BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis

3.1.1 Analisis Masalah

Permasalahan pada penelitian ini adalah masih banyak organisasi dan perusahaan yang masih belum mengoptimalkan penggunan media teknologi dan informasi dengan media internet atau *online*, dan meskipun sudah ada yang mengunakan fasilitas *online* hal tersebut masih dirasa kurang efektif karena pembuatan antara aplikasi server dengan aplikasi klien harus menggunakan bahasa pemrograman yang sama sehingga dalam pembuatan aplikasinya terbatas dalam bahasa pemrograman tertentu. Selain itu, bagaimana mengimplementasikan *web service* dengan aplikasi web dan android sebagai solusi permasalahan tersebut.

3.1.2 Analisis Kebutuhan Data

Sistem yang akan dibangun merupakan sistem untuk menginformasikan lowongan pekerjaan yang tersedia, profil perusahaan yang menyediakan pekerjaan, serta profil pencari kerja. Untuk aplikasi *back end* yang dikelola oleh admin dapat melakukan proses menyetujui publikasi atau penerbitan lowongan pekerjaan,edit,dan delete dan untuk *client* pada aplikasi web dan android dapat

melakukan proses pemanggilan method atau fungsi yang ada pada *web service* untuk mendapatkan data lowongan pekerjaan, profil perusahaan, dan profil pencari kerja. Maka *web service* harus menyediakan data lowongan pekerjaan, profil perusahaan, dan profil pencari kerja agar aplikasi web dan android dapat mengakses data tersebut. Sehingga untuk dibangunnya sistem ini, dibutuhkan data berupa informasi lowongan pekerjaan yang dikelola oleh admin dan berasal dari perusahaan yang mendaftarkan diri ke sistem lowongan pekerjaan ini.

3.1.3 Analisis Kebutuhan Non Fungsional

Kebutuhan non fungsional yang dibutuhkan untuk membangun sistem ini terdiri dari tiga hal, yaitu : kebutuhan perangkat keras, kebutuhan perangkat lunak dan spesifikasi kebutuhan perangkat lunak (SKPL).

3.1.3.1 Analisis Pengguna

Sistem yang akan dibangun ini digunakan oleh tiga jenis pengguna utama yaitu admin, pengunjung umum dan pengunjung terdaftar. Admin dapat melakukan pengelolaan data terhadap aplikasi ini seperti menyetujui untuk menerbitkan atau publikasi lowongan dari member pemberi kerja atau perusahaan, hapus lowongan, edit lowongan dan dapat melakukan hapus data pencari dan pemberi kerja. Pengunjung umum hanya dapat melihat informasi lowongan kerja dan melakukan registrasi menjadi member atau pengunjung terdaftar (pemberi kerja atau pencari kerja). Pengunjung terdaftar terbagi menjadi 2 yaitu member pemberi kerja (Pmk) yang dapat membuat profil perusahaan ketika melakukan registrasi dan juga menambah lowongan kerja baru. Sedangkan member pencari

kerja (Pck) dapat melihat informasi lowongan kerja dan informasi perusahaan, juga membuat profil pribadi ketika melakukan registrasi sebagai member atau pengunjung terdaftar dan melakukan lamaran pekerjaan.

Tabel III. 1 Karakteristik Pengguna

Pengguna	Deskripsi	Usia	Hak Akses	Tingkat Pendidikan	Pengalaman
Administrator Sistem (Staff pengelola data lowongan pekerjaan)	Orang yang mengelola web portal secara keseluruhan, diberikan hak akses penuh untuk melakukan pemeliharaan terhadap web portal	Lebih dari 21 Tahun	Melakukan pengolahan data lowongan pekerjaan, pengolahan data master, lihat data member	Minimal D3 dan dapat mengakses internet	Pernah mengelola web sejenis dan mengerti teknologi *website*
Pemberi Kerja (Perusahaan)	Adalah user yang melakukan registrasi sebagai member perusahaan untuk dapat melakukan input lowongan kerja dan merespon lamaran pekerjaan	Lebih dari 21 Tahun	Membuat profil perusahaan ketika registrasi, input lowongan kerja baru, mengelola data lowongan, melihat daftar pencari kerja, dan merespon lamaran pekerjaan	-	Mengakses internet

Tabel III. 2 Karakteristik Pengguna (Lanjutan)

Pengguna	Deskripsi	Usia	Hak Akses	Tingkat Pendidikan	Pengalaman
Pencari Kerja	Adalah user yang melakukan registrasi sebagai pencari kerja dan dapat melakukan lamaran terhadap lowongan pekerjaan	Lebih dari 17 Tahun	Melihat informasi lowongan pekerjaan, melihat daftar pemberi kerja, mengubah data profil pribadi, dapat melakukan lamaran, dan menerima info lowongan baru sesuai minat kerja melalui email	-	Mengakses internet
Pengunjung Umum	Adalah user yang mengunjungi web portal, dan belum melakukan registrasi untuk menjadi member	Lebih dari 17 Tahun	Melihat informasi lowongan kerja, dapat melakukan pendaftaran sebagai member	-	Mengakses internet

3.1.3.2 Analisis Kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam membangun sistem ini adalah :

1. Processor Intel Core 2 Duo 2.2 GHz
2. Kapasitas RAM 4 GB
3. VGA *Card* 512 MB
4. Kapasitas Hardisk 160 GB
5. Monitor
6. *Lan Card* 10/100 Mbps
7. Koneksi internet 256 Kbps
8. *Keyboard & Mouse*

Spesifikasi kebutuhan perangkat keras (*hardware*) pada mobile android yang digunakan untuk menjalankan aplikasi mobile lowongan pekerjaan ini adalah :

1. Processor 800 MHz
2. Kapasitas RAM 256 MB
3. Memori internal 158 MB
4. Ukuran layar 3,2 inch dengan resolusi HVGA (320x480)
5. Baterai Li-Ion 1350 mAh

Spesifikasi kebutuhan minimal perangkat keras yang digunakan untuk membangun sistem ini adalah :

1. Processor dengan kecepatan minimal 2 GHz
2. RAM minimal 256 MB
3. VGA *Card* minimal 64 MB

4. Kapasitas Harddisk minimal 40 GB
5. Monitor
6. *Lan Card 10/100Mbps*
7. Koneksi *internet* minimal 64 kbps
8. *Keyboard & Mouse*

3.1.3.3 Analisis Kebutuhan Perangkat Lunak

Perangakat lunak yang diperlukan untuk membangun sistem ini adalah :

1. Java Development Kit versi 5 atau 6 , untuk kompilasi kode – kode program.
2. Java Runtime Edition, sebagai *platform* untuk menjalankan sistem.
3. Netbeans IDE, sebagai perangkat lunak yang digunakan untuk membangun *web service*.
4. Eclipse 3.5, sebagai perangkat lunak yang digunakan untuk membangun aplikasi *client* pada android.
5. Adobe Dreamweaver CS5 sebagai perangkat lunak yang digunakan untuk membangun aplikasi web client.
6. Glassfish, sebagai *webserver* untuk aplikasi web service.
7. Apache, sebagai webserver untuk aplikasi web.
8. Android SDK, untuk pengembang aplikasi android.
9. Android Development Tool, sebagai plugin android pada eclipse.
10. Android Virtual *Device*, sebagai emulator untuk menjalankan sistem android.

3.1.3.4 Spesifikasi Kebutuhan Perangkat Lunak (SKPL)

Pembangunan Sistem Lowongan Pekerjaan Dengan Dukungan *Web Service* ini berfungsi sebagai media informasi lowongan kerja dan sarana melamar kerja. Semua itu akan dilakukan secara *online*. Fungsi utama perangkat lunak yang dibangun adalah :

1. Registrasi Member

   Proses registrasi member untuk pengunjung yang ingin bergabung menjadi member aktif di aplikasi Lowongan Pekerjaan dengan menginputkan data profil member lengkap yang akan disimpan ke dalam *database*.

2. Login

   Proses sebelum masuk ke dalam sistem menu utama admin dan member. *User* harus menginputkan data login member atau data login admin agar masuk kedalam sistem. Kemudian sistem akan mengautentikasi *username* dan *password user*. Jika login valid maka *user* dapat mengakses sistem, jika login data tidak valid maka sistem akan memberikan info kepada *user* bahwa *username* atau *password* tidak ditemukan.

3. Lupa *Password*

   Fasilitas lupa *password* untuk mengantisipasi jika *user* admin dan member lupa *password* untuk login ke dalam sistem. *User* hanya menginputkan alamat email ke dalam sistem dan sistem akan memberikan informasi data *password user* melalui alamat email.

4. Logout

   Suatu proses admin dan member untuk keluar dari sistem yang telah digunakan dan jika ingin mengakses sistem maka harus login kembali agar bisa masuk ke sistem.

5. Pencarian Lowongan Pekerjaan

   Sistem menyediakan fungsionalitas pencarian lowongan pekerjaan yang dapat digunakan oleh pengunjung umum dan member. Setelah user menginputkan data cari yang akan dicari, sistem akan memberikan respon apakah data ditemukan atau tidak.

6. Lihat Lowongan Pekerjaan

   Sistem menyediakan fungsionalitas untuk melihat lowongan pekerjaan. Fasilitas ini berfungsi sebagai informasi adanya lowongan pekerjaan dari berbagai perusahaan.

7. *Filter* Lowongan Pekerjaan

   Sistem menyediakan fungsionalitas pada aplikasi *client mobile* android untuk filter lowongan pekerjaan. Fasilitas ini berfungsi untuk menyaring daftar lowongan pekerjaan yang ditampilkan berdasarkan lokasi penempatan, kategori, gaji, dan tipe kerja.

8. Lihat Peta

   Sistem menyediakan fungsionalitas pada untuk melihat peta. Fasilitas ini berfungsi untuk melihat lokasi penempatan dari lowongan pekerjaan pada peta digital.

9. Membagikan Info Lowongan

   Sistem menyediakan fungsionalitas pada aplikasi *client mobile* android untuk membagikan info lowongan pekerjaan. Fasilitas ini berfungsi untuk membagikan info lowongan pekerjaan melalui aplikasi *social networking* atau email yang terdapat pada *device user*.

10. Pengolahan Data Lowongan Kerja

    Sistem menyediakan fungsionalitas proses pengolahan data lowongan kerja yang dikelola oleh admin yakni menyetujui untuk menerbitkan lowongan, dan hapus data lowongan pekerjaan. Sedangkan untuk member pemberi kerja dapat melakukan tambah lowongan baru, edit, dan hapus data lowongan pekerjaan.

11. Pengolahan Data Member

    Sistem menyediakan fungsionalitas proses pengolahan data member dimana admin dapat melihat detil dari data member dan dapat menghapus data member.

12. Pengolahan Data *User*

    Proses pengolahan data *user* atau member untuk mengubah profil pribadi atau profil perusahaan, kemudian disimpan di *database*.

13. *Update* CV

    Sistem menyediakan fungsionalitas untuk meng-*upload* file CV baru bagi member pencari kerja ke dalam database.

14. Pengolahan Data Lamaran Kerja

    Sistem menyediakan fungsionalitas proses pengolahan data lamaran pekerjaan yang masuk bagi member pemberi kerja (perusahaan). Member pemberi kerja (perusahaan) dapat melakukan melihat daftar lamaran yang masuk dan melakukan aksi untuk melihat CV pelamar, menolak, dan menerima lamaran pekerjaan. Member pemberi kerja juga dapat mengirim surat panggilan kepada pelamar jika member perusahaan menerimaa lamaran pekerjaan dari pelamar.

15. Pengolahan Data Master

    Sistem menyediakan fungsionalitas untuk pengolahan data master yang dilakukan admin. Pengolahan data master fasilitas yang berguna bagi admin untuk melakukan penambahan kategori dan admin baru lowongan pekerjaan.

16. *History* Lamaran

    Sistem menyediakan fungsionalitas untuk melihat data *history* lamaran. Fasilitas ini berguna bagi member pencari kerja untuk melihat daftar dari lamaran yang pernah dilakukan dan melihat respon dari perusahaan dari lowongan pekerjaan yang dilamar.

17. Melihat Data Perusahaan

    Sistem menyediakan fungsionalitas untuk melihat data perusahaan. Fasilitas ini berguna untuk membantu *user* untuk melihat data detil dari perusahaan seperti deskripsi perusahaan, alamat, dan kontak perusahaan yang dapat dihubungi oleh *user*.

18. Melihat Data Pencari Kerja

    Sistem menyediakan fungsionalitas untuk melihat data pencari kerja. Fasilitas ini berguna untuk membantu *user* terutama member pemberi kerja untuk melihat spesifikasi dari pencari kerja yang melakukan lamaran ke lowongan perusahaan mereka.

19. Melihat Profil

    Sistem menyediakan fungsionalitas untuk melihat profil member. Fasilitas ini berguna untuk melihat data profil pribadi bagi member yang melakukan login ke aplikasi lowongan pekerjaan.

3.1.4 Analisis Sistem

3.1.4.1 Arsitektur Sistem

Sistem yang akan dibangun mempunyai 2 bagian yaitu di sisi server dan sisi klien. Sisi server akan menyediakan fungsi-fungsi yang terdapat pada *web service*. Sedangkan pada sisi klien terdapat tiga aplikasi yaitu dua aplikasi berbasis web dan satu aplikasi berbasis mobile pada *platform* android yang dapat digunakan oleh *user* melalui internet. Aplikasi web pertama merupakan sistem *back end* yang digunakan *user* administrator untuk mengelola data lowongan pekerjaan dan data member. Administrator berhak atas segala akses data yang sudah ada termasuk untuk menyetujui penerbitan lowongan, edit, dan hapus data lowongan pekerjaan serta dapat menghapus data member. Sedangkan aplikasi web yang kedua adalah sistem *front end* yang digunakan *user* untuk memperoleh informasi mengenai lowongan pekerjaan. *User* yang menggunakan aplikasi web

pada sistem *front end* ini dapat dibedakan menjadi dua jenis, yaitu : pengunjung umum dan member. Kemudian aplikasi klien yang terakhir yang merupakan sistem *front end* yang berbasis mobile pada platform android juga memiliki dua *user* yaitu pengunjung umum dan pengunjung terdaftar atau member.

Selain aplikasi utama pada sisi klien tadi, sistem lowongan pekerjaan juga mempunyai dukungan *web servive* dalam bentuk *client – server*. User yang akan menggunakan dukungan *web service* harus menyediakan program *client*,yang dapat dibuat dengan bantuan WSDL yang telah disediakan. *Client* ini selanjutnya dapat mengakses fungsi-fungsi *web service* yang disediakan oleh *server*. Pada penelitian ini program *client* akan dibuat pada aplikasi web dan mobile pada platform android.

3.1.4.2 Analisis Standar *Web service*

Dalam bab landasan teori telah dijelaskan secara singkat ada dua jenis *web service* yang dapat digunakan, yaitu REST dan SOAP. Tugas akhir ini akan menggunakan *web service* dengan standar SOAP karena pertama, SOAP adalah jenis *web service* yang berorientasi pada *service* yang cocok diterapkan pada aplikasi lowongan pekerjaan yang memang menyediakan layanan untuk memanggil metode yang mengolah *resource* data lowongan, selain itu SOAP memiliki standar yang baik dan jelas. REST tidak memiliki standar, tetapi hanya sebuah arsitektur *web service*. Untuk aplikasi dimana ia mungkin saja berkomunikasi dengan banyak aplikasi lain yang berbeda bahasa maupun *platform*, standar sangat diperlukan karena menjamin aplikasi-aplikasi ini berbicara dalam bahasa yang sama dan memudahkan pengembangan. Penjabaran

SOAP diperjelas dengan disediakannya WSDL, sehingga pengguna *web service* dapat dengan mudah membuat *client* untuk menggunakan SOAP.

1. Analisis Method Web Service

Bagian ini adalah bagian yang paling utama dari fungsi web service. Bagian ini menjelaskan secara rinci fungsi atau method-method apa saja yang tersedia dalam Web Service, parameter apa saja yang diperlukan untuk memanggil sebuah method, dan apa hasil atau tipe data yang dikembalikan oleh method yang dipanggil tersebut.

1) Method *loginPck*

Hal pertama yang harus dilakukan oleh member sebelum melakukan aktivitas dengan aplikasi Lowongan Pekerjaan.

Method *loginPck* merupakan method yang digunakan aplikasi Lowongan Pekerjaan untuk mengambil data login pada basis data Lowongan Pekerjaan sebagai member pencari kerja. Method *loginPck* memiliki beberapa variabel yang dikirimkan sebagai parameter pada saat melakukan *login* ke dalam aplikasi Lowongan Pekerjaan baik aplikasi web maupun mobile. Berikut ini variabel-variabel yang dibutuhkan oleh method *loginPck* pada saat melakukan *login* ke dalam Lowongan Pekerjaan seperti terlihat pada tabel III.3 dan juga prosedurnya pada tabel III.4.

Tabel III. 3 Variabel Method loginPck

Nama Variabel	Isi Variabel	Tipe Variabel
username	*Username* dari pengguna yang akan melakukan login.	GET
password	*Password* dari pengguna yang akan melakukan *login*.	GET

Tabel III. 4 Prosedur Method loginPck

Method *loginPck* (input *username:string*, input *password:string*)	
Masukan	*Username* & *password*
Keluaran	*Array* lowongan pekerjaan dalam bentuk JSON
Skenario Utama Method *loginPck*	
Proses	1. Mengambil *username* & *password* yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan data member pencari kerja sesuai *username* & *password* yang dimasukkan ke dalam *array*.
	4. Mengembalikan data member pencari kerja ke dalam *array*.
Skenario Alternatif Method *loginPck*	
	4a. Mengembalikan nilai kosong jika member pencari kerja dengan *username* & *password* yang dimasukkan tidak ditemukan.

Hasil dari method *loginPck* yaitu berupa data dengan format JSON yang didapat dari basis data Lowongan Pekerjaan. Data dengan format JSON tersebut nantinya akan diuraikan (*parsing*) oleh aplikasi Lowongan Pekerjaan.

Rancangan input dan output dari method *loginPck* dapat kita lihat pada struktur dari dokumen SOAP request dan SOAP response berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:loginPck xmlns:ns2="http://webservice/">
            <username>username_pencarikerja</username>
            <password>password_pencarikerja</password>
        </ns2:loginPck>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
  <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
     <S:Body>
        <ns2:loginPckResponse xmlns:ns2="http://webservice/">
            <return>[
                      {"id_user": data_id_user_pencarikerja,
                       "username":"data_username_pencarikerja",
                      "nama":"data_pencarikerja",
                      "jk":"data_jeniskelamin_pencarikerja",
                      "kota":"data_kota_pencarikerja",
                      "email":"data_email_pencarikerja",
                      "gambar":"path_gambar_pencarikerja"}]
              </return>
          </ns2:loginPckResponse>
      </S:Body>
  </S:Envelope>
```

2) Method *loginPmk*

Method *loginPmk* sama halnya dengan method *loginPck*, tapi method *loginPmk* digunakan untuk proses *login* sebagai member pemberi kerja atau perusahaan. Method *loginPmk* memiliki beberapa variabel yang dikirimkan sebagai parameter pada saat melakukan *login* ke dalam aplikasi Lowongan Pekerjaan baik aplikasi web maupun mobile. Berikut ini variabel-variabel yang

dibutuhkan oleh method *loginPmk* pada saat melakukan *login* ke dalam Lowongan Pekerjaan seperti terlihat pada tabel III.5 dan juga prosedurnya pada tabel III.6.

Tabel III. 5 Variabel Method loginPmk

Nama Variabel	Isi Variabel	Tipe Variabel
username	*Username* dari pengguna yang akan melakukan login.	GET
password	*Password* dari pengguna yang akan melakukan *login*.	GET

Tabel III. 6 Prosedur Method loginPmk

Method *loginPmk* (input *username:string*, input *password:string*)	
Masukan	*Username* & *password*
Keluaran	*Array* lowongan pekerjaan dalam bentuk JSON
Skenario Utama Method *loginPmk*	
Proses	1. Mengambil *username* & *password* yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan data member pemberi kerja sesuai *username* & *password* yang dimasukkan ke dalam *array*.
	4. Mengembalikan data member pemberi kerja ke dalam *array*.
Skenario Alternatif Method *loginPmk*	
	4a. Mengembalikan nilai kosong jika member pencari kerja dengan *username* & *password* yang dimasukkan tidak ditemukan.

Rancangan *input* dan *output* dari method *loginPmk* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:loginPmk xmlns:ns2="http://webservice/">
            <username>username_pemberikerja</username>
            <password>password_pemberikerja</password>
        </ns2:loginPmk>
    </S:Body>
    </S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:loginPmkResponse xmlns:ns2="http://webservice/">
            <return>[
                    {"id_user":data_id_user_pemberikerja,
                     "username":"data_username_pemberikerja",
                     "nama_perusahaan":"data_nama_perusahaan",
                     "kota":"data_kota_pemberikerja",
                     "provinsi":"data_provinsi_pemberikerja",
                     "email":"data_email_pemberikerja",
                     "gambar":"path_gambar_pemberikerja"}]
             </return>
        </ns2:loginPmkResponse>
    </S:Body>
</S:Envelope>
```

3) Method *getJob*

Method *getJob* merupakan method atau fungsi yang digunakan oleh aplikasi Lowongan Pekerjaan untuk mendapatkan semua data lowongan pekerjaan dari basis data lowongan pekerjaan. Method *getJob* tidak memiliki variabel yang menjadi parameter input. Sedangkan prosedur method *getJob* dapat dilihat pada tabel III.7.

Tabel III. 7 Prosedur Method getJob

Method *getJob* ()	
Masukan	-
Keluaran	*Array* lowongan pekerjaan dalam bentuk JSON
Skenario Utama Method *getJob*	
Proses	1. Menerima *request* untuk mengeksekusi method *getJob.*
	2. Menjalankan query sql.
	3. Memasukkan lowongan pekerjaan ke dalam *array.*
	4. Mengembalikan lowongan pekerjaan ke dalam *array.*
Skenario Alternatif Method *getJob*	
	4a. Mengembalikan nilai kosong jika tidak ditemukan lowongan pekerjaan

Rancangan *input* dan *output* dari method *getJob* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:getJob xmlns:ns2="http://webservice/"/>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
   <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
      <S:Body>
        <ns2:getJobResponse xmlns:ns2="http://webservice/">
             <return>[
                    {"idLowongan":data_id_lowongan,
                     "idKategori":data_id_kategori,
```

```
                    "id_user" data_id_user,
                    "idTipe":"data_id_tipekerja",
                    "judul_lowongan":"data_judul_lowongan",
                    "kategori":"data_kategori",
                    "namaperusahaan":"data_namaperusahaan",
                    "jobexpired":" data_tanggalberakhir_lowongan",
                    "tipekerja":"data_tipekerja_lowongan",
                    "kontak_perusahaan":"data_kontak_perusahaan",
                    "lokasiPenempatan":"data_lokasipenempatan",
                    "waktuInput":"data_waktuinput_lowongan"}]
                </return>
            </ns2:getJobResponse>
        </S:Body>
    </S:Envelope>
```

4) Method *getJobByID*

Method *getJobByID* merupakan method atau fungsi yang digunakan oleh aplikasi Lowongan Pekerjaan untuk mendapatkan detail dari lowongan pekerjaan berdasarkan Id dari lowongan pekerjaan dari basis data lowongan pekerjaan. Method *getJobByID* memiliki satu variabel sebagai parameter input yang digunakan untuk mengambil detail lowongan pekerjaan berdasarkan id lowongan pekerjaan dari basis data lowongan pekerjaan.

Berikut ini variabel yang dibutuhkan oleh method *getJobByID* pada saat akan mengambil detail lowongan pekerjaan berdasarkan id lowongan pekerjaan seperti terlihat pada tabel III.8 dan juga prosedurnya pada tabel III.9.

Tabel III. 8 Variabel Method getJobByID

Nama Variabel	Isi Variabel	Tipe Variabel
idLowongan	Id lowongan dari lowongan pekerjaan yang akan diambil detail dari lowongan tersebut	GET

Tabel III. 9 Prosedur Method getJobByID

Method *getJobByID* (input idLowongan:string)	
Masukan	Id Lowongan
Keluaran	Array lowongan pekerjaan dengan id tertentu dalam bentuk JSON
Skenario Utama Method *getJobByID*	
Proses	1. Mengambil ID Lowongan Pekerjaan yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan lowongan pekerjaan dengan ID tertentu ke dalam *array*.
	4. Mengembalikan lowongan pekerjaan berdasarkan ID tertentu ke dalam *array*.
Skenario Utama Method *getJobByID*	
	4a. Mengembalikan nilai kosong jika lowongan pekerjaan dengan id lowongan yang dimasukkan tidak ditemukan.

Rancangan *input* dan *output* dari method *getJobById* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:getJobByID xmlns:ns2="http://webservice/">
            <idLowongan>id_lowongan_yang_dicari</idLowongan>
        </ns2:getJobByID>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
  <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
      <S:Body>
```

```
<ns2:getJobByIDResponse xmlns:ns2="http://webservice/">
      <return>[
            {"idLowongan":data_id_lowongan,
             "idKategori":data_id_kategori,
             "id_user"data_id_user,
             "idTipe":"data_id_tipekerja",
             "judul_lowongan":"data_judul_lowongan",
             "kategori":"data_kategori",
             "namaperusahaan":"data_namaperusahaan",
             "jobexpired":"data_tanggalberakhir_lowongan",
             "tipekerja":"data_tipekerja_lowongan",
             "gaji":"data_gaji",
             "lokasiPenempatan":"data_lokasi_penempatan",
             "lat":"data_latitude_lokasi_penempatan",
            "longi":"data_longitude_lokasi_penempatan ",
             "waktuInput":"data_waktuinput_lowongan",
             "email":"data_email_perusahaan"}]
       </return>
    </ns2:getJobByIDResponse>
  </S:Body>
</S:Envelope>
```

5) Method *getCategory*

Method *getCategory* merupakan method atau fungsi yang digunakan oleh aplikasi Lowongan Pekerjaan untuk mendapatkan semua kategori dari lowongan pekerjaan dari basis data lowongan pekerjaan. Method *getJobById* tidak memiliki variabel yang menjadi parameter input. Sedangkan prosedur method getCategory ini dapat dilihat pada tabel III.10.

Tabel III. 10 Prosedur Method getCategory

Method *getCategory* ()	
Masukan	-
Keluaran	*Array* kategori lowongan pekerjaan dalam bentuk JSON
Skenario Utama Method *getCategory*	
Proses	1. Menerima *request* untuk mengeksekusi method *getCategory*.
	2. Menjalankan query sql.
	3. Memasukkan kategori lowongan pekerjaan ke dalam *array*.
	4. Mengembalikan kategori lowongan pekerjaan ke dalam *array*.
Skenario Alternatif Method *getCategory*	
	4a. Mengembalikan nilai kosong jika belum terdapat data ketegori lowongan pekerjaan

Rancangan *input* dan *output* dari method *getCategory* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:getCategory xmlns:ns2="http://webservice/"/>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:getCategoryResponse xmlns:ns2="http://webservice/">
```

```
            <return>[
                    {"idKategori":data_id_kategori,
                    "kategori":"data_kategori"}]
             </return>
        </ns2:getCategoryResponse>
    </S:Body>
</S:Envelope>
```

6) Method *getJobByCategoryName*

Method *getJobByCategoryName* merupakan method atau fungsi yang digunakan oleh aplikasi Lowongan Pekerjaan untuk memfilter lowongan pekerjaan berdasarkan kategori lowongan pekerjaan yang diambil dari basis data lowongan pekerjaan. Method *getJobByCategoryName* memiliki satu variabel sebagai parameter input yang digunakan untuk mengambil data lowongan pekerjaan berdasarkan kategori lowongan pekerjaan dari basis data lowongan pekerjaan.

Berikut ini variabel yang dibutuhkan oleh method *getJobByCategoryName* pada saat akan memfilter lowongan pekerjaan berdasarkan kategori lowongan pekerjaan seperti terlihat pada tabel III.11 dan juga prosedurnya pada tabel III.12.

Tabel III. 11 Variabel Method getJobByCategoryName

Nama Variabel	Isi Variabel	Tipe Variabel
kategori	Kategori dari lowongan pekerjaan yang akan diambil detail dari lowongan tersebut	GET

Tabel III. 12 Prosedur Method getJobByCategoryName

Method *getJobByCategoryName* (input kategori:string)	
Masukan	Kategori lowongan pekerjaan.
Keluaran	*Array* lowongan pekerjaan berdasarkan kategori dalam bentuk JSON.
Skenario Utama Method *getJobByCategoryName*	
Proses	1. Mengambil kategori lowongan yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan lowongan pekerjaan berdasarkan kategori tertentu ke dalam *array*.
	4. Mengembalikan lowongan pekerjaan dengan kategori tertentu ke dalam *array*.
Skenario Alternatif Method *getJobByCategoryName*	
	4a. Mengembalikan nilai kosong jika lowongan dengan kategori yang dimasukkan tidak ditemukan.

Rancangan *input* dan *output* dari method *getJobCategoryName* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:getJobByCategoryName xmlns:ns2="http://webservice/">
            <kategori>nama_kategori_yang_dicari</kategori>
        </ns2:getJobByCategoryName>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
```

```
    <S:Body>
        <ns2:getJobByCategoryNameResponse xmlns:ns2="http://webservice/">
            <return>[
                    {"idLowongan":data_id_lowongan,
                    "idKategori":data_id_kategori,
                    "id_user":data_id_user,
                    "judul_lowongan":"data_judul_lowongan",
                    "kategori":"data_kategori",
                    "namaperusahaan":"data_namaperusahaan",
                    "jobexpired":"data_tanggalberakhir_lowongan",
                    "tipekerja":"data_tipekerja_lowongan",
                    "kontak_perusahaan":"data_kontak_perusahaan",
                    "lokasiPenempatan":"data_lokasi_penempatan",
                    "waktuInput":"data_waktuinput_lowongan"}]</return>
        </ns2:getJobByCategoryNameResponse>
    </S:Body>
</S:Envelope>
```

7) Method *getJobBySalary*

Method *getJobBySalary* merupakan method atau fungsi yang digunakan oleh aplikasi Lowongan Pekerjaan untuk memfilter lowongan pekerjaan berdasarkan gaji lowongan pekerjaan yang diambil dari basis data lowongan pekerjaan. Method *getJobBySalary* memiliki satu variabel sebagai parameter input yang digunakan untuk mengambil data lowongan pekerjaan berdasarkan gaji lowongan pekerjaan dari basis data lowongan pekerjaan.

Berikut ini variabel yang dibutuhkan oleh method *getJobBySalary* pada saat akan memfilter lowongan pekerjaan berdasarkan gaji lowongan pekerjaan seperti terlihat pada tabel III.13 dan juga prosedurnya pada tabel III.14.

Tabel III. 13 Variabel Method getJobBySalary

Nama Variabel	Isi Variabel	Tipe Variabel
gaji	Besar gaji dari lowongan pekerjaan yang akan diambil detail dari lowongan tersebut	GET

Tabel III. 14 Prosedur Method getJobBySalary

Method *getJobBySalary* (input gaji:string)	
Masukan	Besar gaji lowongan pekerjaan.
Keluaran	*Array* lowongan pekerjaan berdasarkan gaji dalam bentuk JSON.
Skenario Utama Method *getJobBySalary*	
Proses	1. Mengambil tipe gaji lowongan yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan lowongan pekerjaan berdasarkan gaji tertentu ke dalam *array*.
	4. Mengembalikan lowongan pekerjaan dengan gaji tertentu ke dalam *array*.
Skenario Alternatif Method *getJobBySalary*	
	4a. Mengembalikan nilai kosong jika lowongan dengan gaji yang dimasukkan tidak ditemukan.

Rancangan *input* dan *output* dari method *getJobBySalary* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:getJobBySalary xmlns:ns2="http://webservice/">
            <gaji>gaji_yang_dicari</gaji>
        </ns2:getJobBySalary>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
```

```
        <ns2:getJobBySalaryResponse xmlns:ns2="http://webservice/">
            <return>[
                    {"idLowongan":data_id_lowongan,
                    "idKategori":data_id_kategori,
                    "id_user":data_id_user,
                    "judul_lowongan":"data_judul_lowongan",
                    "kategori":"data_kategori",
                    "namaperusahaan":"data_namaperusahaan",
                    "jobexpired":"data_tanggalberakhir_lowongan",
                    "tipekerja":"data_tipekerja_lowongan",
                    "gaji":"data_gaji",
                    "lokasiPenempatan":"data_lokasi_penempatan",
                    "waktuInput":"data_waktuinput_lowongan"}]
             </return>
        </ns2:getJobBySalaryResponse>
    </S:Body>
</S:Envelope>
```

8) Method *getJobByJobType*

Method *getJobByJobType* merupakan method atau fungsi yang digunakan oleh aplikasi Lowongan Pekerjaan untuk memfilter lowongan pekerjaan berdasarkan tipe kerja lowongan pekerjaan yang diambil dari basis data lowongan pekerjaan. Method *getJobByJobType* memiliki satu variabel sebagai parameter input yang digunakan untuk mengambil data lowongan pekerjaan berdasarkan tipe kerja lowongan pekerjaan dari basis data lowongan pekerjaan.

Berikut ini variabel yang dibutuhkan oleh method *getJobByJobType* pada saat akan memfilter lowongan pekerjaan berdasarkan tipe kerja lowongan pekerjaan seperti terlihat pada tabel III.15 dan juga prosedurnya pada tabel III.16.

Tabel III. 15 Varabel Method getJobByJobType

Nama Variabel	Isi Variabel	Tipe Variabel
tipekerja	Jenis tipe kerja dari lowongan pekerjaan yang akan diambil detail dari lowongan tersebut	GET

Tabel III. 16 Prosedur Method getJobByJobType

Method *getJobByJobType* (input tipekerja:string)	
Masukan	Tipe kerja lowongan pekerjaan.
Keluaran	*Array* lowongan pekerjaan berdasarkan tipe kerja dalam bentuk JSON.
Skenario Utama Method *getJobByJobType*	
Proses	1. Mengambil tipe kerja lowongan yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan lowongan pekerjaan berdasarkan tipe kerja tertentu ke dalam *array*.
	4. Mengembalikan lowongan pekerjaan dengan tipe kerja tertentu ke dalam *array*.
Skenario Alternatif Method *getJobByJobType*	
	4a. Mengembalikan nilai kosong jika lowongan dengan tipe kerja yang dimasukkan tidak ditemukan.

Rancangan *input* dan *output* dari method *getJobByJobType* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:getJobByJobType xmlns:ns2="http://webservice/">
            <tipekerja>tipe_kerja_yang_dicari</tipekerja>
        </ns2:getJobByJobType>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:getJobByJobTypeResponse xmlns:ns2="http://webservice/">
```

```
            <return>[
                    {"idLowongan":data_id_lowongan,
                     "idKategori":data_id_kategori,
                     "id_user"data_id_user,
                     "idTipe":"data_id_tipekerja",
                     "judul_lowongan":"data_judul_lowongan",
                     "kategori":"data_kategori",
                     "namaperusahaan":"data_namaperusahaan",
                     "jobexpired":"data_tanggalberakhir_lowongan",
                     "tipekerja":"data_tipekerja_lowongan",
                     "lokasiPenempatan":"data_lokasi_penempatan",
                     "waktuInput":"data_waktuinput_lowongan"}]
              </return>
         </ns2:getJobByJobTypeResponse>
    </S:Body>
</S:Envelope>
```

9) Method *getJobByLocation*

Method *getJobByLocation* merupakan method atau fungsi yang digunakan oleh aplikasi Lowongan Pekerjaan untuk memfilter lowongan pekerjaan berdasarkan lokasi dari lowongan pekerjaan yang diambil dari basis data lowongan pekerjaan. Method *getJobByLocation* memiliki satu variabel sebagai parameter input yang digunakan untuk mengambil data lowongan pekerjaan berdasarkan lokasi lowongan pekerjaan dari basis data lowongan pekerjaan.

Berikut ini variabel yang dibutuhkan oleh method *getJobByLocation* pada saat akan memfilter lowongan pekerjaan berdasarkan lokasi lowongan pekerjaan seperti terlihat pada tabel III.17 dan juga prosedurnya pada tabel III.18.

Tabel III. 17 Variabel Method getJobByLocation

Nama Variabel	Isi Variabel	Tipe Variabel
lokasi	Lokasi dari lowongan pekerjaan yang akan diambil detail dari lowongan tersebut	GET

Tabel III. 18 Prosedur Method getJobByLocation

Method *getJobByLocation* (input lokasi:string)	
Masukan	Lokasi dari lowongan pekerjaan.
Keluaran	*Array* lowongan pekerjaan berdasarkan lokasi dalam bentuk JSON.
Skenario Utama Method *getJobByLocation*	
Proses	1. Mengambil lokasi dari lowongan yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan lowongan pekerjaan berdasarkan lokasi tertentu ke dalam *array*.
	4. Mengembalikan lowongan pekerjaan dengan lokasi tertentu ke dalam *array*.
Skenario Alternatif Method *getJobByLocation*	
	4a. Mengembalikan nilai kosong jika lowongan dengan lokasi yang dimasukkan tidak ditemukan.

Rancangan *input* dan *output* dari method *getJobByLocation* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:getJobByLocation xmlns:ns2="http://webservice/">
            <lokasi>lokasi_lowongan_yang_dicari</lokasi>
        </ns2:getJobByLocation>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
```

```
        <ns2:getJobByLocationResponse xmlns:ns2="http://webservice/">
            <return>[
                    {"idLowongan":data_id_lowongan,
                       "idKategori":data_id_kategori,
                       "id_user"data_id_user,
                       "idTipe":"data_id_tipekerja",
                       "judul_lowongan":"data_judul_lowongan",
                       "kategori":"data_kategori",
                       "namaperusahaan":"data_namaperusahaan",
                       "jobexpired":"data_tanggalberakhir_lowongan",
                       "tipekerja":"data_tipekerja_lowongan",
                       "lokasiPenempatan":"data_lokasi_penempatan",
                       "waktuInput":"data_waktuinput_lowongan"}]
             </return>
        </ns2:getJobByLocationResponse>
    </S:Body>
</S:Envelope>
```

10) Method *getJobByCompany*

Method *getJobByCompany* merupakan method atau fungsi yang digunakan oleh aplikasi Lowongan Pekerjaan untuk memfilter lowongan pekerjaan berdasarkan id pemberi kerja (perusahaan) dari lowongan pekerjaan yang diambil dari basis data lowongan pekerjaan. Method *getJobByCompany* memiliki satu variabel sebagai parameter input yang digunakan untuk mengambil data lowongan pekerjaan berdasarkan id pemberi kerja (perusahaan) dari basis data lowongan pekerjaan.

Berikut ini variabel yang dibutuhkan oleh method *getJobByCompany* pada saat akan memfilter lowongan pekerjaan berdasarkan pemberi kerja (perusahaan) seperti terlihat pada tabel III.19 dan juga prosedurnya pada tabel III.20.

Tabel III. 19 Variabel Method getJobByCompany

Nama Variabel	Isi Variabel	Tipe Variabel
id_user	Id user dari pemberi kerja atau perusahaan	GET

Tabel III. 20 Prosedur Method getJobByCompany

Method *getJobByCompany* (input id_user:string)	
Masukan	Id User
Keluaran	Array lowongan pekerjaan dengan id pemberi kerja tertentu dalam bentuk JSON
Skenario Utama Method *getJobByCompany*	
Proses	1. Mengambil Id pemberi kerja yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan lowongan pekerjaan dengan ID pemberi kerja tertentu ke dalam *array*.
	4. Mengembalikan lowongan pekerjaan berdasarkan Id pemberi kerja tertentu ke dalam *array*.
Skenario Alternatif Method *getJobByCompany*	
	4a. Mengembalikan nilai kosong jika lowongan dengan Id pemberi kerja yang dimasukkan tidak ditemukan.

Rancangan *input* dan *output* dari method *getJobByCompany* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:getJobByCompany xmlns:ns2="http://webservice/">
            <id_user>id_user_pemberikerja</id_user>
        </ns2:getJobByCompany>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:getJobByCompanyResponse xmlns:ns2="http://webservice/">
            <return>[
                    {"idLowongan":data_id_lowongan,
                       "idKategori":data_id_kategori,
                       "id_user"data_id_user,
                       "idTipe":"data_id_tipekerja",
                       "judul_lowongan":"data_judul_lowongan",
                       "kategori":"data_kategori",
                       "namaperusahaan":"data_namaperusahaan",
                       "jobexpired":"data_tanggalberakhir_lowongan",
                       "tipekerja":"data_tipekerja_lowongan",
                       "kontak_perusahaan":"data_kontak_perusahaan",
                       "lokasiPenempatan":"data_lokasi_penempatan",
                       "waktuInput":"data_waktuinput_lowongan"}]
             </return>
        </ns2:getJobByCompanyResponse>
    </S:Body>
</S:Envelope>
```

11) Method *searchJob*

Method *searchJob* merupakan method yang digunakan aplikasi Lowongan Pekerjaan untuk melakukan proses pencarian lowongan pekerjaan berdasarkan keyword lowongan pekerjaan dan lokasi lowongan pekerjaan pada basis data Lowongan Pekerjaan. Method *searchJob* memiliki beberapa variabel yang dikirimkan sebagai parameter pada saat melakukan pencarian. Berikut ini variabel-variabel yang dibutuhkan oleh method *searchJob* pada saat melakukan pencarian ke dalam basis data Lowongan Pekerjaan seperti terlihat pada tabel III.21 dan juga prosedurnya pada tabel III.22.

Tabel III. 21 Variabel Method searchJob

Nama Variabel	Isi Variabel	Tipe Variabel
keyword	*Keyword* dari lowongan pekerjaan yang akan dicari ke field judul lowongan, persyaratan, dan nama perusahaan	GET
lokasi	Lokasi lowongan pekerjaan	GET

Tabel III. 22 Prosedur Method searchJob

Method *searchJob* (input *keyword*:string, input lokasi:string)	
Masukan	*Keyword* lowongan dan lokasi lowongan
Keluaran	*Array* lowongan pekerjaan dalam bentuk JSON
Skenario Utama Method *searchJob*	
Proses	1. Mengambil *keyword* lowongan dan lokasi yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan lowongan pekerjaan berdasarkan parameter pencarian ke dalam *array*.
	4. Mengembalikan lowongan pekerjaan ke dalam *array*.
Skenario Alternatif Method *searchJob*	
	4a. Mengembalikan nilai kosong jika lowongan dengan *keyword* lowongan dan lokasi yang dimasukkan tidak ditemukan.

Rancangan *input* dan *output* dari method *searchJob* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

```
SOAP Request
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
```

```
    <S:Header/>
    <S:Body>
        <ns2:searchJob xmlns:ns2="http://webservice/">
            <keyword>judul_lowongan,persyaratan,nama_perusahaan</keyword>
           <lokasi>lokasi_penempatan_yang_dicari</lokasi>
        </ns2:searchJob>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:searchJobResponse xmlns:ns2="http://webservice/">
            <return>[
                    {"idLowongan":data_id_lowongan,
                       "idKategori":data_id_kategori,
                       "id_user"data_id_user,
                       "judul_lowongan":"data_judul_lowongan",
                       "kategori":"data_kategori",
                       "namaperusahaan":"data_namaperusahaan",
                       "jobexpired":"data_tanggalberakhir_lowongan",
                       "tipekerja":"data_tipekerja_lowongan",
                       "lokasiPenempatan":"data_lokasi_penempatan",
                       "waktuInput":"data_waktuinput_lowongan"}]
                    </return>
        </ns2:searchJobResponse>
    </S:Body>
</S:Envelope>
```

12) Method *registerPmk*

Method *registerPmk* merupakan method yang digunakan aplikasi web Lowongan Pekerjaan untuk melakukan proses registrasi sebagai member pemberi kerja (perusahaan). Method *registerPmk* memiliki beberapa variabel yang dikirimkan sebagai parameter pada saat melakukan penambahan data pemberi kerja. Berikut ini variabel-variabel yang dibutuhkan oleh method *registerPmk* pada saat melakukan penambahan data pemberi kerja ke dalam basis data Lowongan Pekerjaan seperti terlihat pada tabel III.23 dan juga prosedurnya pada tabel III.24.

Tabel III. 23 Variabel Method registerPmk

Nama Variabel	Isi Variabel	Default	Tipe Variabel
id_user	1…N ID dari pemberi kerja dan bersifat auto increment	NULL	GET
username	Username dari pemberi kerja	Kosong	GET
password	Password dari pemberi kerja	Kosong	GET
nama_perusahaan	Nama perusahaan dari pemberi kerja	Kosong	GET
kontak_perusahaan	Kontak perusahaan atau no telepon dari perusahaan pemberi kerja	Kosong	GET
email	Alamat email dari perusahaan pemberi kerja	Kosong	GET
alamat	Alamat dari perusahaan pemberi kerja	Kosong	GET
kota	Kota dari perusahaan pemberi kerja	Kosong	GET
provinsi	Provinsi dari perusahaan pemberi kerja	Kosong	GET
kode_pos	Kode pos perusahaan pemberi kerja	Kosong	GET
deskripsi_perusahaan	Deskripsi atau penjelasan singkat dari perusahaan pemberi kerja	Kosong	GET
gambar	Gambar atau logo dari perusahaan pemberi kerja	Kosong	GET

Tabel III. 24 Prosedur Method registerPmk

Method *registerPmk* (input idPemberi:string, input *username*:string, input *password*:string, input nama_perusahaan:string, input kontak_perusahaan:string, input alamat:string, input kota:string, input provinsi:string, input email:string, input kode_pos:string, input deskripsi_perusahaan:string, input gambar:string,)	
Masukan	ID pemberi kerja, *username*, *password*, nama perusahaan, kontak perusahaan, alamat, kota,provinsi, email, kode pos, bidang kerja, deskripsi perusahaan, gambar
Keluaran	Pesan pemberitahuan bahwa data telah berhasil dikirim
Skenario Utama Method *registerPmk*	
Proses	1. Mengambil parameter registrasi pemberi kerja yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan data ke dalam basis data lowongan pekerjaan.
	4. Mengembalikan pesan bahwa data telah berhasil dikirim.
Skenario Alternatif Method *registerPmk*	
	3a. Mengembalikan pesan error jika terdapat salah satu kolom ada yang kosong atau tidak diisi.

Rancangan *input* dan *output* dari method *registerPmk* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
      <ns2:registerPmk xmlns:ns2="http://webservice/">
         <username>username_pemberikerja</username>
```

```
            <password>password_pemberikerja </password>
            <nama_perusahaan>namaperusahaan_pemberikerja </nama_perusahaan>
            <kontak_perusahaan>kontak_perusahaan</kontak_perusahaan>
            <email>alamat_email_pemberikerja </email>
            <alamat>alamat_ pemberikerja </alamat>
            <kota>kota_ pemberikerja</kota>
            <provinsi>provinsi_ pemberikerja</provinsi>
            <kode_pos>kode_pos_ pemberikerja</kode_pos>
            <deskripsi_perusahaan>deskripsi_perusahaan_ pemberikerja
           </deskripsi_perusahaan>
            <gambar>path_gambar_logoperusahaan</gambar>
        </ns2:registerPmk>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:registerPmkResponse xmlns:ns2="http://webservice/">
            <return>Data berhasil Dikirim</return>
        </ns2:registerPmkResponse>
    </S:Body>
</S:Envelope>
```

13) Method *registerPck*

Method *registerPck* merupakan method yang digunakan aplikasi web Lowongan Pekerjaan untuk melakukan proses registrasi sebagai member pencari kerja. Method *registerPck* memiliki beberapa variabel yang dikirimkan sebagai parameter pada saat melakukan penambahan data pencari kerja. Berikut ini variabel-variabel yang dibutuhkan oleh method *registerPck* pada saat melakukan

penambahan data pencari kerja ke dalam basis data Lowongan Pekerjaan seperti terlihat pada tabel III.25 dan juga prosedurnya pada tabel III.26.

Tabel III. 25 Variabel Method registerPck

Nama Variabel	Isi Variabel	Default	Tipe Variabel
id_user	1…N ID dari pencari kerja dan bersifat auto increment	NULL	GET
username	Username dari pemberi kerja	Kosong	GET
password	Password dari pemberi kerja	Kosong	GET
nama	Nama lengkap dari pencari kerja	Kosong	GET
tgl_lahir	Tanggal lahir dari pencari kerja	Kosong	GET
jk	Jenis kelamin dari pencari kerja	Kosong	GET
alamat	Alamat dari pencari kerja	Kosong	GET
kota	Kota dari pencari kerja	Kosong	GET
provinsi	Provinsi dari pencari kerja	Kosong	GET
no_telepon	No telepon dari pencari kerja	Kosong	GET
email	Alamat email dari pencari kerja	Kosong	GET
spesifikasi	Spesifikasi diri dari pencari kerja	Kosong	GET
pengalaman	Berisi pengalaman yang pernah didapat oleh pencari kerja	Kosong	GET
gambar	Gambar atau logo dari perusahaan pemberi kerja	Kosong	GET
cv	Path cv yang diupload ke situs lowongan pekerjaan	Kosong	GET

Tabel III. 26 Prosedur Method registerPck

Method *registerPmk* (input idPencari:string, input *username*:string, input *password*:string, input nama:string, input tgl_lahir:string, input jk:string, input alamat:string, input provinsi:string, input kota:string, input no_telepon:string, input email:string, input spesifikasi:string, input bidangminat:string, input pengalaman:string, input gambar:string,)	
Masukan	ID pencari kerja, *username*, *password*, nama, tanggal lahir, jenis kelamin, alamat, kota,provinsi, no telepon, email, spesifikasi, pengalaman, gambar,cv
Keluaran	Pesan pemberitahuan bahwa data telah berhasil dikirim
Skenario Utama Method *registerPck*	
Proses	1. Mengambil parameter registrasi pencari kerja yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan data ke dalam basis data lowongan pekerjaan.
	4. Mengembalikan pesan bahwa data telah berhasil dikirim.
Skenario Alternatif Method *registerPck*	
	3a. Mengembalikan pesan error jika terdapat salah satu kolom ada yang kosong atau tidak diisi.

Rancangan *input* dan *output* dari method *registerPck* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:registerPck xmlns:ns2="http://webservice/">
            <username>username_pencarikerja</username>
            <password>password_pencarikerja</password>
```

```
            <nama>nama_pencarikerja</nama>
            <tgl_lahir>tgl_lahir_pencarikerja</tgl_lahir>
            <jk>jeniskelamin_pencarikerja</jk>
            <alamat>alamat_pencarikerja</alamat>
            <kota>kota_pencarikerja</kota>
            <provinsi>provinsi_pencarikerja</provinsi>
            <no_telepon>no_telepon_pencarikerja</no_telepon>
            <email>alamat_email_pencarikerja</email>
            <spesifikasi>spesifikasi_pencarikerja</spesifikasi>
            <pengalaman>pengalaman_pencarikerja</pengalaman>
            <gambar>path_gambar_pencarikerja</gambar>
            <cv>path_file_cv_pencarikerja</cv>
        </ns2:registerPck>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:registerPckResponse xmlns:ns2="http://webservice/">
            <return>Data berhasil Dikirim</return>
        </ns2:registerPckResponse>
    </S:Body>
    </S:Envelope>
```

14) Method *getPck*

Method *getPck* merupakan method atau fungsi yang digunakan oleh aplikasi Lowongan Pekerjaan untuk mendapatkan semua data pencari kerja dari basis data lowongan pekerjaan. Method *getPck* tidak memiliki variabel yang menjadi parameter input. Sedangkan prosedur method *getPck* dapat dilihat pada tabel III.27.

Tabel III. 27 Variabel Method getPck

Method *getPck* ()	
Masukan	-
Keluaran	*Array* pencari kerja dalam bentuk JSON
Skenario Utama Method *getPck*	
Proses	1. Menerima *request* untuk mengeksekusi method *getPck.*
	2. Menjalankan query sql.
	3. Memasukkan pencari kerja ke dalam *array.*
	4. Mengembalikan pencari kerja ke dalam *array.*
Skenario Alternatif Method *getPck*	
	4a. Mengembalikan nilai kosong jika data pencari kerja tidak ditemukan atau masih kosong.

Rancangan *input* dan *output* dari method *getPck* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:getPck xmlns:ns2="http://webservice/"/>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:getPckResponse xmlns:ns2="http://webservice/">
            <return>[
                    {"id_user":data_id_user_pencarikerja,
                    "nama":"data_nama_ pencarikerja ",
                    "tgl_lahir":"tgl_lahir_pencarikerja",
                    "usia":" data_usia_pencarikerja",
                    "jk":" data_jeniskelamin_pencarikerja",
                    "kategori":" data_kategori_pencarikerja",
```

```
                    "alamat":"data_alamat_pencarikerja",
                    "kota":" data_kota_pencarikerja",
                    "email":" data_email_pencarikerja"}]
                </return>
            </ns2:getPckResponse>
        </S:Body>
</S:Envelope>
```

15) Method *getPckByID*

Method *getPckByID* merupakan method atau fungsi yang digunakan oleh aplikasi Lowongan Pekerjaan untuk mendapatkan detail dari pencari kerja berdasarkan Id dari pencari kerja dari basis data lowongan pekerjaan. Method *getPckByID* memiliki satu variabel sebagai parameter input yang digunakan untuk mengambil detail pencari kerja berdasarkan id user pencari kerja dari basis data lowongan pekerjaan.

Berikut ini variabel yang dibutuhkan oleh method *getPckByID* pada saat akan mengambil detail pencari kerja berdasarkan id user pencari kerja seperti terlihat pada tabel III.28 dan juga prosedurnya pada tabel III.29.

Tabel III. 28 Variabel Method getPckByID

Nama Variabel	Isi Variabel	Tipe Variabel
id_user	Id_user dari pencari kerja yang akan diambil informasi detail dari pencari kerja tersebut	GET

Tabel III. 29 Prosedur Method getPckByID

Method *getPckByID* (input id_user:string)	
Masukan	Id User
Keluaran	Array pencari kerja dengan id tertentu dalam bentuk JSON
Skenario Utama Method *getPckByID*	
Proses	1. Mengambil Id Pencari Kerja yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan Pencari Kerja dengan ID tertentu ke dalam *array*.
	4. Mengembalikan Pencari Kerja berdasarkan Id tertentu ke dalam *array*.
Skenario Alternatif Method *getPckByID*	
	4a. Mengembalikan nilai kosong pencari kerja dengan id tertentu yang dimasukkan tidak ditemukan.

Rancangan *input* dan *output* dari method *getPckByID* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:getPckByID xmlns:ns2="http://webservice/">
            <id_user>id_user_pencarikerja_yang_dicari</id_user>
        </ns2:getPckByID>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:getPckByIDResponse xmlns:ns2="http://webservice/">
            <return>[
```

```
                    {"id_user":data_id_user_pencarikerja,
                    "nama":"data_nama_ pencarikerja",
                    "usia":" data_usia_pencarikerja",
                    "jk":" data_jeniskelamin_pencarikerja",
                    "kategori":" data_kategori_pencarikerja",
                    "idKategori":" data_id_kategori_pencarikerja",
                    "alamat":"data_alamat_pencarikerja",
                    "no_telepon":" data_no_telepon_pencarikerja",
                    "email":" data_alamat_email_pencarikerja",
                    "spesifikasi": data_spesifikasi_pencarikerja",
                    "pengalaman":" data_pengalaman_pencarikerja",
                    "gambar":"path_gambar_pencarikerja"}]
                    </return>
        </ns2:getPckByIDResponse>
    </S:Body>
</S:Envelope>
```

16) Method *searchPck*

Method *searchPck* merupakan method yang digunakan aplikasi Lowongan Pekerjaan untuk melakukan proses pencarian pencari kerja berdasarkan keyword pencari kerja pada basis data Lowongan Pekerjaan. Method *searchPck* memiliki satu variabel yang dikirimkan sebagai parameter pada saat melakukan pencarian. Berikut ini variabel yang dibutuhkan oleh method *searchPck* pada saat melakukan pencarian ke dalam basis data Lowongan Pekerjaan seperti terlihat pada tabel III.30 dan juga prosedurnya pada tabel III.31.

Tabel III. 30 Variabel Method searchPck

Nama Variabel	Isi Variabel	Tipe Variabel
keyword	*Keyword* dari pencari kerja yang akan dicari ke field nama, jenis kelamin, spesifikasi, pengalaman	GET

Tabel III. 31 Prosedur Method searchPck

Method *searchPck* (input keyword:string)	
Masukan	*Keyword* pencari kerja
Keluaran	*Array* pencari kerja dalam bentuk JSON
Skenario Utama Method *searchPck*	
Proses	1. Mengambil *keyword* pencari kerja yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan pencari kerja berdasarkan parameter pencarian ke dalam *array.*
	4. Mengembalikan pencari kerja ke dalam *array.*
Skenario Alternatif Method *searchPck*	
	4a. Mengembalikan nilai kosong jika pencari kerja dengan *keyword* yang dimasukkan tidak ditemukan.

Rancangan *input* dan *output* dari method *searchPck* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:searchPck xmlns:ns2="http://webservice/">
            <keyword>nama,jenis_kelamin,spesifikasi,pengalaman
                      pencarikerja</keyword>
        </ns2:searchPck>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:searchPckResponse xmlns:ns2="http://webservice/">
            <return>[
```

```
                    {"id_user": data_id_user_pencarikerja,
                    "nama":"data_nama_ pencarikerja ",
                    "tgl_lahir":" data_tgl_lahir_pencarikerja",
                    "jk":" data_jeniskelamin_pencarikerja",
                    "alamat":" data_alamat_pencarikerja",
                    "kota":" data_kota_pencarikerja",
                    "email":" data_alamat_email_pencarikerja",
                    "spesifikasi":" data_spesifikasi_pencarikerja",
                    "pengalaman":" data_pengalaman_pencarikerja"}]
                    </return>
        </ns2:searchPckResponse>
    </S:Body>
</S:Envelope>
```

17) Method *getPmk*

Method *getPmk* merupakan method atau fungsi yang digunakan oleh aplikasi Lowongan Pekerjaan untuk mendapatkan semua data pemberi kerja dari basis data lowongan pekerjaan. Method *getPmk* tidak memiliki variabel yang menjadi parameter input. Sedangkan prosedur method *getPmk* dapat dilihat pada tabel III.32.

Tabel III. 32 Variabel Method getPmk

Method *getPmk* ()	
Masukan	-
Keluaran	*Array* pemberi kerja dalam bentuk JSON
Skenario Utama Method *getPmk*	
Proses	1. Menerima *request* untuk mengeksekusi method *getPmk.*
	2. Menjalankan query sql.
	3. Memasukkan pemberi kerja ke dalam *array.*
	4. Mengembalikan pemberi kerja ke dalam *array.*
Skenario Alternatif Method *getPmk*	
	4a. Mengembalikan nilai kosong jika data pemberi kerja belum ada atau masih kosong.

Rancangan *input* dan *output* dari method *getPmk* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:getPmk xmlns:ns2="http://webservice/"/>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:getPmkResponse xmlns:ns2="http://webservice/">
            <return>[
                    {"id_user": data_id_user_pemberikerja,
                    "nama_perusahaan":"nama_perusahaan_pemberikerja",
                    "kota":"data_kota_ pemberikerja ",
                    "provinsi":"data_provinsi_ pemberikerja ",
                    "email":" data_email_ pemberikerja ",
                    "gambar":"path_gambar_logoperusahaan"}]
                    </return>
        </ns2:getPmkResponse>
    </S:Body>
</S:Envelope>
```

18) Method *getPmkByID*

Method *getPmkByID* merupakan method atau fungsi yang digunakan oleh aplikasi Lowongan Pekerjaan untuk mendapatkan detail dari pemberi kerja (perusahaan) berdasarkan Id dari pemberi kerja dari basis data lowongan pekerjaan. Method *getPmkByID* memiliki satu variabel sebagai parameter input yang digunakan untuk mengambil detail pemberi kerja berdasarkan id user pemberi kerja dari basis data lowongan pekerjaan.

Berikut ini variabel yang dibutuhkan oleh method *getPmkByID* pada saat akan mengambil detail pemberi kerja berdasarkan id user pemberi kerja seperti terlihat pada tabel III.33 dan juga prosedurnya pada tabel III.34.

Tabel III. 33 Variabel Method getPmkByID

Nama Variabel	Isi Variabel	Tipe Variabel
id_user	Id_user dari pemberi kerja yang akan diambil informasi detail dari pemberi kerja tersebut	GET

Tabel III. 34 Prosedur Method getPmkByID

Method *getPmkByID* (input id_user:string)	
Masukan	Id User
Keluaran	Array pemberi kerja dengan id tertentu dalam bentuk JSON
Skenario Utama Method *getPmkByID*	
Proses	1. Mengambil Id Pemberi Kerja yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan Pemberi Kerja dengan Id tertentu ke dalam *array*.
	4. Mengembalikan Pemberi Kerja berdasarkan Id tertentu ke dalam *array*.
Skenario Alternatif Method *getPmkByID*	
	4a. Mengembalikan nilai kosong jika data pemberi kerja dengan id tertentu yang dimasukkan tidak ditemukan.

Rancangan *input* dan *output* dari method *getPmkByID* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:getPmkByID xmlns:ns2="http://webservice/">
            <id_user>id_pemberikerja_yang_dicari</id_user>
        </ns2:getPmkByID>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:getPmkByIDResponse xmlns:ns2="http://webservice/">
            <return>[
                    {"id_user":data_id_user,
                    "nama_perusahaan":"data_nama_perusahaan",
                    "alamat":"data_alamat_pemberikerja",
                    "email":" data_email_pemberikerja",
                    "kode_pos":"data_kodepos_pemberikerja",
                    "deskripsi_perusahaan":" data_deskripsi_perusahaan",
                    "gambar":"path_gambar_logoperusahaan"}]
             </return>
        </ns2:getPmkByIDResponse>
    </S:Body>
</S:Envelope>
```

19) Method *postJob*

Method *postJob* merupakan method yang digunakan aplikasi Lowongan Pekerjaan untuk melakukan proses tambah lowongan pekerjaan baru. Method *postJob* memiliki beberapa variabel yang dikirimkan sebagai parameter pada saat melakukan penambahan data lowongan kerja. Berikut ini variabel-variabel yang dibutuhkan oleh method *postJob* pada saat melakukan penambahan data lowongan kerja ke dalam basis data Lowongan Pekerjaan seperti terlihat pada tabel III.35 dan juga prosedurnya pada tabel III.36, III.37.

Tabel III. 35 Variabel Method postJob

Nama Variabel	Isi Variabel	Default	Tipe Variabel
idLowongan	1…N ID dari lowongan kerja dan bersifat auto increment	NULL	GET
usernameAdmin	Username dari administrator sistem yang menambahkan lowongan pekerjaan baru	Kosong	GET
id_user	id user dari pemberi kerja (perusahaan) yang menambahkan lowongan pekerjaan baru	Kosong	GET
judul_lowongan	Judul lowongan dari lowongan pekerjaan	Kosong	GET
idKategori	1 sampai 60 Angka 1 sampi 60 merupakan idKategori pada tabel kategori yang direference oleh kolom kategori pada tabel lowongan	Kosong	GET
jobexpired	Date Tanggal terakhir ketika lowongan pekerjaan ini berakhir	Kosong	GET

Tabel III. 36 Variabel Method postJob (Lanjutan)

Nama Variabel	Isi Variabel	Default	Tipe Variabel
tipekerja	1 = Full Time 4 = Magang 2 = Part Time 5 = Other 3 = Kontrak Tipe kerja lowongan pekerjaan	Kosong	GET
gaji	Pilihan gaji dari lowongan pekerjaan. Contoh: Negosiasi, Di bawah 2 Juta, atau Di atas 2 Juta	Kosong	GET
lokasiPenempatan	Diisi kota dari lokasi penempatan lowongan pekerjaan	Kosong	GET
latitude	Titik koordinat latitude dari lokasi lowongan pekerjaan. Contoh: -6.893026	Kosong	GET
longitude	Titik koordinat longitude dari lokasi lowongan pekerjaan. Contoh: 107.60479	Kosong	GET
tanggung_jawab	Text Diisi dengan penjelasan tanggung jawab yang harus dilakukan dari lowongan pekerjaan	Kosong	GET
persyaratan	Text Diisi dengan persyaratan yang dibutuhkan dari lowongan pekerjaan	Kosong	GET
waktuInput	Timestamp Tanggal dan waktu input atau posting lowongan pekerjaan	NULL	GET

Tabel III. 37 Prosedur Method postJob

Method *postJob* (input idLowongan:string, input usernameAdmin:string, input id_user:string, input judul_lowongan:string, input idKategori:string, input jobexpired:string, input tipekerja:string, input gaji:string, input lokasiPenempatan:string, input latitude:string, input longitude:string, input tanggung_jawab:string, input persyaratan:string, input waktuInput:string,)	
Masukan	ID lowongan pekerjaan, *username* admin,id pemberi kerja , judul lowongan pekerjaan, kategori, job expired, tipe kerja, gaji, lokasi penempatan, latitude, longitude, tanggung jawab, persyaratan,waktu input
Keluaran	Pesan pemberitahuan bahwa data telah berhasil dikirim
Skenario Utama Method *postJob*	
Proses	1. Mengambil parameter tambah lowongan pekerjaan yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan data ke dalam basis data lowongan pekerjaan.
	4. Mengembalikan pesan bahwa data telah berhasil dikirim.
Skenario Alternatif Method *postJob*	
	4a. Mengembalikan pesan error jika terdapat salah satu kolom ada yang kosong atau tidak diisi.

Rancangan *input* dan *output* dari method *postJob* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:postJob xmlns:ns2="http://webservice/">
```

```
            <id_user>data_id_user_pemberikerja</id_user>
            <judul_lowongan>judul_lowongan</judul_lowongan>
            <idKategori>id_kategori</idKategori>
            <jobexpired>tanggal_berakhir_lowongan</jobexpired>
            <tipekerja>tipe_kerja</tipekerja>
            <gaji>gaji_lowongan</gaji>
            <lokasiPenempatan>lokasi_penempatan</lokasiPenempatan>
            <latitude>latitude_lokasi_penempatan</latitude>
            <longitude>longitude_lokasi_penempatan</longitude>
              <tanggung_jawab>tanggungjawab_pekerjaan</tanggung_jawab>
            <persyaratan>persyaratan_pencarikerja_yang_dibutuhkan
             </persyaratan>
        </ns2:postJob>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:postJobResponse xmlns:ns2="http://webservice/">
            <return>Data berhasil Dikirim</return>
        </ns2:postJobResponse>
    </S:Body>
</S:Envelope>
```

20) Method *addMinatKerja*

Method *addMinatKerja* merupakan method yang digunakan aplikasi Lowongan Pekerjaan untuk melakukan proses tambah minat kerja pada saat pengunjung situs lowongan pekerjaan melakukan proses registrasi. Method *addMinatKerja* memiliki beberapa variabel yang dikirimkan sebagai parameter pada saat melakukan penambahan data minat kerja. Berikut ini variabel-variabel yang dibutuhkan oleh method *addMinatKerja* pada saat melakukan penambahan data minat kerja ke dalam basis data Lowongan Pekerjaan seperti terlihat pada tabel III.38 dan juga prosedurnya pada tabel III.39.

Tabel III. 38 Variabel Method addMinatKerja

Nama Variabel	Isi Variabel	Default	Tipe Variabel
id_user	id user dari pencari kerja yang menambahkan data minat kerja pekerjaan baru	Kosong	GET
id_category	1…60 Merupakan id kategori dari kategori minat kerja yang terdapat pada tabel kategori	Kosong	GET

Tabel III. 39 Prosedur Method addMinatKerja

Method *addMinatKerja* (input id_user:string, input id_category:string)	
Masukan	Id user pencari kerja, id kategori
Keluaran	Pesan pemberitahuan bahwa data telah berhasil dikirim
Skenario Utama Method *addMinatKerja*	
Proses	1. Mengambil parameter tambah minat kerja yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan data ke dalam basis data lowongan pekerjaan.
	4. Mengembalikan pesan bahwa data telah berhasil dikirim.
Skenario Alternatif Method *addMinatKerja*	
	4a. Mengembalikan pesan error jika terdapat salah satu kolom ada yang kosong atau tidak diisi.

Rancangan *input* dan *output* dari method *addMinatKerja* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:addMinatKerja xmlns:ns2="http://webservice/">
           <id_user>id_user_pencarikerja</id_user>
            <id_category>id_kategori_minat_pencarikerja</id_category >
        </ns2:addMinatKerja >
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2: addMinatKerjaResponse xmlns:ns2="http://webservice/">
            <return>Data berhasil Dikirim</return>
        </ns2: addMinatKerjaResponse>
    </S:Body>
    </S:Envelope>
```

21) Method *getMinat*

Method *getMinat* merupakan method yang digunakan aplikasi Lowongan Pekerjaan untuk mengambil id kategori dari minat kerja dari member pencari kerja. Method *getMinat* memiliki satu variabel yang dikirimkan sebagai parameter pada saat mengambil id kategori dari minat kerja pencari kerja. Berikut ini variabel yang dibutuhkan oleh method *getMinat* pada saat mengambil id kategori dari minat kerja pencari kerja seperti terlihat pada tabel III.40 dan juga prosedurnya pada tabel III.41.

Tabel III. 40 Variabel Method getMinat

Nama Variabel	Isi Variabel	Default	Tipe Variabel
id_pencari	id user dari pencari kerja yang akan diambil id kategori minatnya	Kosong	GET

Tabel III. 41 Prosedur Method getMinat

Method *getMinat* (input id_pencari:string)	
Masukan	Id user pencari kerja
Keluaran	Id kategori minat kerja dari pencari kerja
Skenario Utama Method *getMinat*	
Proses	1. Mengambil parameter id user pencari kerja yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan id kategori minat kerja ke dalam *array.*
	4. Mengembalikan id kategori minat kerja ke dalam *array.*
Skenario Alternatif Method *getMinat*	
	4a. Mengembalikan nilai kosong id kategori dengan id user tertentu tidak ditemukan.

Rancangan *input* dan *output* dari method *getMinat* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
```

```
        <ns2:getMinat xmlns:ns2="http://webservice/">
            <id_pencari>id_user_pencarikerja</id_pencari>
        </ns2:getMinat>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:getMinatResponse xmlns:ns2="http://webservice/">
            <return>[
                 {"idKategori":id_kategori_minat_pencarikerja}]
             </return>
        </ns2:getMinatResponse>
    </S:Body>
</S:Envelope>
```

22) Method *applyJob*

Method *applyJob* merupakan method yang digunakan aplikasi Lowongan Pekerjaan untuk melakukan proses lamaran pekerjaan. Method *applyJob* memiliki dua variabel yang dikirimkan sebagai parameter pada saat melakukan proses lamaran kerja. Berikut ini variabel yang dibutuhkan oleh method *applyJob* pada saat melakukan proses lamaran kerja seperti terlihat pada tabel III.42 dan juga prosedurnya pada tabel III.43.

Tabel III. 42 Variabel Method applyJob

Nama Variabel	Isi Variabel	Default	Tipe Variabel
id_lowongan	idLowongan dari lowongan pekerjaan yang akan dilamar oleh pencari kerja	Kosong	GET
id_user	Id user dari pencari kerja yang melakukan lamaran kerja	Kosong	GET

Tabel III. 43 Prosedur Method applyJob

Method *applyJob* (input id_lowongan:string, id_user:string)	
Masukan	Id lowongan, Id user pencari kerja
Keluaran	Pesan pemberitahuan bahwa lamaran telah berhasil dikirim
Skenario Utama Method *applyJob*	
Proses	1. Mengambil parameter lamaran yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan data ke dalam basis data lowongan pekerjaan.
	4. Mengembalikan pesan bahwa lamaran telah berhasil dikirim.
Skenario Alternatif Method *applyJob*	
	4a. Mengembalikan pesan bahwa user sudah pernah melakukan lamaran pada pekerjaan tersebut jika id lowongan dan id user pencari kerja sudah terdapat pada *database*

Rancangan *input* dan *output* dari method *applyJob* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:applyJob xmlns:ns2="http://webservice/">
            <id_lowongan>id_lowongan_pekerjaan_yang_dilamar</id_lowongan>
           <id_user>id_user_pencarikerja_yang_melamar</id_user>
        </ns2:applyJob>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:applyJobResponse xmlns:ns2="http://webservice/">
            <return>[
                      {"message":"Lamaran        Anda        Berhasil
                Dikirim,<nama perusahaan> Akan Menghubungi Anda.",
                      "status":"1"}]
            </return>
        </ns2:applyJobResponse>
    </S:Body>
</S:Envelope>
```

23) Method *getApplyCompany*

Method *getApplyCompany* merupakan method yang digunakan aplikasi Lowongan Pekerjaan untuk melihat lamaran yang masuk ke sebuah lowongan pekerjaan. Method *getApplyCompany* memiliki satu variabel yang dikirimkan sebagai parameter untuk melihat lamaran pekerjaan yang masuk. Berikut ini variabel yang dibutuhkan oleh method *getApplyCompany* seperti terlihat pada tabel III.44 dan juga prosedurnya pada tabel III.45.

Tabel III. 44 Variabel Method getApplyCompany

Nama Variabel	Isi Variabel	Default	Tipe Variabel
id_lowongan	idLowongan dari lowongan pekerjaan yang dilamar oleh pencari kerja	Kosong	GET

Tabel III. 45 Prosedur Method getApplyCompany

Method *getApplyCompany* (input id_lowongan:string)	
Masukan	Id lowongan pekerjaan
Keluaran	Data pencari kerja yang melamar lowongan pekerjaan
Skenario Utama Method *getApplyCompany*	
Proses	1. Mengambil parameter id lowongan yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan data pencari kerja ke dalam *array*.
	4. Mengembalikan data pencari kerja ke dalam *array*.
Skenario Alterantif Method *getApplyCompany*	
	4a. Mengembalikan nilai kosong jika data pencari kerja yang melakukan lamaran ke lowongan dengan id tertentu yang dimasukkan tidak ditemukan.

Rancangan *input* dan *output* dari method *getApplyCompany* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:getApplyCompany xmlns:ns2="http://webservice/">
            <id_lowongan>id_lowongan_pekerjaan_yang_dilamar</id_lowongan>
        </ns2:getApplyCompany>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:getApplyCompanyResponse xmlns:ns2="http://webservice/">
            <return>[{
```

```
                    "jk":"data_jeniskelamin_pencarikerja",
                    "no_telepon":"data_no_telepon_pencarikerja",
                    "id_user":"data_id_user_pencarikerja",
                    "tgl_lahir":"data_tgl_lahir_pencarikerja",
                    "email":"data_email_pencarikerja",
                    "alamat":"data_alamat_pencarikerja",
                    "nama":"data_nama_pencarikerja",
                    "gambar":"path_gambar_pencarikerja",
                    "cv":"path_file_cv_pencarikerja"}]
                    </return>
        </ns2:getApplyCompanyResponse>
    </S:Body>
</S:Envelope>
```

24) Method *getApplyUserHistory*

Method *getApplyUserHistory* merupakan method yang digunakan aplikasi Lowongan Pekerjaan untuk melihat lamaran yang telah dilakukan oleh member pencari kerja. Method *getApplyUserHistory* memiliki satu variabel yang dikirimkan sebagai parameter untuk melihat lamaran pekerjaan yang pernah dilakukan oleh member pencari kerja. Berikut ini variabel yang dibutuhkan oleh method *getApplyUserHistory* seperti terlihat pada tabel III.46 dan juga prosedurnya pada tabel III.47.

Tabel III. 46 Variabel Method getApplyUserHistory

Nama Variabel	Isi Variabel	Default	Tipe Variabel
id_user	Id user dari pencari kerja yang diambil *history* lamran kerjanya	Kosong	GET

Tabel III. 47 Prosedur Method getApplyUserHistory

Method *getApplyUserHistory* (input id_user:string)	
Masukan	Id user pencari kerja
Keluaran	Data lowongan pekerjaan yang pernah dilamar oleh pencari kerja
Skenario Utama Method *getApplyUserHistory*	
Proses	1. Mengambil parameter id user yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan data lowongan pekerjaan ke dalam *array.*
	4. Mengembalikan data lowongan pekerjaan ke dalam *array.*
Skenario Alternatif Method *getApplyUserHistory*	
	4a. Mengembalikan nilai kosong jika data lowongan pekerjaan yang pernah dilamar oleh user pencari kerja tidak ditemukan.

Rancangan *input* dan *output* dari method *getApplyUserHistory* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:getApplyUserHistory xmlns:ns2="http://webservice/">
            <id_user>id_user_pencarikerja</id_user>
        </ns2:getApplyUserHistory>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
```

```
    <S:Body>
        <ns2:getApplyUserHistoryResponse xmlns:ns2="http://webservice/">
            <return>[
                    {"idLowongan":data_id_lowongan,
                    "idKategori":data_id_kategori,
                    "id_user":data_id_user,
                    "judul_lowongan":"data_judul_lowongan",
                    "kategori":"data_kategori",
                    "namaperusahaan":"data_namaperusahaan",
                    "jobexpired":"data_tanggal_berakhir_lowongan",
                    "tipekerja":"data_tipekerja_lowongan",
                    "kontak_perusahaan":" data_kontak_perusahaan",
                    "lokasiPenempatan":"data_lokasi_penempatan",
                    "waktuInput":"data_waktuinput_lowongan"}]
                    </return>
        </ns2:getApplyUserHistoryResponse>
    </S:Body>
</S:Envelope>
```

25) Method *getJobByCategory*

Method *getJobByCategory* merupakan method yang digunakan aplikasi Lowongan Pekerjaan untuk menampilkan daftar lowongan pekerjaan berdasarkan bidang minat dari member pencari kerja. Method *getJobByCategory* memiliki satu variabel yang dikirimkan sebagai parameter untuk melihat daftar lowongan pekerjaan berdasarkan bidang minat dari member pencari kerja. Berikut ini variabel yang dibutuhkan oleh method *getJobByCategory* seperti terlihat pada tabel III.48 dan juga prosedurnya pada tabel III.49.

Tabel III. 48 Variabel Method getJobByCategory

Nama Variabel	Isi Variabel	Default	Tipe Variabel
idKategori	Id ketegori yang merupakan bidang minat dari pemberi kerja yang mereference ke tabel kategori	Kosong	GET

Tabel III. 49 Prosedur Method getjobByCategory

Method *getJobByCategory* (input idKategori:string)	
Masukan	Id kategori lowongan pekerjaan.
Keluaran	*Array* lowongan pekerjaan berdasarkan kategori dalam bentuk JSON.
Skenario Utama Method *getJobByCategory*	
Proses	1. Mengambil id kategori lowongan yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan lowongan pekerjaan berdasarkan kategori tertentu ke dalam *array*.
	4. Mengembalikan lowongan pekerjaan dengan kategori tertentu ke dalam *array*.
Skenario Alternatif Method *getJobByCategory*	
	4a. Mengembalikan nilai kosong jika data lowongan pekerjaan dengan id kategori yang dimasukkan tidak ditemukan.

Rancangan *input* dan *output* dari method *getJobByCategory* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:getJobByCategoryName xmlns:ns2="http://webservice/">
            <kategori>nama_kategori_lowongan</kategori>
        </ns2:getJobByCategoryName>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:getJobByCategoryNameResponse xmlns:ns2="http://webservice/">
            <return>[
                    {"{"idLowongan":data_id_lowongan,
                    "idKategori":data_id_kategori,
                    "id_user":data_id_user,
                    "judul_lowongan":"data_judul_lowongan",
                    "kategori":"data_kategori",
                    "namaperusahaan":"data_namaperusahaan",
                    "jobexpired":"data_tanggal_berakhir_lowongan",
                    "tipekerja":"data_tipekerja_lowongan",
                    "kontak_perusahaan":" data_kontak_perusahaan",
                    "lokasiPenempatan":"data_lokasi_penempatan",
                    "waktuInput":"data_waktuinput_lowongan"}]
             </return>
        </ns2:getJobByCategoryNameResponse>
    </S:Body>
    </S:Envelope>
```

26) Method *updateJob*

Method *updateJob* merupakan method yang digunakan aplikasi Lowongan Pekerjaan untuk melakukan proses *edit* data lowongan pekerjaan. Method *updateJob* memiliki beberapa variabel yang dikirimkan sebagai parameter pada saat melakukan perubahanan atau *edit* data lowongan kerja. Berikut ini variabel-variabel yang dibutuhkan oleh method *updateJob* pada saat melakukan perubahanan atau *edit* data lowongan kerja ke dalam basis data Lowongan Pekerjaan seperti terlihat pada tabel III.50 dan III.51 dan juga prosedurnya pada tabel III.52.

Tabel III. 50 Variabel Method updateJob

Nama Variabel	Isi Variabel	Default	Tipe Variabel
idLowongan	1…N ID dari lowongan pekerjaan yang akan di-*edit*	idLowongan	GET
judul_lowongan	Judul lowongan dari lowongan pekerjaan	judul_lowongan	GET
idKategori	1 sampai 60 Angka 1 sampi 60 merupakan idKategori pada tabel kategori yang direference oleh kolom kategori pada tabel lowongan	idKategori	GET
jobexpired	Date Tanggal terakhir ketika lowongan pekerjaan ini berakhir	jobexpired	GET
tipekerja	1 = Full Time 4 = Magang 2 = Part Time 5 = Other 3 = Kontrak Tipe kerja lowongan pekerjaan	tipekerja	GET
gaji	Pilihan gaji dari lowongan pekerjaan. Contoh: Negosiasi, Di bawah 2 Juta, atau Di atas 2 Juta	gaji	GET
lokasiPenempatan	Diisi kota dari lokasi penempatan lowongan pekerjaan	lokasiPenempatan	GET

Tabel III. 51 Variabel Method updateJob (Lanjutan)

Nama Variabel	Isi Variabel	Default	Tipe Variabel
latitude	Titik koordinat latitude dari lokasi lowongan pekerjaan. Contoh: -6.893026	latitude	GET
longitude	Titik koordinat longitude dari lokasi lowongan pekerjaan. Contoh: 107.60479	longitude	GET
tanggung_jawab	Text Diisi dengan penjelasan tanggung jawab yang harus dilakukan dari lowongan pekerjaan	tanggung_jawab	GET
persyaratan	Text Diisi dengan persyaratan yang dibutuhkan dari lowongan pekerjaan	persyaratan	GET

Tabel III. 52 Prosedur Method updateJob

Method *updateJob* (input idLowongan:string, input judul_lowongan:string, input idKategori:string, input jobexpired:string, input tipekerja:string, input gaji:string, input lokasiPenempatan:string, input latitude:string, input longitude:string, input tanggung_jawab:string, input persyaratan:string)	
Masukan	ID lowongan pekerjaan, judul lowongan pekerjaan, kategori, job expired, tipe kerja, gaji, lokasi penempatan, latitude, longitude, tanggung jawab, persyaratan,
Keluaran	Pesan pemberitahuan bahwa data telah berhasil diupdate
Skenario Utama Method *updateJob*	
Proses	1. Mengambil parameter *update* lowongan pekerjaan yang dimasukkan.
	2. Menjalankan query sql.
	3. Memasukkan data ke dalam basis data lowongan pekerjaan.
	4. Mengembalikan pesan bahwa data telah berhasil diupdate.
Skenario Alternatif Method *updateJob*	
	4a. Mengembalikan pesan error jika terdapat salah satu kolom ada yang kosong atau tidak diisi.

Rancangan *input* dan *output* dari method *updateJob* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:updateJob xmlns:ns2="http://webservice/">
        <id_user>data_id_user_pemberikerja</id_user>
```

```
            <judul_lowongan>judul_lowongan</judul_lowongan>
            <idKategori>id_kategori</idKategori>
            <jobexpired>tanggal_berakhir_lowongan</jobexpired>
            <tipekerja>tipe_kerja</tipekerja>
            <gaji>gaji_lowongan</gaji>
            <lokasiPenempatan>lokasi_penempatan</lokasiPenempatan>
            <latitude>latitude_lokasi_penempatan</latitude>
            <longitude>longitude_lokasi_penempatan</longitude>
              <tanggung_jawab>tanggungjawab_pekerjaan</tanggung_jawab>
            <persyaratan>persyaratan_pencarikerja_yang_dibutuhkan
             </persyaratan>
            </ns2:updateJob>
        </S:Body>
    </S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:updateJobResponse xmlns:ns2="http://webservice/">
            <return>Data berhasil Di Update</return>
        </ns2:updateJobResponse>
    </S:Body>
</S:Envelope>
```

27) Method *deleteJob*

Method *deleteJob* merupakan method yang digunakan aplikasi Lowongan Pekerjaan untuk data lowongan pekerjaan. Method *deleteJob* memiliki satu variabel yang dikirimkan sebagai parameter untuk menghapus data lowongan pekerjaan. Berikut ini variabel yang dibutuhkan oleh method *deleteJob* seperti terlihat pada tabel III.53 dan juga prosedurnya pada tabel III.54.

Tabel III. 53 Variabel Method deleteJob

Nama Variabel	**Isi Variabel**	**Default**	**Tipe Variabel**
idLowongan	1…N ID dari lowongan pekerjaan yang akan di-*edit*	Kosong	GET

Tabel III. 54 Prosedur Method deleteJob

Method *deleteJob* (input idLowongan:string)	
Masukan	ID Lowongan
Keluaran	Pesan pemberitahuan bahwa data telah berhasil dihapus
Proses	1. Mengambil ID Lowongan Pekerjaan yang dimasukkan.
	2. Menghapus lowongan pekerjaan sesuai dengan ID Lowongan yang dimasukkan.
	3. Mengembalikan pesan bahwa data telah berhasil dihapus.

Rancangan *input* dan *output* dari method *deleteJob* dapat kita lihat pada struktur dari dokumen SOAP *request* dan SOAP *response* berikut.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:deleteJob xmlns:ns2="http://webservice/">
            <idLowongan>id_lowongan_yang_akan_dihapus</idLowongan>
        </ns2:deleteJob>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:deleteJobResponse xmlns:ns2="http://webservice/">
            <return>Data berhasil Di Hapus</return>
        </ns2:deleteJobResponse>
    </S:Body>
</S:Envelope>
```

3.1.4.3 Analisis WSDL

SOAP sebagai standar pertukaran pesan dalam *web service* mampu berjalan sendiri tanpa bantuan WSDL untuk mendeskripsikannya. Tetapi, hal ini akan menyulitkan pengembangan, karena tanpa bantuan WSDL, para pengembang harus bertanya-tanya seperti apa bentuk pesan yang seharusnya dikirim kepada atau diterima dari penyedia layanan *web service*. Dengan bantuan WSDL, para pengembang telah dibantu karena telah mengetahui seperti apa pesan yang seharusnya dikirim atau diterima. WSDL dalam pembuatannya terlihat sangat menyulitkan dan membingungkan, tetapi dengan bantuan *library* dan *tools* yang telah beredar di *internet*, pembuatannya menjadi mudah dan cepat.

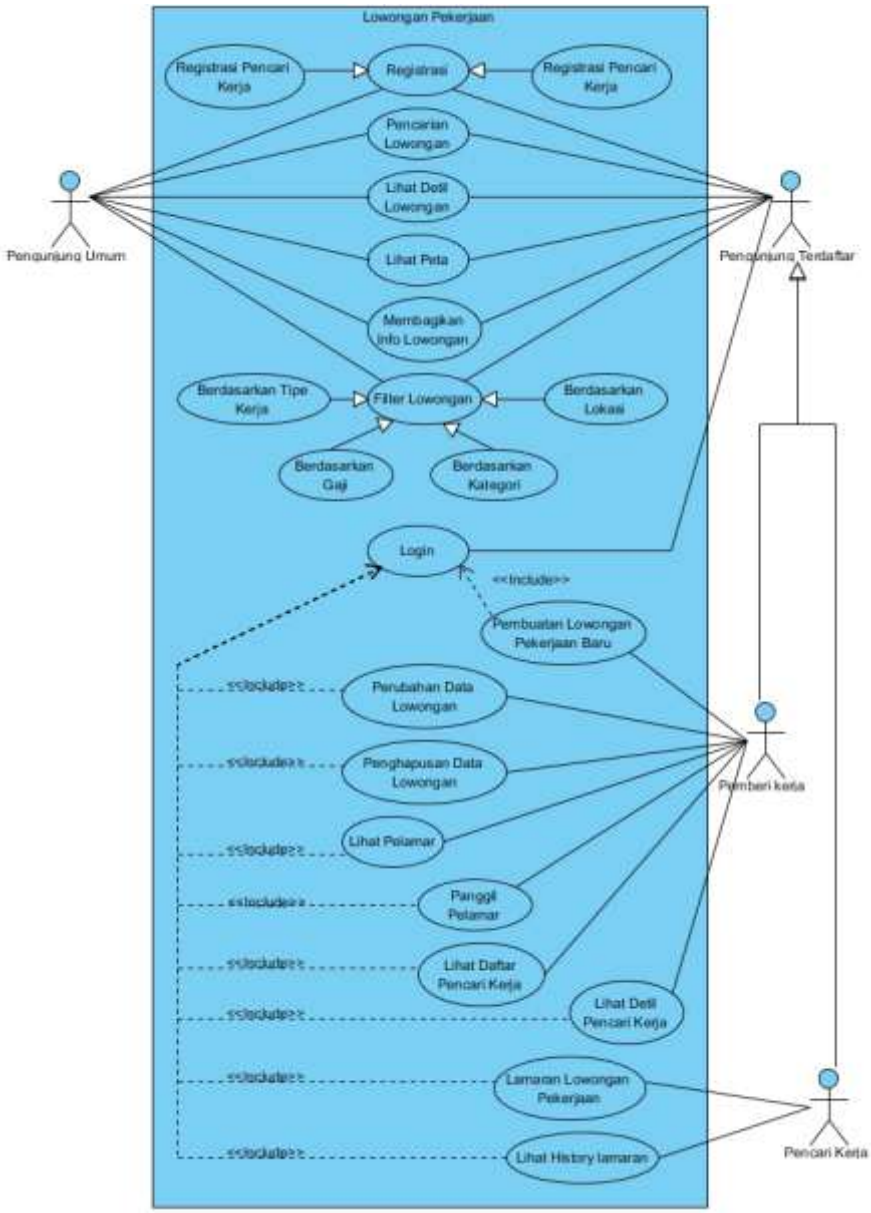
3.1.5 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sistem yang akan dibangun sehingga sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan proses bisnis.

Analisis dimodelkan dengan menggunakan UML (*Unified Modeling Language*) versi 2. Tahap-tahap pemodelan dalam analisis menggunakan UML antara lain use case diagaram, activity diagram, sequence diagram, dan class diagram.

3.1.5.1 Analisis Pada Aplikasi Android

1. **Use Case Diagram**



Gambar III. 1 Use Case diagram sistem pada aplikasi android

a. Identifikasi Aktor

Pada tabel III.55 berisi daftar aktor dan deskripsi peran meraka masing-masing dalam aplikasi ini.

Tabel III. 55 Definisi Aktor pada aplikasi android

NO	Aktor	Deskripsi
A-01	Pengunjung Umum	Merupakan pengguna non member dari aplikasi lowongan pekerjaan pada android
A-02	Pemberi Kerja	Merupakan pengguna dari aplikasi lowongan pekerjaan pada android sebagai member pemberi kerja yang akan dapat menambahkan lowongan pekerjaan baru.
A-03	Pencari Kerja	Merupakan pengguna dari aplikasi lowongan pekerjaan pada android sebagai member pencari kerja yang dapat melakukan *apply* lowongan.

b. Identifikasi Use Case

Berdasarkan use case diagram di atas maka di buatlah skenario dari tiap proses yang ada pada use case diagram tersebut. Pada tabel III.56 menggambarkan skenario login lowongan pekerjaan pada use case *Login*.

Tabel III. 56 Skenario Use Case Login

Identifikasi	
Nomor	UC-M-01
Nama Use Case	Login
Deskripsi	Proses untuk melakukan *login* ke dalam aplikasi Lowongan Pekerjaan
Aktor	Pengunjung Umum
Skenario Utama(UC-M-01-S01)	
Kondisi awal	Running Program dan Tampil Menu Utama
Aksi aktor	**Respon Sistem**
1. Memilih menu login	2. Menampilkan menu *login*
3. Mengisi data *login*	
4. Menekan tombol *login*	5. Sistem melakukan validasi *field login*
	6. Memanggil fungsi login
	7. Sistem melakukan *authentikasi login*
	8. Menampilkan halaman member
Skenario Alternatif (UC-M-01-S02) Login namun terdapat pesan error	
	5a. Menampilkan pesan kesalahan, karena *username* atau *password* belum terisi atau masih kosong.
Skenario Alternatif (UC- M-01-S03) Login namun data user tidak ditemukan atau gagal login	
	7a. Menampilkan pesan kesalahan login gagal karena user tidak ditemukan.
Kondisi akhir	*Login* Berhasil.

Pada tabel III.57 menggambarkan skenario mencari data lowongan pekerjaan pada use case Pencarian Lowongan.

Tabel III. 57 Skenario Use Case Pencarian Lowongan.

Identifikasi	
Nomor	UC-M-02
Nama Use Case	Pencarian Lowongan
Deskripsi	Proses untuk mencari data lowongan pekerjaan yang akan di tampilkan urut berdasarkan tanggal posting lowongan pekerjaan terbaru
Aktor	Pengunjung Umum, Member
Skenario Utama(UC-M-02-S01)	
Kondisi awal	Running Program dan Tampil Menu Utama
Aksi aktor	**Respon Sistem**
1. Memilih menu *Search*	2. Menampilkan form inputan
3. Memasukkan parameter pencarian pekerjaan	4. Memanggil fungsi searchJob
	5. Mengambil *return value* dari *web service* dalam bentuk SOAP
	6. Memparsing data dari *return value*
	7. Menampilkan data hasil parsing
Skenario Alternatif (UC-M-02-S02) Pencarian Lowongan namun terdapat pesan error	
	4a. Menampilkan pesan kesalahan, inputan pencarian belum diisi atau belum lengkap, atau tidak ada koneksi *internet* atau tidak dapat terhubung ke *server*.
Skenario Alternatif (UC- M-02-S03) Pencarian Lowongan namun data tidak ditemukan	
	7a. menampilkan pesan kesalahan, misal karena data yang di cari tidak ditemukan.
Kondisi akhir	Data lowongan pekerjaan berhasil ditampilkan di android

Pada tabel III.58 menggambarkan skenario filter data lowongan pekerjaan berdasarkan lokasi lowongan pekerjaan tersebut pada use case Filter Lowongan Berdasarkan Lokasi.

Tabel III. 58 Skenario Use Case Filter Lowongan Berdasarkan Lokasi.

Identifikasi	
Nomor	UC-M-03
Nama Use Case	Filter Lowongan Berdasarkan Lokasi
Deskripsi	Proses untuk memfilter data lowongan pekerjaan yang akan di tampilkan berdasarkan lokasi lowongan pekerjaan
Aktor	Pengunjung Umum, Member
Skenario Utama(UC-M-03-S01)	
Kondisi awal	Berada pada halaman daftar lowongan pekerjaan
Aksi aktor	**Respon Sistem**
1. Memilih menu filter lowongan pekerjaan berdasarkan lokasi	2. Menampilkan form inputan
3. Memilih lokasi lowongan pekerjaan	4. Memanggil fungsi getJobByLocation
	5. Mengambil *return value* dari *web service* dalam bentuk SOAP
	6. Memparsing data dari *return value*
	7. Menampilkan data hasil parsing
Skenario Alternatif (UC- M-03-S02) Filter Lowongan Berdasarkan Lokasi namun terdapat pesan error	
	9.a Menampilkan pesan kesalahan, misalnya karena data yang di filter tidak ada
Kondisi akhir	Data lowongan pekerjaan berdasarkan lokasi berhasil ditampilkan di android

Pada tabel III.59 menggambarkan skenario filter data lowongan pekerjaan berdasarkan kategori pekerjaan pada use case Filter Lowongan Berdasarkan Kategori.

Tabel III. 59 Skenario Use Case Filter Lowongan Berdasarkan Kategori.

Identifikasi	
Nomor	UC-M-04
Nama Use Case	Filter Lowongan Berdasarkan Kategori
Deskripsi	Proses untuk memfilter data lowongan pekerjaan yang akan di tampilkan berdasarkan kategori pekerjaan
Aktor	Pengunjung Umum, Member
Skenario Utama(UC-M-04-S01)	
Kondisi awal	Berada pada halaman daftar lowongan pekerjaan
Aksi aktor	**Respon Sistem**
1. Memilih menu filter lowongan pekerjaan berdasarkan kategori.	2. Menampilkan form inputan
3. Memilih kategori lowongan pekerjaan	4. Memanggil fungsi getJobByCategory
	5. Mengambil *return value* dari *web service* dalam bentuk SOAP
	6. Memparsing data dari *return value*
	7. Menampilkan data hasil parsing
Skenario Alternatif (UC-M-04-S02) Filter Lowongan Berdasarkan Kategori namun terdapat pesan error	
	9.a Menampilkan pesan kesalahan, misalnya karena data yang di filter tidak ada
Kondisi akhir	Data lowongan berdasarkan kategori berhasil ditampilkan di android

Pada tabel III.60 menggambarkan skenario filter data lowongan pekerjaan berdasarkan gaji tertentu pada use case Filter Lowongan Berdasarkan Gaji.

Tabel III. 60 Skenario Use Case Filter Lowongan Berdasarkan Gaji.

Identifikasi	
Nomor	UC-M-05
Nama Use Case	Filter Lowongan Berdasarkan Gaji
Deskripsi	Proses untuk memfilter data lowongan pekerjaan yang akan di tampilkan berdasarkan gaji
Aktor	Pengunjung Umum, Member
Skenario Utama (UC-M-05-S01)	
Kondisi awal	Berada pada halaman daftar lowongan pekerjaan
Aksi aktor	**Respon Sistem**
1. Memilih menu filter lowongan pekerjaan berdasarkan salary.	2. Menampilkan form inputan
3. Memilih rentang gaji	4. Memanggil fungsi getJobBySalary
	5. Mengambil *return value* dari *web service* dalam bentuk SOAP
	6. Memparsing data dari *return value*
	7. Menampilkan data hasil parsing
Skenario Alternatif (UC-M-05-S02) Filter Lowongan Berdasarkan Gaji namun terdapat pesan error	
	7.a Menampilkan pesan kesalahan, misalnya karena data yang di filter tidak ada.
Kondisi akhir	Data lowongan berdasarkan gaji berhasil ditampilkan di android

Pada tabel III.61 menggambarkan skenario filter data lowongan pekerjaan berdasarkan tipe pekerjaan pada use case Filter Lowongan Berdasarkan Tipe Kerja.

Tabel III. 61 Skenario Use Case Filter Lowongan Berdasarkan Tipe Kerja.

Identifikasi	
Nomor	UC-M-06
Nama Use Case	Filter Lowongan Berdasarkan Tipe Kerja
Deskripsi	Proses untuk menfilter data lowongan pekerjaan yang akan di tampilkan berdaskan tipe pekerjaan.
Aktor	Pengunjung Umum, Member
Skenario Utama(UC-M-06-S01)	
Kondisi awal	Berada pada halaman daftar lowongan pekerjaan
Aksi aktor	**Respon Sistem**
1. Memilih menu filter lowongan pekerjaan berdasarkan tipe pekerjaan.	2. Menampilkan form inputan
3. Memilih tipe pekerjaan	4. Memanggil fungsi getJobByJobType
	5. Mengambil *return value* dari *web service* dalam bentuk SOAP
	6. Memparsing data dari *return value*
	7. Menampilkan data hasil parsing
Skenario Alternatif(UC-M-06-S02) Filter Lowongan Berdasarkan Tipe Kerja namun data tidak ditemukan	
	7.a Menampilkan pesan kesalahan, misalnya karena data yang di filter tidak ada.
Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel III.62 menggambarkan skenario mengambil data lowongan pekerjaan pada use case Lihat Daftar Lowongan.

Tabel III. 62 Skenario Use Case Lihat Daftar Lowongan.

Identifikasi	
Nomor	UC-M-07
Nama Use Case	Lihat Daftar Lowongan
Deskripsi	Proses untuk mengambil seluruh data lowongan pekerjaan yang ada
Aktor	Pengunjung Umum, Member
Skenario Utama(UC-M-07-S01)	
Kondisi awal	Running Program dan Tampil Menu Utama
Aksi aktor User	**Respon Sistem**
1. Memilih menu semua pekerjaan	2. Memanggil fungsi getJob
	3. Mengambil *return value* dari *web service* dalam bentuk SOAP
	4. Melakukan parsing data
	5. Menampilkan data hasil parsing
Skenario Alternatif (UC-M-07-S02) Lihat Daftar Lowongan namun terdapat pesan error	
	2a. Menampilkan pesan kesalahan, misalnya karena tidak ada koneksi *internet* atau tidak dapat terhubung ke *server*.
Kondisi akhir	Data lowongan pekerjaan berhasil ditampilkan di android

Pada tabel III.63 menggambarkan skenario mengambil data lowongan pekerjaan berdasarkan id pada use case Lihat Detil Lowongan.

Tabel III. 63 Skenario Use Case Lihat Detil Lowongan.

Identifikasi	
Nomor	UC-M-08
Nama Use Case	Lihat Detil Lowongan
Deskripsi	Proses untuk mengambil data lowongan pekerjaan berdasarkan id pekerjaan
Aktor	Pengunjung Umum, Member
Skenario Utama (UC-M-08-S01)	
Kondisi awal	Berada pada halaman daftar lowongan pekerjaan
Aksi aktor	**Respon Sistem**
1. Memilih lowongan pekerjaan pada listView	2. Menggambil parameter id lowongan pekerjan
	3. Memanggil fungsi getJobByID
	4. Mengambil *return value* dari *web service* dalam bentuk SOAP
	5. Melakukan parsing data
	6. Menampilkan data hasil parsing
Skenario Alternatif (UC-M-08-S02) Lihat Detil Lowongan namun terdapat pesan error	
	4a. Menampilkan pesan kesalahan, misalnya karena tidak ada koneksi *internet* atau tidak dapat terhubung ke *server*.
Kondisi akhir	Data detil lowongan pekerjaan berhasil ditampilkan di android

Pada tabel III.64 menggambarkan skenario menampilkan peta lokasi lowongan pekerjaan pada use case Lihat Peta.

Tabel III. 64 Skenario Use Case Lihat Peta.

Identifikasi	
Nomor	UC-M-09
Nama Use Case	Lihat Peta
Deskripsi	Proses untuk menampilkan peta dari lokasi lowongan pekerjaan
Aktor	Pengunjung Umum, Member
Skenario Utama(UC-M-09-S01)	
Kondisi awal	Berada pada halaman detail lowongan pekerjaan
Aksi aktor	**Respon Sistem**
1. Menekan image view peta di sebelah lokasi lowongan pekerjaan	2. Mengambil latitude dan longitude dari lokasi lowongan pekerjaan.
	3. Mencari posisi latitude dan longitude pada google map
	4. Menampilkan peta
Kondisi akhir	Peta berhasil di tampilkan

Pada tabel III.65 menggambarkan skenario untuk *share* lowongan pekerjaan melalui aplikasi *social network* maupun email yang tersedia di *device user* pada use case Membagikan Info Lowongan.

Tabel III. 65 Skenario Use Case Membagikan Info Lowongan

Identifikasi	
Nomor	UC-M-10
Nama Use Case	Membagikan Info Lowongan
Deskripsi	Proses untuk membagikan lowongan pekerjaan melalui aplikasi *social networking* atau email
Aktor	Pengunjung Umum, Member
Skenario Utama(UC-M-06-S01)	
Kondisi awal	Berada pada halaman detail lowongan pekerjaan
Aksi aktor	**Respon Sistem**
1. Menekan image view *share*	2. Menampilkan pilihan aplikasi untuk *share* lowongan pekerjaan
3. Memilih salah satu aplikasi sebagai media untuk *share* lowongan pekerjaan	4. Mengambil data lowongan pekerjaan berupa judul lowongan, *job expired* dan lokasi lowongan pekerjaan
	5. Memasukkan parameter lowongan pekerjaan yang akan dibagikan ke aplikasi
6. Mengeksekusi tombol bagi atau kirim	7. Menampilkan pesan lowongan pekerjaan berhasil diposting atau dikirim.
Kondisi akhir	Data lowongan pekerjaan berhasil diposting

Pada tabel III.66 menggambarkan skenario mengambil data pencari kerja pada use case Lihat Daftar Pencari Kerja.

Tabel III. 66 Skenario Use Case Lihat Daftar Pencari Kerja

Identifikasi	
Nomor	UC-M-11
Nama Use Case	Lihat Daftar Pencari Kerja
Deskripsi	Proses untuk mengambil seluruh data lowongan pekerjaan yang ada
Aktor	Pemberi kerja
Skenario Utama(UC-M-11-S01)	
Kondisi awal	Login sebagai pemberi kerja berhasil dan berada pada halaman member
Aksi aktor User	**Respon Sistem**
1. Memilih tab pencari kerja	2. Memanggil fungsi getPck
	3. Mengambil *return value* dari *web service* dalam bentuk SOAP
	4. Melakukan parsing data
	5. Menampilkan data hasil parsing
Skenario Alternatif (UC-M-11-S02) Lihat Daftar Pencari Kerja namun terdapat pesan error	
	2a. Menampilkan pesan kesalahan, misalnya karena tidak ada koneksi *internet* atau tidak dapat terhubung ke *server*.
Kondisi akhir	Data pencari kerja berhasil ditampilkan di android

Pada tabel III.67 menggambarkan skenario mengambil data pencari kerja berdasarkan id pada use case Lihat Detil Pencari Kerja.

Tabel III. 67 Skenario Use Case Lihat Detil Pencari Kerja

Identifikasi	
Nomor	UC-M-12
Nama Use Case	Lihat Detil Pencari Kerja
Deskripsi	Proses untuk mengambil data pencari kerja berdasarkan id pencari kerja
Aktor	Pemberi kerja
Skenario Utama (UC-M-12-S01)	
Kondisi awal	Login sebagai pemberi kerja berhasil dan berada pada tab pencari kerja
Aksi aktor	**Respon Sistem**
1. Memilih pencari kerja pada *list view*	2. Menggambil parameter id pencari kerja
	3. Memanggil fungsi getPckByID
	4. Mengambil *return value* dari *web service* dalam bentuk SOAP
	5. Melakukan parsing data
	6. Menampilkan data hasil parsing
Skenario Alternatif (UC-M-12-S02) Lihat Detil Pencari Kerja namun terdapat pesan error	
	4a. Menampilkan pesan kesalahan, misalnya karena tidak ada koneksi *internet* atau tidak dapat terhubung ke *server*.
Kondisi akhir	Data detil pncari kerja berhasil ditampilkan di android

Pada tabel III.68 menggambarkan skenario untuk melakukan proses tambah lowongan baru pada use case Pembuatan Lowongan Pekerjaan Baru.

Tabel III. 68 Skenario Use Case Pembuatan Lowongan Pekerjaan Baru

Identifikasi	
Nomor	UC-M-13
Nama Use Case	Pembuatan Lowongan Pekerjaan Baru
Deskripsi	Proses untuk menambah lowongan pekerjaan baru
Aktor	Pemberi kerja
Skenario Utama (UC-M-13-S01)	
Kondisi awal	Login sebagai pemberi kerja dan koneksi ke *web service* berhasil
Aksi aktor	**Respon Sistem**
1. Memilih menu tambah lowongan pekerjaan	2. Menampilkan form inputan
3. Memasukkan parameter tambah lowongan pekerjaan baru	4. Sistem melakukan validasi *field* tambah lowongan
	5. Memanggil fungsi postJob
	6. Mengirim data lowongan pekerjaan baru ke database
	7. Lowongan berhasil ditambahkan dan sistem menampilkan data lowongan pekerjaan
Skenario Alternatif (UC-M-13-S02) Pembuatan Lowongan Pekerjaan Baru namun terdapat pesan error	
	4a. Menampilkan pesan kesalahan, misalnya karena *field* masih ada yang kosong atau parameter input *invalid*
Kondisi akhir	Data lowongan pekerjaan baru berhasil ditambahkan

Pada tabel III.69 menggambarkan skenario untuk melakukan proses mengubah data lowongan pekerjaan pada use case Perubahan Data Lowongan.

Tabel III. 69 Skenario Use Case Perubahan Data Lowongan

Identifikasi	
Nomor	UC-M-14
Nama Use Case	Perubahan Data Lowongan
Deskripsi	Proses untuk mengubah data lowongan pekerjaan
Aktor	Pemberi kerja
Skenario Utama (UC-M-14-S01)	
Kondisi awal	Login sebagai pemberi kerja dan koneksi ke *web service* berhasil
Aksi aktor	**Respon Sistem**
1. Memilih lowongan pekerjaan pada *list view*	2. Menampilkan pilihan *edit* lowongan
3. Memilih pilihan *edit* lowongan	4. Menampilkan form inputan
5. Memasukkan parameter *edit* lowongan pekerjaan	6. Sistem melakukan validasi *field edit* lowongan
	7. Memanggil fungsi updateJob
	8. Mengirim data lowongan pekerjaan ke database
	9. Lowongan berhasil di-*edit* dan sistem menampilkan data lowongan pekerjaan
Skenario Alternatif (UC-M-14-S02) Perubahan Data Lowongan namun terdapat pesan error	
	6a. Menampilkan pesan kesalahan, misalnya karena *field* masih ada yang kosong atau parameter input *invalid*
Kondisi akhir	Data lowongan pekerjaan baru berhasil ditambahkan

Pada tabel III.70 menggambarkan skenario untuk melakukan proses lamar lowongan pekerjaan baru pada use case Lamaran Lowongan Pekerjaan.

Tabel III. 70 Skenario Use Case Lamaran lowongan Pekerjaan

Identifikasi	
Nomor	UC-M-15
Nama Use Case	Lamaran Lowongan Pekerjaan
Deskripsi	Proses untuk melamar lowongan pekerjaan
Aktor	Pencari kerja
Skenario Utama (UC-M-15-S01)	
Kondisi awal	Login sebagai pencari kerja dan berada pada halaman daftar lowongan pekerjaan
Aksi aktor	**Respon Sistem**
1. Memilih menu *apply*	2. Mengambil id user dan id lowongan yang dilamar.
	3. Memanggil fungsi *applyJob*
	4. Mengambil *return value* dari *web service* dalam bentuk SOAP
	5. Memparsing data dari *return value*
	6. Menampilkan data hasil parsing lamaran berhasil dikirim
Skenario Alternatif (UC-M-15-S02) Lamaran Lowongan Pekerjaan namun terdapat pesan error	
	6a. Menampilkan pesan kesalahan lamaran gagal dikirim karena sudah pernah melakukan lamaran pada lowongan pekerjaan yang dilamar
Kondisi akhir	Lamaran berhasil dikirim

Pada tabel III.71 menggambarkan skenario untuk melihat para pelamar yang sudah melakukan lamaran pada suatu lowongan pekerjaan pada use case Lihat Pelamar.

Tabel III. 71 Skenario Use Case Lihat Pelamar

Identifikasi	
Nomor	UC-M-16
Nama Use Case	Lihat Pelamar
Deskripsi	Proses untuk melihat pelamar pada lowongan pekerjaan
Aktor	Pemberi kerja
Skenario Utama (UC-M-16-S01)	
Kondisi awal	Login sebagai pemberi kerja dan berada pada halaman daftar lowongan pekerjaan
Aksi aktor	**Respon Sistem**
1. Memilih lowongan pada *list view*	2. Menampilkan pilihan lihat pelamar
3. Memilih pilihan lihat pelamar	4. Mengambil id lowongan yang dipilih.
	5. Memanggil fungsi *getApply*
	6. Mengambil *return value* dari *web service* dalam bentuk SOAP
	7. Memparsing data dari *return value*
	8. Menampilkan data hasil parsing berupa daftar pelamar
Skenario Alternatif (UC-M-16-S02) Lihat Pelamar namun terdapat pesan error	
	8a. Menampilkan pesan kesalahan, misal karena tidak ada pelamar pada lowongan pekerjaan yang dipilih
Kondisi akhir	Data pelamar berhasil ditampilkan di android

Pada tabel III.72 menggambarkan skenario untuk melihat history lamaran pekerjaan yang pernah dilakukan oleh pencari kerja pada use case Lihat History Lamaran.

Tabel III. 72 Skenario Use Case Lihat History Lamaran

Identifikasi	
Nomor	UC-M-17
Nama Use Case	Lihat History Lamaran
Deskripsi	Proses untuk melihat *history* lamaran yang pernah dilakukan oleh member pencari kerja
Aktor	Pencari kerja
Skenario Utama (UC-M-17-S01)	
Kondisi awal	Login sebagai pencari kerja dan berada pada halaman member pencari kerja
Aksi aktor	**Respon Sistem**
1. Memilih tab *history* lamaran	2. Memanggil fungsi getApplyUserHistory
	3. Mengambil *return value* dari *web service* dalam bentuk SOAP
	4. Memparsing data dari *return value*
	5. Menampilkan data hasil parsing berupa daftar *history* lamaran
Skenario Alternatif (UC-M-17-S02) Lihat Pelamar namun terdapat pesan error	
	5a. Menampilkan pesan kesalahan, misal karena belum pernah melakukan lamaran
Kondisi akhir	Data *history* lamaran berhasil ditampilkan di android

Pada tabel III.73 menggambarkan skenario untuk memanggil pelamar dengan mengirimkan email ke alamat email pelamar seperti seperti terlihat pada use case Panggil Pelamar.

Tabel III. 73 Skenario Use Case Panggil Pelamar

Identifikasi	
Nomor	UC-M-18
Nama Use Case	Panggil Pelamar
Deskripsi	Proses untuk memanggil pelamar dengan mengirimkan panggilan ke alamat email dari pelamar
Aktor	Pemberi kerja
Skenario Utama (UC-M-18-S01)	
Kondisi awal	Login sebagai pemberi kerja dan berada pada halaman pelamar
Aksi aktor	**Respon Sistem**
1. Memilih pelamar pada *list view*	2. Menampilkan pilihan panggil pelamar
3. Memilih pilihan panggil pelamar	4. Menampilkan form panggil pelamar
5. Memasukkan parameter panggil pelamar	6. Mengambil parameter panggil pelamar
	7. Mengirimkan email ke pelamar
	8. Menampilkan pesan email terkirim
Skenario Alternatif (UC-M-18-S02) Panggil Pelamar namun terdapat pesan error	
	8a. Menampilkan pesan kesalahan email gagal terkirim
Kondisi akhir	Panggil pelamar berhasil dilakukan

Pada tabel III.74 menggambarkan skenario untuk melakukan registrasi menjadi member lowongan pekerjaan pada use case Registrasi.

Tabel III. 74 Skenario Use Case Registrasi

Identifikasi	
Nomor	UC-M-19
Nama Use Case	Registrasi
Deskripsi	Proses untuk registrasi menjadi member lowongan pekerjaan
Aktor	Pengunjung Umum
Skenario Utama (UC-M-19-S01)	
Kondisi awal	Running Program dan Tampil Menu Utama
Aksi aktor	**Respon Sistem**
1. Memilih menu *Sign Up*	2. Menampilkan pilihan tipe member
3. Memilih tipe member	4. Menampilkan form registrasi sesuai member yang dipilih
5. Memasukkan parameter registrasi	
6. Menekan tombol registrasi	7. Sistem melakukan validasi *field* registrasi
	8. Memanggil fungsi registrasi
	9. Sistem melakukan *authentikasi* registrasi
	10. Menampilkan pesan registrasi berhasil
Skenario Alternatif (UC-M-19-S02) Registrasi namun terdapat pesan error	
	7a. Menampilkan pesan kesalahan, misal semua kolom harus terisi. 10a. Menampilkan pesan kesalahan registrasi gagal
Kondisi akhir	Proses registrasi berhasil

2. **Activity Diagram**

a. Activity diagram Login

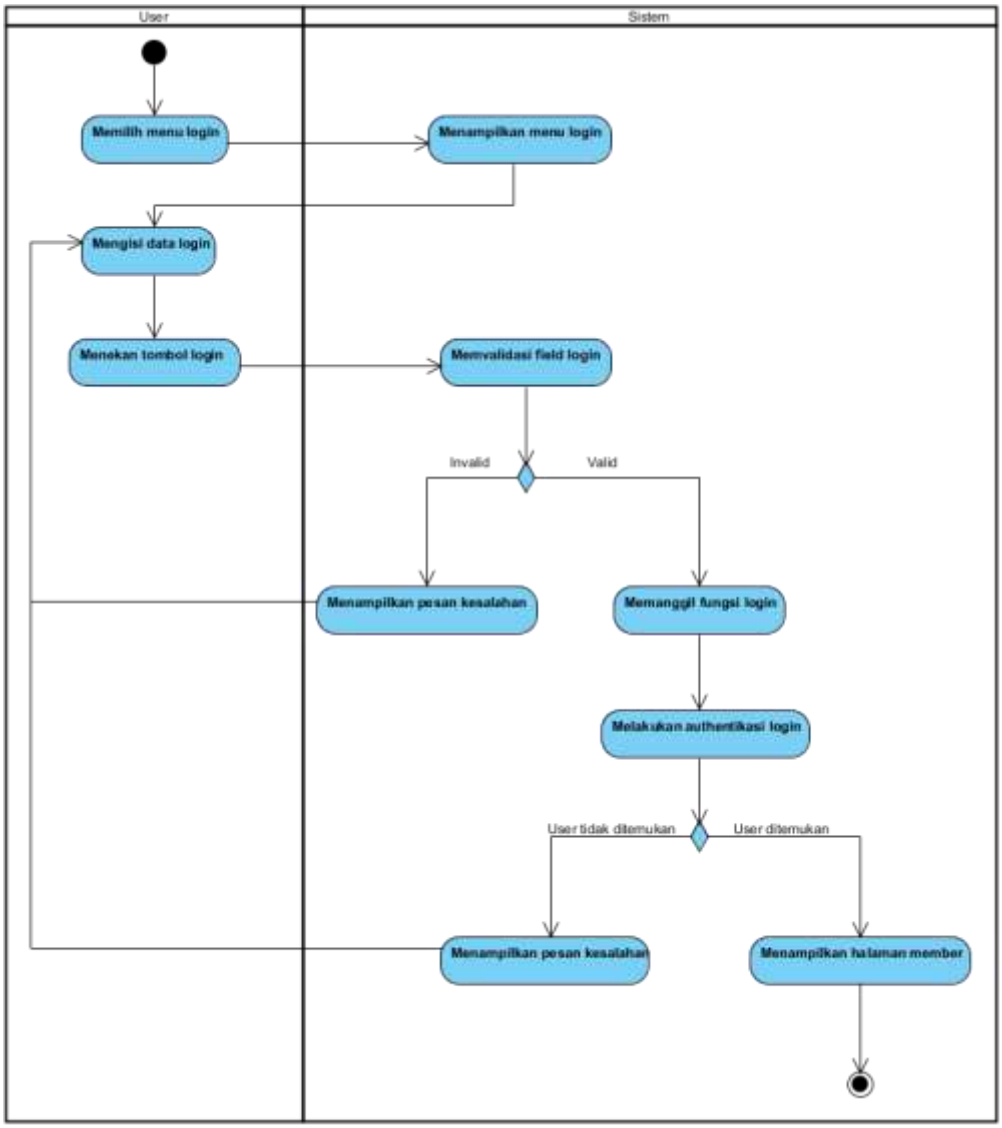
Aktivitas *login* merupakan aktivitas yang dilakukan oleh user biasa atau bukan user member untuk melakukan *login* sebagai member pemberi kerja atau pencari kerja Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu *login*, pada aktivitas ini user menjalankan menu *login* yang ada pada menu utama.
2. Menampilkan menu login, pada aktivitas ini sistem menampilkan halaman *login* kepada user.
3. Mengisi data *login*, pada aktivitas ini user memasukkan *username* dan *password* serta memilih tipe tipe user untuk melakukan *login* sebagai pemberi kerja atau pencari kerja.
4. Menekan tombol login, pada aktivitas ini user menekan tombol *login* untuk mengeksekusi proses *login*.
5. Memvalidasi field *login*, pada aktivitas ini sistem akan memeriksa kolom *username* dan *password* sudah diisi atau belum.
6. Menampilkan pesan kesalahan, aktivitas ini akan ditampilkan oleh sistem jika *username* dan *password* yang dimasukkan tidak valid.
7. Memanggil fungsi *login*, pada aktivitas ini jika parameter input pada *form login* valid maka sistem akan memanggil fungsi login.
8. Melakukan authentikasi *login*, pada aktivitas ini sistem akan melakukan authentikasi *username* dan *password* dengan nilai *return* dari *webservice*, jika *username* dan *password* cocok atau ditemukan maka sistem akan

menampilkan halaman member, sebaliknya sistem akan menampilkan pesan kesalahan jika *username* dan *password* tidak ditemukan.

9. Menampilkan pesan kesalahan, aktivitas ini akan ditampilkan oleh sistem jika *username* dan *password* yang dimasukkan tidak ditemukan.
10. Menampilkan halaman member, pada aktivitas ini sistem akan menampilkan halaman member jika proses *login* berhasil.

Seluruh aktivitas login terlihat pada *activity diagram* gambar III.2.



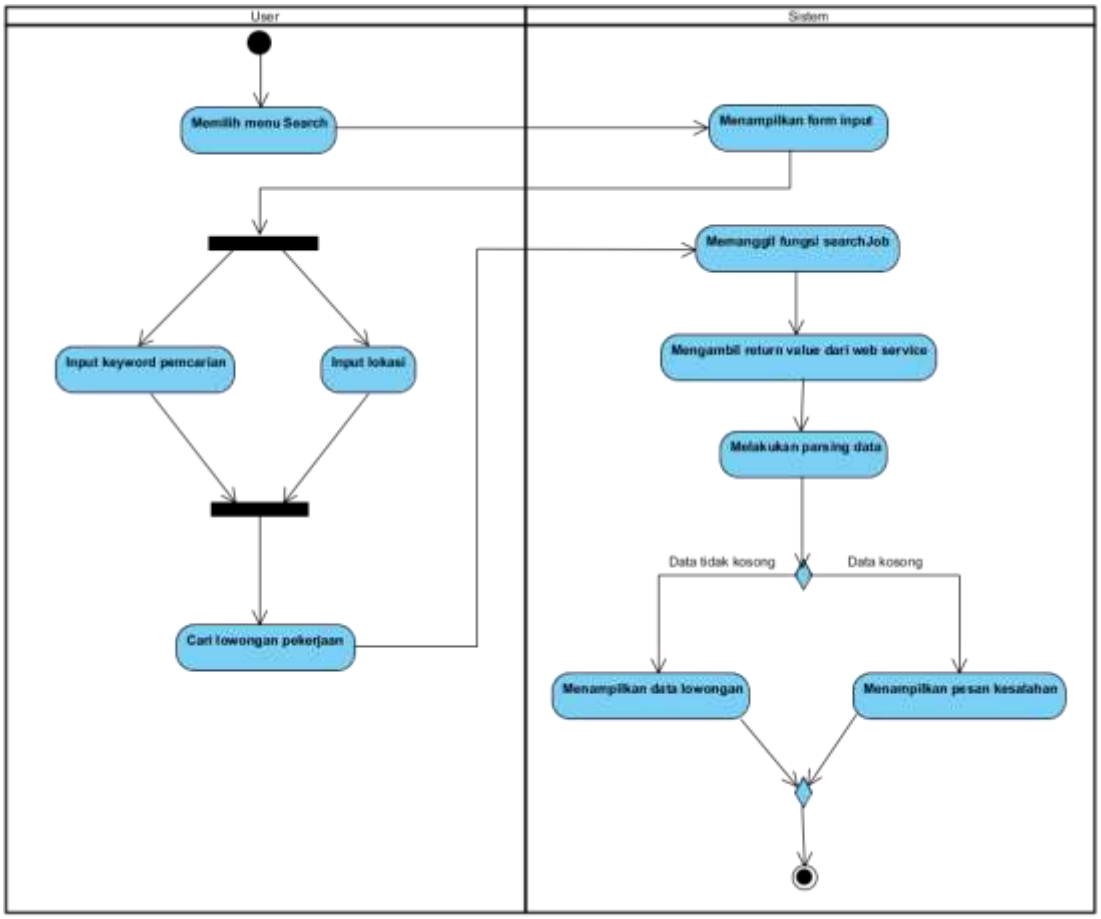
Gambar III. 2 Activity Diagram login

b. Activity diagram Pencarian Lowongan

Aktivitas Pencarian Lowongan merupakan aktivitas yang dilakukan oleh user untuk mencari data lowongan pekerjaan yang ada pada database dengan *web service*. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu *search*, pada aktivitas ini user menjalankan menu search yang ada pada menu utama.
2. Menginputkan parameter pencarian pada form input.
3. Memanggil fungsi searchJob, pada aktivitas ini sistem memanggil fungsi searchJob.
4. Mengambil *return value* dari *web service*, pada aktivitas ini sistem mengambil nilai yang di kembalikan oleh *web service* berupa hasil dari eksekusi method pada *web service*.
5. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
6. Menampilkan data lowongan pekerjaan, pada aktivitas ini sistem menampilkan hasil parsing kedalam bentuk *list view*.

Seluruh aktivitas Pencarian Lowongan terlihat pada *activity diagram* gambar III.3.



Gambar III. 3 Activity Diagram Pencarian lowongan.

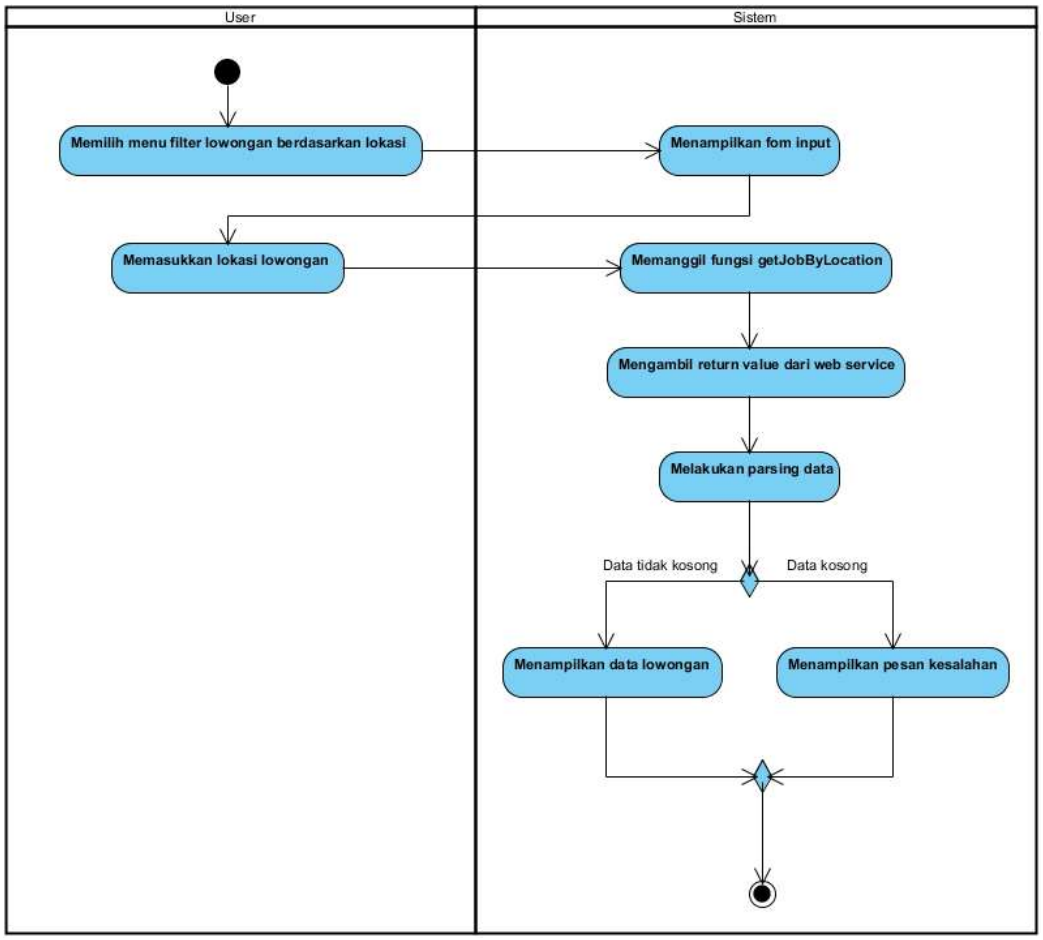
c. Activity diagram Filter Lowongan Berdasarkan Lokasi

Aktivitas Filter Lowongan Berdasarkan Lokasi merupakan aktivitas yang dilakukan oleh user untuk memfilter data lowongan pekerjaan berdasarkan lokasi lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu *filter* lowongan berdasarkan lokasi, pada aktivitas ini user memilih menu untuk memfilter data berdasarkan lokasi lowongan pekerjaan.

2. Menampilkan menu lokasi lowongan pekerjaan, pada aktivitas ini sistem menampilkan menu lokasi yang harus dipilih oleh user.
3. Memilih lokasi, pada aktivitas ini user memilih lokasi lowongan pekerjaan untuk memfilter data lowongan.
4. Memanggil fungsi byLocation, pada aktivitas ini sistem memanggil fungsi byLocation yang ada pada *web service*.
5. Mengambil *return value* dari web service, pada aktivitas ini sistem mengambil nilai yang dikembalikan oleh web service berupa hasil dari eksekusi method pada web service.
6. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
7. Menampilkan datalowongan pekerjaan, pada aktivitas ini sistem menampilkan hasil parsing kedalam bentuk list view jika data hasil parsing tidak kosong.
8. Menampilkan pesan kesalahan, pada aktivitas ini sistem menampilkan pesan kesalahan jika data hasil parsing kosong.

Seluruh aktivitas Filter Lowongan Berdasarkan Lokasi terlihat pada *activity diagram* gambar III.4.



Gambar III. 4 Activity Diagram Filter Lowongan Berdasarkan Lokasi.

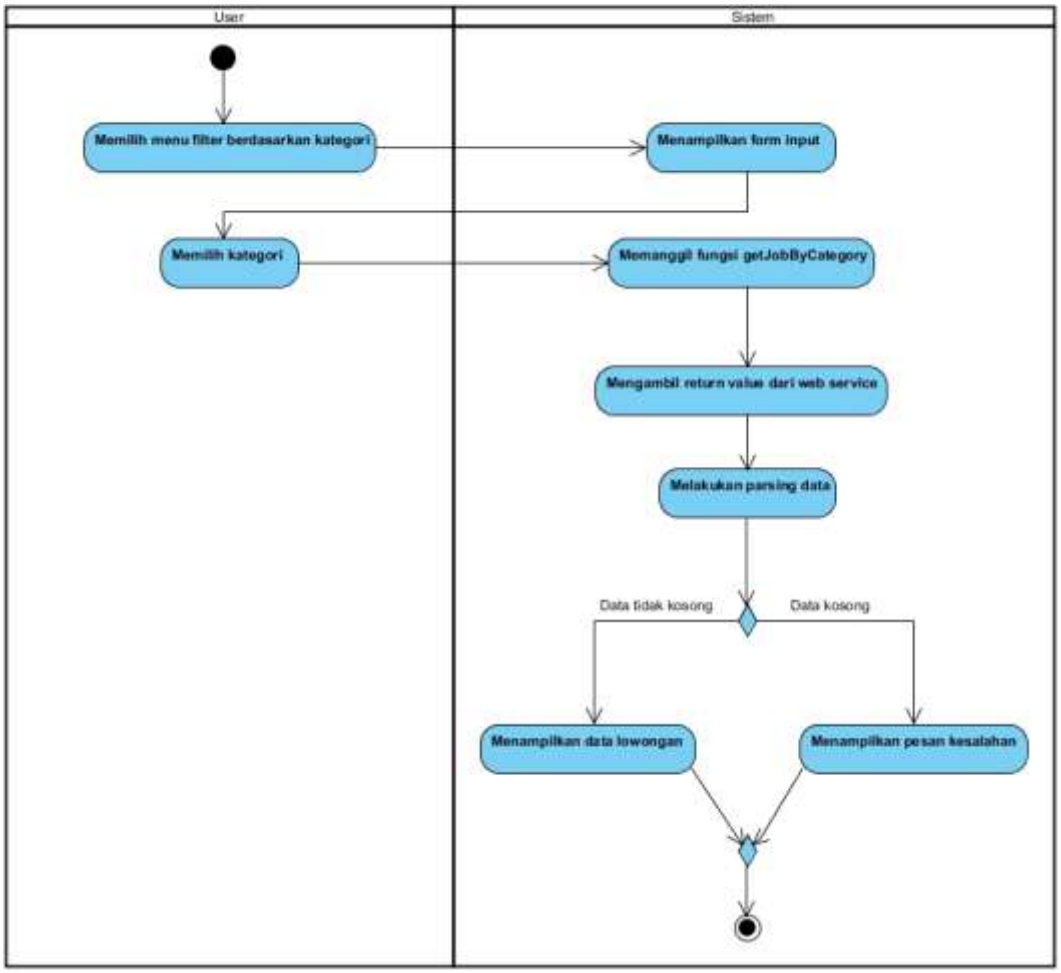
d. Activity diagram Filter Lowongan Berdasarkan Kategori

Aktivitas Filter Lowongan Berdasarkan Kategori merupakan aktivitas yang dilakukan oleh user untuk memfilter data lowongan pekerjaan berdasarkan kategori lowongan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu *filter* lowongan berdasarkan kategori, pada aktivitas ini user memilih menu untuk memfilter data berdasarkan kategori lowongan.

2. Menampilkan menu kategori lowongan pekerjaan, pada aktivitas ini sistem menampilkan menu kategori yang harus dipilih oleh user.
3. Memilih kategori, pada aktivitas ini user memilih kategori lowongan pekerjaan untuk memfilter data lowongan.
4. Memanggil fungsi byCategory, pada aktivitas ini sistem memanggil fungsi byCategory yang ada pada *web service*.
5. Mengambil *return value* dari web service, pada aktivitas ini sistem mengambil nilai yang dikembalikan oleh web service berupa hasil dari eksekusi method pada web service.
6. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
7. Menampilkan data lowongan pekerjaan, pada aktivitas ini sistem menampilkan hasil parsing kedalam bentuk *list view* jika data hasil parsing tidak kosong.
8. Menampilkan pesan kesalahan, pada aktivitas ini sistem menampilkan pesan kesalahan jika data hasil *parsing* kosong.

Seluruh aktivitas Filter Lowongan Berdasarkan Kategori terlihat pada *activity diagram* gambar III.5.



Gambar III. 5 Activity Diagram Lowongan Berdasarkan Kategori

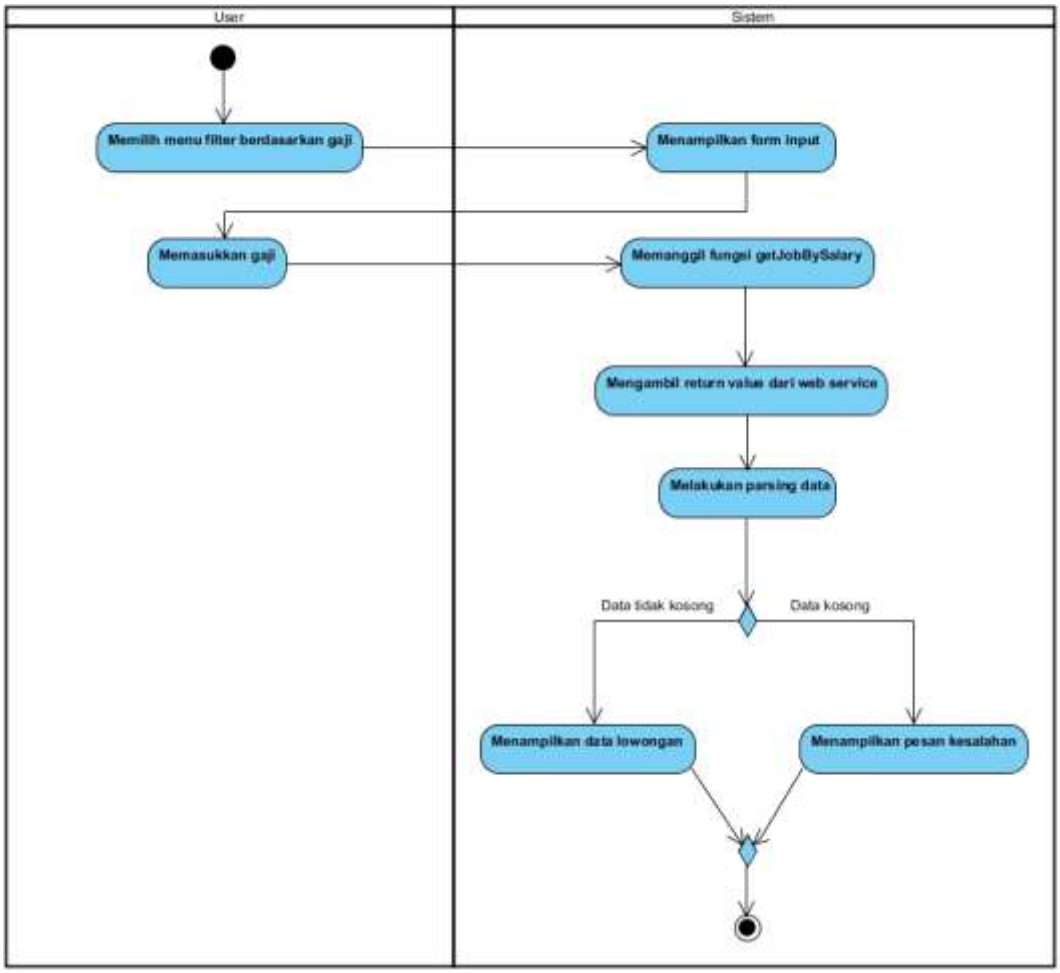
e. Activity diagram Filter Lowongan Berdasarkan Gaji

Aktivitas Filter Lowongan Berdasarkan Gaji merupakan aktivitas yang dilakukan oleh user untuk memfilter data lowongan pekerjaan berdasarkan rentang gaji lowongan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu *filter* lowongan berdasarkan gaji, pada aktivitas ini user memilih menu untuk memfilter data berdasarkan gaji lowongan pekerjaan.

2. Menampilkan menu gaji lowongan pekerjaan, pada aktivitas ini sistem menampilkan menu gaji yang harus dipilih oleh user.
3. Memilih gaji, pada aktivitas ini user memilih gaji lowongan pekerjaan untuk memfilter data lowongan.
4. Memanggil fungsi bySalary, pada aktivitas ini sistem memanggil fungsi bySalary yang ada pada *web service*.
5. Mengambil *return value* dari *web service*, pada aktivitas ini sistem mengambil nilai yang dikembalikan oleh *web service* berupa hasil dari eksekusi method pada *web service*.
6. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
7. Menampilkan data lowongan pekerjaan, pada aktivitas ini sistem menampilkan hasil parsing kedalam bentuk *list view* jika data hasil *parsing* tidak kosong.
8. Menampilkan pesan kesalahan, pada aktivitas ini sistem menampilkan pesan kesalahan jika data hasil *parsing* kosong.

Seluruh aktivitas Filter Lowongan Berdasarkan Gaji terlihat pada *activity diagram* gambar III.6.



Gambar III. 6 Activity Diagram Filter Lowongan Berdasarkan Gaji

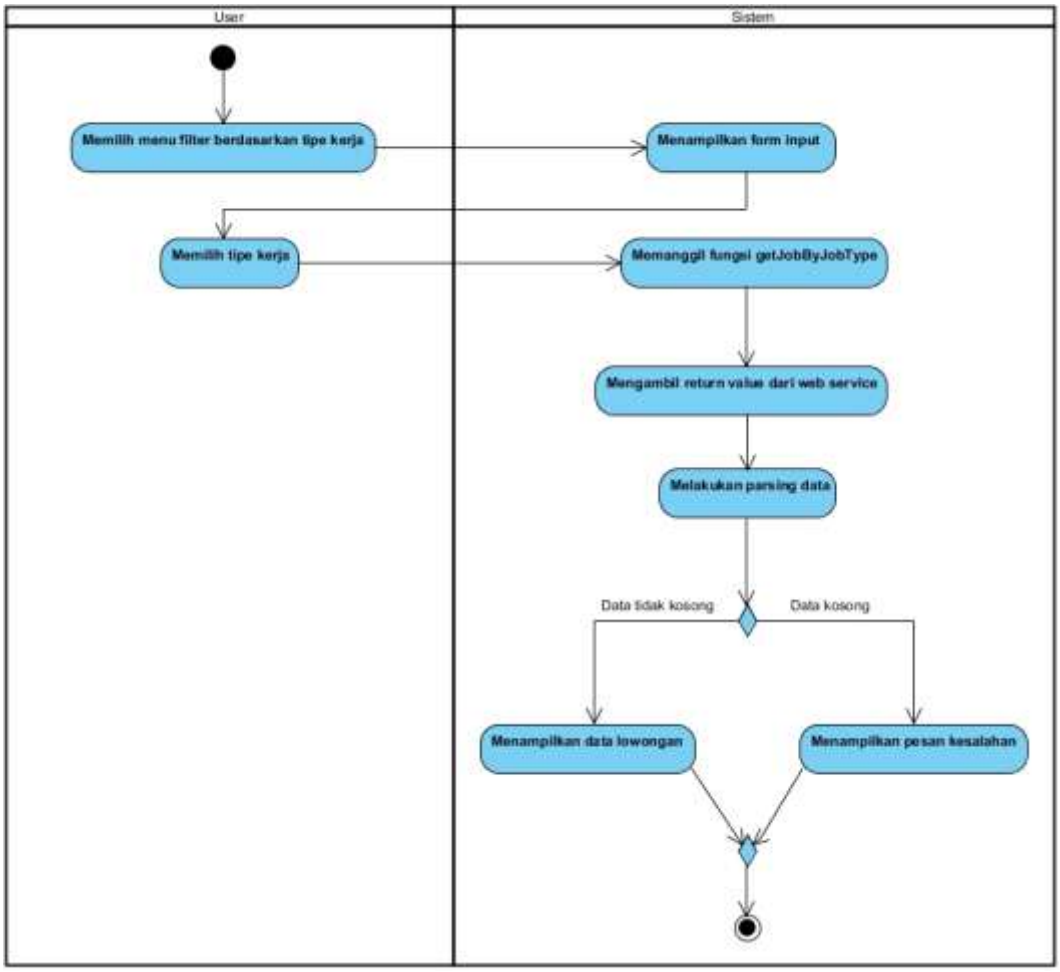
f. Activity diagram Filter Lowongan Berdasarkan Tipe Kerja

Aktivitas Filter Lowongan Berdasarkan Tipe Kerja merupakan aktivitas yang dilakukan oleh user untuk memfilter data lowongan pekerjaan berdasarkan tipe kerja lowongan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu *filter* lowongan berdasarkan tipe kerja, pada aktivitas ini user memilih menu untuk memfilter data berdasarkan tipe kerja lowongan pekerjaan.

2. Menampilkan menu tipe kerja lowongan pekerjaan, pada aktivitas ini sistem menampilkan menu tipe kerja yang harus dipilih oleh user.
3. Memilih tipe kerja, pada aktivitas ini user memilih tipe kerja lowongan pekerjaan untuk memfilter data lowongan.
4. Memanggil fungsi byJobType, pada aktivitas ini sistem memanggil fungsi byJobType yang ada pada *web service*.
5. Mengambil *return value* dari web service, pada aktivitas ini sistem mengambil nilai yang dikembalikan oleh *web service* berupa hasil dari eksekusi method pada *web service*.
6. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
7. Menampilkan data lowongan pekerjaan, pada aktivitas ini sistem menampilkan hasil parsing kedalam bentuk list view jika data hasil *parsing* tidak kosong.
8. Menampilkan pesan kesalahan, pada aktivitas ini sistem menampilkan pesan kesalahan jika data hasil *parsing* kosong.

Seluruh aktivitas Filter Lowongan Berdasarkan Tipe Kerja terlihat pada *activity diagram* gambar III.7.



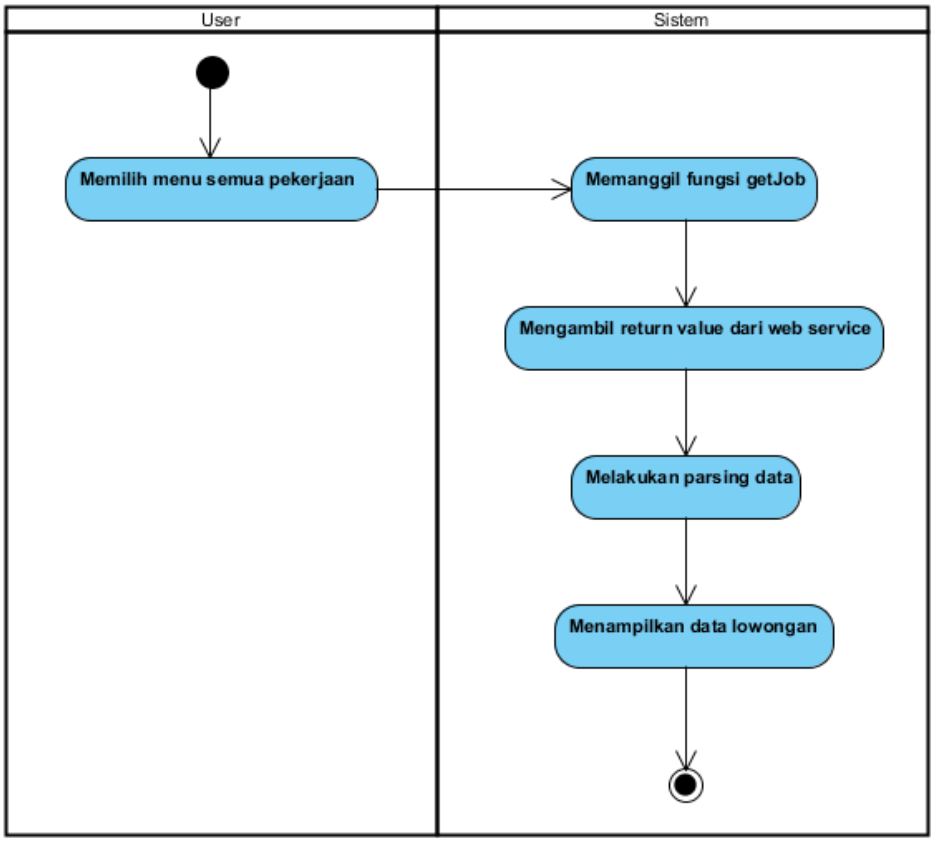
Gambar III. 7 Activity Diagram Filter Lowongan Berdasarkan Tipe Kerja

g. Activity diagram Lihat Daftar Lowongan

Aktivitas Lihat Daftar Lowongan merupakan aktivitas yang dilakukan oleh user untuk melihat daftar lowongan pekerjaan yang ada pada *database* dengan *web service*. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Menjalankan menu semua pekerjaan, pada aktivitas ini user menjalankan menu semua pekerjaan yang ada pada sistem.
2. Memanggil fungsi getJob, pada aktivitas ini sistem memanggil fungsi getJob yang ada pada *web service*.
3. Mengambil *return value* dari *web service*, pada aktivitas ini sistem mengambil nilai yang dikembalikan oleh *web service* berupa hasil dari eksekusi method pada *web service*.
4. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
5. Menampilkan data lowongan pekerjaan, pada aktivitas ini sistem menampilkan hasil *parsing* ke dalam bentuk *list view*.

Seluruh aktivitas Lihat Daftar Lowongan terlihat pada *activity diagram* gambar III.8.



Gambar III. 8 Activity Diagram Lihat Daftar Lowongan.

h. Activity diagram Lihat Detil Lowongan

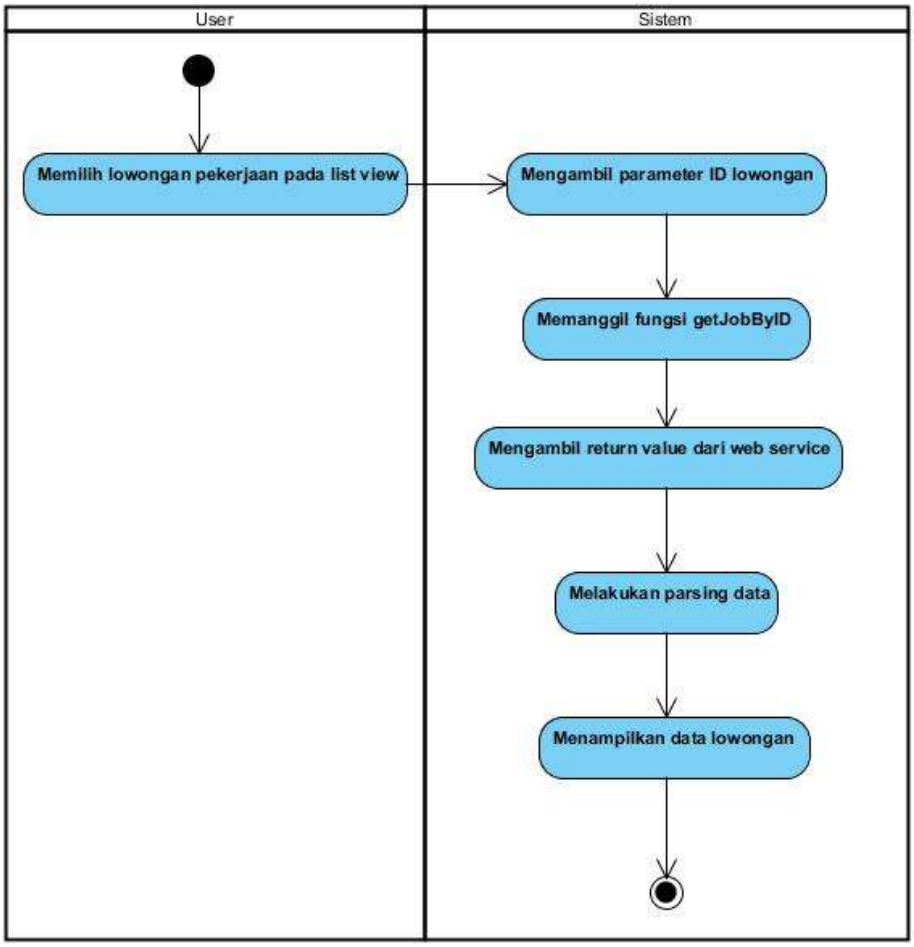
Aktivitas Lihat Detil Lowongan merupakan aktivitas yang dilakukan oleh user untuk melihat detail dari data lowongan pekerjaan berdasarkan id lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih lowongan pekerjaan pada *list view*, pada aktivitas ini user memilih lowongan pekerjaan untuk menampilkan detail dari lowongan pekerjaan yang dipilih.
2. Mengambil parameter id lowongan pekerjaan, pada aktivitas ini sistem mengambil id lowongan pekerjaan, dimana id lowongan pekerjaan

didapat ketika user memilih lowongan pekerjaan sebelumnya pada *list view*.

3. Memanggil fungsi getJobtByID, pada aktivitas ini sistem memanggil fungsi getJobtByID yang ada pada *web service*.
4. Mengambil *return value* dari web service, pada aktivitas ini sistem mengambil nilai yang dikembalikan oleh *web service* berupa hasil dari eksekusi method pada *web service*.
5. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
6. Menampilkan data lowongan pekerjaan, pada aktivitas ini sistem menampilkan hasil *parsing*.

Seluruh aktivitas Lihat Detil Lowongan terlihat pada *activity diagram* gambar III.9.



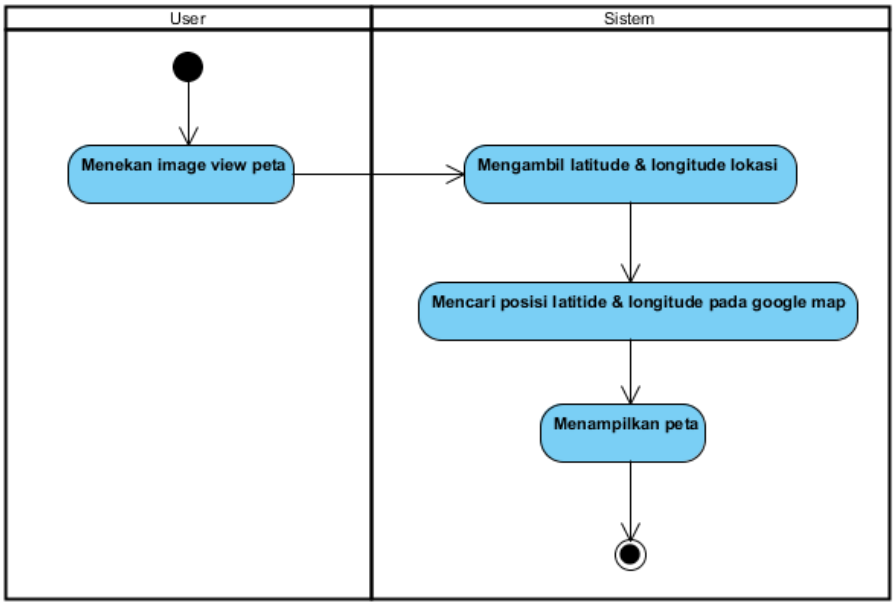
Gambar III. 9 Activity Diagram Lihat Detil Lowongan.

i. Activity diagram Lihat Peta

Aktivitas Lihat Peta merupakan aktivitas yang dilakukan oleh user untuk melihat lokasi penempatan dari lowongan pekerjaan di peta digital (google map). Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Menekan *image view* peta, pada aktivitas ini user menekan sebuah ikon image view peta yang terdapat di sebelah lokasi penempatan lowongan pekerjaan.
2. Mengambil latitude dan longitude, pada aktivitas ini sistem mengambil latitude dan longitude dari lokasi penempatan lowongan pekerjaan.
3. Mencari posisi latitude dan longitude pada google map, pada aktivitas ini sistem akan mencari lokasi berdasarkan latitude dan longitude dari lokasi penempatan lowongan pekerjaan.
4. Menampilkan peta, pada aktivitas ini sistem akan menampilkan peta sesuai nilai dari latitude dan longitude.

Seluruh aktivitas Lihat Peta terlihat pada *activity diagram* gambar III.10.



Gambar III. 10 Activity Diagram Lihat Peta

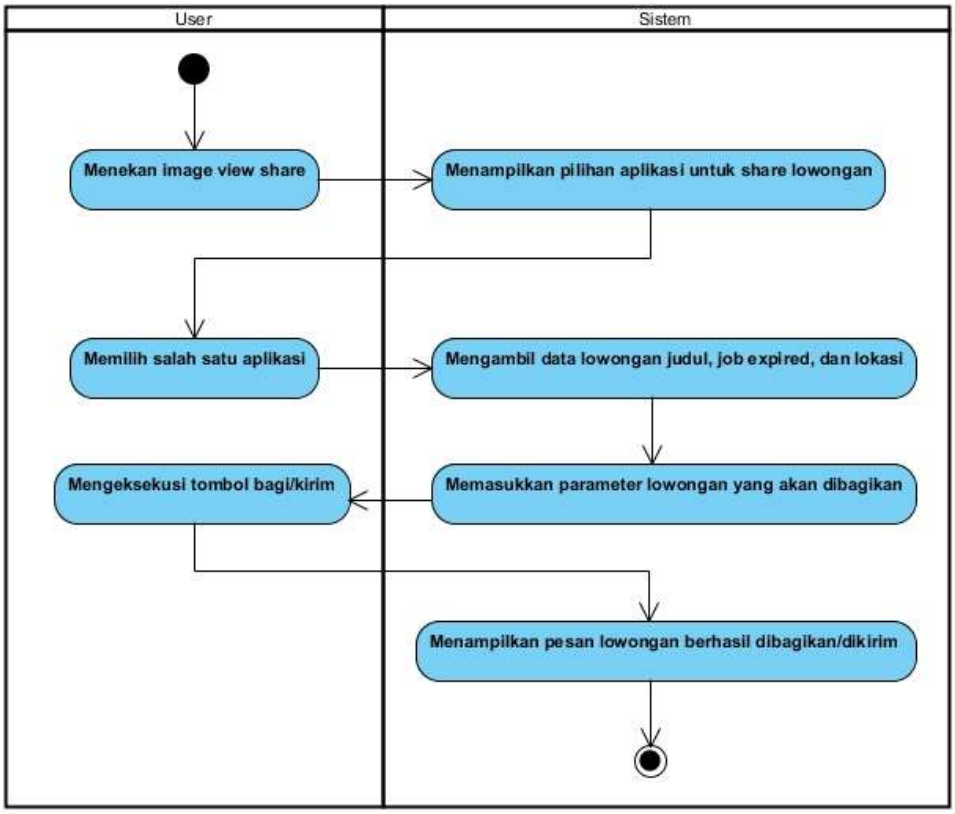
j. Activity diagram Membagikan Info Lowongan

Aktivitas Membagikan Info Lowongan merupakan aktivitas yang dilakukan oleh user untuk membagikan info tentang lowongan pekerjaan ini melalui media *social network* ataupun email. Dalam aktivitas ini terdapat beberapa aktivitas yaitu:

1. Menekan image view *share*, pada aktivitas ini user menekan sebuah ikon image view *share* yang terdapat di list view dari detail lowongan pekerjaan.
2. Menampilkan pilihan aplikasi untuk share lowongan, pada aktivitas ini sistem akan menampilkan aplikasi yang terdapat di device user yang dapat digunakan sebagai media untuk membagikan info lowongan pekerjaan.
3. Memilih salah satu aplikasi, pada aktivitas user memilih salah satu media *share* yang akan digunakan untuk membagikan info lowongan pekerjaan.
4. Mengambil data lowongan judul, *job expired*, dan lokasi, pada aktivitas ini sistem mengambil beberapa data lowongan pekerjaan yaitu judul lowongan, tanggal job *expired*, dan lokasi yang akan dibagikan.
5. Memasukkan parameter lowongan yang akan dibagikan, pada aktivitas ini sistem memasukkan parameter lowongan yang akan dibagikan ke dalam aplikasi.
6. Mengeksekusi tombol bagi/kirim, pada aktivitas ini user mengeksekusi proses bagi/kirim info lowongan pekerjaan dengan menekan tombol bagi/kirim.

7. Menampilkan pesan lowongan berhasil dibagikan/dikirim, pada aktivitas ini aplikasi akan menampilkan pesan bahwa lowongan pekerjaan berhasil dibagikan atau dikirim.

Seluruh aktivitas Membagikan Info Lowongan terlihat pada *activity diagram* gambar III.11.



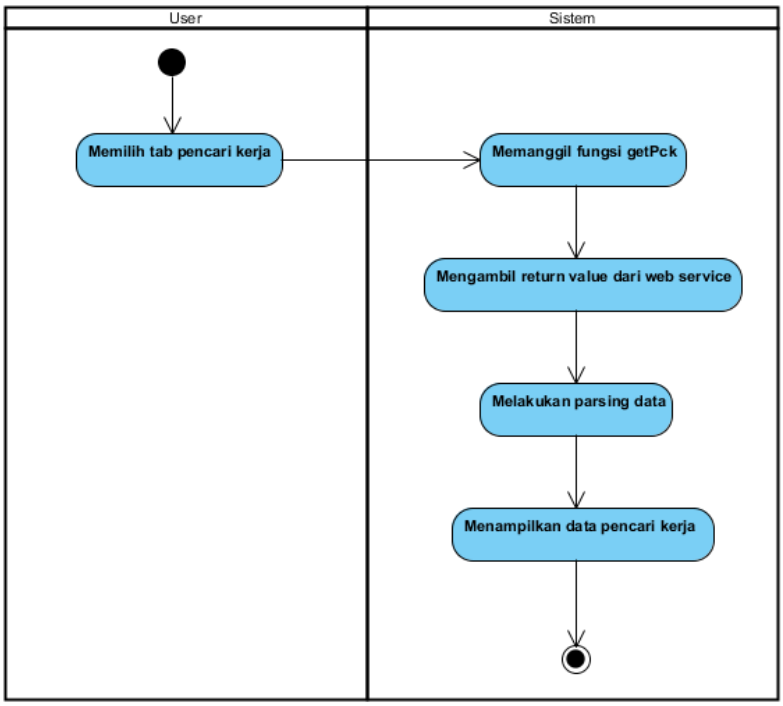
Gambar III. 11 Activity Diagram Membagikan Info Lowongan

k. Activity diagram Lihat Daftar Pencari Kerja

Aktivitas Lihat Daftar Pencari Kerja merupakan aktivitas yang dilakukan oleh member pemberi kerja untuk mangambil data pencari kerja yang ada pada *database* dengan *web service*. Dalam aktivitas ini terdapat beberapa aktivitas yaitu:

1. Memilih tab pencari kerja, pada aktivitas ini user memilih menu tab pencari kerja yang ada pada sistem.
2. Memanggil fungsi getPck, pada aktivitas ini sistem memanggil fungsi getPck yang ada pada *web service*.
3. Mengambil *return value* dari *web service*, pada aktivitas ini sistem mengambil nilai yang dikembalikan oleh *web service* berupa hasil dari eksekusi method pada *web service*.
4. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
5. Menampilkan data pencari kerja, pada aktivitas ini sistem menampilkan hasil *parsing* ke dalam bentuk *list view*.

Seluruh aktivitas Lihat Daftar Pencari Kerja terlihat pada *activity diagram* gambar III.12.



Gambar III. 12 Activity Diagram Lihat Daftar Pencari Kerja

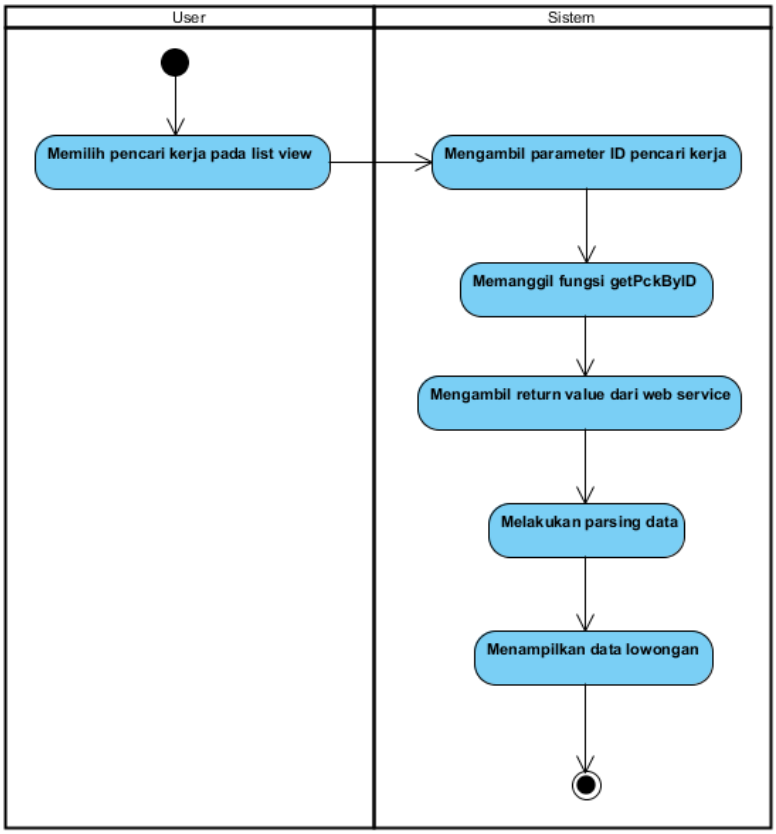
l. Activity diagram Lihat Detil Pencari Kerja

Aktivitas Lihat Detil Pencari Kerja merupakan aktivitas yang dilakukan oleh user untuk mangambil detail dari data pencari kerja yang ada pada *web service* berdasarkan id pencari kerja. Dalam aktivitas ini terdapat beberapa aktivitas yaitu:

1. Memilih pencari kerja pada *list view*, pada aktivitas ini user memilih pencari kerja untuk menampilkan detail dari lowongan pekerjaan yang dipilih.
2. Mengambil parameter id pencari kerja, pada aktivitas ini sistem mengambil id pencari kerja, dimana id pencari kerja didapat ketika user memilih pencari kerja sebelumnya pada *list view*

3. Memanggil fungsi getPckByID, pada aktivitas ini sistem memanggil fungsi getPckByID yang ada pada *web service.*
4. Mengambil *return value* dari web service, pada aktivitas ini sistem mengambil nilai yang dikembalikan oleh *web service* berupa hasil dari eksekusi method pada *web service*.
5. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
6. Menampilkan data lowongan pekerjaan, pada aktivitas ini sistem menampilkan hasil *parsing*.

Seluruh aktivitas Lihat Detil Pencari Kerja terlihat pada *activity diagram* gambar III.13.



Gambar III. 13 Activity Diagram Lihat Detil Pencari Kerja

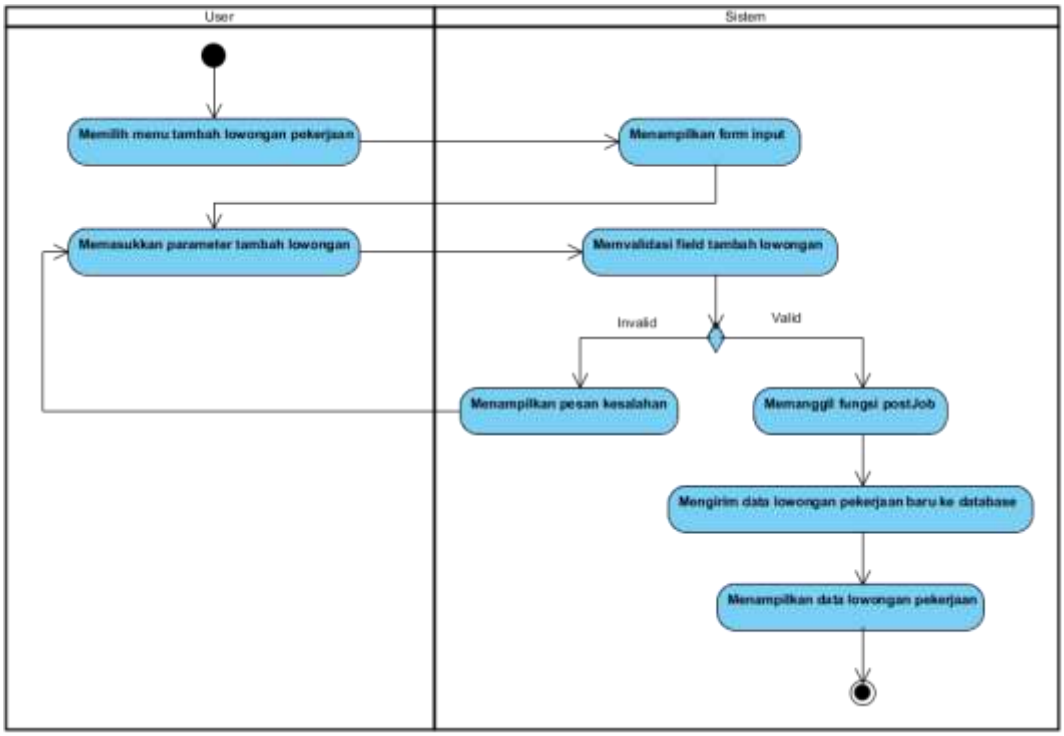
m. Activity diagram Pembuatan Lowongan Pekerjaan Baru

Aktivitas Pembuatan Lowongan Pekerjaan Baru merupakan aktivitas yang dilakukan oleh member pemberi kerja untuk menambahkan data lowongan pekerjaan baru ke dalam database melalui method yang disediakan oleh *web service*. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu tambah lowongan pekerjaan, pada aktivitas ini *user* memilih menu tambah lowongan yang terdapat pada sistem.
2. Menampilkan form *input*, pada aktivitas ini sistem akan menampilkan halaman form tambah lowongan pekerjaan baru.
3. Memasukkan parameter tambah lowongan, pada aktivitas ini *user* akan memasukkan parameter pada *field* tambah lowongan pekerjaan baru.
4. Memvalidasi *field* tambah lowongan, pada aktivitas ini sistem akan memeriksa parameter yang telah dimasukkan oleh *user* apakah telah sesuai dengan ketentuan parameter *input* apa belum.
5. Memanggil fungsi *postJob*, pada aktivitas jika parameter tambah lowongan pekerjaan baru valid maka sistem akan memanggil fungsi *postJob*.
6. Menampilkan pesan kesalahan, pada aktivitas ini sistem menampilkan pesan kesalahan jika parameter tambah lowongan pekerjaan baru tidak valid.
7. Mengirim data lowongan pekerjaan baru ke *database*, pada aktivitas ini sistem akan mengirim data lowongan pekerjaan baru ke dalam *database*.

8. Menampilkan data lowongan pekerjaan, pada aktivitas ini sistem akan menampilkan daftar lowongan pekerjaan termasuk lowongan pekerjaan yang baru saja ditambahkan ke database.

Seluruh aktivitas Pembuatan Lamaran Pekerjaan terlihat pada *activity diagram* gambar III.14.



Gambar III. 14 Activity Diagram Pembuatan Lowongan Pekerjaan Baru

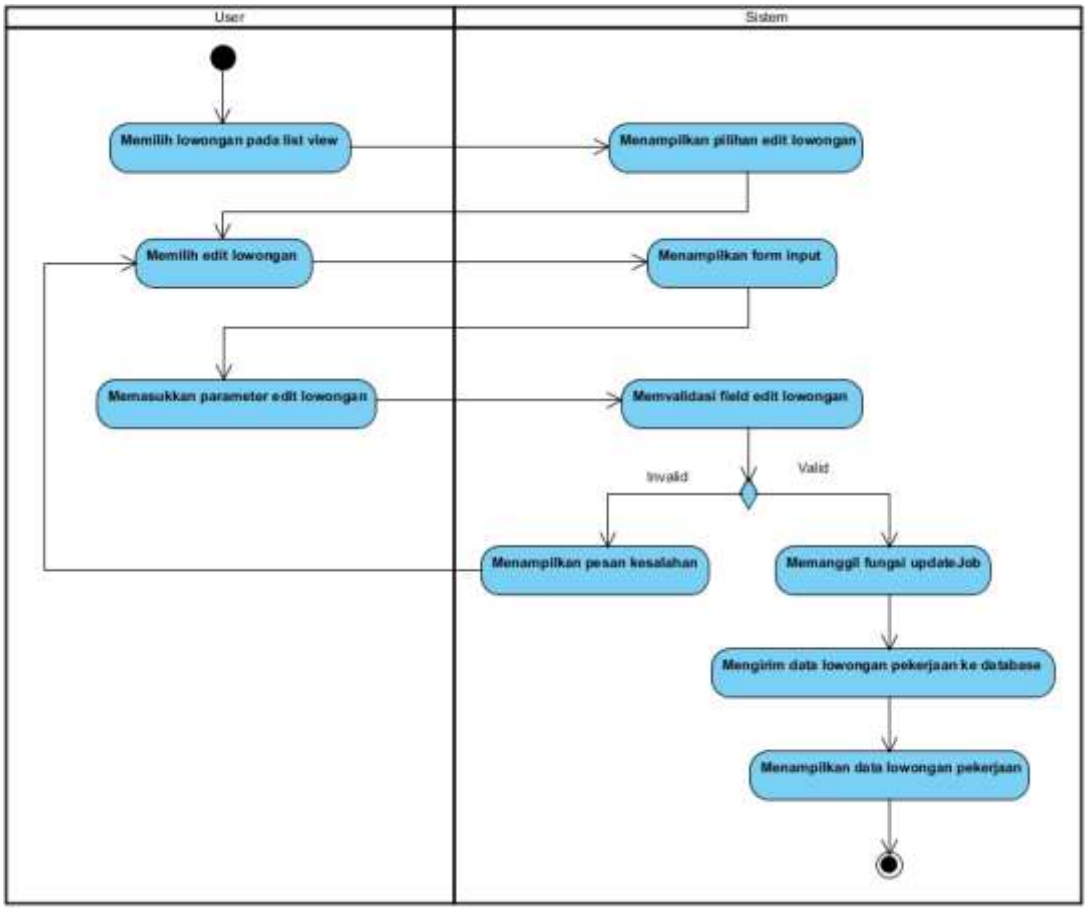
n. Activity diagram Perubahan Data Lowongan

Aktivitas Perubahan Data Lowongan merupakan aktivitas yang dilakukan oleh member pemberi kerja untuk mengubah atau edit data lowongan pekerjaan ke dalam database melalui method yang disediakan oleh *web service*. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih lowongan pada *list view*, pada aktivitas ini *user* memilih lowongan yang terdapat pada *list view*.

2. Menampilkan pilihan edit lowongan, pada aktivitas ini sistem akan menampilkan pilihan untuk melakukan *edit* pada lowongan pekerjaan yang dipilih.
3. Memilih *edit* lowongan, pada aktivitas ini user pemberi kerja memilih untuk melakukan *edit* lowongan pekerjaan.
4. Menampilkan form *input*, pada aktivitas ini sistem akan menampilkan halaman form *edit* lowongan pekerjaan.
5. Memasukkan parameter *edit* lowongan, pada aktivitas ini *user* akan memasukkan parameter pada *field edit* lowongan pekerjaan.
6. Memvalidasi *field* edit lowongan, pada aktivitas ini sistem akan memeriksa parameter yang telah dimasukkan oleh *user* apakah telah sesuai dengan ketentuan parameter *input* apa belum.
7. Memanggil fungsi *updateJob*, pada aktivitas jika parameter *edit* lowongan pekerjaan valid maka sistem akan memanggil fungsi *updateJob*.
8. Menampilkan pesan kesalahan, pada aktivitas ini sistem menampilkan pesan kesalahan jika parameter *edit* lowongan pekerjaan tidak valid.
9. Mengirim data lowongan pekerjaan ke *database*, pada aktivitas ini sistem akan mengirim data lowongan pekerjaan ke dalam *database*.
10. Menampilkan data lowongan pekerjaan, pada aktivitas ini sistem akan menampilkan daftar lowongan pekerjaan termasuk lowongan pekerjaan yang baru saja di-*edit*.

Seluruh aktivitas Perubahan Data Lowongan terlihat pada *activity diagram* gambar III.15.



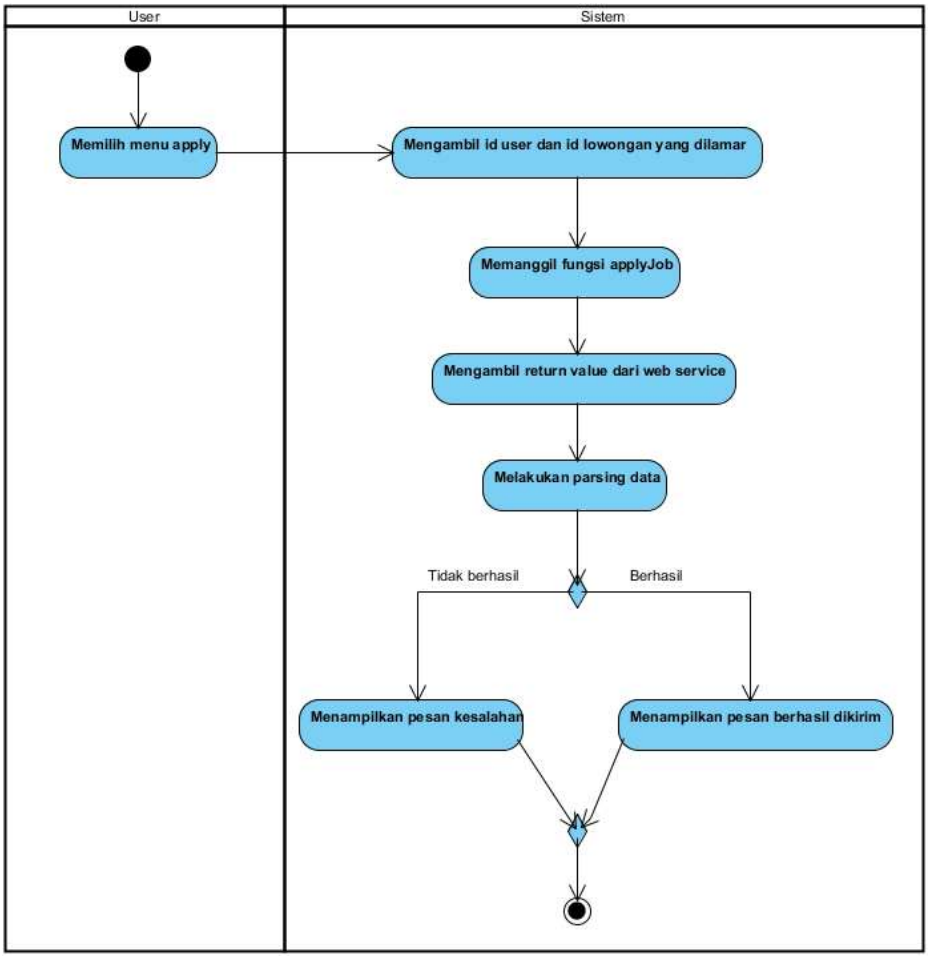
Gambar III. 15 Activity Diagram Perubahan Data Lowongan

o. Activity diagram Lamaran Lowongan Pekerjaan

Aktivitas Lamaran Lowongan Pekerjaan merupakan aktivitas yang dilakukan oleh member pencari kerja untuk melakukan proses lamar kerja kepada sebuah lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu *apply*, pada aktivitas ini user memilih menu *apply* yang terdapat pada sistem.
2. Mengambil id user dan id lowongan yang dilamar, pada aktivitas ini sistem mengambil id user pencari kerja dan id lowongan pekerjaan yang dilamar sebagai parameter input untuk memanggil fungsi *applyJob*.
3. Memanggil fungsi *applyJob*, pada aktivitas ini sistem akan memanggil fungsi *applyJob.*
4. Memasukkan data pencari kerja dan email perusahaan, pada aktivitas ini sistem akan memasukkan data pencari kerja sebagai isi email dan alamat email perusahaan sebagai email tujuan bersamaan dengan menampilkan aplikasi email kepada *user*.
5. Mengambil *return value* dari *web service*, pada aktivitas ini sistem mengambil nilai yang dikembalikan oleh *web service* berupa hasil dari eksekusi method pada *web service*.
6. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
7. Menampilkan pesan berhasil dikirim, pada aktivitas ini sistem menampilkan pesan pemberitahuan kalau pesan lamaran telah berhasil dikirimkan.
8. Menampilkan pesan kesalahan, pada aktivitas ini sistem akan menampilkan pesan kesalahan jika member pencari kerja sudah pernah melakukan proses lamar pada lowongan pekerjaan yang sama.

Seluruh aktivitas Lamaran Lowongan Pekerjaan terlihat pada *activity diagram* gambar III.16.



Gambar III. 16 Activity Diagram Lamaran Lowongan Pekerjaan

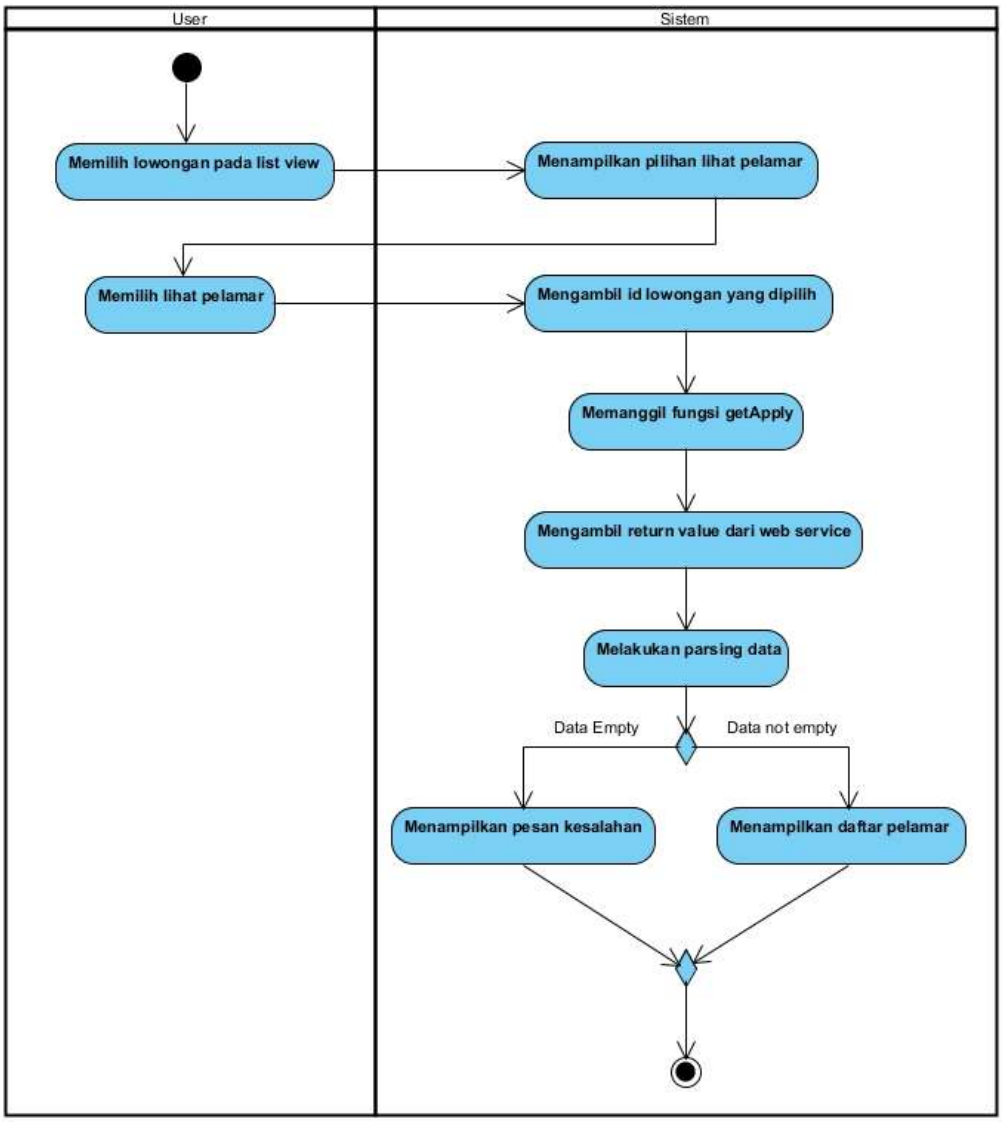
p. Activity diagram Lihat Pelamar

Aktivitas Lihat Pelamar merupakan aktivitas yang dilakukan oleh member pemberi kerja untuk melihat para pelamar yang telah melakukan lamaran ke lowongan pekerjaan yang dibuat oleh perusahaannya. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih lowongan pada *list view*, pada aktivitas ini *user* memilih lowongan yang terdapat pada *list view*.

2. Menampilkan pilihan lihat pelamar, pada aktivitas ini sistem akan menampilkan pilihan untuk melihat pelamar dari lowongan pekerjaan yang dipilih.
3. Memilih lihat pelamar, pada aktivitas ini user pemberi kerja memilih untuk melihat pelamar.
4. Memanggil fungsi *getApply*, pada aktivitas jika parameter *edit* lowongan pekerjaan valid maka sistem akan memanggil fungsi *updateJob*.
5. Mengambil *return value* dari web service, pada aktivitas ini sistem mengambil nilai yang dikembalikan oleh *web service* berupa hasil dari eksekusi method pada *web service.*
6. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
7. Menampilkan pesan kesalahan, pada aktivitas ini sistem menampilkan pesan kesalahan jika data pelamar kosong.
8. Menampilkan daftar pelamar, pada aktivitas ini sistem akan menampilkan daftar pelamar dari lowongan pekerjaan yang dipilih.

Seluruh aktivitas Lihat Pelamar terlihat pada *activity diagram* gambar III.17.



Gambar III. 17 Activity Diagram Lihat Lamaran

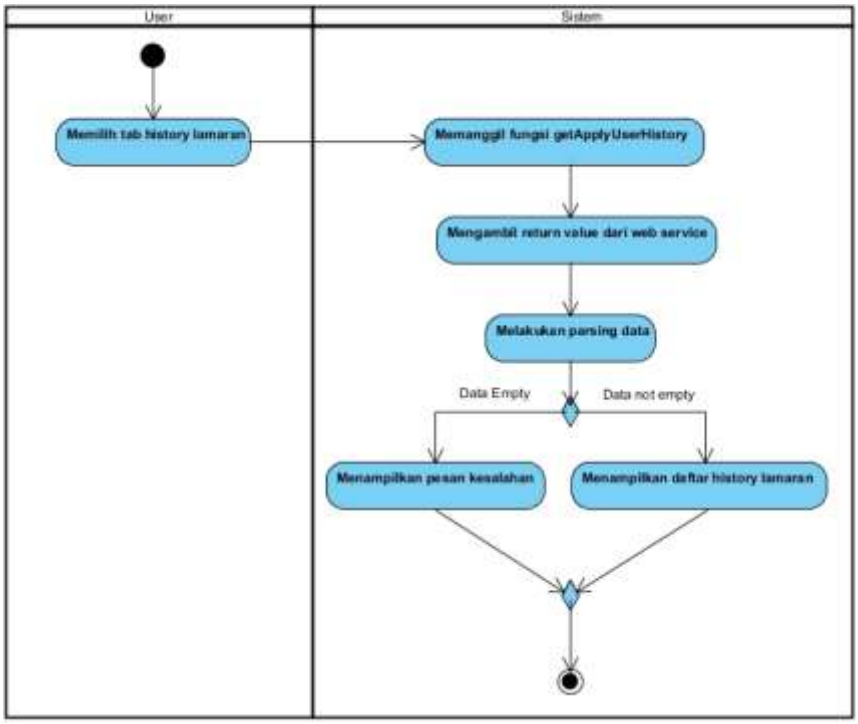
q. Activity diagram Lihat History Lamaran

Aktivitas Lihat *History* Lamaran merupakan aktivitas yang dilakukan oleh member pencari kerja untuk melihat *history* lamaran yang pernah dilakukan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih tab *history* lamaran, pada aktivitas ini *user* memilih tab *history* lamaran yang terdapat pada menu member.

2. Memanggil fungsi *getApply*, pada aktivitas ini sistem akan memanggil fungsi *getApply*.
3. Mengambil *return value* dari web service, pada aktivitas ini sistem mengambil nilai yang dikembalikan oleh *web service* berupa hasil dari eksekusi method pada *web service*.
4. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
5. Menampilkan pesan kesalahan, pada aktivitas ini sistem menampilkan pesan kesalahan jika data pelamar kosong.
6. Menampilkan daftar *history* lamaran, pada aktivitas ini jika data ditemukan maka sistem akan menampilkan daftar *history* lamaran dalam bentuk *list view* di android.

Seluruh aktivitas Lihat *History* Lamaran terlihat pada *activity diagram* gambar III.18.



Gambar III. 18 Activity Diagram Lihat History Lamaran

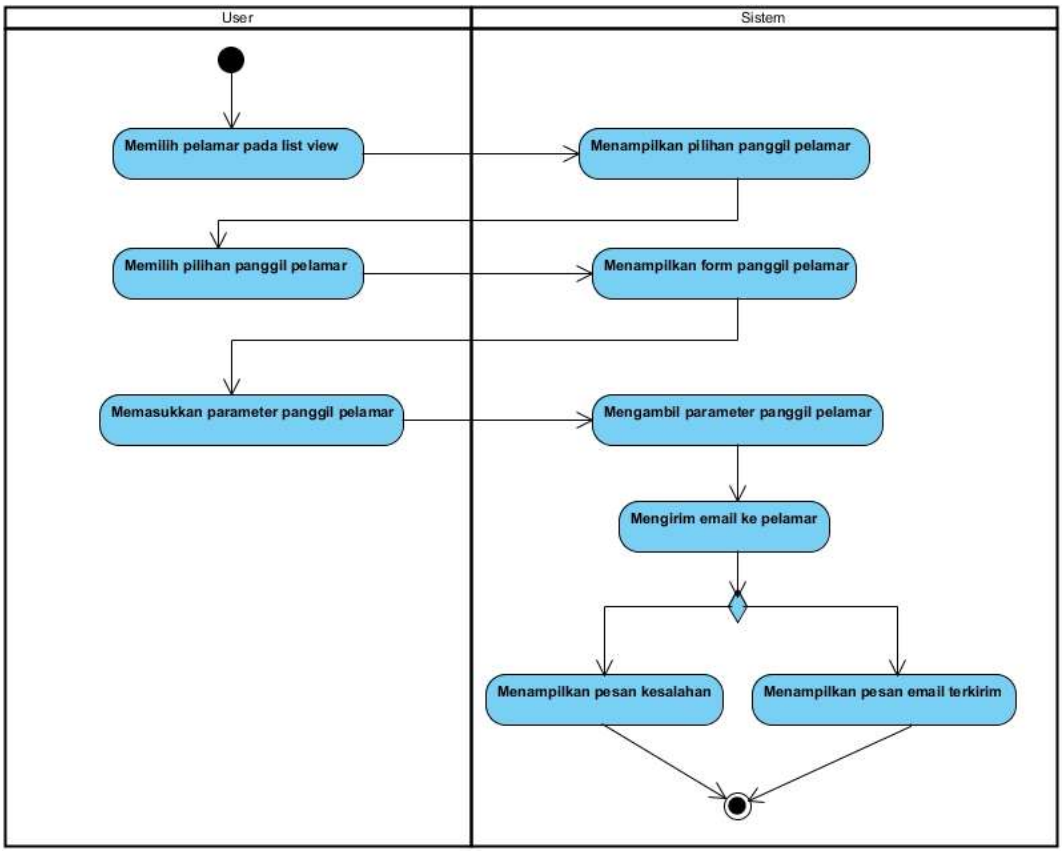
r. Activity diagram Panggil Pelamar

Aktivitas Lihat Panggil Pelamar merupakan aktivitas yang dilakukan oleh member pemberi kerja untuk memanggil pelamar dengan mengirimkan panggilan yang dikirim melalui email ke alamat email pelamar. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih lowongan pada *list view*, pada aktivitas ini *user* memilih lowongan yang terdapat pada *list view*.
2. Menampilkan pilihan panggil pelamar, pada aktivitas ini sistem akan menampilkan pilihan untuk panggil pelamar dari lowongan pekerjaan yang dipilih.
3. Memilih pilihan panggil pelamar, pada aktivitas ini user pemberi kerja memilih untuk memanggil pelamar.

4. Menampilkan form panggil pelamar, pada aktivitas sistem akan menampilkan form berupa alamat email tujuan,*subject* dan *body* dari surat panggilan yang akan dikirimkan melalui email.
5. Memasukkan parameter panggil pelamar, pada aktivitas *user* pemberi kerja memasukkan parameter panggil pelamar pada form panggil pelamar.
6. Mengambil parameter panggil pelamar, pada aktivitas ini sistem akan mengambil nilai parameter berupa alamat tujuan email, dan *subject* dan *body* yang telah dimasukkan oleh *user* pemberi kerja.
7. Mengirim email ke pelamar, pada aktivitas ini sistem akan mengirimkan email ke alamat email pelamar.
8. Menampilkan pesan terkirim, pada aktivitas ini sistem akan menampilkan pesan bahwa email terkirim jika email berhasil dikirim.
9. Menampilkan pesan kesalahan, aktivitas ini akan dilakukan jika email gagal dikirim.

Seluruh aktivitas Panggil Pelamar terlihat pada *activity diagram* gambar III.19.



Gambar III. 19 Activity Diagram Panggil Pelamar

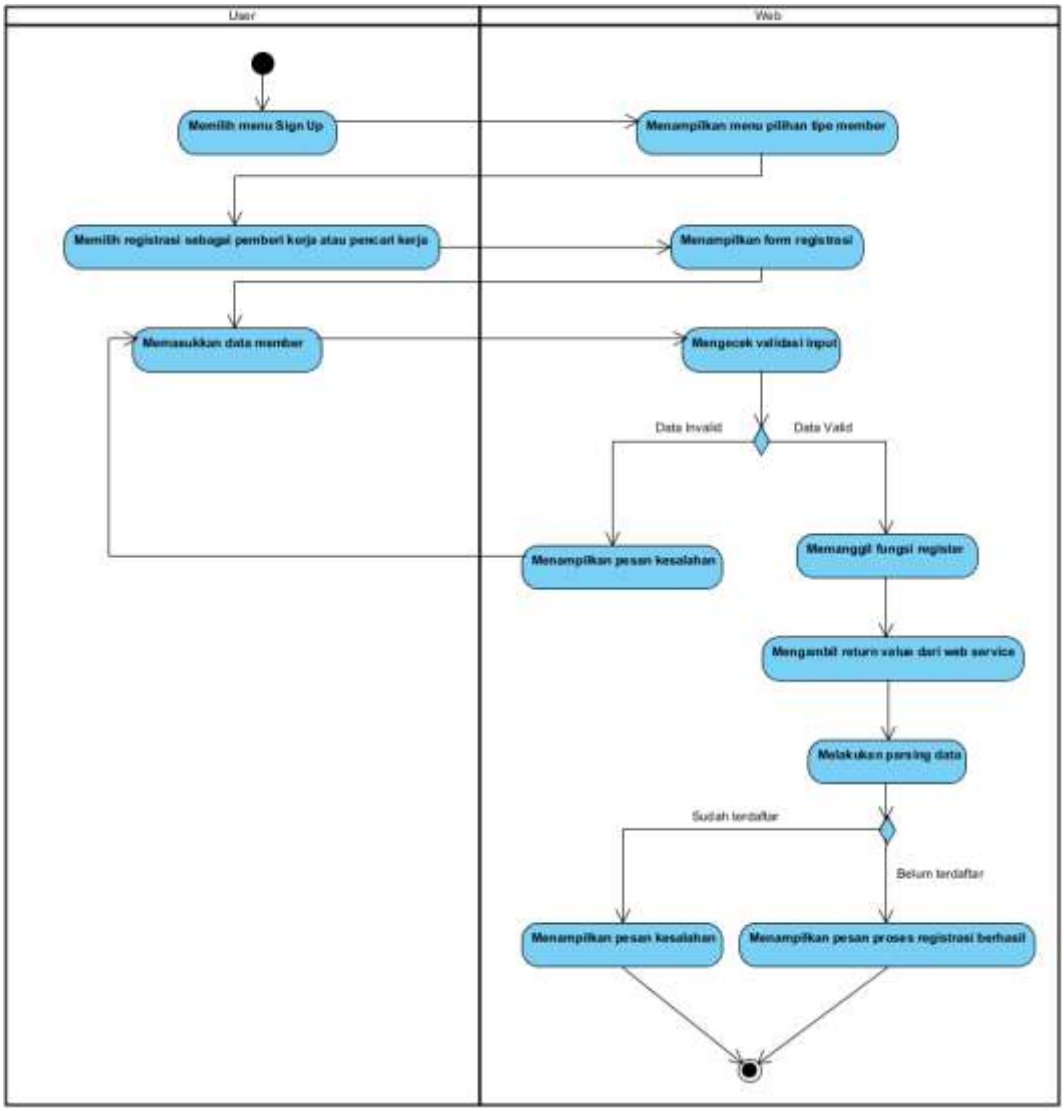
s. Activity diagram Registrasi

Aktivitas Lihat Registrasi merupakan aktivitas yang dilakukan oleh pengunjung umum melakukan registrasi menjadi member pemberi kerja atau pencari kerja aplikasi mobile android lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu *Sign Up*, pada aktivitas ini *user* memilih menu *Sign Up* yang terdapat pada menu *option* pada halaman utama.
2. Menampilkan menu pilihan tipe member, pada aktivitas ini sistem akan menampilkan tipe member untuk melakukan registrasi.

3. Memilih registrasi sebagai pemberi kerja atau pencari kerja, pada aktivitas ini *user* memilih untuk melakukan registrasi sebagai member pemberi kerja atau pencari kerja.
4. Menampilkan form registrasi, pada aktivitas ini sistem akan menampilkan form registrasi sessuai dengan pilihan *user* sebelumnya.
5. Memasukkan data member, pada aktivitas ini *user* akan memasukkan data yang dibutuhkan pada form registrasi.
6. Mengecek validasi input, pada aktivitas ini sistem akan memeriksa kolom pada form registrasi apakah sudah terisi semua atau masih kosong.
7. Menampilkan pesan kesalahan, aktivitas ini akan ditampilkan oleh sistem jika kolom pada form registrasi masih ada yang kosong.
8. Memanggil fungsi *register*, pada aktivitas ini sistem akan memanggil fungsi registerPmk atau registerPck sesuai dengan pilihan registrasi dari *user* saat awal registrasi.
9. Mengambil *return value* dari web service, pada aktivitas ini sistem mengambil nilai yang dikembalikan oleh *web service* berupa hasil dari eksekusi method pada *web service.*
10. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
11. Menampilkan pesan kesalahan, aktivitas ini akan ditampilkan oleh sistem jika proses registrasi gagal karena member sudah terdaftar.
12. Menampilkan pesan proses registrasi berhasil, pada aktivitas ini sistem akan menampilkan pesan bahwa registrasi berhasil dilakukan.

Seluruh aktivitas Registrasi terlihat pada *activity diagram* gambar III.20.

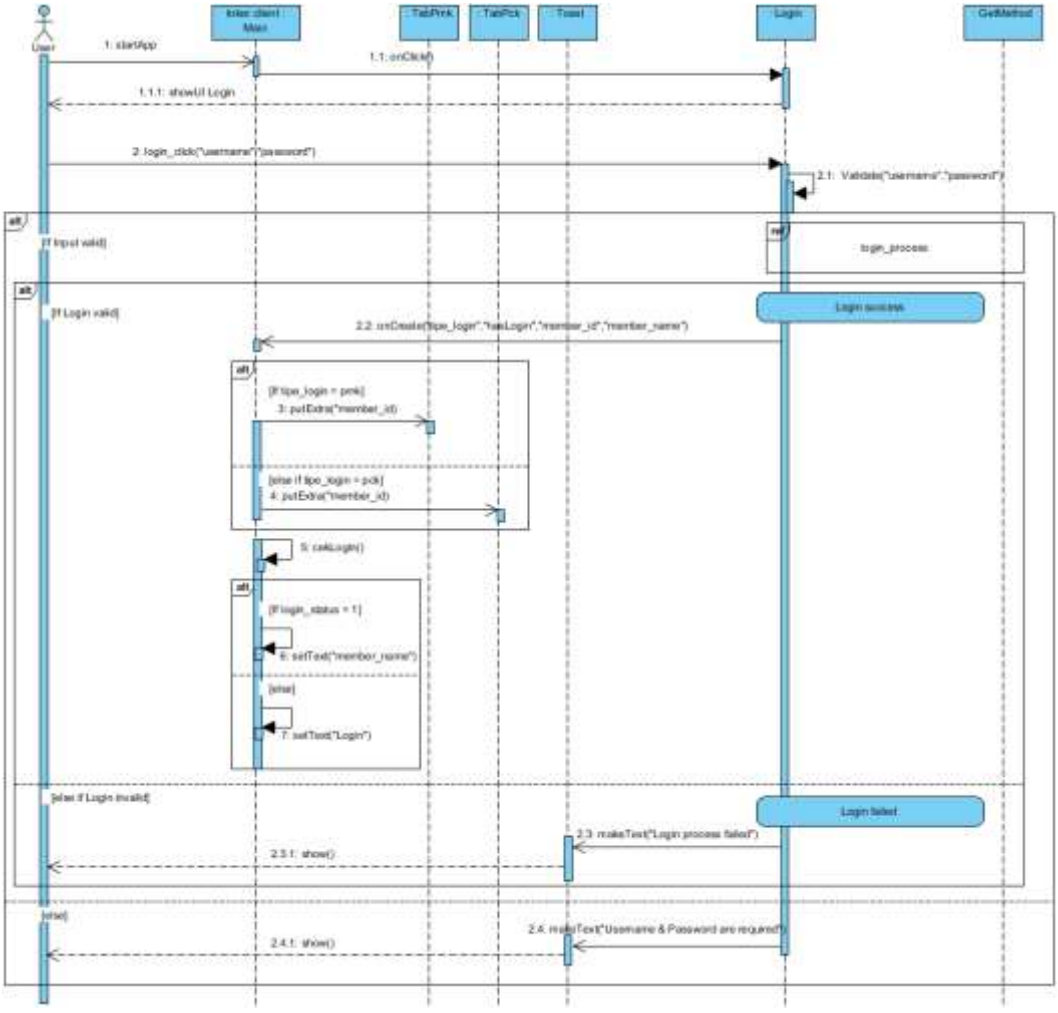


Gambar III. 20 Activity Diagram Register

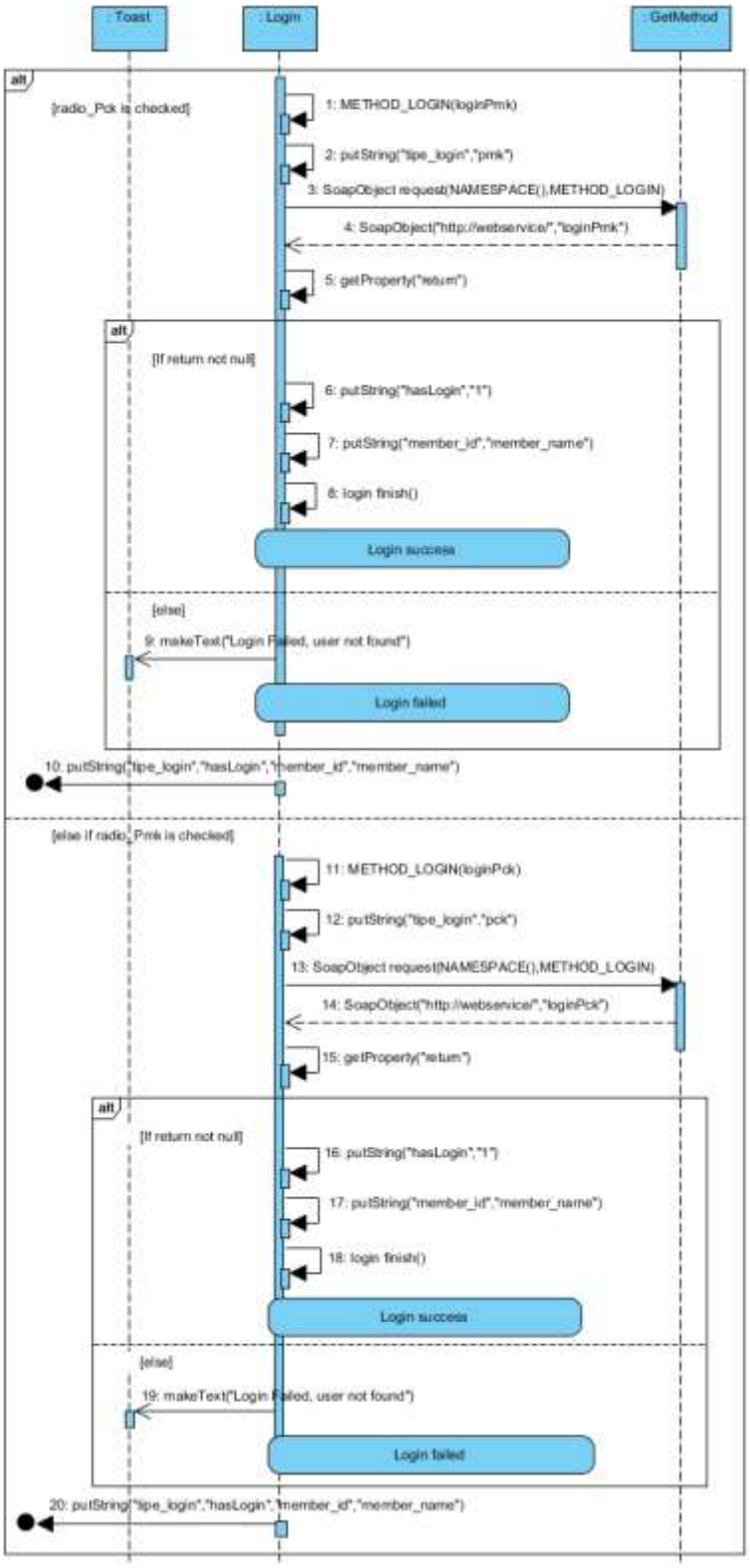
3. Sequence Diagram

Diagram sequence merupakan salah satu *Interaction Diagram* yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut. Berikut ini adalah sequence diagram dari sistem yang akan dibangun.

a. Sequence diagram Login

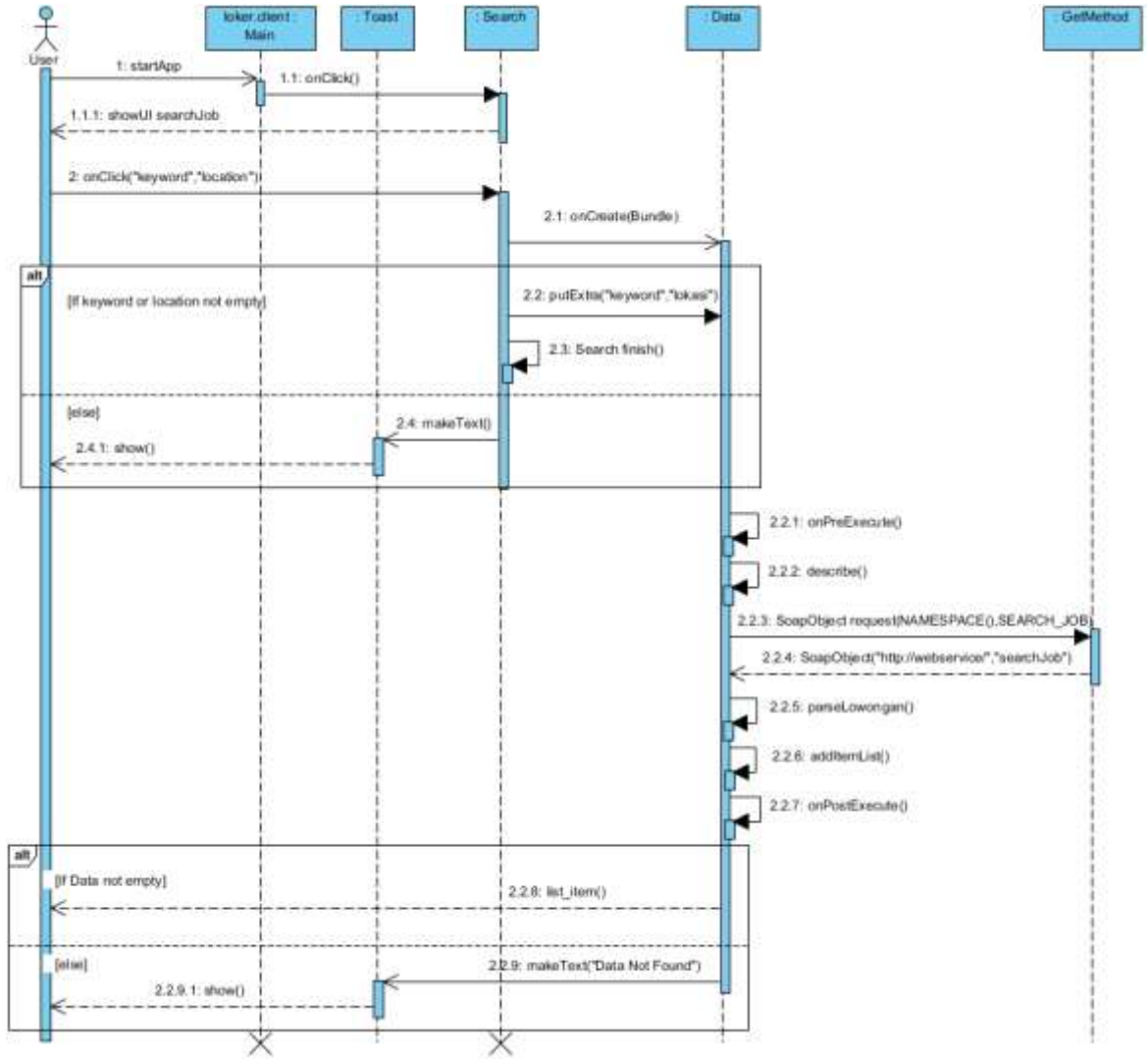


Gambar III. 21 Sequence Diagram Login



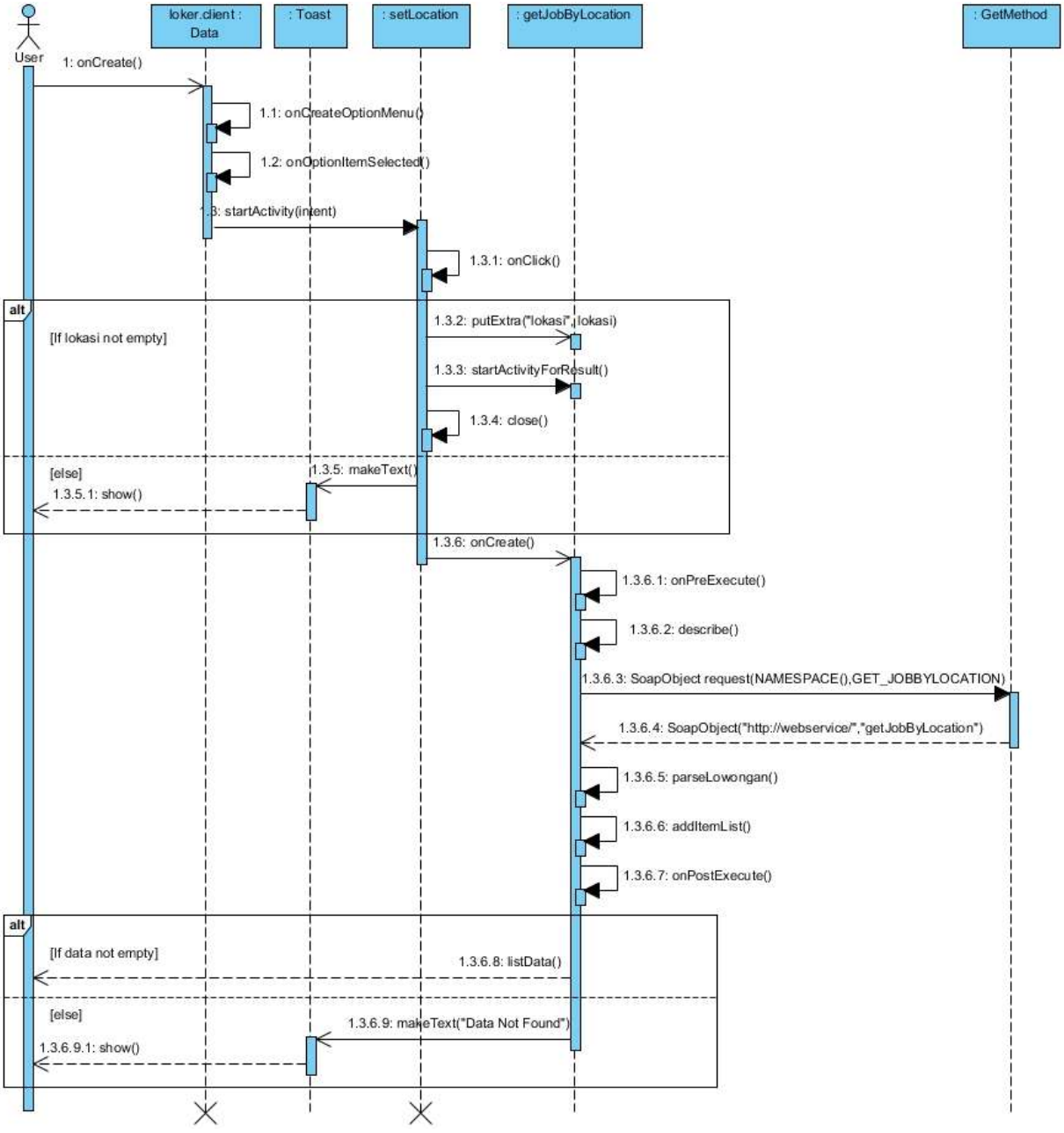
Gambar III. 22 Sequence Diagram Referenesi Interaksi login_process

b. Sequence diagram Pencarian Lowongan



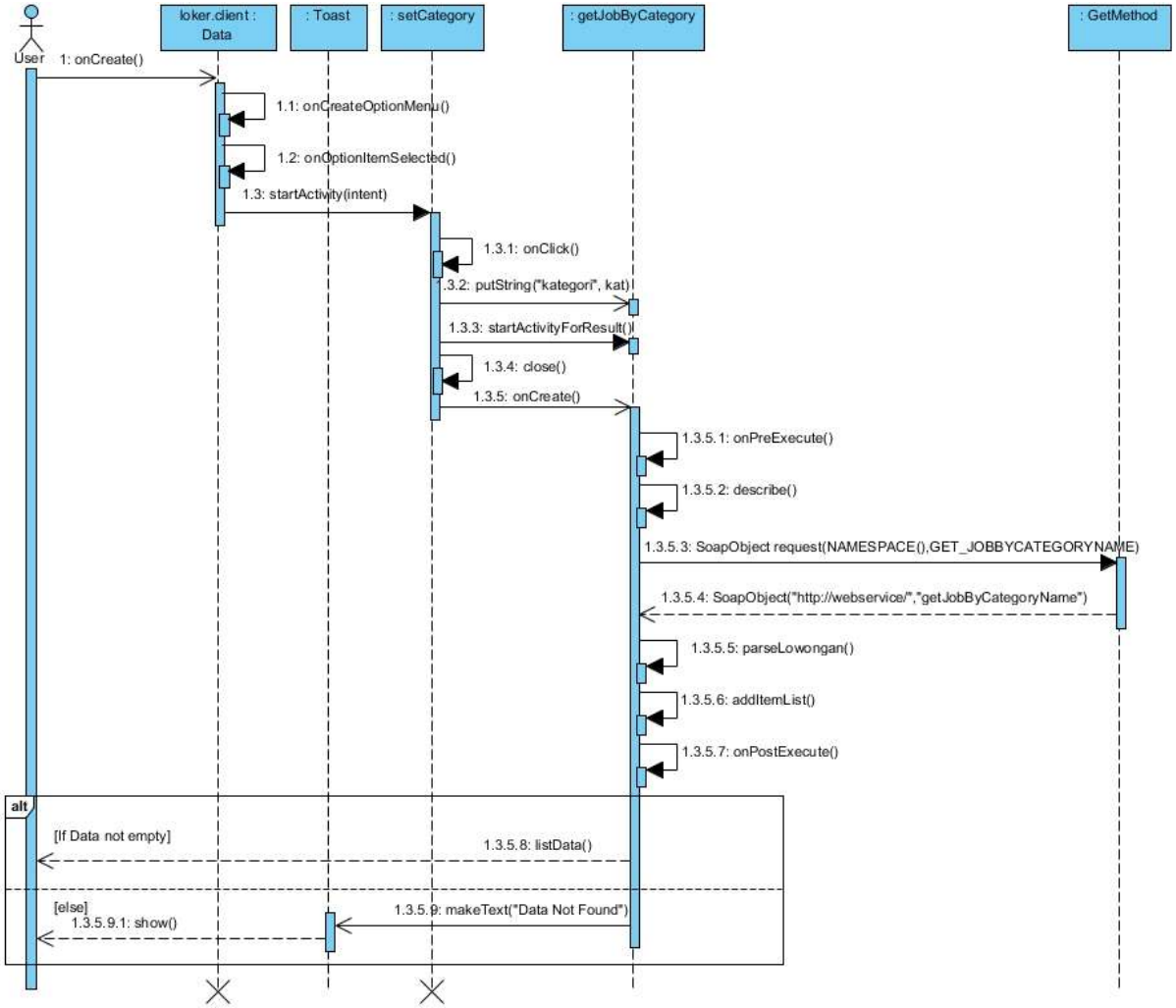
Gambar III. 23 Sequence Diagram Pencarian Lowongan.

c. Sequence diagram Filter Lowongan Berdasarkan Lokasi



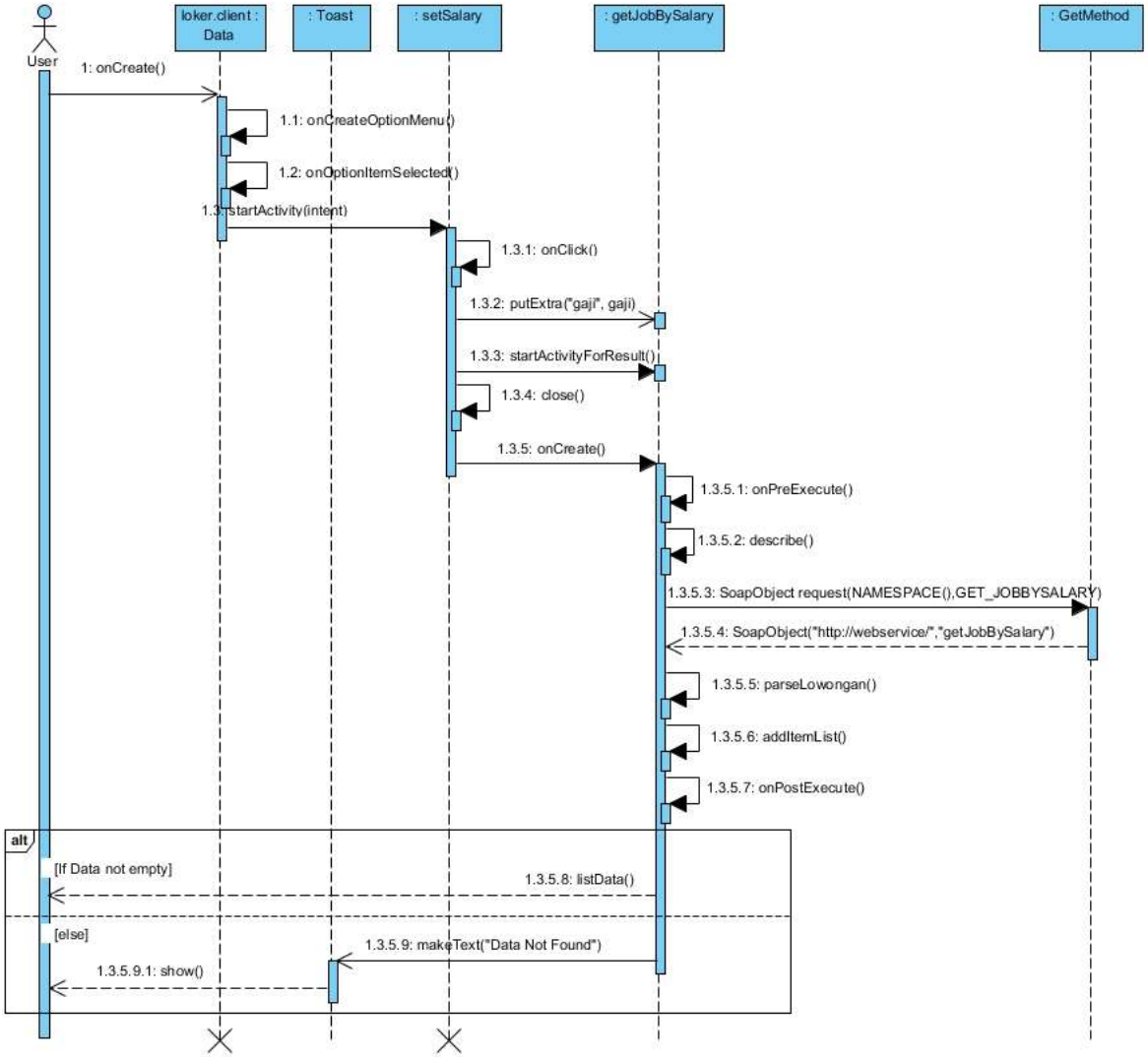
Gambar III. 24 Sequence Diagram Filter Lowongan Berdasarkan Lokasi.

d. Sequence diagram Filter Lowongan Berdasarkan Kategori



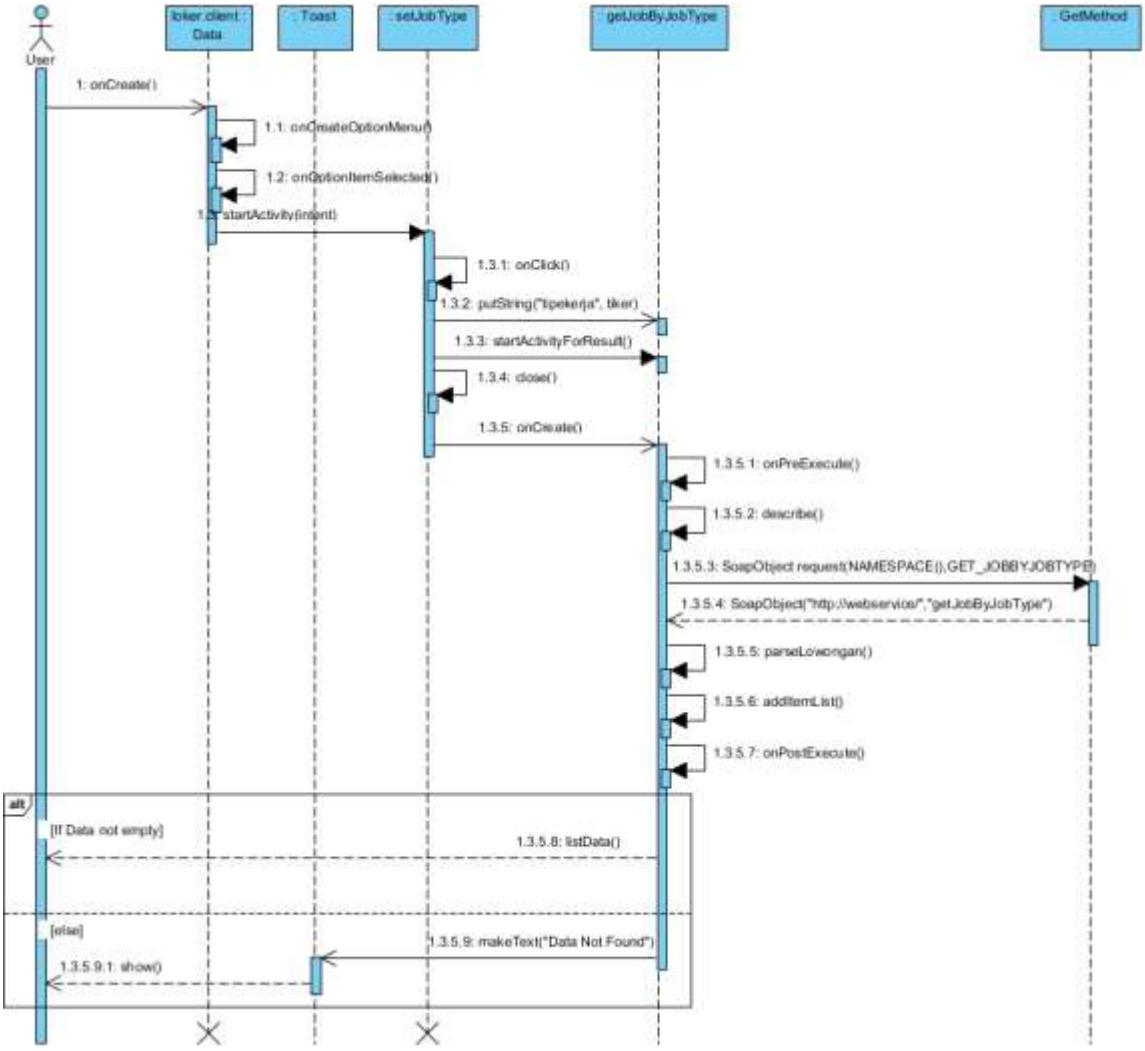
Gambar III. 25 Sequence Diagram Filter Lowongan Berdasarkan Kategori.

e. Sequence diagram Filter Lowongan Berdasarkan Gaji



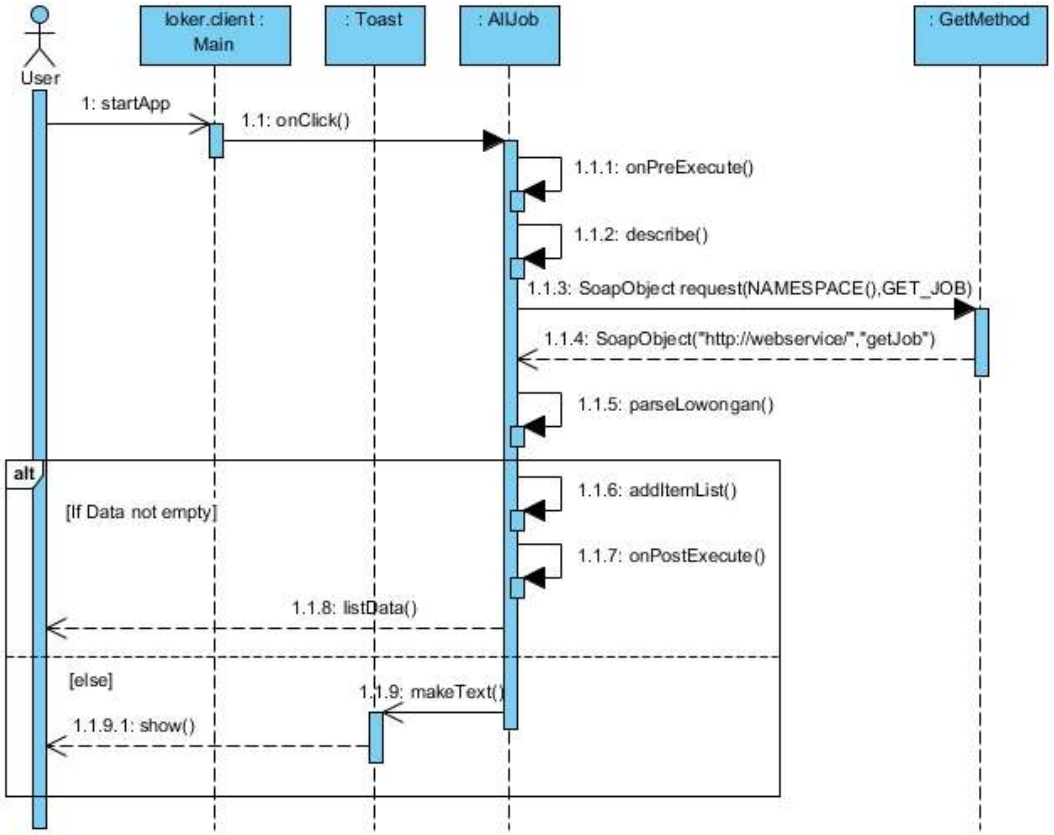
Gambar III. 26 Sequence Diagram Filter Lowongan Berdasarkan Gaji

f. Sequence diagram Filter Lowongan Berdasarkan Tipe Kerja



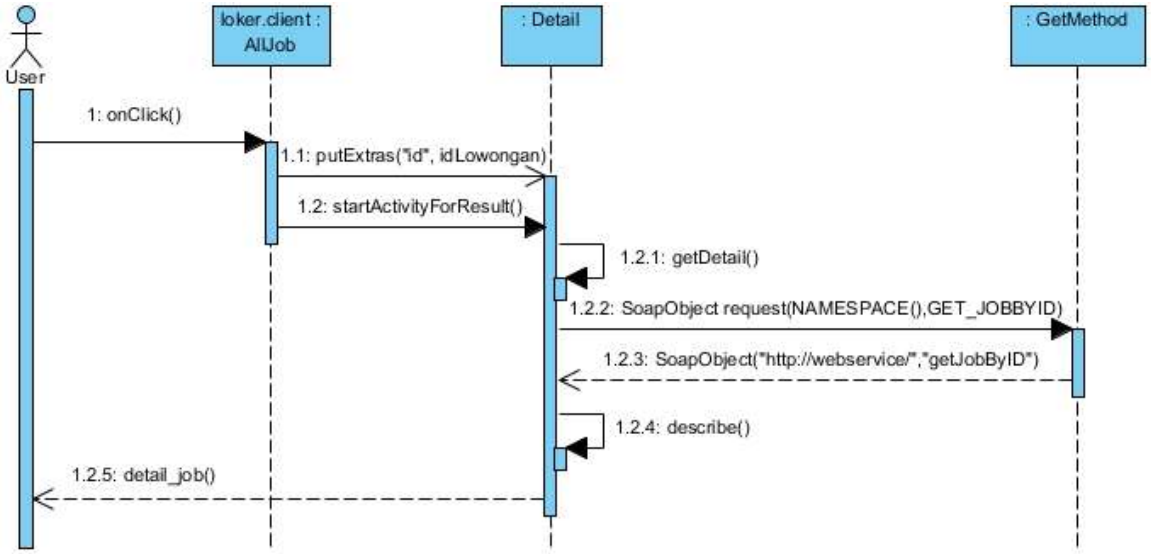
Gambar III. 27 Sequence Diagram Filter Lowongan Berdasarkan Tipe Kerja

g. Sequence diagram Lihat Daftar Lowongan



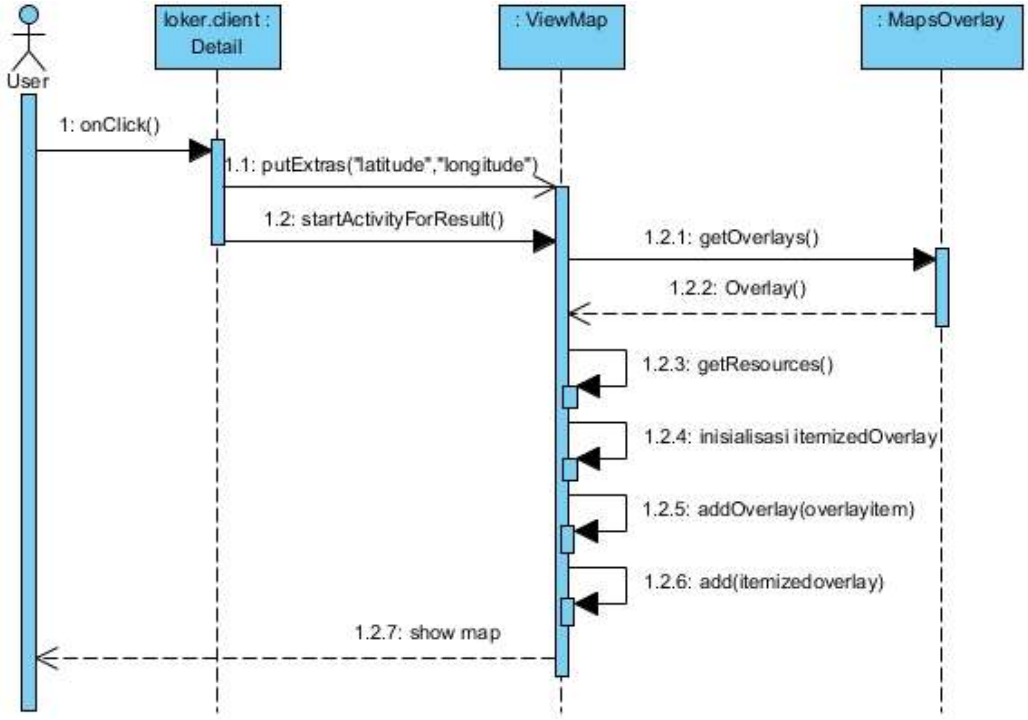
Gambar III. 28 Sequence Diagram Lihat Daftar Lowongan.

h. Sequence diagram Lihat Detil Lowongan



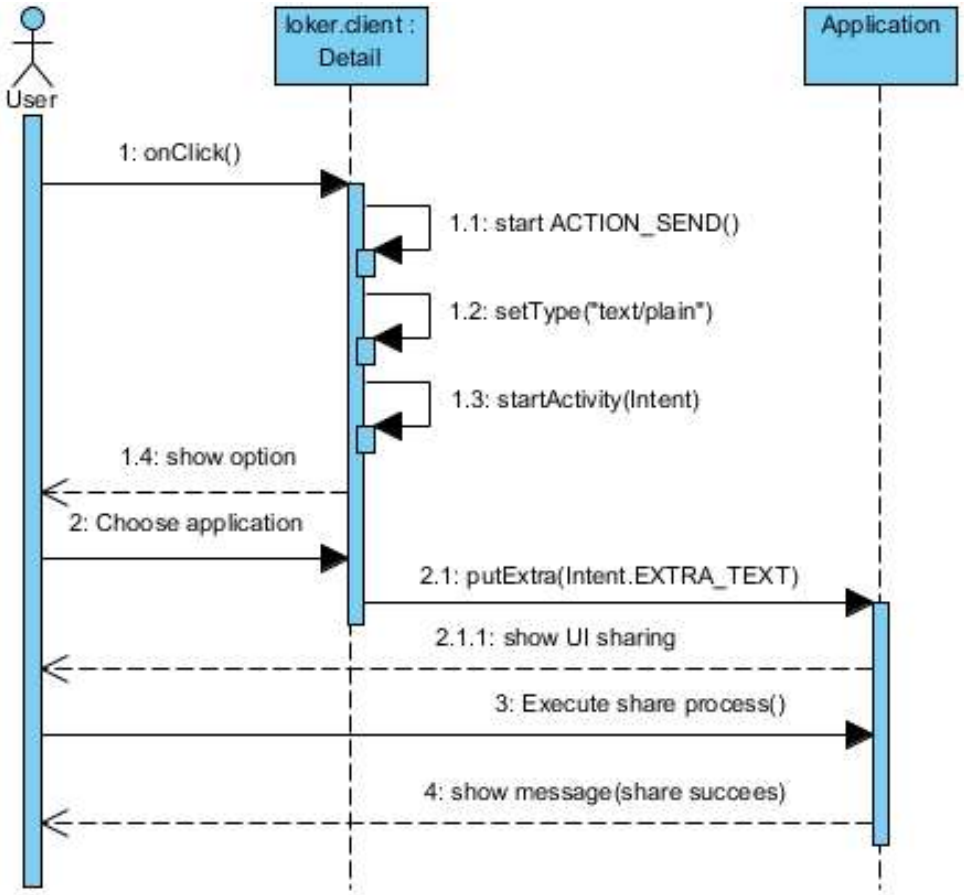
Gambar III. 29 Sequence Diagram Lihat Detil Lowongan.

i. Sequence diagram Lihat Peta



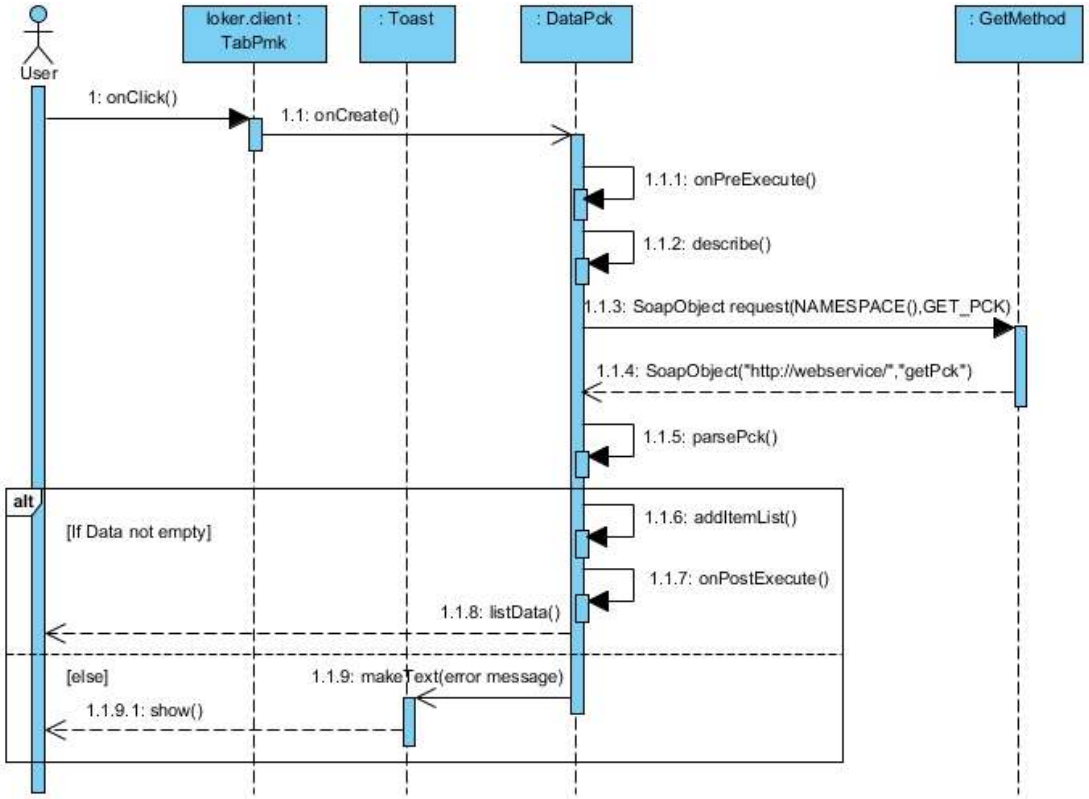
Gambar III. 30 Sequence Diagram Lihat Peta

j. Sequence diagram Membagikan Info Lowongan



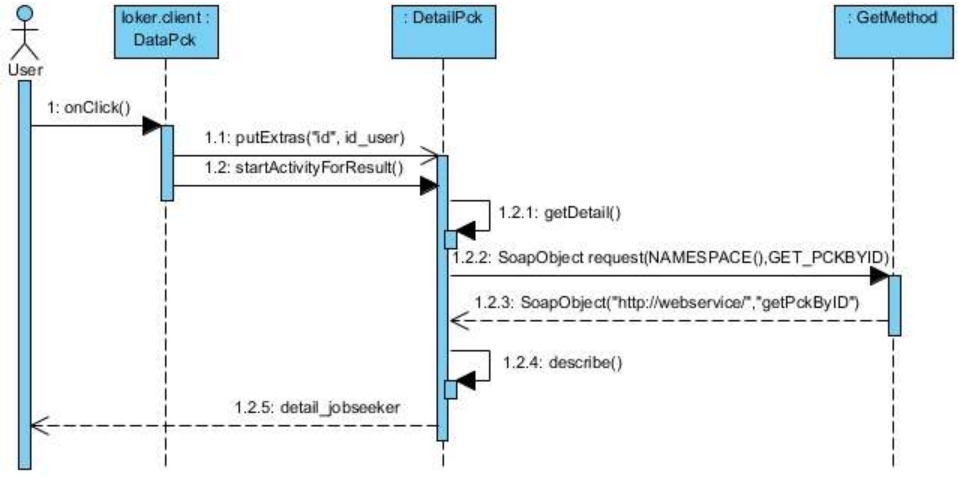
Gambar III. 31 Sequence Diagram Membagikan Info Lowongan

k. Sequence diagram Lihat Daftar Pencari Kerja



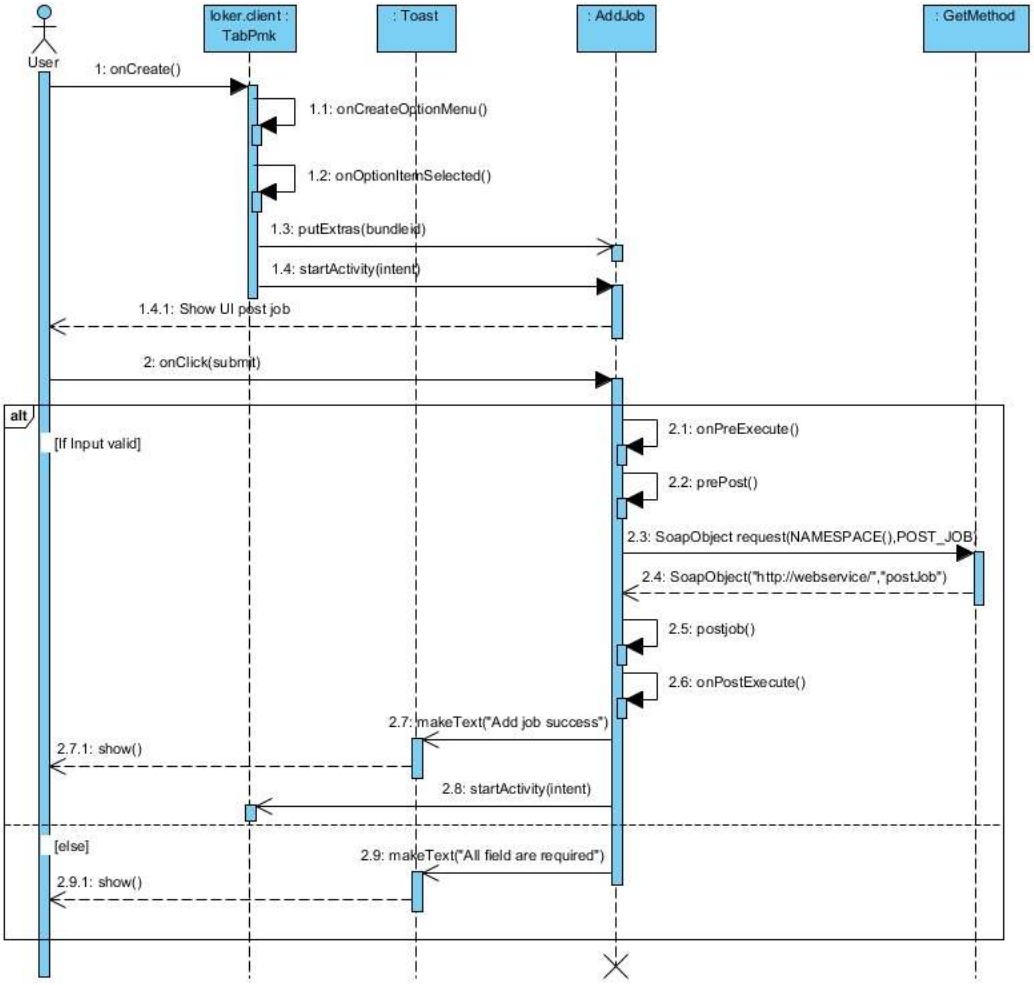
Gambar III. 32 Sequence Diagram Lihat Daftar Pencari Kerja

l. Sequence diagram Lihat Detil Pencari Kerja



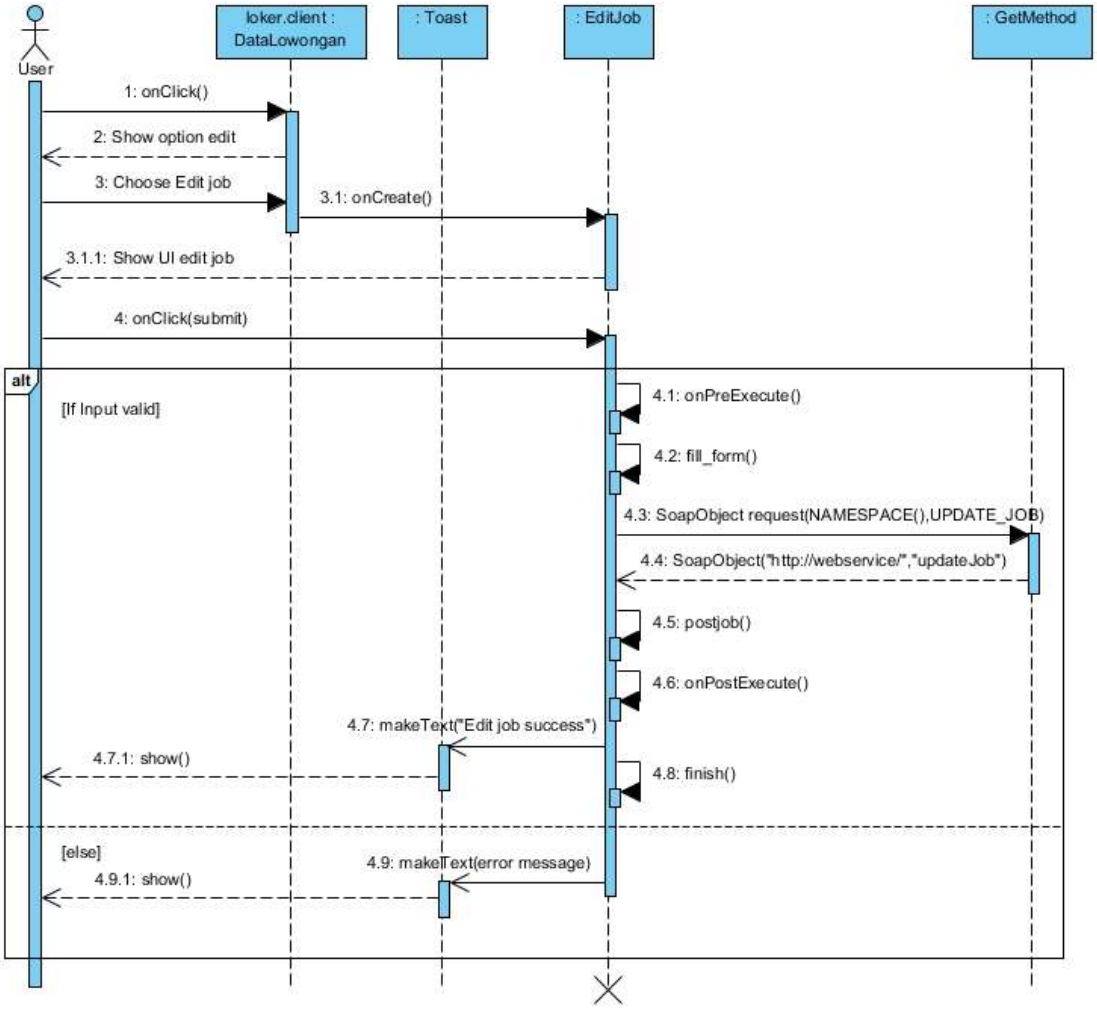
Gambar III. 33 Sequence Diagram Lihat Detil Pencari Kerja

m. Sequence diagram Pembuatan Lowongan Pekerjaan Baru



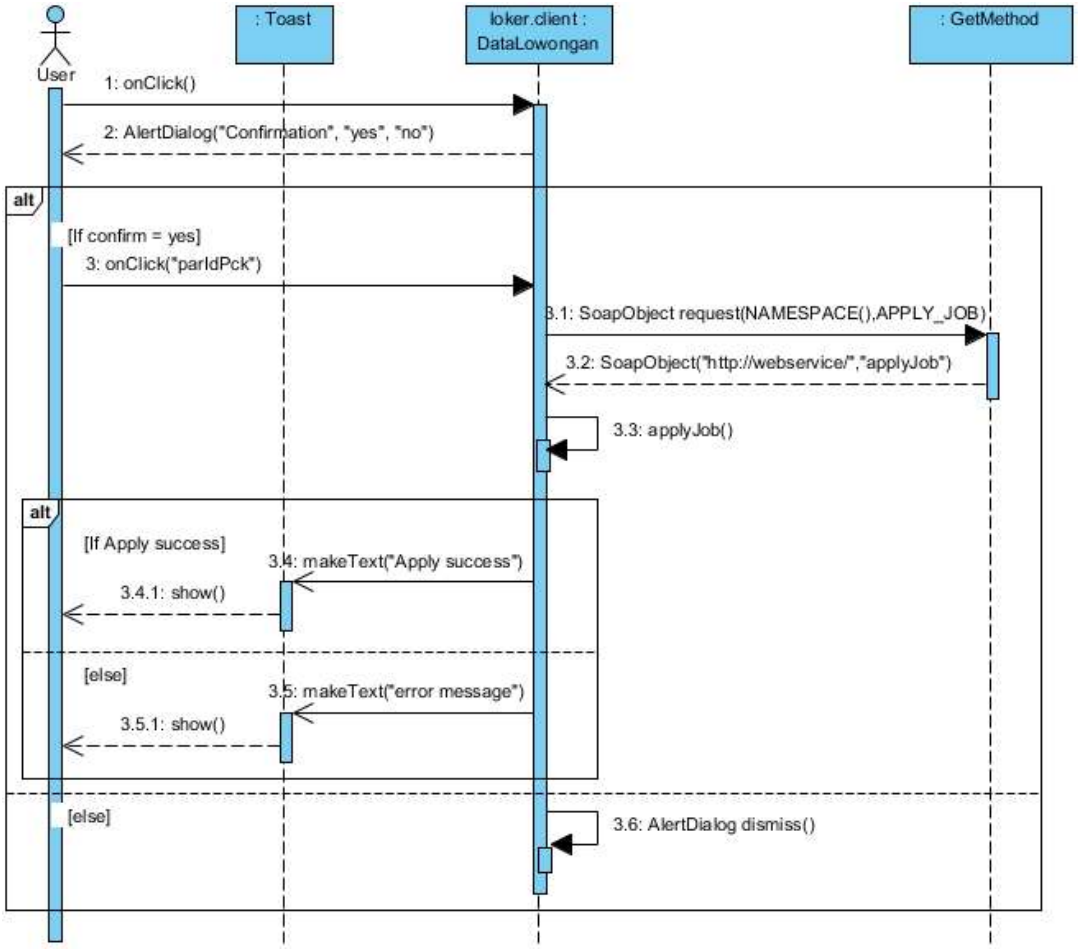
Gambar III. 34 Sequence Diagram Pembuatan Lowongan Pekerjaan Baru

n. Sequence diagram Perubahan Data Lowongan



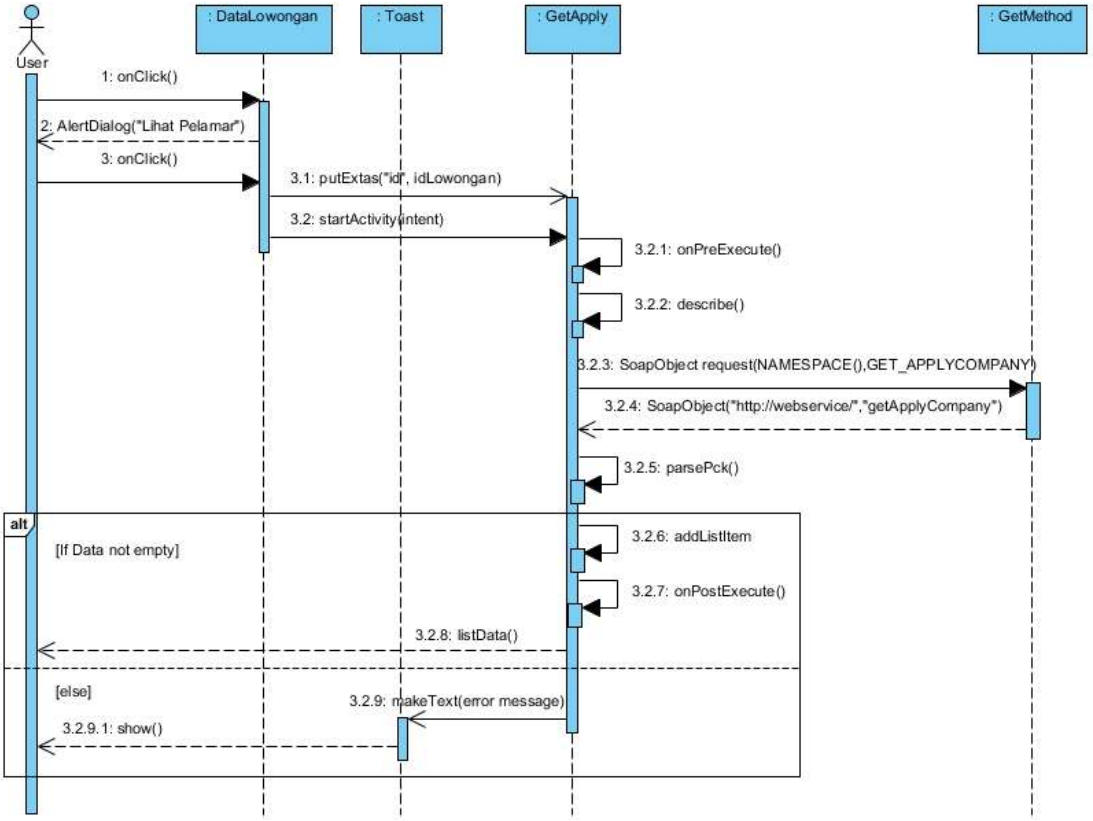
Gambar III. 35 Sequence Diagram Perubahan Data Lowongan

o. Sequence diagram Lamaran Lowongan Pekerjaan



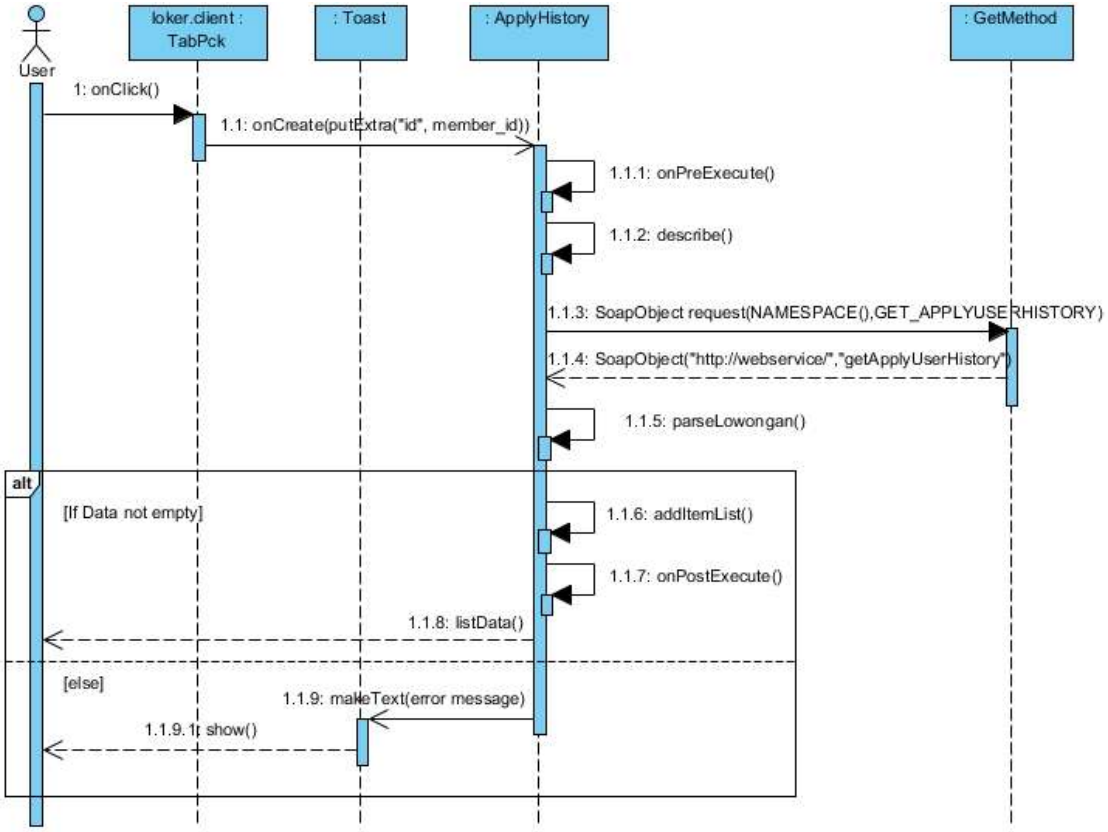
Gambar III. 36 Sequence Diagram Lamaran Lowongan Pekerjaan

p. Sequence diagram Lihat Pelamar



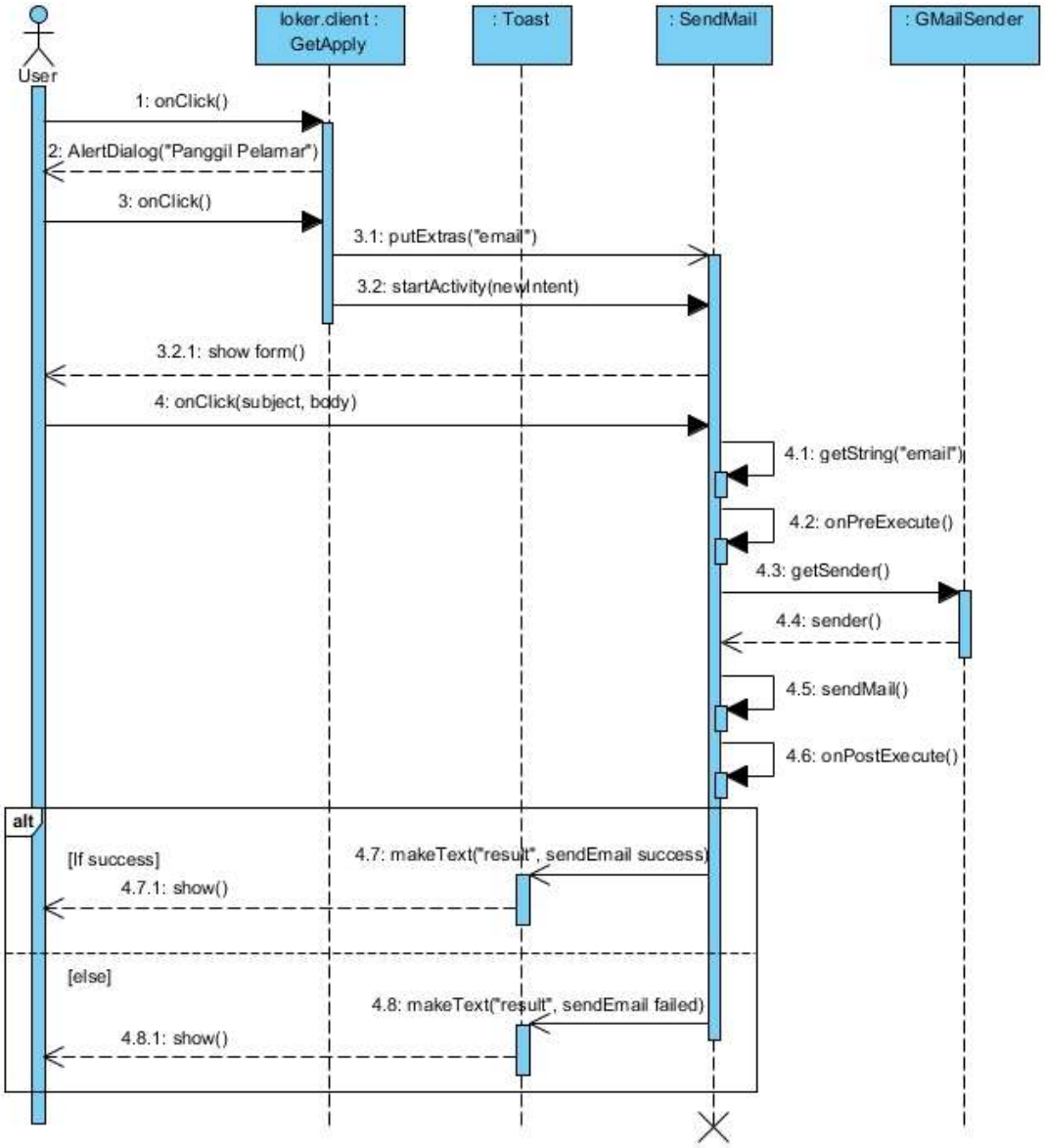
Gambar III. 37 Sequence Diagram Lihat Pelamar

q. Sequence diagram Lihat History Lamaran



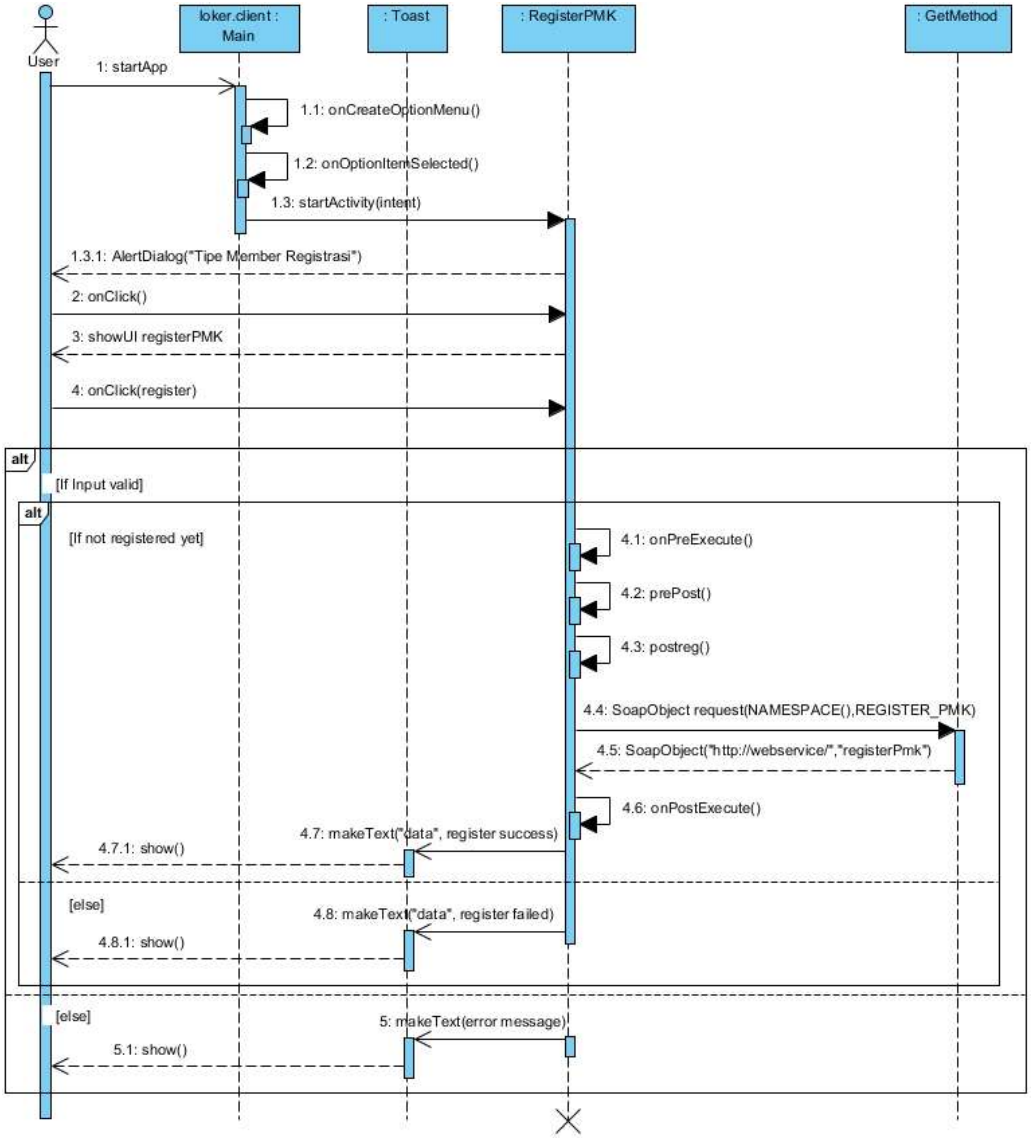
Gambar III. 38 Sequence Diagram Lihat History Lamaran

r. Sequence diagram Panggil Pelamar



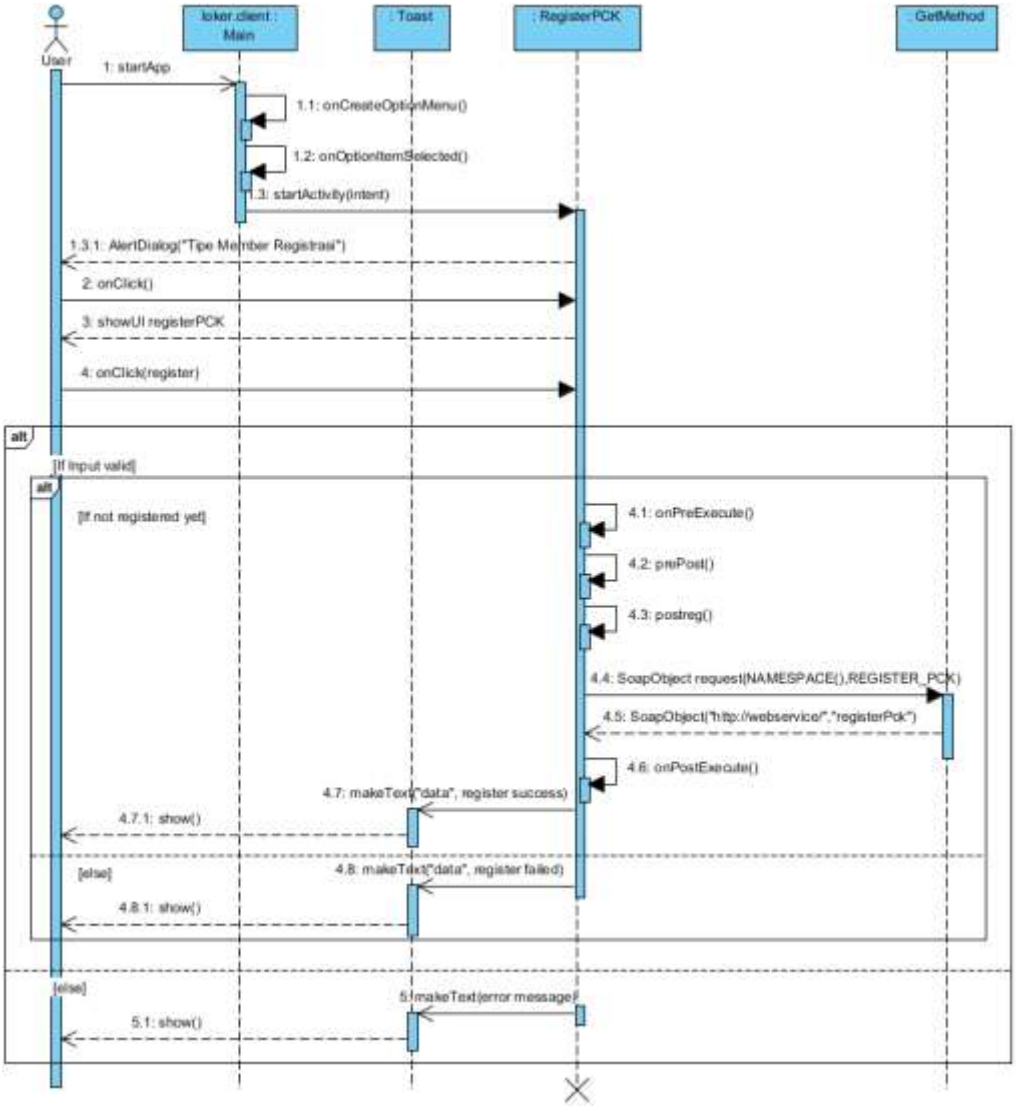
Gambar III. 39 Sequence Diagram Panggil Pelamar

s. Sequence diagram Register Pemberi Kerja



Gambar III. 40 Sequence Diagram Register Pemberi Kerja

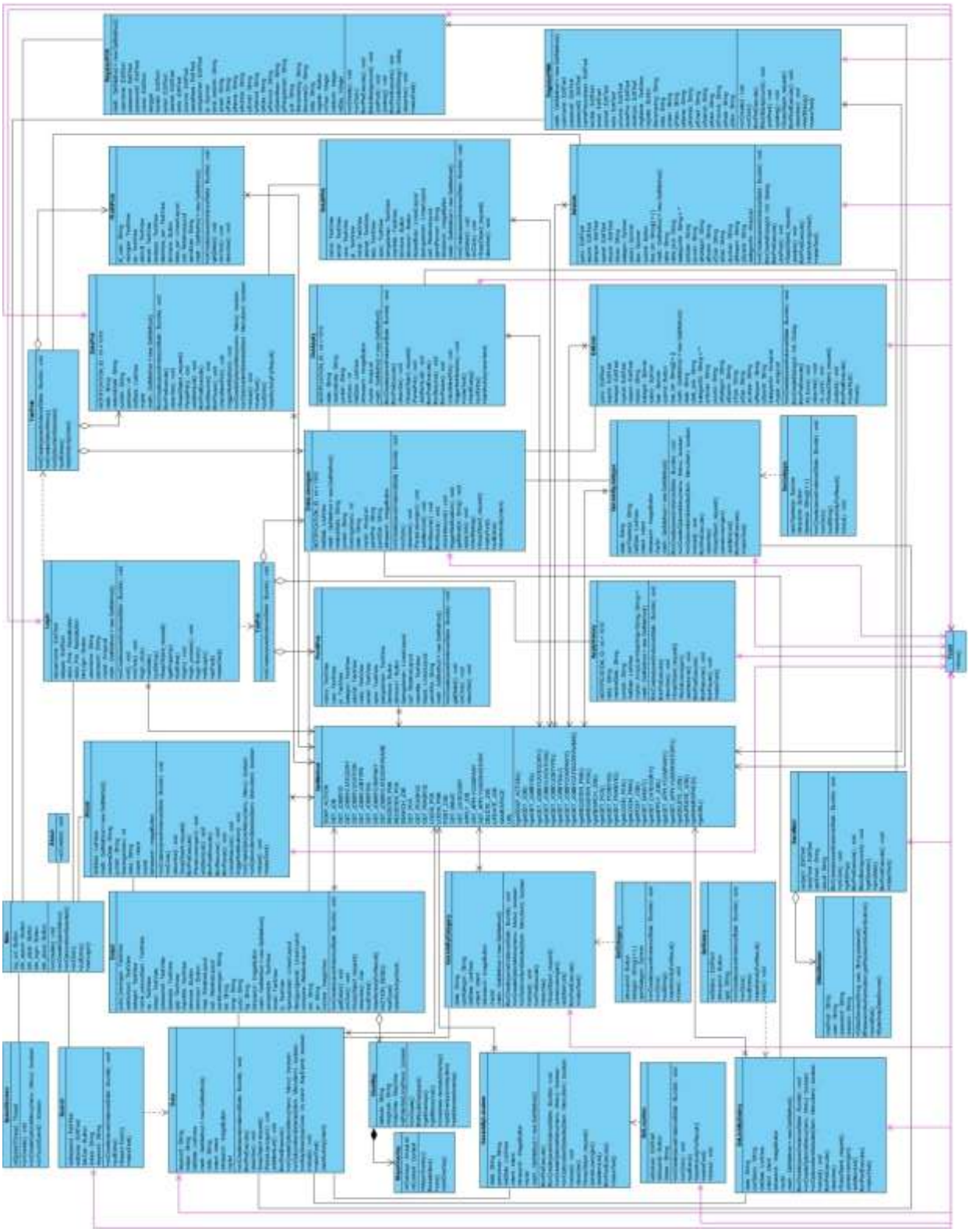
t. Sequence diagram Register Pencari Kerja



Gambar III. 41 Sequence Diagram Register Pencari Kerja

4. Class Diagram

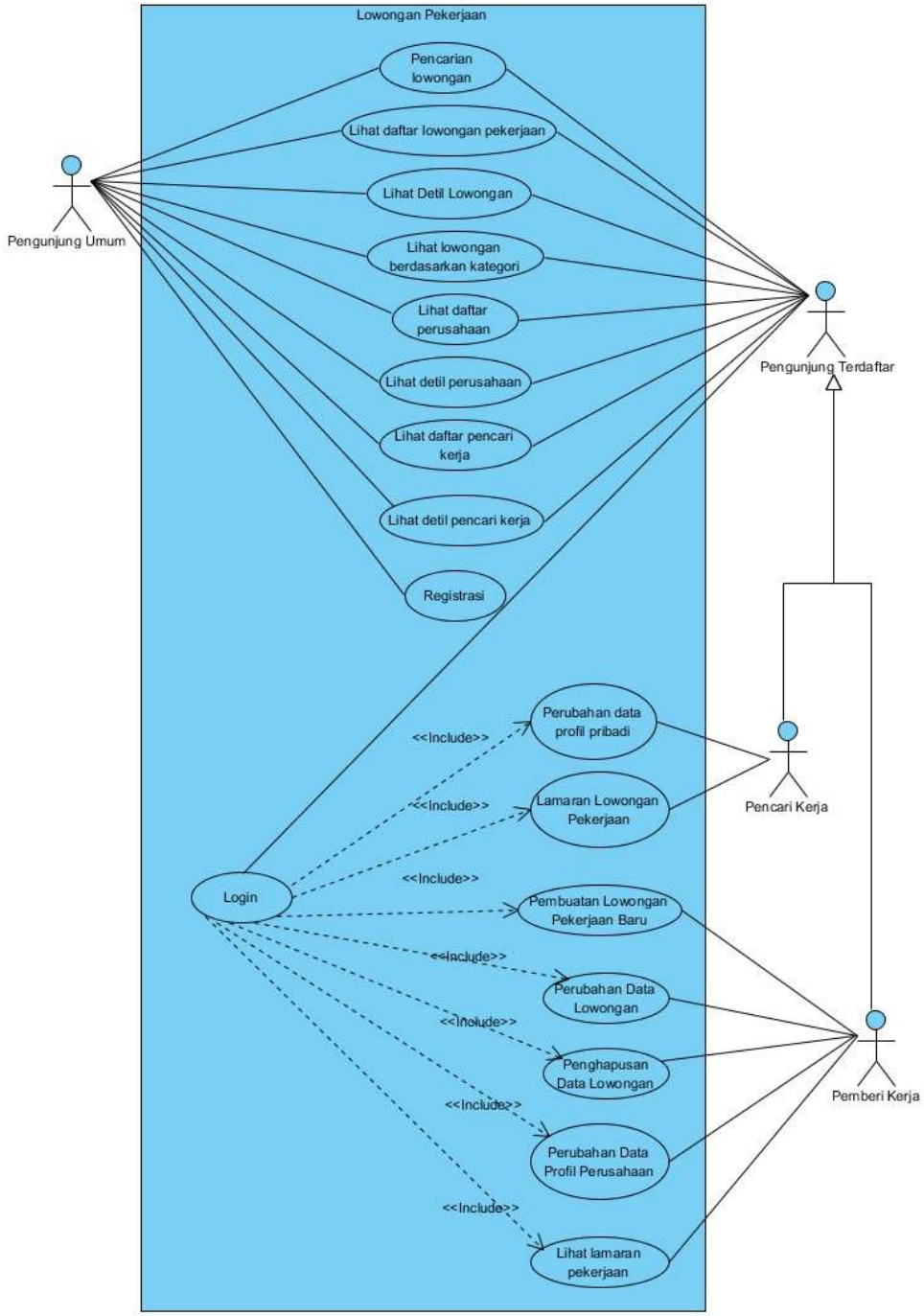
Class diagram menggambarkan struktur dan deskripsi class beserta hubungan satu sama lain seperti Asosiasi, Generalisasi dan lain sebagainya.



Gambar III. 42 Class Diagram Aplikasi Android

3.1.5.2 Analisis Pada Aplikasi Web

1. **Use Case Diagram**



Gambar III. 43 Use Case diagram sistem pada aplikasi web

a. Identifikasi Aktor

Pada tabel III.75 berisi daftar aktor dan deskripsi peran meraka masing-masing dalam aplikasi ini.

Tabel III. 75 Definisi Aktor pada aplikasi web

NO	Aktor	Deskripsi
B-01	Pengunjung Umum	Merupakan pengguna dari aplikasi web lowongan pekerjaan yang tidak terdaftar
B-02	Pemberi Kerja	Merupakan pengguna dari aplikasi web lowongan pekerjaan sebagai member pemberi kerja yang dapat menambahkan dan mengelola lowongan pekerjaan, dan menerima lamaran dari member pencari kerja.
B-03	Pencari Kerja	Merupakan pengguna dari aplikasi web lowongan pekerjaan sebagai member pencari kerja yang dapat melakukan *apply* atau lamaran kepada perusahaan yang memiliki lowongan pekerjaan di web lowongan pekerjaan.

b. Identifikasi Use Case

Berdasarkan use case diagram di atas maka di buatlah skenario dari tiap proses yang ada pada use case diagram tersebut. Pada tabel III.76

menggambarkan skenario mencari data lowongan pekerjaan pada use case Pencarian lowongan.

Tabel III. 76 Skenario Use Case Pencarian lowongan

Identifikasi	
Nomor	UC-W-01
Nama Use Case	Pencarian lowongan
Deskripsi	Proses untuk mencari data lowongan pekerjaan yang akan di tampilkan urut berdasarkan tanggal posting lowongan pekerjaan terbaru
Aktor	Pengunjung Umum, Pengunjung Terdaftar
Skenario Utama(UC-W-01-S01)	
Kondisi awal	Membuka alamat situs lowongan pekerjaan dan tampil menu utama
Aksi aktor	**Respon Sistem**
1. Memilih menu *button* Pencarian	2. Menampilkan form pencarian
3. Menginputkan parameter pencarian	
4. Menekan tombol cari	5. Memanggil fungsi searchJob
	6. Mengambil *return value* dari *web service* dalam bentuk SOAP
	7. Memparsing data dari *return value*
	8. Menampilkan data hasil parsing
Skenario Alternatif (UC- W-01-S03 Pencarian lowongan namun data tidak ditemukan	
	8a. menampilkan pesan kesalahan, misal karena data yang dicari tidak ditemukan.
Kondisi akhir	Data berhasil ditampilkan

Pada tabel III.77 menggambarkan skenario untuk melihat list atau daftar lowongan pekerjaan di situs lowongan pekerjaan pada use case Lihat daftar lowongan pekerjaan.

Tabel III. 77 Skenario Use Case Lihat daftar lowongan pekerjaan

Identifikasi	
Nomor	UC-W-02
Nama Use Case	Lihat daftar lowongan pekerjaan
Deskripsi	Proses untuk menampilkan daftar lowongan pekerjaan di situs lowongan pekerjaan
Aktor	Pengunjung Umum, Pengunjung Terdaftar
Skenario Utama(UC-W-02-S01)	
Kondisi awal	Membuka alamat situs lowongan pekerjaan dan tampil menu utama
Aksi aktor	**Respon Sistem**
1. Memilih menu Daftar lowongan	2. Memanggil fungsi getJob
	3. Mengambil *return value* dari *web service* dalam bentuk SOAP
	4. Memparsing data dari *return value*
	5. Menampilkan data hasil parsing berupa daftar lowongan pekerjaan
Kondisi akhir	Data daftar lowongan pekerjaan berhasil ditampilkan

Pada tabel III.78 menggambarkan skenario mengambil data lowongan pekerjaan berdasarkan id pada use case Lihat detil lowongan.

Tabel III. 78 Skenario Use Case Lihat detil lowongan

Identifikasi		
Nomor	UC-W-03	
Nama Use Case	Lihat detil lowongan	
Deskripsi	Proses untuk melihat detil lowongan pekerjaan	
Aktor	Pengunjung Umum, Pengunjung Terdaftar	
Skenario Utama (UC-W-03-S01)		
Kondisi awal	Berada pada halaman daftar lowongan	
Aksi aktor		**Respon Sistem**
1. Memilih lowongan pekerjaan pada *list view*		2. Menggambil parameter id lowongan pekerjan
		3. Memanggil fungsi getJobByID
		4. Mengambil *return value* dari *web service* dalam bentuk SOAP
		5. Memparsing data dari *return value*
		6. Menampilkan detil lowongan pekerjaan
Kondisi akhir		Detil lowongan berhasil ditampilkan

Pada tabel III.79 menggambarkan skenario filter data lowongan pekerjaan berdasarkan kategori pekerjaan pada use case Lihat lowongan berdasarkan kategori.

Tabel III. 79 Skenario Use Case Lihat lowongan berdasarkan kategori

Identifikasi	
Nomor	UC-W-04
Nama Use Case	Lihat lowongan berdasarkan kategori
Deskripsi	Proses untuk melihat lowongan pekerjaan berdasarkan kategori tertentu
Aktor	Pengunjung Umum, Pengunjung Terdaftar
Skenario Utama(UC-W-04-S01)	
Kondisi awal	Berada pada halaman daftar lowongan
Aksi aktor	**Respon Sistem**
1. Memilih salah satu kategori lowongan pekerjaan	2. Mengambil parameter kategori yang dipilih oleh user
	3. Memanggil fungsi getJobByCategory
	4. Mengambil *return value* dari *web service* dalam bentuk SOAP
	5. Memparsing data dari *return value*
	6. Menampilkan data hasil parsing
Skenario Alternatif (UC-W-04-S02) Lihat lowongan berdasarkan kategori namun terdapat pesan error	
	6.a Menampilkan pesan kesalahan, misalnya karena data pada kategori yang dipilih tidak ada
Kondisi akhir	Data berhasil ditampilkan

Pada tabel III.80 menggambarkan skenario untuk melihat list atau daftar perusahaan yang terdaftar di situs lowongan pekerjaan pada use case Lihat daftar perusahaan.

Tabel III. 80 Skenario Use Case Lihat daftar perusahaan

Identifikasi	
Nomor	UC-W-05
Nama Use Case	Lihat daftar perusahaan
Deskripsi	Proses untuk menampilkan daftar perusahaan yang terdaftar di situs lowongan pekerjaan
Aktor	Pengunjung Umum, Pengunjung Terdaftar
Skenario Utama(UC-W-05-S01)	
Kondisi awal	Membuka alamat situs lowongan pekerjaan dan tampil menu utama
Aksi aktor	**Respon Sistem**
1. Memilih menu Daftar Perusahaan	2. Memanggil fungsi getPmk
	3. Mengambil *return value* dari *web service* dalam bentuk SOAP
	4. Memparsing data dari *return value*
	5. Menampilkan data hasil parsing
Skenario Alternatif (UC-W-05-S02) Lihat daftar perusahaan namun terdapat pesan error	
	5.a Menampilkan pesan kesalahan, misalnya karena data perusahaan masih kosong atau tidak ada
Kondisi akhir	Data daftar perusahaan berhasil ditampilkan

Pada tabel III.81 menggambarkan skenario mengambil data perusahaan berdasarkan id pada use case Lihat detil perusahaan.

Tabel III. 81 Skenario Use Case Lihat detil perusahaan

Identifikasi	
Nomor	UC-W-06
Nama Use Case	Lihat detil perusahaan
Deskripsi	Proses untuk melihat detil perusahaan sebagai pemberi kerja
Aktor	Pengunjung Umum, Pengunjung Terdaftar
Skenario Utama (UC-W-06-S01)	
Kondisi awal	Berada pada halaman daftar perusahaan
Aksi aktor	**Respon Sistem**
1. Memilih salah satu perusahaan pada *list view*	2. Menggambil parameter id perusahaan
	3. Memanggil fungsi getPmkByID
	4. Mengambil *return value* dari *web service* dalam bentuk SOAP
	5. Memparsing data dari *return value*
	6. Menampilkan detil perusahaan
Kondisi akhir	Detil perusahaan berhasil ditampilkan

Pada tabel III.82 menggambarkan skenario untuk melihat list atau daftar pencari kerja yang terdaftar di situs lowongan pekerjaan pada use case Lihat daftar perusahaan.

Tabel III. 82 Skenario Use Case Lihat daftar pencari kerja

Identifikasi	
Nomor	UC-W-07
Nama Use Case	Lihat daftar pencari kerja
Deskripsi	Proses untuk menampilkan daftar pencari kerja yang telah mendaftarkan diri di situs lowongan pekerjaan
Aktor	Pengunjung Umum, Pengunjung Terdaftar
Skenario Utama(UC-W-07-S01)	
Kondisi awal	Membuka alamat situs lowongan pekerjaan dan tampil menu utama
Aksi aktor	**Respon Sistem**
1. Memilih menu Daftar Pencari Kerja	2. Memanggil fungsi getPck
	3. Mengambil *return value* dari *web service* dalam bentuk SOAP
	4. Memparsing data dari *return value*
	5. Menampilkan data hasil parsing berupa daftar pencari kerja
Skenario Alternatif (UC-W-07-S02) Lihat daftar pencari kerja namun terdapat pesan error	
	5.a Menampilkan pesan kesalahan, misalnya karena data pencari kerja masih kosong atau tidak ada
Kondisi akhir	Data berhasil ditampilkan

Pada tabel III.83 menggambarkan skenario mengambil data pencari kerja berdasarkan id pada use case Lihat detil pencari kerja.

Tabel III. 83 Skenario Use Case Lihat detil pencari kerja

Identifikasi	
Nomor	UC-W-08
Nama Use Case	Lihat detil pencari kerja
Deskripsi	Proses untuk melihat detil pencari kerja
Aktor	Pengunjung Umum, Pengunjung Terdaftar
Skenario Utama (UC-W-08-S01)	
Kondisi awal	Berada pada halaman daftar pencari kerja
Aksi aktor	**Respon Sistem**
1. Memilih salah satu pencari kerja pada *list view*	2. Menggambil parameter id pencari kerja
	3. Memanggil fungsi getPckByID
	4. Mengambil *return value* dari *web service* dalam bentuk SOAP
	5. Memparsing data dari *return value*
	6. Menampilkan detil pencari kerja
Kondisi akhir	Detil pencari kerja berhasil ditampilkan

Pada tabel III.84 menggambarkan skenario pendaftaran pada situs lowongan pekerjaan pada use case Registrasi.

Tabel III. 84 Skenario Use Case Registrasi

Identifikasi	
Nomor	UC-W-09
Nama Use Case	Registrasi
Deskripsi	Proses untuk mendaftarkan diri sebagai pencari kerja atau sebagai pemberi kerja
Aktor	Pengunjung Umum
Skenario Utama(UC-W-09-S01)	
Kondisi awal	Membuka alamat situs lowongan pekerjaan dan tampil menu utama
Aksi aktor	**Respon Sistem**
1. Memilih menu *Login*	2. Menampilkan form login dan tombol *sign up*
3. Memilih tombol *sign up*	4. Menampilkan halaman pilihan tipe member registrasi
5. Memilih menu registrasi sebagai pemberi kerja atau pencari kerja	6. Menampilkan form registrasi
7. Mengisi form data member pemberi kerja atau data pribadi	8. Mengecek validasi data
	9. Memanggil fungsi register
	10. Menyimpan data dalam database
	11. Menampilkan pesan proses registrasi berhasil
Skenario Alternatif (UC-M-09-S02) Registrasi namun terdapat pesan error	
	8a. Menampilkan pesan kesalahan, misalnya inputan data pribadi atau pemberi kerja belum diisi atau belum lengkap.
	10a. Menampilkan pesan kesalahan, misalnya karena data member sudah terdaftar
Kondisi akhir	Proses registrasi berhasil

Pada tabel III.85 menggambarkan skenario untuk login di situs lowongan pekerjaan pada use case Login.

Tabel III. 85 Skenario Use Case Login

Identifikasi	
Nomor	UC-W-10
Nama Use Case	Login
Deskripsi	Proses untuk login sebagai member pemberi kerja atau pencari kerja di situs lowongan pekerjaan
Aktor	Pengunjung Terdaftar
Skenario Utama(UC-W-10-S01)	
Kondisi awal	Membuka alamat situs lowongan pekerjaan dan tampil halaman utama
Aksi aktor	**Respon Sistem**
1. Memilih menu Login	2. Menampilkan halaman login
3. Mengisi data pada form login	4. Mengecek validasi inputan data member
	5. Memanggil fungsi login
	6. Cek validasi data member
	7. Menampilkan menu utama member terdaftar
Skenario Alternatif (UC-W-10-S02) Login namun terdapat pesan error	
	4.a Menampilkan pesan kesalahan, misalnya karena data yang diinputkan salah
	6.a Login gagal karena data yang diinputkan belum terdaftar sebagai member
Kondisi akhir	Login berhasil

Pada tabel III.86 menggambarkan skenario untuk merubah data profil pencari kerja di situs lowongan pekerjaan pada use case Perubahan data profil pribadi.

Tabel III. 86 Skenario Use Case Perubahan data profil pribadi

Identifikasi	
Nomor	UC-W-11
Nama Use Case	Perubahan data profil pribadi
Deskripsi	Proses untuk mengedit atau mengubah data/profil pribadi ketika sudah terdaftar sebagai member pencari kerja di situs lowongan pekerjaan
Aktor	Pencari Kerja
Skenario Utama(UC-W-11-S01)	
Kondisi awal	Berhasil melakukan login sebagai pencari kerja dan berada pada halaman member
Aksi aktor	**Respon Sistem**
1. Memilih menu Edit profil	2. Menampilkan menu form edit profil pribadi
3. Mengubah data pada form data pribadi	4. Mengecek validasi inputan data member
	5. Memanggil fungsi editPck
	6. Menyimpan data ke dalam database
	7. Menampilkan pesan bahwa data telah berhasil disimpan
Skenario Alternatif (UC-W-11-S02) Edit profil pribadi namun terdapat pesan error	
	4.a Menampilkan pesan kesalahan, misalnya karena data yang diinputkan salah
Kondisi akhir	Menampilkan konfirmasi bahwa data telah disimpan dan edit profil pribadi berhasil

Pada tabel III.87 menggambarkan skenario untuk melakukan proses lamaran kerja oleh pencari kerja di situs lowongan pekerjaan pada use case Lamaran lowongan pekerjaan.

Tabel III. 87 Skenario Use Case Lamaran lowongan pekerjaan

Identifikasi	
Nomor	UC-W-12
Nama Use Case	Lamaran lowongan pekerjaan
Deskripsi	Proses untuk melakukan lamaran dari lowongan pekerjaan oleh member pencari kerja di situs lowongan pekerjaan
Aktor	Pencari Kerja
Skenario Utama(UC-W-12-S01)	
Kondisi awal	Berhasil melakukan login sebagai pencari kerja dan berada pada halaman detil lowongan pekerjaan
Aksi aktor	**Respon Sistem**
1. Menekan tombol *apply*	2. Memanggil fungsi applyJob
	3. Menyimpan data ke dalam database
	4. Menampilkan pesan bahwa lamaran telah berhasil dilakukan
Skenario Alternatif (UC-W-12-S02) Lamaran lowongan pekerjaan jika user belum login sebagai member pencari kerja	
	2.a Sistem akan langsung mengarahkan ke halaman login
Kondisi akhir	Menampilkan konfirmasi bahwa lamaran telah berhasil dilakukan

Pada tabel III.88 menggambarkan skenario untuk membuat data lowongan pekerjaan baru di situs lowongan pekerjaan pada use case Pembuatan lowongan pekerjaan baru.

Tabel III. 88 Skenario Use Case Pembuatan lowongan pekerjaan baru

Identifikasi	
Nomor	UC-W-13
Nama Use Case	Pembuatan lowongan pekerjaan baru
Deskripsi	Proses untuk membuat lowongan pekerjaan baru
Aktor	Member pemberi kerja
Skenario Utama(UC-W-13-S01)	
Kondisi awal	Berhasil login sebagai pemberi kerja
Aksi aktor	**Respon Sistem**
1. Memilih menu/halaman member	2. Menampilkan halaman member
3. Memilih *button* tambah lowongan	4. Menampilkan form tambah lowongan
5. Mengisi data pada form tambah lowongan	6. Mengecek validasi inputan data lowongan
	7. Memanggil fungsi postJob
	8. Menyimpan data lowongan baru ke database
	9. Menampilkan pesan bahwa data lowongan baru telah berhasil disimpan
Skenario Alternatif (UC-W-13-S02) Pembuatan lowongan pekerjaan baru namun terdapat pesan error	
	6.a Menampilkan pesan kesalahan, misalnya karena data yang diinputkan salah
Kondisi akhir	Tambah lowongan baru berhasil

Pada tabel III.89 menggambarkan skenario untuk melakukan proses edit atau merubah data lowongan pekerjaan oleh pemberi kerja di situs lowongan pekerjaan pada use case Perubahan data lowongan.

Tabel III. 89 Skenario Use Case Perubahan data lowongan

Identifikasi	
Nomor	UC-W-14
Nama Use Case	Perubahan data lowongan
Deskripsi	Proses untuk mengedit atau merubah data lowongan pekerjaan
Aktor	Member pemberi kerja
Skenario Utama(UC-W-14-S01)	
Kondisi awal	Berhasil login sebagai pemberi kerja dan berada pada halaman member
Aksi aktor	**Respon Sistem**
1. Memilih lowongan pekerjaan	2. Menampilkan *option edit* lowongan
3. Memilih *option* edit lowongan	4. Menampilkan form edit lowongan
5. Mengubah data lowongan pekerjaan	6. Mengecek validasi inputan data lowongan
	7. Memanggil fungsi postJob
	8. Menyimpan data lowongan baru ke database
	9. Menampilkan pesan bahwa data lowongan baru telah berhasil disimpan
Skenario Alternatif (UC-W-14-S02) Perubahan data lowongan namun terdapat pesan error	
	6.a Menampilkan pesan kesalahan, misalnya karena data yang diinputkan salah
Kondisi akhir	Edit data lowongan pekerjaan berhasil

Pada tabel III.90 menggambarkan skenario untuk melakukan proses hapus data lowongan pekerjaan oleh pemberi kerja di situs lowongan pekerjaan pada use case Penghapusan lowongan.

Tabel III. 90 Skenario Use Case Penghapusan Lowongan

Identifikasi	
Nomor	UC-W-15
Nama Use Case	Penghapusan lowongan
Deskripsi	Proses menghapus data lowongan pekerjaan
Aktor	Member pemberi kerja
Skenario Utama(UC-W-15-S01)	
Kondisi awal	Berhasil login sebagai pemberi kerja dan berada pada halaman member
Aksi aktor	**Respon Sistem**
1. Memilih lowongan pekerjaan	2. Menampilkan *delete* lowongan
3. Memilih *option* delete lowongan	4. Menampilkan pesan konfirmasi hapus lowongan
5. Memilih tombol hapus lowongan	6. Mengambil parameter id lowongan
	7. Memanggil fungsi deleteJob
	8. Menghapus data lowongan pekerjaan
	9. Menampilkan pesan bahwa data lowongan berhasil dihapus
Kondisi akhir	Penghapusan data lowongan pekerjaan berhasil

Pada tabel III.91 menggambarkan skenario untuk melakukan proses perubahan data profil perusahaan oleh pemberi kerja di situs lowongan pekerjaan pada use case Perubahan data profil perusahaan.

Tabel III. 91 Use Case Perubahan data profil perusahaan

Identifikasi	
Nomor	UC-W-16
Nama Use Case	Perubahan data profil perusahaan
Deskripsi	Proses untuk meng-*update* atau mengubah data profil perusahaan
Aktor	Member pemberi kerja
Skenario Utama(UC-W-16-S01)	
Kondisi awal	Berhasil login sebagai pemberi kerja dan berada pada halaman member
Aksi aktor	**Respon Sistem**
1. Memilih menu *button* Edit Profil	2. Menampilkan halaman edit profil perusahaan
3. Mengubah data profil perusahaan	4. Mengecek validasi inputan data profil perusahaan
	5. Memanggil fungsi updatePmk
	6. Menyimpan perubahan data profil ke database
	7. Menampilkan pesan bahwa data profil perusahaan telah berhasil diupdate
Skenario Alternatif (UC-W-16-S02) Perubahan data profil perusahaan namun terdapat pesan error	
	4.a Menampilkan pesan kesalahan, misalnya karena data yang diinputkan salah
Kondisi akhir	Update / perubahan data profil perusahaan berhasil

Pada tabel III.92 menggambarkan skenario untuk melihat daftar lamaran pekerjaan dari pencari kerja pada use case Lihat lamaran pekerjaan.

Tabel III. 92 Skenario Use Case Lihat lamaran pekerjaan

Identifikasi	
Nomor	UC-W-17
Nama Use Case	Lihat lamaran pekerjaan
Deskripsi	Proses untuk melihat daftar lamaran pekerjaan dari pencari kerja
Aktor	Member pemberi kerja
Skenario Utama(UC-W-17-S01)	
Kondisi awal	Berhasil login sebagai pemberi kerja
Aksi aktor	**Respon Sistem**
1. Memilih menu *button* Lamaran	2. Menampilkan halaman lamaran pekerjaan
3. Memilih salah satu lamaran pada *list view* lamaran	4. Menampilkan halaman detil lamaran dari pencari kerja
Kondisi akhir	Update / perubahan data profil perusahaan berhasil

2. **Activity Diagram**

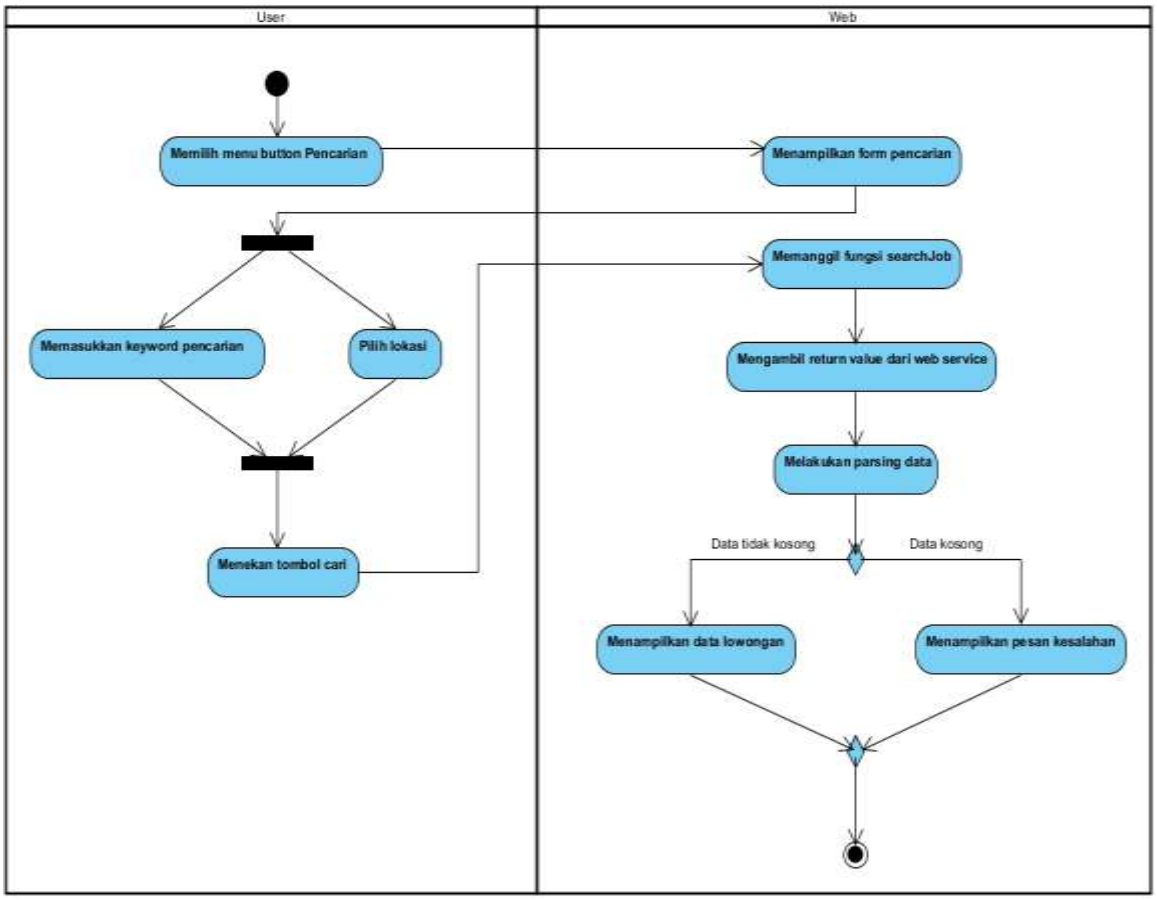
a. Activity diagram Pencarian lowongan

Aktivitas Pencarian lowongan merupakan aktivitas yang dilakukan user untuk mencari lowongan pekerjaan berdasarkan *keyword* dan lokasi lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Membuka menu *button* pencarian, pada aktivitas ini *user* menekan menu *button* pencarian yang terdapat pada halaman utama.
2. Menampilkan form pencarian, pada aktivias ini sistem menampilkan form pencarian untuk mencari lowongan pekerjaan.
3. Memasukkan keyword dan memilih lokasi lowongan pekerjaan, pada aktivitas ini user memasukkan keyword serta memilih lokasi dari lowongan pekerjaan.
4. Menekan tombol cari, pada aktivitas ini user mengklik tombol cari untuk mengeksekusi proses pencarian.
5. Memanggil fungsi searchJob, pada aktivitas ini sistem memanggil fungsi searchJob.
6. Mengambil *return value* dari *web service*, pada aktivitas ini sistem mengambil nilai yang di kembalikan oleh *web service* berupa hasil dari eksekusi method pada *web service*.
7. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
8. Menampilkan data lowongan pekerjaan, pada aktivitas ini sistem menampilkan hasil parsing kedalam bentuk *list view*.

9. Menampilkan pesan kesalahan, aktivitas ini akan dilakukan jika data lowongan yang dicari berdasarkan parameter input dari user tidak ditemukan.

Seluruh aktivitas Pencarian lowongan terlihat pada *activity diagram* gambar III.44.



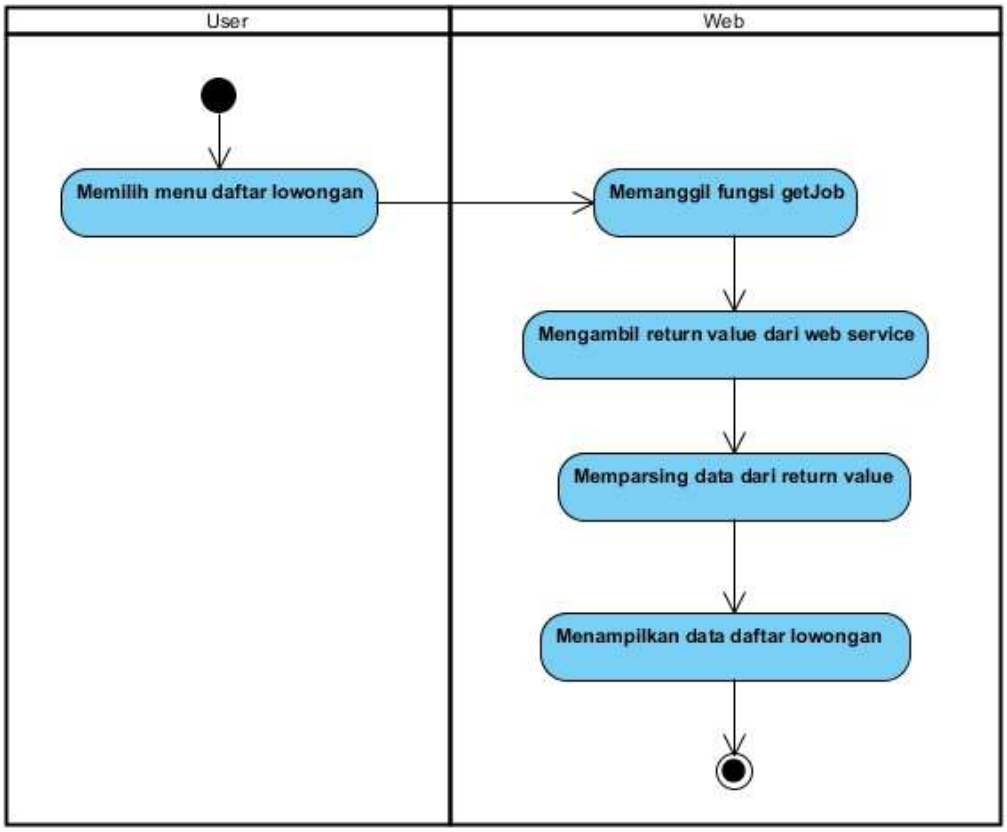
Gambar III. 44 Activity Diagram Pencarian lowongan

b. Activity diagram Lihat daftar lowongan

Aktivitas Lihat daftar lowongan merupakan aktivitas yang dilakukan user untuk melihat daftar lowongan pekerjaan yang terdapat di situs lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu daftar lowongan, pada aktivitas ini *user* memilih menu daftar lowongan.
2. Memanggil fungsi getJob, pada aktivitas ini sistem akan memanggil fungsi getJob.
3. Mengambil *return value* dari web service, pada aktivitas ini sistem mengambil nilai yang di kembalikan oleh web service berupa hasil dari eksekusi *method* pada web service.
4. Melakukan parsing data, pada aktivitas ini sistem memparsing respon dari web service agar data yang diterima dapat dibaca.
5. Menampilkan data daftar lowongan, pada aktivitas ini sistem menampilkan daftar lowongan pekerjaan dalam bentuk *list view*.

Seluruh aktivitas Lihat detil lowongan terlihat pada *activity diagram* gambar III.45.



Gambar III. 45 Activity Diagram Lihat daftar lowongan

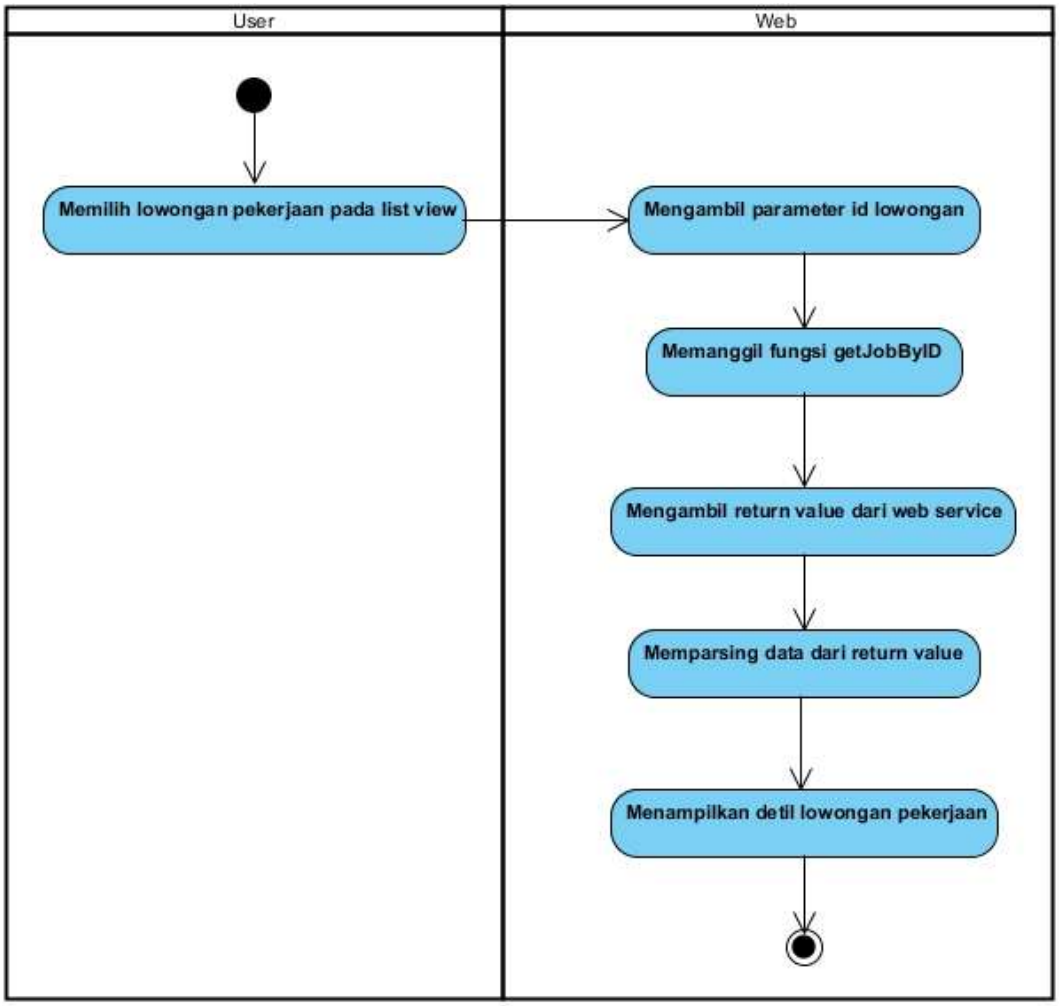
c. Activity diagram Lihat detil lowongan

Aktivitas Lihat detil lowongan merupakan aktivitas yang dilakukan user untuk melihat detil dari lowongan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih lowongan pekerjaan pada list view, pada aktivitas ini *user* mengklik lowongan perusahaan untuk melihat detil dari lowongan tersebut.
2. Mengambil parameter id lowongan, pada aktivias ini sistem mengambil id dari lowongan pekerjaan yang dipilih oleh user sebagai parameter untuk menampilkan detil lowongan.

3. Memanggil fungsi getJob, pada aktivitas ini sistem akan memanggil fungsi getJob.
4. Mengambil *return value* dari *web service*, pada aktivitas ini sistem mengambil nilai yang di kembalikan oleh *web service* berupa hasil dari eksekusi method pada *web service*.
5. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
6. Menampilkan detil lowongan pekerjaan, pada aktivitas ini sistem menampilkan detil lowongan pekerjaan.

Seluruh aktivitas Lihat detil lowongan terlihat pada *activity diagram* gambar III.46.



Gambar III. 46 Activity Diagram Lihat detil lowongan

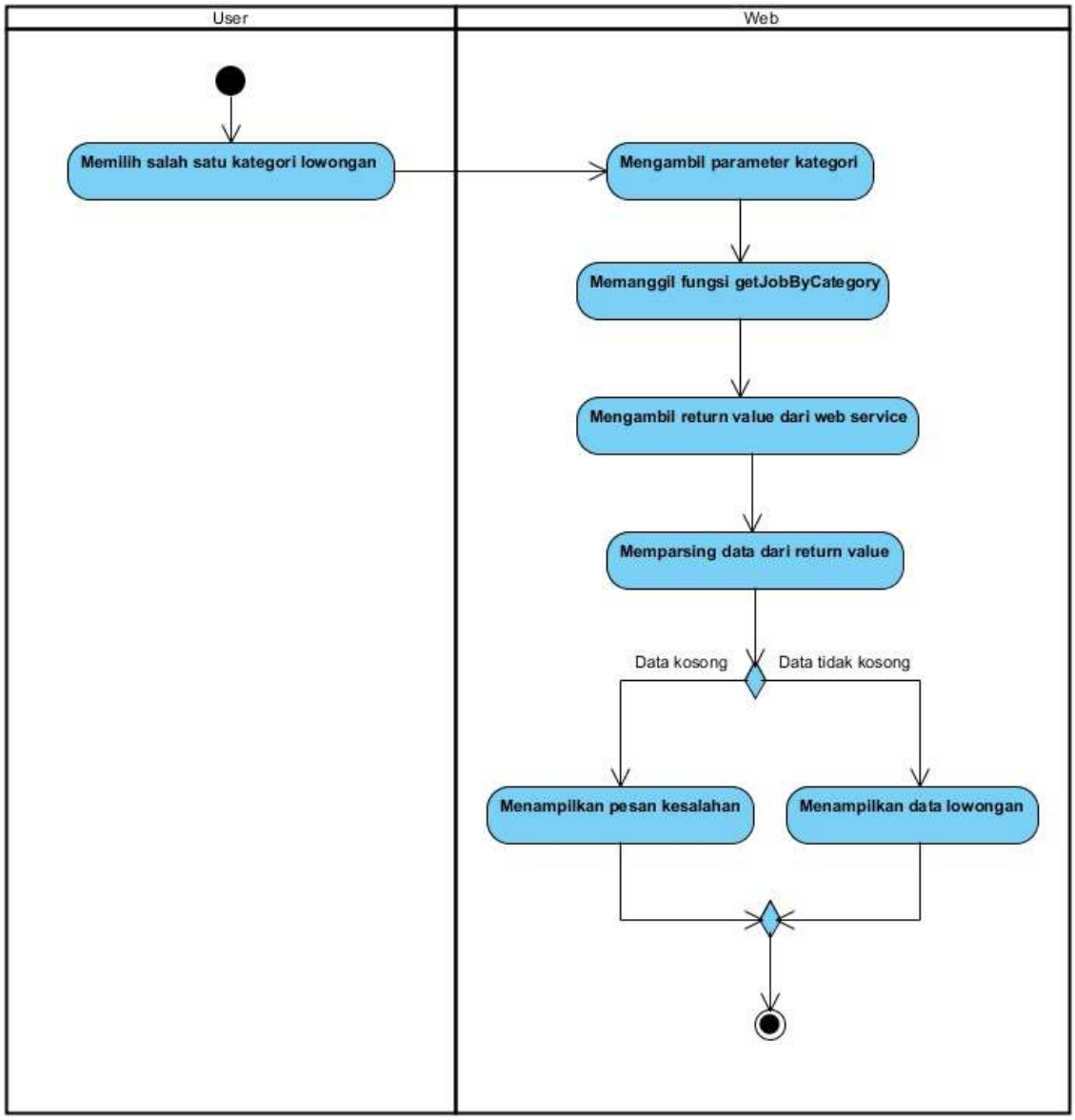
d. Activity diagram Lihat lowongan berdasarkan kategori

Aktivitas Lihat lowongan berdasarkan kategori merupakan aktivitas yang dilakukan user untuk melihat lowongan pekerjaan berdasarkan kategori tertentu. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih salah satu kategori lowongan, pada aktivitas ini *user* mengklik salah satu kategori lowongan pekerjaan.

2. Mengambil parameter kategori, pada aktivias ini sistem mengambil kategori dari lowongan pekerjaan yang dipilih oleh *user* sebagai parameter untuk menampilkan data lowongan berdasarkan kategori yang dipilih.
3. Memanggil fungsi getJobByCategory, pada aktivitas ini sistem akan memanggil fungsi getJobByCategory.
4. Mengambil *return value* dari *web service*, pada aktivitas ini sistem mengambil nilai yang di kembalikan oleh *web service* berupa hasil dari eksekusi method pada *web service*.
5. Melakukan *parsing* data, pada aktivitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
6. Menampilkan pesan kesalahan, aktivitas ini terjadi jika data kosong.
7. Menampilkan data lowongan pekerjaan, pada aktivitas ini sistem menampilkan daftar lowongan pekerjaan berdasarkan kategori yang telah dipilih oleh *user*.

Seluruh aktivitas Lihat lowongan berdasarkan kategori terlihat pada *activity diagram* gambar III.47.



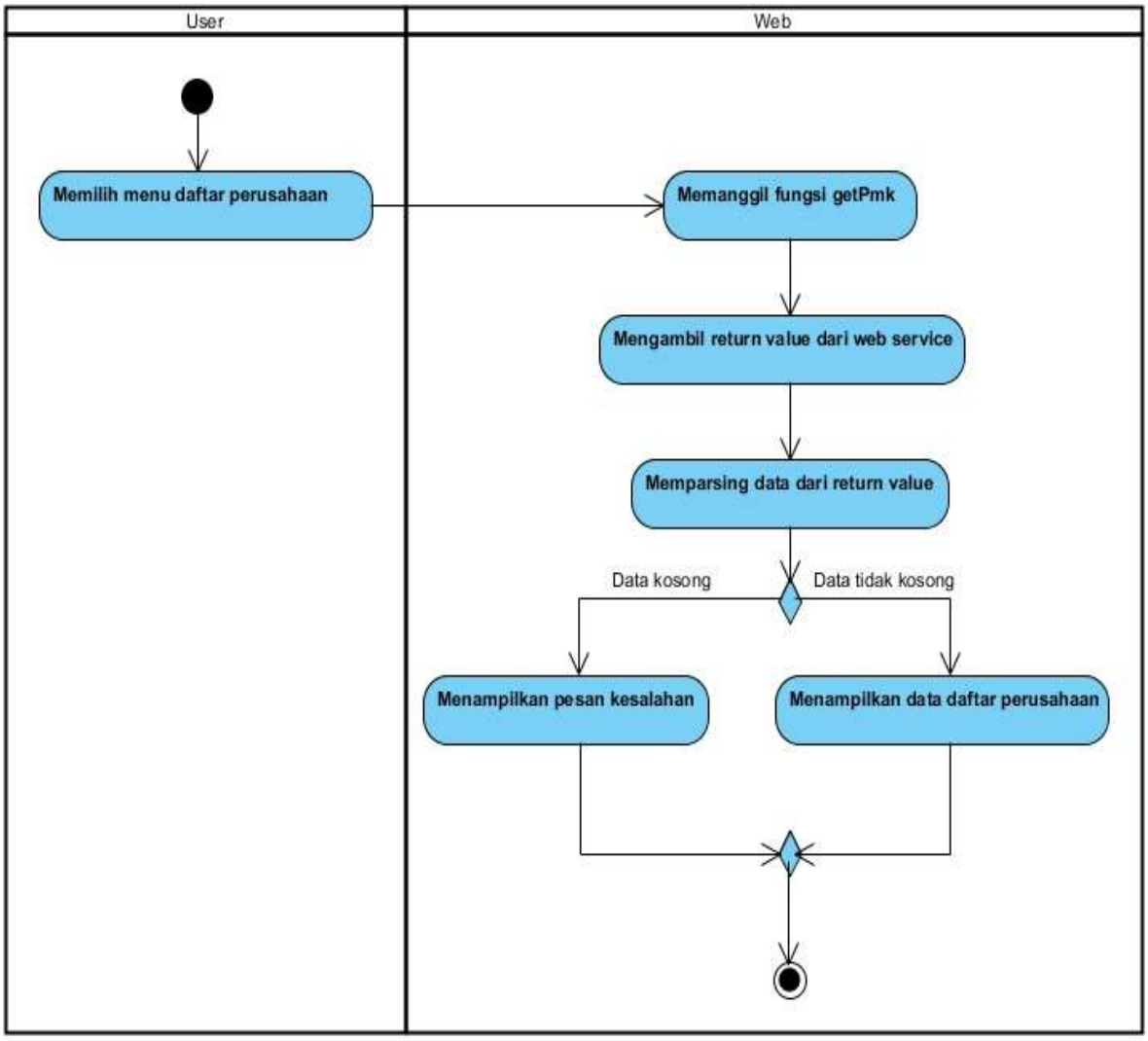
Gambar III. 47 Activity Diagram Lihat lowongan berdasarkan kategori

e. Activity diagram Lihat daftar perusahaan

Aktivitas Lihat daftar perusahaan merupakan aktivitas yang dilakukan user untuk melihat daftar (list) perusahaan yang terdaftar di situs lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu daftar perusahaan, pada aktivitas ini *user* mengklik menu daftar perusahaan.
2. Memanggil fungsi getPmk, pada aktivias ini sistem memanggil fungsi getPmk.
3. Mengambil *return value* dari *web service*, pada aktivitas ini sistem mengambil nilai yang di kembalikan oleh *web service* berupa hasil dari eksekusi method pada *web service*.
4. Melakukan parsing data, pada aktivitas ini sistem memparsing respon dari web service agar data yang diterima dapat dibaca.
5. Menampilkan pesan kesalahan, aktivitas ini terjadi jika data kosong.
6. Menampilkan data daftar perusahaan, pada aktivitas ini sistem menampilkan daftar perusahaan yang telah terdaftar di situs lowongan pekerjaan.

Seluruh aktivitas Lihat daftar perusahaan terlihat pada *activity diagram* gambar III.48.



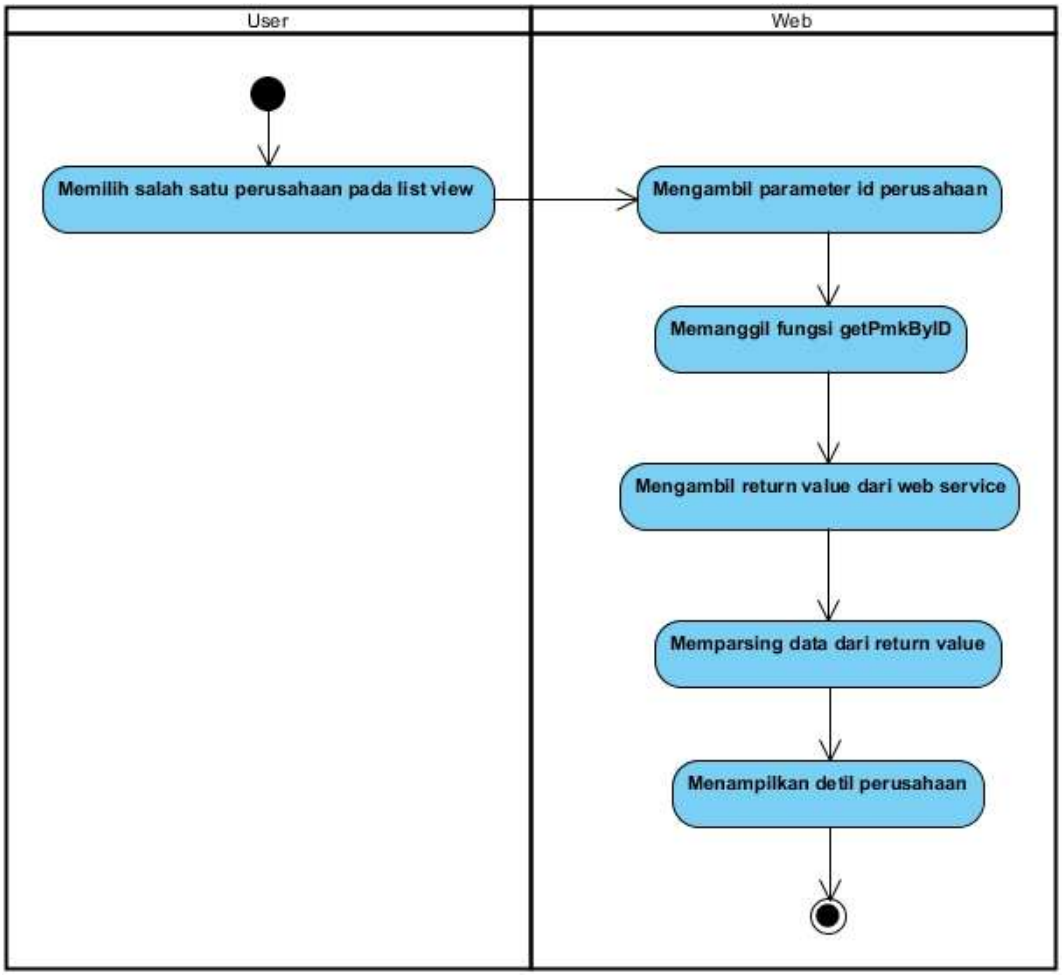
Gambar III. 48 Activity Diagram Lihat daftar perusahaan

f. Activity diagram Lihat detil perusahaan

Aktivitas Lihat detil perusahaan merupakan aktivitas yang dilakukan *user* untuk melihat detil dari perusahaan yang terdaftar di situs lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih salah satu perusahaan pada *list view*, pada aktivitas ini *user* memilih salah satu perusahaan yang terdapat pada *list view*.
2. Memanggil parameter id lowongan, pada aktivitas ini sistem akan mengambil id lowongan yang dipilih user yang akan digunakan sebagai parameter untuk melihat detil lowongan.
3. Memanggil fungsi getPmkByID, pada aktivias ini sistem memanggil fungsi getPmkByID.
4. Mengambil return value dari web service, pada aktivitas ini sistem mengambil nilai yang di kembalikan oleh web service berupa hasil dari eksekusi method pada web service.
5. Melakukan parsing data, pada aktivitas ini sistem memparsing respon dari web service agar data yang diterima dapat dibaca.
6. Menampilkan detil perusahaan, pada aktivitas ini sistem menampilkan detil perusahaan.

Seluruh aktivitas Lihat detil perusahaan terlihat pada *activity diagram* gambar III.49.



Gambar III. 49 Activity Diagram Lihat detil perusahaan

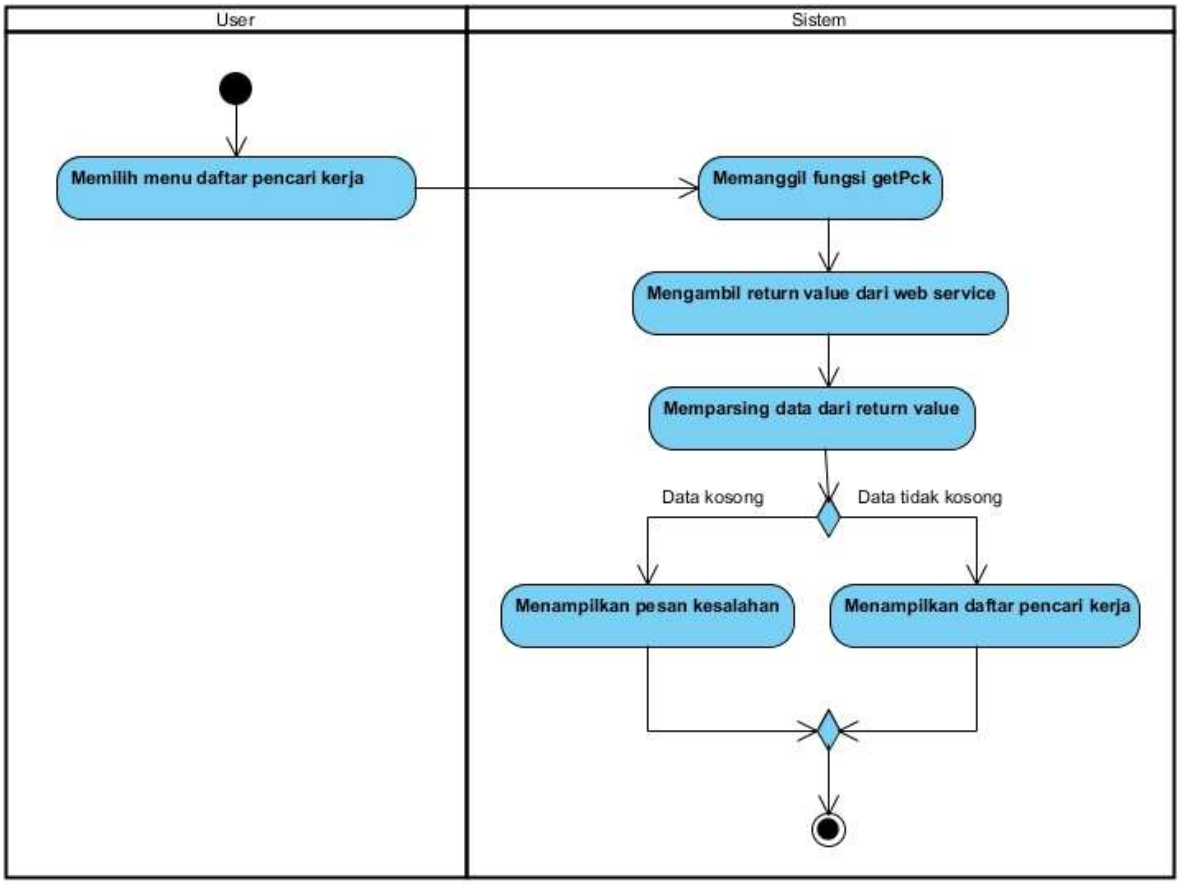
g. Activity diagram Lihat daftar pencari kerja

Aktivitas Lihat daftar pencari kerja merupakan aktivitas yang dilakukan user untuk melihat daftar (list) pencari kerja yang terdaftar di situs lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu daftar pencari kerja, pada aktivitas ini *user* mengklik menu hyperlink daftar pencari kerja.
2. Memanggil fungsi getPck, pada aktivias ini sistem memanggil fungsi getPck.

3. Mengambil return value dari web service, pada aktivitas ini sistem mengambil nilai yang di kembalikan oleh web service berupa hasil dari eksekusi method pada web service.
4. Melakukan parsing data, pada aktivitas ini sistem memparsing respon dari web service agar data yang diterima dapat dibaca.
5. Menampilkan pesan kesalahan, aktivitas ini terjadi jika data kosong.
6. Menampilkan daftar pencari kerja, pada aktivitas ini sistem menampilkan daftar pencari kerja yang telah terdaftar di situs lowongan pekerjaan.

Seluruh aktivitas Lihat daftar pencari kerja terlihat pada *activity diagram* gambar III.50.



Gambar III. 50 Activity Diagram Lihat daftar pencari kerja

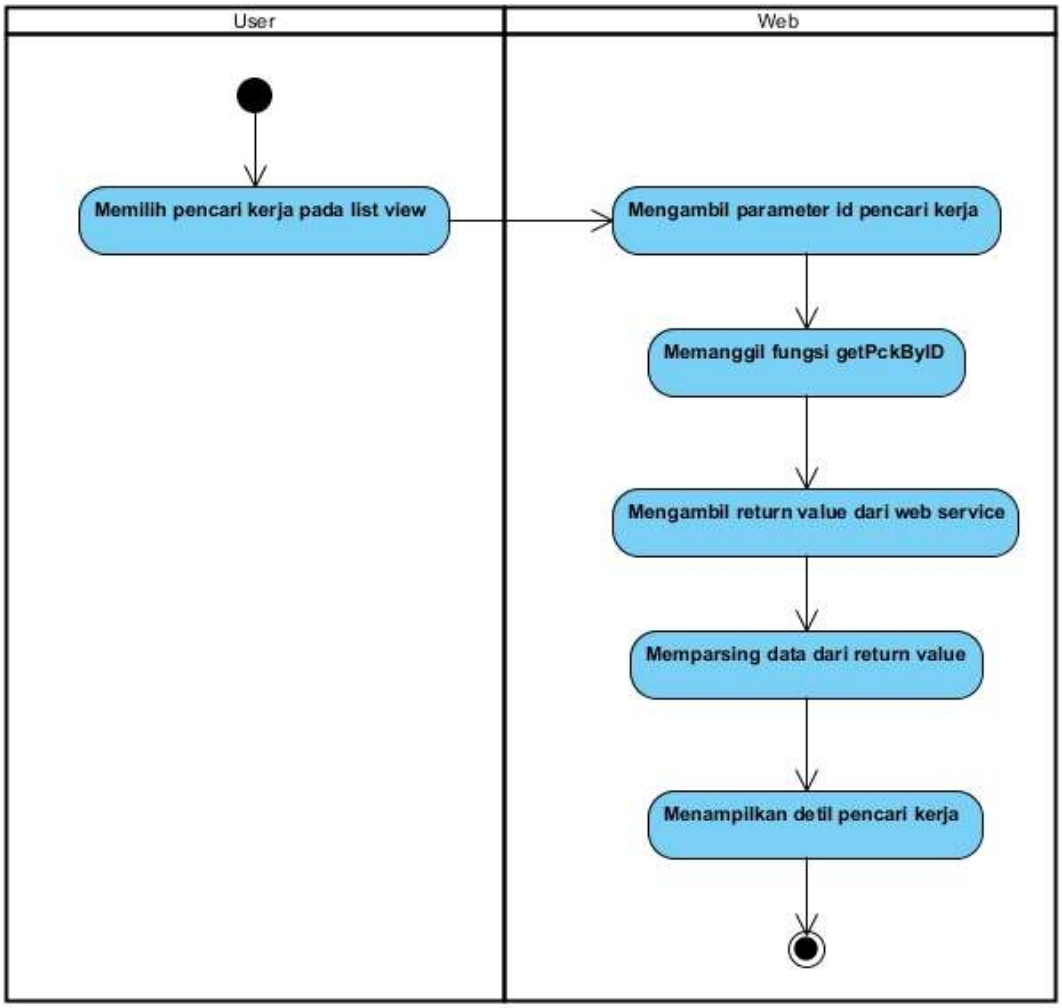
h. Activity diagram Lihat detil pencari kerja

Aktivitas Lihat detil pencari kerja merupakan aktivitas yang dilakukan *user* untuk melihat detil dari pencari kerja yang terdaftar di situs lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih pencari kerja pada *list view*, pada aktivitas ini *user* memilih salah satu pencari kerja yang terdapat pada *list view*.
2. Memanggil parameter id pencari kerja, pada aktivitas ini sistem akan mengambil id pencari kerja yang dipilih user yang akan digunakan sebagai parameter untuk melihat detil pencari kerja.

3. Memanggil fungsi getPckByID, pada aktivias ini sistem memanggil fungsi getPckByID.
4. Mengambil return value dari web service, pada aktivitas ini sistem mengambil nilai yang di kembalikan oleh web service berupa hasil dari eksekusi method pada web service.
5. Melakukan parsing data, pada aktivitas ini sistem memparsing respon dari web service agar data yang diterima dapat dibaca.
6. Menampilkan detil pencari kerja, pada aktivitas ini sistem menampilkan detil pencari kerja.

Seluruh aktivitas Lihat detil pencari kerja terlihat pada *activity diagram* gambar III.51.



Gambar III. 51 Activity Diagram Lihat detil pencari kerja

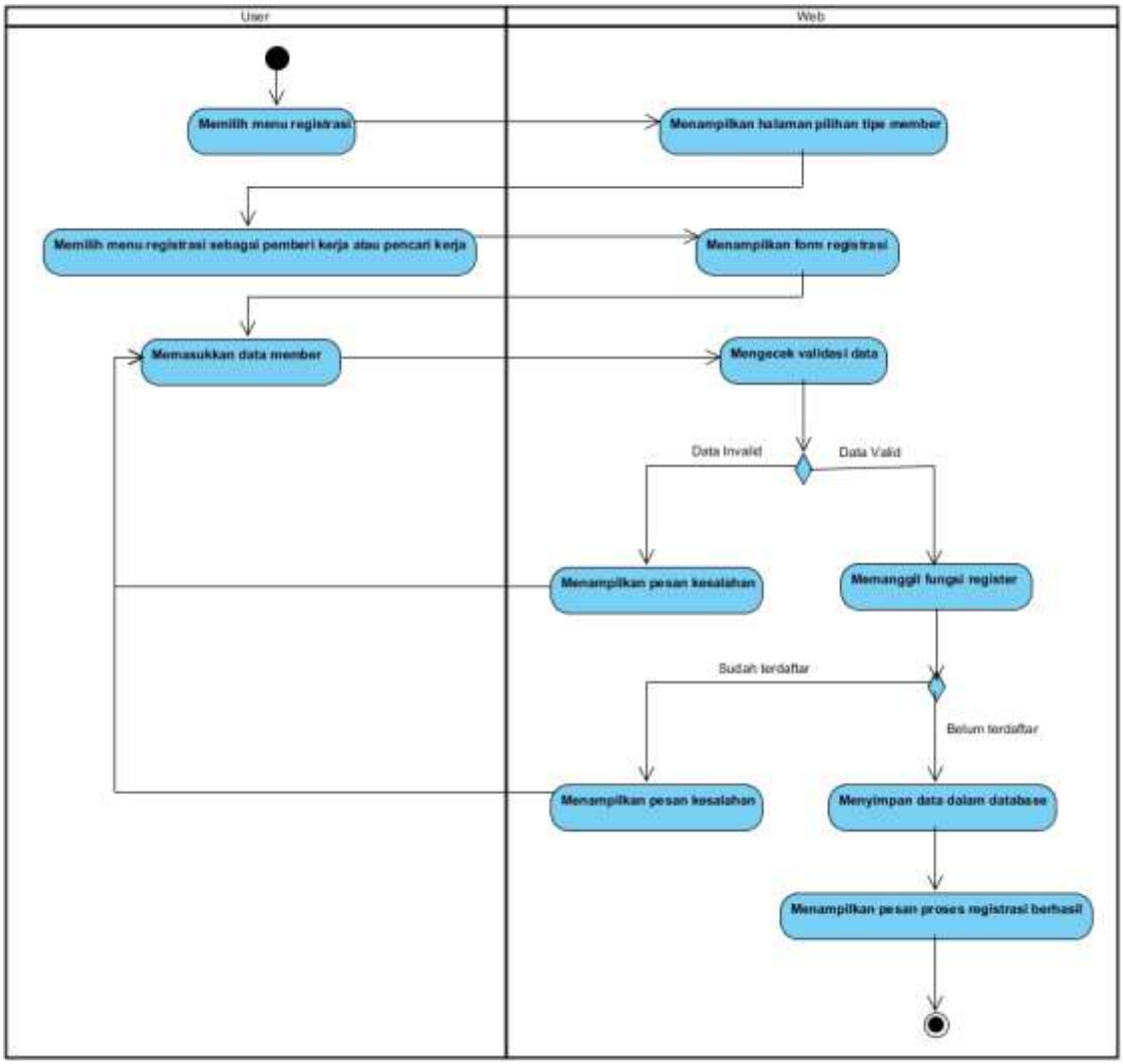
i. Activity diagram Registrasi

Aktivitas registrasi merupakan aktivitas yang dilakukan user untuk mendaftarkan diri sebagai pemberi kerja atau pencari kerja dan membuat data pribadi atau perusahaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu registrasi, pada aktivitas ini *user* memilih tombol registrasi untuk memulai proses registrasi.
2. Menampilkan pilihan tipe member, pada aktivitas ini sistem akan menampilkan pilihan tipe member untuk melakukan registrasi.

3. Memilih menu registrasi sebagai pemberi kerja atau pencari kerja, pada aktivitas ini *user* memilih menu registrasi sebagai pemberi kerja atau pencari kerja.
4. Menampilkan form registrasi, pada aktivitas ini sistem akan menampilkan menu form registrasi sesuai dengan pilihan *user* sebelumnya.
5. Memasukkan data *member*, pada aktivitas ini user mengisi data diri pada form yang telah tersedia.
6. Mengecek validasi data, pada aktivitas ini sistem akan mengecek apakah data yang diinputkan oleh *user* telah sesuai dengan format input atau belum.
7. Menampilkan pesan kesalahan, aktivitas ini terjadi jika data yang diinputkan salah atau belum sesuai dengan format atau belum diisi.
8. Memanggil fungsi login, pada aktivitas ini sistem akan memanggil fungsi loginPmk untuk registrasi sebagai pemberi kerja atau fungsi loginPck untuk registrasi sebagai pencari kerja.
9. Menyimpan data dalam database, pada aktivitas ini sistem akan menyimpan data member baru ke dalam database.
10. Menampilkan pesan kesalahan, aktivitas ini terjadi jika data member sudah terdaftar sebelumnya.
11. Menampilkan pesan registrasi berhasil, pada aktivitas ini sistem akan menampilkan pesan bahwa proses registrasi berhasil.

Seluruh aktivitas Registrasi terlihat pada *activity diagram* gambar III.52.



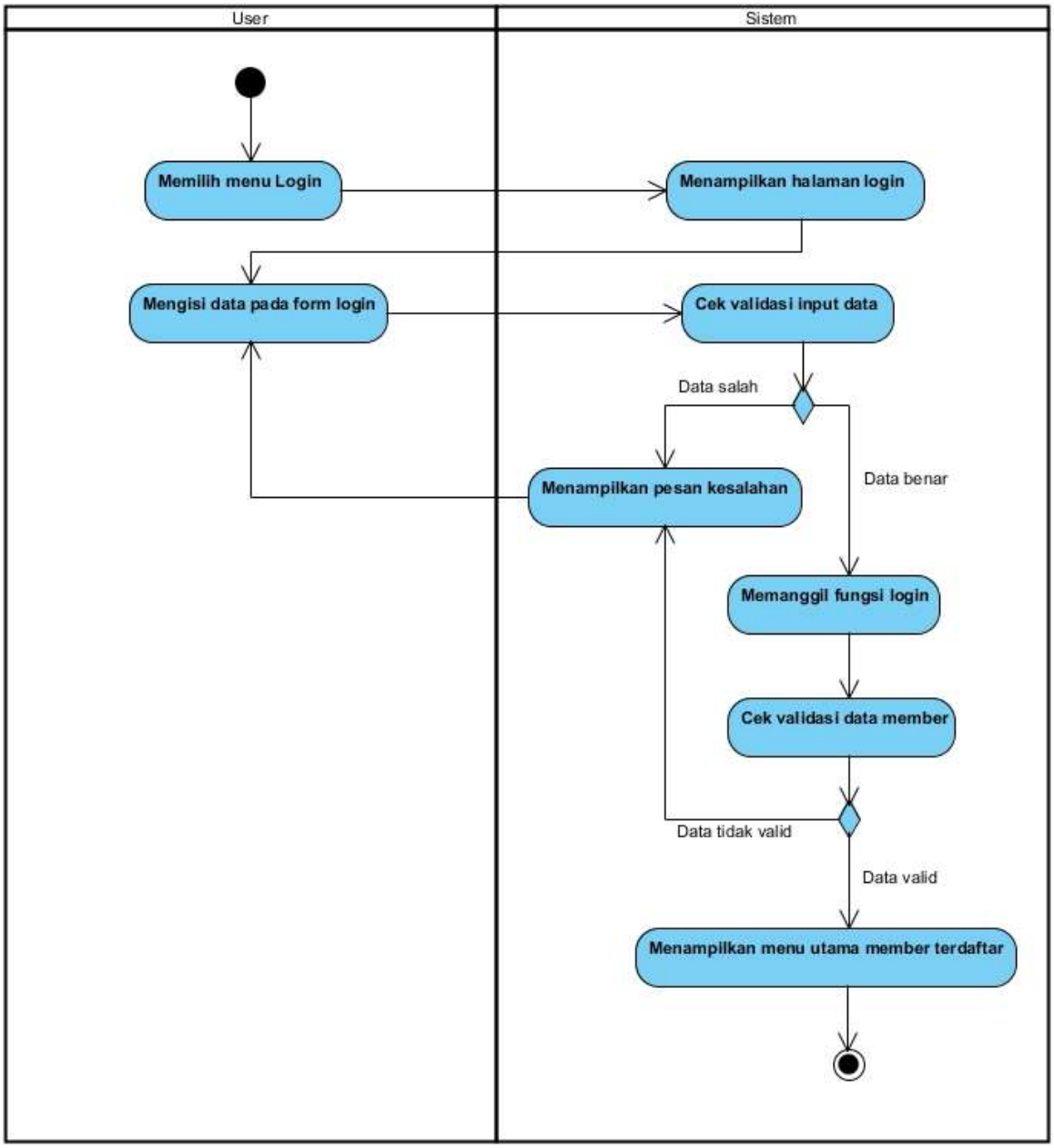
Gambar III. 52 Activity Diagram Registrasi

j. Activity diagram Login

Aktivitas *login* merupakan aktivitas yang dilakukan *user* untuk mandapatkan akses untuk mengelola lowongan pekerjaan jika *login* sebagai pemberi kerja dan membuat profil data pribadi sebagai pencari kerja. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu *login*, pada aktivitas ini *user* meneekan menu *login* untuk membuka halaman *login*.
2. Menampilkan halaman login, pada aktivitas ini sistem menampilkan form *login*.
3. Mengisi data pada form *login*, pada aktivitas ini user memasukkan data pada form *login*.
4. Cek validasi input data, pada aktivitas ini sistem mengecek inputan dari user apakah sudah sesuai dengan format input atau belum.
5. Menampilkan pesan kesalahan, aktivitas ini terjadi jika data yang diinputkan salah atau belum sesuai dengan format atau belum diisi.
6. Memanggil fungsi login, pada aktivitas ini sistem akan memanggil fungsi loginPmk untuk pemberi kerja atau loginPck untuk pencari kerja.
7. Cek validasi data member, pada aktivitas ini sistem mencocokan data member yang user inputkan dengan data member yang ada pada database.
8. Menampilkan pesan kesalahan, aktivitas ini terjadi jika input data salah atau data member tidak valid.
9. Menampilkan menu utama member terdaftar, aktivitas ini terjadi jika data member yang *user* inputkan ada pada database lalu sistem menampilkan menu utama member terdaftar, dimana terdapat menu tambahan yang membedakan menu member terdaftar dengan member tidak terdaftar.

Seluruh aktivitas login terlihat pada *activity diagram* gambar III.53.



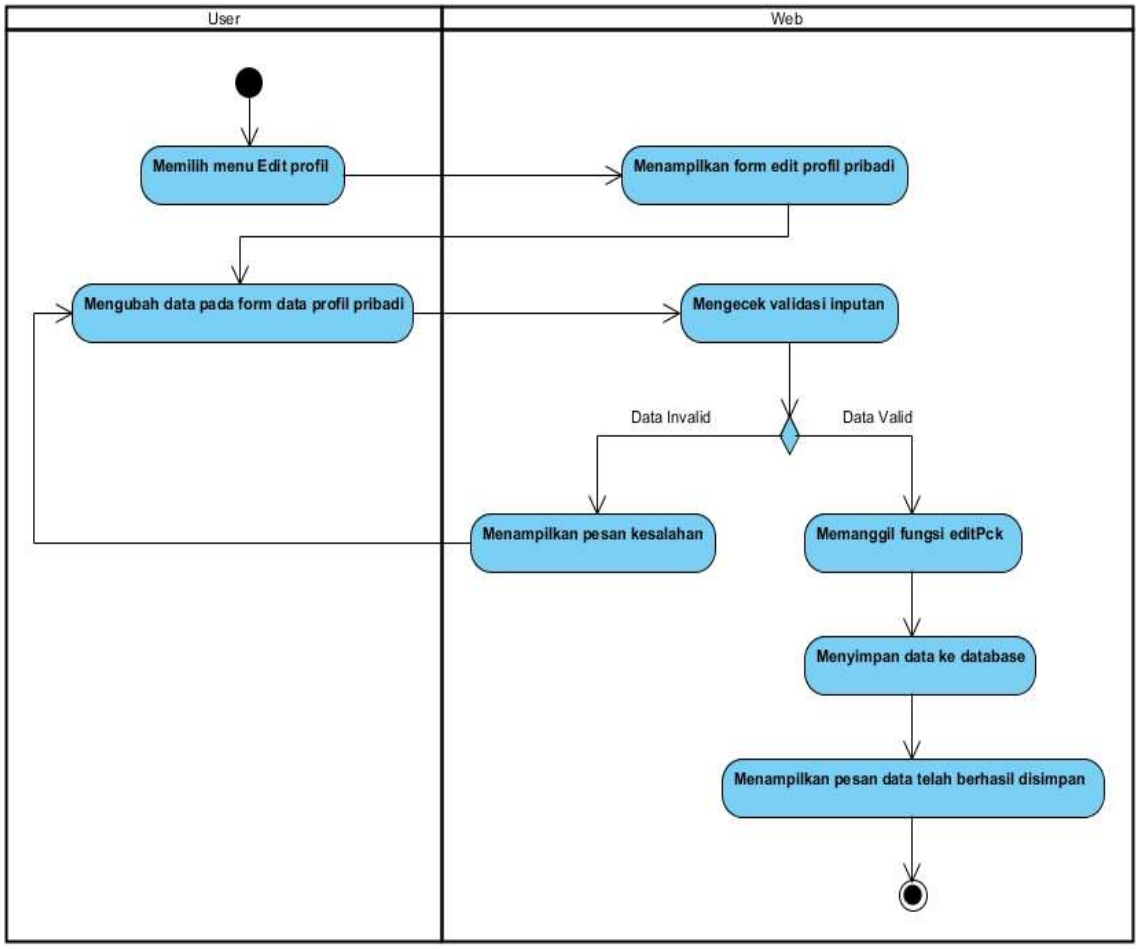
Gambar III. 53 Activity Diagram Login

k. Activity diagram Perubahan data profil pribadi

Aktivitas Perubahan data profil pribadi merupakan aktivitas yang dilakukan *user* pencari kerja untuk melakukan perubahan data profil pribadi . Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu Edit profil, pada aktivitas ini *user* menekan menu *button* Edit profil untuk melakukan perubahan data profil pribadi.
2. Menampilkan form edit profil pribadi, pada aktivitas ini sistem akan menampilkan halaman form edit profil pribadi.
3. Mengubah data pada form data profil pribadi, pada aktivitas ini user akan mengubah data profil pribadi pada form profil pribadi.
4. Mengecek validasi inputan, pada aktivitas ini sistem akan mengecek inputan data profil pribadi sudah sesuai format atau belum.
5. Menampilkan pesan kesalahan, aktivitas ini terjadi jika data yang diinputkan salah atau belum sesuai dengan format atau belum diisi.
6. Memanggil fungsi editPck, pada aktivitas ini sistem akan memanggil fungsi editPck.
7. Menyimpan data ke database, pada aktivitas ini sistem akan menyimpan perubahan data profil pribadi ke dalam database.
8. Menampilkan pesan data telah berhasil disimpan, pada aktivitas ini sistema akan menampilkan pesan pemberitahuan bahwa data profil pribadi telah berhasil disimpan.

Seluruh aktivitas Perubahan data profil pribadi terlihat pada *activity diagram* gambar III.54.



Gambar III. 54 Activity Diagram Perubahan data profil pribadi

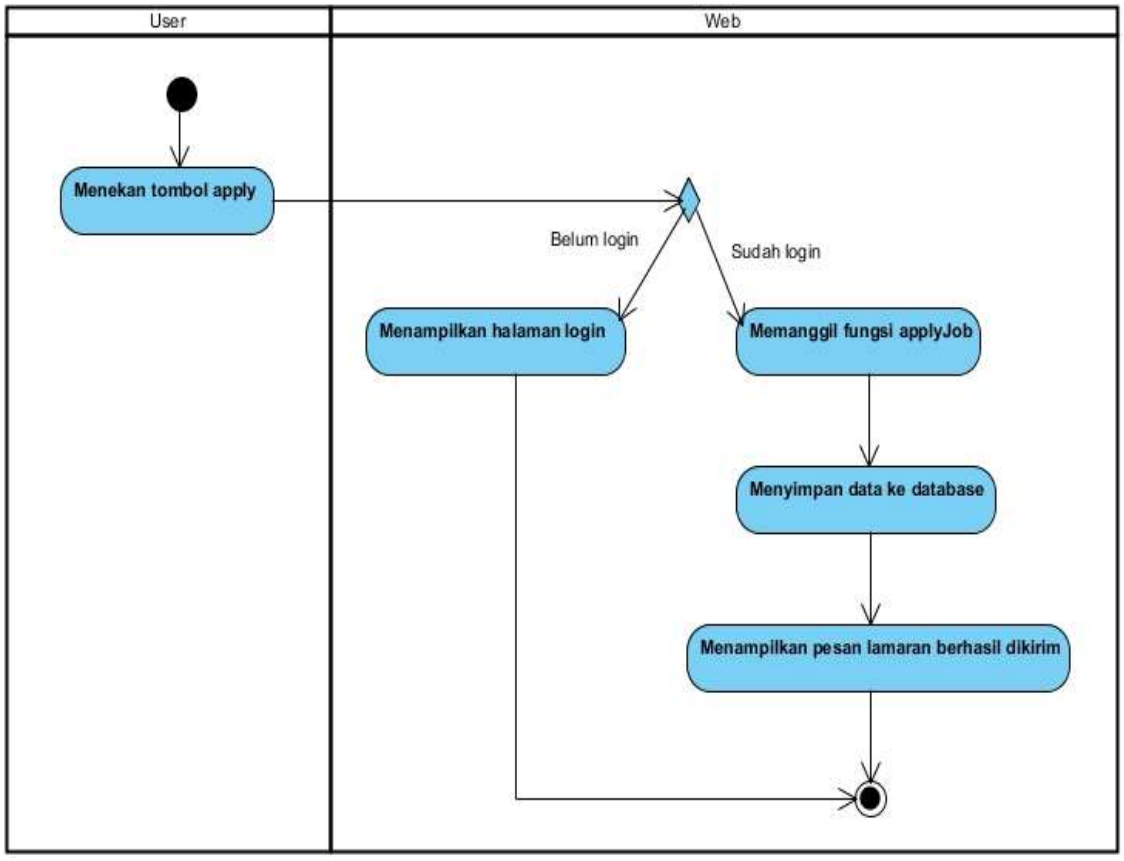
l. Activity diagram Lamaran lowongan pekerjaan

Aktivitas Lamaran lowongan pekerjaan merupakan aktivitas yang dilakukan *user* pencari kerja melakukan proses lamaran pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Menekan tombol *apply*, pada aktivitas ini *user* pencari kerja menekan tombol *apply* untuk melakukan proses lamaran terhadap lowongan pekerjaan yang sedang dibuka oleh pencari kerja.
2. Menampilkan halaman login, aktivitas ini terjadi jika *user* belum melakukan login sebagai member pencari kerja.

3. Memanggil fungsi applyJob, pada aktivitas ini sistem akan memanggil fungsi applyJob.
4. Menyimpan data ke database, pada aktivitas ini sistem akan menyimpan perubahan data member pencari kerja yang melakukan proses lamaran ke dalam database.
5. Menampilkan pesan lamaran berhasil dikirim, pada aktivitas ini sistem akan menampilkan pesan bahwa lamaran telah berhasil dikirim.

Seluruh aktivitas Perubahan data profil pribadi terlihat pada *activity diagram* gambar III.55.



Gambar III. 55 Activity Diagram Lamaran lowongan pekerjaan

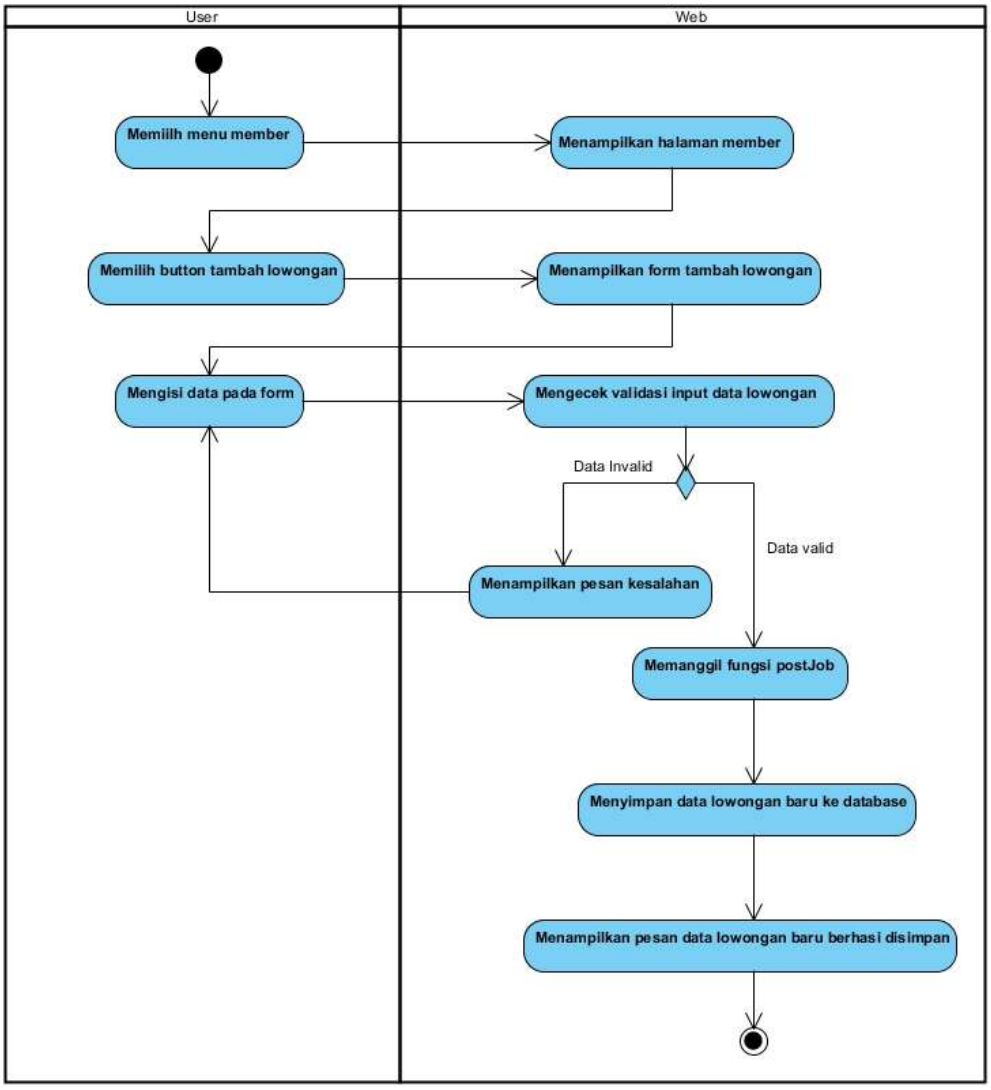
m. Activity diagram Pembuatan lowongan pekerjaan baru

Aktivitas Buat lowongan pekerjaan baru merupakan aktivitas yang dilakukan *user* pemberi kerja untuk membuat data lowongan pekerjaan baru. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu member, pada aktivitas ini *user* pemberi kerja membuka halaman member untuk melakukan proses tambah lowongan baru.
2. Membuka halaman member, pada aktivitas ini sistem menampilkan halaman member dan *button* Tambah Lowongan.
3. Memilih *button* tambah lowongan, pada aktivitas ini user menekan tombol tambah lowongan yang terdapat pada member pemberi kerja.
4. Menampilkan form tambah lowongan, pada aktivitas ini sistem menampilkan halaman form tambah lowongan.
5. Mengisi data pada form, pada aktivitas ini user memasukkan data lowongan pekerjaan baru pada form yang tersedia.
6. Mengecek validasi input data lowongan, pada aktivitas ini sistem mengecek inputan dari user apakah sudah sesuai dengan format input atau belum.
7. Menampilkan pesan kesalahan, aktivitas ini terjadi jika input data salah.
8. Memanggil fungsi postJob, pada aktivitas ini sistem akan memanggil fungsi postJob.
9. Menyimpan data lowongan baru ke dalam database, pada aktivitas ini sistem akan menyimpan data lowongan pekerjaan baru ke dalam database.

10. Menampilkan pesan data lowongan baru berhasil disimpan, pada aktivitas ini sistem akan menampilkan pesan bahwa data lowongan baru telah berhasil disimpan.

Seluruh aktivitas Buat lowongan pekerjaan baru terlihat pada *activity diagram* gambar III.56.



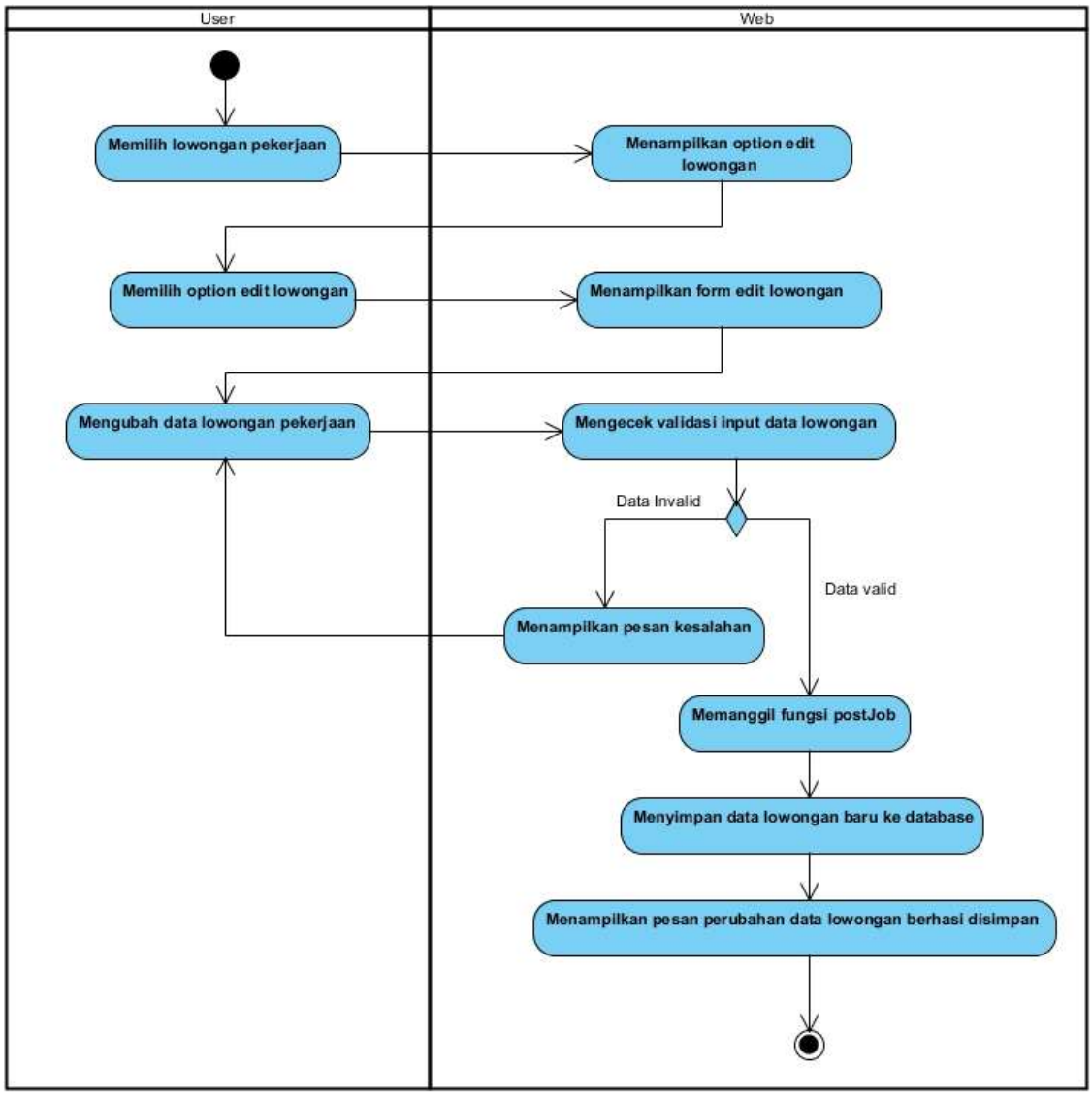
Gambar III. 56 Activity Diagram Buat lowongan pekerjaan baru

n. Activity diagram Perubahan data lowongan

Aktivitas Perubahan data lowongan merupakan aktivitas yang dilakukan *user* pemberi kerja untuk mengubah data lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih lowongan pekerjaan, pada aktivitas ini *user* pemberi kerja memilih lowongan pekerjaan yang akan diedit.
2. Menampilkan *option edit* lowongan, pada aktivitas ini sistem akan menampilkan pilihan edit lowongan untuk melakukan perubahan pada data lowongan pekerjaan.
3. Memilih *option edit* lowongan, pada aktivitas ini pemberi kerja memilih *option* untuk *edit* lowongan pekerjaan.
4. Mengecek validasi input data lowongan, pada aktivitas ini sitem akan memeriksa input data lowongan apakah terisi atau belum dan sudah sesuai format atau belum.
5. Menampilkan pesan kesalahan, aktivitas ini terjadi jika data yang diinputkan salah atau belum sesuai dengan format atau belum diisi.
6. Memanggil fungsi postJob, pada aktivitas ini sistem akan memanggil fungsi postJob.
7. Menyimpan data lowongan baru ke database, pada aktivitas ini sistem akan menyimpan perubahan data lowongan pekerjaan ke dalam database.
8. Menampilkan pesan perubahan data lowongan berhasil disimpan, pada aktivitas ini sistem akan menampilkan pesan bahwa perubahan data lowongan telah berhasil disimpan.

Seluruh aktivitas Perubahan data lowongan terlihat pada *activity diagram* gambar III.57.



Gambar III. 57 Activity Diagram Perubahan data lowongan

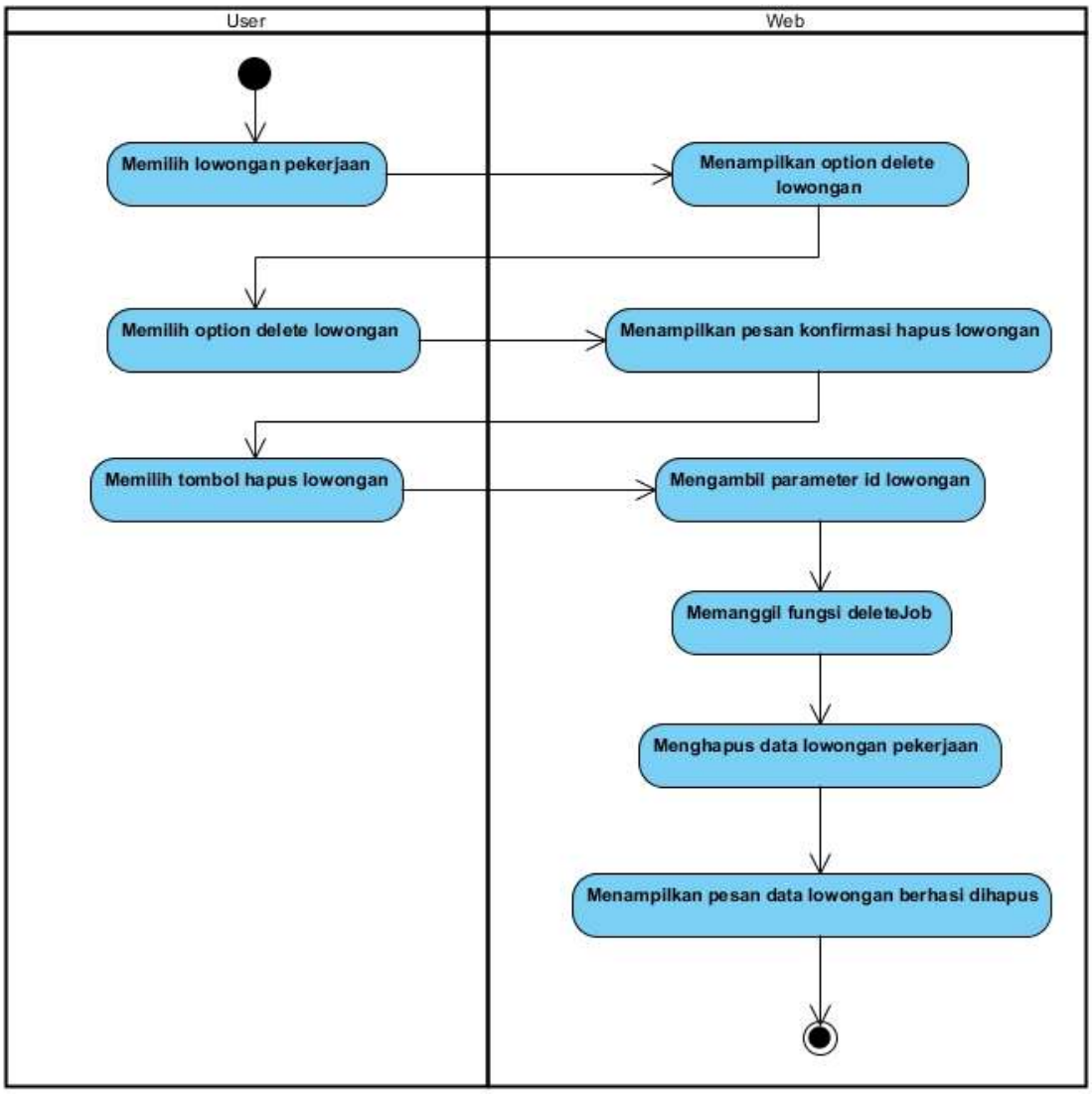
o. Activity diagram Penghapusan data lowongan

Aktivitas Penghapusan data lowongan merupakan aktivitas yang dilakukan *user* pemberi kerja menghapus data lowonga pekerjaan yang pernah dibuat oleh

member pemberi kerja tersebut. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih lowongan pekerjaan, pada aktivitas ini *user* pemberi kerja memilih lowongan pekerjaan yang akan diedit.
2. Menampilkan *option delete* lowongan, pada aktivitas ini sistem akan menampilkan pilihan *delete* lowongan untuk melakukan hapus data lowongan pekerjaan.
3. Memilih *option delete* lowongan, pada aktivitas ini pemberi kerja memilih *option* untuk *delete* lowongan pekerjaan.
4. Menampilkan pesan konfirmasi hapus lowongan, pada aktivitas ini sistem menampilkan pesan konfirmasi untuk melanjutkan proses hapus lowongan pekerjaan.
5. Memilih tombol hapus lowongan, pada aktivitas ini pemberi kerja memilih pilihan “Ya” untuk melanjutkan proses hapus lowongan pekerjaan.
6. Memanggil fungsi deleteJob, pada aktivitas ini sistem akan memanggil fungsi deleteJob.
7. Menghapus data lowongan pekerjaan, pada aktivitas ini sistem akan menghapus data lowongan pekerjaan dari database.
8. Menampilkan pesan data lowongan berhasil dihapus, pada aktivitas ini sistem akan menampilkan pesan bahwa data lowongan telah berhasil dihapus.

Seluruh aktivitas Penghapusan data lowongan terlihat pada *activity diagram* gambar III.58.



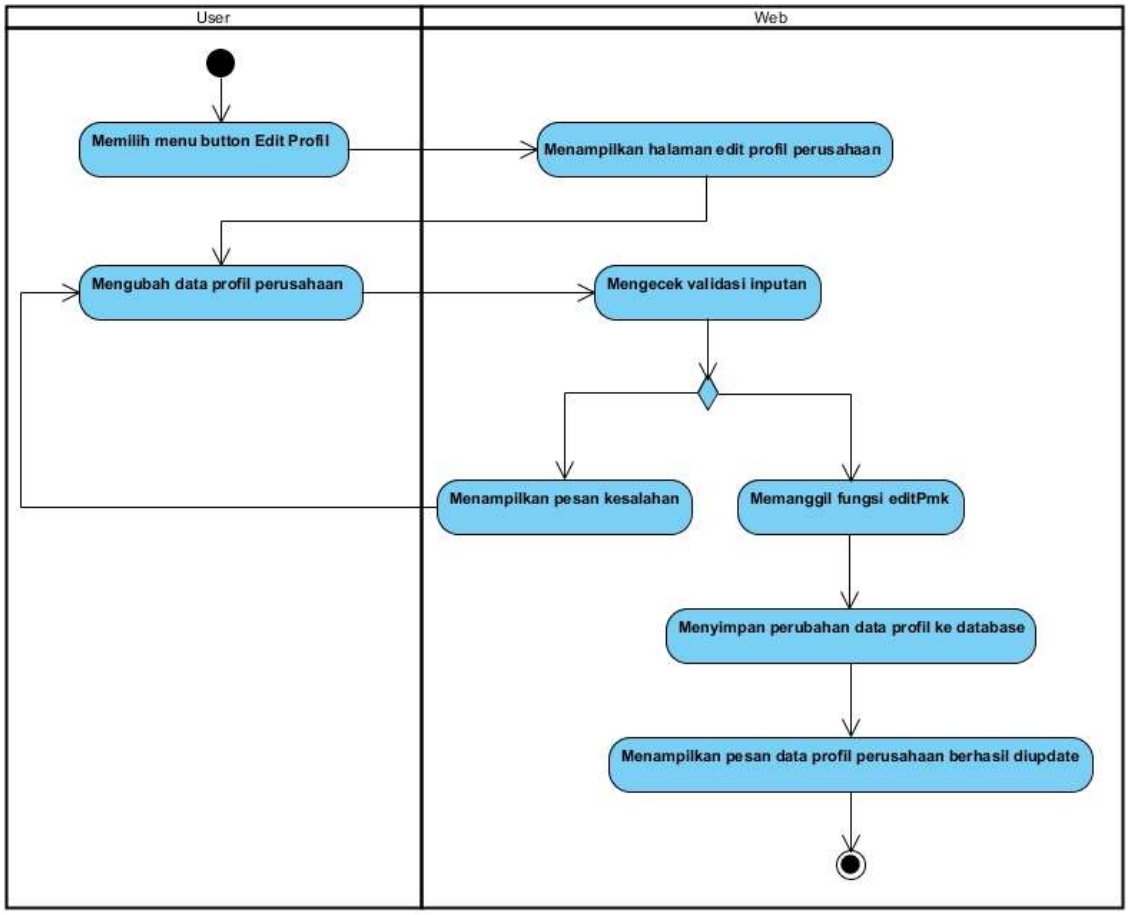
Gambar III. 58 Activity Diagram Penghapusan data lowongan

p. Activity diagram Perubahan data profil perusahaan

Aktivitas Perubahan data profil perusahaan merupakan aktivitas yang dilakukan *user* pemberi kerja untuk mengubah data profil perusahaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu Edit profil, pada aktivitas ini *user* menekan menu *button* Edit profil untuk melakukan perubahan data profil perusahaan.
2. Menampilkan halaman edit profil perusahaan, pada aktivitas ini sistem akan menampilkan halaman form edit profil perusahaan.
3. Mengubah data profil perusahaan, pada aktivitas ini user akan mengubah data profil perusahaan pada form profil perusahaan.
4. Mengecek validasi inputan, pada aktivitas ini sistem akan mengecek inputan data profil perusahaan sudah diisi atau belum dan sesuai format atau belum.
5. Menampilkan pesan kesalahan, aktivitas ini terjadi jika data yang diinputkan salah atau belum sesuai dengan format atau belum diisi.
6. Memanggil fungsi ediPmk, pada aktivitas ini sistem akan memanggil fungsi ediPmk.
7. Menyimpan perubahan data profil ke database, pada aktivitas ini sistem akan menyimpan perubahan data profil pribadi ke dalam database.
8. Menampilkan pesan data profil perusahaan telah berhasil diupdate, pada aktivitas ini sistema akan menampilkan pesan pemberitahuan bahwa data profil perusahaan telah berhasil diupdate atau diperbaharui.

Seluruh aktivitas Perubahan data profil perusahaan terlihat pada *activity diagram* gambar III.59.



Gambar III. 59 Activity Diagram Perubahan data profil perusahaan

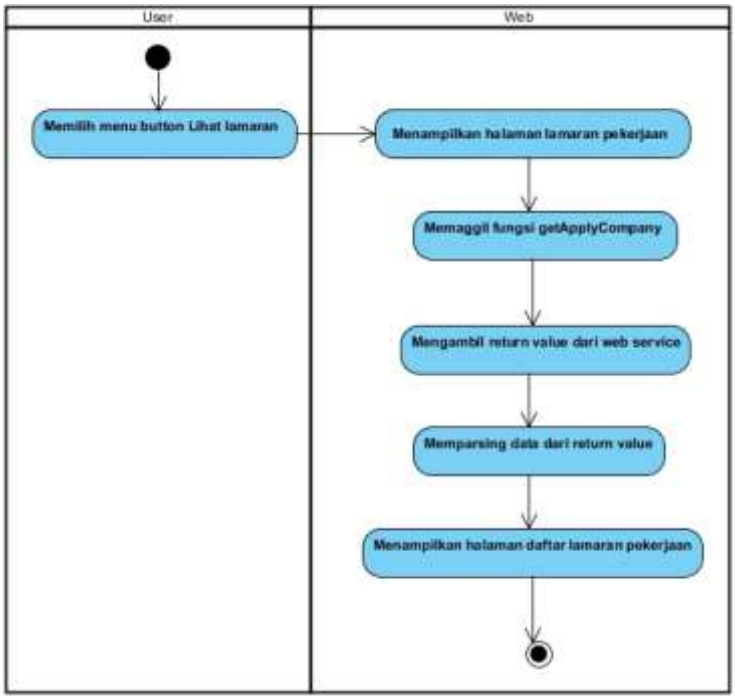
q. Activity diagram Lihat lamaran pekerjaan

Aktivitas Lihat lamaran pekerjaan merupakan aktivitas yang dilakukan *user* pemberi kerja melihat lamaran yang masuk ke perusahaannya berdasarkan lowongan yang telah dibuat oleh perusahaannya. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu button Lihat lamaran, pada aktivitas ini *user* menekan menu *button* Lamaran untuk melihat lamaran yang masuk ke perusahaannya.

2. Menampilkan halaman lamaran pekerjaan, pada aktivitas ini sistem akan menampilkan halaman lamaran pekerjaan.
3. Memanggil fungsi getApplyCompany, pada aktivitas ini sistem akan memanggil fungsi getApplyCompany.
4. Mengambil return value dari web service, pada aktivitas ini sistem mengambil nilai yang di kembalikan oleh web service berupa hasil dari eksekusi method pada web service.
5. Melakukan parsing data, pada aktivitas ini sistem memparsing respon dari web service agar data yang diterima dapat dibaca.
6. Menampilkan halaman daftar lamaran pekerjaan, pada aktivitas ini sistem akan menampilkan daftar lamaran pekerjaan.

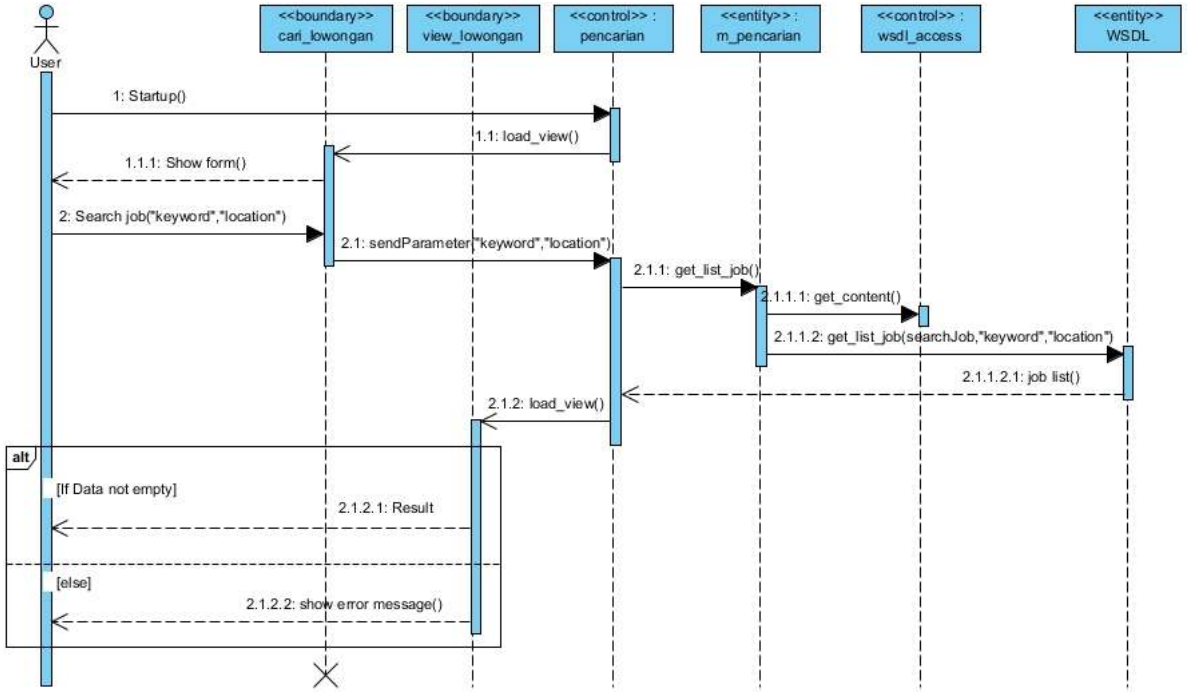
Seluruh aktivitas Lihat lamaran pekerjaan terlihat pada *activity diagram* gambar III.60.



Gambar III. 60 Activity Diagram Lihat lamaran pekerjaan

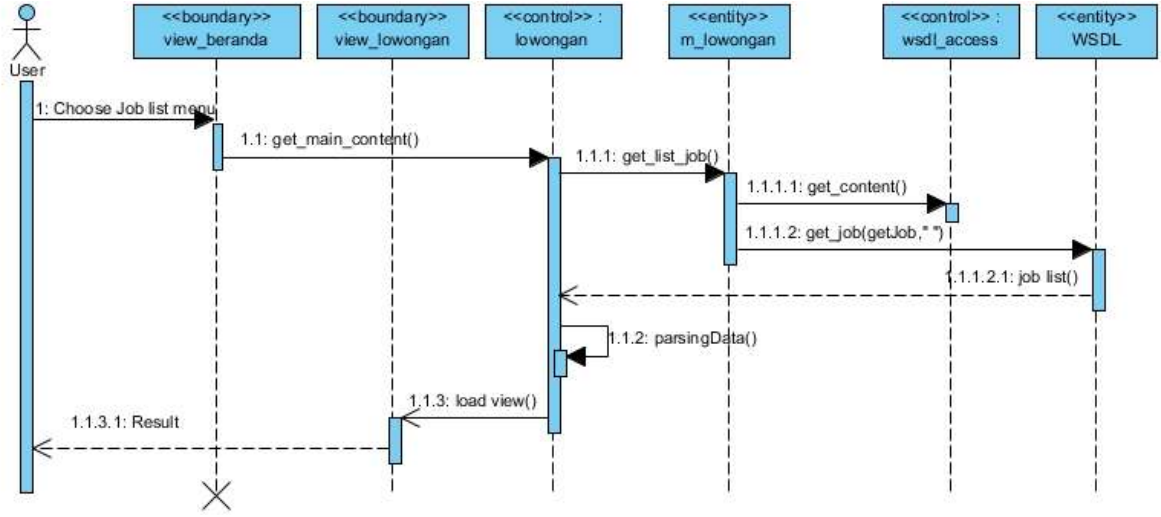
3. Sequence Diagram

a. Sequence diagram Pencarian lowongan



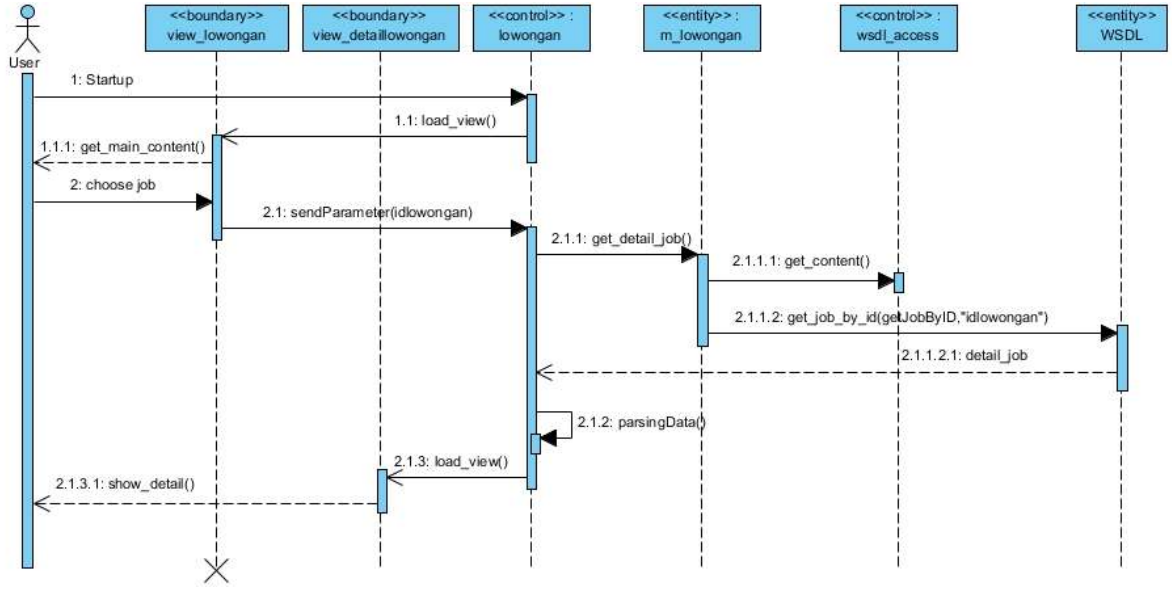
Gambar III. 61 Sequence Diagram Registrasi

b. Sequence diagram Lihat daftar lowongan



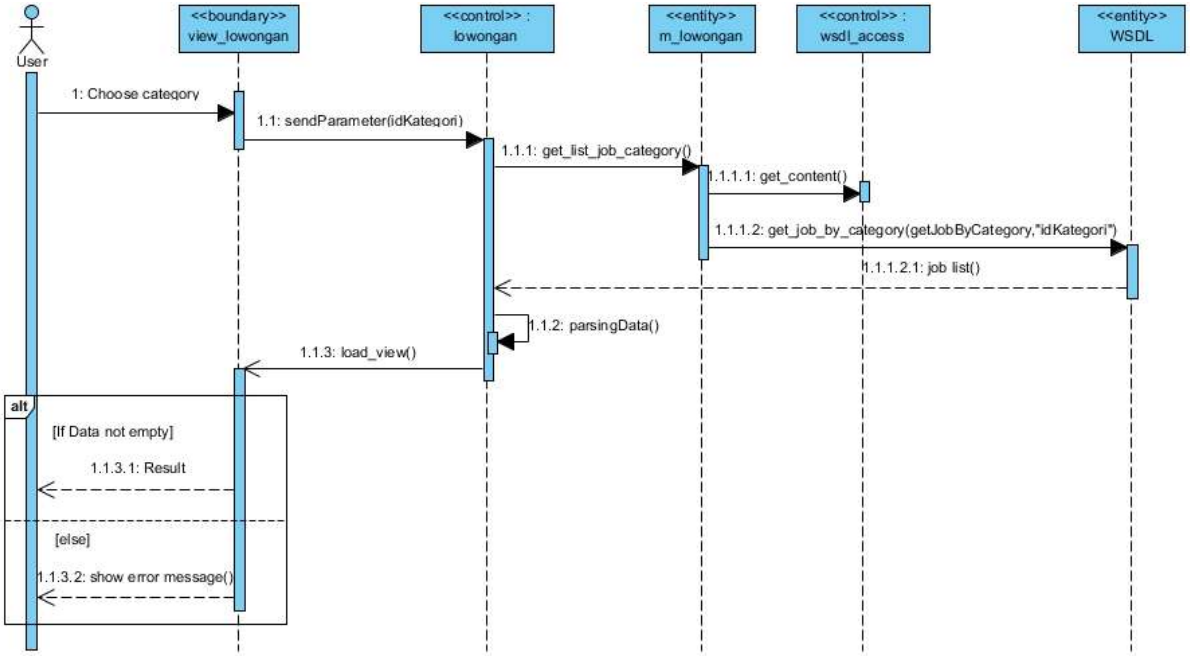
Gambar III. 62 Sequnce Diagram Lihat daftar lowongan

c. Sequence diagram Lihat detil lowongan



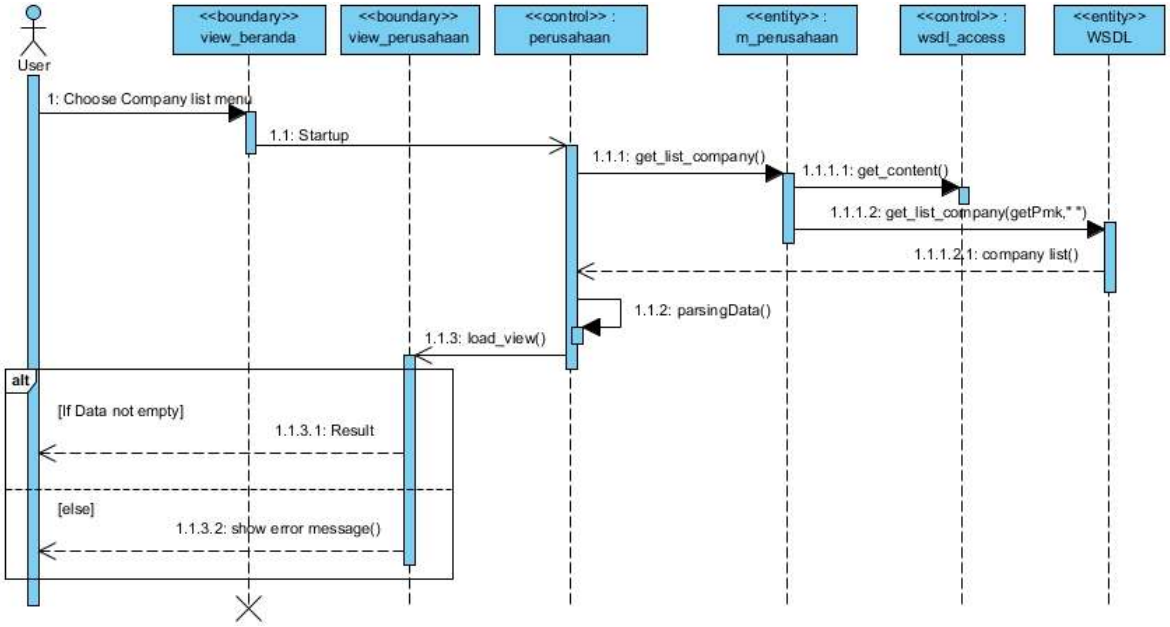
Gambar III. 63 Sequence Diagram Lihat detil lowongan

d. Sequence diagram Lihat lowongan berdasarkan kategori



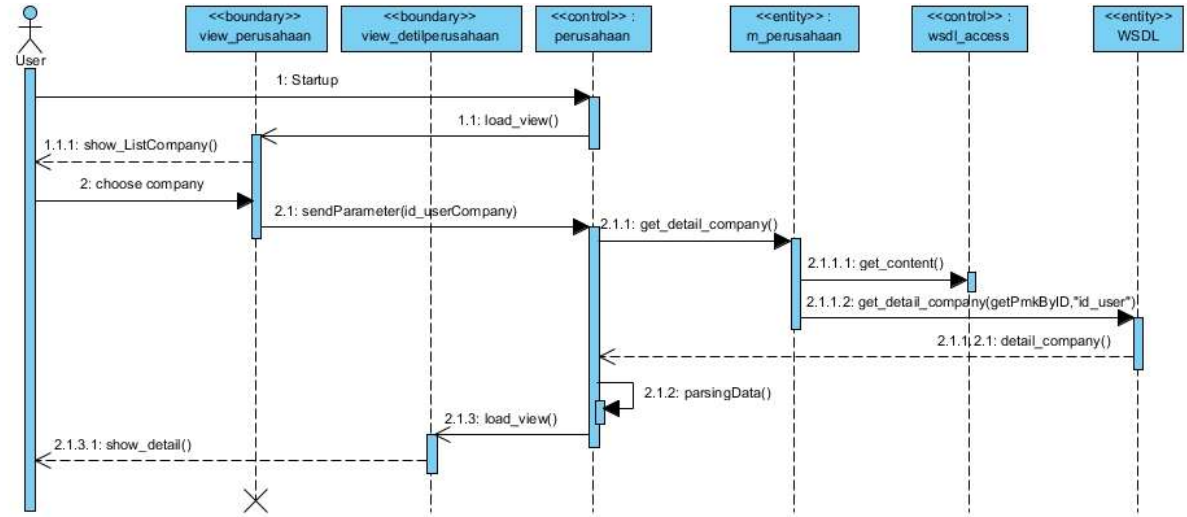
Gambar III. 64 Sequence Diagram Lihat lowongan berdasarkan kategori

e. Sequence diagram Lihat daftar perusahaan



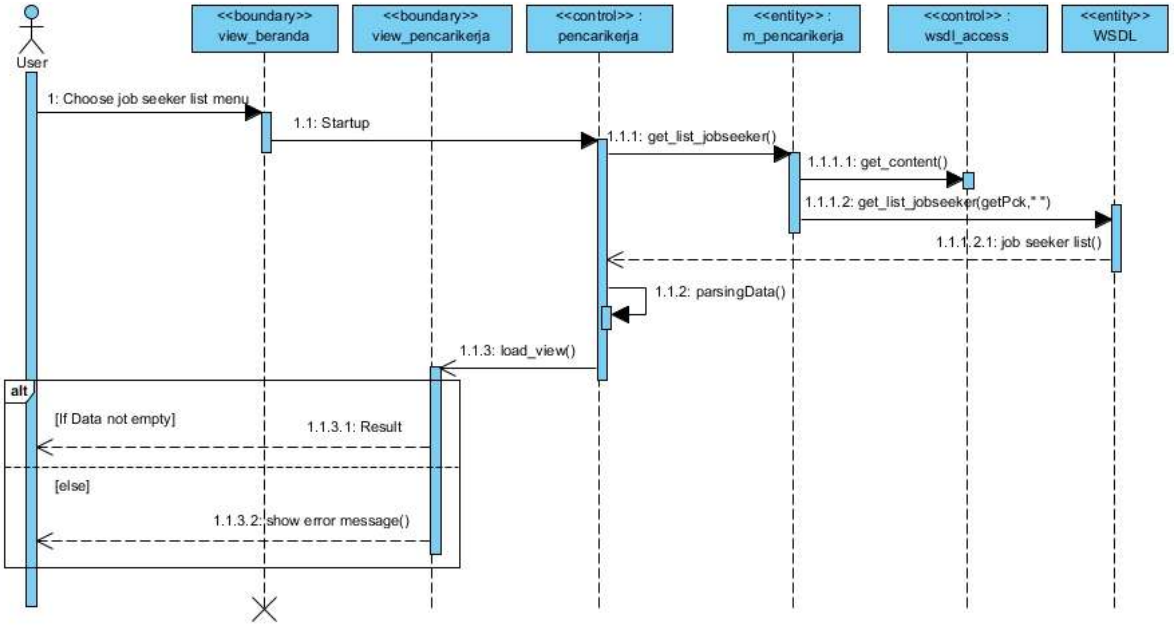
Gambar III. 65 Sequence Diagram Lihat daftar perusahaan

f. Sequence diagram Lihat detil perusahaan



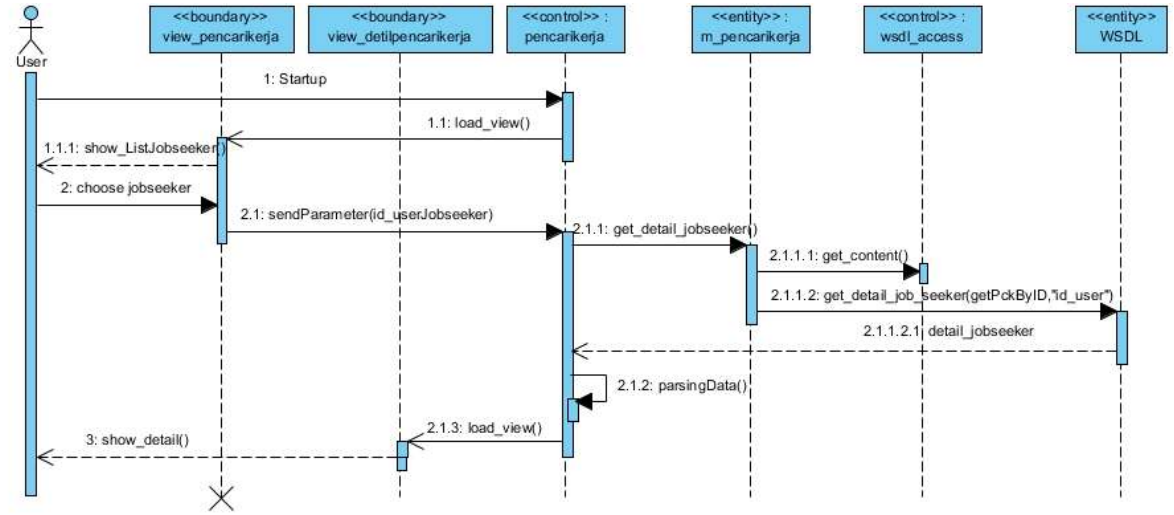
Gambar III. 66 Sequence Diagram Lihat detil perusahaan

g. Sequence diagram Lihat daftar pencari kerja



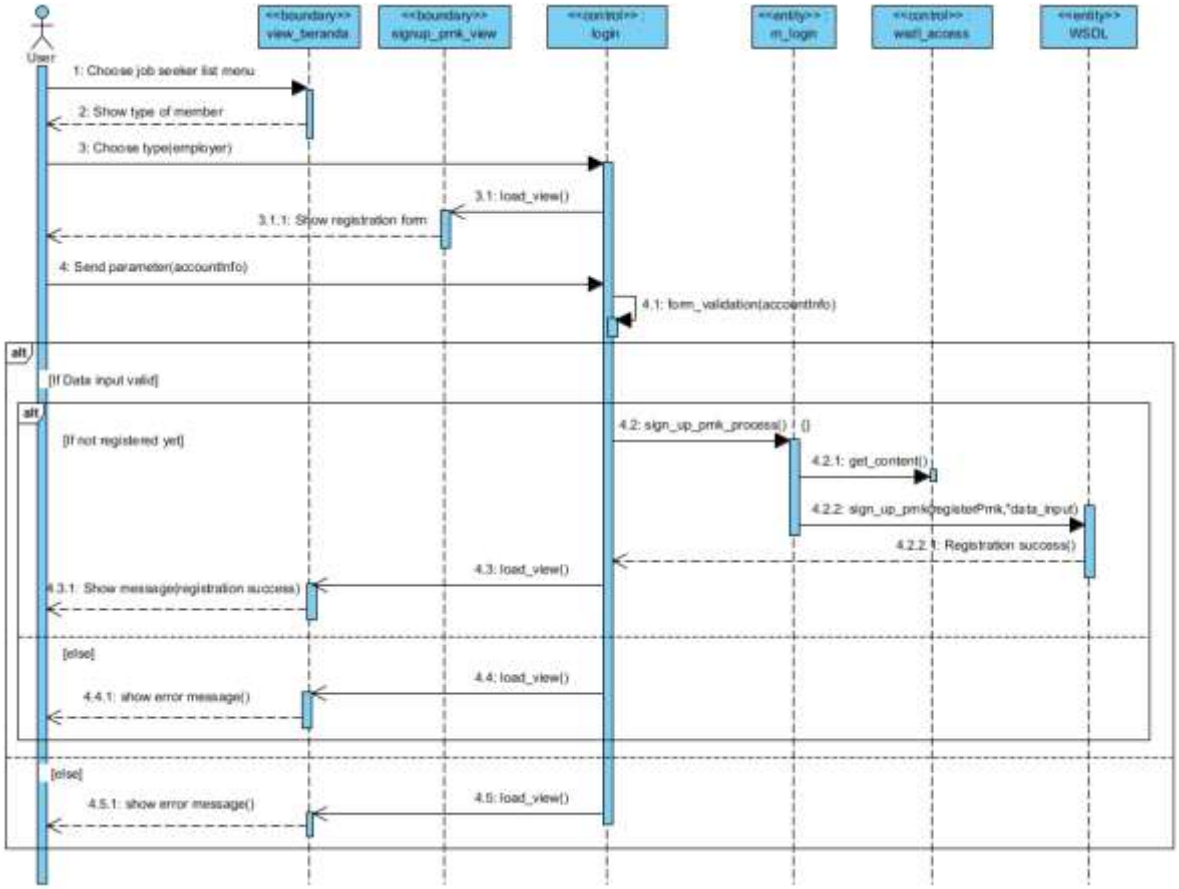
Gambar III. 67 Sequence Diagram Lihat daftar pencari kerja

h. Sequence diagram Lihat detil pencari kerja



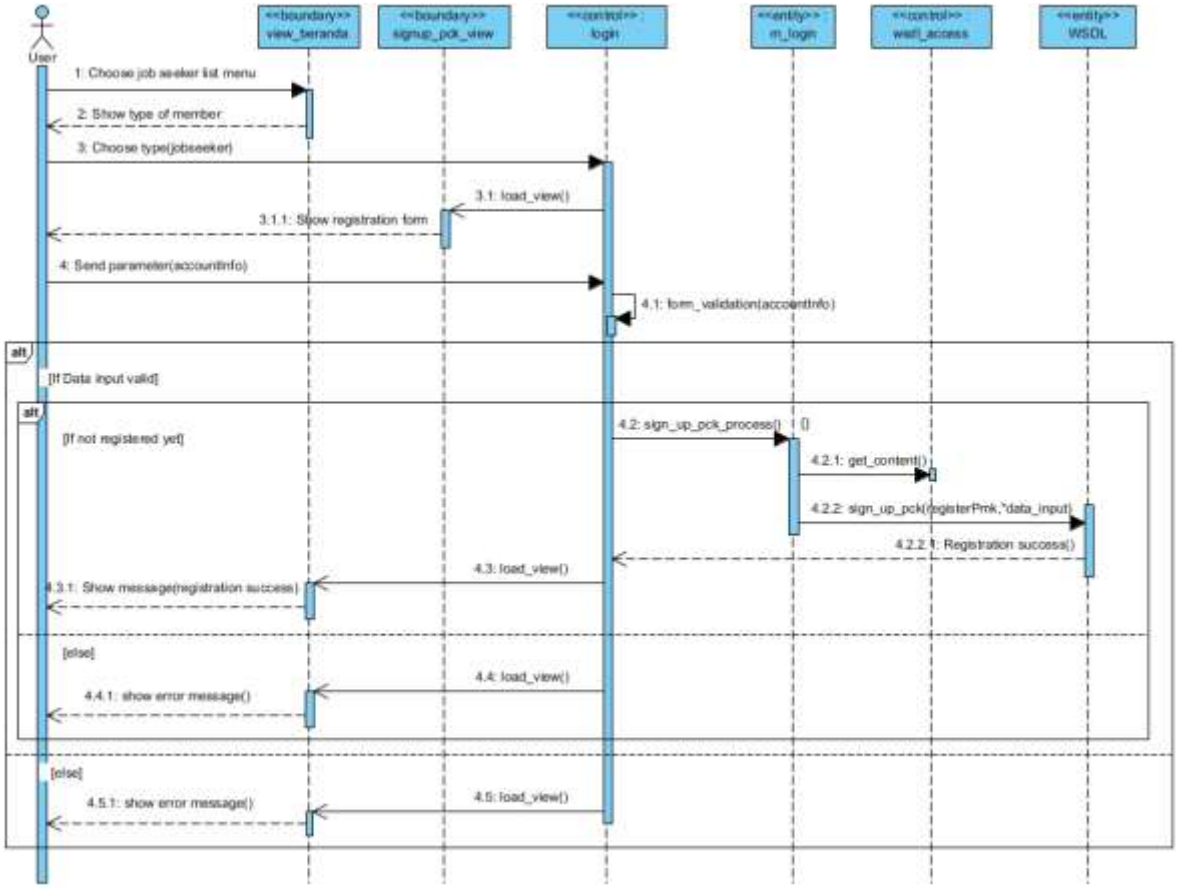
Gambar III. 68 Sequence Diagram Lihat detil pencari kerja

i. Sequence diagram Registrasi Pemberi Kerja



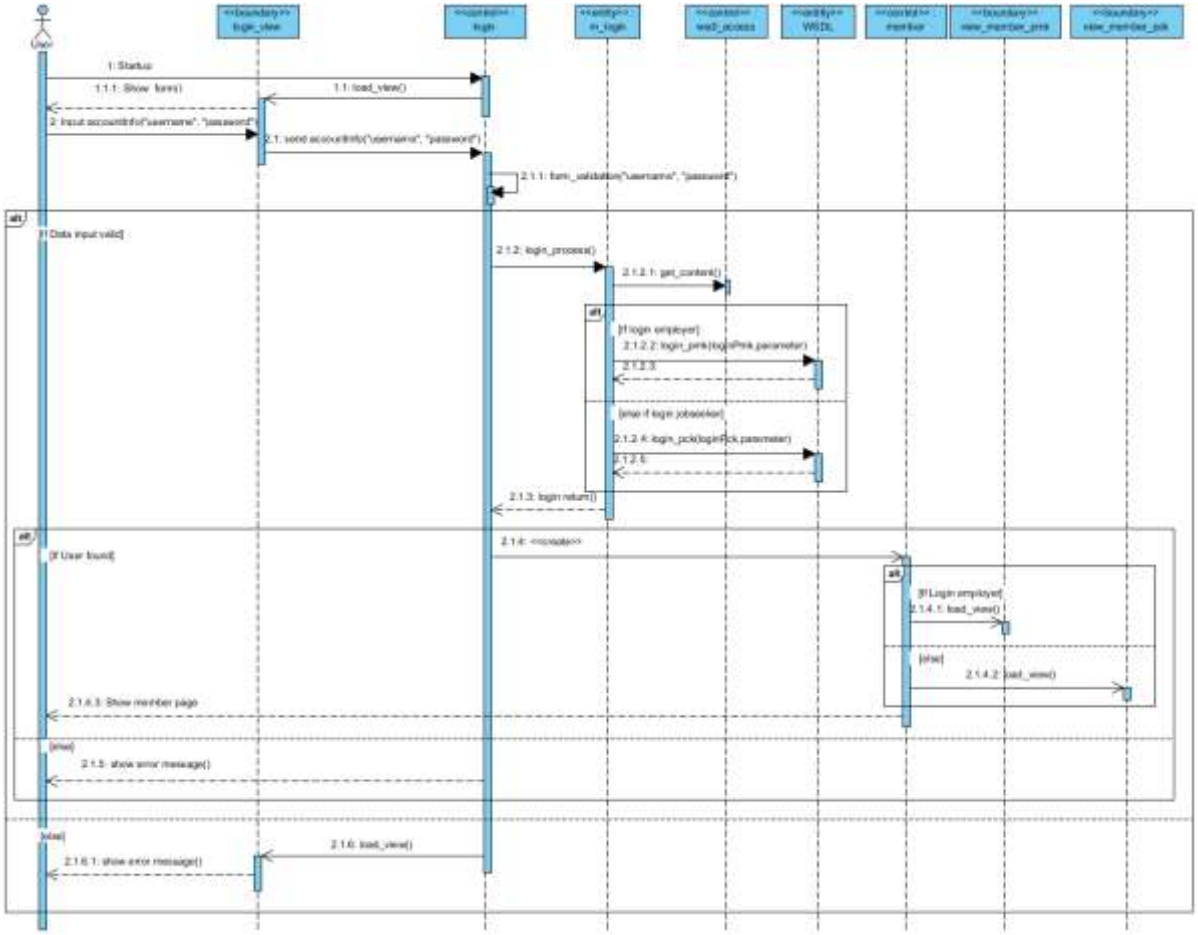
Gambar III. 69 Sequence Diagram Registrasi Pemberi Kerja

j. Sequence diagram Registrasi Pencari Kerja



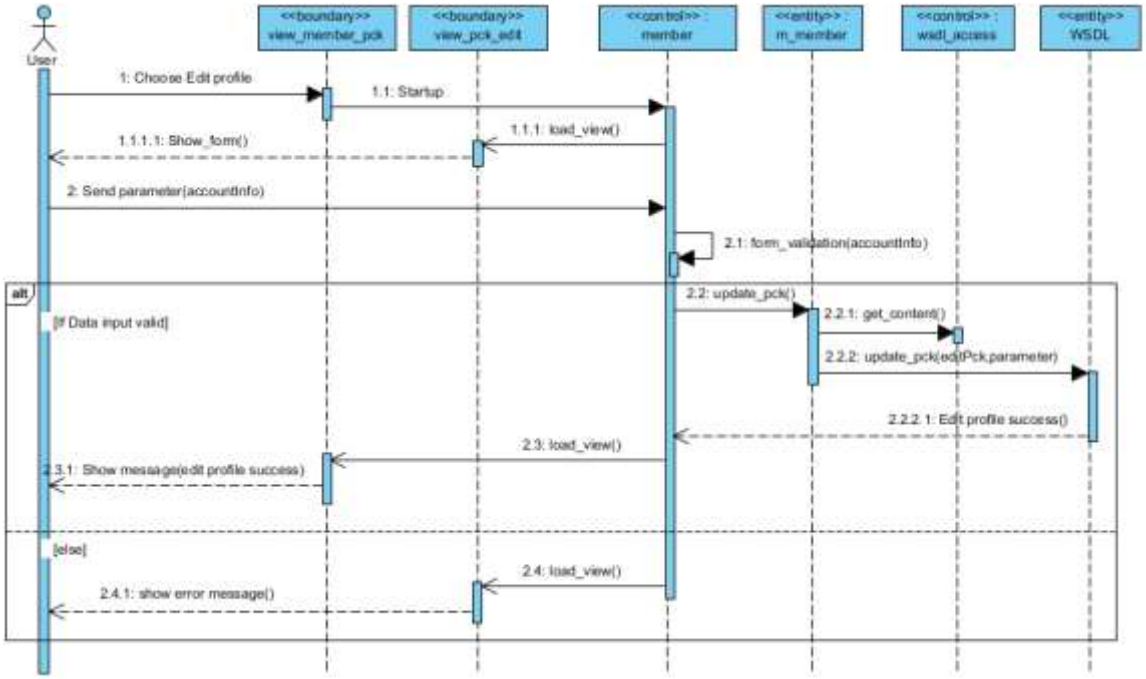
Gambar III. 70 Sequence Diagram Registrasi Pencari Kerja

k. Sequence diagram Login



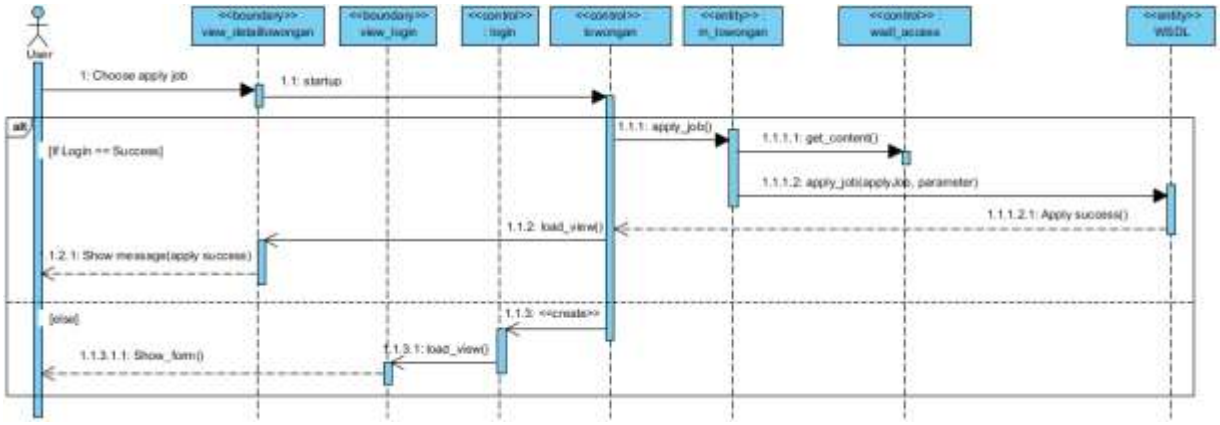
Gambar III. 71 Sequence Diagram Login

l. Sequence diagram Perubahan data profil pribadi



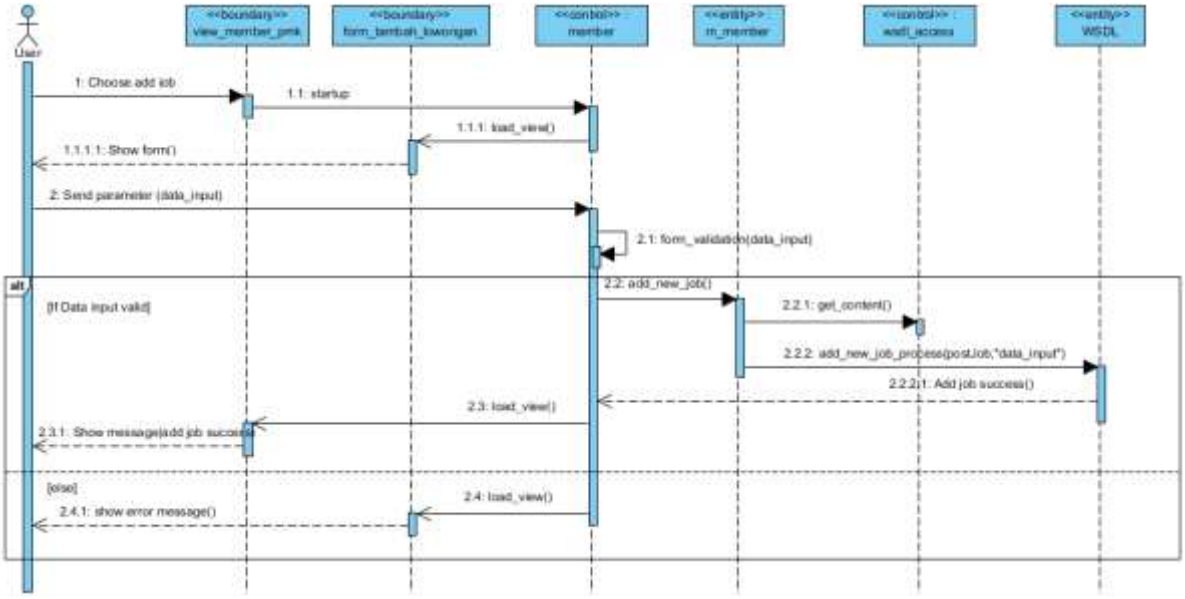
Gambar III. 72 Sequence Diagram Perubahan data profil pribadi

m. Sequence diagram Lamaran lowongan pekerjaan



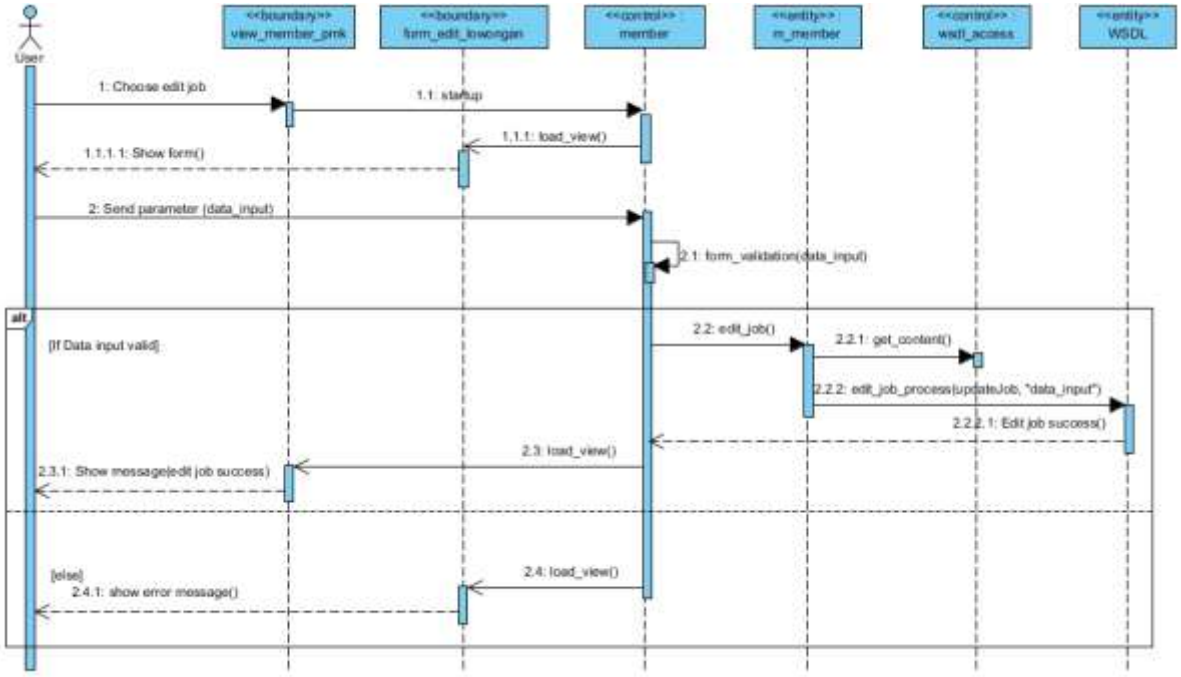
Gambar III. 73 Sequence Diagram Lamaran lowongan pekerjaan

n. Sequence diagram Pembuatan lowongan pekerjaan baru



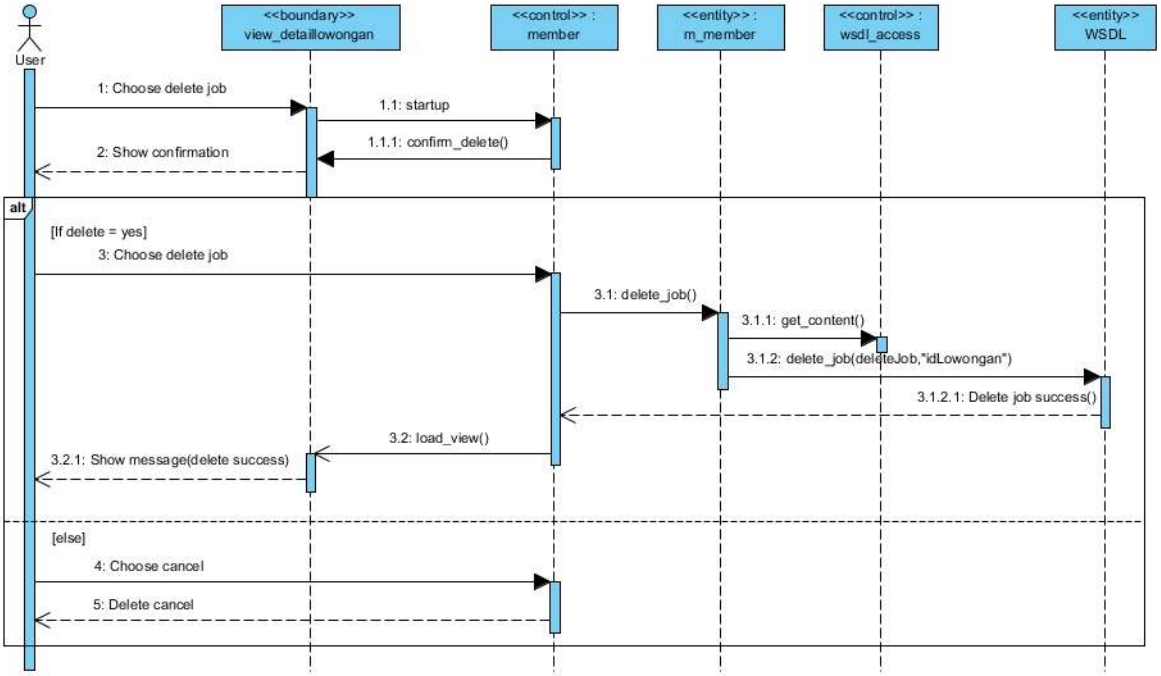
Gambar III. 74 Sequence Diagram Pembuatan lowongan pekerjaan baru

o. Sequence diagram Perubahan data lowongan



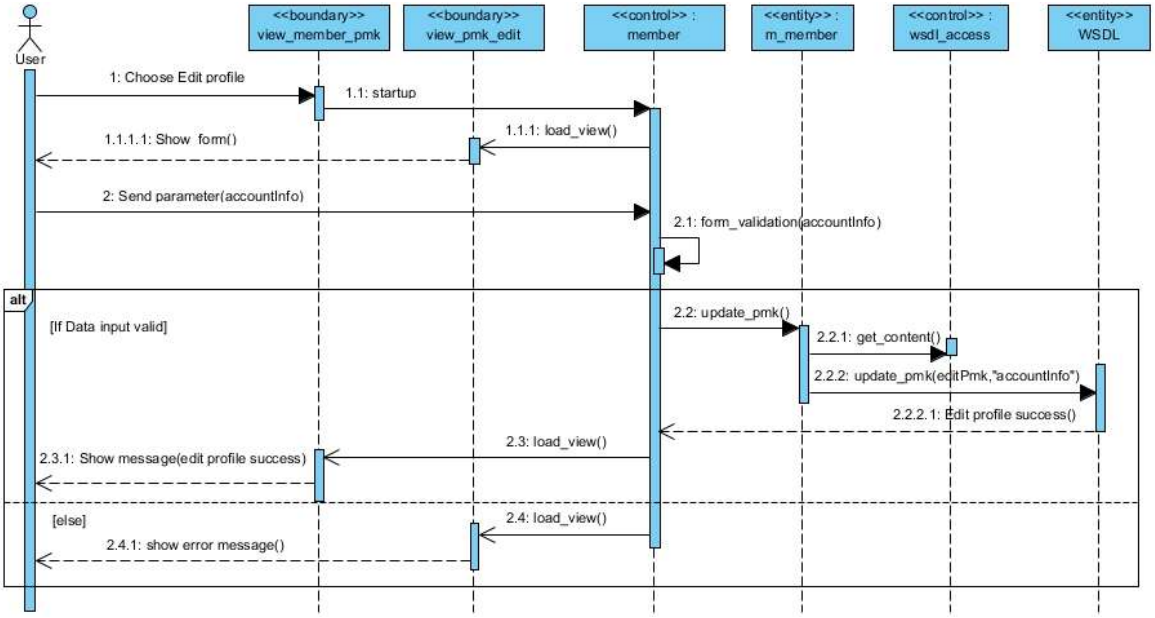
Gambar III. 75 Sequence Diagram Perubahan data lowongan

p. Sequence diagram Penghapusan data lowongan



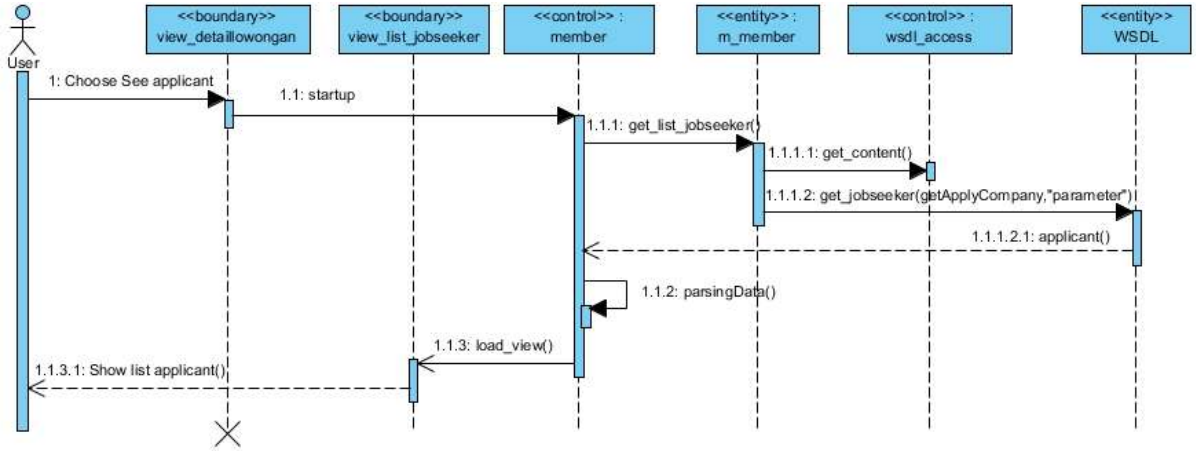
Gambar III. 76 Sequence Diagram Pengahapusan data lowongan

q. Sequence diagram Perubahan data profil perusahaan



Gambar III. 77 Sequence Diagram Perubahan data profil perusahaan

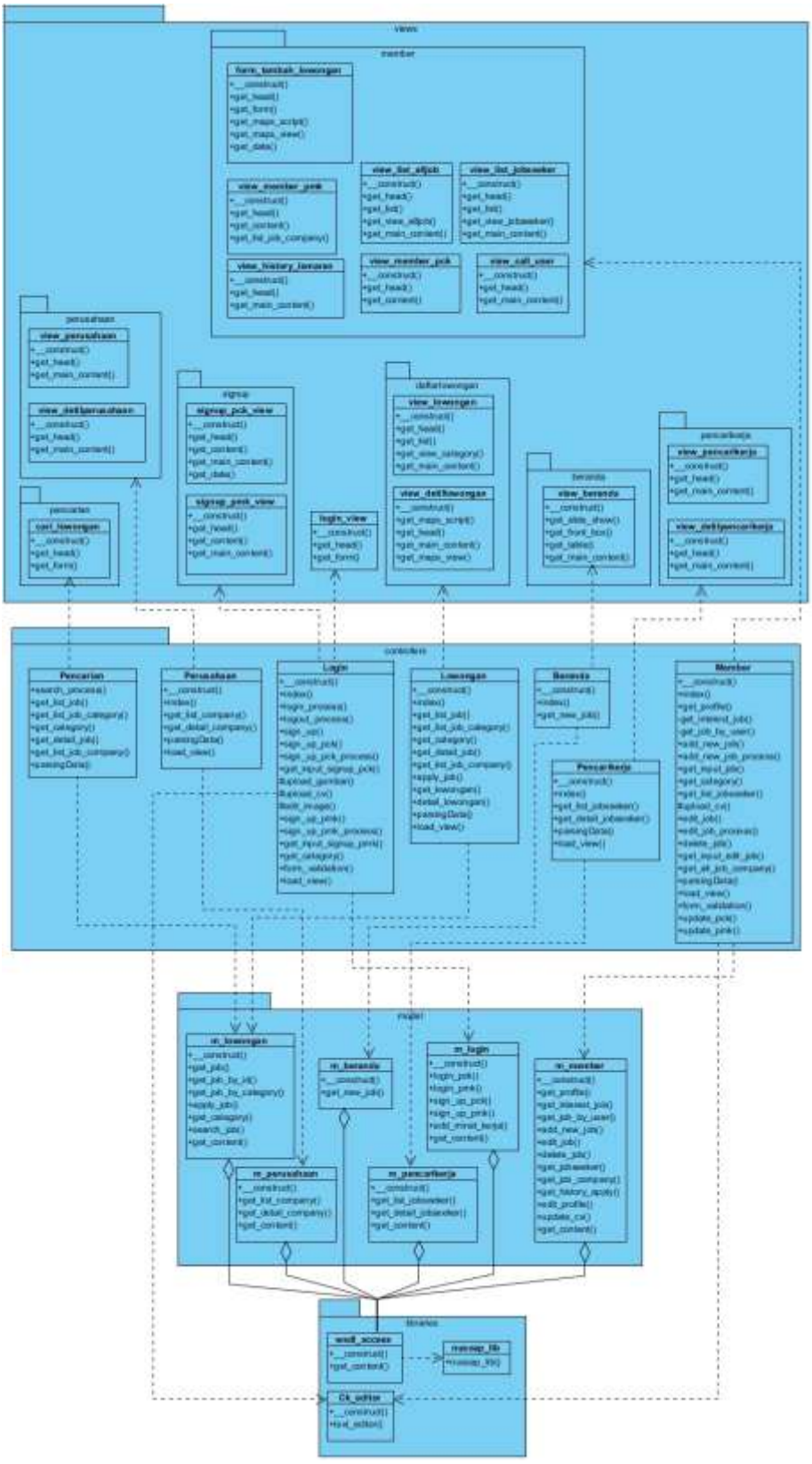
r. Sequence diagram Lihat lamaran pekerjaan



Gambar III. 78 Sequence Diagram Lihat lamaran pekerjaan

4. Class Diagram

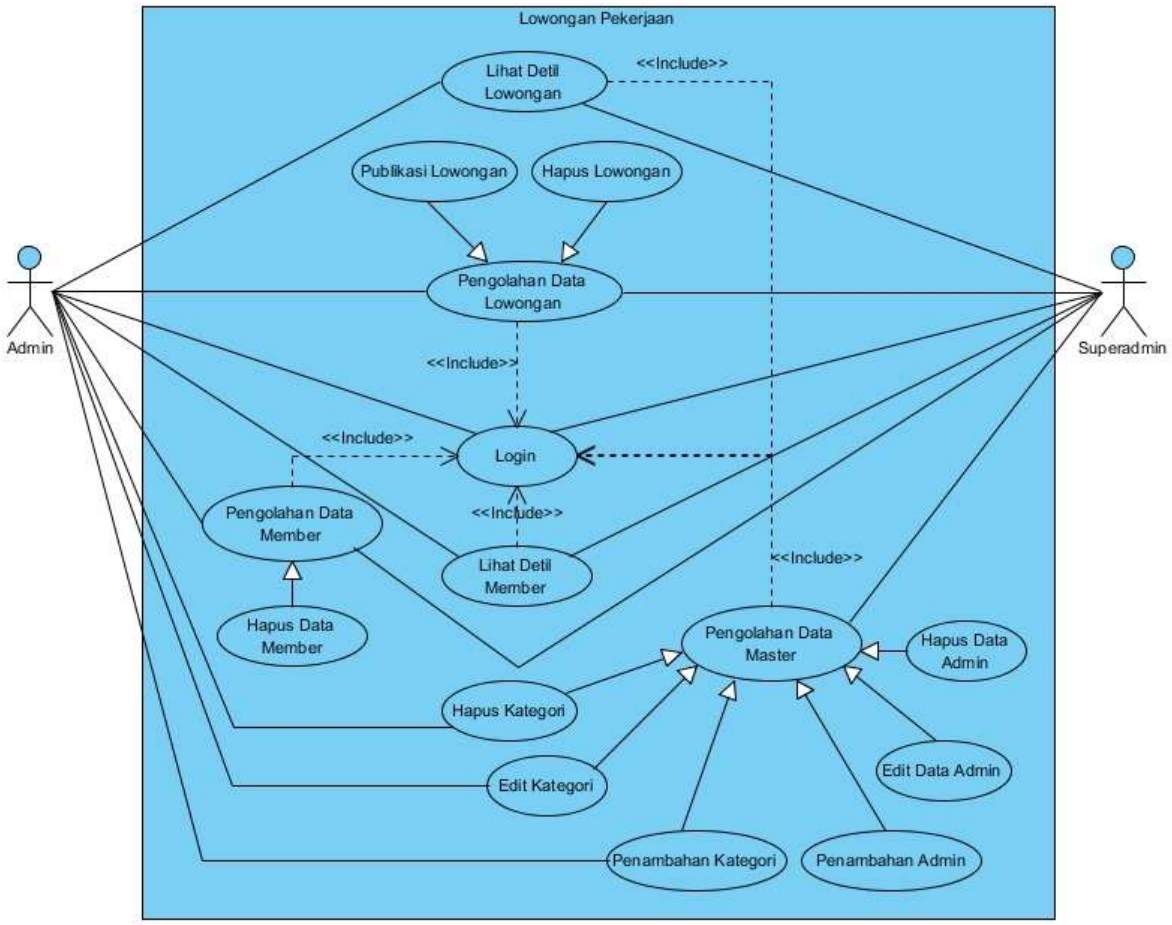
Class diagram menggambarkan struktur dan deskripsi class beserta hubungan satu sama lain seperti Asosiasi, Generalisasi dan lain sebagainya.



Gambar III. 79 Class Diagram aplikasi web

3.1.5.3 Analisis Pada Web Admin

1. **Use Case Diagram**



Gambar III. 80 Use case diagram web admin

a. Identifikasi Aktor

Pada tabel III.93 berisi daftar aktor dan deskripsi peran meraka masing-masing dalam sistem ini.

Tabel III. 93 Definisi aktor dalam web admin.

NO	Aktor	Deskripsi
C-01	Admin	Merupakan pengguna pada aplikasi web yang mempunyai hak akses untuk melakukan pengelolaan data lowongan, dan juga data member dan data master untuk penambahan kategori.
C-02	Superadmin	Merupakan pengguna pada aplikasi web yang mempunyai hak akses untuk melakukan pengelolaan data lowongan, dan juga data member dan data master untuk penambahan kategori dan penambahan data admin.

b. Identifikasi Use Case

Berdasarkan use case diagram di atas maka dibuatlah skenario dari tiap proses yang ada pada use case diagram tersebut. Pada tabel III.94 menggambarkan skenario pada use case *login*.

Tabel III. 94 Skenario Use case login.

Identifikasi	
Nomor	UC-WA-01
Nama Use Case	Login
Deskripsi	Proses autentikasi admin untuk masuk pada sistem
Aktor	Admin, Superadmin
Skenario Utama(UC-WA-01-S01)	
Kondisi awal	Koneksi ke database berhasil dan tampil halaman *login*
Aksi aktor	**Respon Sistem**
1. Membuka halaman utama	2. Menampilkan form login
3. Mengisi data *username* dan	4. Mencocokkan *username* dan *password*

password pada form login	yang ada pada database
	5. *Login* berhasil.
Skenario Alternatif (UC-WA-01-S02) Login namun terdapat pesan kesalahan	
	5a. *Login* gagal, sistem menampilkan pesan kesalahan. Misal karena *username* dan *password* tidak ada pada database atau *username* dan *password* tidak cocok.
Kondisi akhir	Halaman utama web admin tampil.

Pada tabel III.95 menggambarkan skenario untuk melakukan persetujuan untuk menerbitkan atau publikasi lowongan pekerjaan dari member pemberi kerja pada use case Publikasi Lowongan.

Tabel III. 95 Skenario Use case Publikasi Lowongan.

Identifikasi	
Nomor	UC-WA-02
Nama Use Case	Publikasi Lowongan
Deskripsi	Proses untuk melakukan persetujuan untuk menerbitkan atau publikasi lowongan pekerjaan dari member pemberi kerja
Aktor	Admin, Superadmin
Skenario Utama(UC-WA-02-S01)	
Kondisi awal	Login sebagai admin berhasil
Aksi aktor	**Respon Sistem**
1. Melakukan ceklist pada lowongan yang disetujui untuk dipublikasi	2. Mengambil Id lowongan yang dipublikasi
	3. Merubah status lowongan menjadi *publish*
	4. Menampilkan lowongan pekerjaan
Kondisi akhir	Lowongan pekerjaan dari member pemberi kerja berhasil dipublikasi

Pada tabel III.96 menggambarkan skenario untuk melihat detil lowongan pekerjaan yang dilakukan oleh admin pada use case Lihat Detil Lowongan.

Tabel III. 96 Skenario Use case Lihat Detil Lowongan.

Identifikasi	
Nomor	UC-WA-03
Nama Use Case	Lihat Detil Lowongan
Deskripsi	Proses untuk melihat detil dari lowongan pekerjaan
Aktor	Admin, Superadmin
Skenario Utama(UC-WA-03-S01)	
Kondisi awal	Login sebagai admin berhasil dan berada pada halaman utama web admin
Aksi aktor	**Respon Sistem**
1. Memilih menu tab data lowongan	2. Melakukan *select* lowongan pekerjaan ke *database*
	3. Menampilkan daftar lowongan pekerjaan
4. Memilih tombol detail	5. Melakukan *select* detil lowongan berdasarkan id lowongan yang dipilih
	6. Menampilkan detil lowongan pekerjaan
Skenario Alternatif (UC-WA-03-S02) Lihat Detil Lowongan namun terdapat pesan kesalahan	
	3a. Menampilkan pesan kesalahan, karena data lowongan masih kosong. 6a. Menampilkan pesan kesalahan, misal karena koneksi ke *database* gagal atau tidak ada koneksi internet
Kondisi akhir	Detil lowongan pekerjaan berhasil ditampilkan

Pada tabel III.97 menggambarkan skenario untuk menghapus data lowongan pekerjaan pada use case Hapus Data Lowongan.

Tabel III. 97 Skenario Use case Hapus Data Lowongan.

Identifikasi	
Nomor	UC-WA-04
Nama Use Case	Hapus Data Lowongan
Deskripsi	Proses untuk menghapus data lowongan pekerjaan
Aktor	Admin, Superadmin
Skenario Utama(UC-WA-04-S01)	
Kondisi awal	Login sebagai admin berhasil dan berada pada halaman utama web admin
Aksi aktor	**Respon Sistem**
1. Memilih menu tab data lowongan	2. Menampilkan daftar lowongan pekerjaan
3. Memilih hapus lowongan pekerjaan	4. Menampilkan pesan konfirmasi hapus lowongan
5. Memilih pilihan “Ya” untuk hapus lowongan	6. Menghapus lowongan yang dipilih dari *database*
Skenario Alternatif (UC-WA-04-S02) Hapus Data Lowongan jika memilih pilihan “Tidak” untuk hapus lowongan	
	6a. Proses hapus lowongan berakhir dan data lowongan tidak dihapus dari *database*
Kondisi akhir	Data lowongan pekerjaan berhasil dihapus dari *database*

Pada tabel III.98 menggambarkan skenario untuk melihat detil member lowongan pekerjaan pada use case Lihat Detil Member .

Tabel III. 98 Skenario Use Case Lihat Detil Member

Identifikasi	
Nomor	UC-WA-05
Nama Use Case	Lihat Detil Member
Deskripsi	Proses untuk melihat detil dari member lowongan pekerjaan
Aktor	Admin, Superadmin
Skenario Utama(UC-WA-05-S01)	
Kondisi awal	Login sebagai admin berhasil dan berada pada halaman utama web admin
Aksi aktor	**Respon Sistem**
1. Memilih menu tab data pemberi / pencari kerja	2. Melakukan *select* data member ke *database*
	3. Menampilkan daftar pencari kerja / pemberi kerja
4. Memilih tombol detail	5. Melakukan *select* detil member berdasarkan id member yang dipilih
	6. Menampilkan detil member yang dipilih
Skenario Alternatif (UC-WA-05-S02) Lihat Detil Member namun terdapat pesan kesalahan	
	6a. Menampilkan pesan kesalahan, misal karena koneksi ke *database* gagal atau tidak ada koneksi internet
Kondisi akhir	Detil member berhasil ditampilkan

Pada tabel III.99 menggambarkan skenario untuk menghapus data member lowongan pekerjaan pada use case Hapus Data Member.

Tabel III. 99 Skenario Use Case Hapus Data Member.

Identifikasi	
Nomor	UC-WA-06
Nama Use Case	Hapus Data Member
Deskripsi	Proses untuk menghapus data member lowongan pekerjaan
Aktor	Admin, Superadmin
Skenario Utama(UC-WA-06-S01)	
Kondisi awal	Login sebagai admin berhasil dan berada pada halaman utama web admin
Aksi aktor	**Respon Sistem**
1. Memilih menu tab data pemberi / pencari kerja	2. Menampilkan daftar pemberi / pencari kerja
3. Memilih hapus data member	4. Menampilkan pesan konfirmasi hapus member
5. Memilih pilihan “Ya” untuk hapus member	6. Menghapus member yang dipilih dari *database*
	7. Menampilkan pesan member berhasil dihapus
Skenario Alternatif (UC-WA-06-S02) Hapus Data Member jika memilih pilihan “Tidak” untuk hapus member	
	6a. Proses hapus member berakhir dan data member tidak dihapus dari *database*
Kondisi akhir	Data member berhasil dihapus dari *database*

Pada tabel III.100 menggambarkan skenario untuk pengolahan data master penambahan kategori lowongan pekerjaan pada use case Penambahan Kategori .

Tabel III. 100 Skenario Use Case Penambahan Kategori

Identifikasi	
Nomor	UC-WA-07
Nama Use Case	Penambahan Kategori
Deskripsi	Proses untuk pengolahan data master untuk menambah kategori lowongan pekerjaan
Aktor	Admin, Superadmin
Skenario Utama(UC-WA-07-S01)	
Kondisi awal	Login sebagai admin berhasil dan berada manu tab data master
Aksi aktor	**Respon Sistem**
1. Memilih submenu kategori	2. Menampilkan kategori lowongan pekerjaan
3. Memilih tambah data	4. Menampilkan form tambah kategori lowongan
5. Memasukkan data kategori baru	6. Mengecek validasi input kategori
	7. Menyimpan data kategori baru ke dalam *database*
	8. Menampilkan pesan kategori berhasil diinputkan
Skenario Alternatif (UC-WA-07-S02) Penambahan Kategori namun terdapat pesan kesalahan	
	8a. Menampilkan pesan kesalahan, karena field kategori masih kosong atau kategori sudah ada sebelumnya.
Kondisi akhir	Jumlah data kategori pada *database* bertambah

Pada tabel III.101 menggambarkan skenario untuk pengolahan data master edit kategori lowongan pekerjaan pada use case Edit Kategori .

Tabel III. 101 Skenario Use Case Edit Kategori

Identifikasi	
Nomor	UC-WA-08
Nama Use Case	Edit Kategori
Deskripsi	Proses untuk pengolahan data master untuk mengedit nama kategori lowongan pekerjaan
Aktor	Admin, Superadmin
Skenario Utama(UC-WA-08-S01)	
Kondisi awal	Login sebagai admin berhasil dan berada pada halaman utama web admin
Aksi aktor	**Respon Sistem**
1. Memilih submenu kategori	2. Menampilkan kategori lowongan pekerjaan
3. Memilih edit kategori	4. Menampilkan form edit kategori lowongan
5. Mengubah data kategori	6. Mengecek validasi input data kategori
	7. Menyimpan perubahan data kategori ke dalam *database*
	8. Menampilkan pesan kategori berhasil diinputkan
Skenario Alternatif (UC-WA-08-S02) Edit Kategori namun terdapat pesan kesalahan	
	8a. Menampilkan pesan kesalahan, karena field kategori masih kosong atau kategori sudah ada sebelumnya.
Kondisi akhir	Perubahan data kategori berhasil dilakukan

Pada tabel III.102 menggambarkan skenario untuk menghapus data kategori lowongan pekerjaan pada use case Hapus Kategori.

Tabel III. 102 Skenario Use Case Hapus Kategori.

Identifikasi	
Nomor	UC-WA-09
Nama Use Case	Hapus Kategori
Deskripsi	Proses untuk menghapus data member lowongan pekerjaan
Aktor	Admin, Superadmin
Skenario Utama(UC-WA-09-S01)	
Kondisi awal	Login sebagai admin berhasil dan berada pada halaman utama web admin
Aksi aktor	**Respon Sistem**
1. Memilih submenu kategori	2. Menampilkan kategori lowongan pekerjaan
3. Memilih hapus kategori	4. Menampilkan pesan konfirmasi hapus member
5. Memilih pilihan “Ya” untuk hapus kategori	6. Menghapus kategori yang dipilih dari *database*
	7. Menampilkan pesan kategori berhasil dihapus
Skenario Alternatif (UC-WA-09-S02) Hapus Kategori jika memilih pilihan “Tidak” untuk hapus kategori	
	6a. Proses hapus kategori berakhir dan data member tidak dihapus dari *database*
Kondisi akhir	Data kategori lowongan pekerjaan berhasil dihapus dari *database*

Pada tabel III.103 menggambarkan skenario untuk pengolahan data master penambahan data admin lowongan pekerjaan pada use case Penambahan Admin

Tabel III. 103 Skenario Use Case Penambahan Admin

Identifikasi	
Nomor	UC-WA-10
Nama Use Case	Penambahan Admin
Deskripsi	Proses untuk pengolahan data master untuk menambah data admin lowongan pekerjaan
Aktor	Superadmin
Skenario Utama(UC-WA-10-S01)	
Kondisi awal	Login sebagai admin berhasil dan berada pada halaman utama web admin

Aksi aktor	**Respon Sistem**
1. Membuka halaman data master	2. Menampilkan halaman data master
3. Memilih submenu user	4. Menampilkan daftar admin lowongan pekerjaan
5. Memilih tambah data	6. Menampilkan form tambah admin lowongan pekerjaan
7. Memasukkan data admin baru	8. Mengecek validasi input data admin
	9. Menyimpan data kategori baru ke dalam *database*
	10. Menampilkan pesan data admin berhasil diinputkan
Skenario Alternatif (UC-WA-10-S02) Penambahan Admin namun terdapat pesan kesalahan	
	8a. Menampilkan pesan kesalahan, karena data input kosong atau belum diisi.
Kondisi akhir	Jumlah data admin pada *database* bertambah

Pada tabel III.104 menggambarkan skenario untuk pengolahan data master edit data admin lowongan pekerjaan pada use case Edit Data Admin.

Tabel III. 104 Skenario Use Case Edit Data Admin

Identifikasi	
Nomor	UC-WA-11
Nama Use Case	Edit Data Admin
Deskripsi	Proses untuk pengolahan data master untuk edit data admin lowongan pekerjaan
Aktor	Superadmin
Skenario Utama(UC-WA-11-S01)	
Kondisi awal	Login sebagai admin berhasil dan berada pada halaman utama web admin
Aksi aktor	**Respon Sistem**
1. Membuka halaman data master	2. Menampilkan halaman data master
3. Memilih submenu user	4. Menampilkan daftar admin lowongan pekerjaan
5. Memilih edit admin	6. Menampilkan form edit admin lowongan pekerjaan
7. Memasukkan data admin baru	8. Mengecek validasi input data admin
	9. Menyimpan perubahan data kategori ke dalam *database*
Skenario Alternatif (UC-WA-11-S02) Edit Admin namun terdapat pesan kesalahan	
	8a. Menampilkan pesan kesalahan, karena data input kosong atau belum diisi.
Kondisi akhir	Perubahan data kategori berhasil dilakukan

Pada tabel III.105 menggambarkan skenario untuk menghapus data admin lowongan pekerjaan pada use case Hapus Admin.

Tabel III. 105 Skenario Use Case Hapus Admin.

Identifikasi	
Nomor	UC-WA-12
Nama Use Case	Hapus Admin
Deskripsi	Proses untuk menghapus data admin lowongan pekerjaan
Aktor	Superadmin
Skenario Utama(UC-WA-12-S01)	
Kondisi awal	Login sebagai admin berhasil dan berada pada halaman utama web admin
Aksi aktor	**Respon Sistem**
1. Membuka menu tab data master	2. Menampilkan halaman data master
3. Memilih submenu user	4. Menampilkan data admin lowongan pekerjaan
5. Memilih hapus admin	6. Menampilkan pesan konfirmasi hapus admin
7. Memilih pilihan “Ya” untuk hapus admin	8. Menghapus adminyang dipilih dari *database*
Skenario Alternatif (UC-WA-12-S02) Hapus Admin jika memilih pilihan “Tidak” untuk hapus admin	
	6a. Proses hapus kategori berakhir dan data admin tidak dihapus dari *database*
Kondisi akhir	Data admin lowongan pekerjaan berhasil dihapus dari *database*

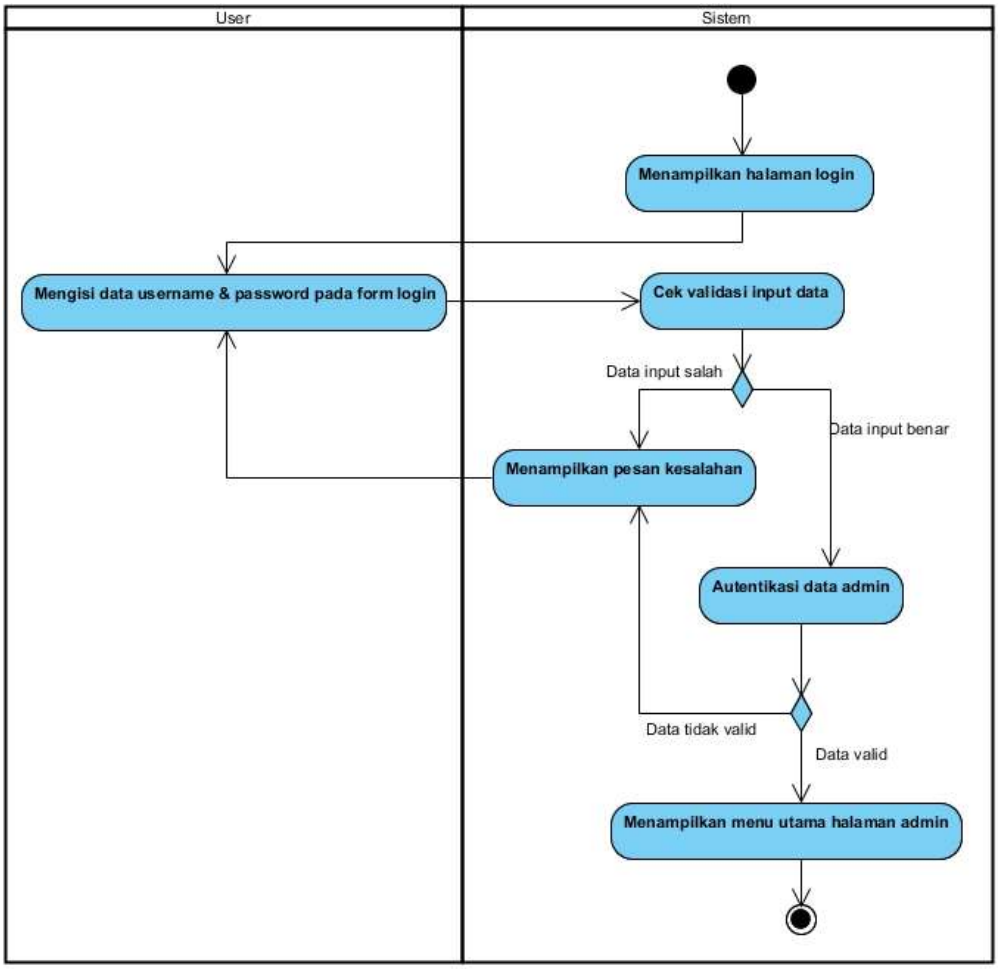
2. **Activity Diagram**

a. Activity Diagram Login

Aktivitas login merupakan aktivitas yang dilakukan admin untuk mandapatkan akses untuk mengelola lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Menampilkan halaman login, pada aktivitas ini sistem akan menampilkan halaman login ketika *user* admin ingin masuk ke halaman utama admin kelola lowongan pekerjaan .
2. Mengisi data *username* dan *password* pada form login, pada aktivitas ini admin memasukkan *username* dan *password*.
3. Cek validasi input data, pada aktivitas ini sistem akan memeriksa apakah *field username* & *password* sudah terisi atau belum.
4. Menampilkan pesan kesalahan, aktivitas ini terjadi jika *field username* dan *password* masih kosong.
5. Autentikasi data admin, pada aktivitas ini sistem mencocokan username dan *password* yang admin inputkan dengan *username* dan *password* yang ada pada database.
6. Menampilkan pesan kesalahan, aktivitas ini terjadi jika *username* dan *password* tidak ada dalam *database*.
7. Menampilkan menu utama halaman admin, aktivitas ini terjadi jika *username* dan *password* yang admin inputkan ada pada *database* lalu sistem menampilkan menu kelola lowongan pekerjaan.

Seluruh aktivitas login terlihat pada *activity diagram* gambar III.81.



Gambar III. 81 Activity Diagram proses Login

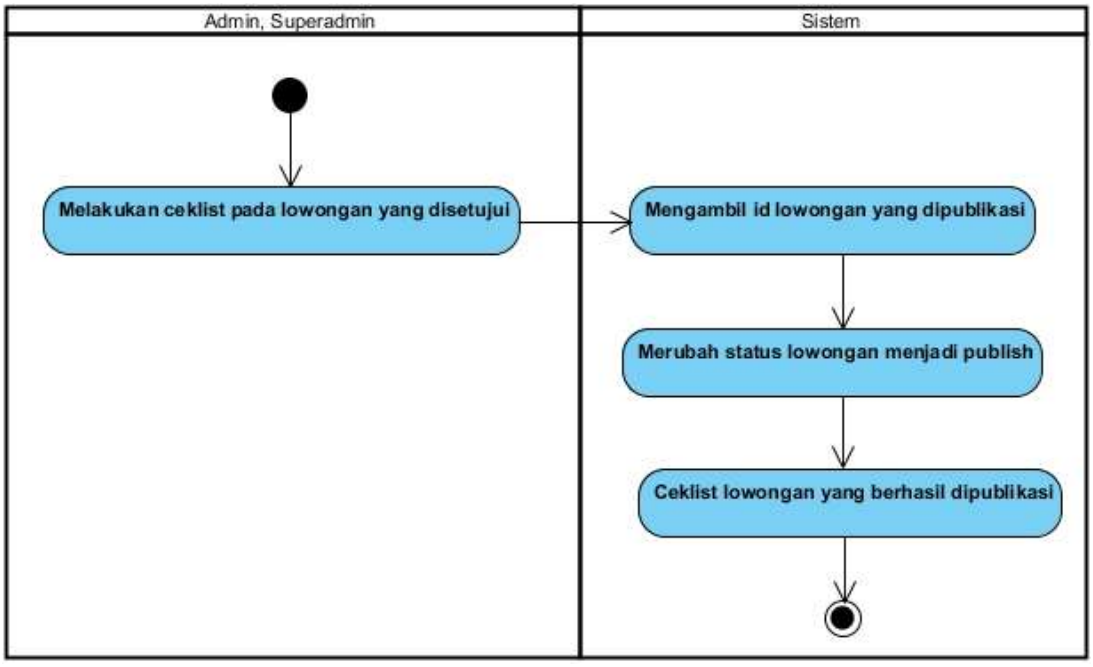
b. Activity Diagram Publikasi Lowongan

Aktivitas tambah lowongan pekerjaan merupakan aktivitas yang dilakukan oleh admin untuk menambahkan data lowongan baru. Dalam aktivitas ini terdapat beberapa tahapan aktivitas, yaitu :

1. Melakukan ceklist pada lowongan yang disetujui, dalam aktivitas ini admin melakukan ceklist pada *checkbox* lowongan pekerjaan yang disetujui untuk dipublikasi atau diterbitkan.

2. Mengambil id lowongan yang dipublikasi, aktivitas ini merupakan aktivitas yang dilakukan oleh sistem untuk mengambil id lowongan pekerjaan yang dipilih oleh admin untuk dipublikasi.
3. Mengubah status lowongan menjadi *publish*, pada aktivitas ini sistem akan merubah status lowongan yang sebelumnya belum disetujui untuk diterbitkan menjadi *publish* ketika lowongan berhasil diterbitkan.
4. Ceklist lowongan yang berhasil dipublikasi, pada aktivitas ini jika lowongan sudah berhasil diterbitkan atau dipublikasi maka sistem akan menceklist *checkbox* lowongan pekerjaan yang dipublikasi.

Seluruh aktivitas publikasi lowongan pekerjaan terlihat pada *activity diagram* gambar III.82.



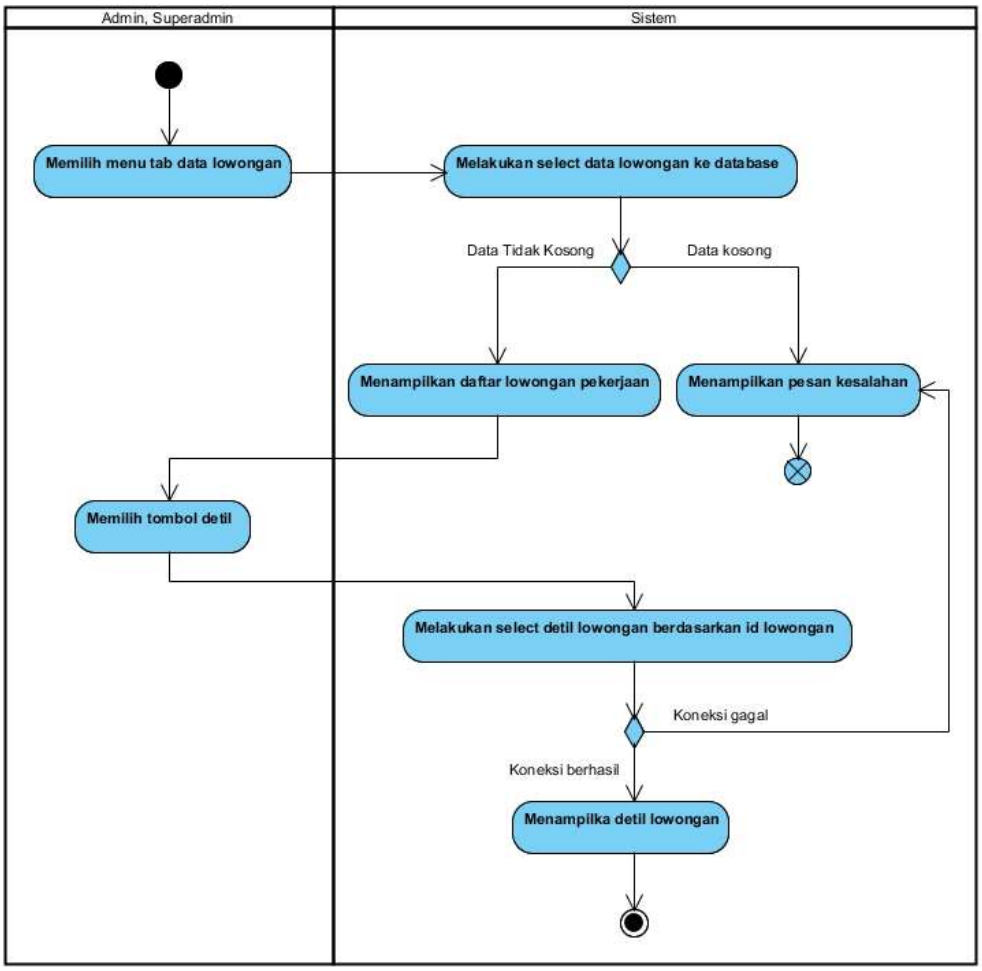
Gambar III. 82 Activity Diagram Publikasi Lowongan

c. Activity Diagram Lihat Detil Lowongan

Aktivitas edit lowongan merupakan aktivitas yang dilakukan oleh admin untuk mengedit lowongan pekerjaan yang sudah ada dalam database. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu tab data lowongan, pada aktivitas ini admin memilih menu tab lowongan untuk melihat daftar data lowongan pekerjaan.
2. Melakukan *select* data lowongan ke database, pada aktivitas ini sistem akan mengambil data lowongan pekerjaan yang terdapat pada *database*.
3. Menampilkan daftar lowongan pekerjaan, pada aktivitas ini jika data hasil proses *select* ke *database* tidak kosong maka sistem akan menampilkan daftar data lowongan pekerjaan.
4. Menampilkan pesan kesalahan, jika data hasil proses *select* ke *database* kosong maka sistem akan menampilkan pesan kesalahan bahwa data lowongan tidak ditemukan.
5. Memilih tombol detil, pada aktivitas ini admin memiilh tombol detil untuk melihat detil dari lowongan pekerjaan yang dipilih.
6. Melakukan *select* detil lowongan berdasarkan id lowongan, pada aktivitas ini sistem akan melakukan *select* data detil lowongan pekerjaan berdasarkan id lowongan ke *database*.
7. Menampilkan pesan kesalahan, aktivitas ini terjadi jika koneksi ke *database* gagal atau tidak ada koneksi internet.
8. Menampilkan detil lowongan, pada aktivitas ini sistem menampilkan detil lowongan pekerjaan yang dipilih oleh admin.

Seluruh aktivitas lihat detil lowongan pekerjaan terlihat pada *activity diagram* gambar III.83.



Gambar III. 83 Activity Diagram Lihat Detil Lowongan.

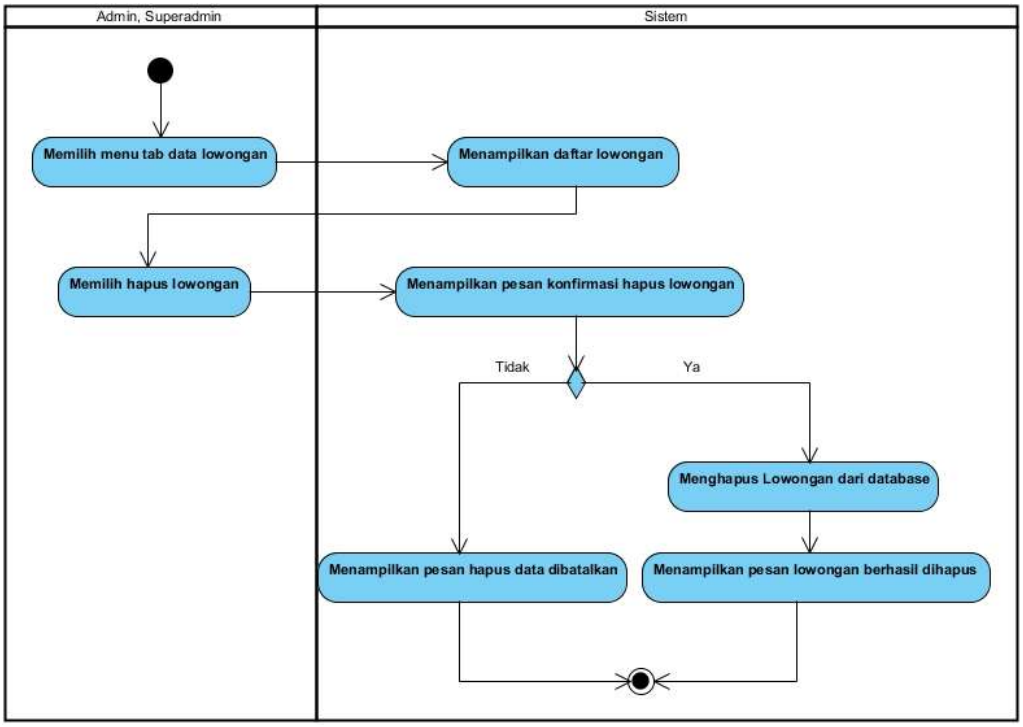
d. Activity Diagram Hapus Lowongan

Aktivitas hapus lowongan merupakan aktivitas yang dilakukan oleh admin untuk menghapus lowongan pekerjaan yang ada dalam database. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu tab data lowongan, pada aktivitas ini admin memilih menu tab lowongan untuk melihat daftar data lowongan pekerjaan.

2. Menampilkan daftar lowongan pekerjaan, pada aktivitas ini sistem akan menampilkan daftar data lowongan pekerjaan.
3. Memilih hapus lowongan, pada aktivitas ini admin memilih tombol hapus untuk melakukan hapus lowongan.
4. Menampilkan pesan konfirmasi hapus lowongan, pada aktivitas ini sistem akan menampilkan pesan konfirmasi terlebih dahulu sebelum admin melakukan proses hapus lowongan pekerjaan.
5. Menghapus lowongan dari *database*, jika admin memilih untuk tetap menghapus lowongan pekerjaan maka pada aktivitas ini sistem akan menghapus data lowongan pekerjaan dari database.
6. Menampilkan pesan lowongan berhasil dihapus, pada aktivitas ini sistem menampilkan pesan lowongan berhasil dihapus.
7. Menampilkan pesan hapus data dibatalkan, aktivitas ini terjadi jika admin membatalkan untuk menghapus lowongan pekerjaan.

Seluruh aktivitas hapus lowongan terlihat pada *activity diagram* gambar III.84.



Gambar III. 84 Activity Diagram Hapus Lowongan.

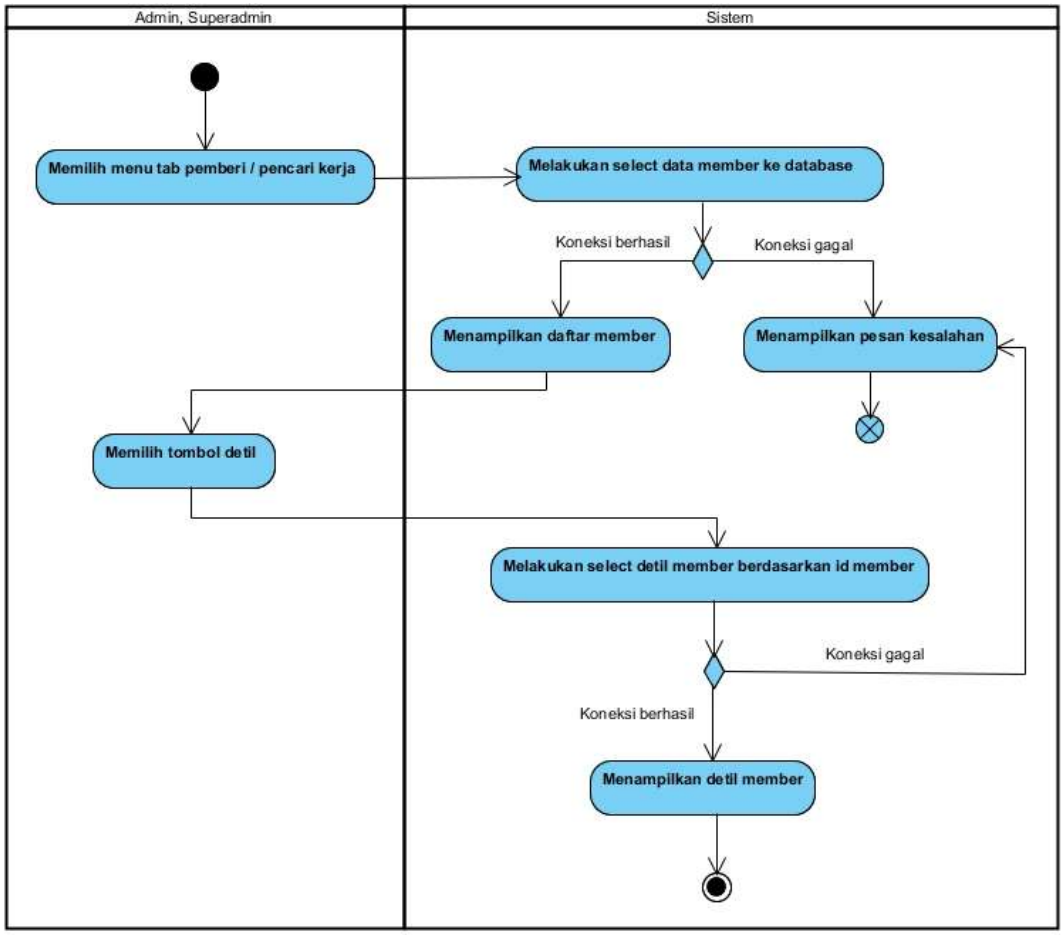
e. Activity Diagram Lihat Detil Member

Aktivitas lihat detil member merupakan aktivitas yang dilakukan oleh admin untuk melihat detil dari member pemberi atau pencari kerja yang terdaftar di sistem lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu tab pemberi / pencari kerja, pada aktivitas ini admin memilih menu tab pemberi / pencari kerja untuk melihat daftar pemberi / pencari kerja yang terdaftar di sistem lowongan pekerjaan.
2. Melakukan *select* data member ke *database*, pada aktivitas ini sistem akan mengambil daftar member yang terdapat dalam *database*.
3. Menampilkan data member, jika hasil *select* data ke *database* tidak kosong maka sistem akan menampilkan daftar member (pemberi / pencari kerja).

4. Menampilkan pesan kesalahan, jika hasil *select* data ke *database* kosong maka sistem akan menampilkan pesan bahwa member tidak ditemukan atau belum ada member (pemberi / pencari kerja)
5. Memilih tombol detil, pada aktivitas ini admin memiilh tombol detil untuk melihat detil dari member pemberi / pencari kerja.
6. Melakukan select detil member berdasarkan id member, pada aktivitas ini sistem akan mengambil data detil dari lowongan pekerjaan yang dipilih oleh admin.
7. Menampilkan detil member, jika koneksi ke *database* berhasil maka sistem akan menampilkan detil dari lowongan pekerajaan.
8. Menampilkan pesan kesalahan, jika koneksi ke *database* gagal maka sistem akan menampilkan pesan kesalahan.

Seluruh aktivitas lihat detil member terlihat pada *activity diagram* gambar III.85.



Gambar III. 85 Activity Diagram Lihat Detil Member

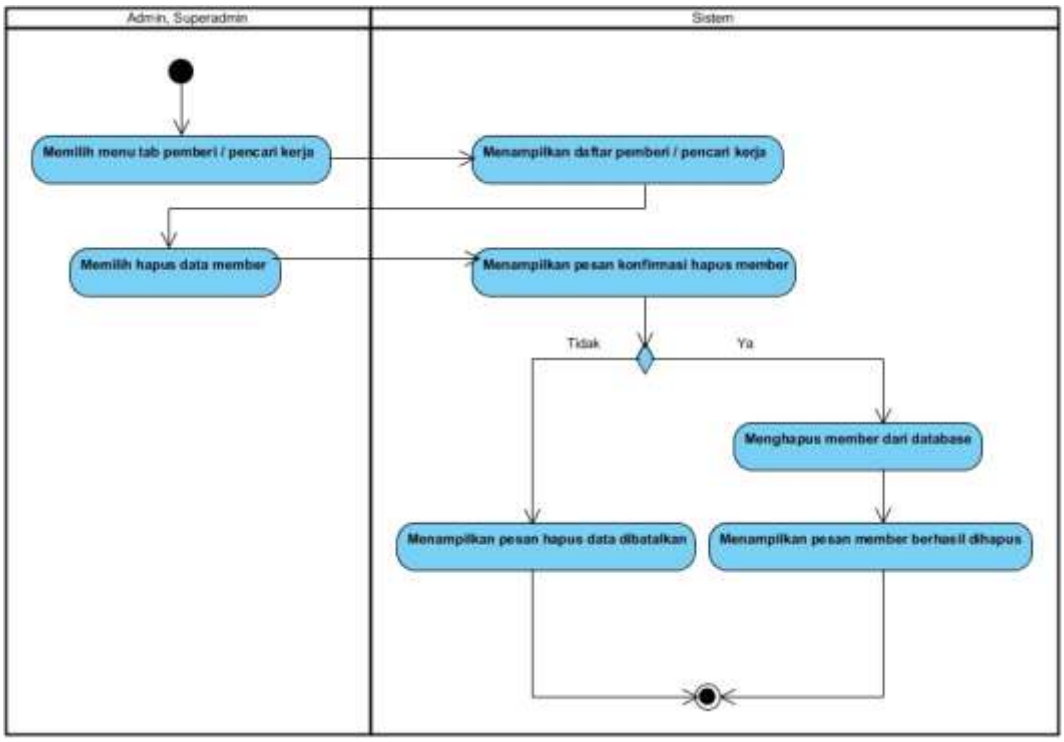
f. Activity Diagram Hapus Data Member

Aktivitas hapus data member merupakan aktivitas yang dilakukan oleh admin untuk menghapus data member pemberi / pencari kerja yang ada dalam database. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih menu tab pemberi / pencari kerja, pada aktivitas ini admin memilih menu tab pemberi / pencari kerja untuk melihat daftar pemberi / pencari kerja yang terdaftar di sistem lowongan pekerjaan.

2. Menampilkan daftar pemberi / pencari kerja, pada aktivitas ini sistem menampilkan daftar pemberi / pencari kerja yang terdaftar di sistem lowongan pekerjaan.
3. Memilih hapus data member, pada aktivitas ini admin memilih tombol hapus untuk melakukan hapus data member pemberi / pencari kerja.
4. Menampilkan pesan konfirmasi hapus member, pada aktivitas ini sistem akan menampilkan pesan konfirmasi terlebih dahulu sebelum admin melakukan proses hapus member pemberi / pencari kerja.
5. Menghapus member dari *database*, jika admin memilih untuk tetap menghapus data pemberi / pencari kerja maka pada aktivitas ini sistem akan menghapus data pemberi / pencari kerja dari *database.*
6. Menampilkan pesan member berhasil dihapus, pada aktivitas ini sistem menampilkan pesan data member berhasil dihapus.
7. Menampilkan pesan hapus data dibatalkan, aktivitas ini terjadi jika admin membatalkan untuk menghapus data pemberi / pencari kerja.

Seluruh aktivitas hapus data member terlihat pada activity diagram gambar III.86.



Gambar III. 86 Activity Diagram Hapus Data Member

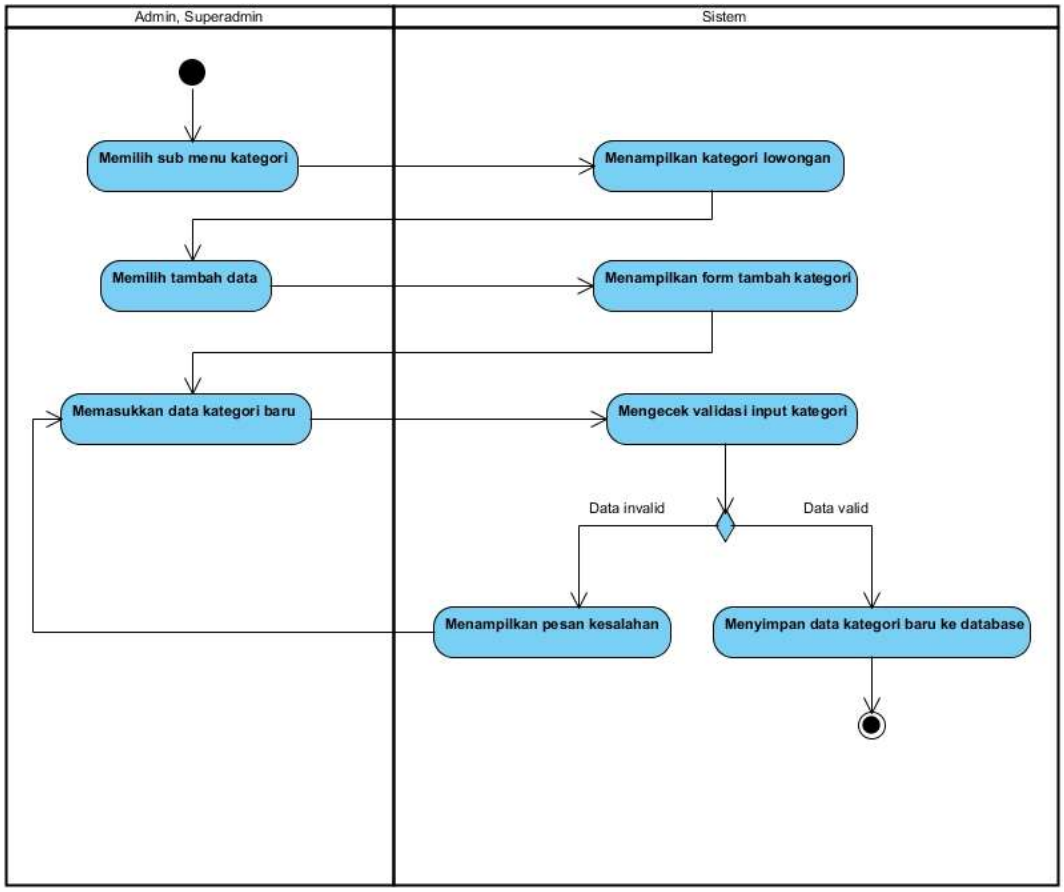
g. Activity Diagram Penambahan Kategori

Aktivitas hapus lowongan merupakan aktivitas yang dilakukan oleh admin untuk menghapus lowongan pekerjaan yang ada dalam database. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih sub menu kategori, pada aktivitas ini admin memilih sub menu kategori untuk menampilkan daftar kategori lowongan.
2. Menampilkan kategori lowongan, pada aktivitas ini sistem akan menampilkan daftar kategori lowongan pekerjaan.
3. Memilih tambah data, pada aktivitas ini admin memilih tombol tambah data untuk melakukan proses tambah data kategori.

4. Menampilkan form tambah kategori, pada aktivitas ini sistem akan menampilkan form untuk tambah data kategori lowongan pekerjaan.
5. Memasukkan data kategori baru, pada aktivitas ini admin memasukkan data kategori baru pada form tambah kategori.
6. Mengecek validasi input kategori, pada aktivitas ini sistem akan mengecek *field* pada form tambah data kategori apakah sudah valid atau belum.
7. Menampilkan pesan kesalahan, aktivitas ini terjadi jika data input pada form tambah data kategori tidak valid karena *field* kosong atau karena data kategori sudah ada.
8. Menyimpan data kategori baru ke *database*, pada aktivitas ini jika data kategori baru valid maka sistem akan menyimpan data kategori baru ke dalam *database*.

Seluruh aktivitas penambahan kategori terlihat pada activity diagram gambar III.87.



Gambar III. 87 Activity Diagram Penambahan Kategori

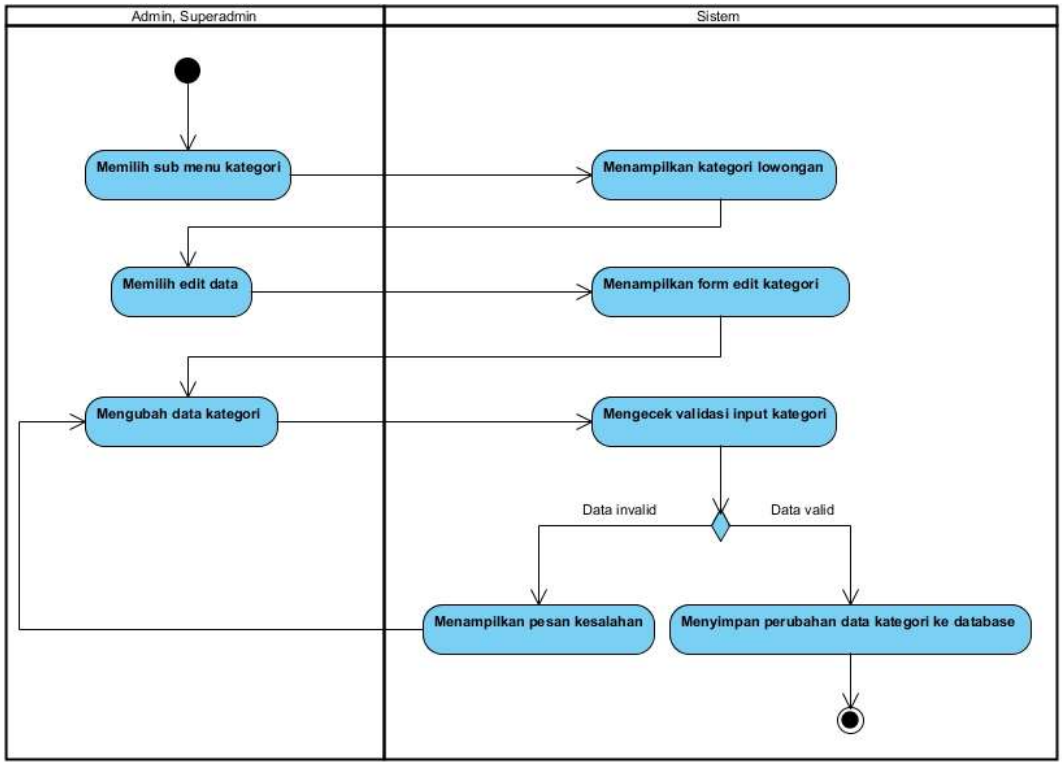
h. Activity Diagram Edit Kategori

Aktivitas edit kategori merupakan aktivitas yang dilakukan oleh admin untuk merubah data kategori lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih sub menu kategori, pada aktivitas ini admin memilih sub menu kategori untuk menampilkan daftar kategori lowongan.
2. Menampilkan kategori lowongan, pada aktivitas ini sistem menampilkan daftar kategori lowongan pekerjaan.

3. Memilih edit kategori, pada aktivitas ini admin memilih tombol edit untuk mengubah data kategori lowongan pekerjaan.
4. Menampilkan form edit kategori, pada aktivitas ini sistem akan menampilkan form edit kategori lowongan pekerjaan yang dipilih.
5. Mengubah data kategori, pada aktivitas ini admin mengubah data kategori yang dipilih.
6. Mengecek validasi input kategori, pada aktivitas ini sistem akan mengecek *field* pada form tambah data kategori apakah sudah valid atau belum.
7. Menampilkan pesan kesalahan, aktivitas ini terjadi jika data input pada form tambah data kategori tidak valid karena *field* kosong atau karena data kategori sudah ada.
8. Menyimpan perubahan data kategori ke *database*, pada aktivitas ini jika perubahan data kategori valid maka sistem akan menyimpan perubahan data kategori ke dalam *database.*

Seluruh aktivitas lihat edit kategori terlihat pada activity diagram gambar III.88.



Gambar III. 88 Activity Diagran Edit Kategori

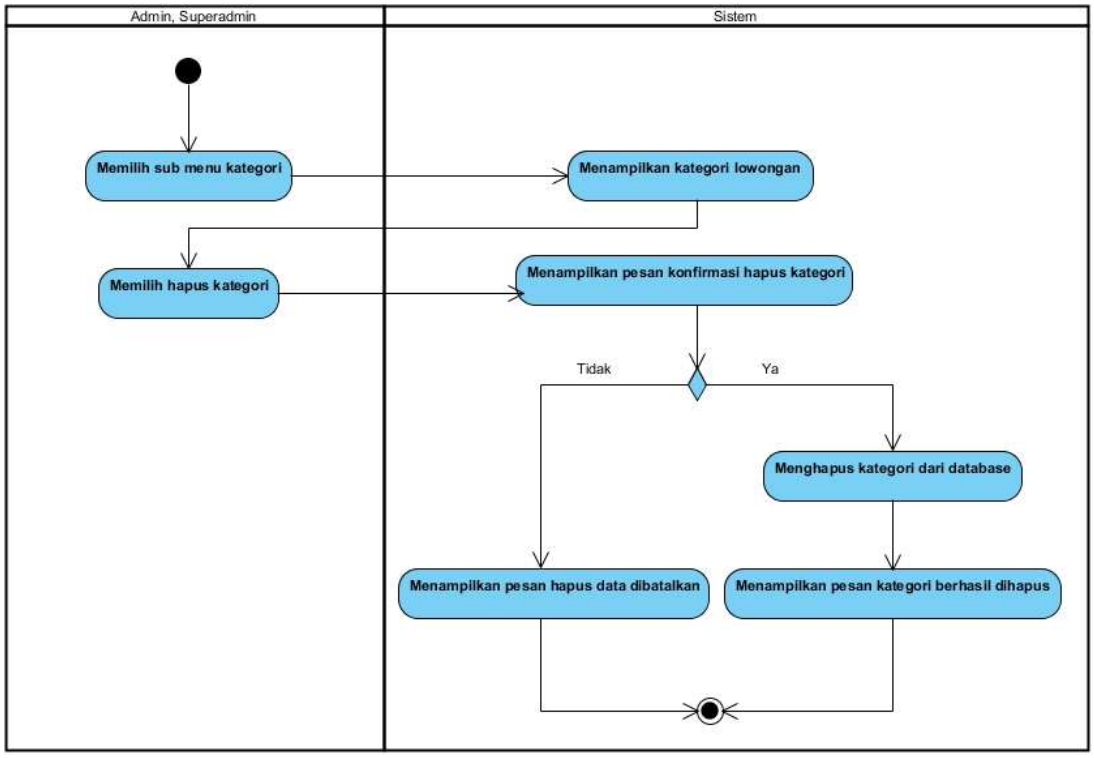
i. Activity Diagram Hapus Kategori

Aktivitas hapus lowongan merupakan aktivitas yang dilakukan oleh admin untuk menghapus lowongan pekerjaan yang ada dalam database. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih sub menu kategori, pada aktivitas ini admin memilih sub menu kategori untuk menampilkan daftar kategori lowongan.
2. Menampilkan kategori lowongan, pada aktivitas ini sistem menampilkan daftar kategori lowongan pekerjaan.
3. Memilih hapus kategori, pada aktivitas ini admin memilih tombol hapus untuk menghapus data kategori lowongan pekerjaan.

4. Menampilkan pesan konfirmasi hapus kategori, pada aktivitas ini sistem akan menampilkan pesan konfirmasi terlebih dahulu sebelum admin melakukan proses hapus kategori lowongan pekerjaan.
5. Menghapus kategori dari *database*, jika admin memilih untuk tetap menghapus data kategori maka pada aktivitas ini sistem akan menghapus data kategori dari *database.*
6. Menampilkan pesan hapus data dibatalkan, aktivitas ini terjadi jika admin membatalkan untuk menghapus data kategori.
7. Menampilkan pesan kategori berhasil dihapus, pada aktivitas ini sistem akan menampilkan pesan kategori berhasil dihapus jika proses hapus data kategori berhasil dilakukan.

Seluruh aktivitas hapus kategori terlihat pada activity diagram gambar III.89.



Gambar III. 89 Activity Diagram Hapus Kategori

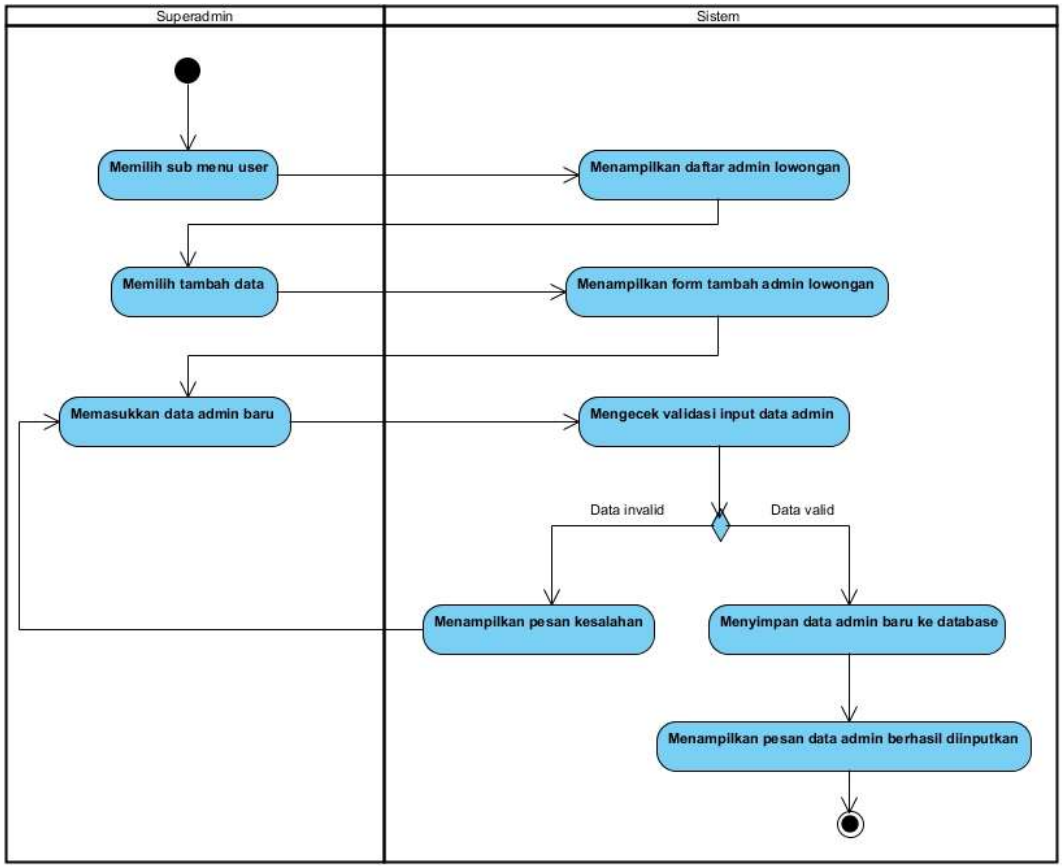
j. Activity Diagram Penambahan Admin

Aktivitas penambahan admin merupakan aktivitas yang dilakukan oleh superadmin untuk menambah data *user* admin sistem lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih sub menu user, pada aktivitas ini admin memilih sub menu user untuk menampilkan daftar user admin sistem lowongan pekerjaan.
2. Menampilkan daftar admin lowongan, pada aktivitas ini sistem akan menampilkan daftar admin lowongan pekerjaan.
3. Memilih tambah data, pada aktivitas ini admin memilih tombol tambah data untuk melakukan proses tambah data admin.

4. Menampilkan form tambah admin lowongan, pada aktivitas ini sistem akan menampilkan form untuk tambah data admin lowongan pekerjaan.
5. Memasukkan data admin baru, pada aktivitas ini admin memasukkan data admin baru pada form tambah admin baru.
6. Mengecek validasi input data admin, pada aktivitas ini sistem akan mengecek *field* pada form tambah data admin apakah sudah valid atau belum.
7. Menampilkan pesan kesalahan, aktivitas ini terjadi jika data input pada form tambah data admin tidak valid karena *field* kosong atau karena data admin sudah ada.
8. Menyimpan data admin baru ke *database*, pada aktivitas ini jika data admin baru valid maka sistem akan menyimpan data admin baru ke dalam *database*.
9. Menampilkan pesan data admin berhasil diinputkan, pada aktivitas ini jika proses penambahan data admin berhasil maka sistem akan menampilkan pesan bahwa data admin berhasil diinputkan.

Seluruh aktivitas penambahan admin terlihat pada activity diagram gambar III.90.



Gambar III. 90 Activity Diagram Penambahan Admin

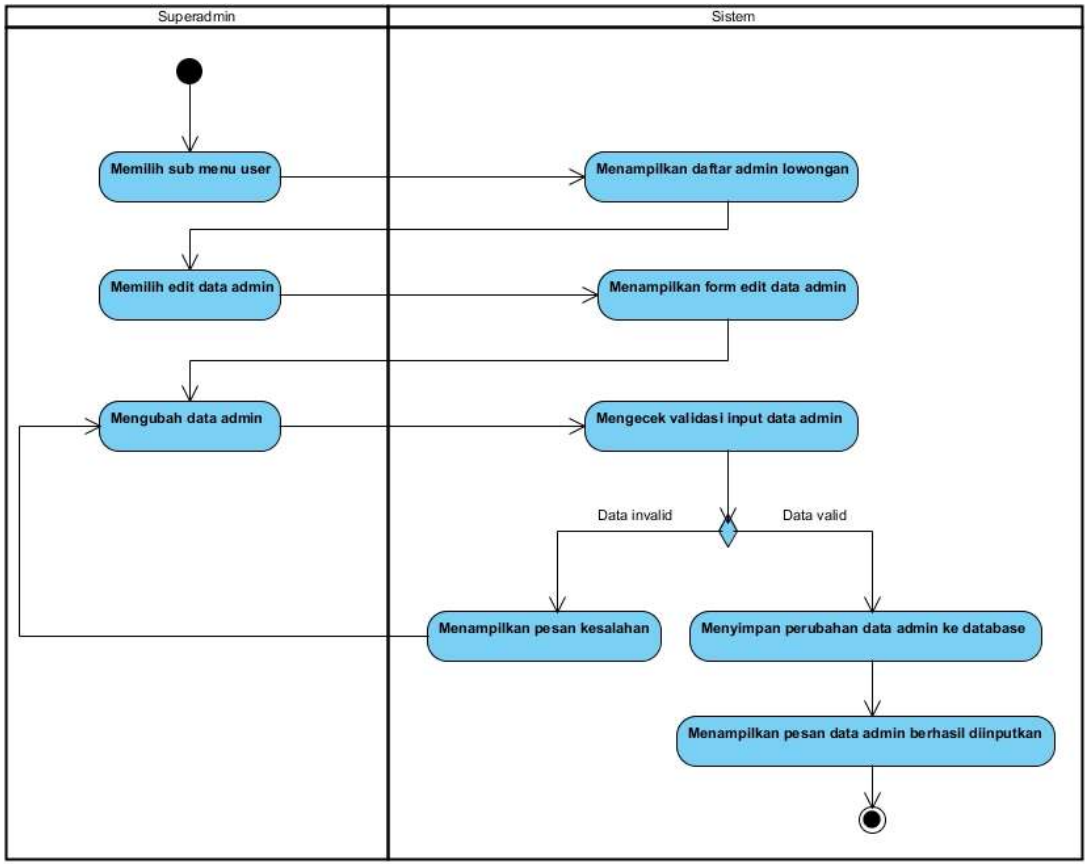
k. Activity Diagram Edit Data Admin

Aktivitas edit data admin merupakan aktivitas yang dilakukan oleh superadmin untuk mengubah data admin sistem lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih sub menu *user*, pada aktivitas ini admin memilih sub menu user untuk menampilkan daftar user admin sistem lowongan pekerjaan.
2. Menampilkan daftar admin lowongan, pada aktivitas ini sistem akan menampilkan daftar admin lowongan pekerjaan.
3. Memilih edit data admin, pada aktivitas ini admin memilih tombol edit untuk melakukan proses edit data admin.

4. Menampilkan form edit data admin, pada aktivitas ini sistem akan menampilkan form untuk mengubah data admin lowongan pekerjaan yang dipilih.
5. Mengubah data admin, pada aktivitas ini admin mengubah data admin pada form edit data admin.
6. Mengecek validasi input data admin, pada aktivitas ini sistem akan mengecek *field* pada form tambah data admin apakah sudah valid atau belum.
7. Menampilkan pesan kesalahan, aktivitas ini terjadi jika data input pada form tambah data admin tidak valid karena *field* kosong atau karena data admin sudah ada.
8. Menyimpan perubahan data admin ke *database*, pada aktivitas ini jika perubahan data admin valid maka sistem akan menyimpan perubahan data admin ke dalam *database*.
9. Menampilkan pesan data admin berhasil diinputkan, pada aktivitas ini jika proses perubahan data admin berhasil maka sistem akan menampilkan pesan bahwa data admin berhasil diinputkan.

Seluruh aktivitas edit data admin terlihat pada activity diagram gambar III.91.



Gambar III. 91 Activity Diagram Edit Data Admin

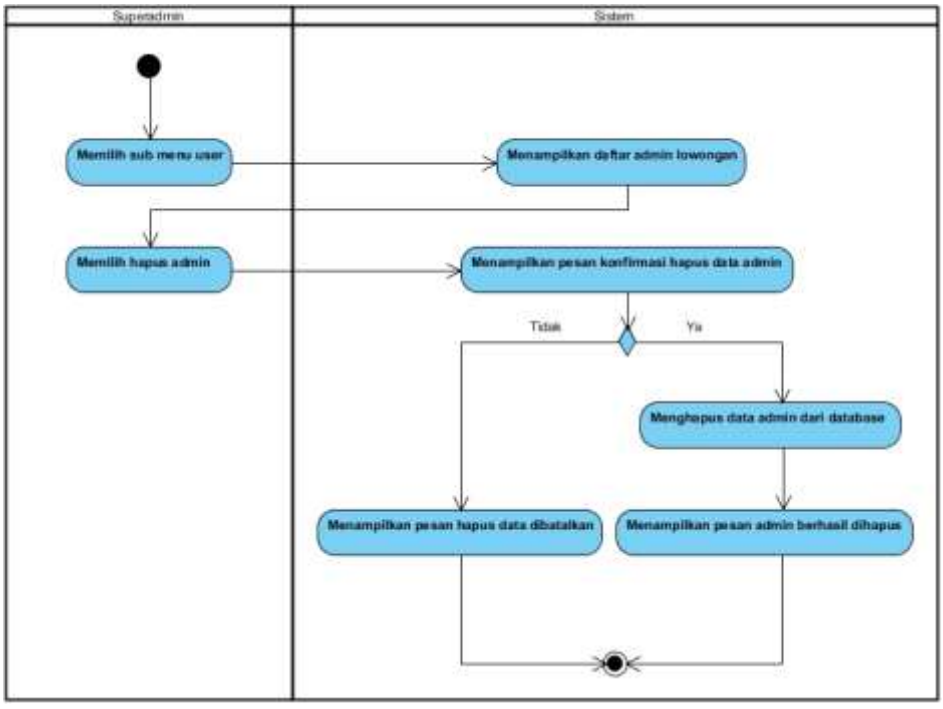
l. Activity Diagram Hapus Data Admin

Aktivitas hapus data admin merupakan aktivitas yang dilakukan oleh superadmin untuk menghapus data admin sistem lowongan pekerjaan. Dalam aktivitas ini terdapat beberapa aktivitas yaitu :

1. Memilih sub menu user, pada aktivitas ini admin memilih sub menu user untuk menampilkan daftar user admin sistem lowongan pekerjaan.
2. Menampilkan daftar admin lowongan, pada aktivitas ini sistem akan menampilkan daftar admin lowongan pekerjaan.
3. Memilih hapus admin, pada aktivitas ini admin memilih tombol hapus untuk melakukan proses hapus data admin.

4. Menampilkan pesan konfirmasi hapus data admin, pada aktivitas ini sistem akan menampilkan pesan konfirmasi terlebih dahulu sebelum admin melakukan proses data admin sistem lowongan pekerjaan.
5. Menghapus data admin dari *database*, jika admin memilih untuk tetap menghapus data admin maka pada aktivitas ini sistem akan menghapus data admin dari *database*.
6. Menampilkan pesan hapus data dibatalkan, aktivitas ini terjadi jika admin membatalkan untuk menghapus data admin.
7. Menampilkan pesan admin berhasil dihapus, pada aktivitas ini sistem akan menampilkan pesan admin berhasil dihapus jika proses hapus data admin berhasil dilakukan.

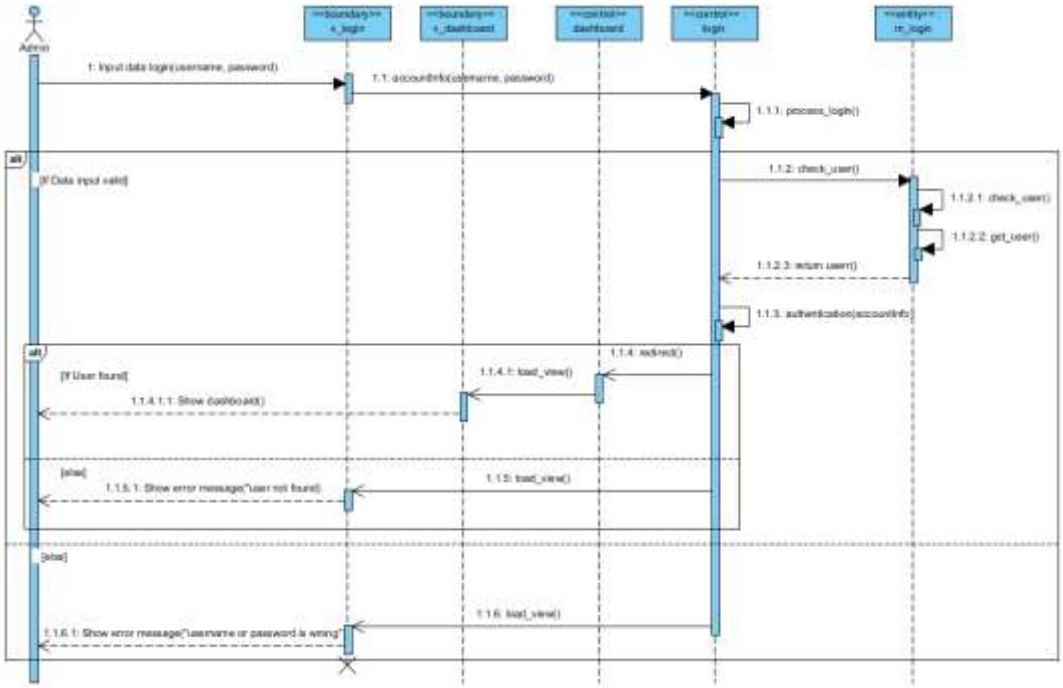
Seluruh aktivitas lihat detil member terlihat pada activity diagram gambar III.92.



Gambar III. 92 Activity Diagram Hapus Data Admin

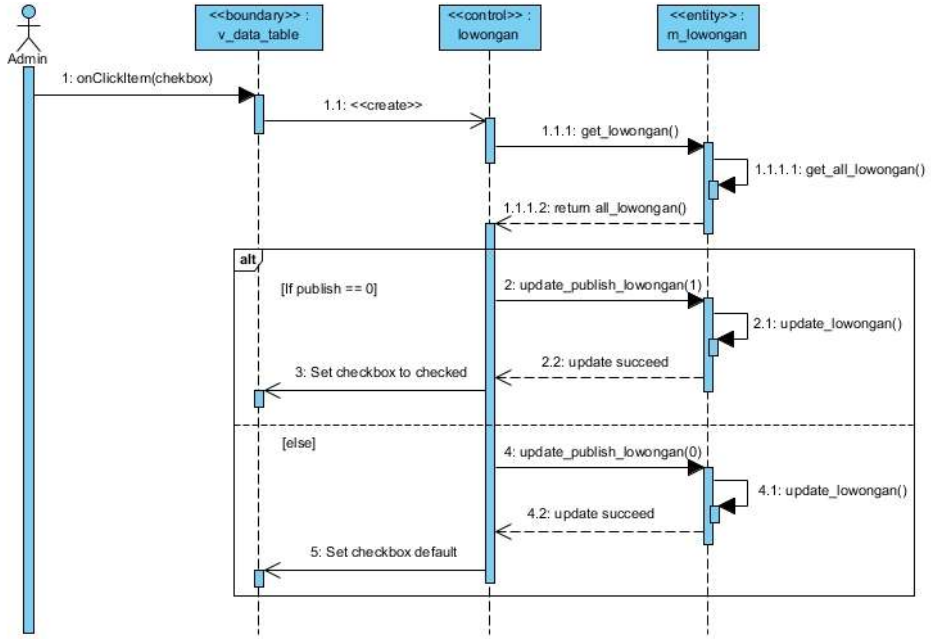
3. Sequence Diagram

a. Sequence Diagram Login



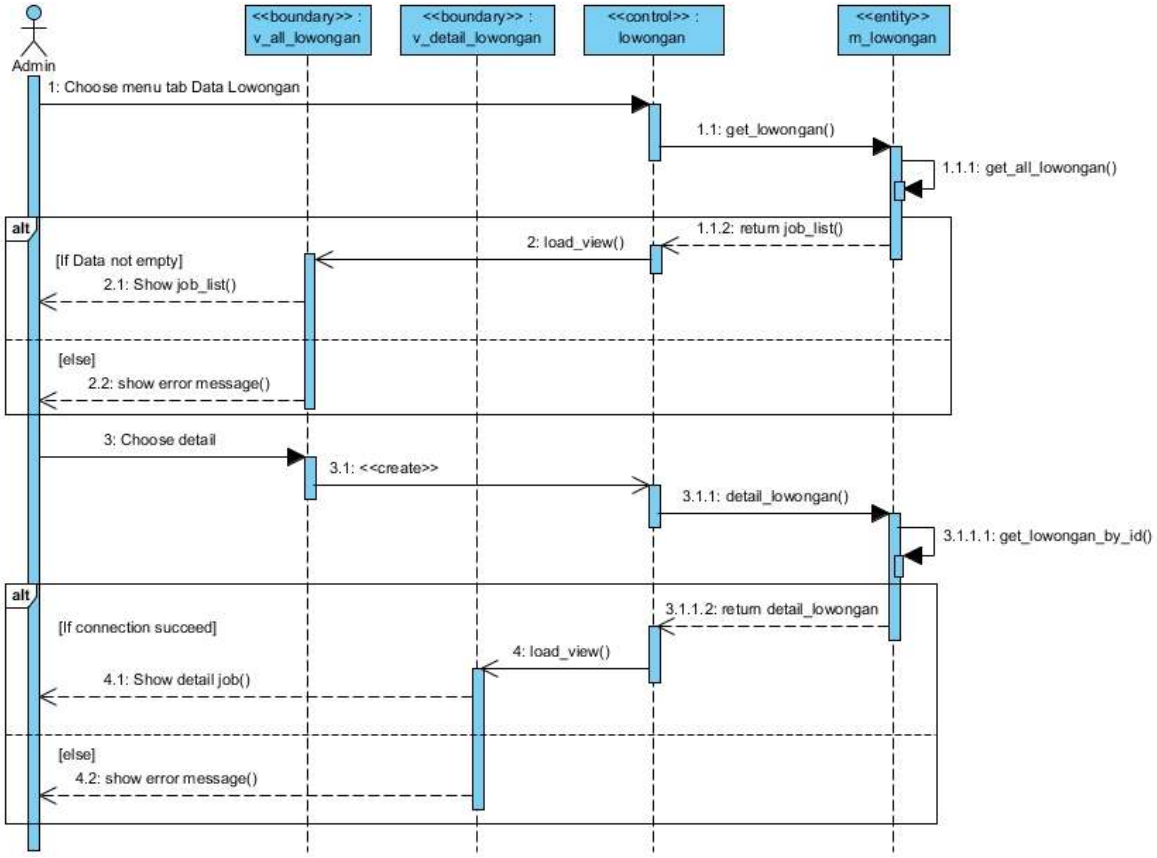
Gambar III. 93 Sequence Diagram proses Login

b. Sequence Diagram Publikasi Lowongan



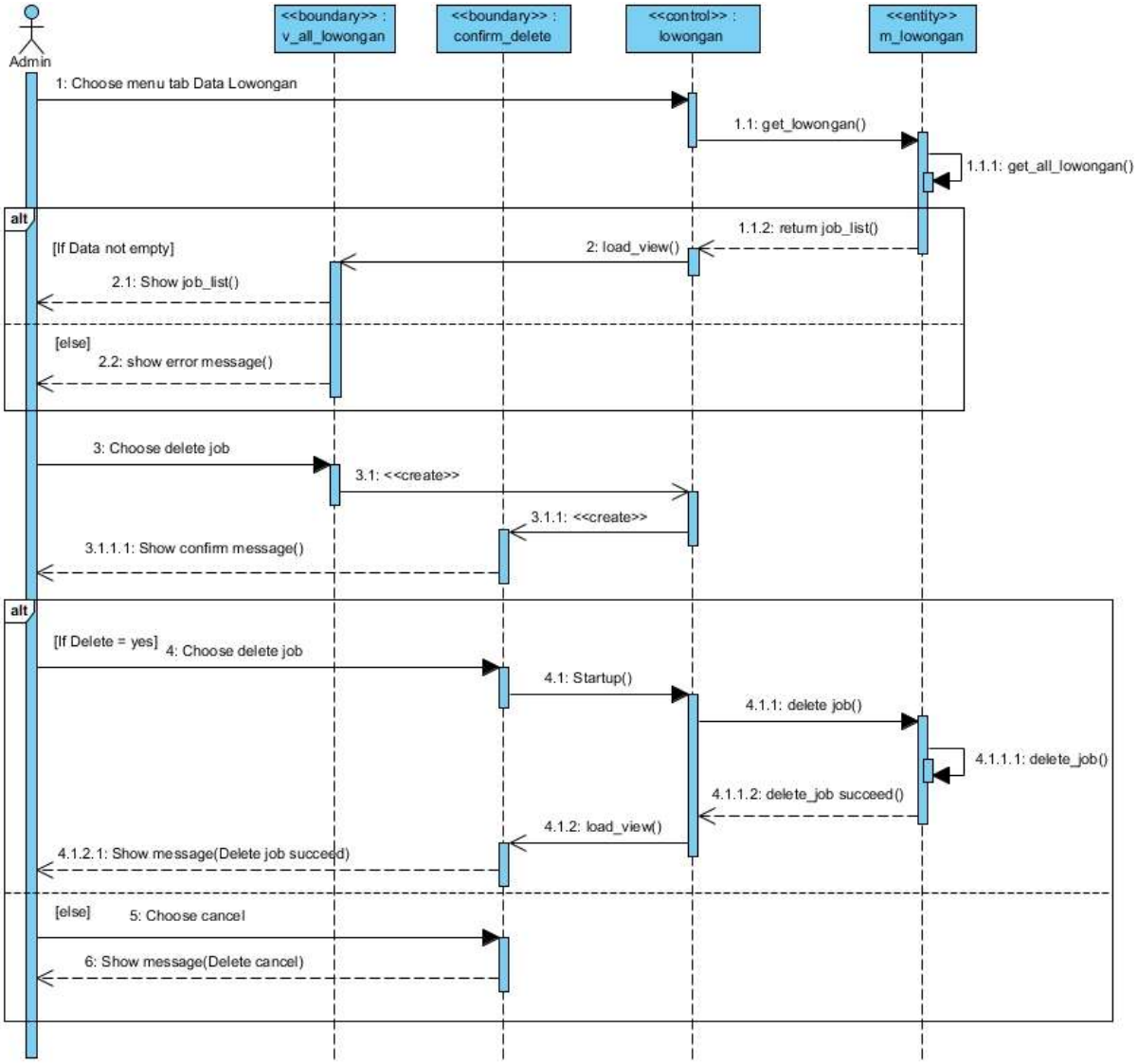
Gambar III. 94 Sequence Diagram Publikasi Lowongan.

c. Sequence Diagram Lihat Detil Lowongan



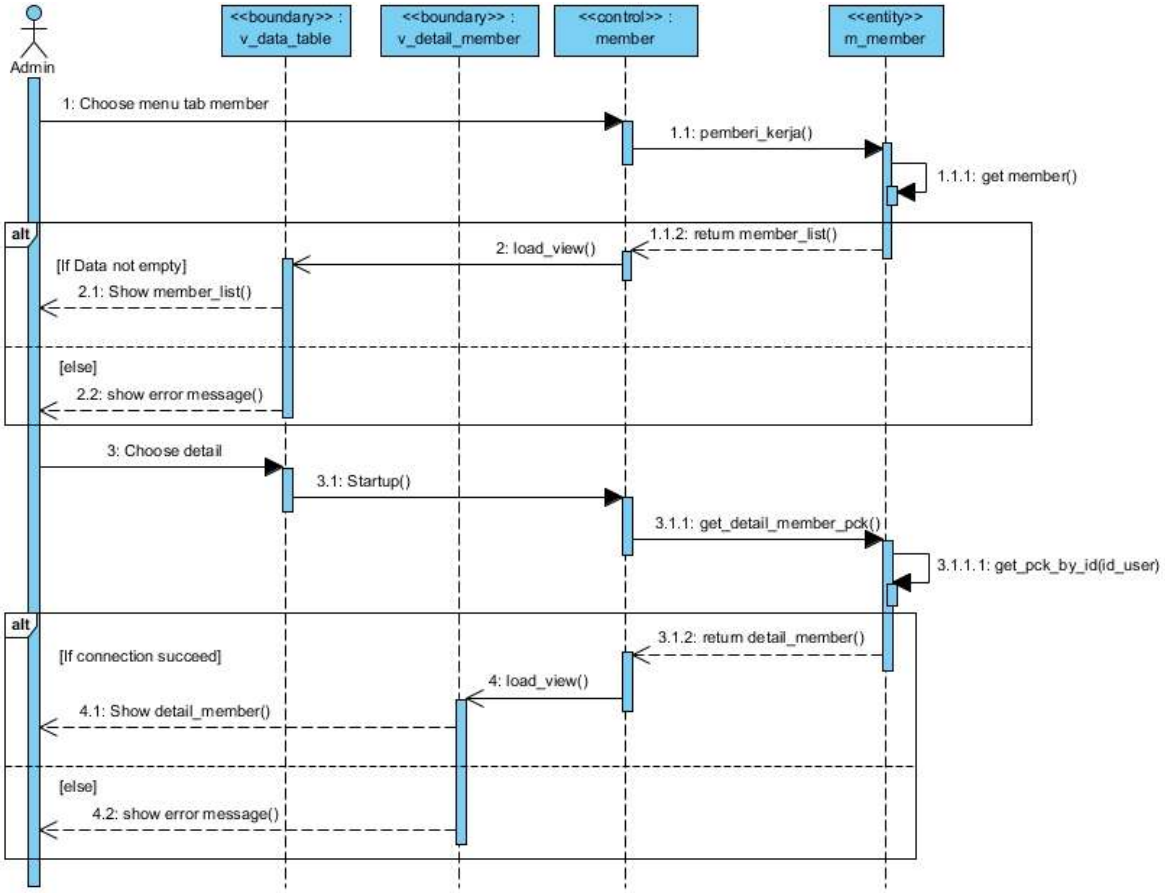
Gambar III. 95 Sequence Diagram Lihat Detil Lowongan.

d. Sequence Diagram Hapus Lowongan



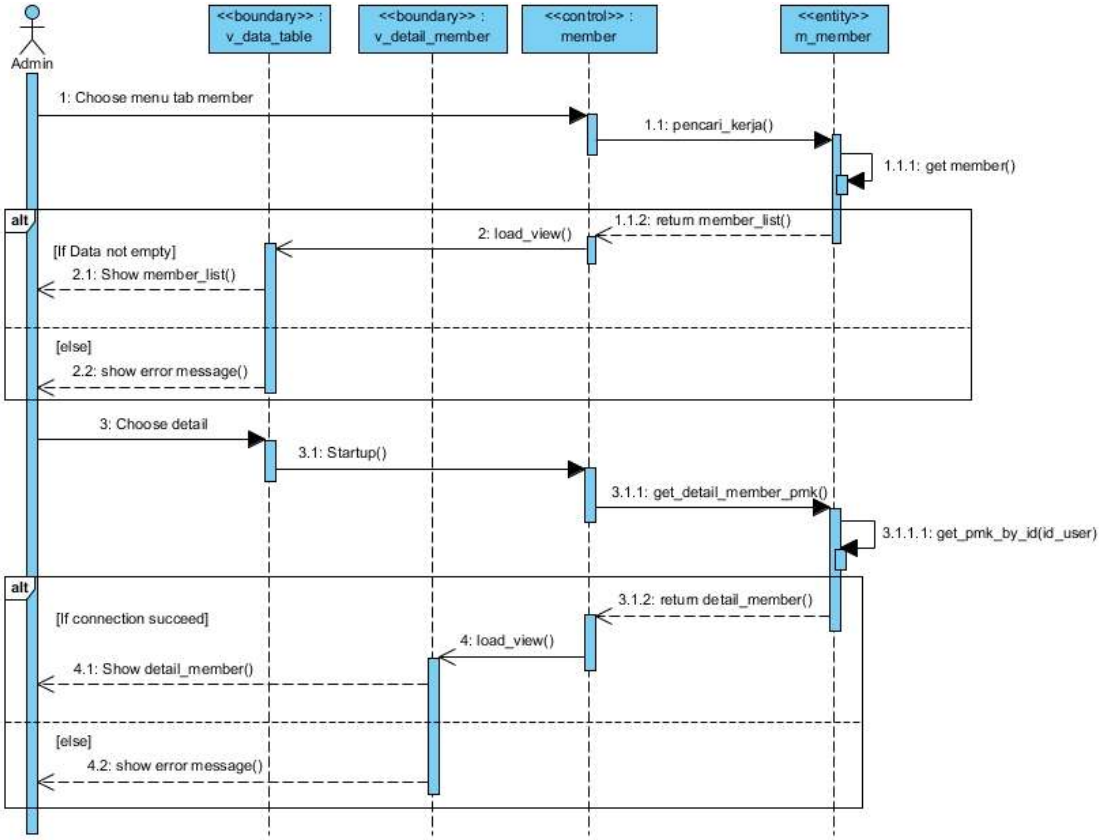
Gambar III. 96 Sequence Diagram Hapus Lowongan.

e. Sequence Diagram Lihat Detil Member Pemberi Kerja



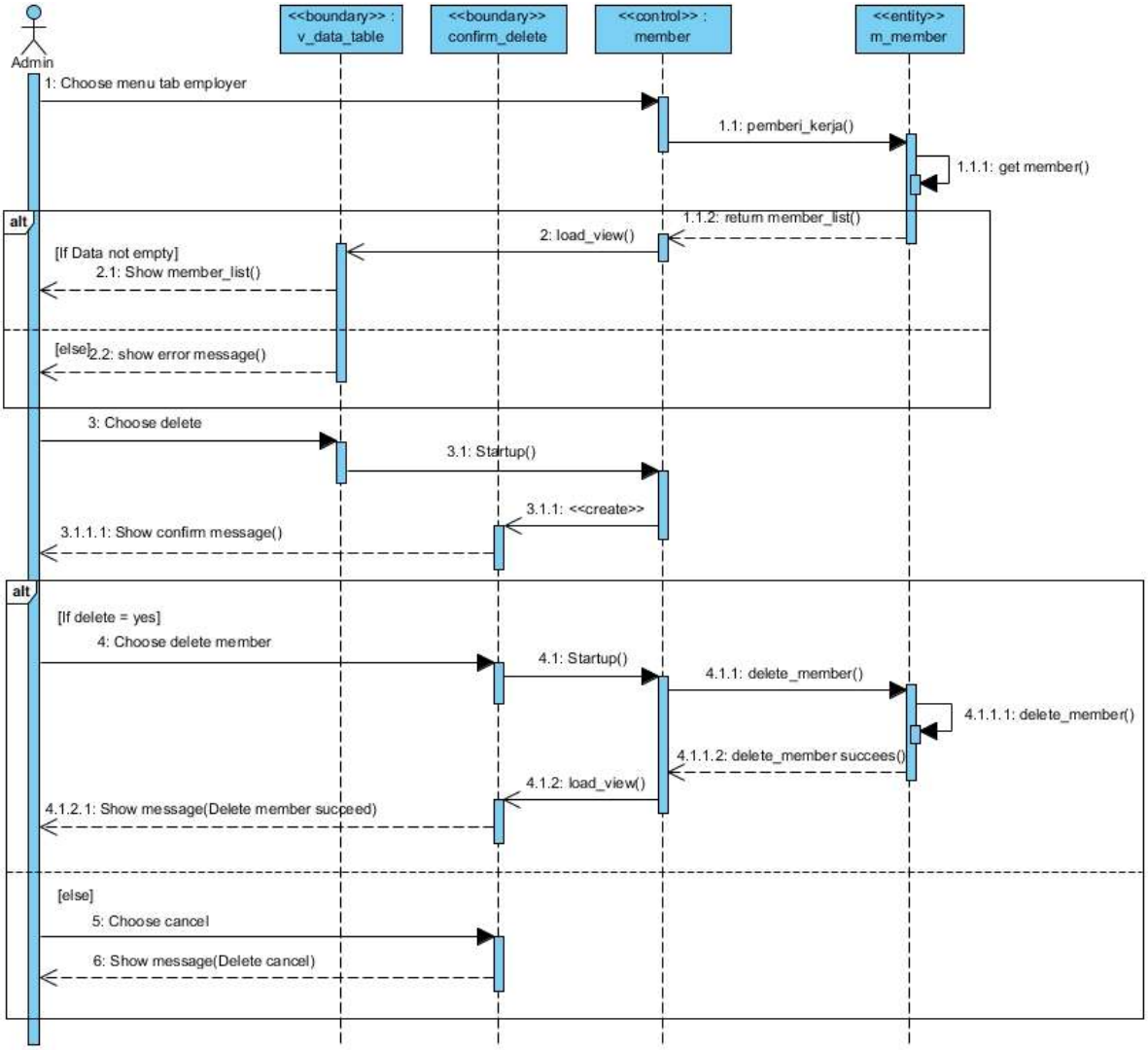
Gambar III. 97 Sequence Diagram Lihat Member Pemberi Kerja

f. Sequence Diagram Lihat Detil Member Pencari Kerja



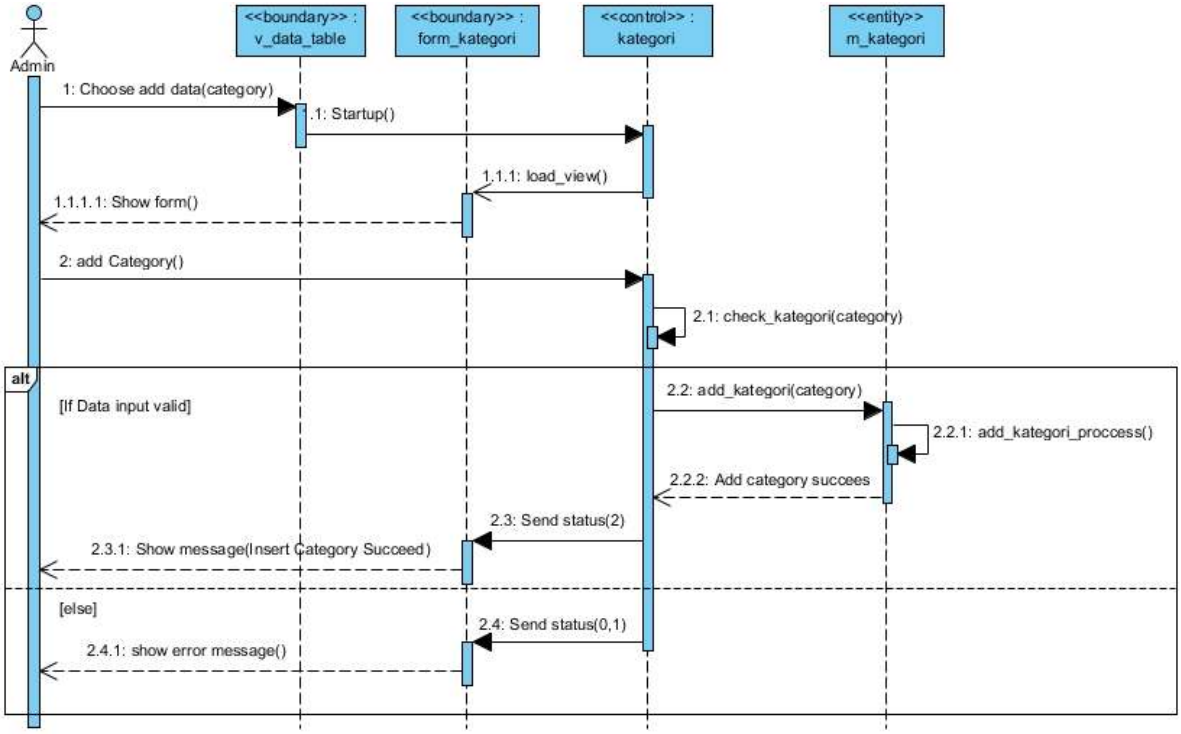
Gambar III. 98 Sequence Diagram Lihat Member Pencari Kerja

g. Sequence Diagram Hapus Data Member



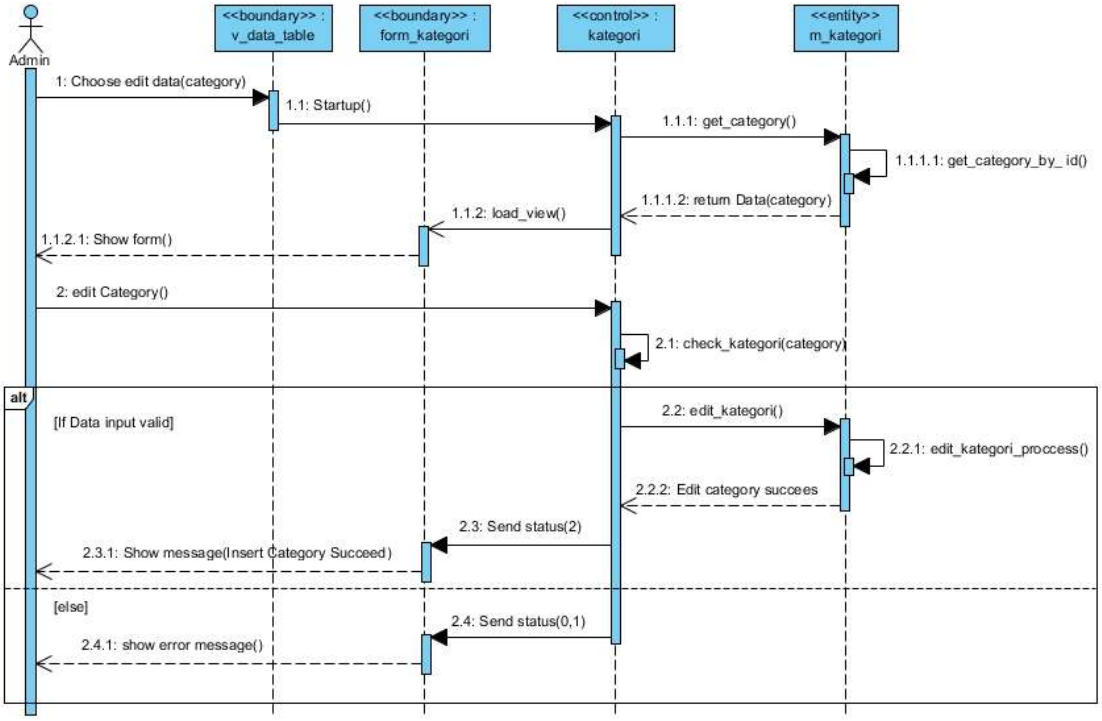
Gambar III. 99 Sequence Diagram Hapus Data Member

h. Sequence Diagram Penambahan Kategori



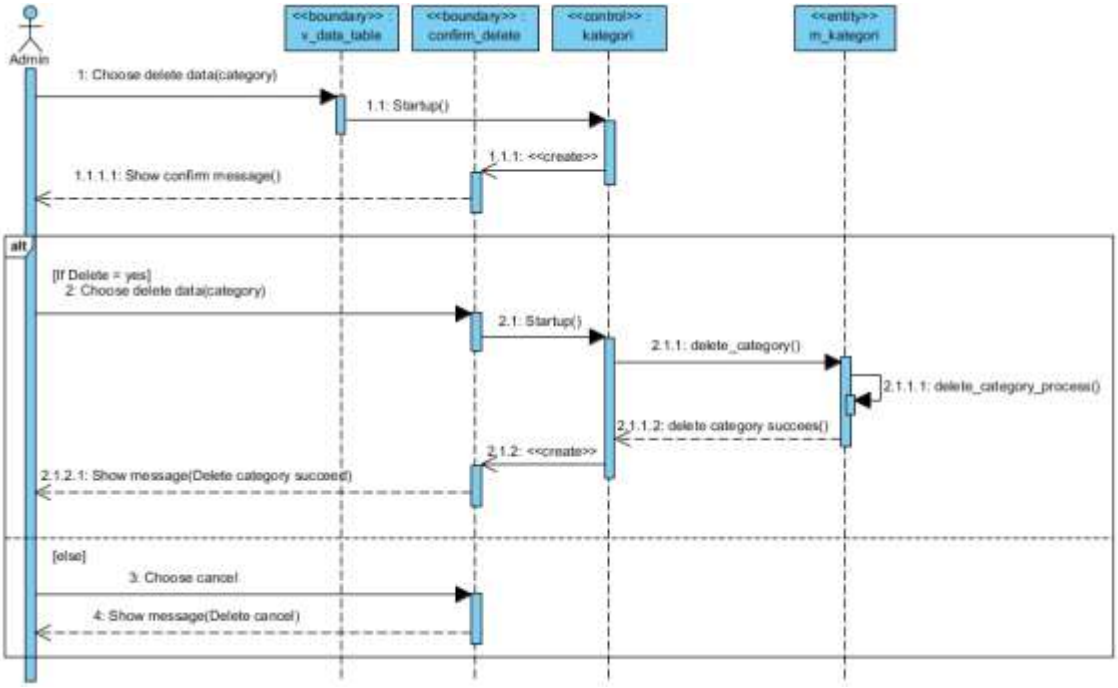
Gambar III. 100 Sequence Diagram Penambahan Kategori

i. Sequence Diagram Edit Kategori



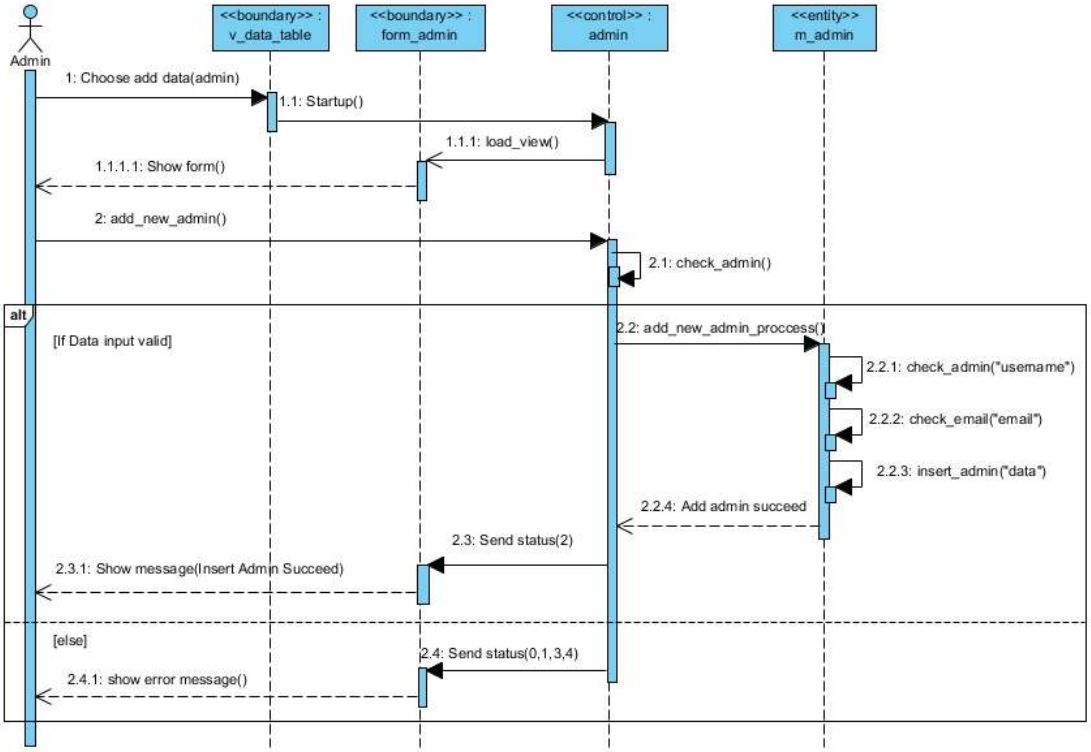
Gambar III. 101 Sequence Diagram Edit Kategori

j. Sequence Diagram Hapus Kategori



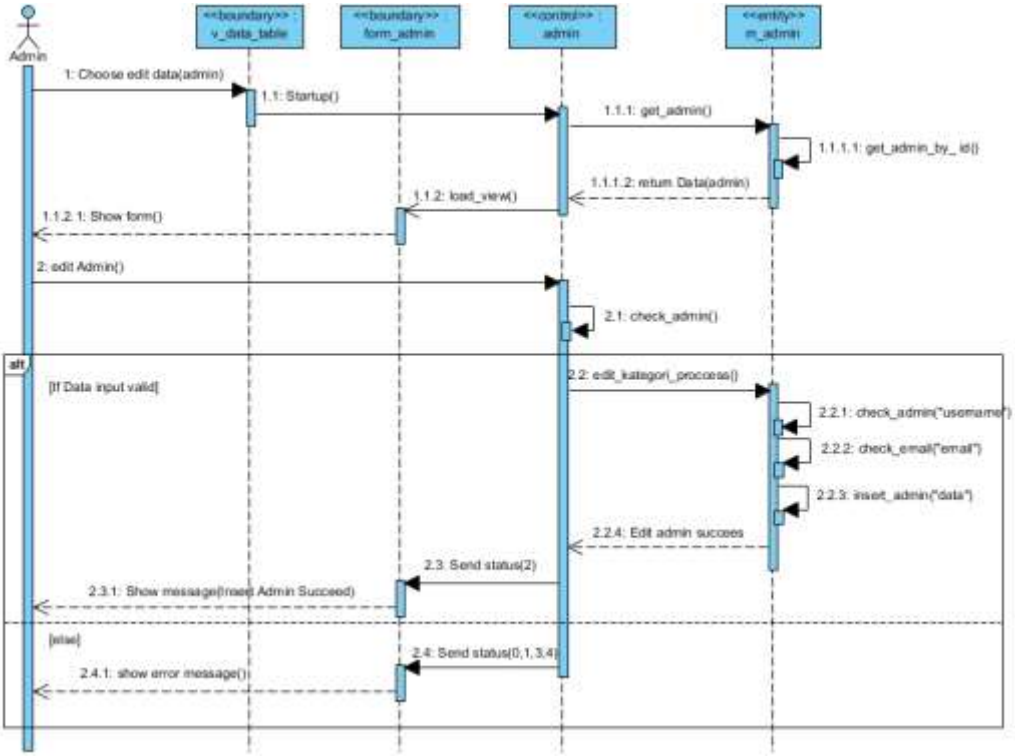
Gambar III. 102 Sequence Diagram Hapus Kategori

k. Sequence Diagram Penambahan Admin



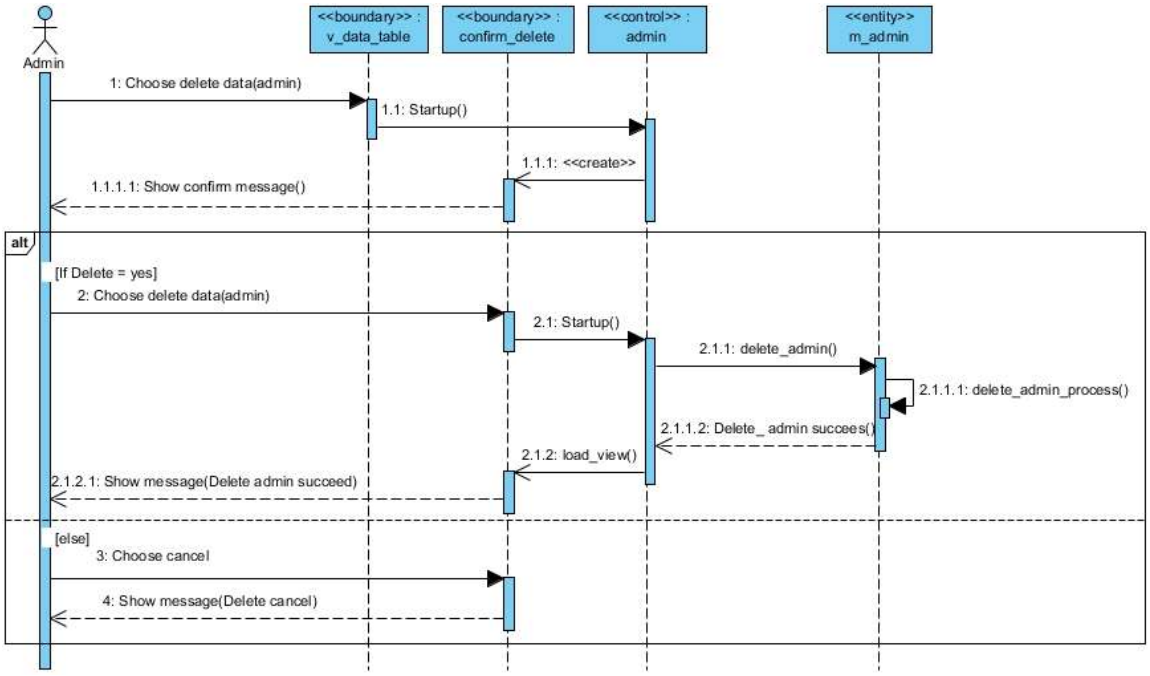
Gambar III. 103 Sequence Diagram Penambahan Admin

l. Sequence Diagram Edit Data Admin



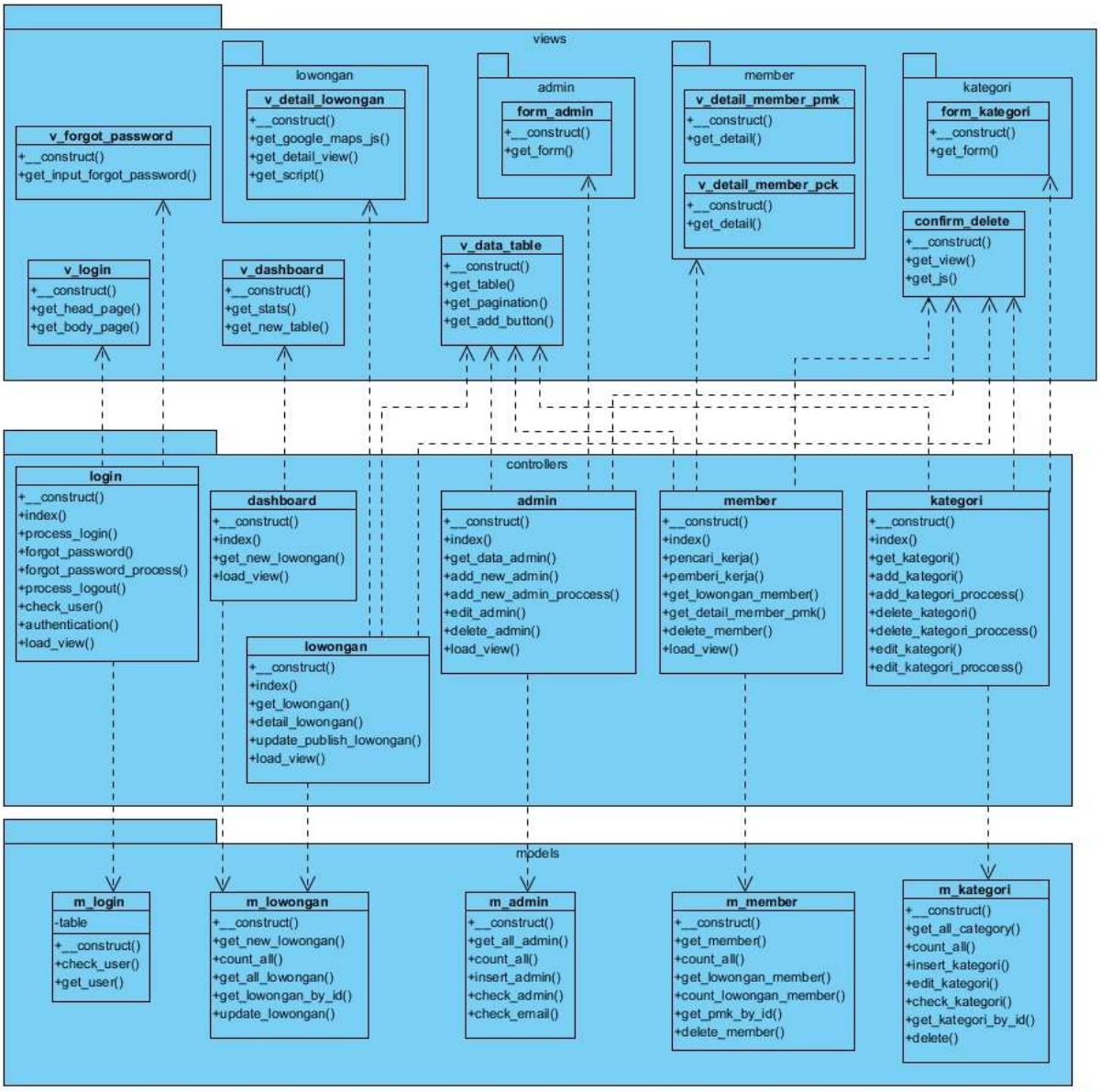
Gambar III. 104 Sequence Diagram Edit Data Admin

m. Sequence Diagram Hapus Data Admin



Gambar III. 105 Sequence Diagram Hapus Data Admin

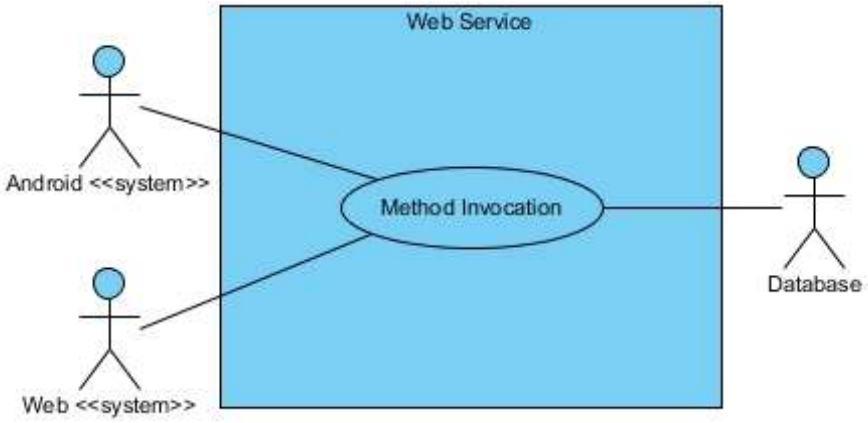
4. Class Diagram



Gambar III. 106 Class Diagram Web Admin

3.1.5.4 Analisis Pada Web Service

1. **Use Case Diagram**



Gambar III. 107 Use case diagram web service

a. Identifikasi Aktor

Pada tabel III.106 berisi daftar aktor dan deskripsi peran meraka masing-masing dalam sistem ini.

Tabel III. 106 Definisi aktor dalam web service.

NO	Aktor	Deskripsi
D-01	Android	Merupakan sistem klien berbasis *mobile* android yang mengakses web service.
D-02	Web	Merupakan sistem klien berbasis web yang mengakses web service.
D-03	Database	Merupakan sistem yang menyimpan seluruh data lowongan pekerjaan

b. Identifikasi Use Case

Berdasarkan use case diagram di atas maka dibuatlah skenario dari tiap proses yang ada pada use case diagram tersebut. Pada tabel III.107 menggambarkan skenario pada use case *Method Invocation*.

Tabel III. 107 Skenario Use Case Method Invocation

Identifikasi		
Nomor	UC-WS-01	
Nama Use Case	Method invocation	
Deskripsi	Proses pemanggilan method yang dilakukan oleh aplikasi *client* web dan android	
Aktor	Web, Android, Database	
Skenario Utama(UC-WS-01-S01)		
Kondisi awal	Koneksi ke sistem berhasil	
Aksi aktor Web / Android	**Respon Sistem**	**Aksi Database**
1. Memanggil method	2. Mencari Method	
	3. Menjalankan Method	4. Menjalankan query
	5. Menampilkan respon berupa hasil query dalam bentuk SOAP	
Kondisi akhir	Sistem menampilkan respon dalam bentuk SOAP	

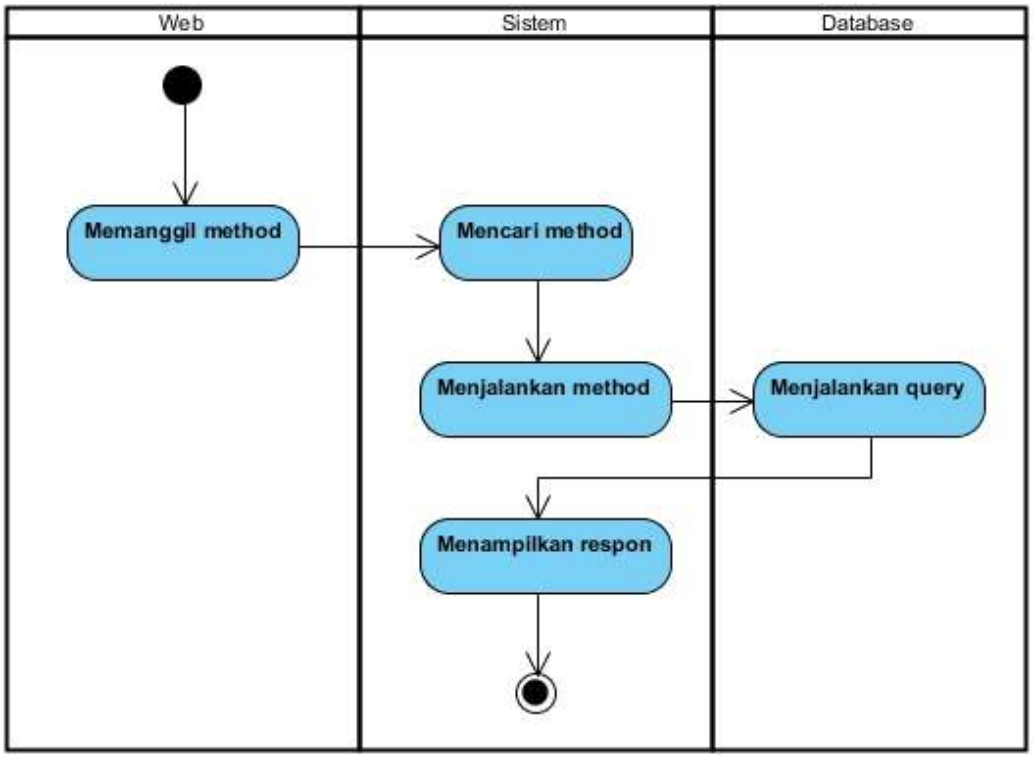
2. **Activity Diagram**

a. Activity Diagram *Method Invocation*

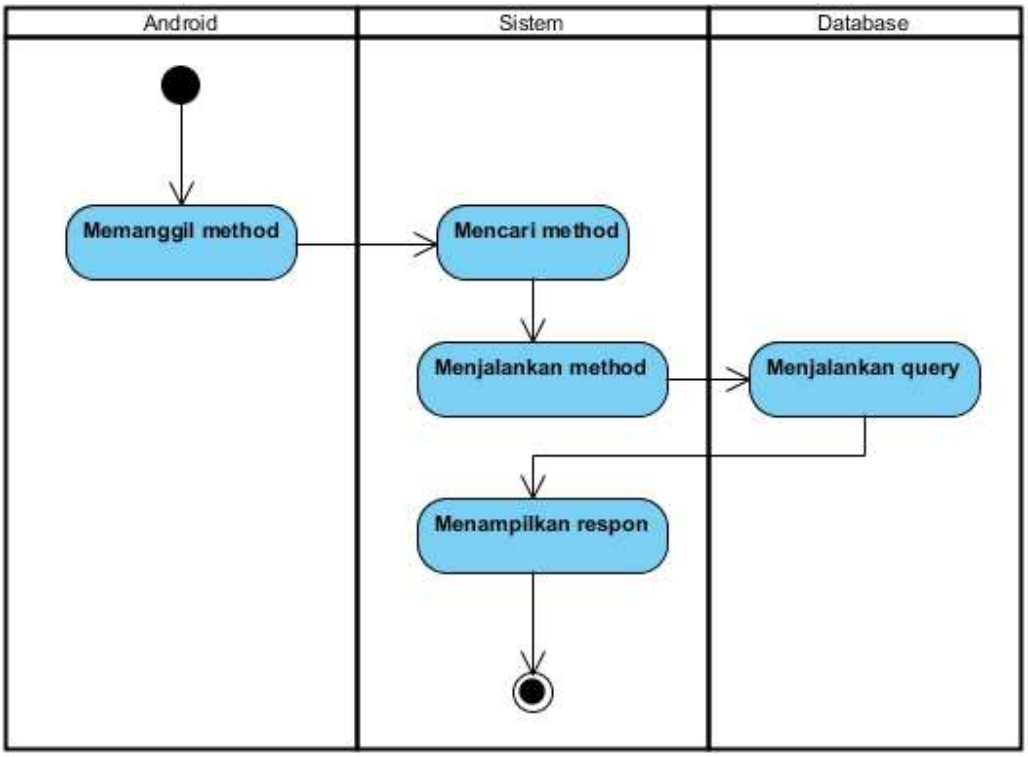
Aktivitas *Method Invocation* merupakan aktivitas yang dilakukan oleh android dan web untuk memanggil method yang ada pada sistem. Dalam aktivitas ini terdapat beberapa tahapan aktivitas , yaitu :

1. Memanggil method, dalam aktivitas ini aplikasi web dan android memanggil method dengan mengakses*:*
   http://lowongankerja.sytes.net:8080/serviceLowongan/DataLowonganService?WSDL
2. Mencari Method, aktivitas ini merupakan aktivitas yang dilakukan oleh sistem untuk mencari method yang dipanggil oleh aplikasi web dan android.
3. Menjalankan method, pada aktivitas ini sistem menjalankan method yang aplikasi web dan android panggil.
4. Menampilkan respon, pada aktivitas ini sistem menampilkan respon berupa hasil query yang ditampilkan dalam bentuk SOAP.

Seluruh aktivitas method invocation dari aplikasi mobile android dan web terlihat pada *activity diagram* gambar III.108 dan gambar III.109.



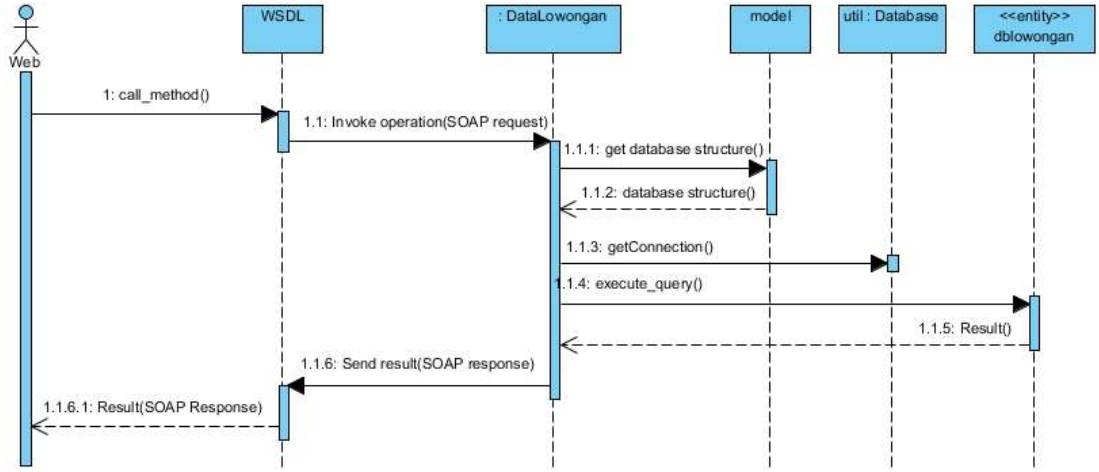
Gambar III. 108 Activity Diagram Method Invocation aplikasi web



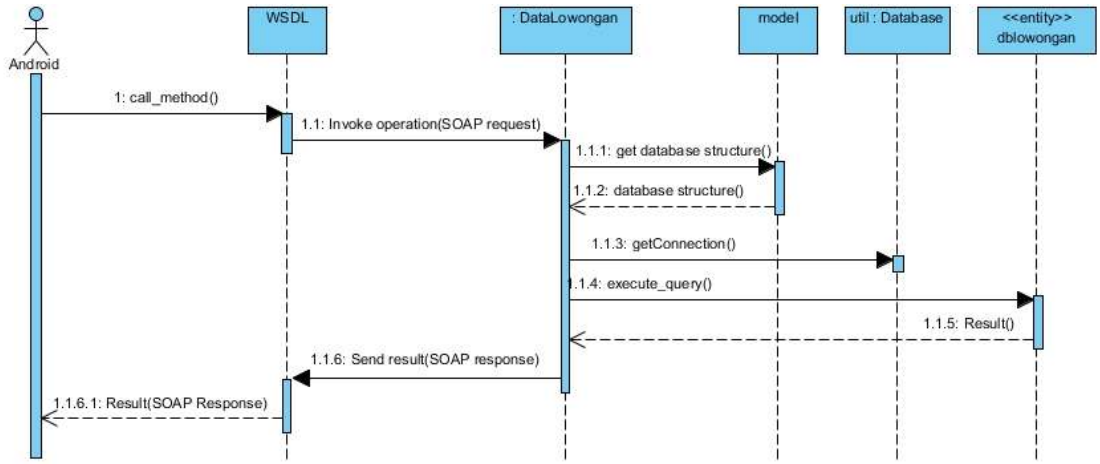
Gambar III. 109 Activity Diagram Method Invocation aplikasi android

3. Sequence Diagram

a. Sequence Diagram Method Invocation



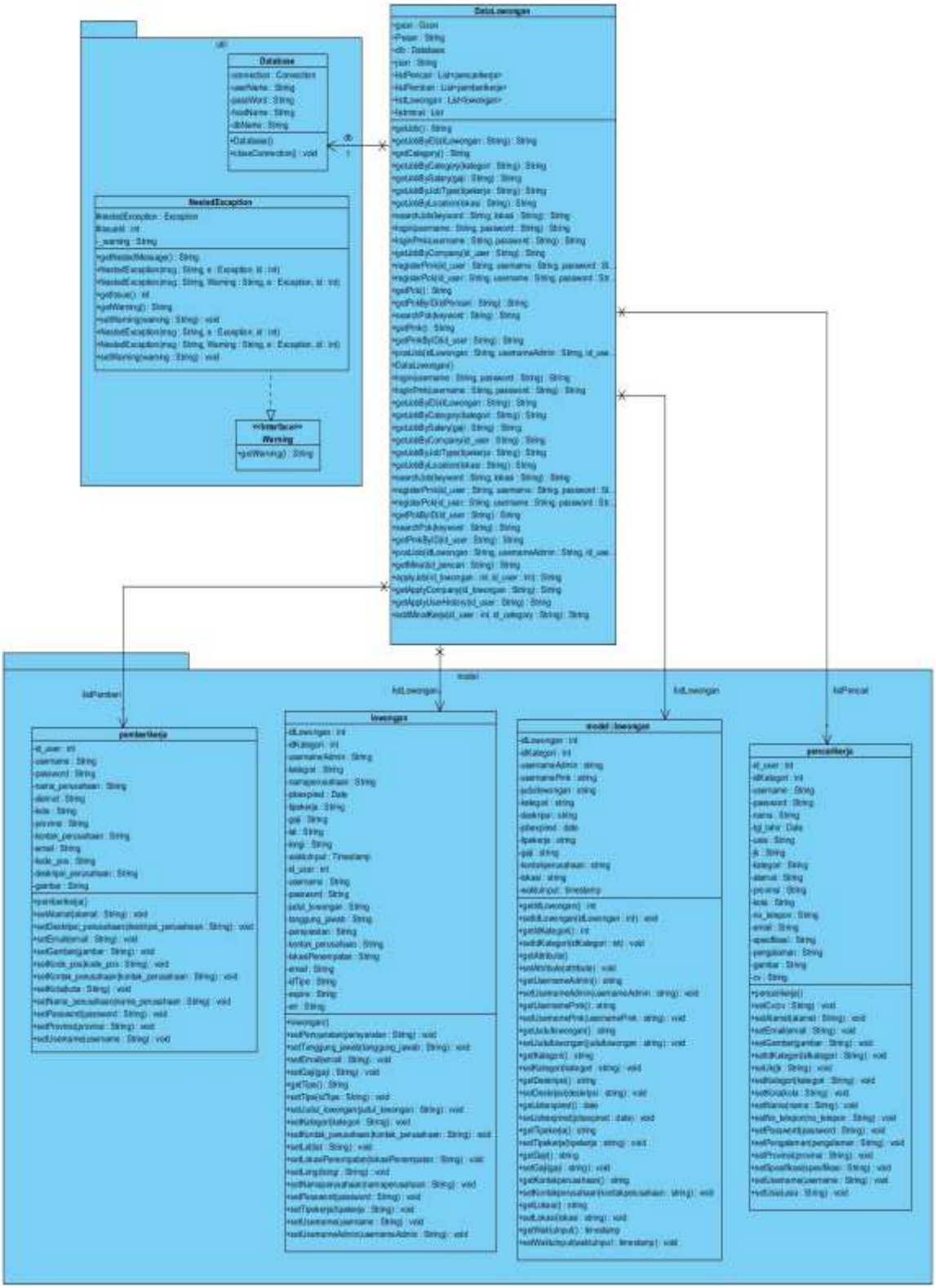
Gambar III. 110 Sequence Diagram Method Invocation aplikasi web



Gambar III. 111 Sequence Diagram Method Invocation aplikasi android

4. Class Diagram

Class diagram menggambarkan struktur dan deskripsi class beserta hubungan satu sama lain seperti Asosiasi, Generalisasi dan lain sebagainya.



Gambar III. 112 Class Diagram Sistem Web Service

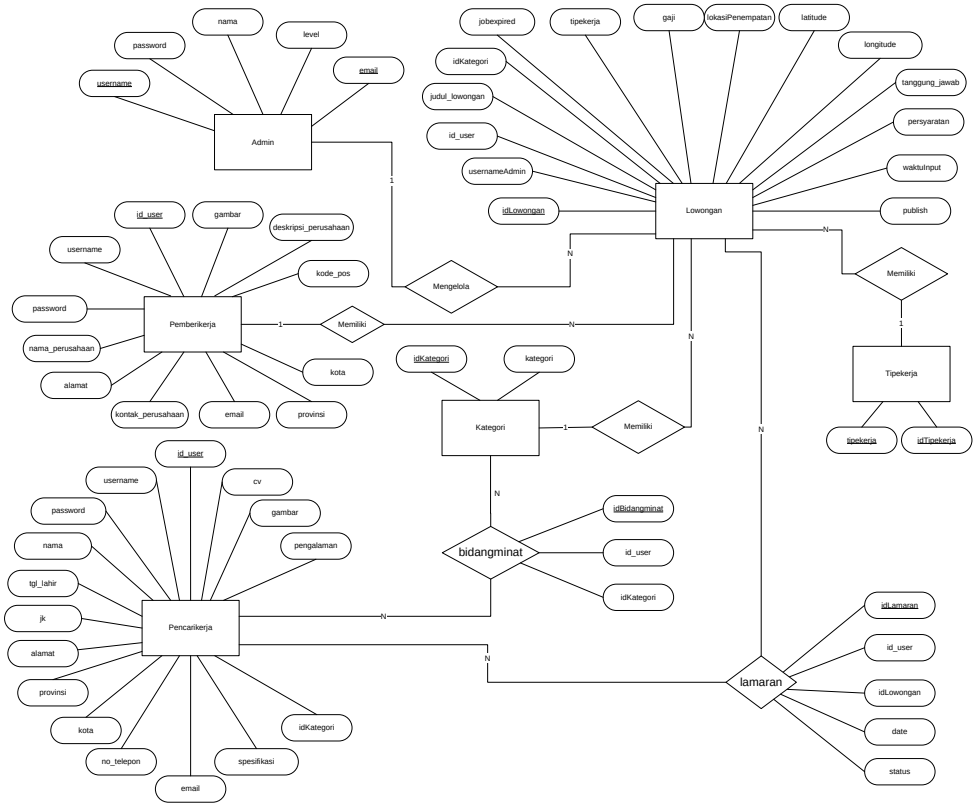
3.2 Perancangan

3.2.1 Perancangan Basis Data

Perancangan basis data adalah menciptakan atau merancang kumpulan data yang terhubung dan tersimpan secara bersama-sama.

3.2.1.1 Entity Relationship Diagram

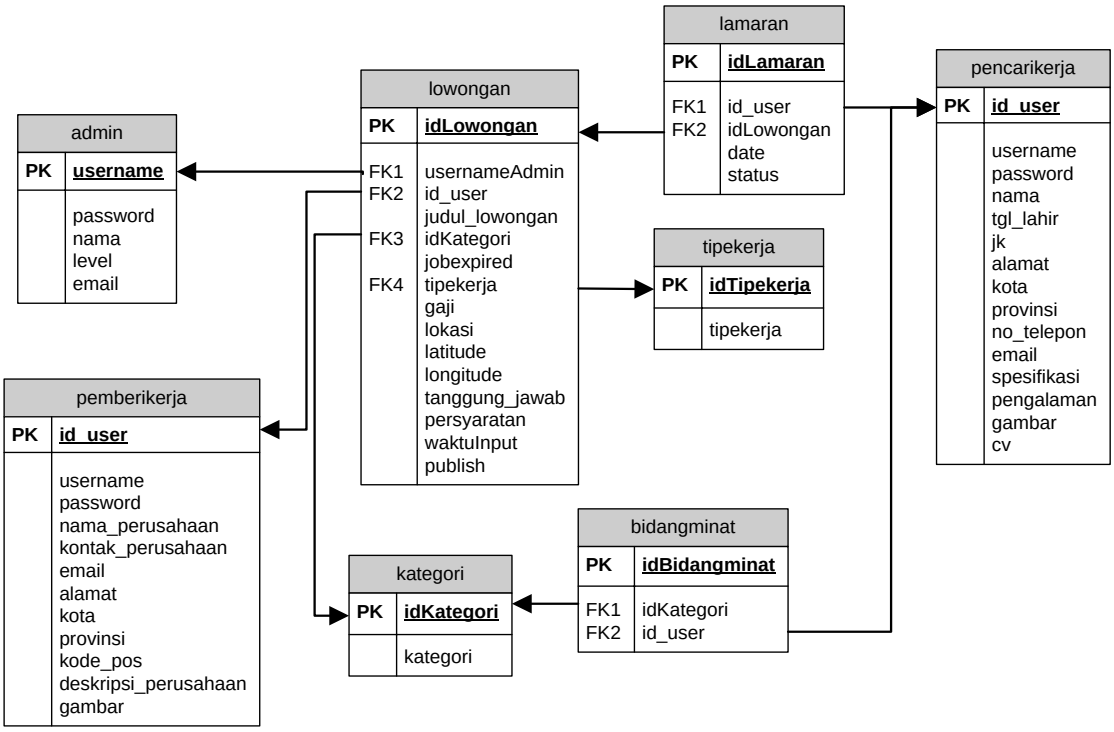
Dalam memodelkan data dan menggambarkan hubungan antara data yang ada pada sistem digunakan alat bantu yaitu diagram E-R. Sistem yang akan dibangun dapat dilihat hubungan antar entitas. Untuk itu sistem yang akan dirancang memiliki usulan ERD yaitu:



Gambar III. 113 ER Diagram

3.2.1.2 Skema Relasi

Skema relasi menggambarkan hubungan antar data, arti data dan batasannya dijelaskan dengan baris dan kolom.



Gambar III. 114 Skema Relasi

3.2.1.3 Perancangan Struktur Tabel

1. **Struktur Tabel Pada Server**

Tabel-tabel yang terdapat dalam basis data yang digunakan dalam sistem ini adalah sebagai berikut :

a. Tabel Admin

Tabel ini adalah perancangan untuk tabel admin yang akan digunakan pada pembangunan sistem. Struktur tabel admin adalah sebagai berikut :

Tabel III. 108 Struktur Tabel Admin

No	Nama Field	Type	Ukuran	Keterangan
1	username	Varchar	8	**Primary Key*
2	password	Varchar	41	
3	nama	Varchar	50	
4	level	Enum	(‘ADMIN’, ’SUPERADMIN’)	
5	email	Varchar	100	

b. Tabel Lowongan

Tabel ini adalah perancangan untuk tabel lowongan yang akan digunakan untuk menampung seluruh data lowongan. Struktur tabel lowongan adalah sebagai berikut:

Tabel III. 109 Strukur Tabel Lowongan

No	Nama Field	Type	Ukuran	Keterangan
1	idLowongan	Integer	5	**Primary Key*
2	usernameAdmin	Varchar	20	***Foreign key, Reference* admin (username)
3	id_user	Integer	5	**Foreign key, *Reference* pemberikerja (id_user)
4	judul_lowongan	Varchar	100	
5	idKategori	Integer	5	***Foreign key, Reference* kategori (idKategori)

6	jobexpired	Date		
7	tipekerja	Integer	5	***Foreign key, Reference* tipekerja (idTipekerja)
8	gaji	Varchar	100	
9	lokasiPenempatan	Varchar	100	
10	latitude	Decimal	8,5	
11	longitude	Decimal	8,5	
12	tanggung_jawab	Text		
13	persyaratan	Text		
14	waktuInput	Timestamp		
15	publish	Int	1	

c. Tabel Kategori

Tabel ini adalah perancangan untuk tabel kategori yang akan digunakan untuk menampung kategori dari lowongan. Struktur tabel kategori adalah sebagai berikut:

Tabel III. 110 Strukur Tabel Kategori

No	Nama Field	Type	Ukuran	Keterangan
1	idKategori	Integer	5	**Primary Key*
2	kategori	Varchar	100	

d. Tabel Pemberikerja

Tabel ini adalah perancangan untuk tabel pemberi kerja yang akan digunakan untuk menampung seluruh data dari pemberi kerja. Struktur tabel pemberikerja adalah sebagai berikut:

Tabel III. 111 Strukur Tabel Pemberikerja

No	Nama Field	Type	Ukuran	Keterangan
1	id_user	Integer	5	**Primary Key*
2	username	Varchar	20	*Unique*
3	password	Varchar	41	
4	nama_perusahaan	Varchar	50	
5	kontak_perusahaan	Varchar	15	
6	email	Varchar	50	*Unique*
7	alamat	Varchar	100	
8	kota	Varchar	20	
9	provinsi	Varchar	20	
10	kode_pos	Varchar	5	
12	deskripsi_perusahaan	Text		
13	gambar	Varchar	100	

e. Tabel Pencarikerja

Tabel ini adalah perancangan untuk tabel pencari kerja yang akan digunakan untuk menampung seluruh data dari pencari kerja. Struktur tabel pencarikerja adalah sebagai berikut:

Tabel III. 112 Strukur Tabel Pencarikerja

No	Nama Field	Type	Ukuran	Keterangan
1	id_user	Integer	5	**Primary Key*
2	username	Varchar	20	*Unique*
3	password	Varchar	41	
4	nama	Varchar		
5	tgl_lahir	Date	100	
6	jk	Varchar	12	
7	alamat	Varchar	100	
8	kota	Varchar	20	
9	provinsi	Varchar	20	
10	no_telepon	Varchar	15	
11	email	Varchar	50	*Unique*
12	spesifikasi	Text		
13	pengalaman	Text		
14	gambar	Varchar	100	
15	cv	Varchar	255	

f. Tabel Tipekerja

Tabel ini adalah perancangan untuk tabel tipe kerja yang akan digunakan untuk menampung tipe kerja dari lowongan. Struktur tabel tipekerja adalah sebagai berikut:

Tabel III. 113 Strukur Tabel Tipekerja

No	Nama Field	Type	Ukuran	Keterangan
1	idTipekerja	Integer	5	**Primary Key*
2	tipekerja	Varchar	15	

g. Tabel Bidangminat

Tabel ini adalah perancangan untuk tabel bidang minat yang akan digunakan untuk menampung bidang minat dari pencari kerja. Struktur tabel bidangminat adalah sebagai berikut:

Tabel III. 114 Struktur tabel bidangminat

No	Nama Field	Type	Ukuran	Keterangan
1	idBidangminat	Integer	5	**Primary Key*
2	id_user	Integer	5	***Foreign key*, *Reference* pencarikerja (id_user)
3	idKategori	Integer	5	***Foreign key*, *Reference* kategori (idKategori)

h. Tabel Lamaran

Tabel ini adalah perancangan untuk tabel lamaran yang akan digunakan untuk menampung lamaran dari pencari kerja. Struktur tabel lamaran adalah sebagai berikut:

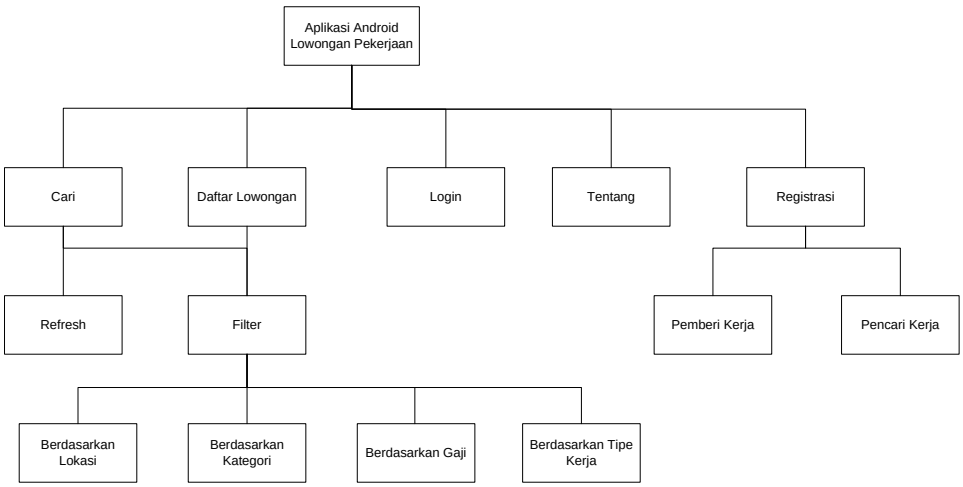
Tabel III. 115 Struktur tabel lamaran

No	Nama Field	Type	Ukuran	Keterangan
1	idLamaran	Integer	5	**Primary Key*
2	id_user	Integer	10	***Foreign key*, *Reference* pencarikerja (id_user)
3	idLowongan	Integer	10	***Foreign key*, *Reference* lowongan (idLowongan)
4	date	Date		
5	status	Enum	(‘Di Kirim’, ‘Di Panggil)	

3.2.2 Perancangan Struktur Menu

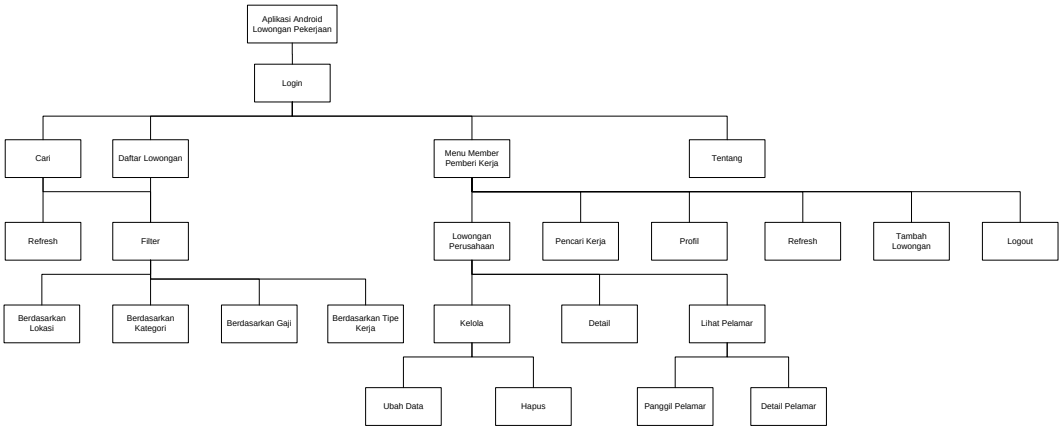
3.2.2.1 Perancangan Struktur Menu Aplikasi Android

1. Struktur Menu Pengunjung Umum



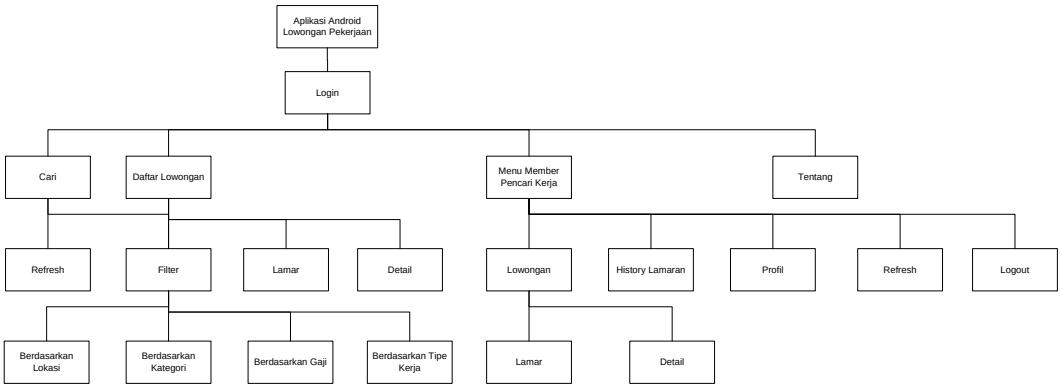
Gambar III. 115 Struktur Menu Pengunjung Umum Aplikasi Android

2. Struktur Menu Member Pemberi Kerja



Gambar III. 116 Struktur Menu Member Pemberi Kerja Aplikasi Android

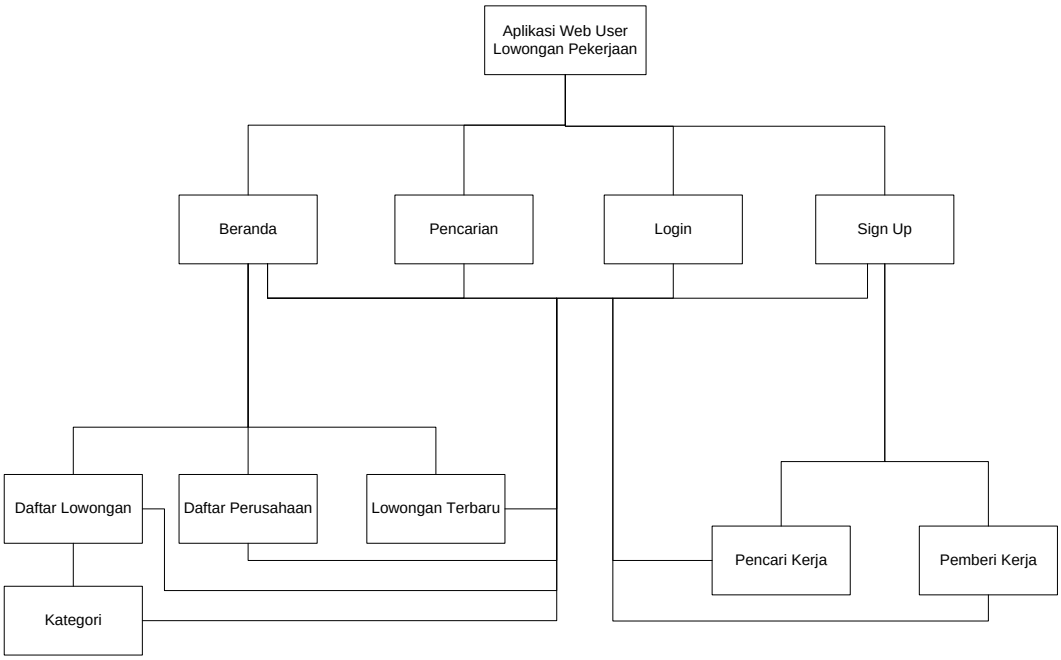
3. Struktur Menu Member Pencari Kerja



Gambar III. 117 Struktur Menu Member Pencari Kerja Aplikasi Android

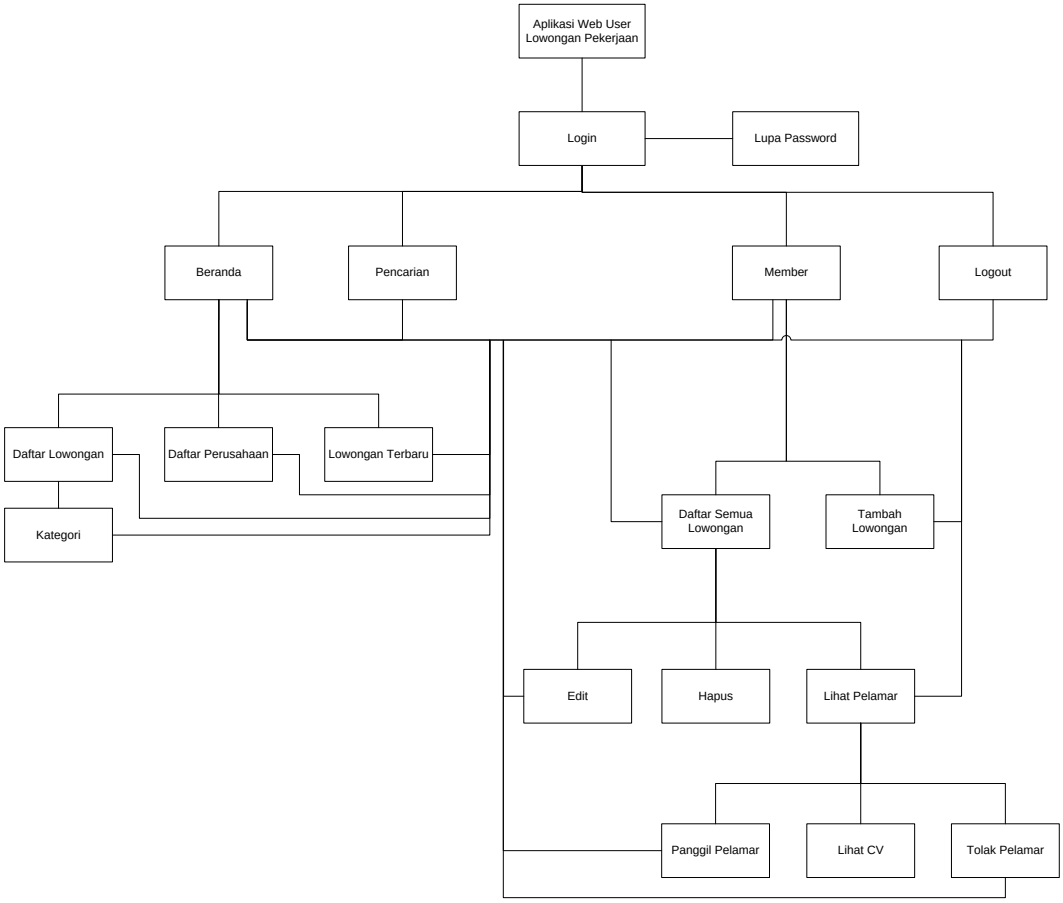
3.2.2.2 Perancangan Struktur Menu Aplikasi Web

1. Struktur Menu Pengunjung Umum



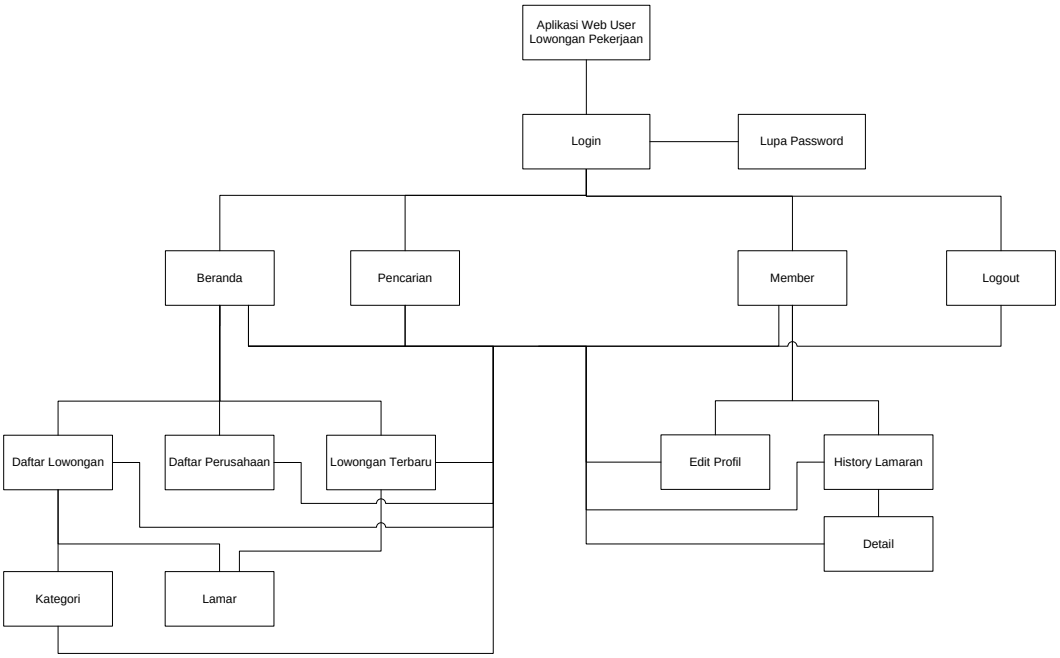
Gambar III. 118 Struktur Menu Pengunjung Umum Aplikasi Web

2. Struktur Menu Member Pemberi Kerja



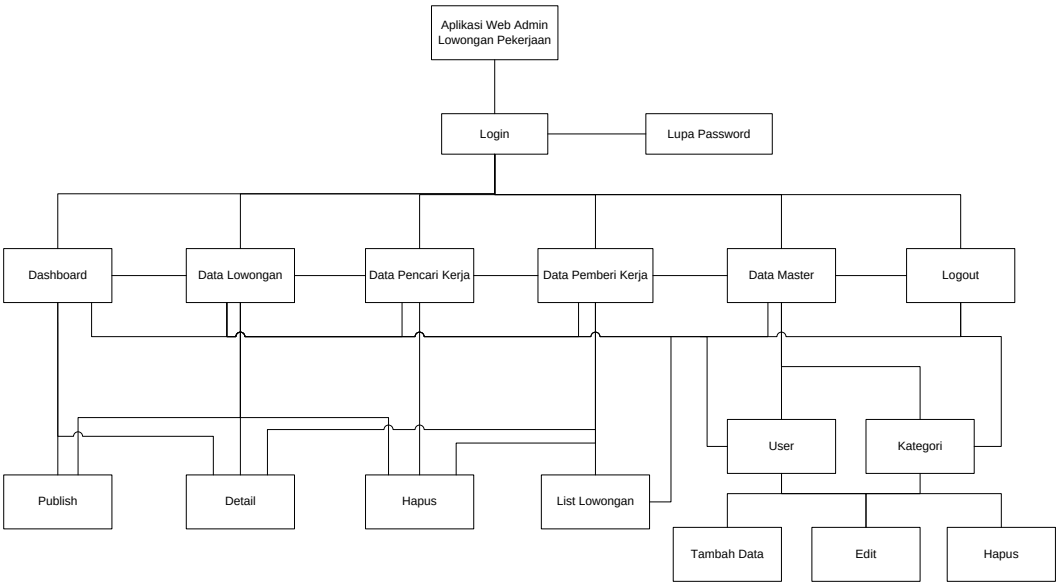
Gambar III. 119 Struktur Menu Member Pemberi Kerja Aplikasi Web

3. Struktur Menu Member Pencari Kerja



Gambar III. 120 Struktur Menu Member Pencari Kerja Aplikasi Web

3.2.2.3 Perancangan Struktur Menu Web Admin

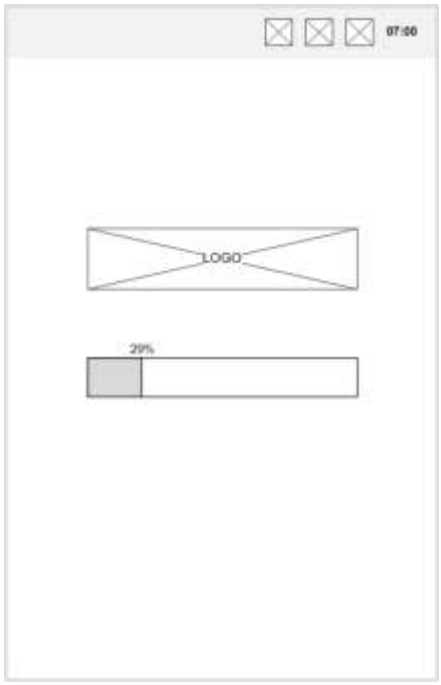


Gambar III. 121 Struktur Menu Web Admin

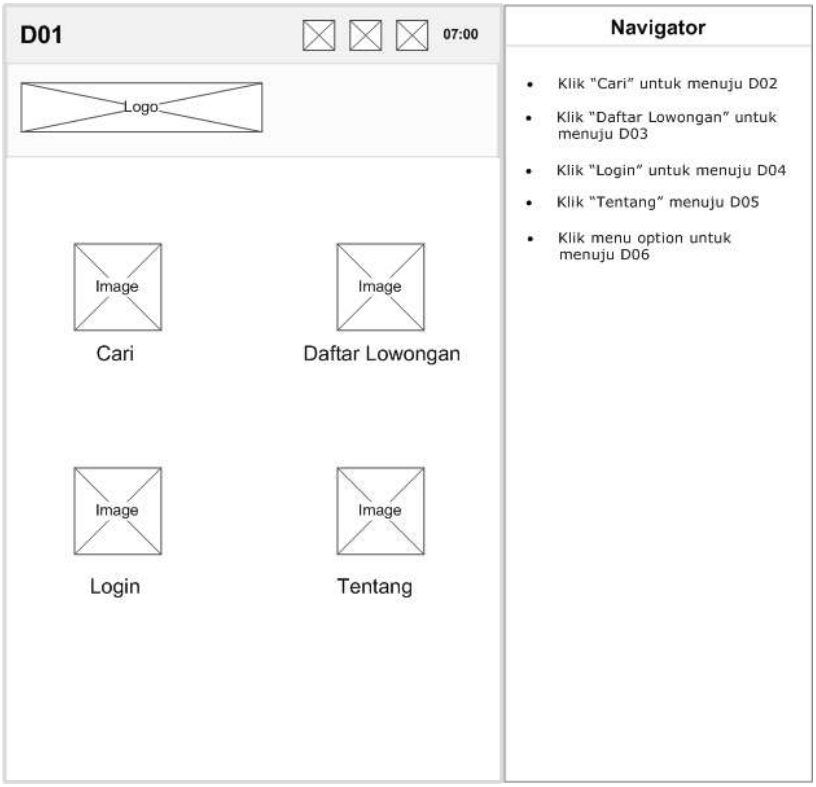
3.2.3 Perancangan Antarmuka

Interface atau antar muka merupakan tampilan dari suau program aplikasi yang berperan sebagai media komunikasi yang digunakan sebagai sarana berdialog antara program dengan *user*. Sistem yang akan dibangun diharapkan menyediakan *interface* yang mudah dipahami dan digunakan oleh *user*. Berikut ini adalah perancangan antarmuka pada sistem yang akan dibuat.

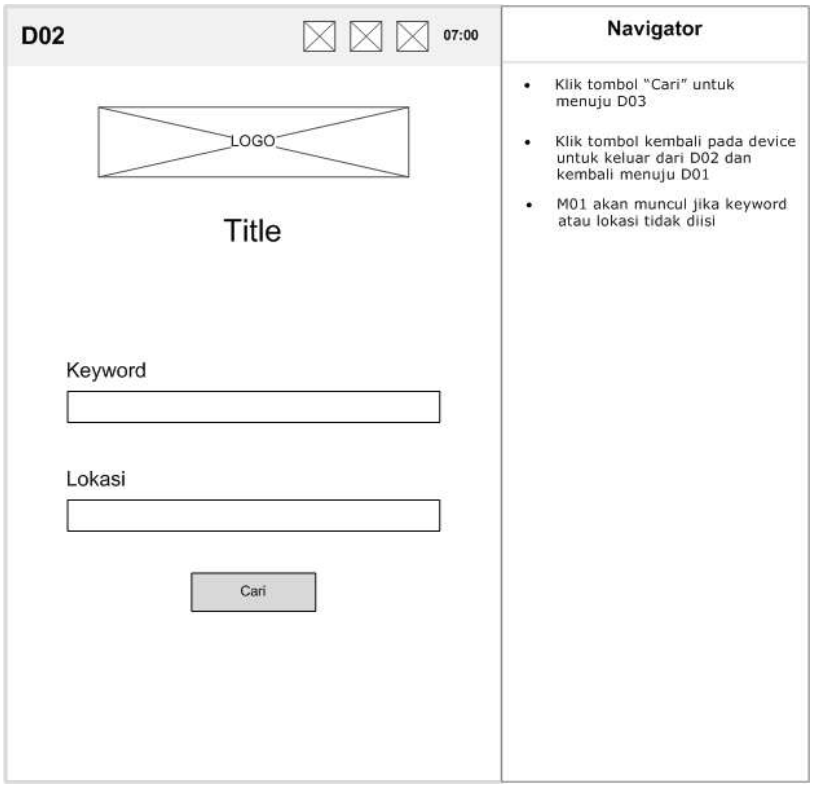
3.2.3.1 Perancangan Antarmuka Android



Gambar III. 122 Perancangan Antarmuka Splash Screen



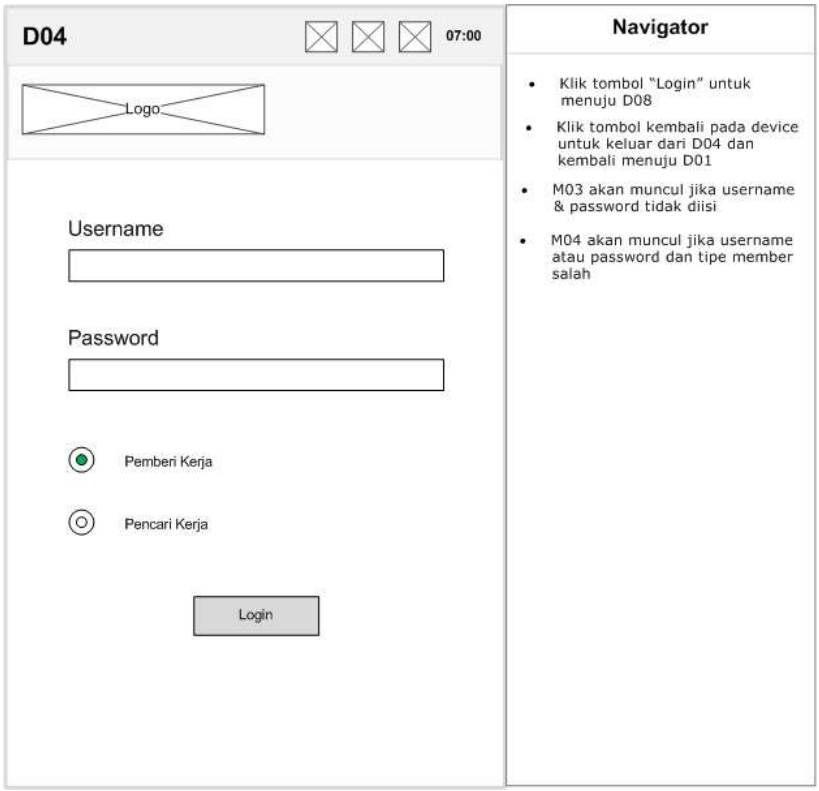
Gambar III. 123 Perancangan Antarmuka menu utama aplikasi android



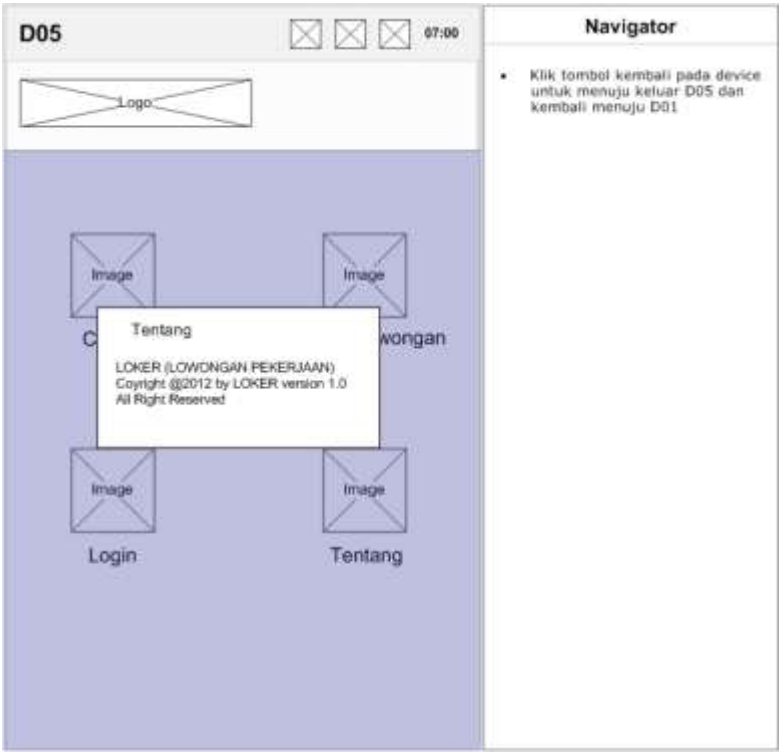
Gambar III. 124 Perancangan Antarmuka Pencarian Lowongan



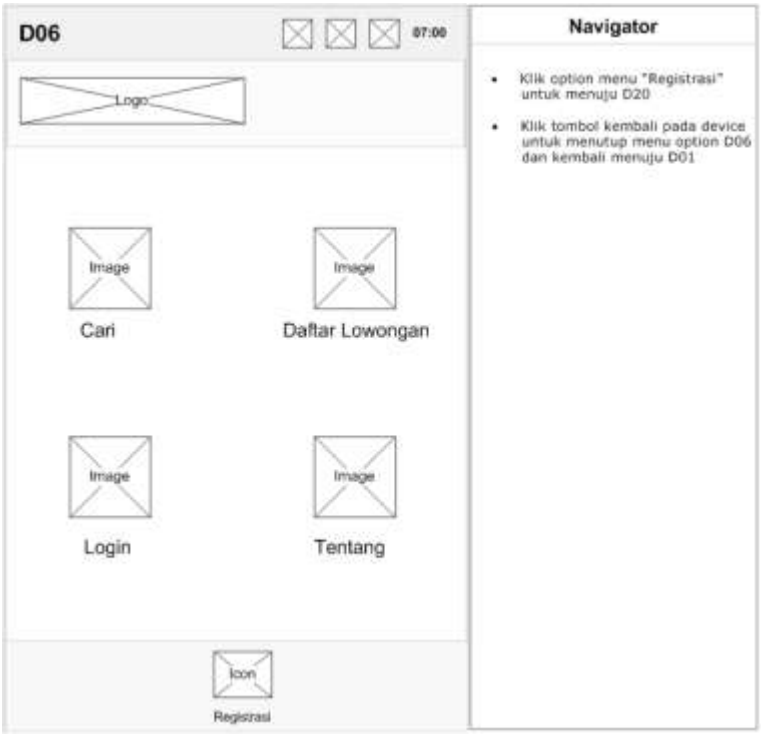
Gambar III. 125 Perancangan antarmuka Daftar Lowongan



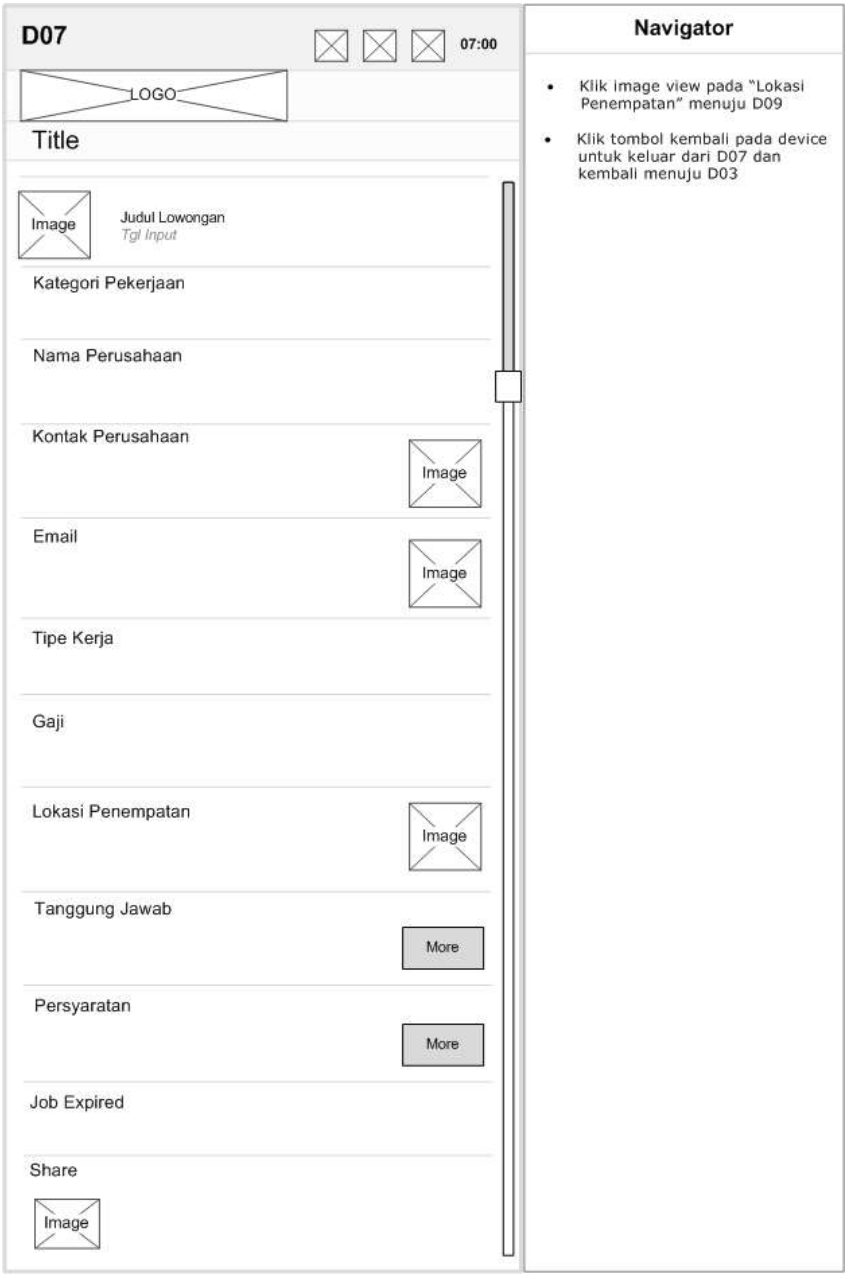
Gambar III. 126 Perancangan antarmuka menu Login member



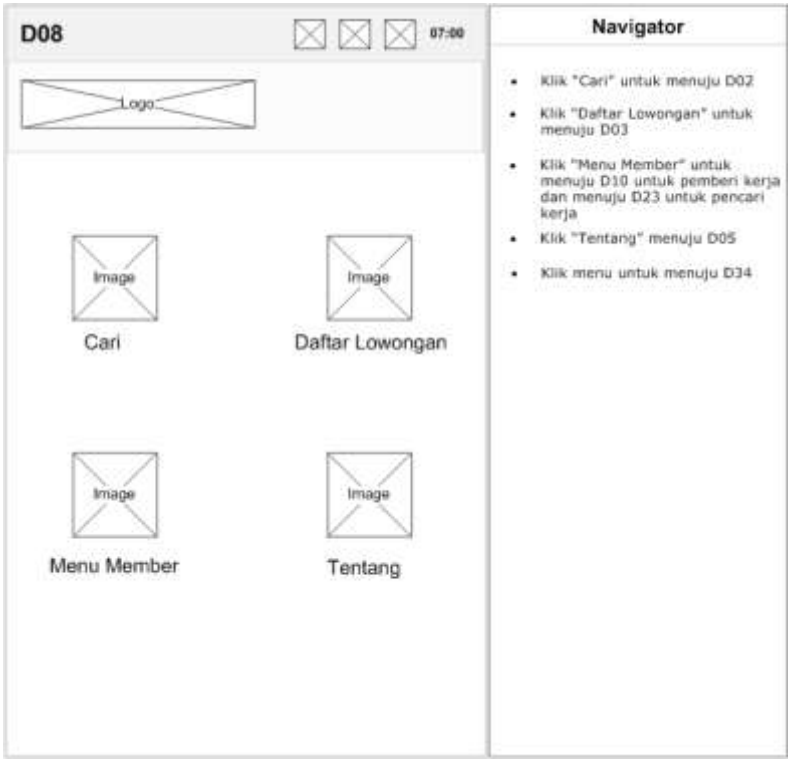
Gambar III. 127 Perancangan antarmuka menu Tentang



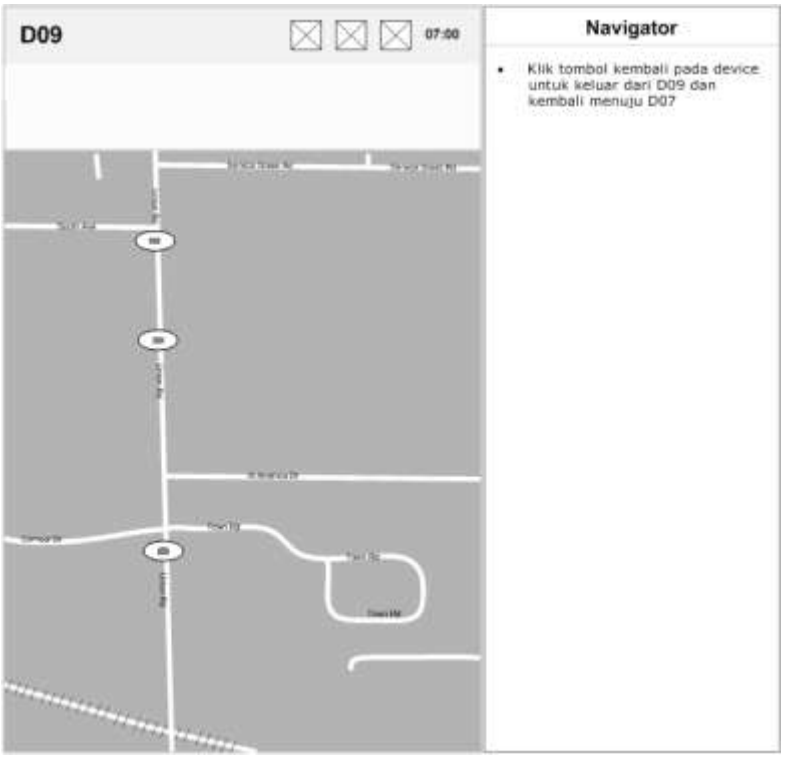
Gambar III. 128 Perancangan antarmuka menu Registrasi



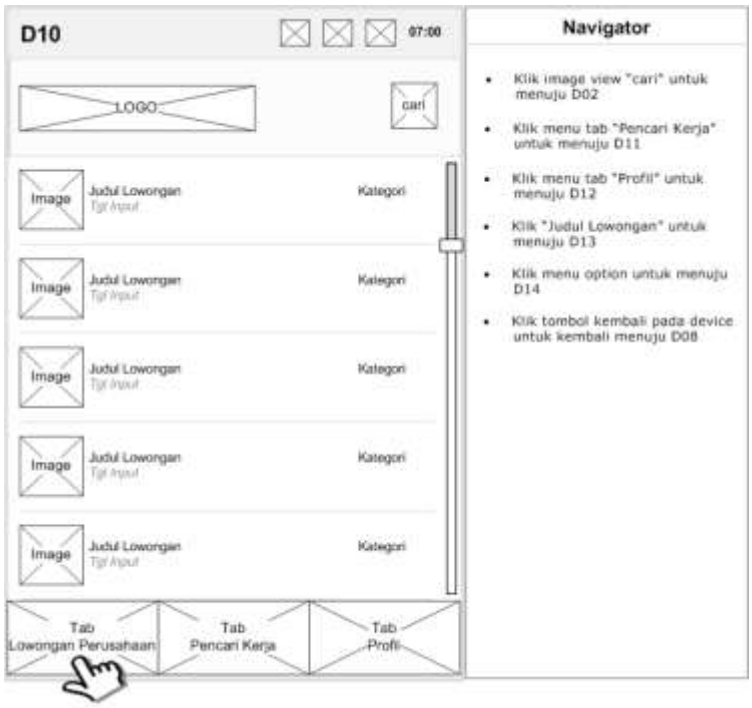
Gambar III. 129 Perancangan antarmuka Detil Lowongan



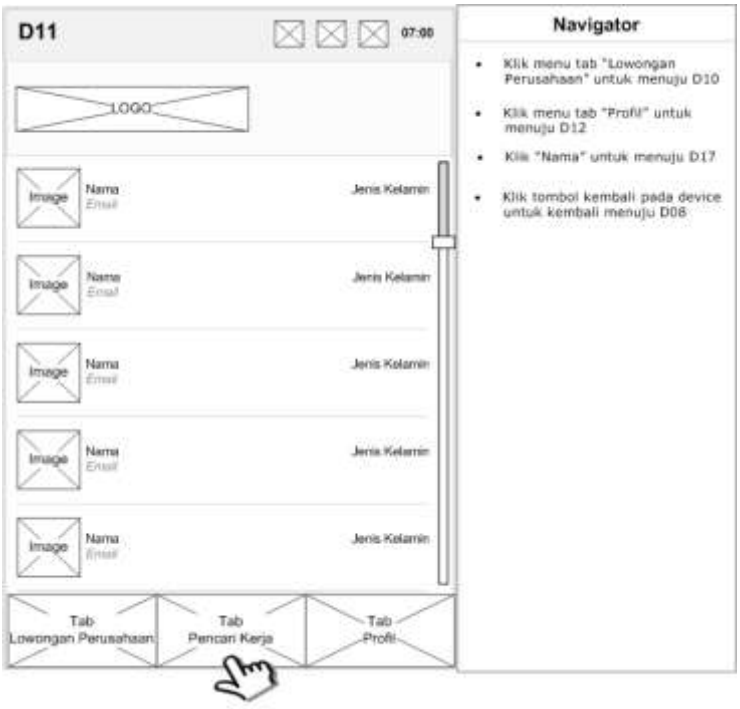
Gambar III. 130 Perancangan antarmuka Menu Utama Member



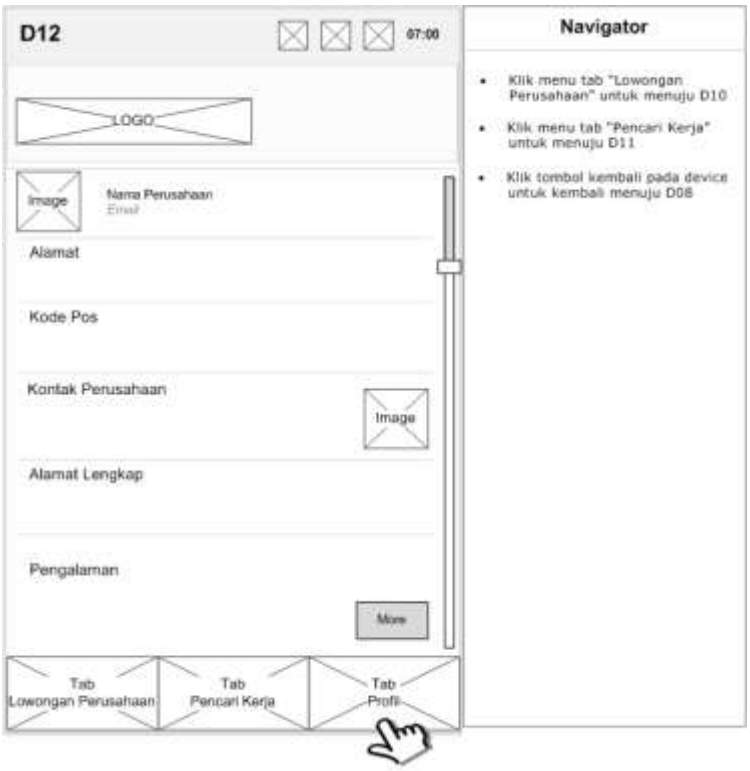
Gambar III. 131 Perancangan antarmuka halaman Lihat Peta



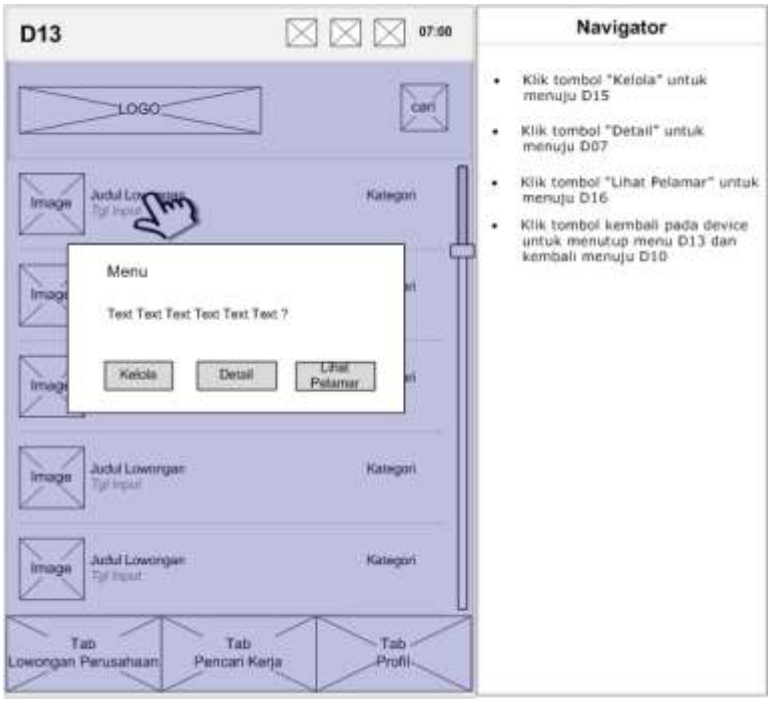
Gambar III. 132 Perancangan Antarmuka Halaman Member Pemberi Kerja Tab Lowongan Perusahaan



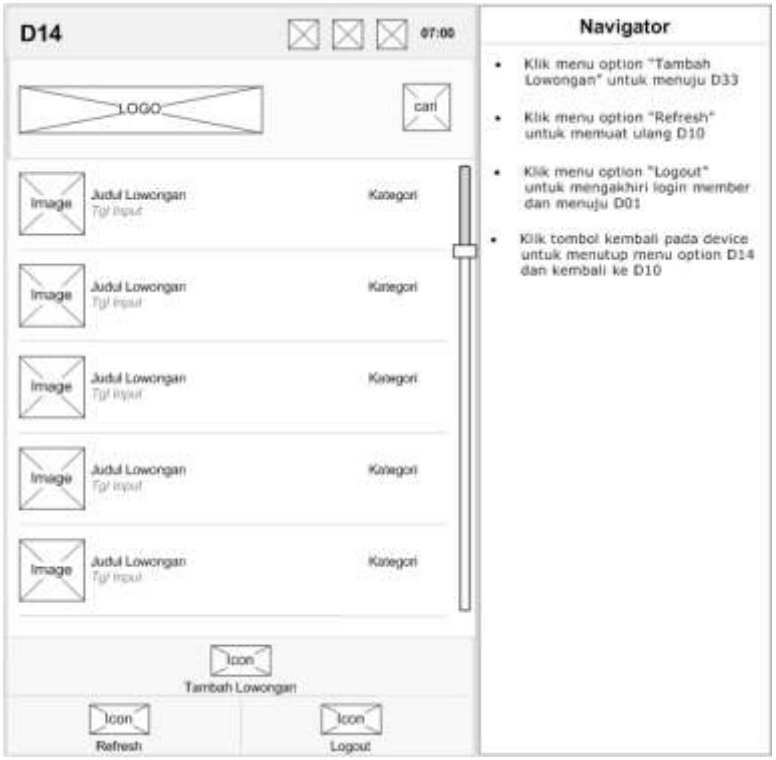
Gambar III. 133 Perancangan Antarmuka Halaman Member Pemberi Kerja Tab Pencari Kerja



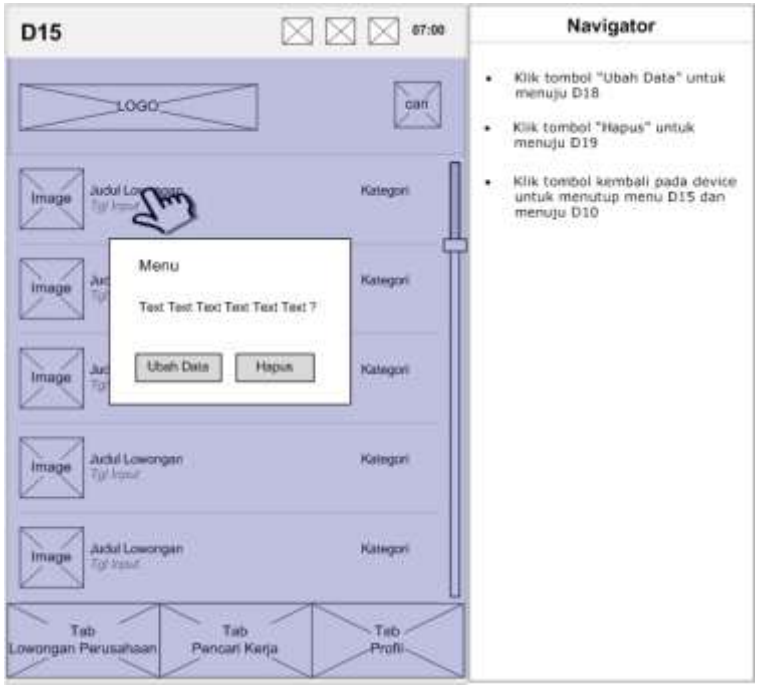
Gambar III. 134 Perancangan Antarmuka Halaman Member Pemberi Kerja Tab Profil



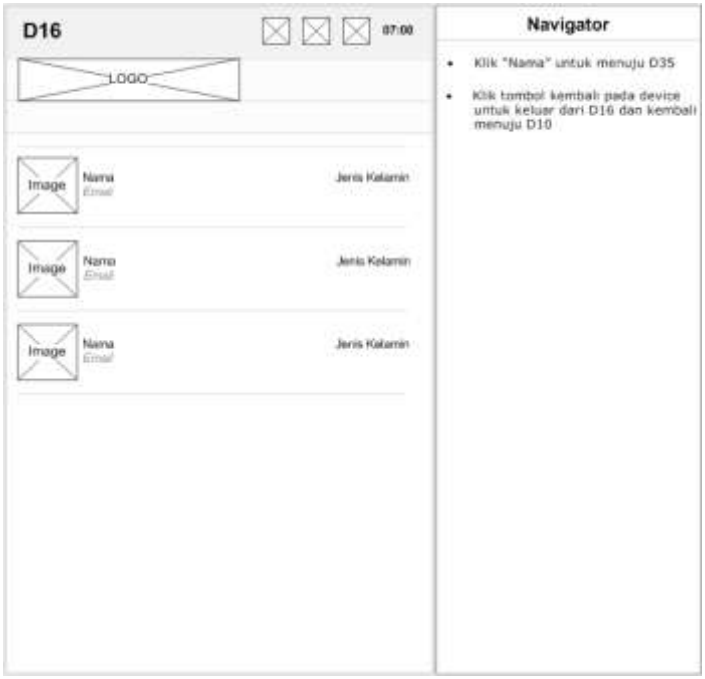
Gambar III. 135 Perancangan Antarmuka Halaman Member Pemberi Kerja Menu Lowongan



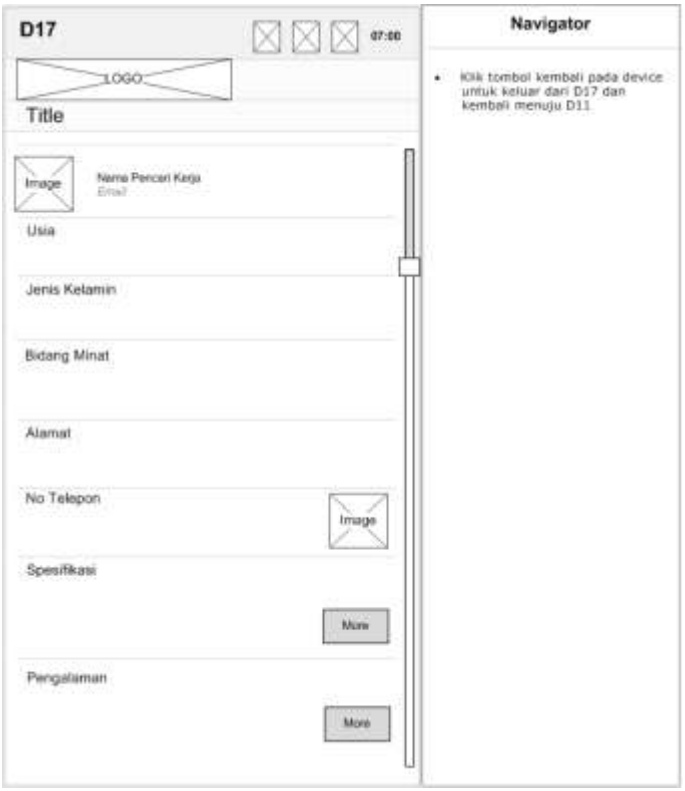
Gambar III. 136 Perancangan Antarmuka Halaman Menu Option Member Pemberi Kerja



Gambar III. 137 Perancangan Antarmuka Halaman Member Pemberi Kerja Menu Kelola Lowongan



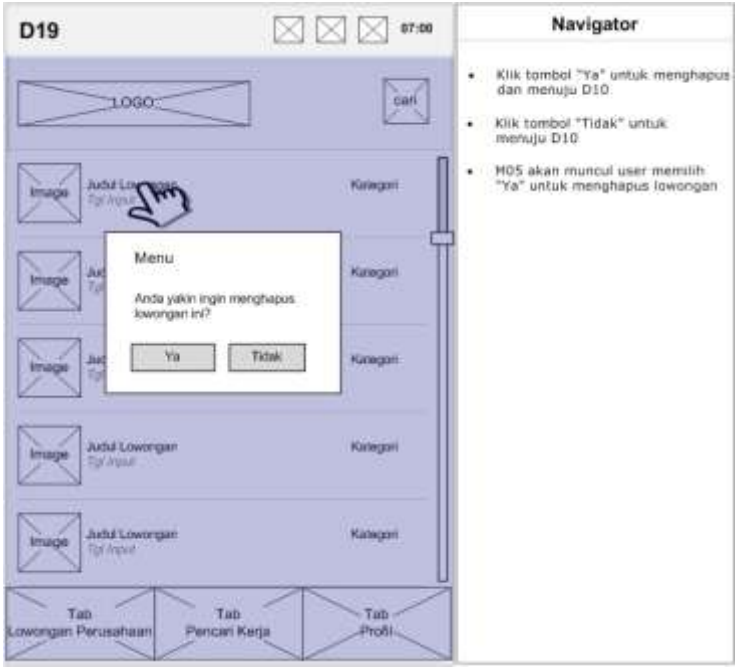
Gambar III. 138 Perancangan Antarmuka Halaman Daftar Pelamar Kerja



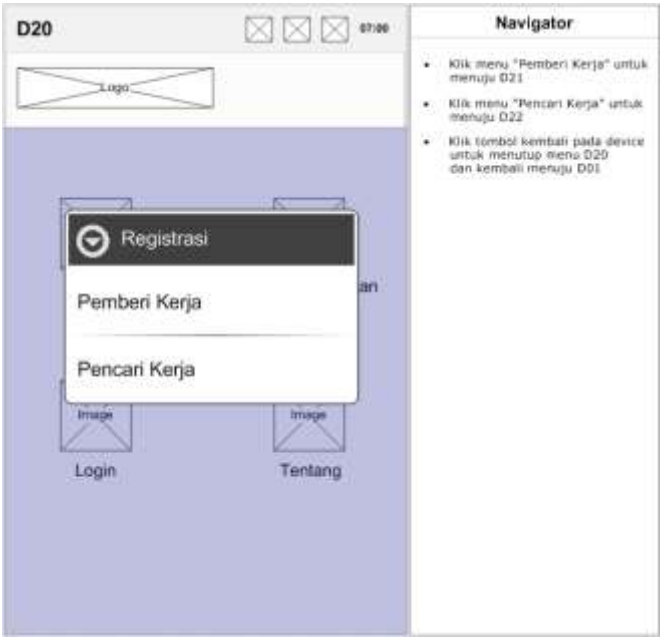
Gambar III. 139 Perancangan Antarmuka Halaman Detail Pelamar atau Pencari Kerja



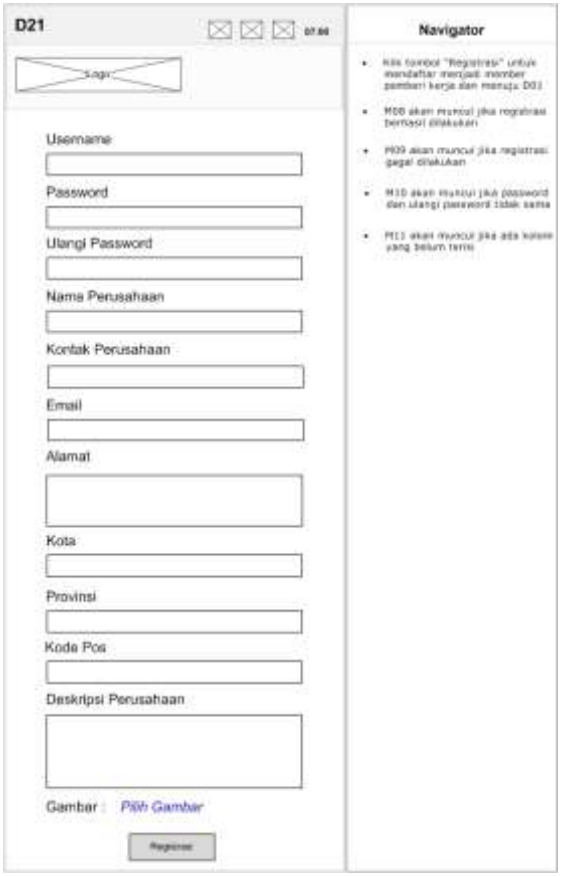
Gambar III. 140 Perancangan Antarmuka Halaman Ubah Data Lowongan



Gambar III. 141 Perancangan Antarmuka Menu Konfirmasi Hapus Lowongan



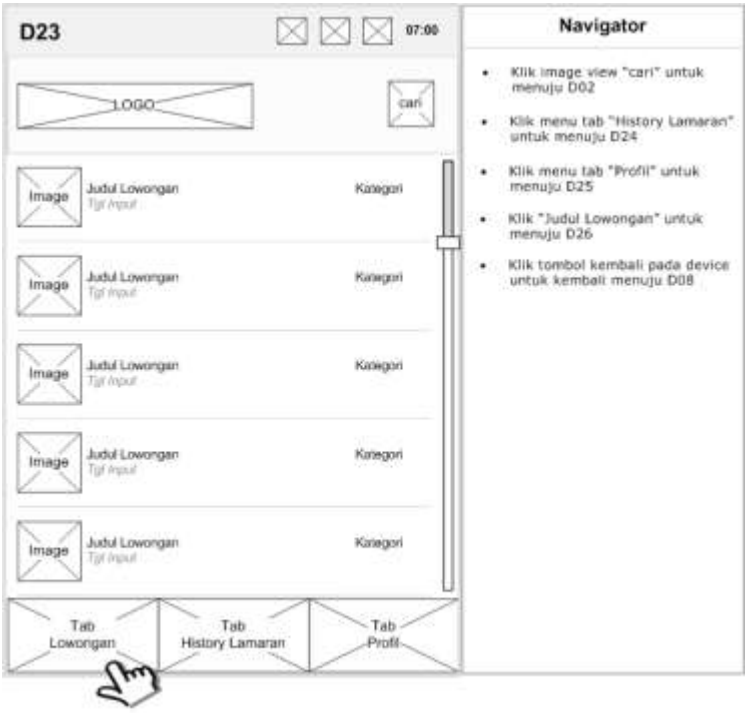
Gambar III. 142 Perancangan Antarmuka Menu Registrasi



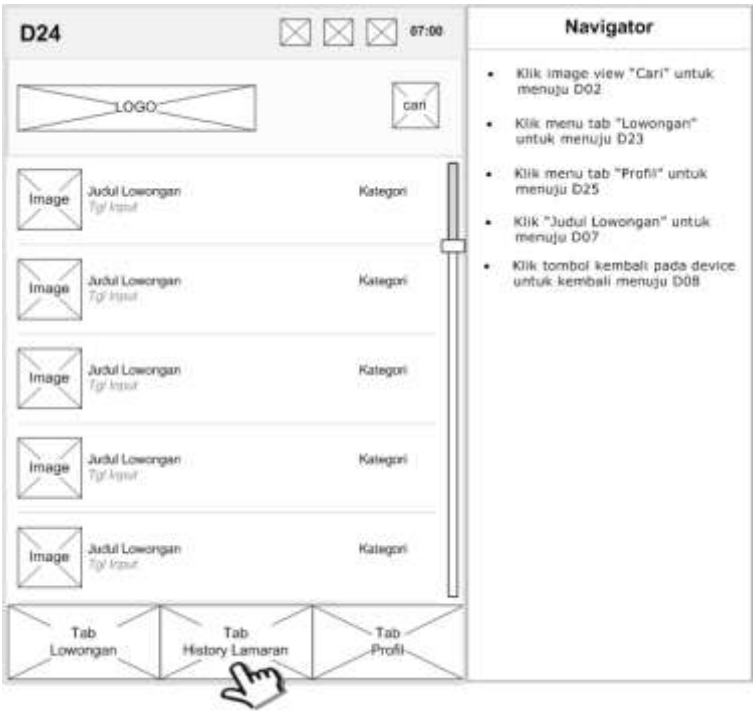
Gambar III. 143 Perancangan Antarmuka Halaman Registrasi Member Pemberi Kerja



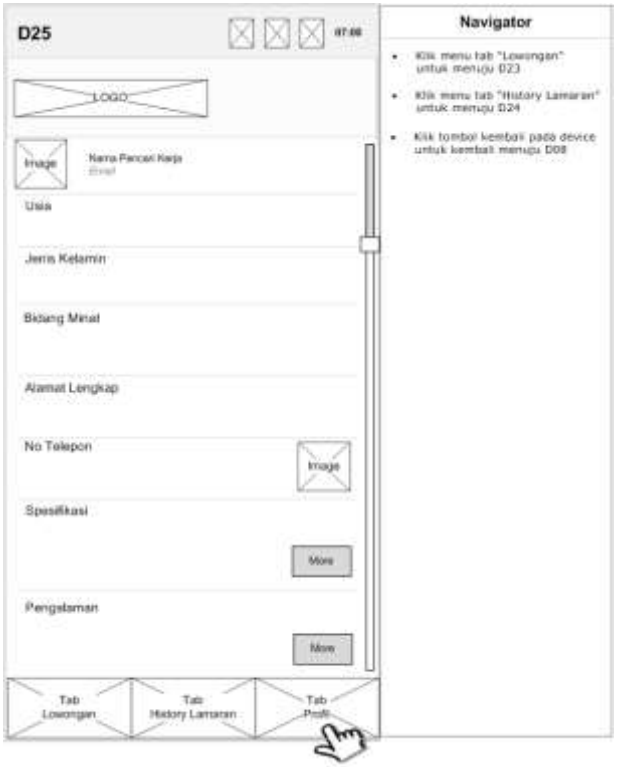
Gambar III. 144 Perancangan Antarmuka Halaman Registrasi Member Pencari Kerja



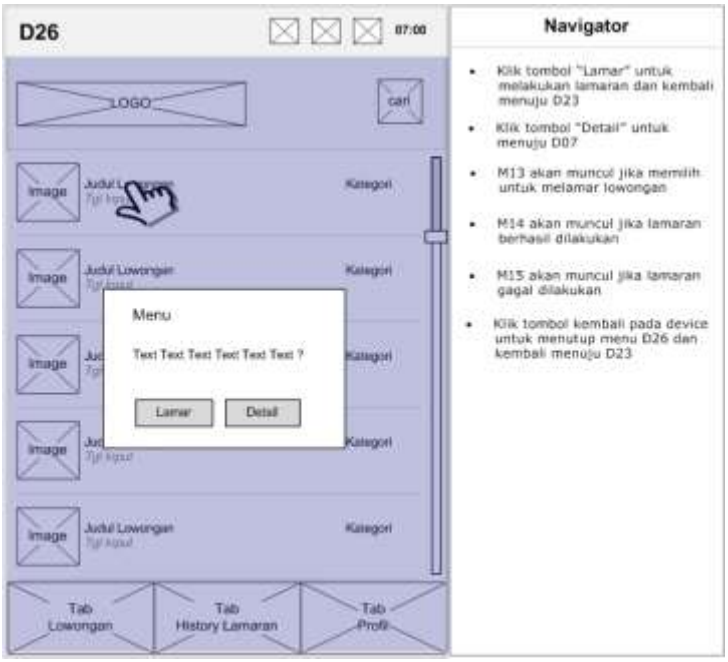
Gambar III. 145 Perancangan Antarmuka Halaman Member Pencari Kerja Tab Lowongan



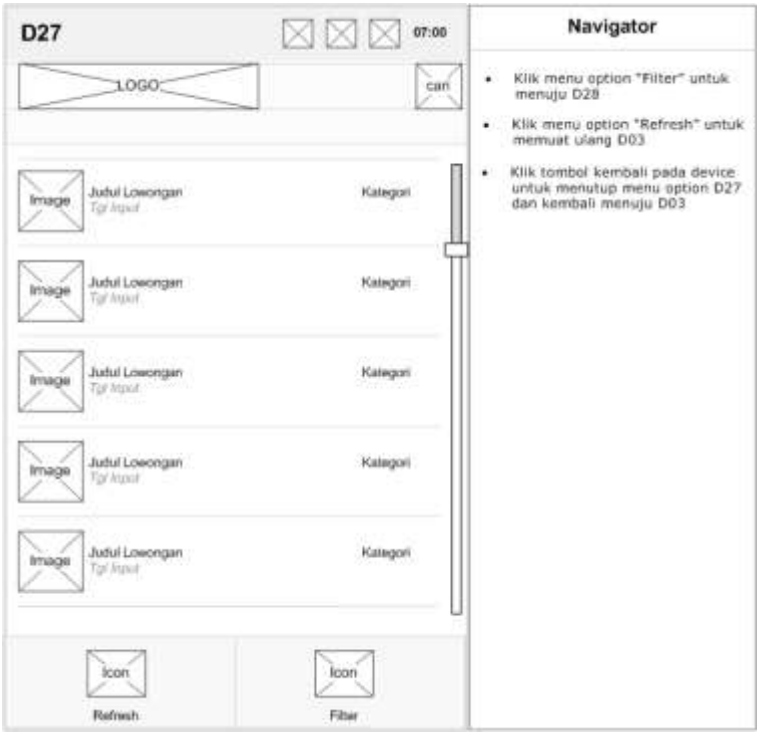
Gambar III. 146 Perancangan Antarmuka Halaman Member Pencari Kerja Tab History Lowongan



Gambar III. 147 Perancangan Antarmuka Halaman Member Pencari Kerja Tab Profil



Gambar III. 148 Perancangan Antarmuka Halaman Member Pencari Kerja Menu Lowongan



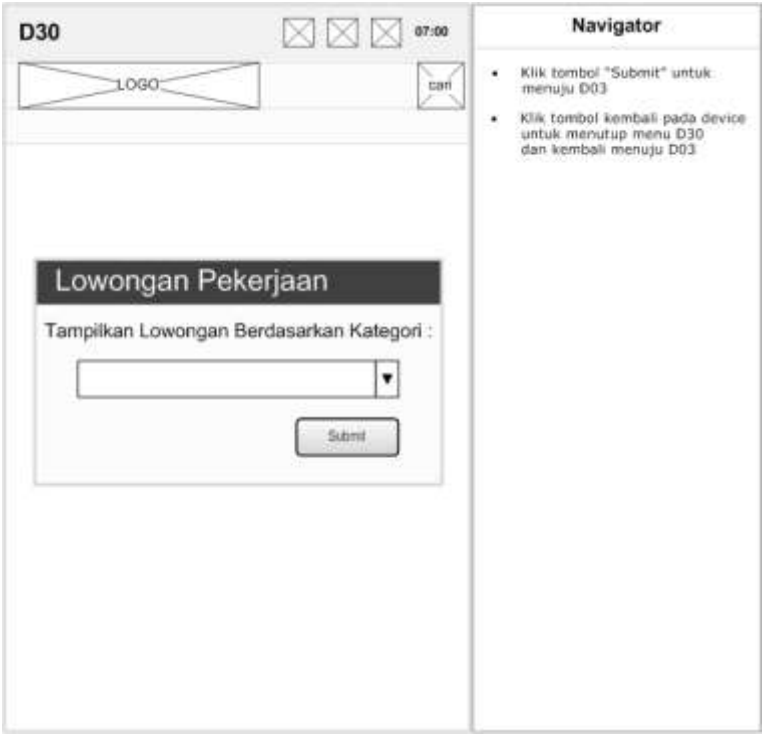
Gambar III. 149 Perancangan Antarmuka Menu Option Halaman Daftar Lowongan



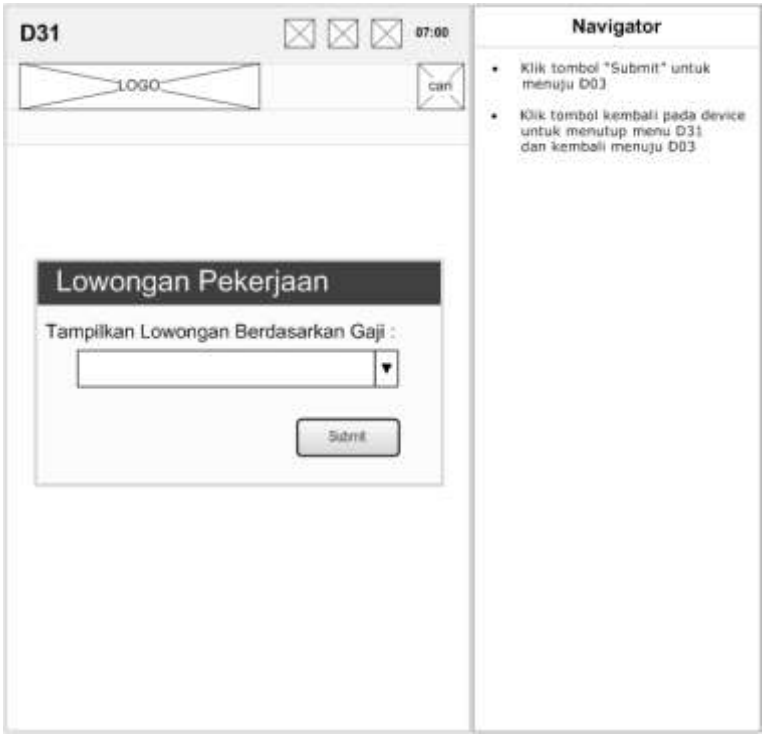
Gambar III. 150 Perancangan Antarmuka Menu Filter Lowongan



Gambar III. 151 Perancangan Antarmuka Menu Filter Lowongan Berdasarkan Lokasi



Gambar III. 152 Perancangan Antarmuka Menu Filter Lowongan Berdasarkan Kategori



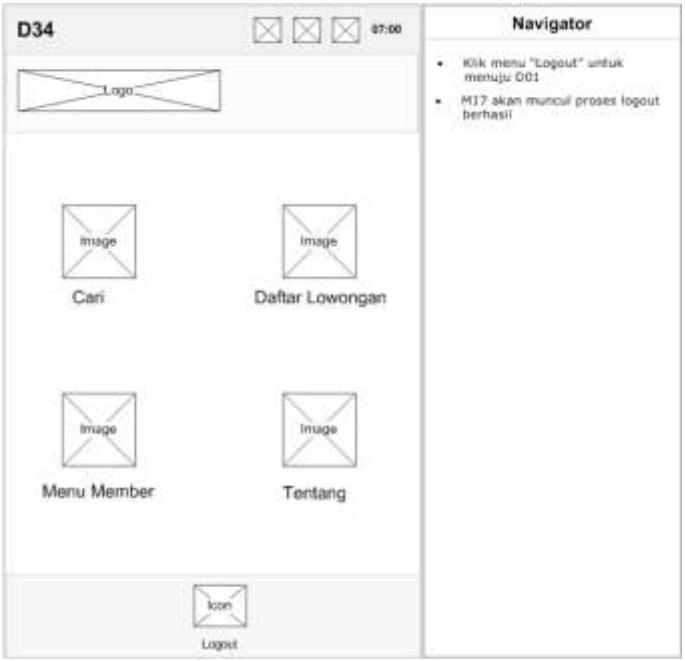
Gambar III. 153 Perancangan Antarmuka Menu Filter Lowongan Berdasarkan Gaji



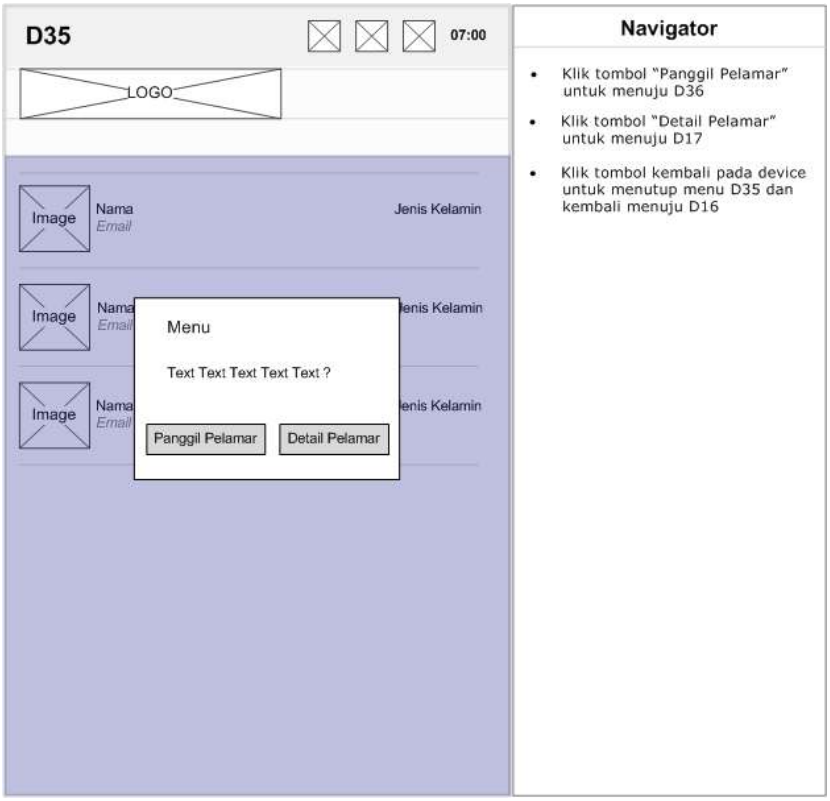
Gambar III. 154 Perancangan Antarmuka Menu Filter Lowongan Berdasarkan Tipe Kerja



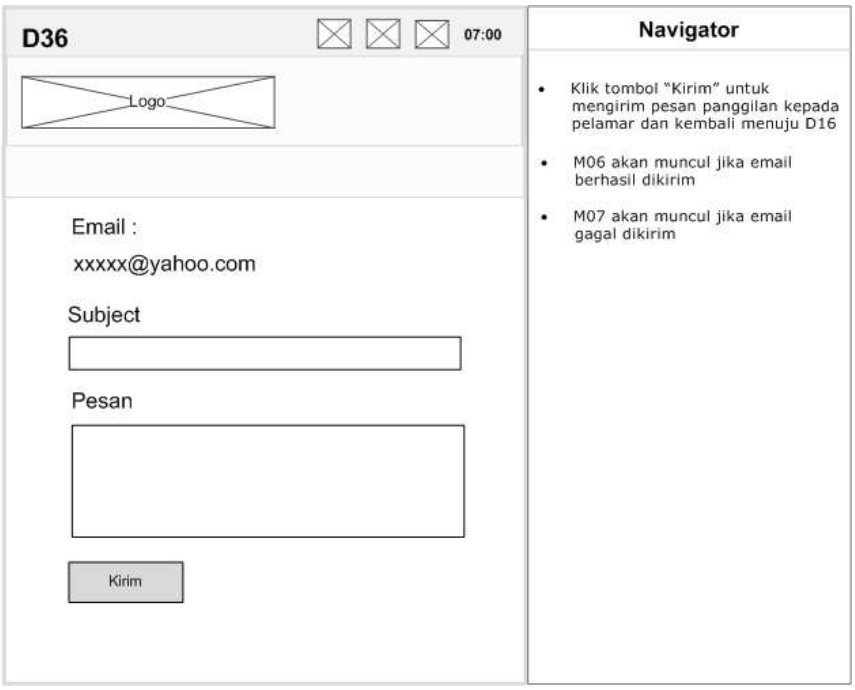
Gambar III. 155 Perancangan Antarmuka Halaman Tambah Lowongan



Gambar III. 156 Perancangan Antarmuka Menu Logout



Gambar III. 157 Perancangan Antarmuka Menu Pelamar

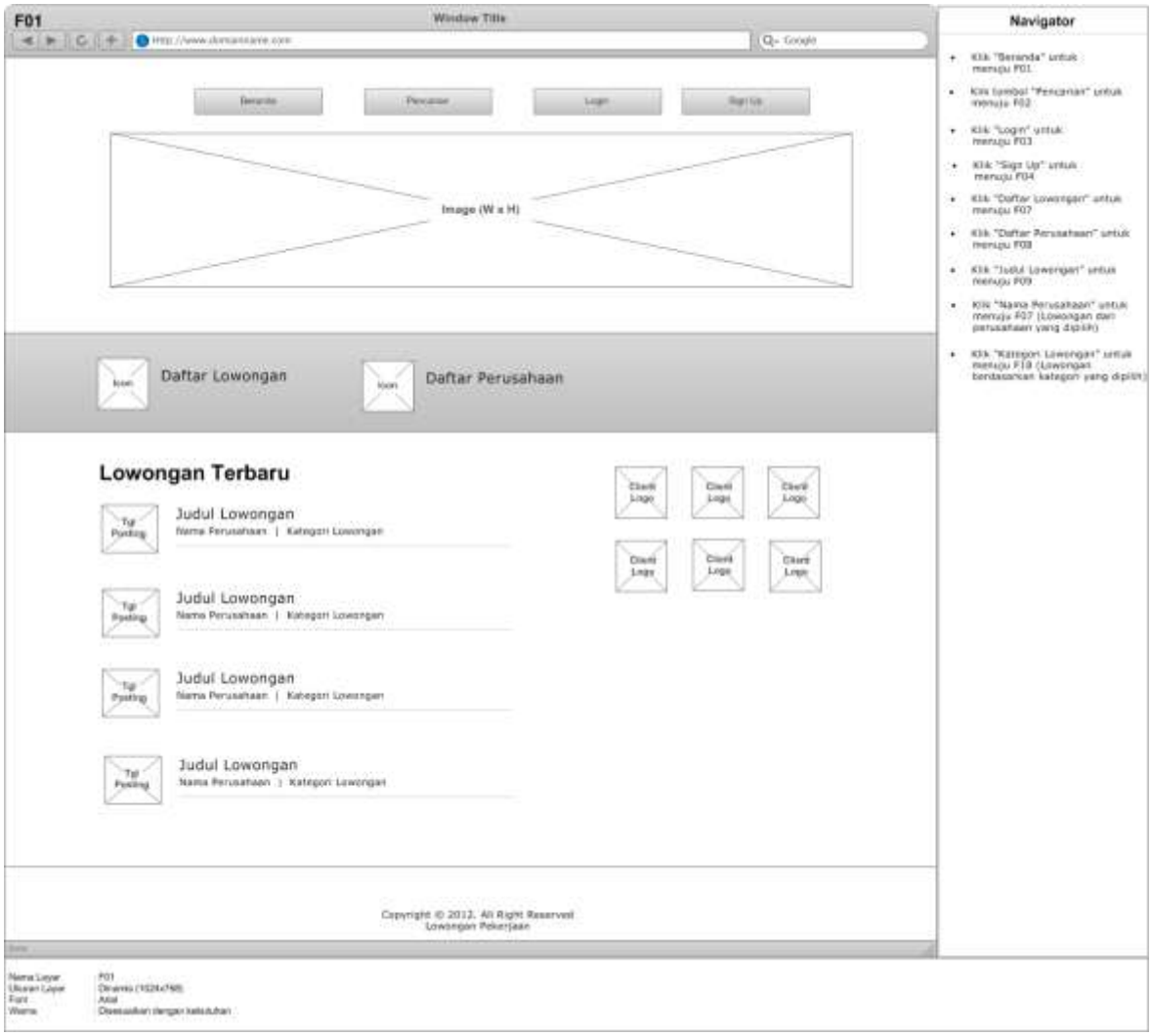


Gambar III. 158 Perancangan Antarmuka Halaman Panggil Pelamar

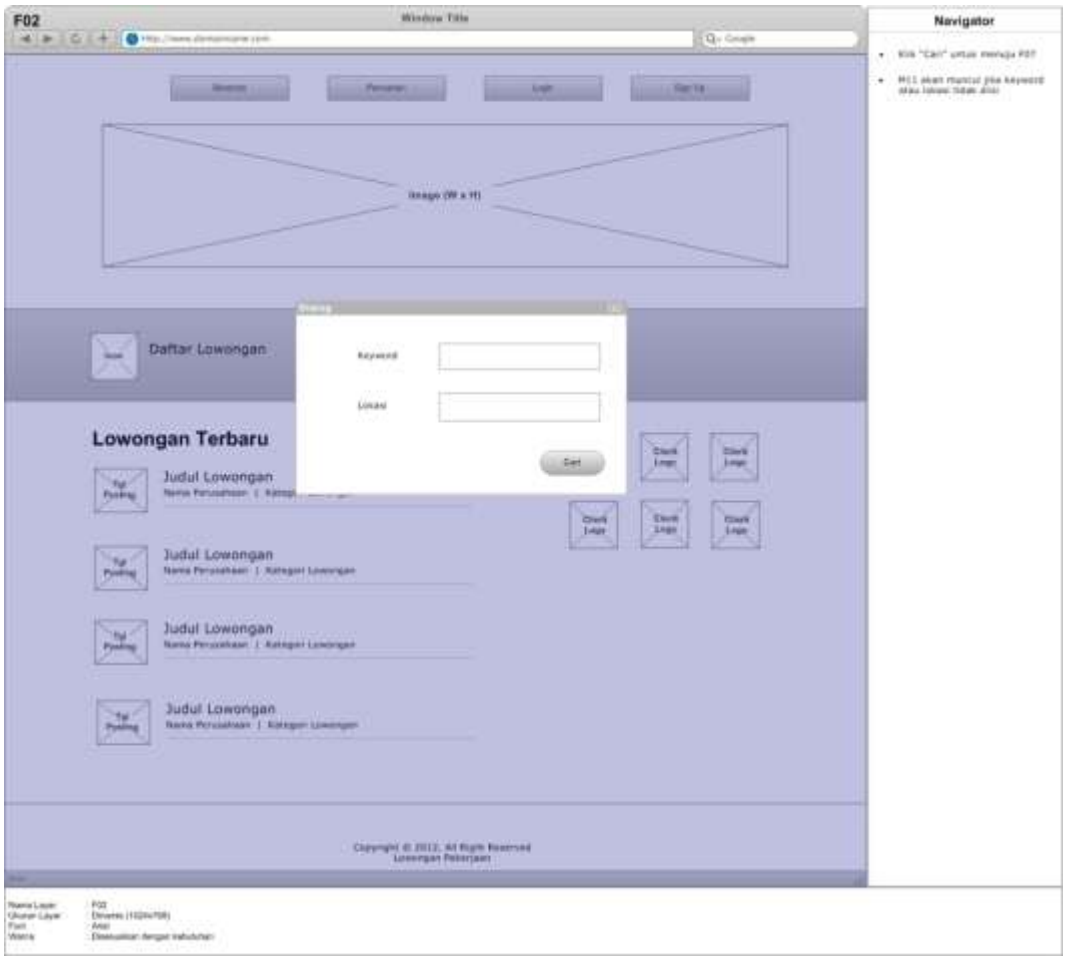


Gambar III. 159 Perancangan Pesan Aplikasi Android

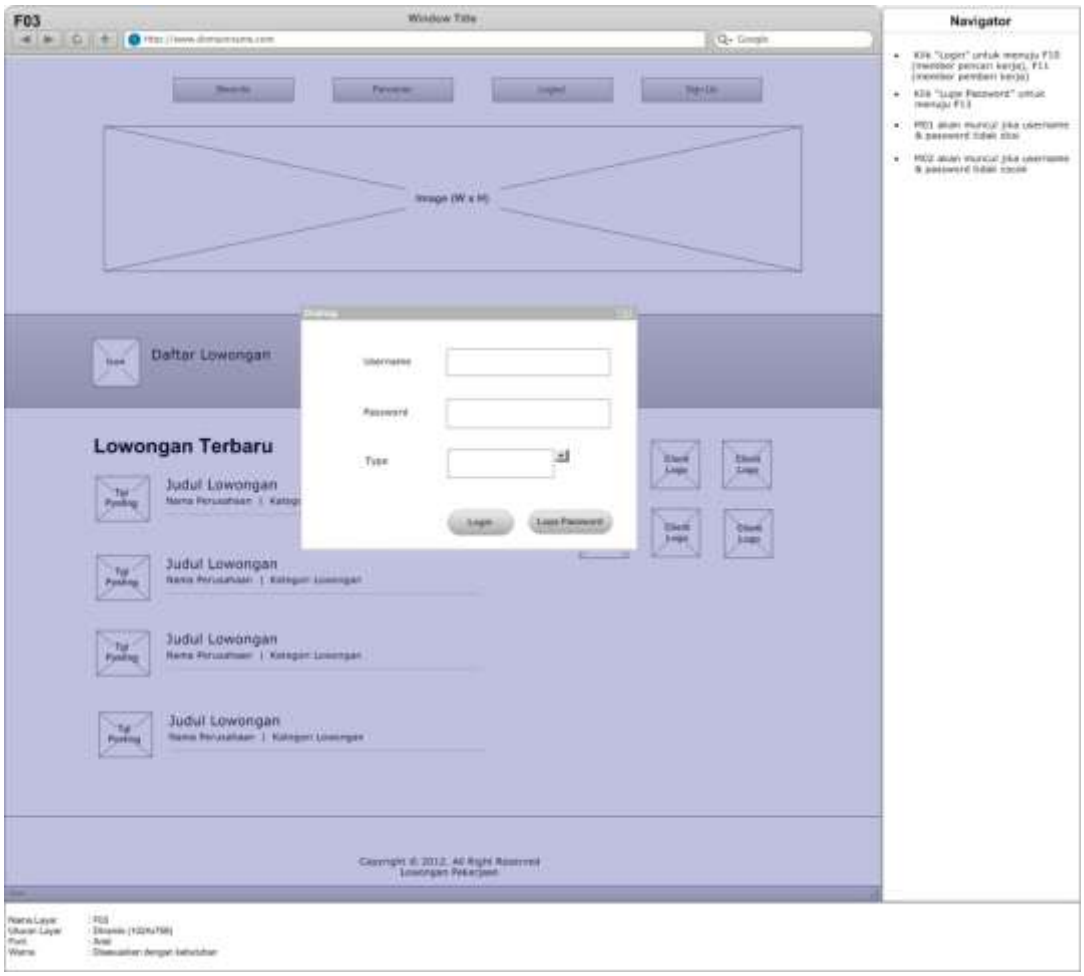
3.2.3.2 Perancangan Antarmuka Web User



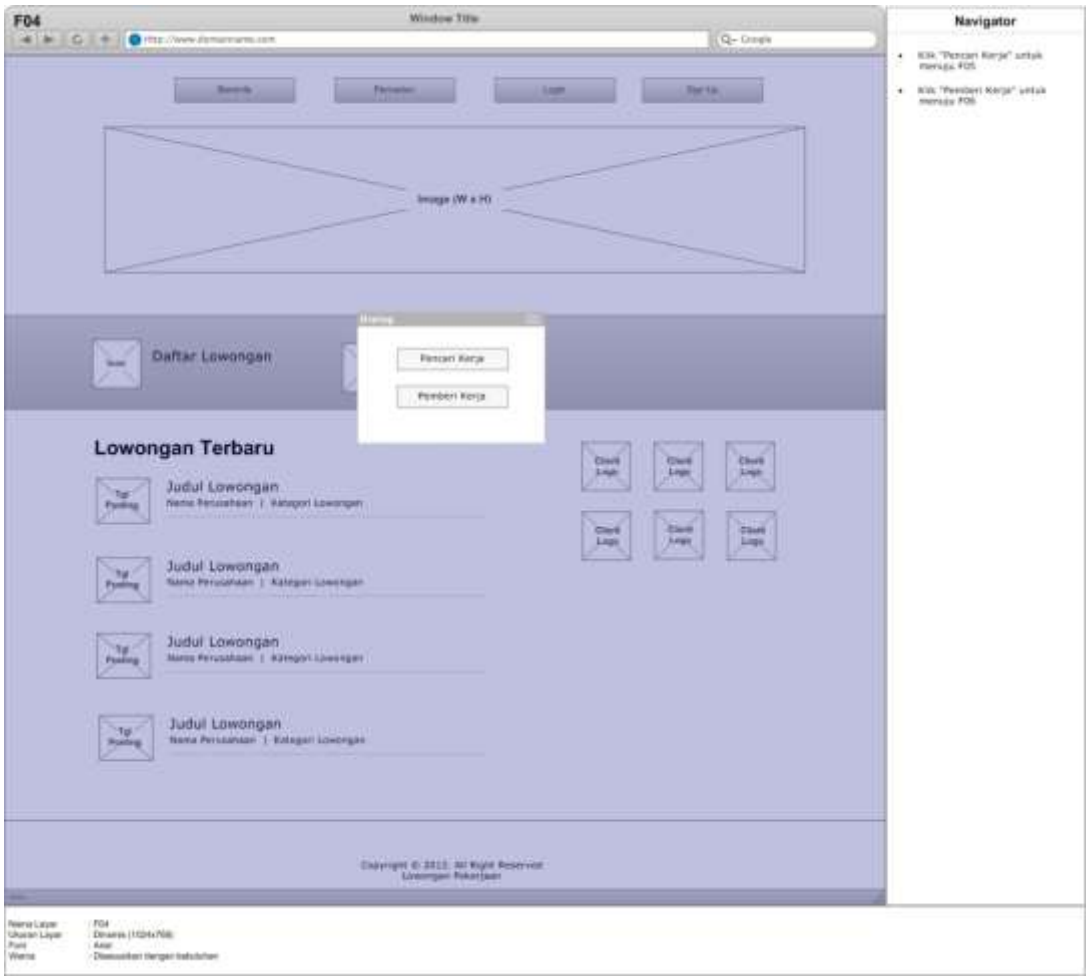
Gambar III. 160 Perancangan halaman menu utama (Beranda)



Gambar III. 161 Perancangan menu pencarian



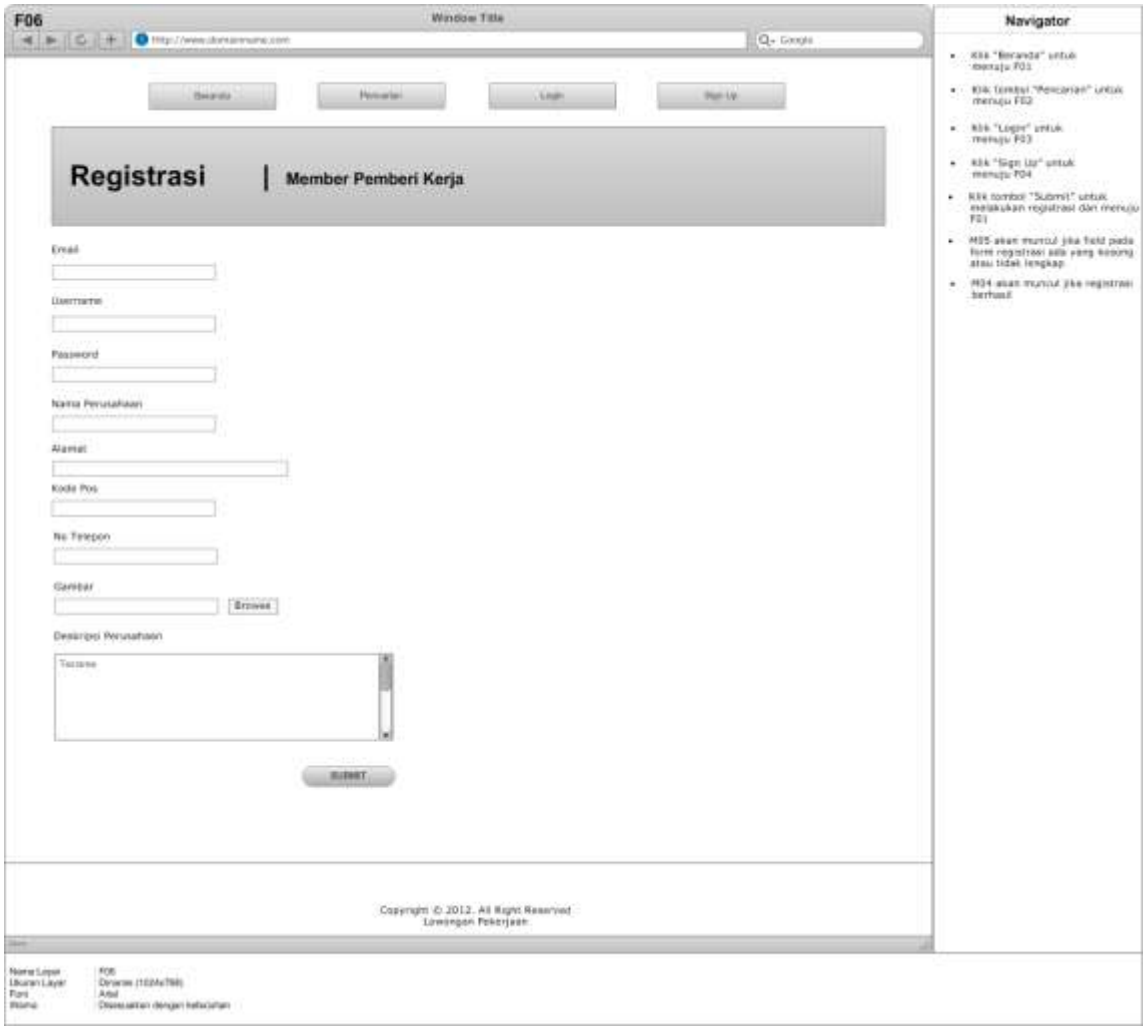
Gambar III. 162 Perancangan menu login



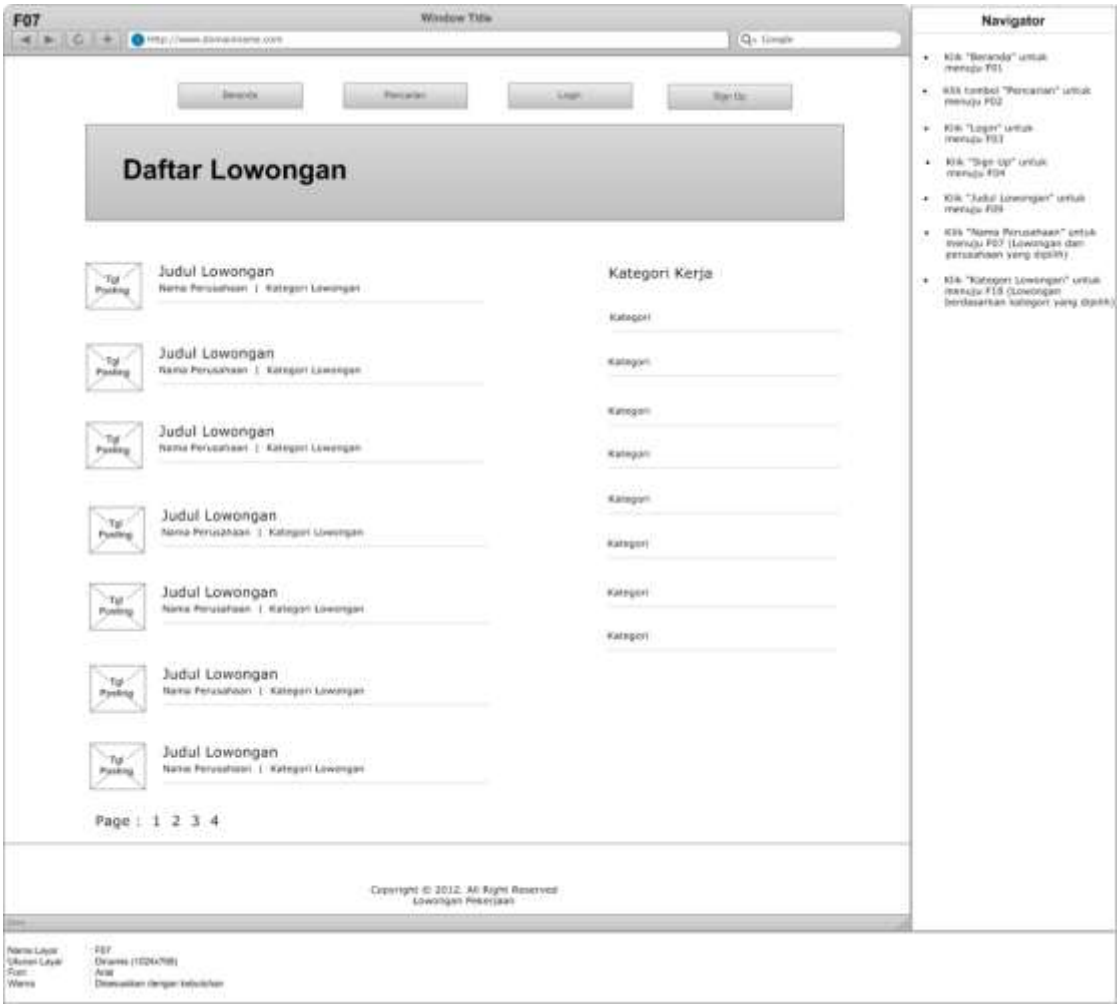
Gambar III. 163 Perancangan menu register



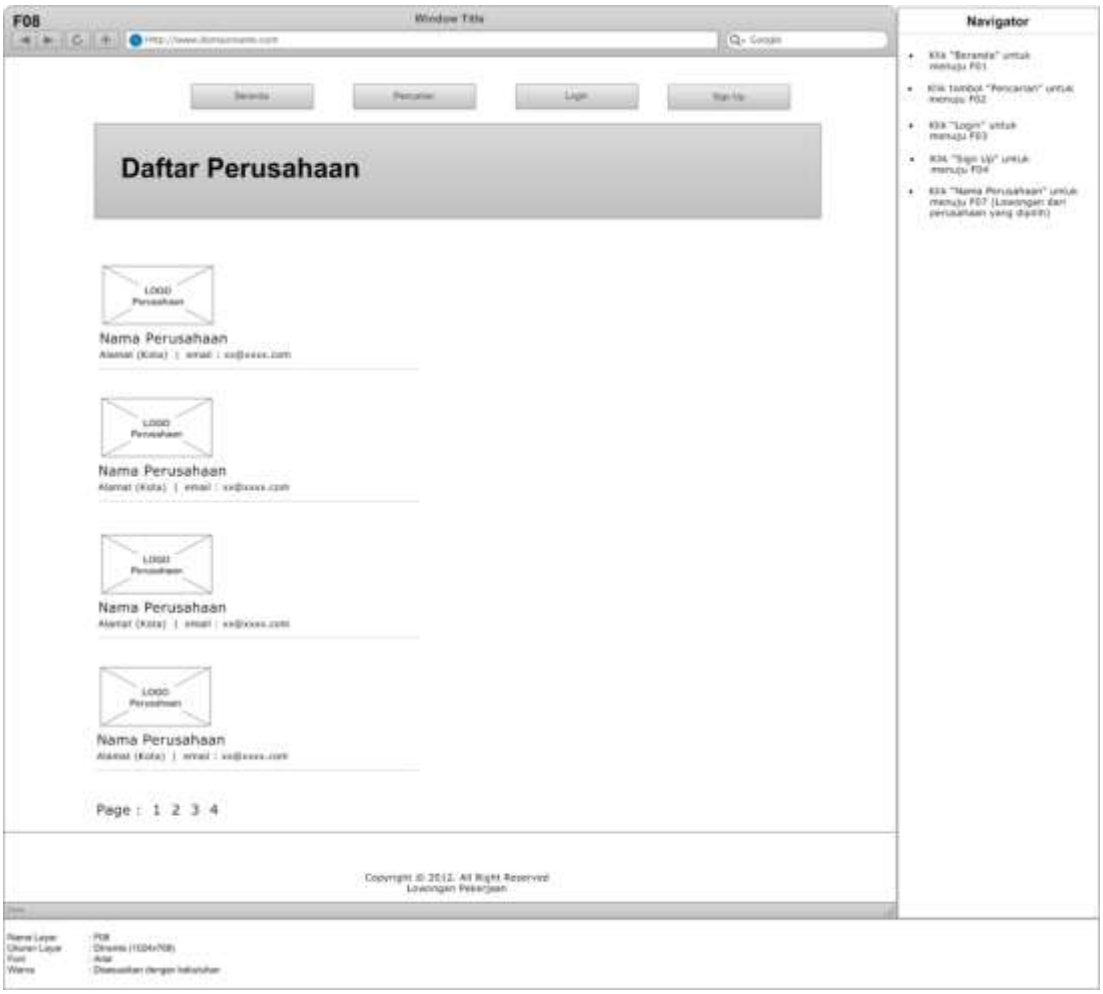
Gambar III. 164 Perancangan halaman Registrasi member pencari kerja



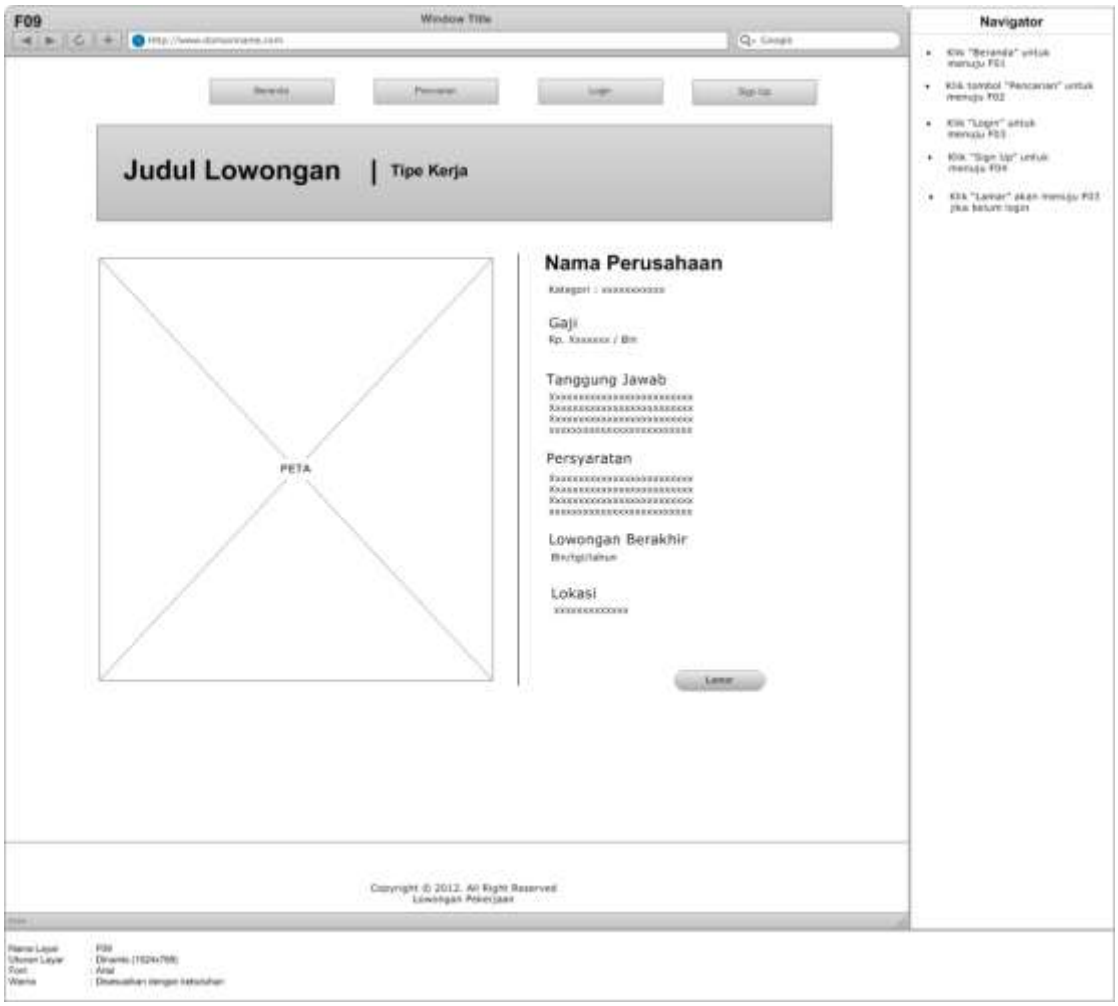
Gambar III. 165 Perancangan halaman Registrasi member pemberi kerja



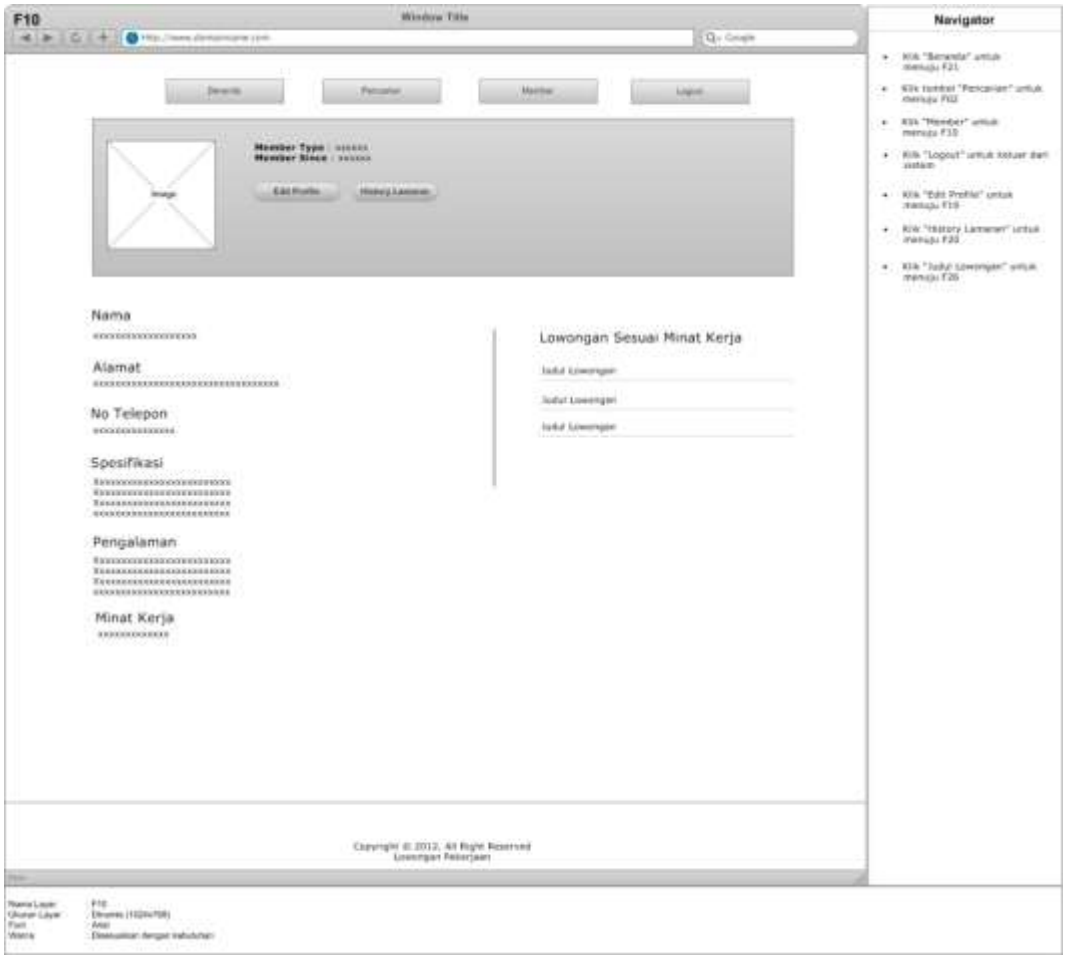
Gambar III. 166 Perancangan halaman Daftar Lowongan Pekerjaan



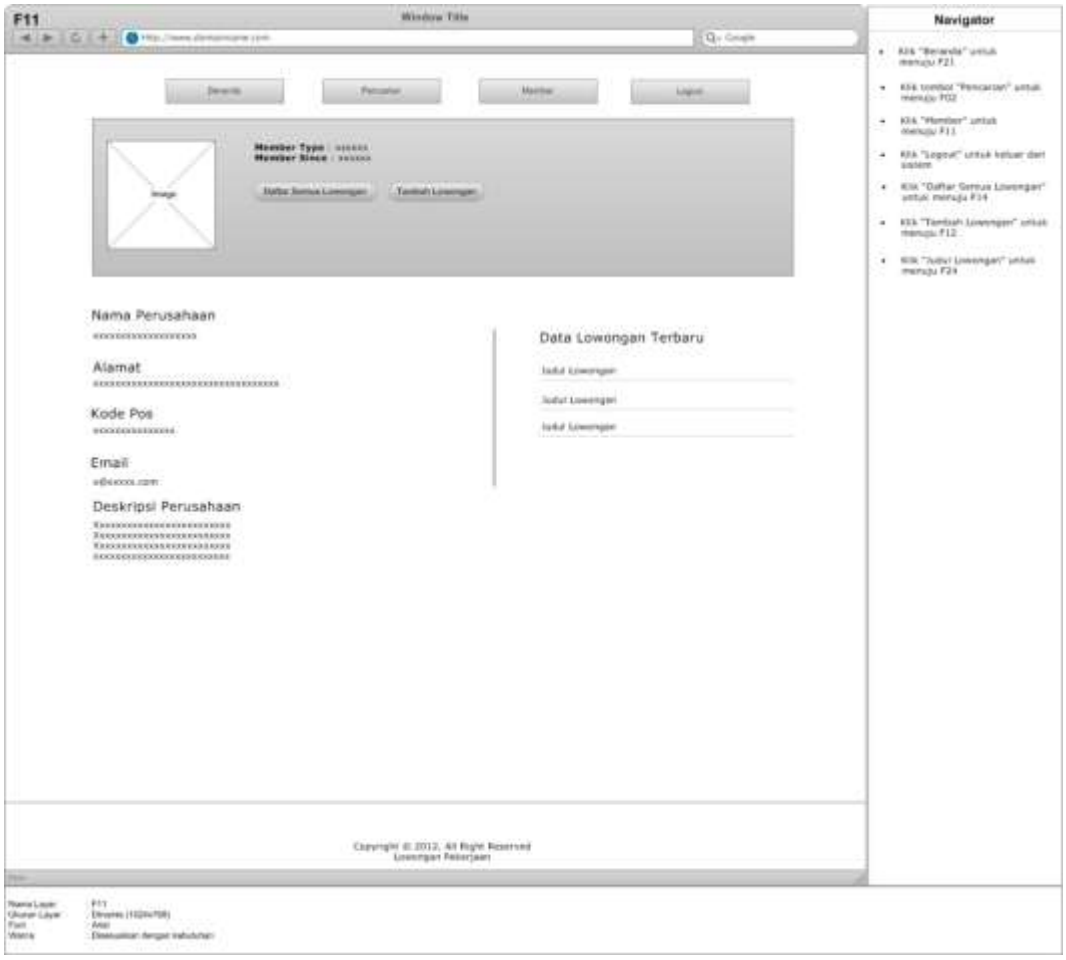
Gambar III. 167 Perancangan halaman Daftar perusahaan



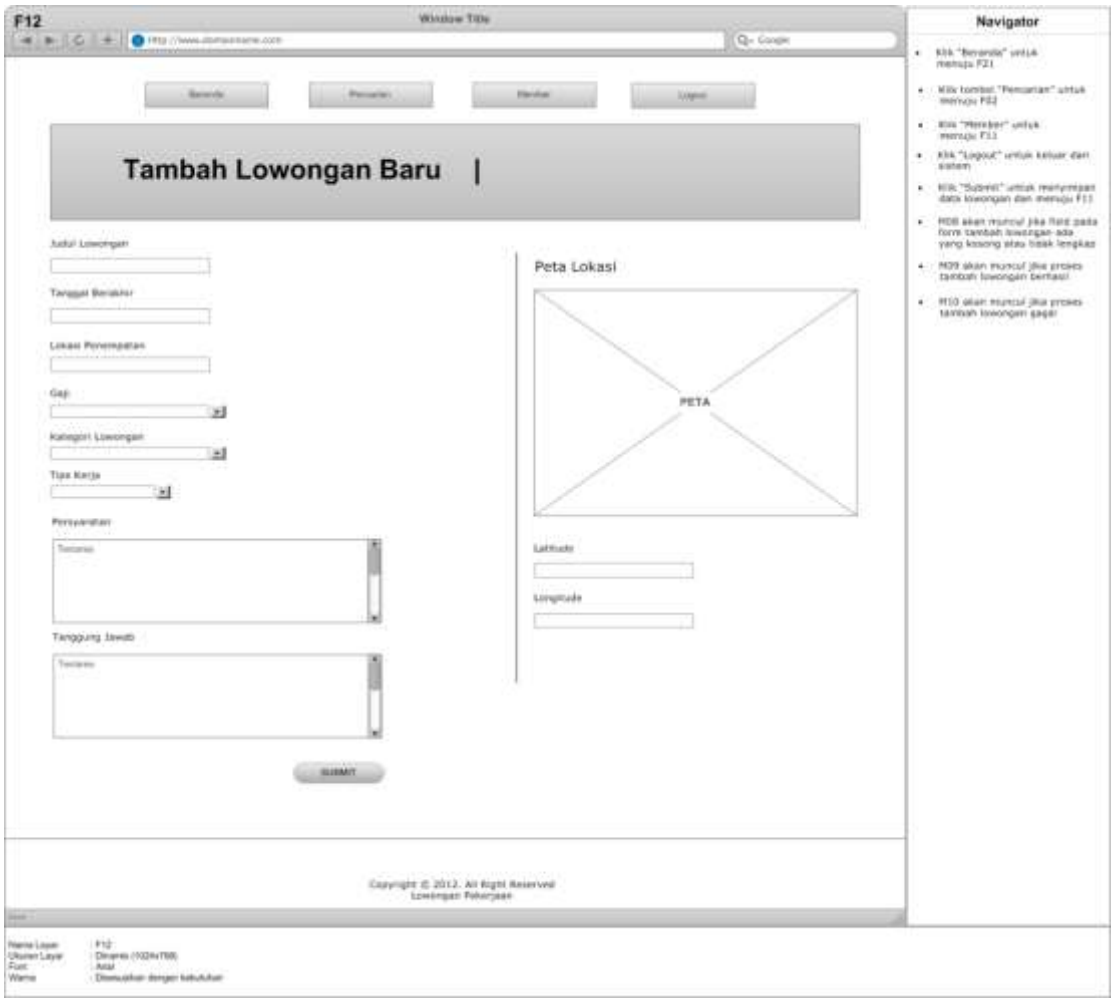
Gambar III. 168 Perancangan halaman Detil lowongan pekerjaan



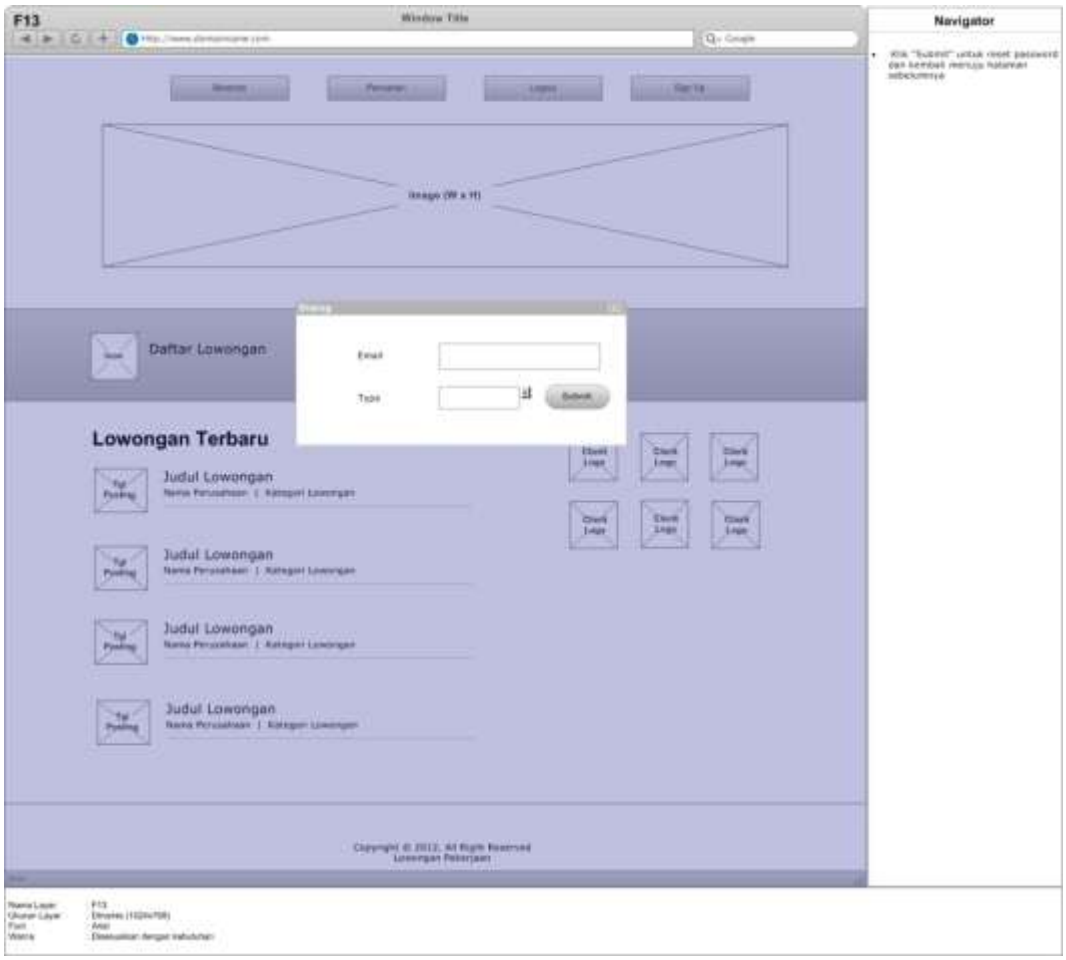
Gambar III. 169 Perancangan halaman Member Pencari kerja



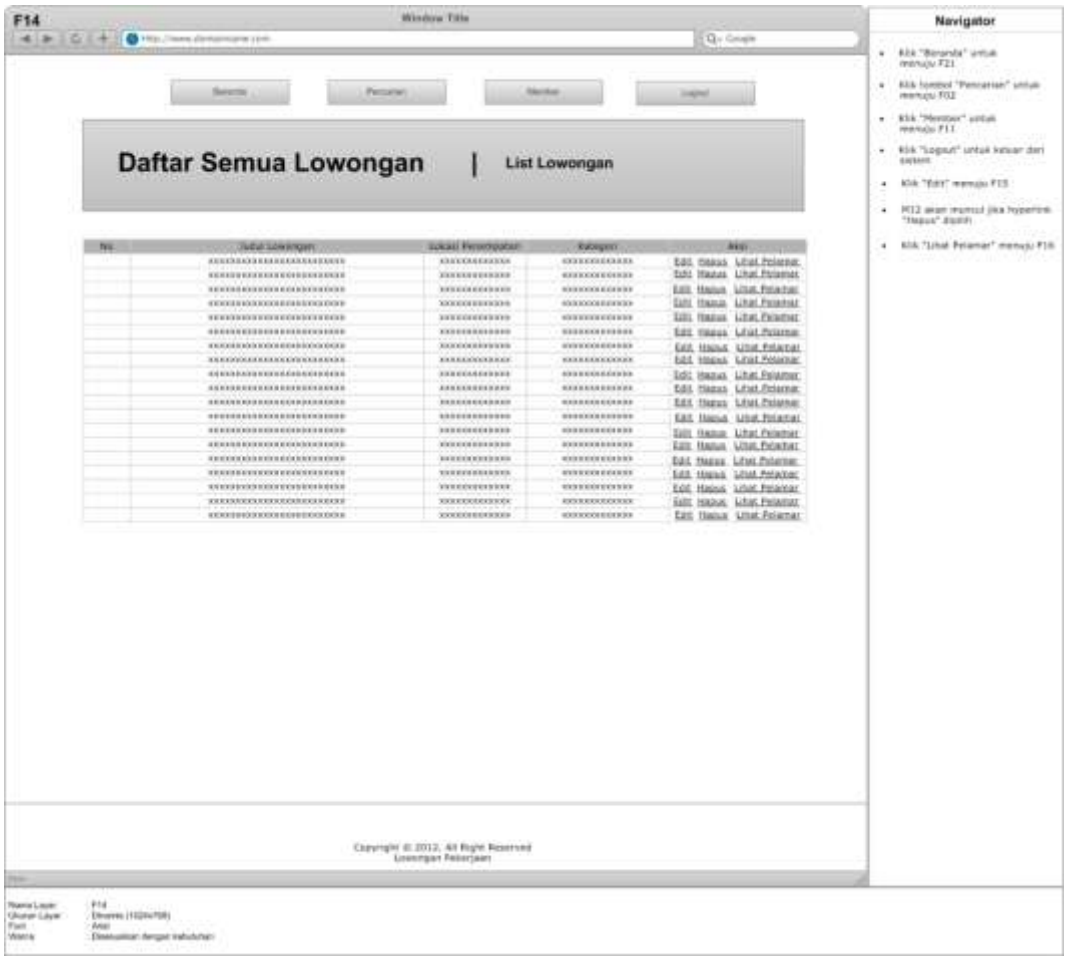
Gambar III. 170 Perancangan halaman Member Pemberi kerja



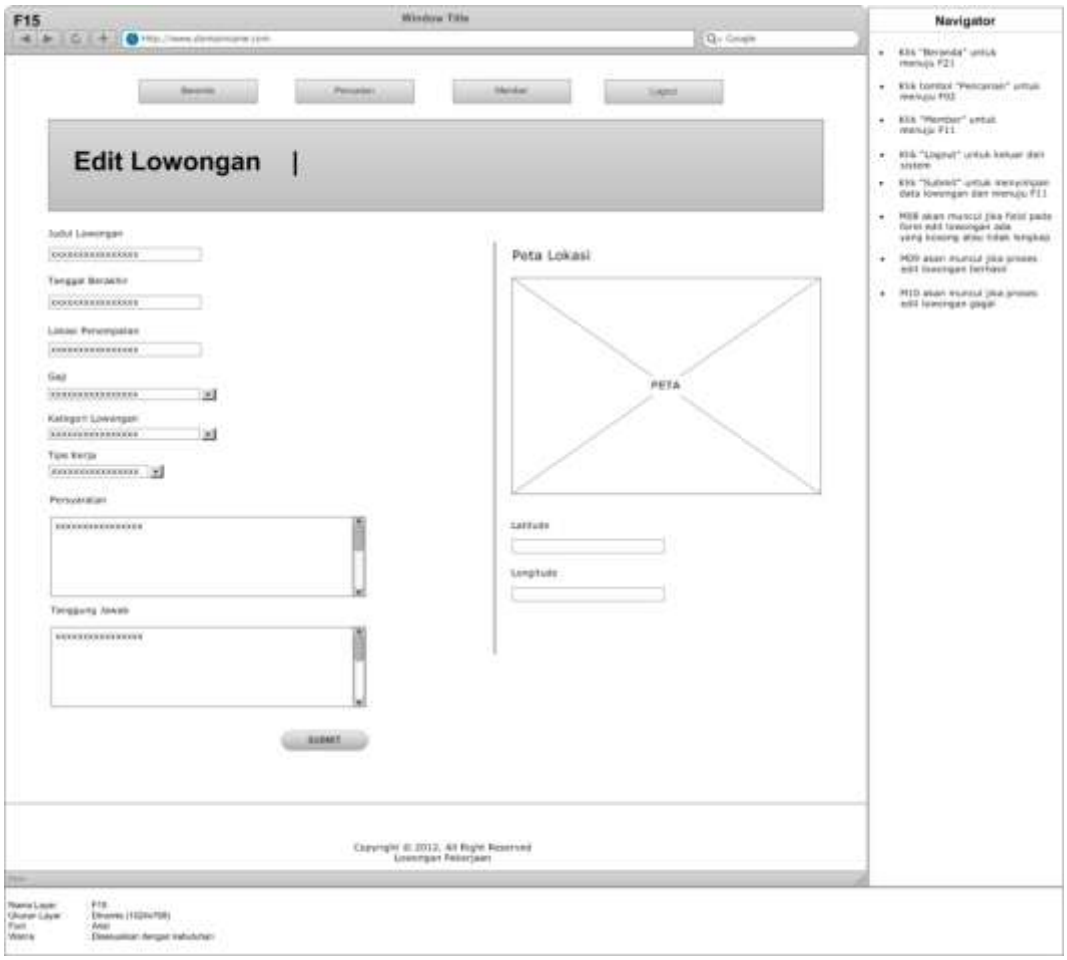
Gambar III. 171 Perancangan halaman Tambah lowongan baru



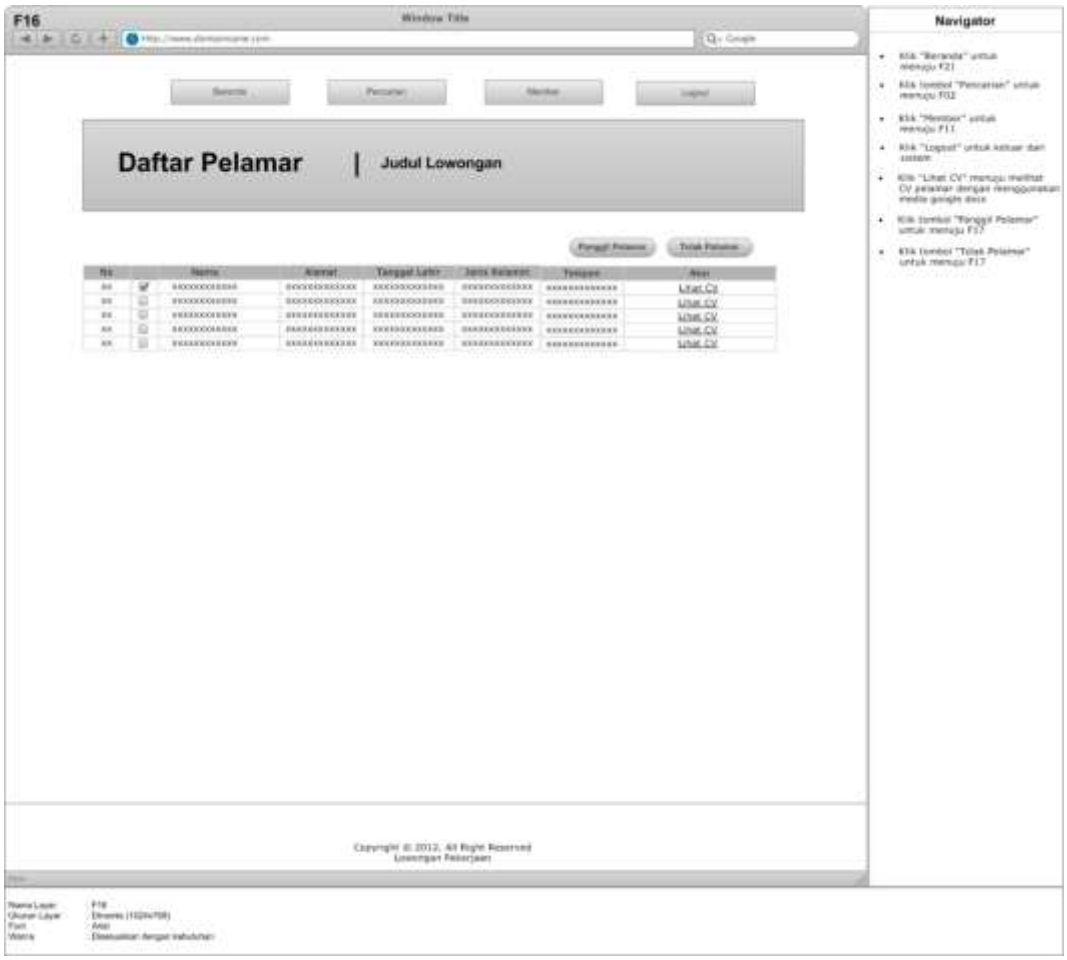
Gambar III. 172 Perancangan menu lupa password



Gambar III. 173 Perancangan halaman Member Pemberi Kerja Daftar Semua Lowongan



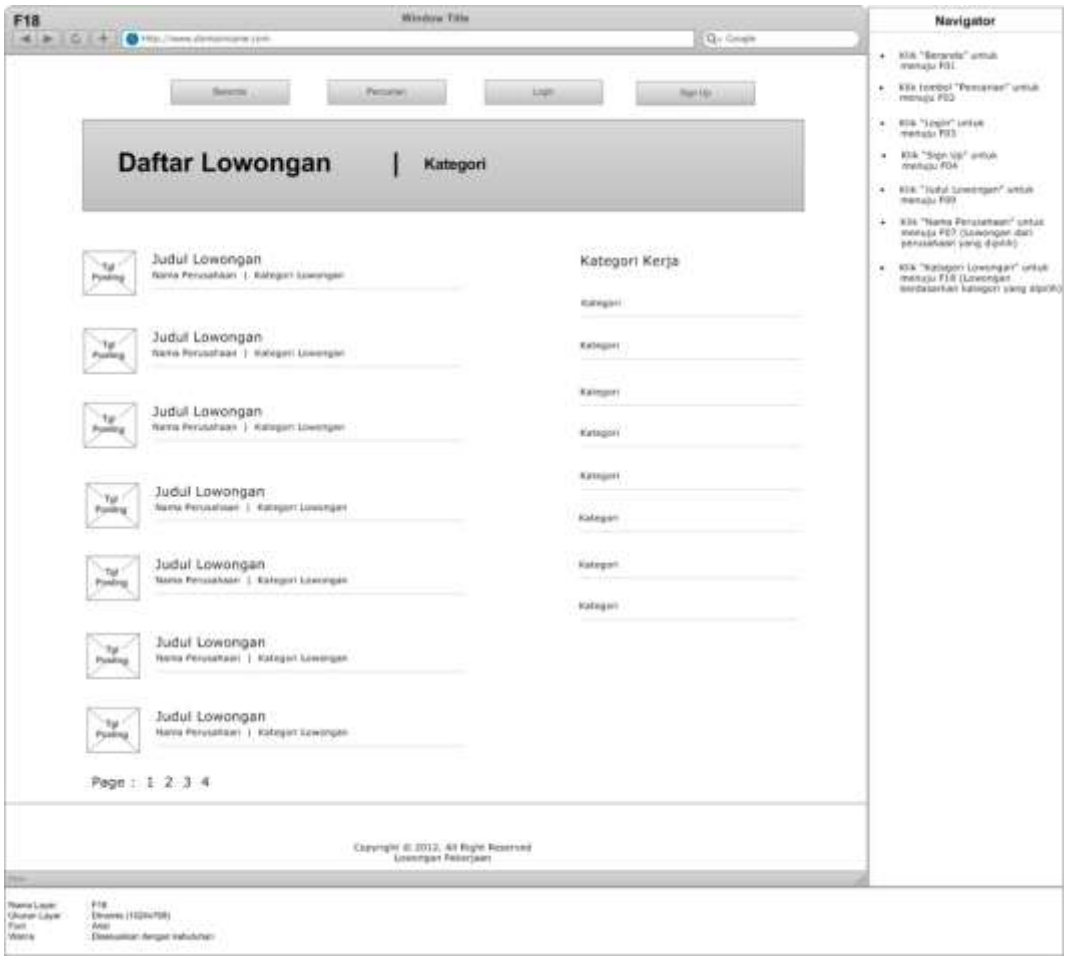
Gambar III. 174 Perancangan halaman Member Pemberi Kerja Edit Lowongan



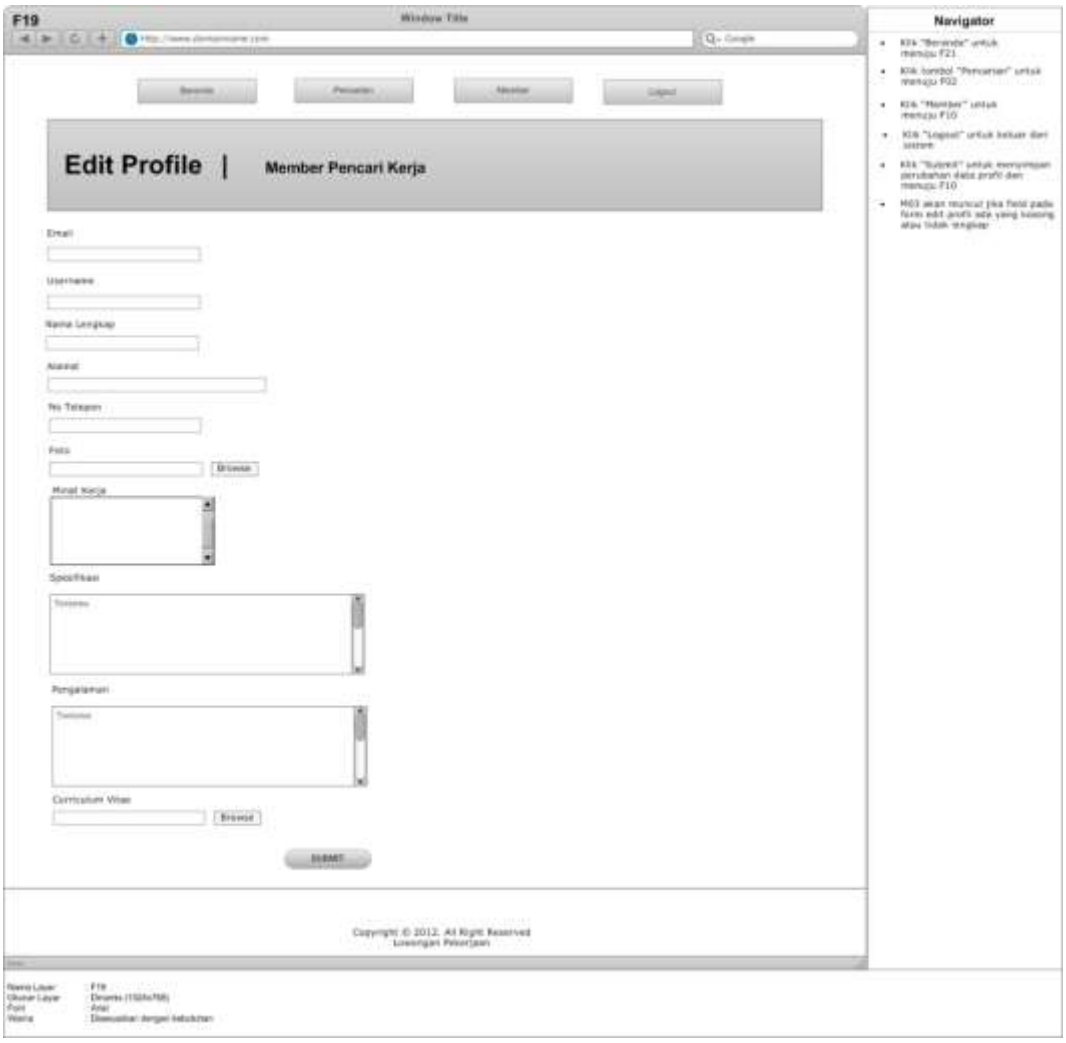
Gambar III. 175 Perancangan halaman Member Pemberi Kerja Lihat Pelamar



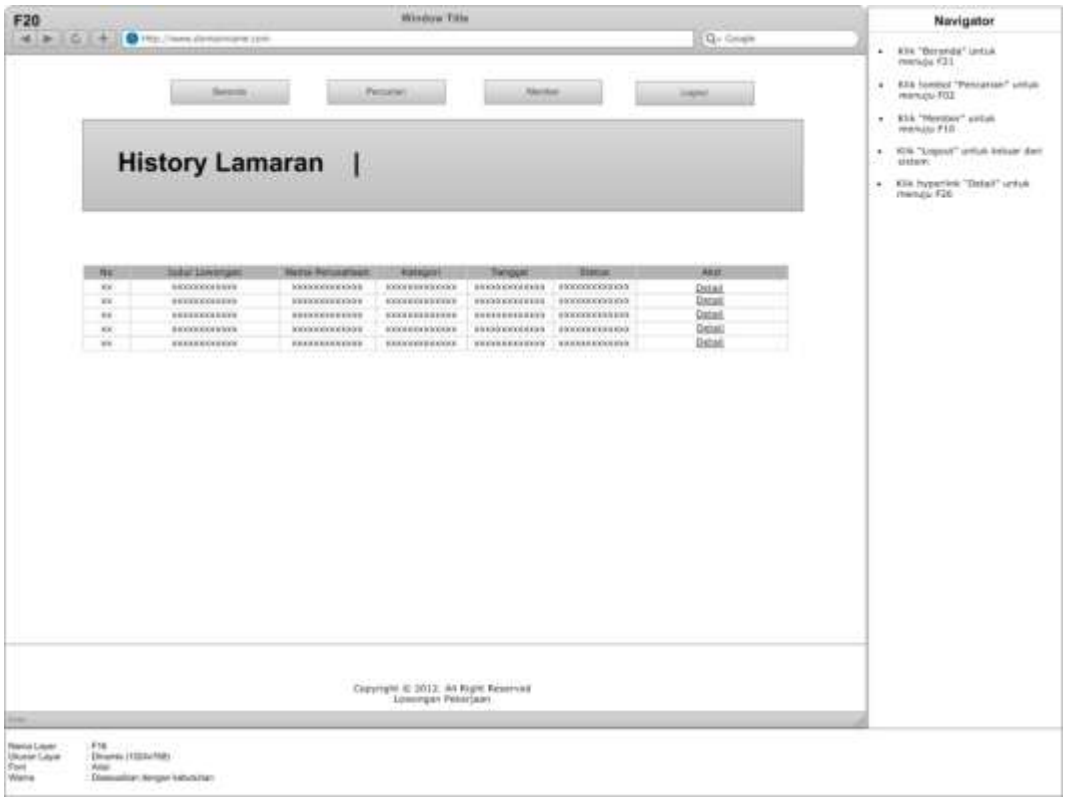
Gambar III. 176 Perancangan halaman Member Pemberi Kerja Panggil Pelamar



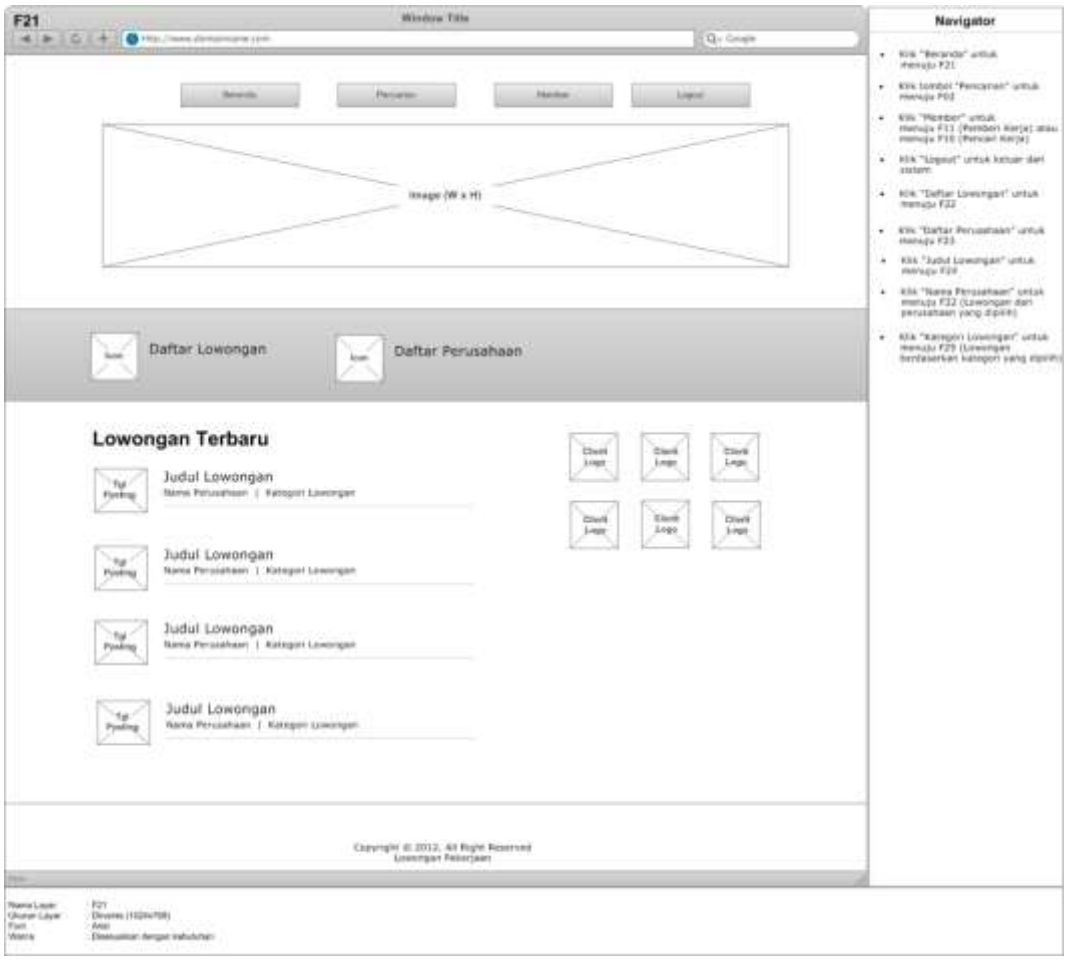
Gambar III. 177 Perancangan halaman Daftar Lowongan Berdasarkan Kategori



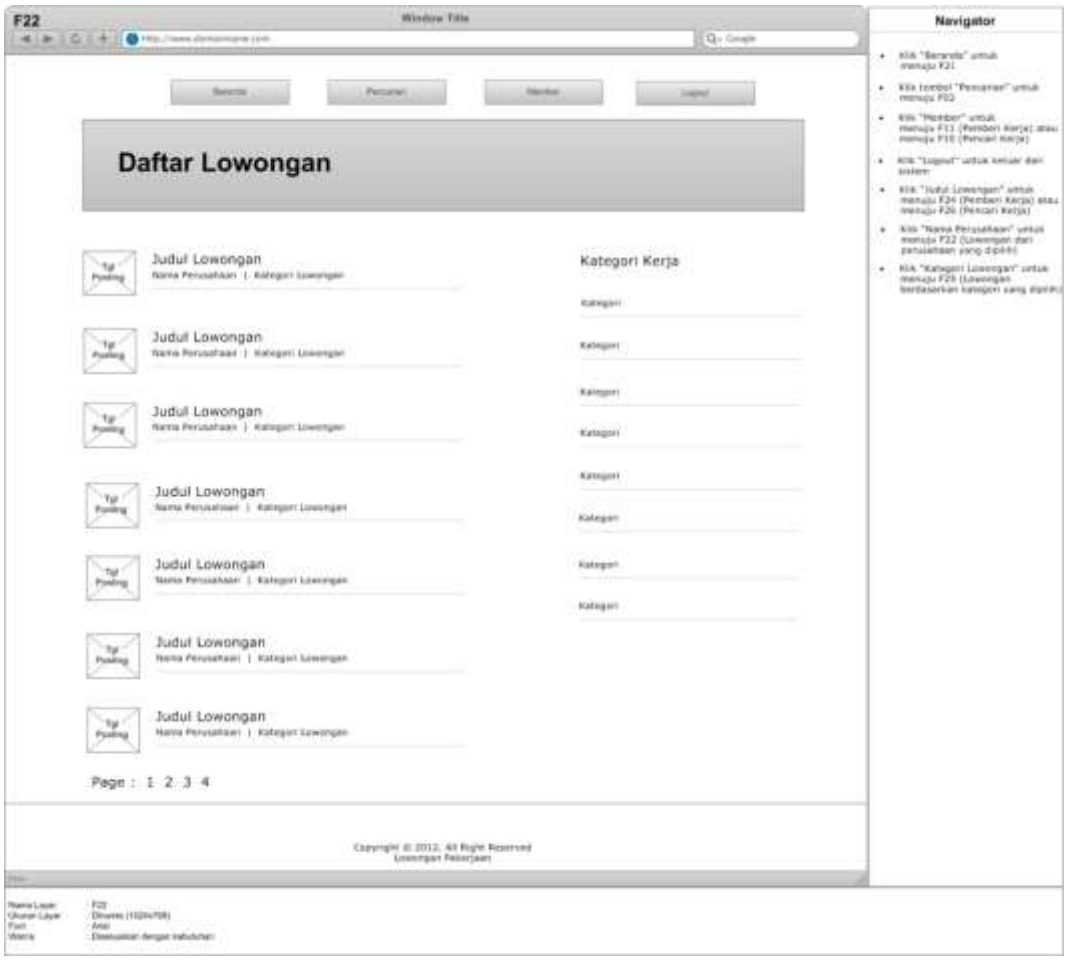
Gambar III. 178 Perancangan halaman Member Pencari Kerja Edit profil



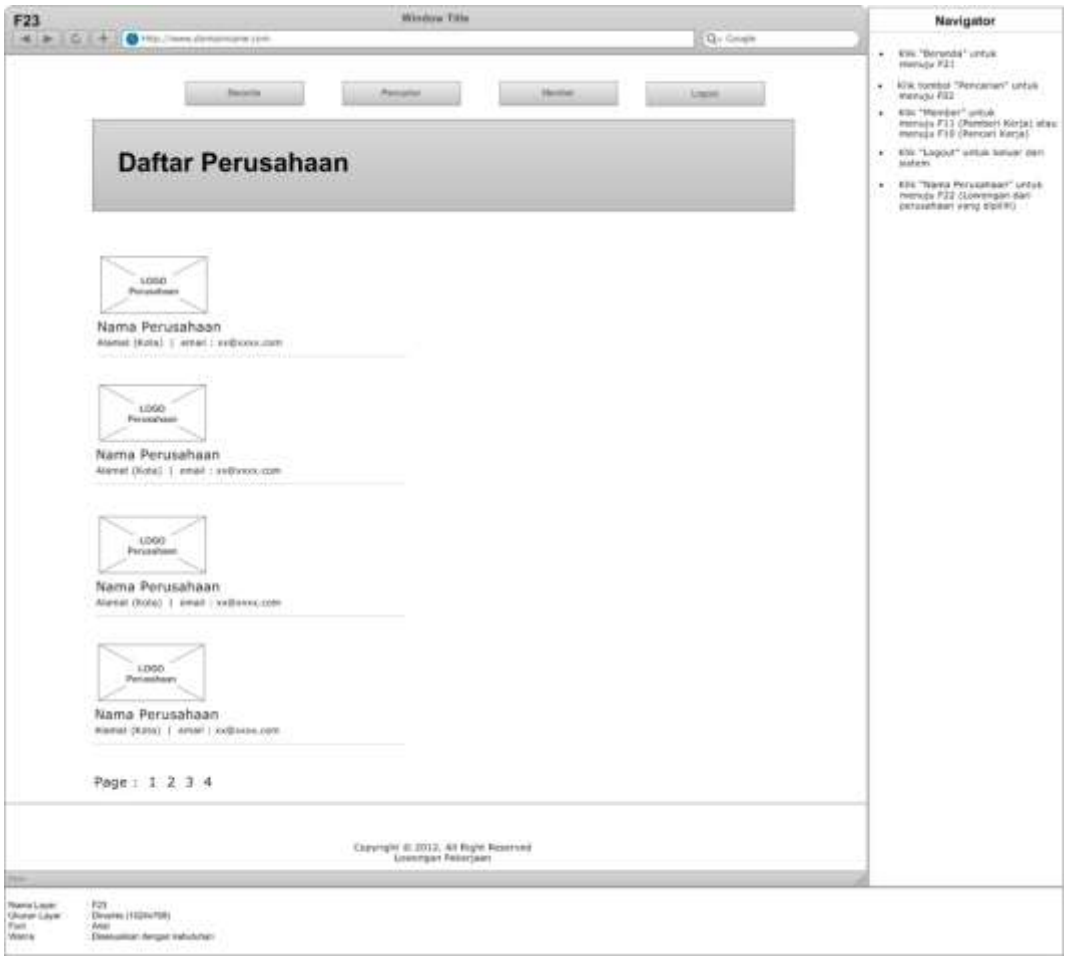
Gambar III. 179 Perancangan halaman Member Pencari History Lamaran



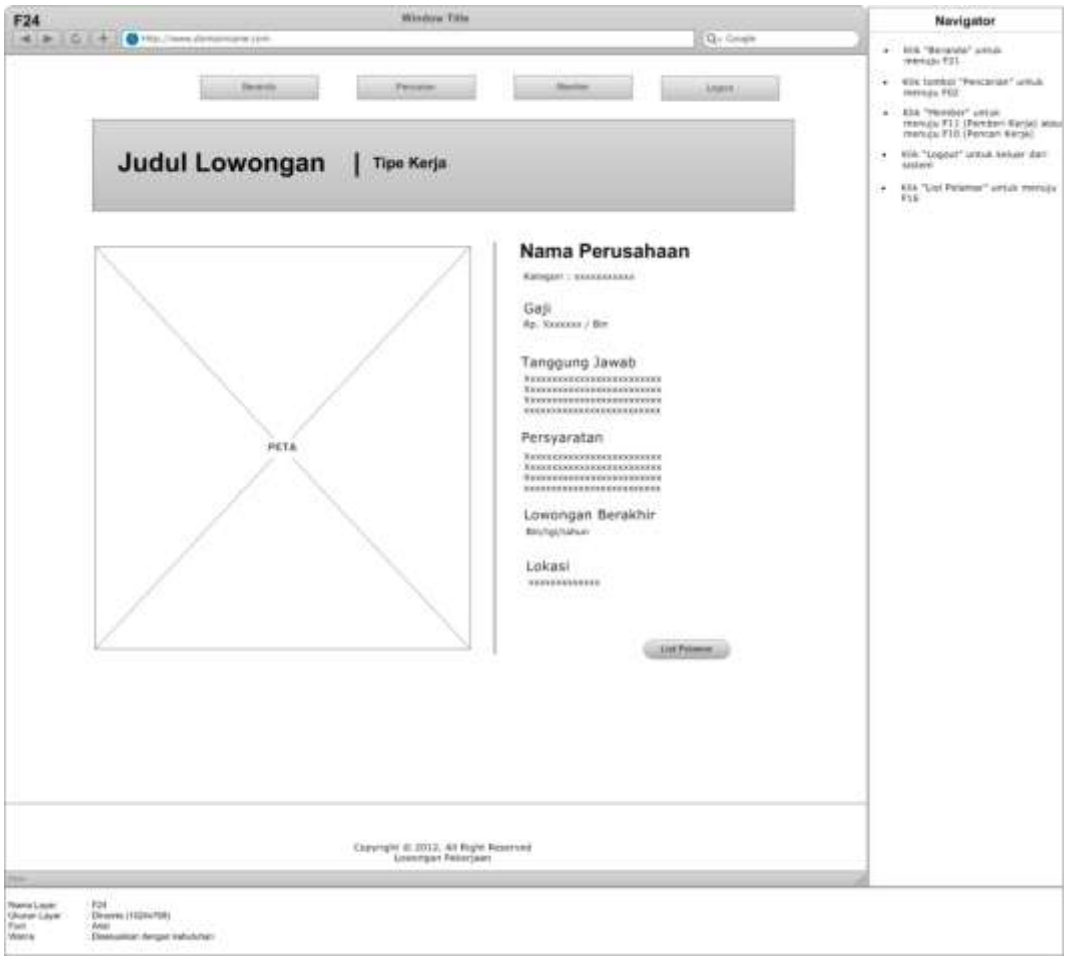
Gambar III. 180 Perancangan halaman Beranda untuk Member Lowongan Pekerjaan



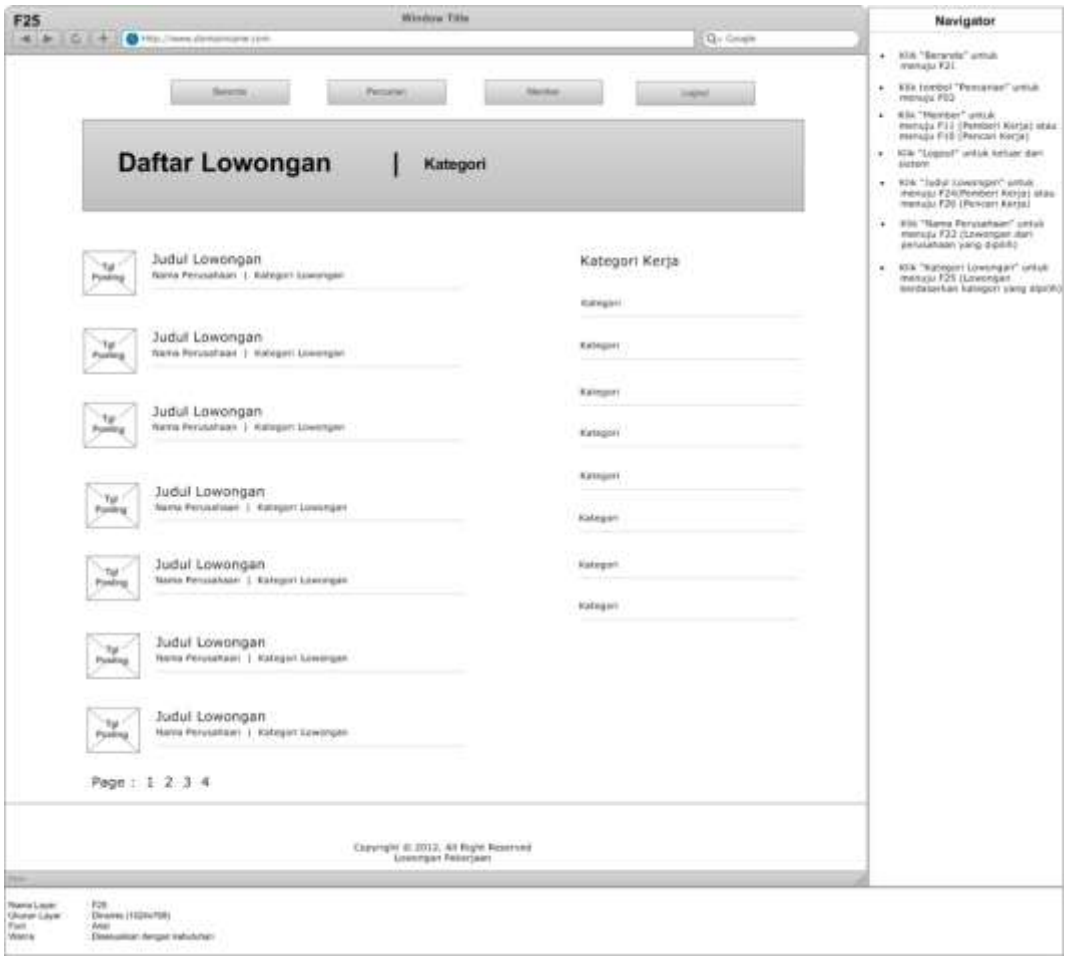
Gambar III. 181 Perancangan halaman Daftar Lowongan untuk Member Lowongan Pekerjaan



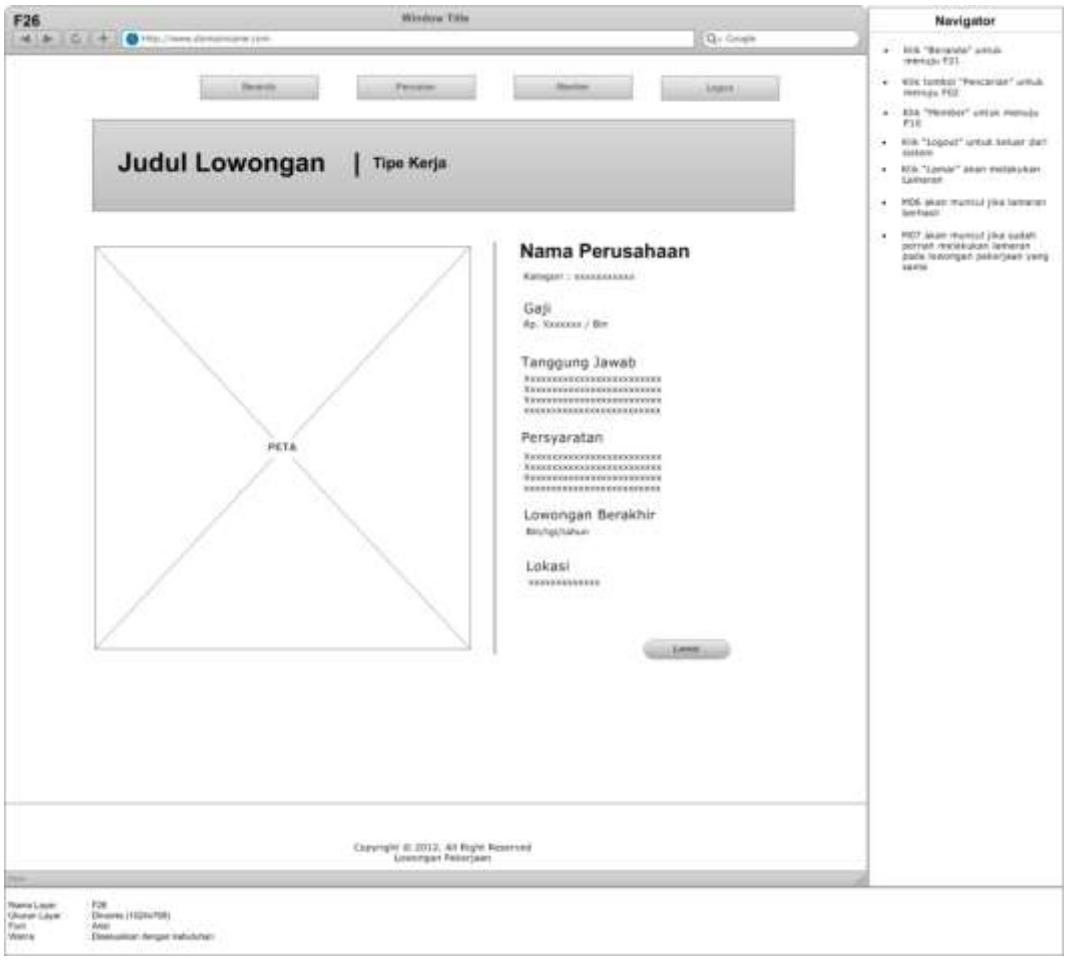
Gambar III. 182 Perancangan halaman Daftar Perusahaan untuk Member Lowongan Pekerjaan



Gambar III. 183 Perancangan halaman Detail Lowongan untuk Member Pemberi Kerja



Gambar III. 184 Perancangan halaman Daftar Lowongan Berdasarkan Kategori untuk Member Lowongan Pekerjaan



Gambar III. 185 Perancangan halaman Detail Lowongan untuk Member Pencari Kerja

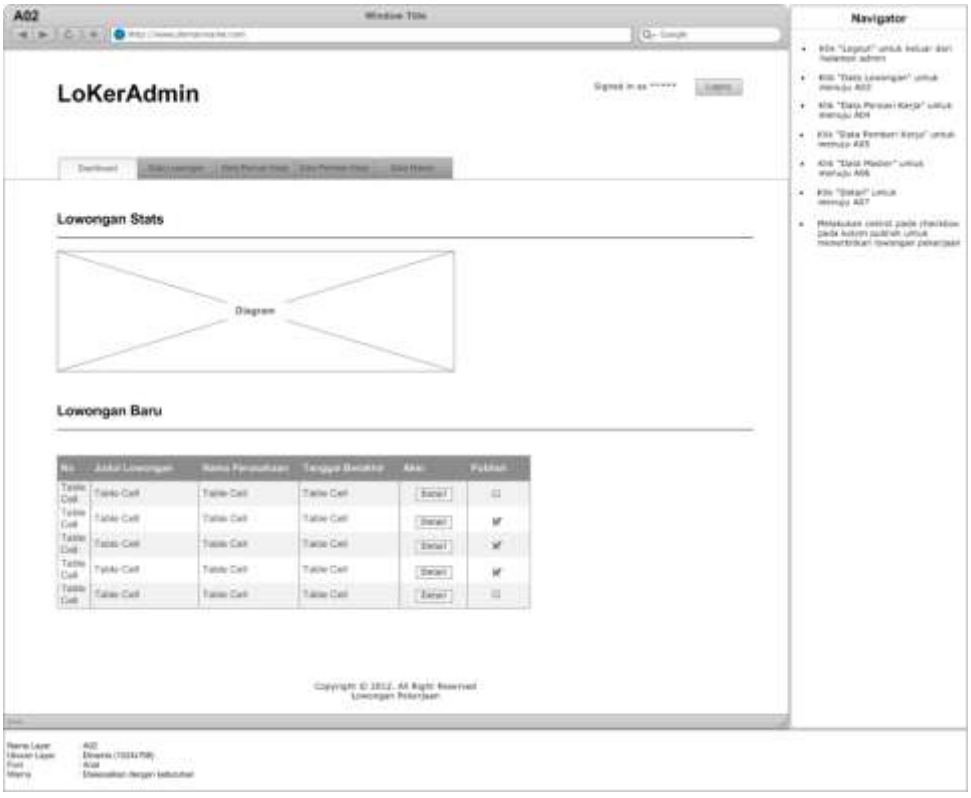


Gambar III. 186 Perancangan pesan web user

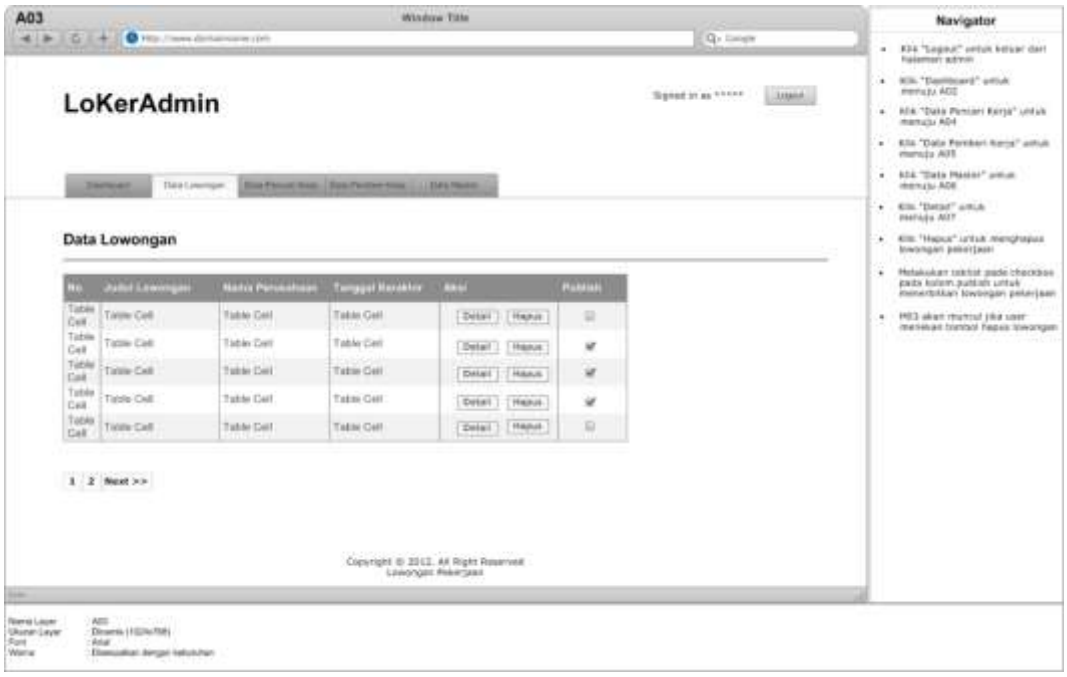
3.2.3.3 Perancangan Antarmuka Admin



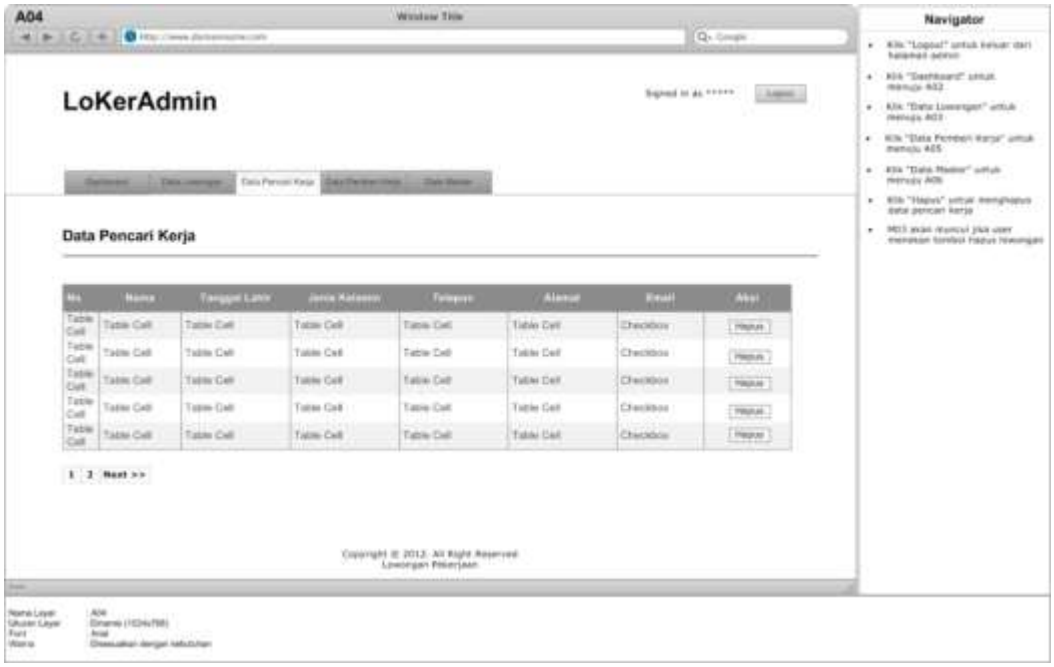
Gambar III. 187 Perancangan halaman login admin



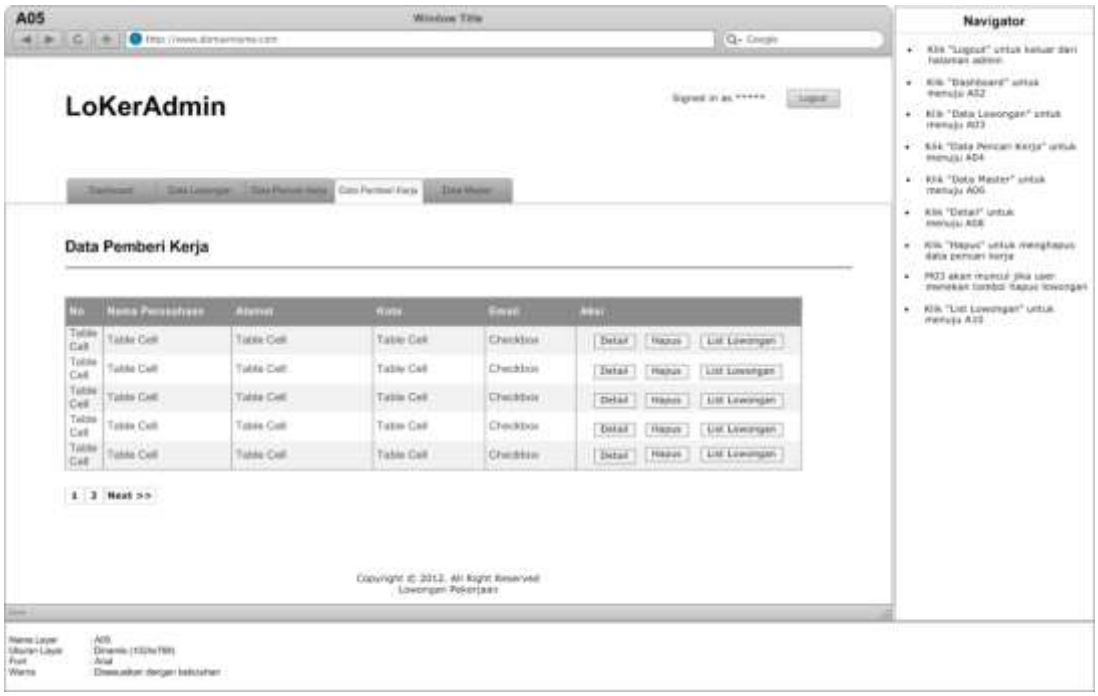
Gambar III. 188 Perancangan halaman utama admin



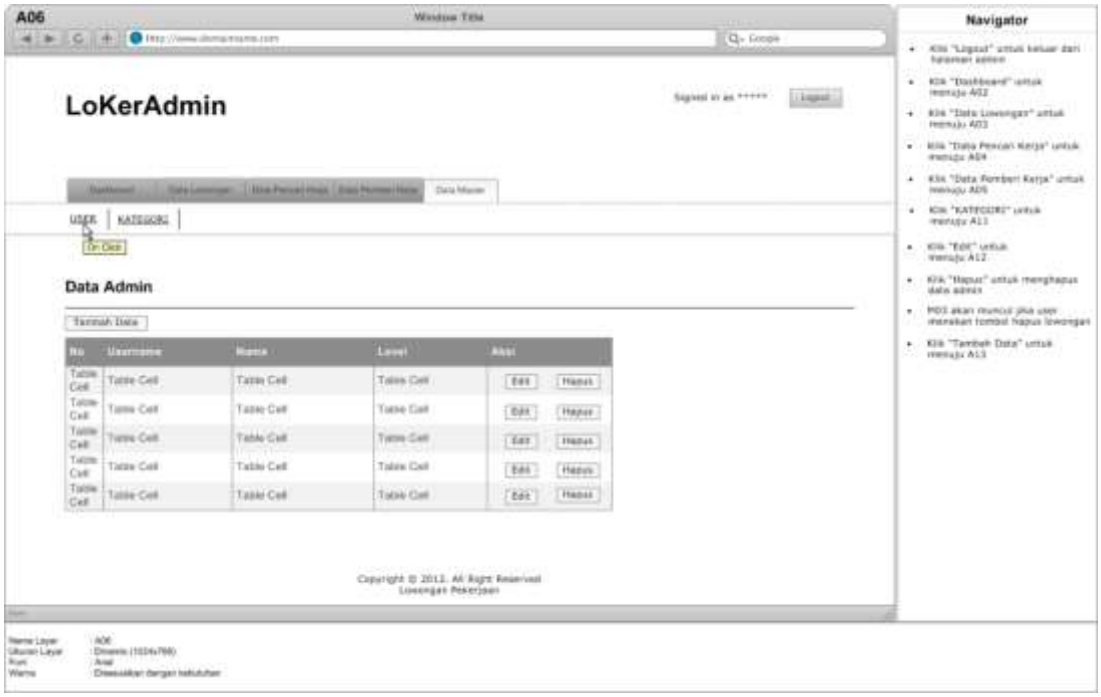
Gambar III. 189 Perancangan halaman admin data lowongan



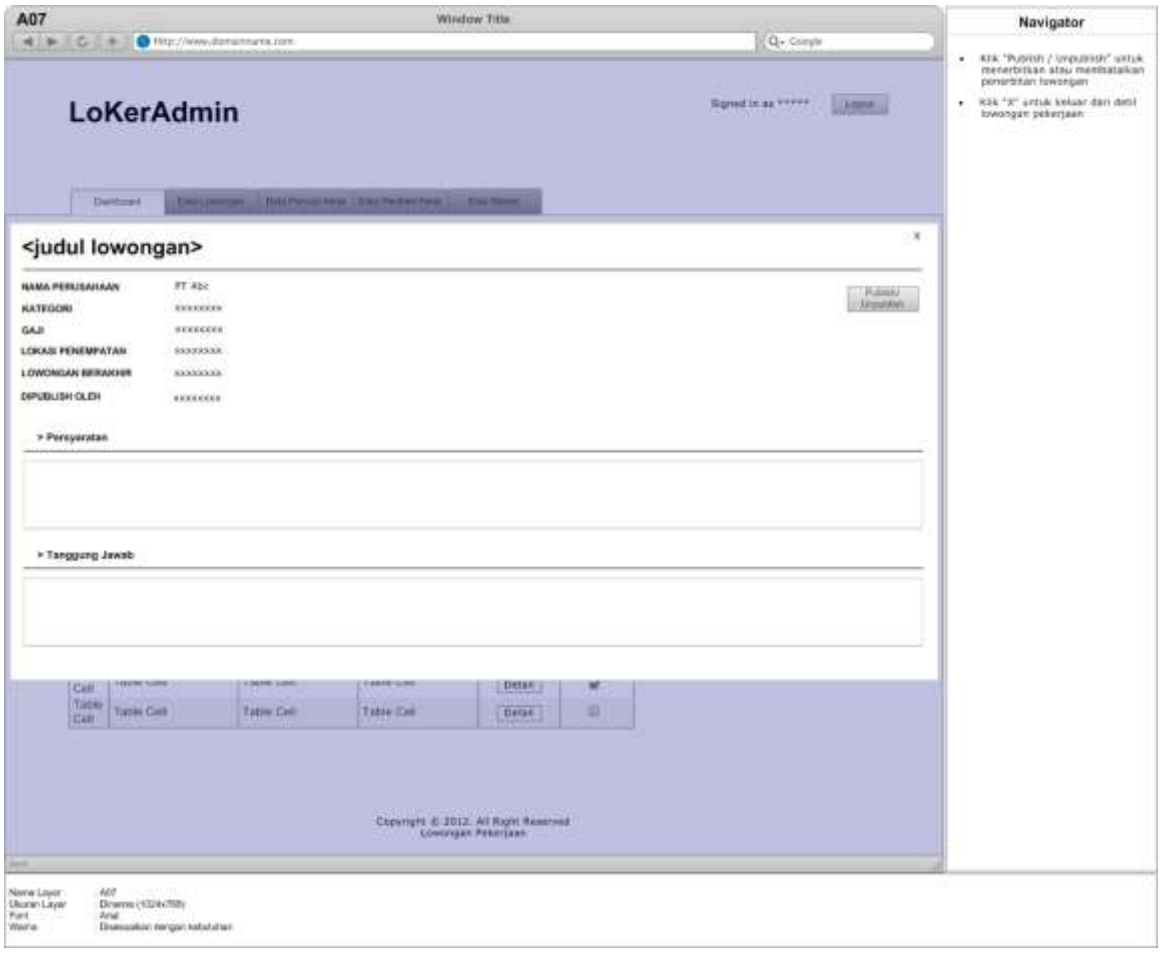
Gambar III. 190 Perancangan halaman admin data pencari kerja



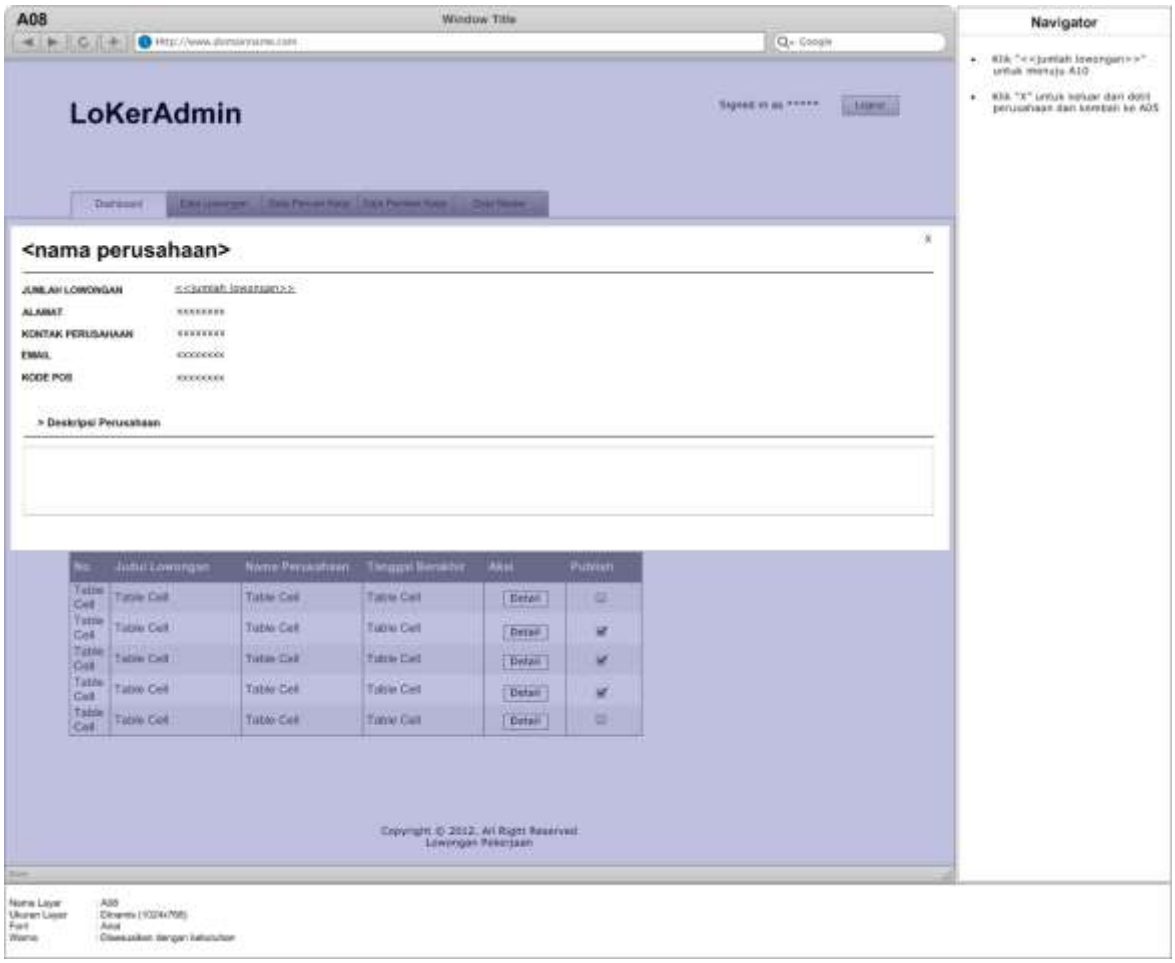
Gambar III. 191 Perancangan halaman admin data pemberi kerja



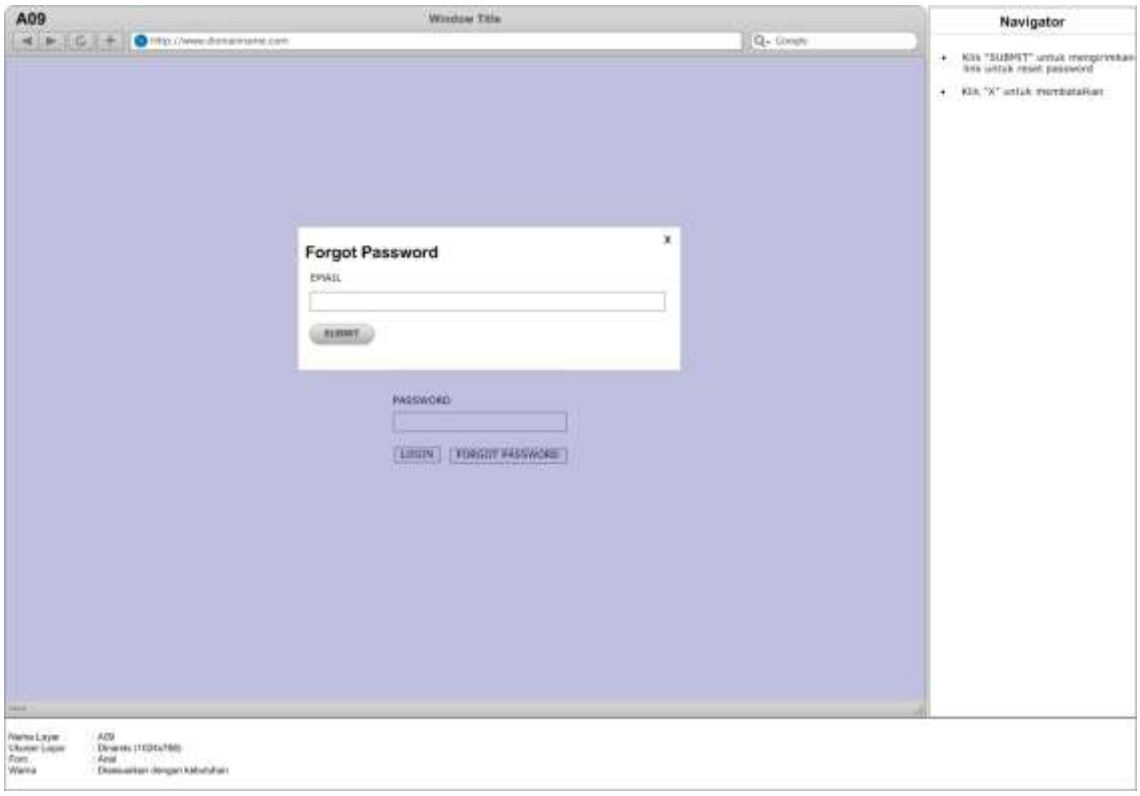
Gambar III. 192 Perancangan halaman data master user admin



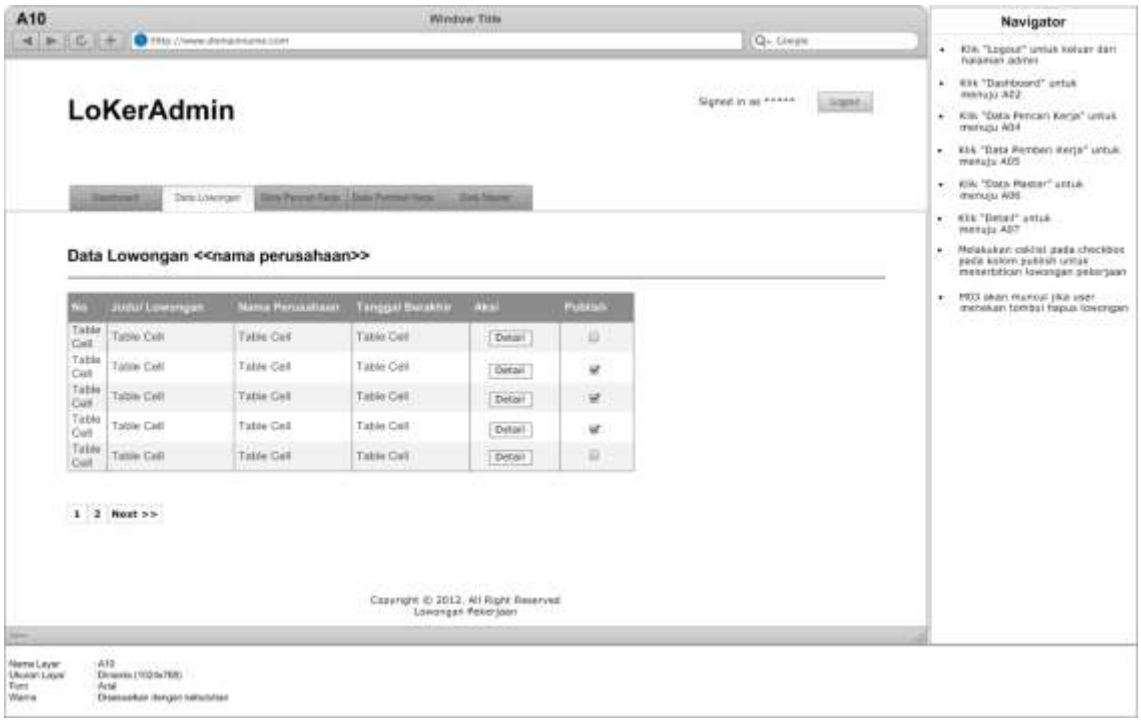
Gambar III. 193 Perancangan halaman admin detil lowongan pekerjaan



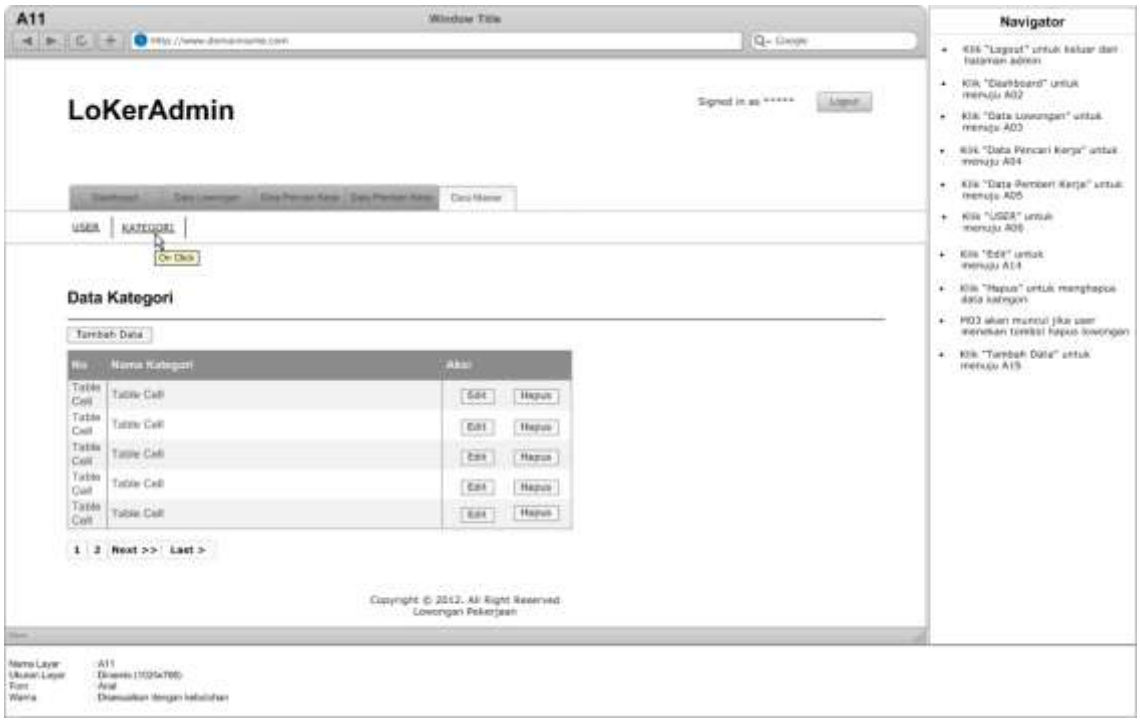
Gambar III. 194 Perancangan halaman admin detil perusahaan



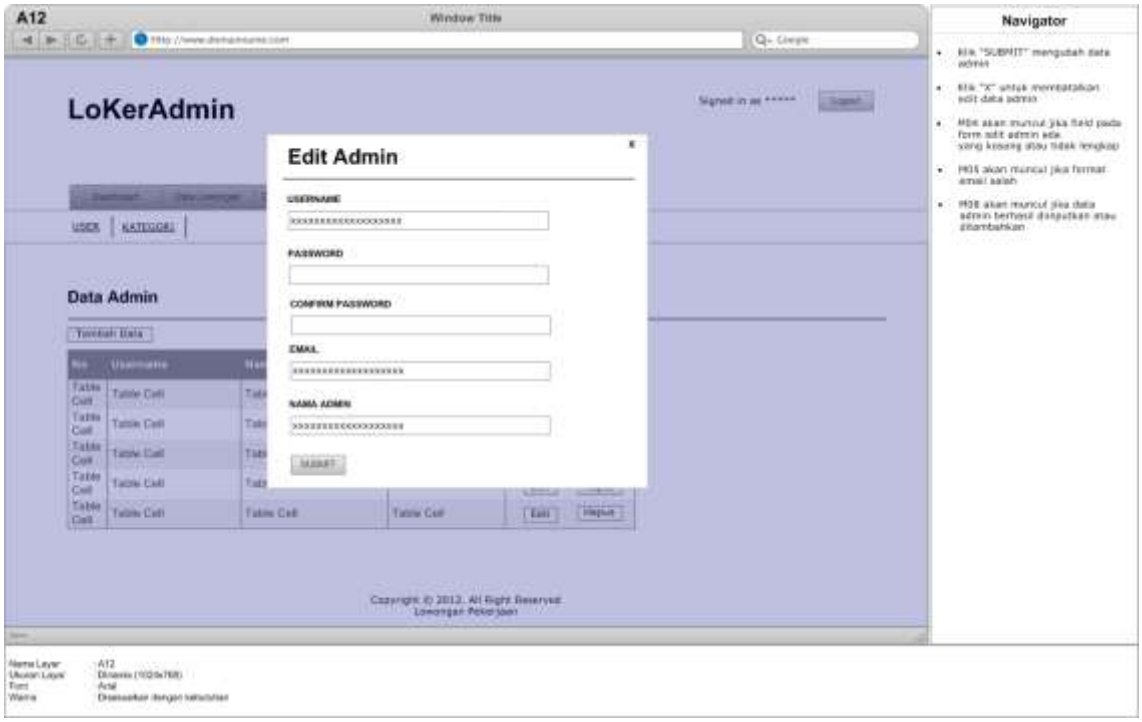
Gambar III. 195 Perancangan halaman admin lupa password



Gambar III. 196 Perancangan halaman admin daftar lowongan perusahaan



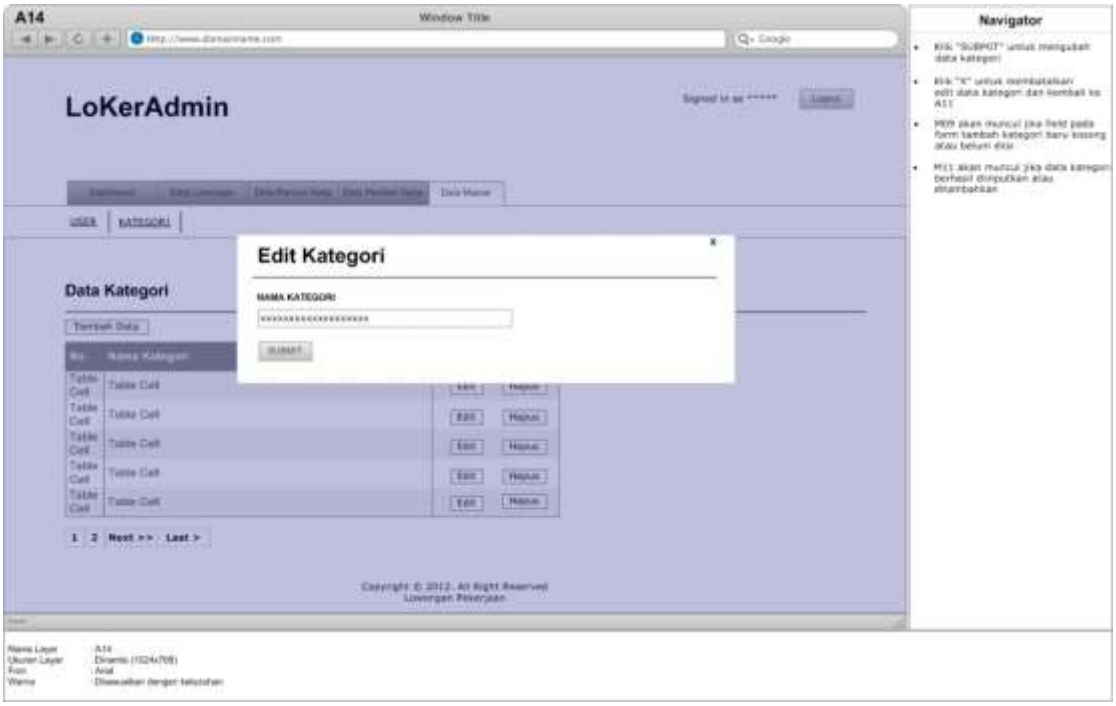
Gambar III. 197 Perancangan halaman data master kategori



Gambar III. 198 Perancangan halaman edit data admin



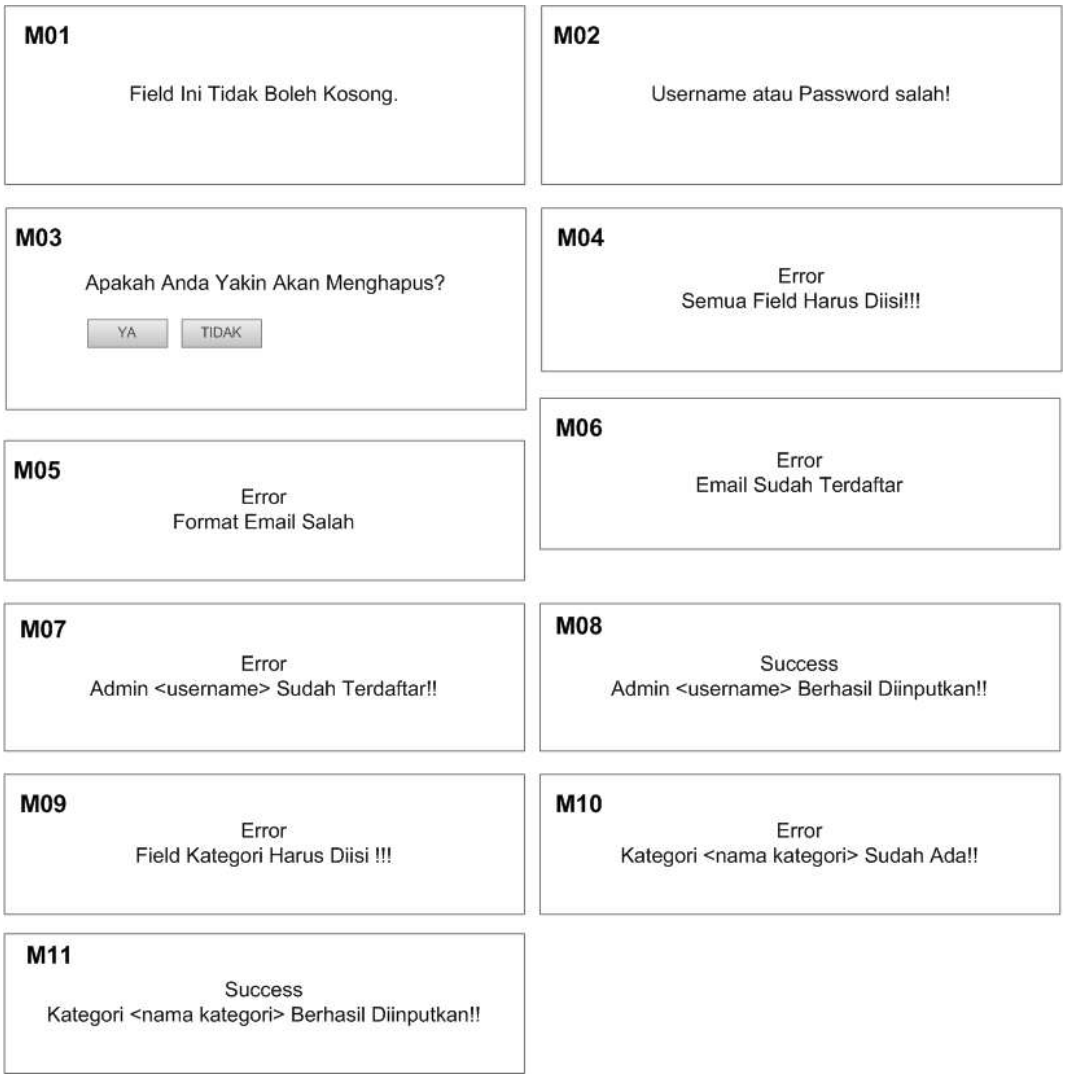
Gambar III. 199 Perancangan halaman tambah data admin



Gambar III. 200 Perancangan halaman edit kategori



Gambar III. 201 Perancangan halaman tambah data kategori

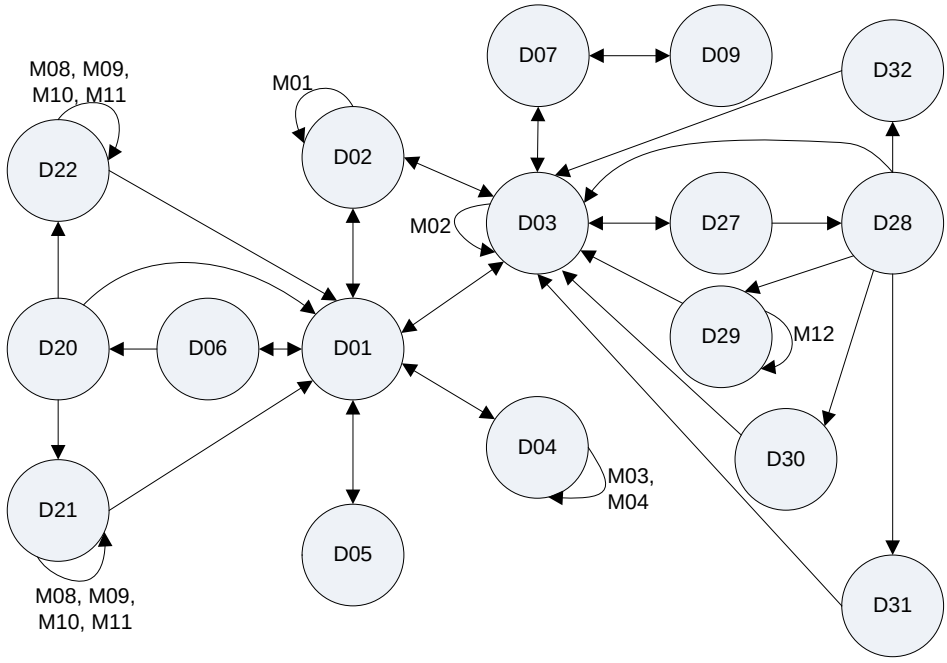


Gambar III. 202 Perancangan pesan web admin

3.2.4 Perancangan Jaringan Semantik

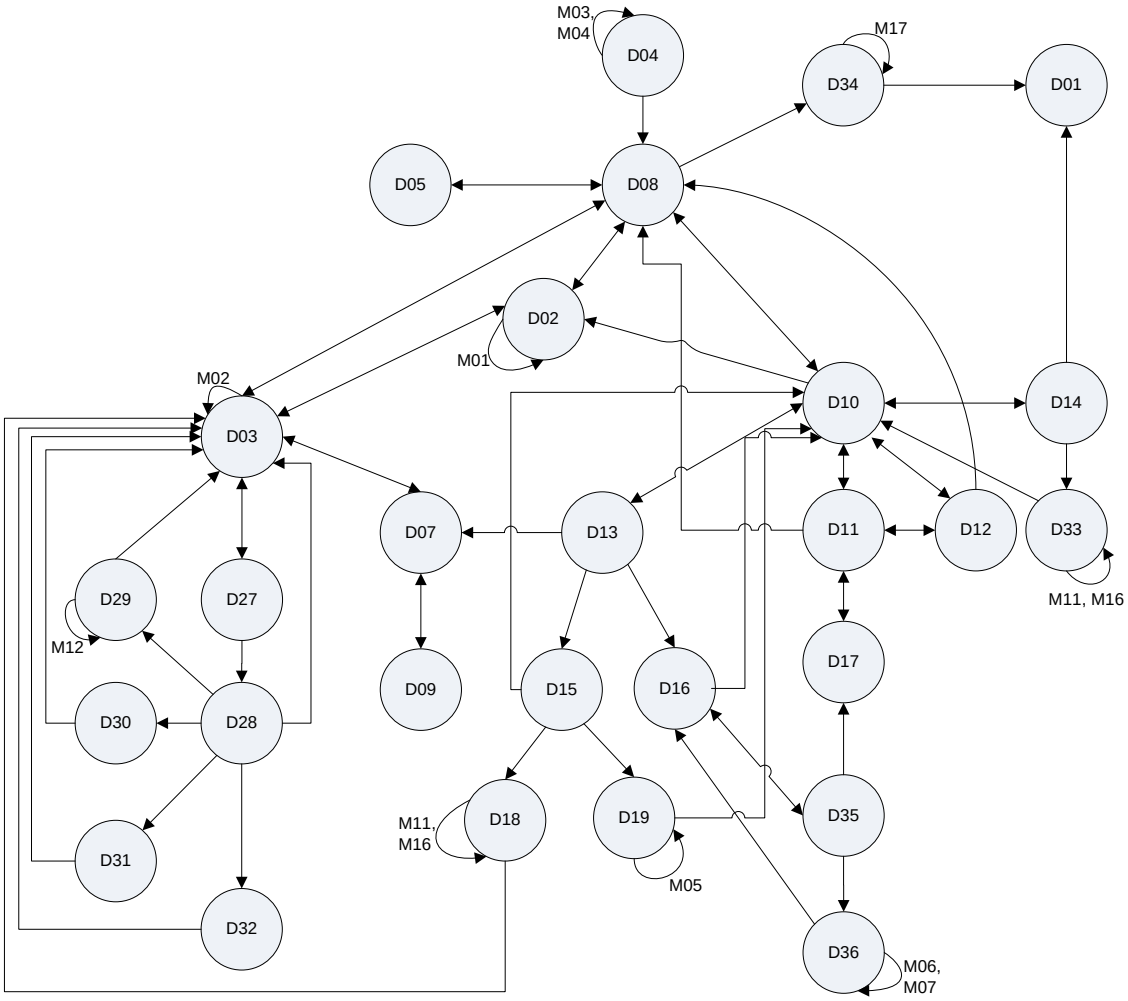
3.2.4.1 Jaringan Semantik Aplikasi Android

1. Jaringan Semantik Pengunjung Umum



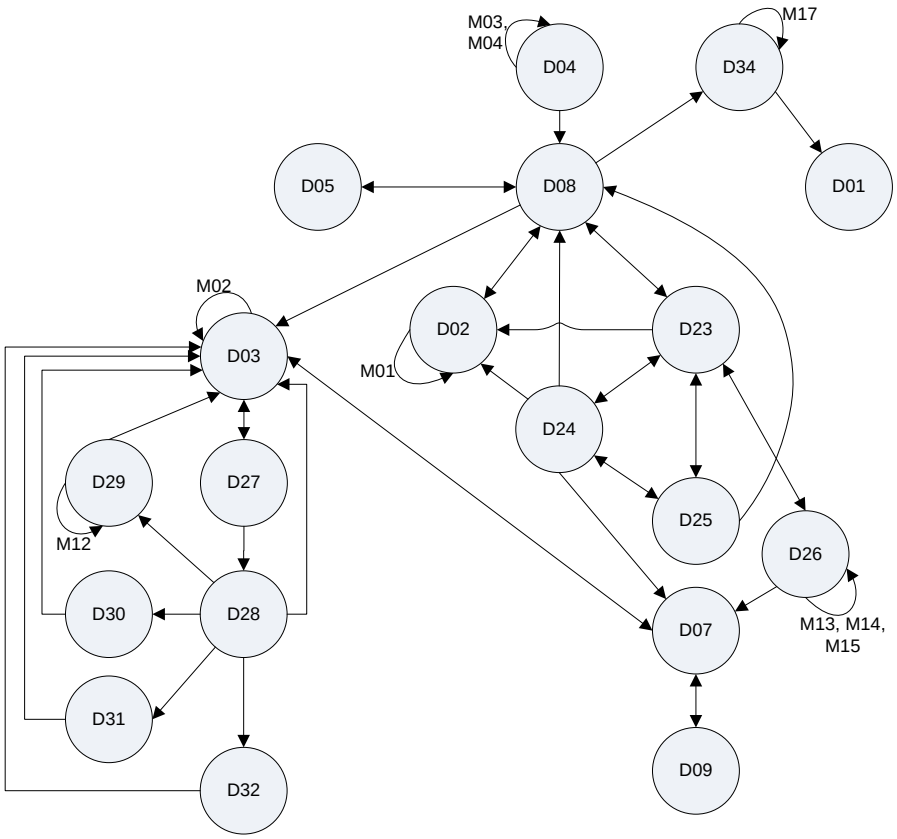
Gambar III. 203 Jaringan Semantik Aplikasi Android Pengunjung Umum

2. Jaringan Semantik Member Pemberi Kerja



Gambar III. 204 Jaringan Semantik Aplikasi Android Member Pemberi Kerja

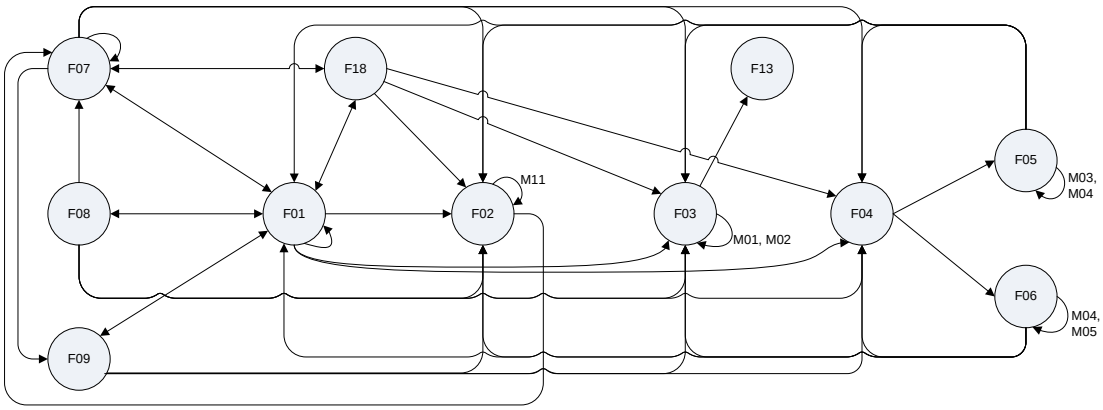
3. Jaringan Semantik Member Pencari Kerja



Gambar III. 205 Jaringan Semantik Aplikasi Android Member Pencari Kerja

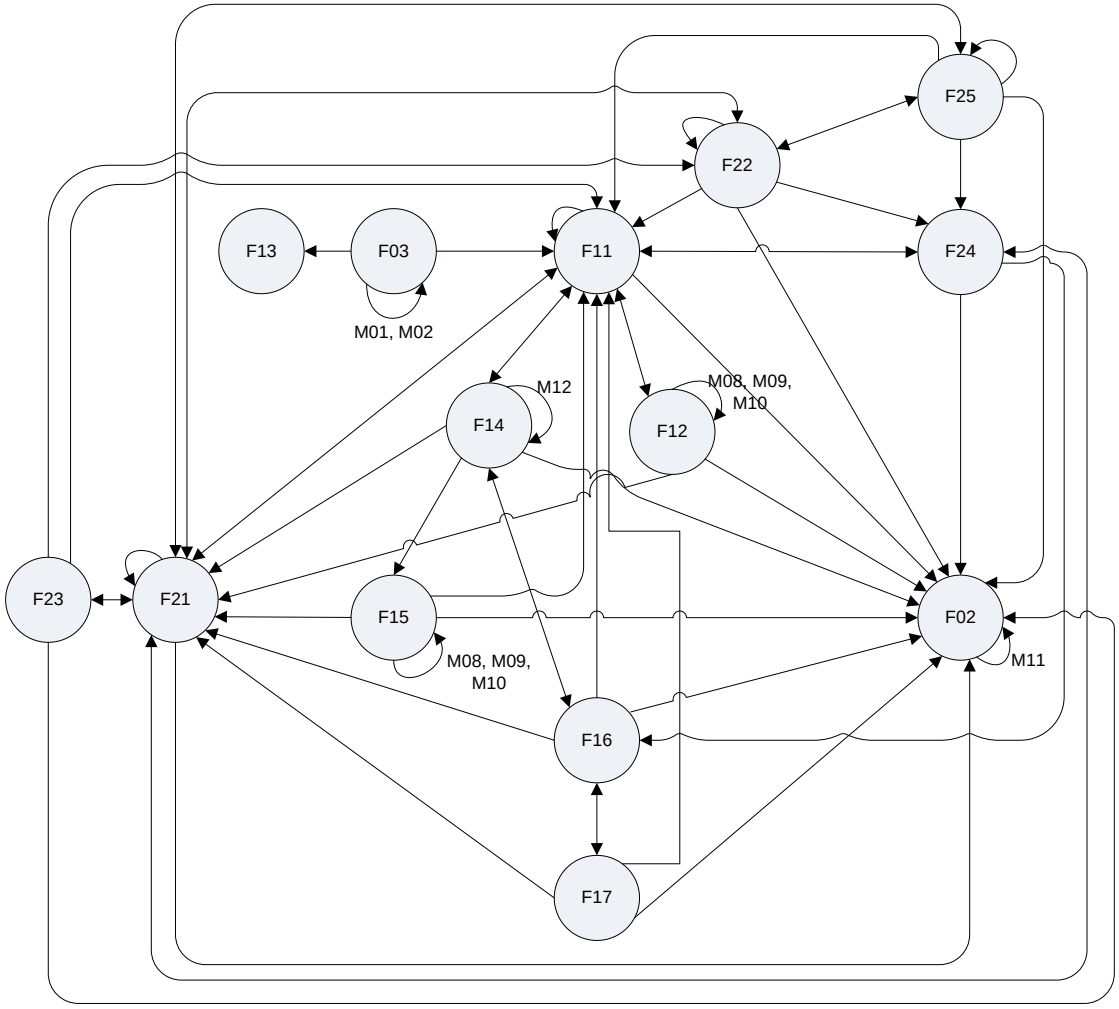
3.2.4.2 Jaringan Semantik Aplikasi Web User

1. Jaringan Semantik Pengunjung Umum



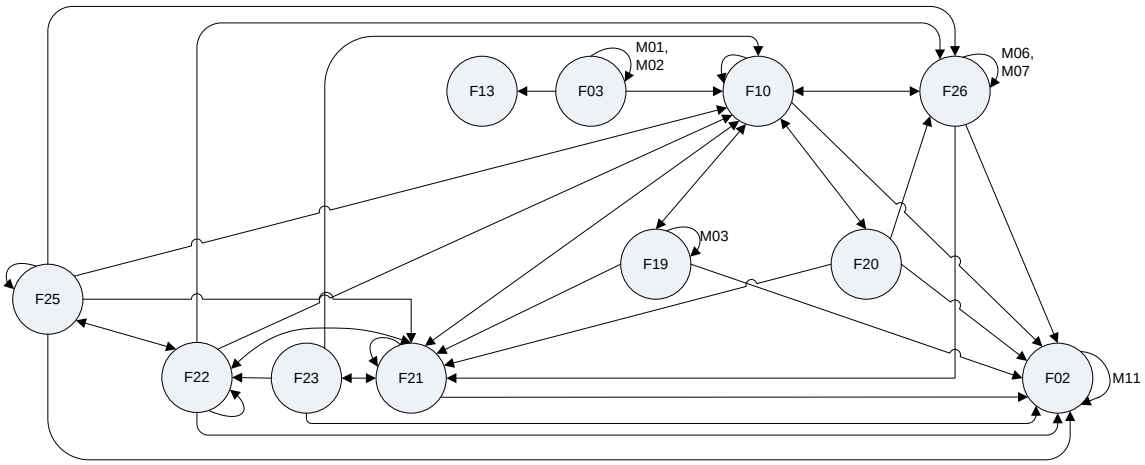
Gambar III. 206 Jaringan Semantik Aplikasi Web User Pengunjung Umum

2. Jaringan Semantik Member Pemberi Kerja



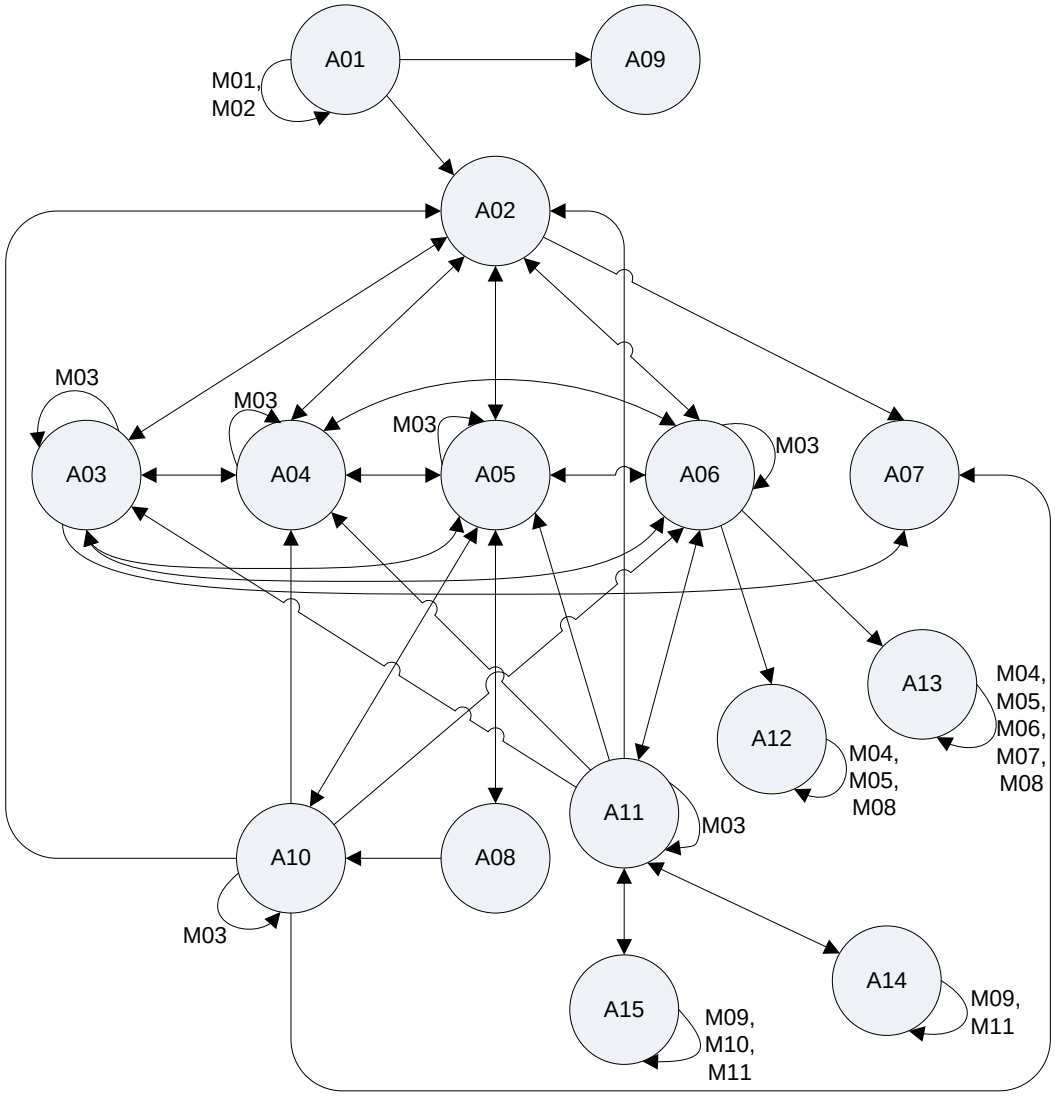
Gambar III. 207 Jaringan Semantik Aplikasi Web User Member Pemberi Kerja

3. Jaringan Semantik Member Pencari Kerja



Gambar III. 208 Jaringan Semantik Aplikasi Web User Member Pencari Kerja

3.2.4.3 Jaringan Semantik Aplikasi Admin



Gambar III. 209 Jaringan Semantik Aplikasi Web Admin