BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem

Pada tahap perancangan ini akan dijelaskan mengenai perancangan aplikasi pada objek yang digunakan, perancangan arsitektur program yang akan dibuat, perancangan tampilan dan perancangan menu. Analisis dan perancangan adalah serangkaian kegiatan yang selalu beriringan dalam setiap pengembangan *software*, sebagai sebuah hubungan sebab dan akibat yang memunculkan sebuah siklus hidup sistem. Walaupun dalam kenyataannya pengembangan sistem yang sederhana, aktivitas ini tidak tampak.

4.1.1 Tujuan Perancangan Sistem

Tujuan dari perancangan ini merupakan suatu kegiatan pengembangan prosedur dan proses yang sedang berjalan untuk menghasilkan sesuatu yang baru atau memperbaharui yang ada untuk meningkatkan kinerja sistem itu sendiri sehingga dapat memenuhi hasil yang diinginkan. Rancangan aplikasi yang baru, akan diterapkan suatu kegiatan untuk menemukan dan mengembangkan metode, prosedur dan proses suatu data agar tujuan dari suatu organisasi dapat tercapai.

Adapun tujuan dari tahap perancangan ini adalah untuk menghasilkan perancangan aplikasi berupa pemodelan menggunakan pendekatan berorientasi objek sehingga dapat memperbaiki atau meningkatkan efisiensi kerja sistem yang sedang berjalan.

4.1.2 Gambaran Umum Sistem Yang Diusulkan

Gambaran umum tentang aplikasi yang diusulkan pada proses perancangan ini adalah suatu aplikasi yang digunakan untuk membuat jadwal kegiatan upacara dengan harapan mampu menangani permasalahan yang ada pada penjadwalan yang sedang berjalan pada PT.Pindad.

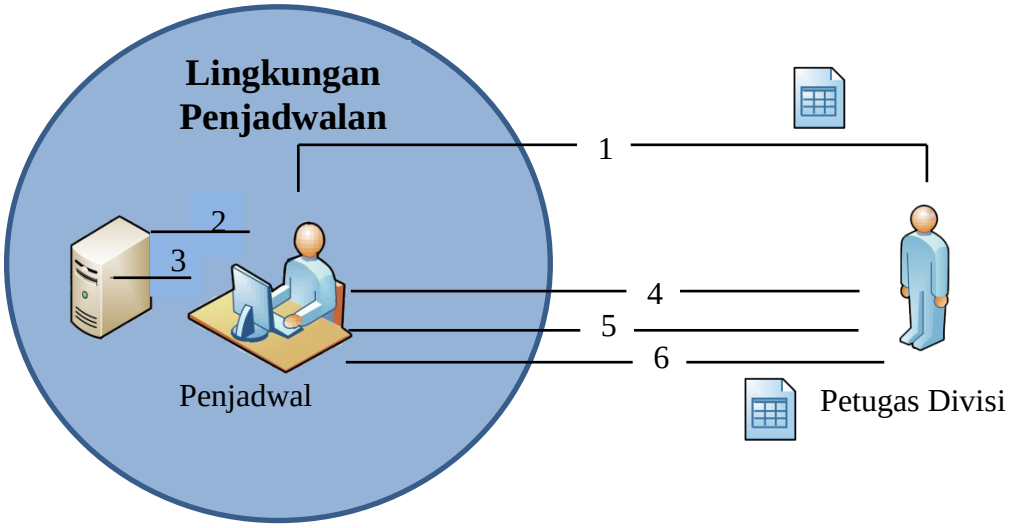
Aplikasi ini sendiri dibangun atas dasar kebutuhan akan pembuatan jadwal kegiatan upacara dengan memperhatikan ketersediaan pegawai yang dapat mengikuti kegiatan upacara. Pengguna dari aplikasi ini dikhususkan pada petugas penjadwalan yang telah ditunjuk oleh koordinator atau ketua pelaksana kegiatan upacara.

4.1.3 Perancangan Prosedur Yang Diusulkan

Pada tahap ini, perancangan prosedur yang diusulkan dilakukan untuk memperbaiki atau meningkatkan efisiensi kerja. Tahap perancangan yang digambarkan merupakan tahap perancangan untuk membangun suatu sistem dan mengkonfigurasikan komponen-komponen perangkat lunak dan perangkat keras sehingga menghasilkan sistem yang baik. Tahapan perancangan prosedur ini akan dijelaskan dengan menggunakan pemodelan sistem informasi berorientasi objek dengan *UML*.

4.1.3.1 Proses Bisnis

Proses bisnis target sistem adalah sebagai berikut :



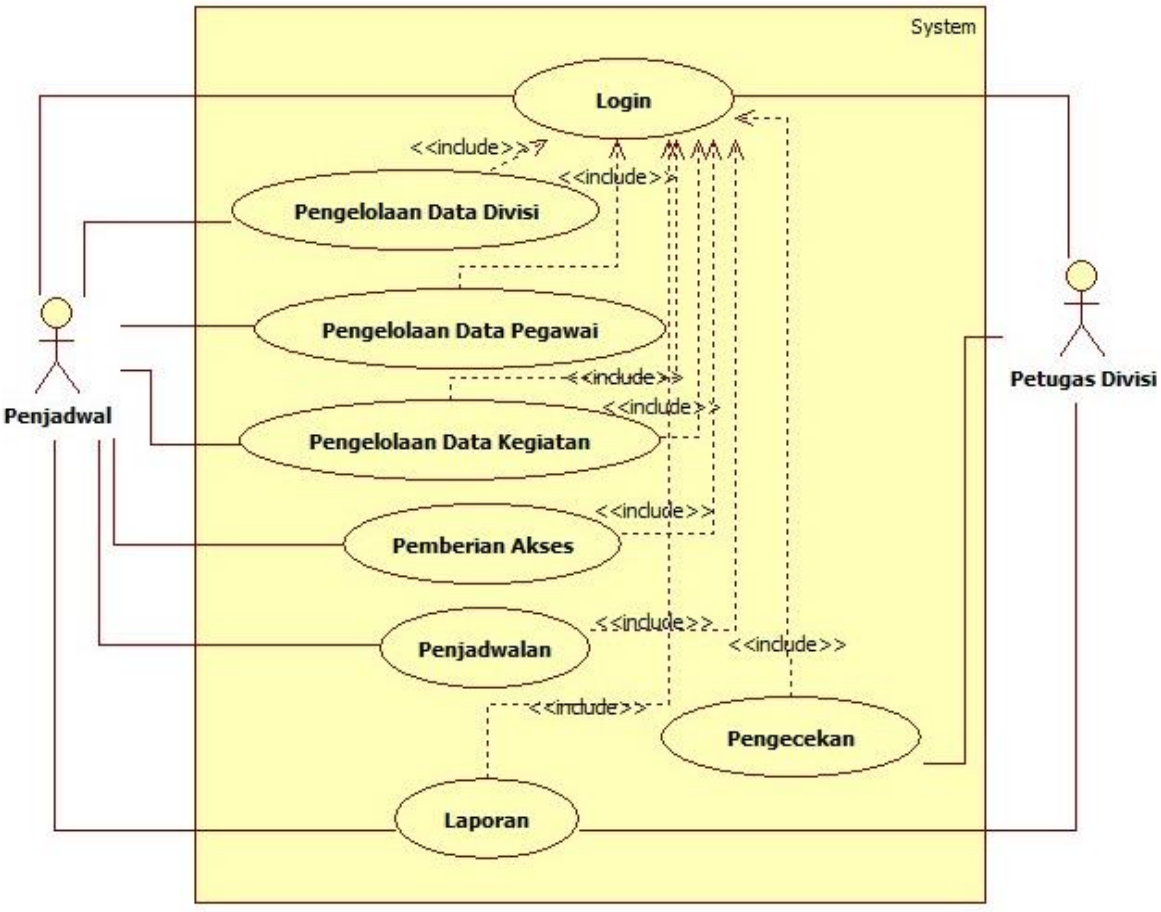
Gambar 4.1 Proses Bisnis Penjadwalan Kegiatan Upacara

Keterangan:

1. Penjadwal menerima export data pegawai yang akan menjadi anggota upacara kepada penjadwal.
2. Penjadwal menyimpan data pegawai ke database penjadwalan.
3. Penjadwal membuat tanggal dilaksanakannya kegiatan upacara beserta jadwal pegawai yang menjadi anggota upacara pada kegiatan upacara yang telah dibuat.
4. Petugas penjadwal memberikan laporan kegiatan kepada petugas divisi untuk dilakukan pengecekan apakah pegawai tidak dalam keadaan sakit(rawat inap), dinas ke luar kota, atau cuti.
5. Penjadwal menerima hasil pengecekan data pegawai.
6. Sistem akan mencetak laporan kegiatan upacara.

4.1.3.2 *Use Case Diagram*

Use case diagram adalah diagram yang menyajikan interaksi antara *use case* dan aktor. Dimana aktor dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai (*user*), bukan berdasarkan alur sebuah kejadian.



Gambar 4.2 *Use Case Diagram* Penjadwalan yang Diusulkan

Tabel 4.1 Daftar Deskripsi *Use Case* Usulan

Kode *Use Case*	Nama *Use Case*	Deskripsi
UC-JKU-01	*Login*	Untuk Mengakses Aplikasi
UC-JKU-02	Pengolahan Data Divisi	Menambah, menghapus, mengubah data divisi yang pegawai nya akan mengikuti kegiatan upacara.
UC-JKU-03	Pengolahan Data Pegawai	Menambah, menghapus, mengubah data pegawai yang pegawai akan menjadi anggota upacara.
UC-JKU-04	Pengolahan Data Kegiatan	Menambah, menghapus, mengubah data kegiatan upacara.
UC-JKU-05	Penjadwalan	Membuat jadwal pelaksanaan kegiatan upacara berdasarkan pegawai dari setiap divisi dan data kegiatan pelaksanaan kegiatan upacara.
UC-JKU-06	Pemberian Akses	Menambah, menghapus, mengubah data akses yang akan diberikan.
UC-JKU-07	Pengecekan	Pengecekan terhadap pegawai yang akan menjadi anggota upacara apakah pegawai tersebut tidak dalam keadaan sakit(rawat inap), dinas ke luar kota, atau cuti
UC-JKU-08	Laporan	Membuat dan mencetak laporan / jadwal yang berkaitan dengan kegiatan upacara

4.1.3.3 Skenario *Use Case*

Skenario *Use Case* digunakan untuk memudahkan dalam menganalisa skenario yang akan gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut. Setiap *use case* di atas harus dideskripsikan dalam dokumen. Dokumen ini merupakan definisi apa yang harus dilakukan oleh sistem ketika actor mengaktifkan *use case*. Berikut ini adalah dokumentasi *use case* untuk *Use Case Diagram* Penjadwalan Kegiatan Upacara yang diusulkan oleh penulis.

Tabel 4.2 Skenario *Use Case Login*

No.*Use Case*	UC-JKU-01
Nama *Use Case*	*Login*
Deskripsi *Use Case*	*Use case* ini memungkin kan pengguna untuk masuk dan mengakses ke dalam aplikasi
Aktor	Penjadwal, Petugas Divisi.
Kondisi Awal	Penjadwal dan Petugas Divisi membuka aplikasi
Main Flow	
Aktor	**Reaksi Sistem**
1. Aktor membuka aplikasi.	2. Menampilkan *form login*. Verifikasi *username* dan *password* di database.
3. Aktor mengisi *username* dan *password.*	4. Verifikasi *username* dan *password* di *database.*
	5. Aplikasi menampilkan menu utama.
Skenario Alternatif-Validasi Kegagalan	
1. Salah satu atau kedua field masih kosong.	2. Keluar pesan : Data masih kosong.

3. Aktor memasukan *username* dan *password* yang salah.	4. Keluar pesan : *username* atau *password* yang dimasukan salah.

Tabel 4.3 Skenario *Use Case* Pengelolaan Data Divisi

No.*Use Case*	UC-JKU-02
Nama *Use Case*	Pengelolaan Data Divisi
Deskripsi *Use Case*	*Use Case* ini memungkinkan semua proses pengolahan data divisi seperti menambah, hapus dan edit. Data-data tersebut adalah data divisi yang pegawainya mengikuti kegiatan upacara.
Aktor	Penjadwal.
Kondisi Awal	1. Aktor sudah *login* ke aplikasi 2. Menu utama untuk admin.
Main Flow	
Aktor	**Reaksi Sistem**
1. Aktor memilih submenu Data pada menu utama untuk admin.	2. Menampilkan *form* yang dipilih oleh aktor.
3. Melakukan pengolahan data.	
4. Menekan tombol tambah, mengisi field pada *form* yang dipilih kemudian menyimpan data yang ingin ditambahkan.	
5. Memilih data di table pada *form* yang dipilih untuk diedit.	
	6. Aplikasi memeriksa field apakah masih kosong atau tidak, jika ya tampil pesan field masih kosong, Jika tidak sistem melakukan proses penambahan data pada database, jika berhasil tampil

	pesan data berhasil ditambah atau diedit dan hasilnya tampil dalam tabel pada *form*.
	7. Aplikasi menyimpan hasil pengolahan data ke database.
Skenario Alternatif-Validasi Kegagalan	
1. Aktor menyimpan data dengan menekan sebuah button tanpa mengisi field-field pada *form*.	2. *Form* pengisian data ditampilkan dan ada pesan kesalahan yang menyatakan field harus diisi.
	3. Data tidak ditambahkan, diedit, atau dihapus pada database.

Tabel 4.4 Skenario *Use Case* Pengelolaan Data Pegawai

No.*Use Case*	UC-JKU-03
Nama *Use Case*	Pengelolaan Data Pegawai
Deskripsi *Use Case*	*Use Case* ini memungkinkan semua proses pengolahan data pegawai seperti menambah, hapus dan edit. Data-data tersebut adalah data pegawai yang akan menjadi anggota upacara.
Aktor	Penjadwal.
Kondisi Awal	1. Aktor sudah *login* ke aplikasi 2. Menu utama untuk admin.
Main Flow	
Aktor	**Reaksi Sistem**
1. Aktor memilih submenu Data pada menu utama untuk admin.	2. Menampilkan *form* yang dipilih oleh actor.
3. Menekan tombol tambah, mengisi field pada *form* yang dipilih kemudian menyimpan data yang ingin ditambahkan.	
4. Memilih data di table pada *form*	

yang dipilih untuk diedit.	
	5. Aplikasi memeriksa field apakah masih kosong atau tidak, jika ya tampil pesan field masih kosong, Jika tidak sistem melakukan proses penambahan data pada database, jika berhasil tampil pesan data berhasil ditambah atau diedit dan hasilnya tampil dalam tabel pada *form*.
	6. Aplikasi menyimpan hasil pengolahan data ke database.
Skenario Alternatif-Validasi Kegagalan	
1. Aktor menyimpan data dengan menekan sebuah button tanpa mengisi field-field pada *form*.	2. *Form* pengisian data ditampilkan dan ada pesan kesalahan yang menyatakan field harus diisi.
	3. Data tidak ditambahkan, diedit, atau dihapus pada database.

Tabel 4.5 Skenario *Use Case* Pengelolaan Data Kegiatan

No.*Use Case*	UC-JKU-04
Nama *Use Case*	Pengelolaan Data Kegiatan
Deskripsi *Use Case*	*Use Case* ini memungkinkan semua proses pengolahan data pegawai seperti menambah, hapus dan edit. Data-data tersebut adalah data kegiatan pelaksanaan kegiatan upacara.
Aktor	Penjadwal.
Kondisi Awal	1. Aktor sudah *login* ke aplikasi 2. Menu utama untuk admin.
Main Flow	

Aktor	Reaksi Sistem
1. Aktor memilih submenu Data pada menu utama untuk admin.	2. Menampilkan *form* yang dipilih oleh aktor.
3. Menekan tombol tambah, mengisi field pada *form* yang dipilih kemudian menyimpan data yang ingin ditambahkan.	
4. Memilih data di tabel pada *form* yang dipilih untuk diedit.	
	5. Aplikasi memeriksa field apakah masih kosong atau tidak, jika ya tampil pesan field masih kosong, Jika tidak sistem melakukan proses penambahan data pada database, jika berhasil tampil pesan data berhasil ditambah atau diedit dan hasilnya tampil dalam tabel pada *form*.
	6. Aplikasi menyimpan hasil pengolahan data ke database.
Skenario Alternatif-Validasi Kegagalan	
1. Aktor menyimpan data dengan menekan sebuah button tanpa mengisi field-field pada form.	2. *Form* pengisian data ditampilkan dan ada pesan kesalahan yang menyatakan field harus diisi.
	3. Data tidak ditambahkan, diedit, atau dihapus pada database.

Tabel 4.6 Skenario *Use Case* Pemberian Akses

No.*Use Case*	UC-JKU-05
Nama *Use Case*	Pemberian Akses
Deskripsi *Use Case*	*Use Case* ini memungkinkan semua proses pengolahan data akses seperti menambah, hapus dan edit. Data-data tersebut adalah data hak akses bagi pengguna untuk menggunakan aplikasi.
Aktor	Penjadwal.
Kondisi Awal	1. Aktor sudah *login* ke aplikasi 2. Menu utama untuk admin.
Main Flow	
Aktor	**Reaksi Sistem**
1. Aktor memilih submenu Data pada menu utama untuk admin.	2. Menampilkan *form* yang dipilih oleh aktor.
3. Menekan tombol tambah, mengisi field pada *form* yang dipilih kemudian menyimpan data yang ingin ditambahkan.	
4. Memilih data di table pada *form* yang dipilih untuk diedit.	
	5. Aplikasi memeriksa field apakah masih kosong atau tidak, jika ya tampil pesan field masih kosong, Jika tidak sistem melakukan proses penambahan data pada database, jika berhasil tampil pesan data berhasil ditambah atau diedit dan hasilnya tampil dalam tabel pada *form*.
	6. Aplikasi menyimpan hasil pengolahan data ke database.

Skenario Alternatif-Validasi Kegagalan	
1. Aktor menyimpan data dengan menekan sebuah button tanpa mengisi field-field pada *form*.	2. *Form* pengisian data ditampilkan dan ada pesan kesalahan yang menyatakan field harus diisi.
	3. Data tidak ditambahkan, diedit, atau dihapus pada database.

Tabel 4.7 Skenario *Use Case* Penjadwalan

No.*Use Case*	UC-JKU-06
Nama *Use Case*	Penjadwalan
Deskripsi *Use Case*	*Use Case* ini ditujukan untuk membuat penjadwalan kegiatan upacara.
Aktor	Penjadwal.
Kondisi Awal	1. Aktor sudah *login* ke aplikasi 2. Menu utama untuk admin.
Main Flow	
Aktor	**Reaksi Sistem**
1. Aktor memilih submenu Penjadwalan pada menu utama untuk admin.	2. Menampilkan *form* yang dipilih oleh aktor.
3. Menekan tombol isi data, mengisi field serta pegawai yang ada pada tabel pegawai pada *form* yang dipilih kemudian menyimpan data yang ingin diproses.	
4. Menekan tombol ganti, Memilih data di table anggota upacara pada *form* yang dipilih untuk diganti dengan pegawai lain.	

	5. Aplikasi memeriksa apakah pegawai yang menjadi anggota upacara pada divisi yang divisi yang dipilih melebihi kuota yang ditentukan atau tidak, jika ya tampil pesan kuota untuk divisi tersebut sudah terpenuhi, Jika tidak sistem melakukan proses penambahan data pada database, jika berhasil tampil pesan data input berhasil.
	6. Aplikasi menyimpan hasil pengolahan data ke database.
Skenario Alternatif-Validasi Kegagalan	
1. Aktor menyimpan data dengan menekan sebuah button tanpa mengisi field-field pada *form*.	2. *Form* pengisian data ditampilkan dan ada pesan kesalahan yang menyatakan field harus diisi.
	3. Data tidak ditambahkan, diedit, atau dihapus pada database.

Tabel 4.8 Skenario *Use Case* Pengecekan

No.*Use Case*	UC-JKU-07
Nama *Use Case*	Pengecekan
Deskripsi *Use Case*	*Use Case* ini ditujukan untuk men-cek pegawai apakah pegawai tidak dalam keadaan sakit(rawat inap), dinas ke luar kota, atau cuti. .
Aktor	Petugas Divisi.
Kondisi Awal	1. Aktor sudah *login* ke aplikasi 2. Menu utama untuk divisi
Main Flow	
Aktor	**Reaksi Sistem**

1. Aktor memilih submenu Laporan pada menu utama untuk divisi	2. Menampilkan *form* yang dipilih oleh aktor.
3. Pada menu utama Divisi, aktor memilih no.kegiatan.	
	4. Menampilkandata yang dipilih

Tabel 4.9 Skenario *Use Case* Laporan

No.*Use Case*	UC-JKU-08
Nama *Use Case*	Laporan
Deskripsi *Use Case*	*Use Case* ini ditujukan untuk melihat jadwal kegiatan upacara yang sudah dibuat
Aktor	Penjadwal, Petugas Divisi.
Kondisi Awal	1. Aktor sudah *login* ke aplikasi. 2. Menu utama untuk admin. 3. Menu utama untuk divisi.
Main Flow	
Aktor	**Reaksi Sistem**
1. Aktor memilih submenu Laporan pada menu utama untuk admin dan menu utama untuk divisi	2. Menampilkan *form* yang dipilih oleh aktor.
3. Pada menu utama Admin, actor penjadwal memilih no.kegiatan lalu menekan tombol print, untuk melihat seluruh data pegawai yang menjadi anggota upacara, dan pada menu utama Divisi actor petugas divisi memilih no.kegiatan lalu menekan tombol print, untuk melihat data pegawai yang	

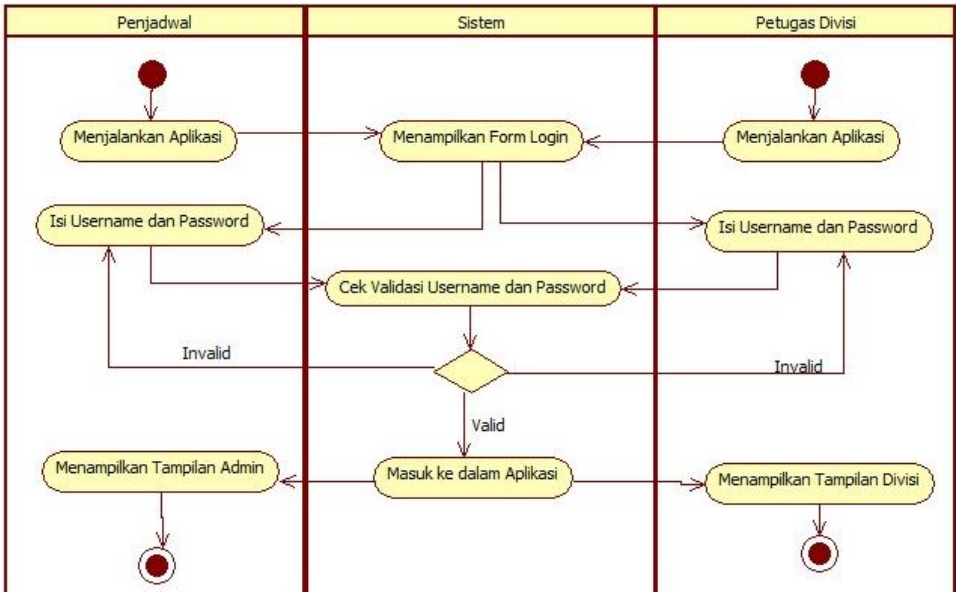
menjadi anggota upacara pada divisi tersebut tersebut.	
	4. Menampilkan data yang akan di print.

4.1.3.4 *Activity Diagram*

Activity diagram menggambarkan aliran fungsionalitas dari sistem baru yang dibuat. Dalam diagram ini akan digambarkan berbagai aliran aktivitas dalam sistem baru, yang bertujuan untuk mengetahui alur proses pada sistem yang diusulkan.

1. *Activity Diagram Login*

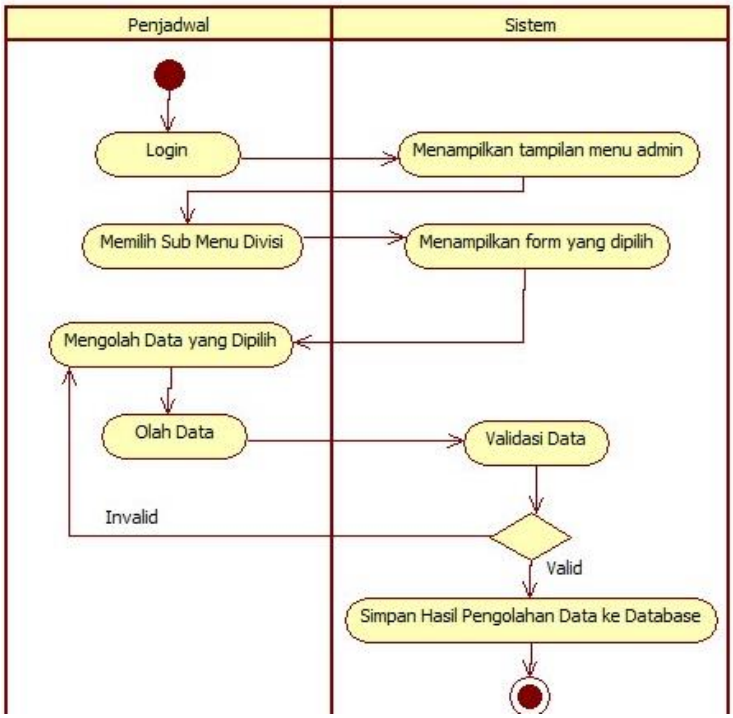
   Aktivitas diagram *login* yang diusulkan adalah sebagai berikut :



Gambar 4.3 *Activity Diagram Login*

2. *Activity Diagram* Pengelolaan Data Divisi

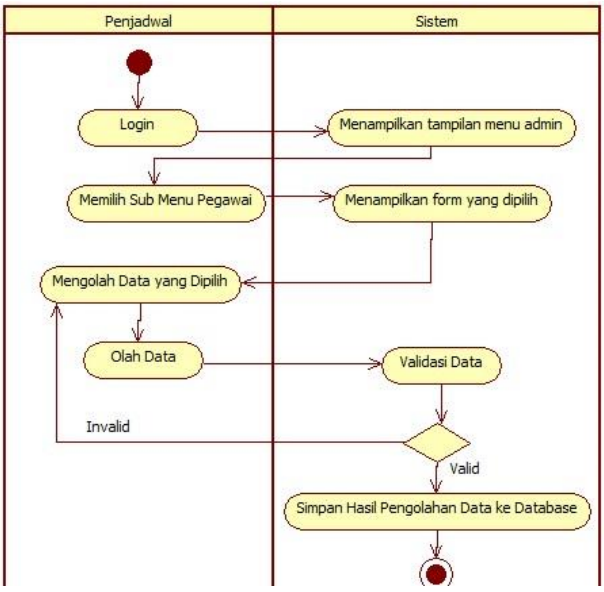
   Aktivitas diagram pengelolaan data divisi yang diusulkan adalah sebagai berikut :



Gambar 4.4 *Activity Diagram* Pengelolaan Data Divisi

3. *Activity Diagram* Pengelolaan Data Pegawai

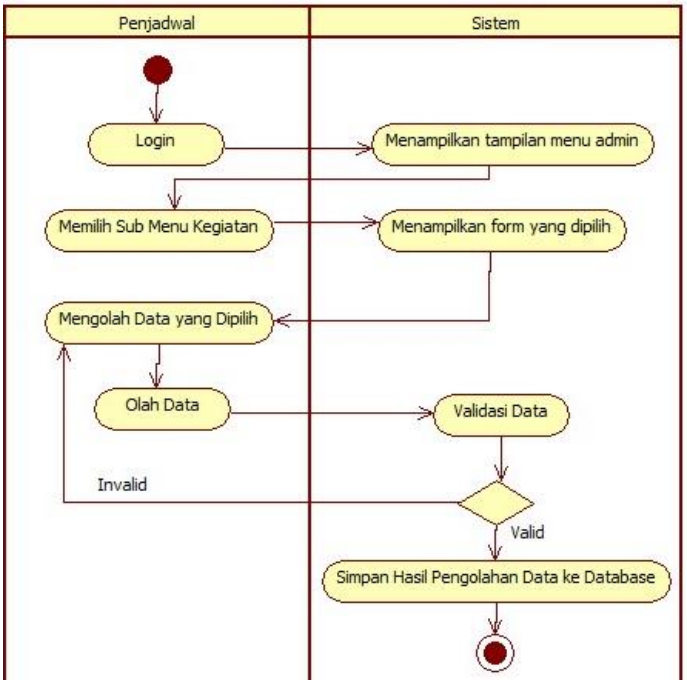
Aktivitas diagram pengelolaan data pegawai yang diusulkan adalah sebagai berikut :



Gambar 4.5 *Activity Diagram* Pengelolaan Data Pegawai

4. *Activity Diagram* Pengelolaan Data Kegiatan

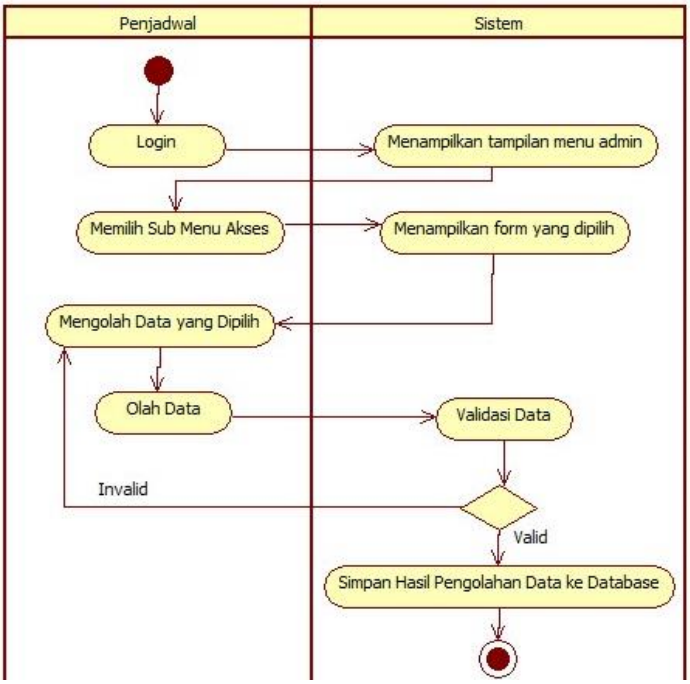
Aktivitas diagram data kegiatan yang diusulkan adalah sebagai berikut:



Gambar 4.6 *Activity Diagram* Pengelolaan Data Kegiatan

5. *Activity Diagram* Pemberian Akses

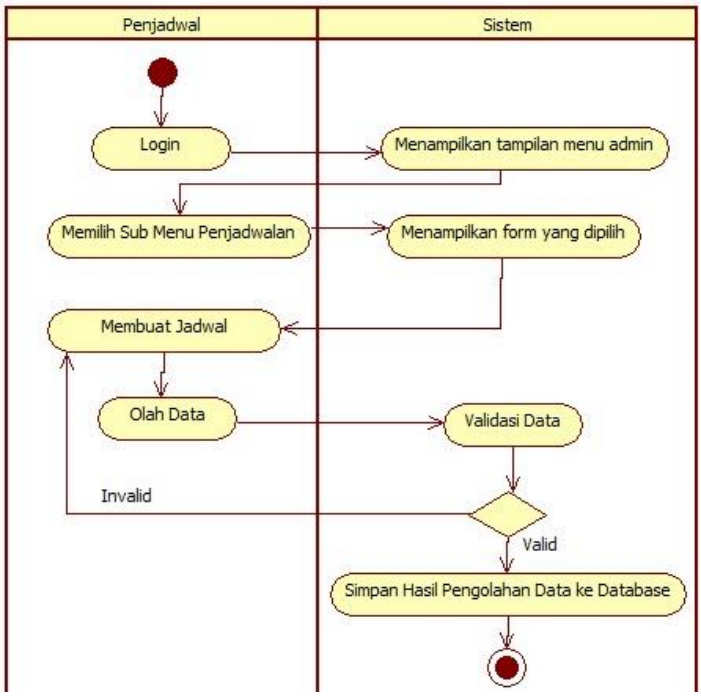
Aktivitas diagram pemberian akses yang diusulkan adalah sebagai berikut :



Gambar 4.7 *Activity Diagram* Pemberian Akses

6. *Activity Diagram* Penjadwalan

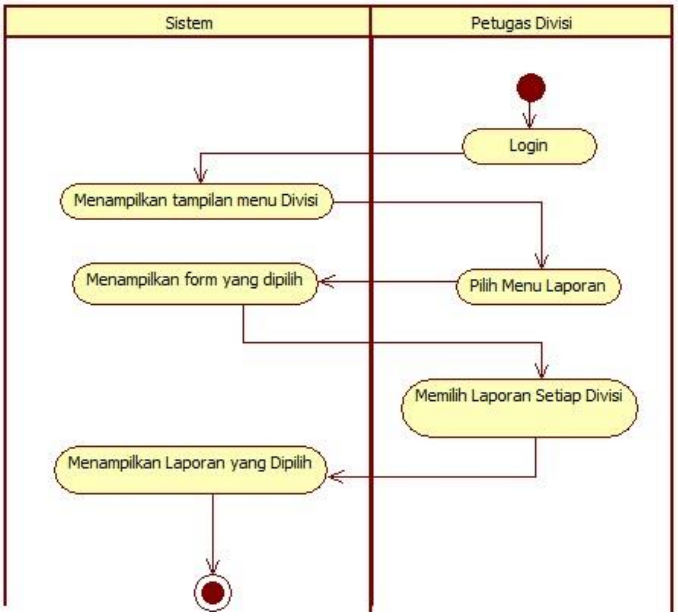
Aktivitas diagram penjadwalan diusulkan adalah sebagai berikut:



Gambar 4.8 *Activity Diagram* Penjadwalan

7. *Activity Diagram* Pengecekan

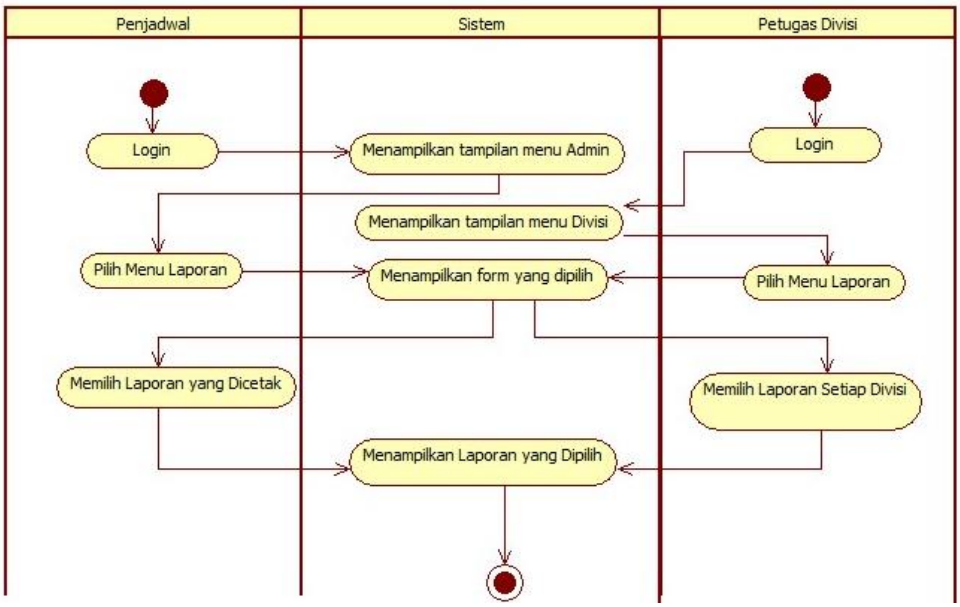
Aktivitas diagram yang diusulkan adalah sebagai berikut :



Gambar 4.9 *Activity Diagram* Pengecekan

8. *Activity Diagram* Laporan

Aktivitas diagram laporan yang diusulkan adalah sebagai berikut :



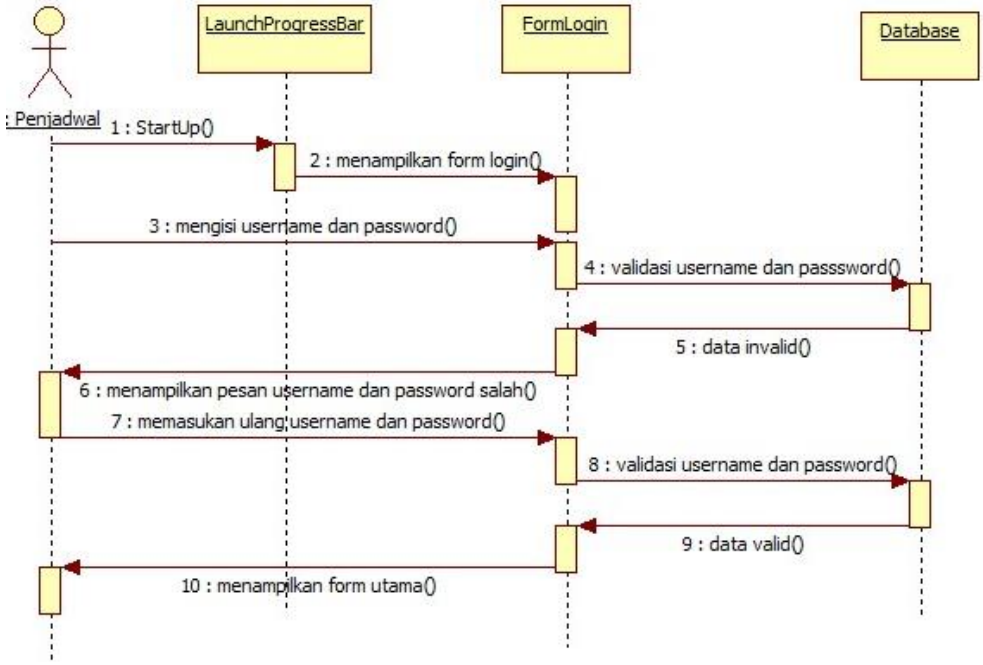
Gambar 4.10 *Activity Diagram* Laporan

4.1.3.5 *Sequence Diagram*

Sequence Diagram adalah *interaction diagram* yang memperlihatkan event-event yang berurutan sepanjang berjalannya waktu. Masing-masing *Sequence Diagram* akan menggambarkan aliran-aliran pada suatu *use case*. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antar object, juga interaksi antar objek, dan menunjukkan sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama, pesan diwakili oleh garis dengan tanda panah, dan waktu yang ditunjukkan dengan proses vertikal. Berikut adalah *sequence diagram* yang ada pada aplikasi penjadwalan kegiatan upacara.

1. *Sequence Diagram Login*

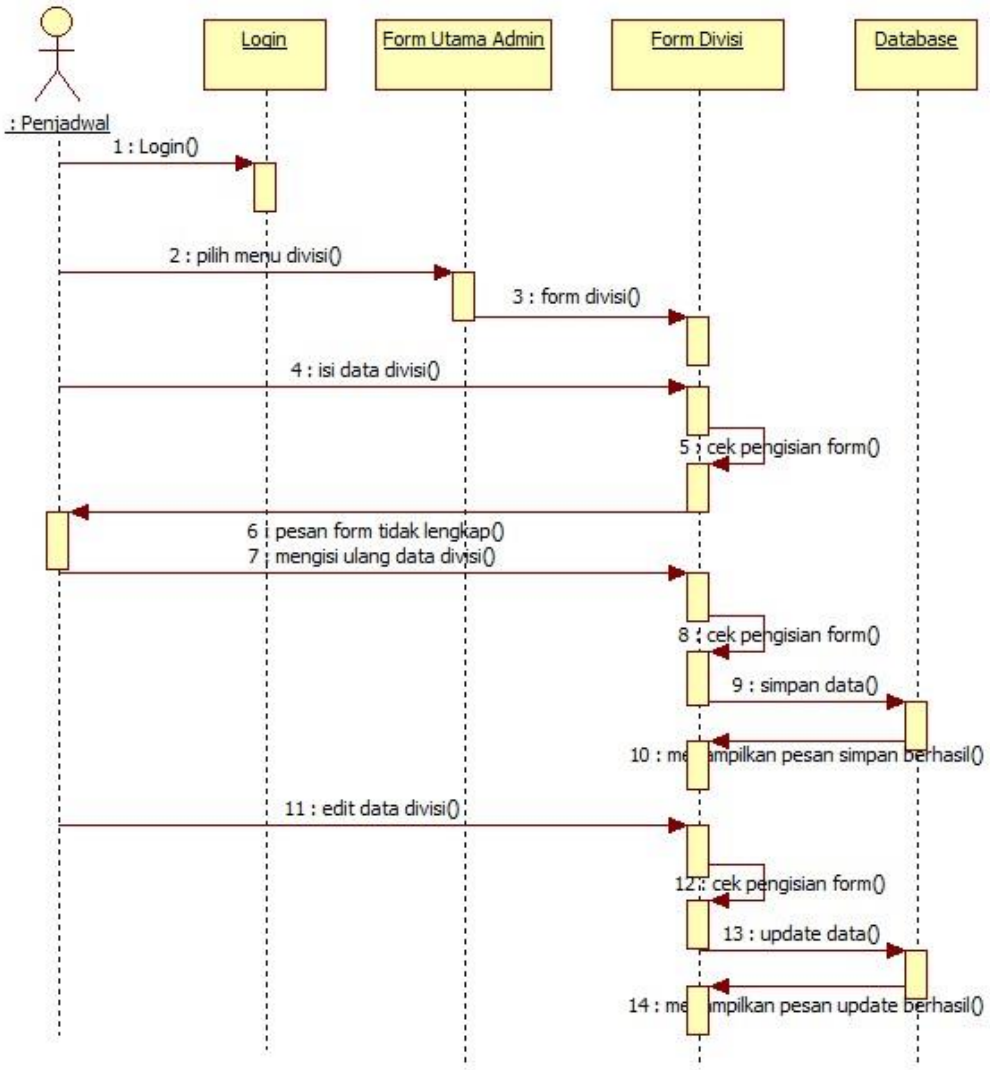
   Berikut merupakan *sequence diagram* untuk *login* :



Gambar 4.11 *Sequence Diagram Login*

2. *Sequence Diagram* Pengelolaan Data Divisi

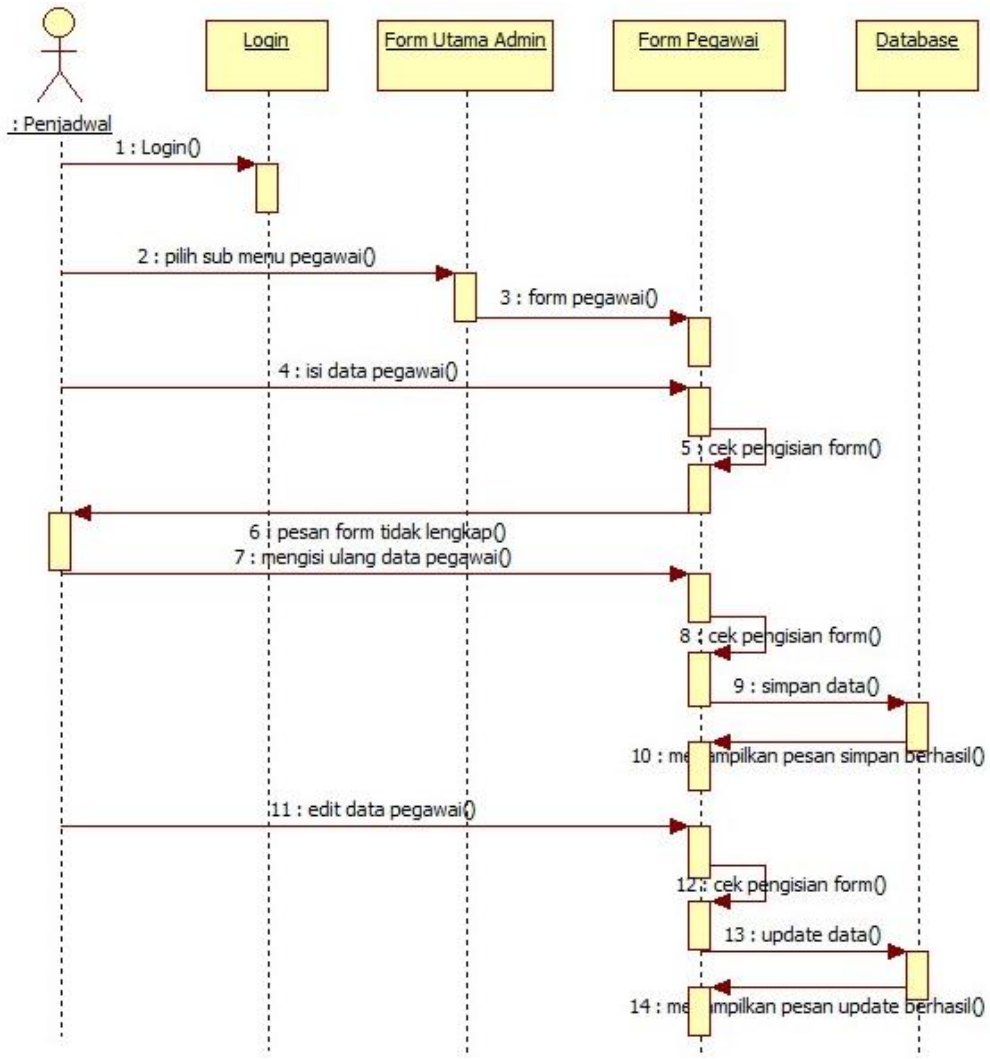
   Berikut merupakan *sequence diagram* untuk pengelolaan data divisi :



Gambar 4.12 *Sequence Diagram* Pengelolaan Data Divisi

3. *Sequence Diagram* Pengelolaan Data Pegawai

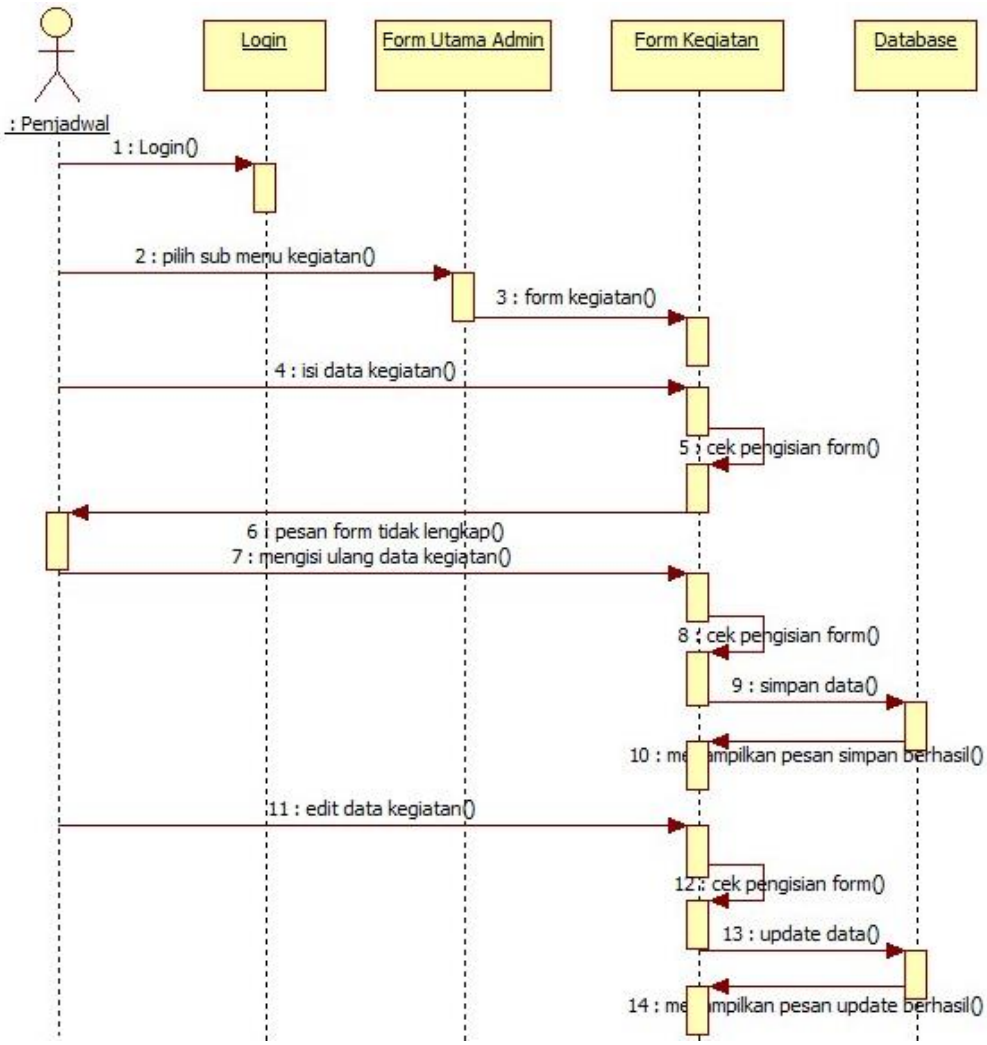
   Berikut merupakan *sequence diagram* untuk pengelolaan data pegawai :



Gambar 4.13 *Sequence Diagram* Pengelolaan Data Pegawai

4. *Sequence Diagram* Pengelolaan Data Kegiatan

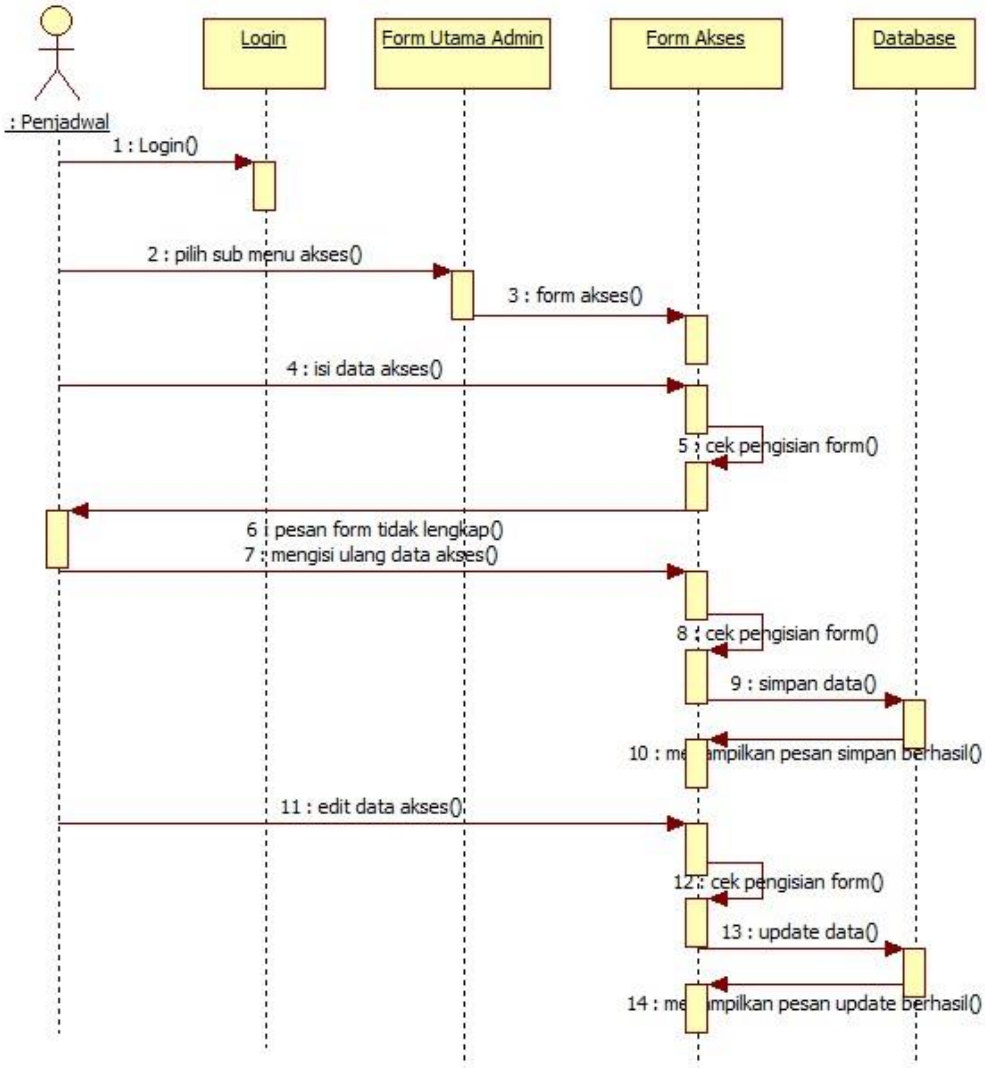
   Berikut merupakan *sequence diagram* untuk pengelolaan data kegiatan :



Gambar 4.14 *Sequence Diagram* Pengelolaan Data Kegiatan

5. *Sequence Diagram* Pemberian Akses

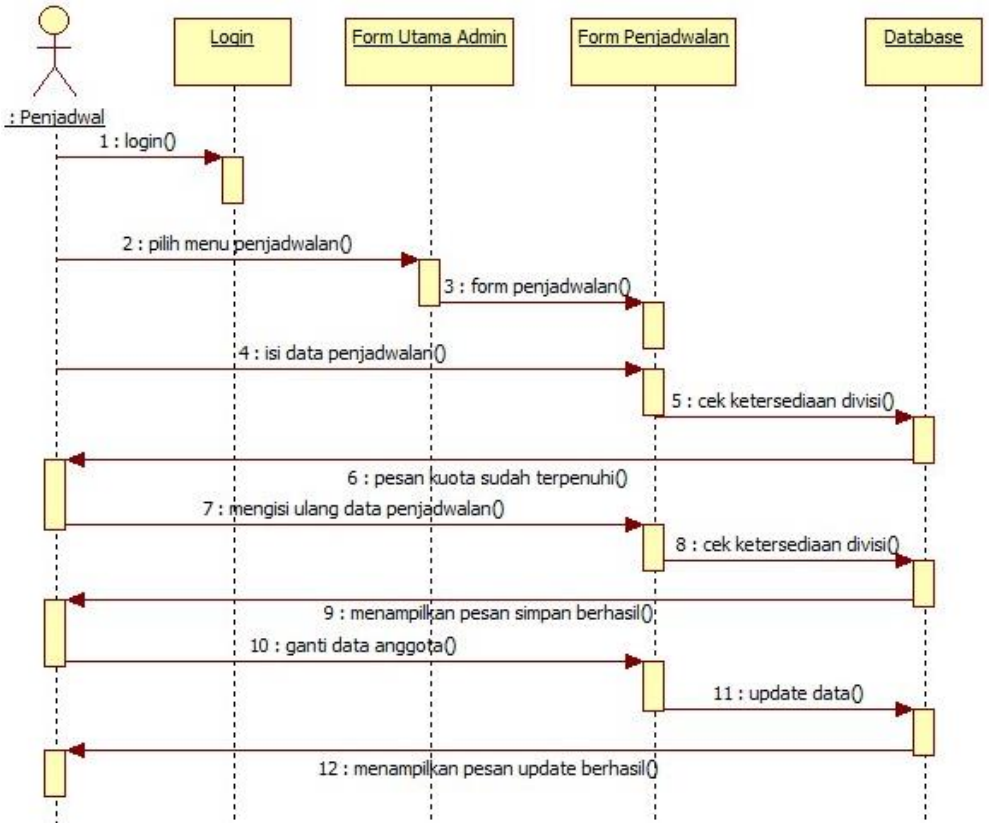
Berikut merupakan *sequence diagram* untuk pemberian akses :



Gambar 4.15 *Sequence Diagram* Pemberian Akses

6. *Sequence Diagram* Penjadwalan

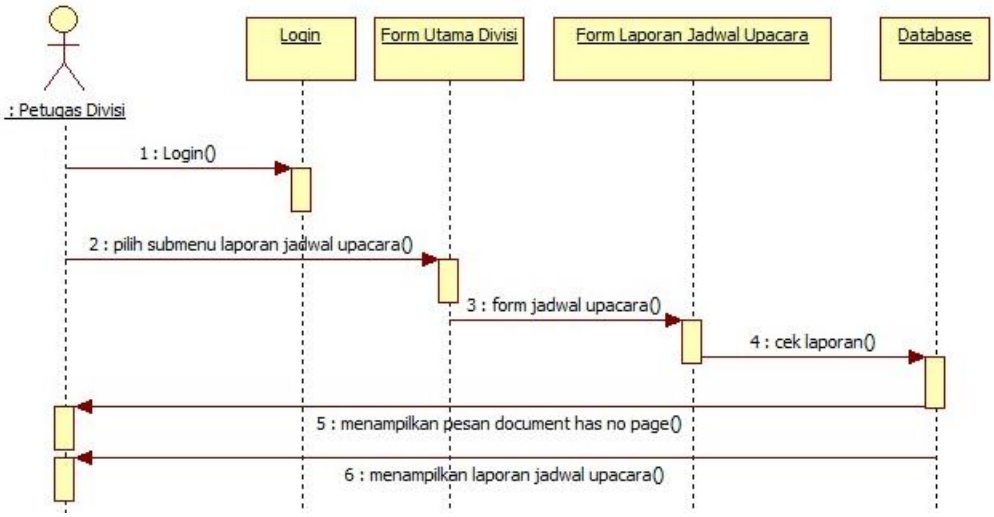
Berikut merupakan *sequence diagram* untuk penjadwalan :



Gambar 4.16 *Sequence Diagram* Penjadwalan

7. *Sequence Diagram* Pengecekan

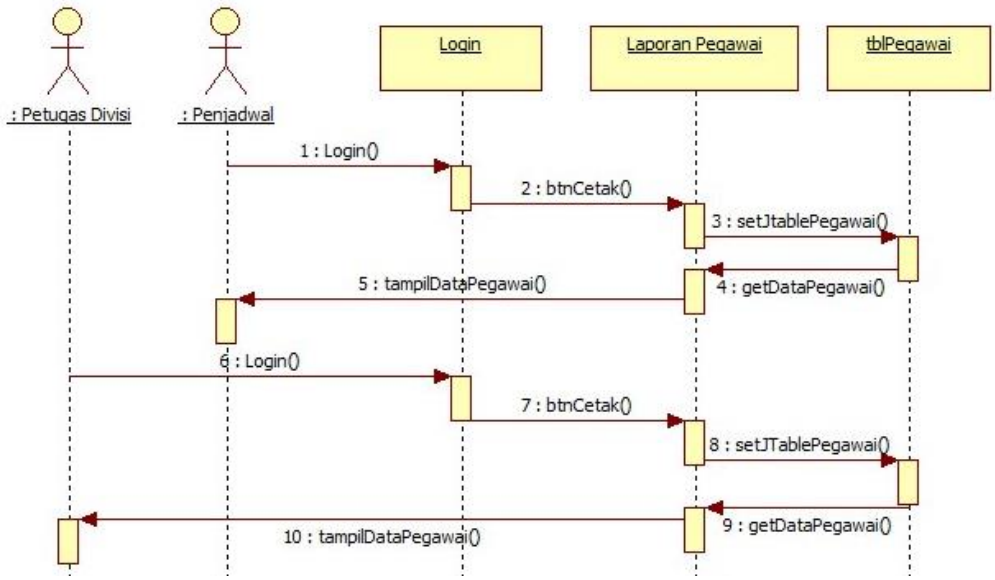
Berikut merupakan *sequence diagram* untuk pengecekan :



Gambar 4.17 *Sequence Diagram* Pengecekan

8. *Sequence Diagram* Laporan Data Pegawai

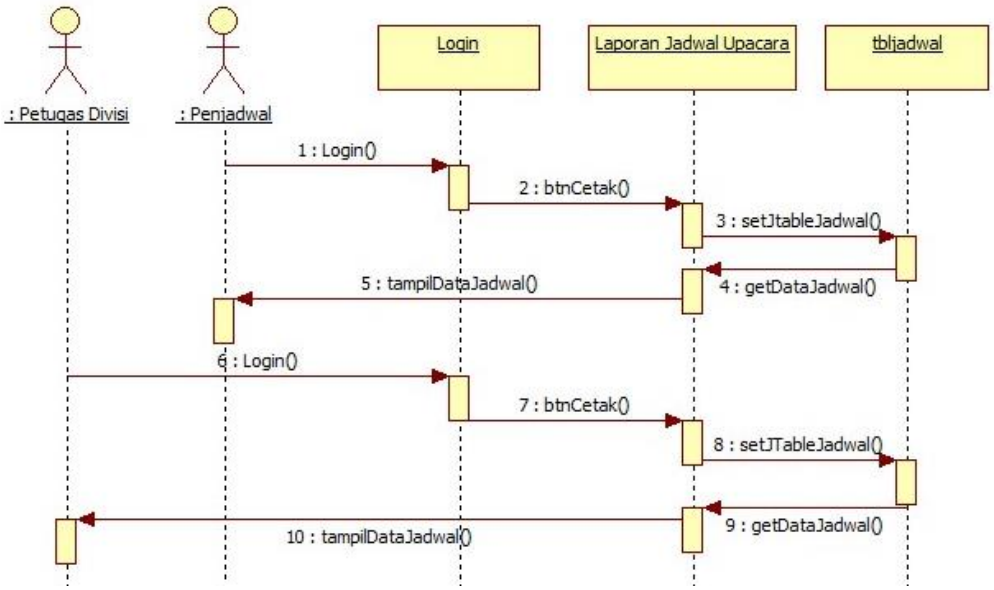
Berikut merupakan *sequence diagram* untuk laporan data pegawai :



Gambar 4.18 *Sequence Diagram* Laporan Data Pegawai

9. *Sequence Diagram* Laporan Jadwal Upacara

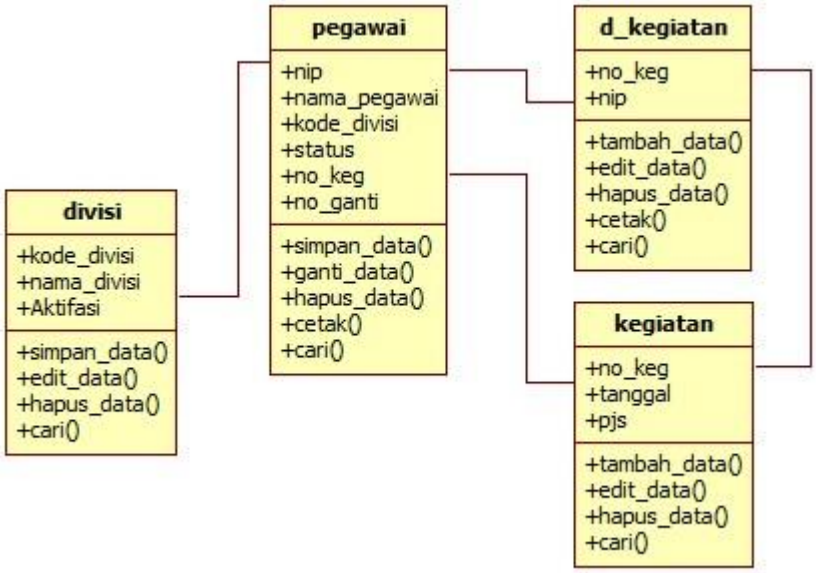
Berikut merupakan *sequence diagram* untuk laporan jadwal upacara :



Gambar 4.19 *Sequence Diagram* Laporan Jadwal Upacara

4.1.3.6 *Class Diagram*

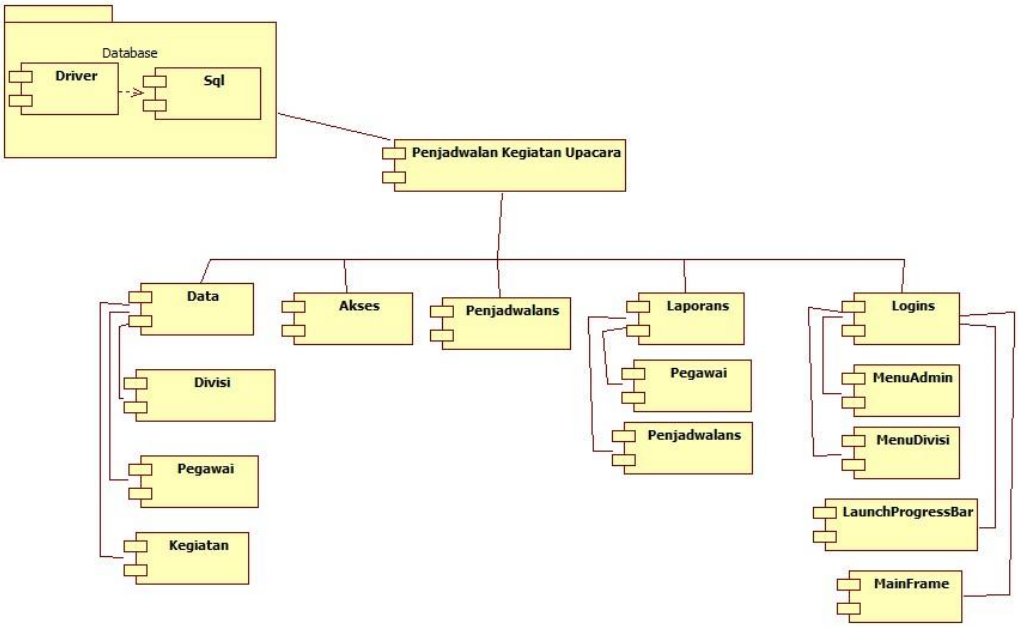
Class diagram mendepkripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. Class diagram juga menunjukan property dan operasi sebuah class dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. Berikut *Class Diagram* Penjadwalan Kegiatan Upacara yang diusulkan.



Gambar 4.20 *Class Diagram*

4.1.3.7 *Component Diagram*

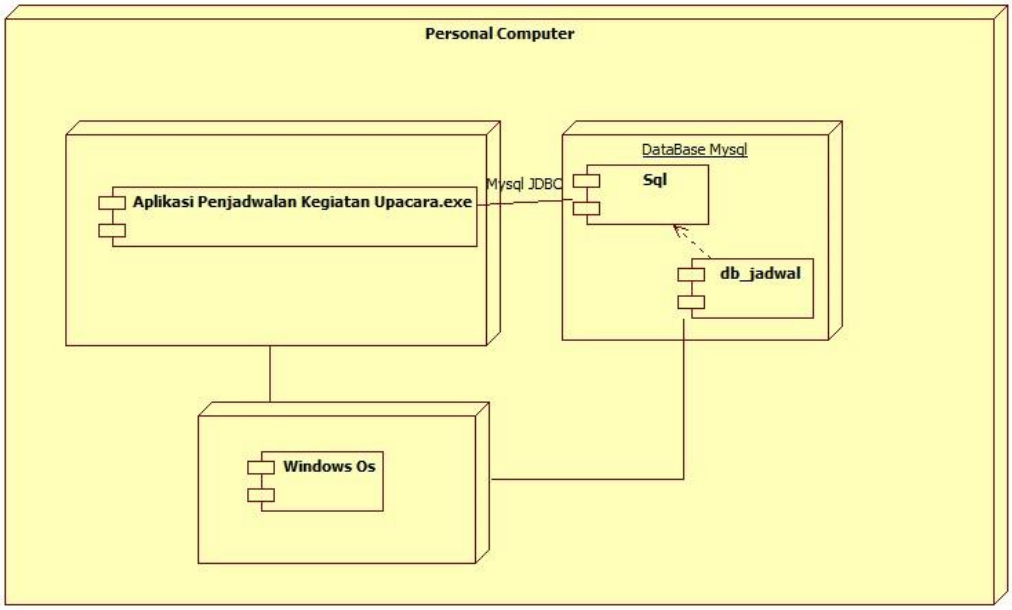
Component diagram menggambarkan modul *software* yang bersama-sama membangun sistem. Komponen-komponen dipetakan ke masing-masing *class* sesuai dengan bahasa untuk implementasi dan *source code*-nya. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan. Berikut diagram komponen dari perancangan aplikasi penjadwalan kegiatan upacara.



Gambar 4.21 *Component Diagram*

4.1.3.8 *Deployment Diagram*

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian software yang berjalan pada bagian-bagian hardware yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen hardware tersebut. Jadi penggambaran arsitektur fisik sebuah aplikasi yang melibatkan perangkat, baik perangkat lunak maupun perangkat keras yang disebut dengan *node* dan menunjukkan bagaimana komponen perangkat lunak dan keras ini bekerja sama akan digambarkan dalam deployment diagram.



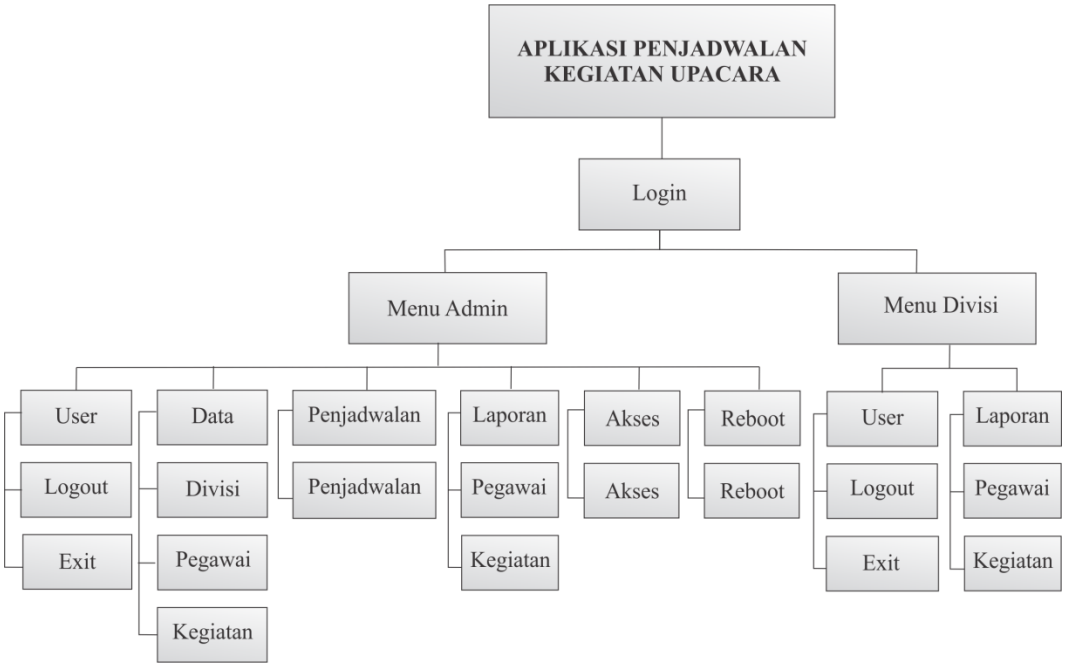
Gambar 4.22 *Deployment Diagram*

4.2 Perancangan Antar Muka

Perancangan antar muka merupakan tahapan untuk membuat tampilan atau design dari sistem yang akan dibuat. Rancangan tampilan yang dibuat meliputi rancangan input dan rancangan output dari sistem yang akan dibuat. Perancangan perancangan antar muka pada Aplikasi Penjadwalan Kegiatan Upacara pada PT.Pindad adalah sebagai berikut.

4.2.1 Struktur Menu

Perancangan menu dibuat sebagai alat antar muka dengan pengguna untuk memudahkan pengoperasian aplikasi. Berikut rancangan menu aplikasi ini :



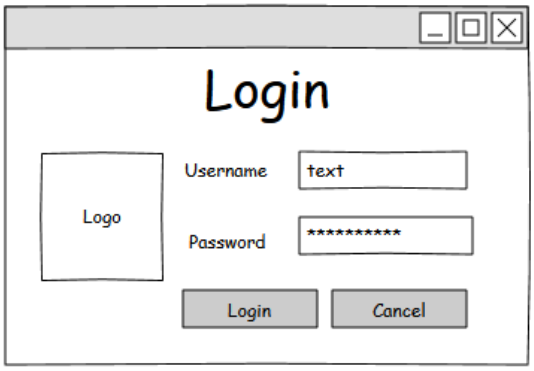
Gambar 4.23 Struktur Menu

4.2.2 Perancangan *Input*

Perancangan *input* diperlukan untuk menentukan tampilan program yang berfungsi sebagai tempat memasukan data. Perancangan input merupakan perancangan tampilan yang akan digunakan guna memasukkan data pada sistem untuk kemudian diproses. Dalam perancangan input ini, data yang dimasukkan akan mempengaruhi hasil yang ditampilkan. Adapun perancangan-perancangan input yang ada dalam perancangan ini adalah :

1. *Form Login*

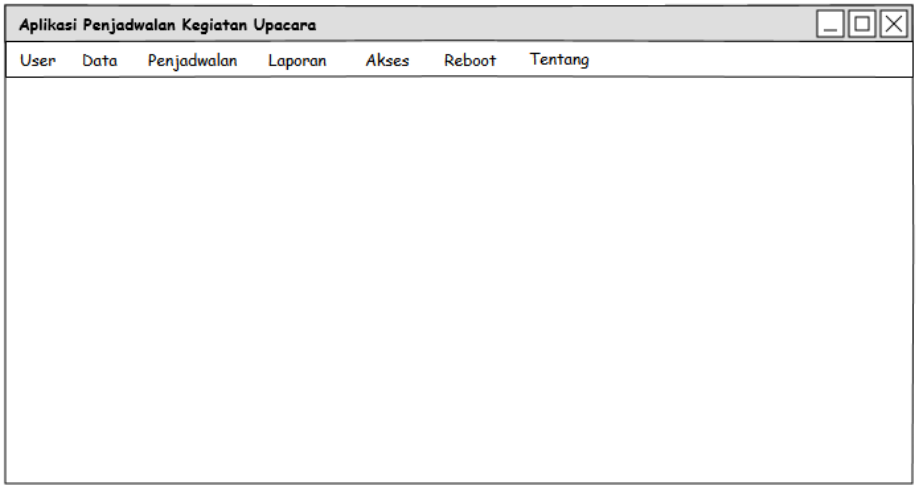
   Rancangan *form* ini berfungsi untuk menginputkan *username* dan *password* ketika kita akan *login*.



Gambar 4.24 Perancangan *Form Login*

2. *Form* Menu Admin

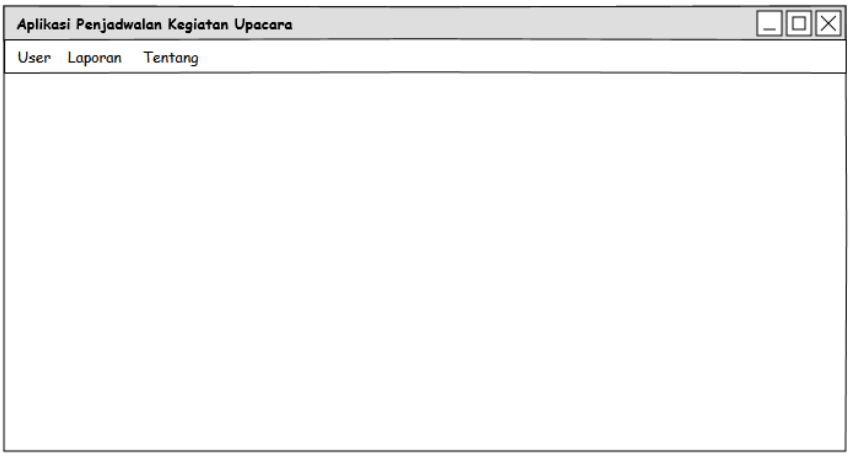
   Rancangan *form* ini berfungsi sebagai halaman utama untuk admin.



Gambar 4.25 Perancangan *Form* Menu Admin

3. *Form* Menu Divisi

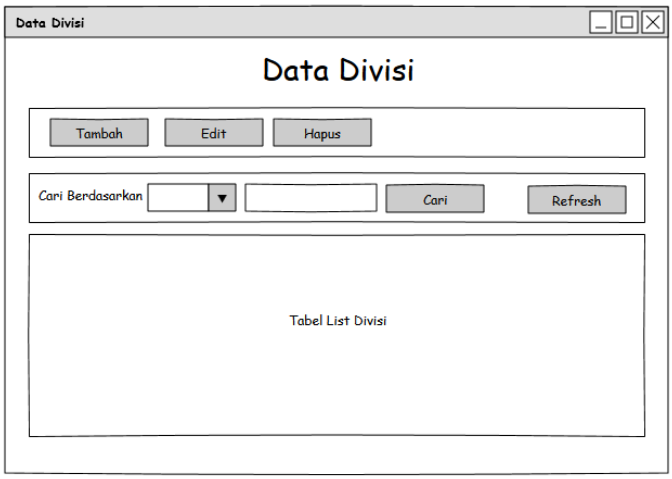
   Rancangan *form* ini berfungsi sebagai halaman utama untuk Divisi.



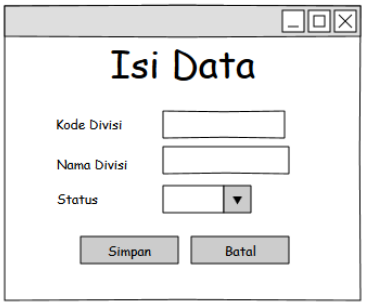
Gambar 4.26 Perancangan *Form* Menu Divisi

4. *Form* Data Divisi

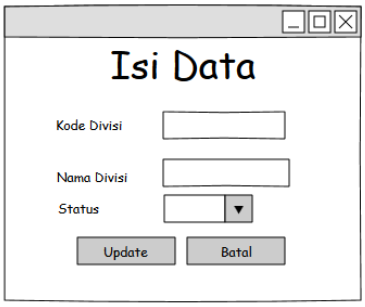
   Rancangan *form* ini berfungsi untuk menginputkan data divisi yang mengikuti kegiatan upacara. Di mana didalamnya terdapat *fields* untuk mengisikan kode divisi, serta nama divisi.



Gambar 4.27 Perancangan *Form* Data Divisi



Gambar 4.28 Perancangan Dialog Tambah Data Divisi



Gambar 4.29 Perancangan Dialog Edit Data Divisi

Keterangan :

a. *Field* Kode Divisi di isi secara manual dari data divisi yang ada.
b. *Field* Nama Divisi di isi secara manual dari data divisi yang ada.
c. *Combo box* status di isi secara optional.
d. Tombol Tambah digunakan untuk menambah data divisi, sebelum menekan tombol ini, dialog tambah data divisi tidak akan muncul. Setelahnya, maka untuk menyimpan data yang telah ditambah, tekan tombol Simpan.
e. Tombol Edit digunakan untuk mengedit data nama divisi yang dipilih dari table, setelah menekan tombol edit maka akan muncul dialog edit data divisi. Untuk menyimpan data yang telah diedit, tekan tombol Update.

f. Tombol Hapus digunakan untuk menghapus data divisi yang dipilih dari tabel.

g. Tombol Batal digunakan bila ingin membatalkan pengolahan data.

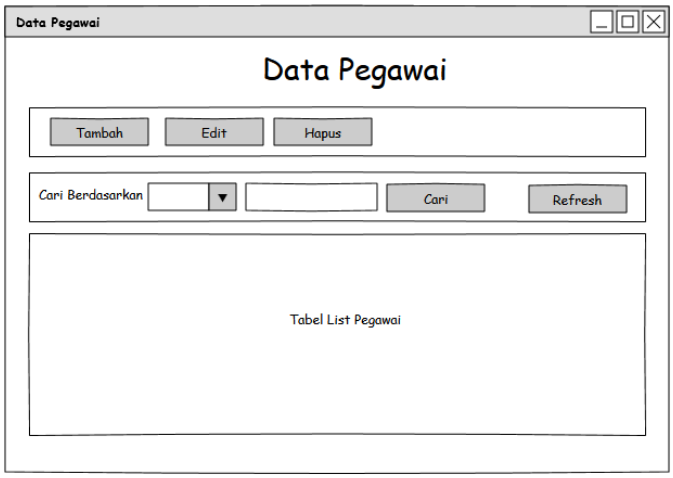
h. Tombol Cari digunakan untuk mencari data divisi dengan cara mengisi *field* berdasarkan *combo box* yang dipilih.

i. Tombol Refresh digunakan untuk merefresh data divisi yang ada pada tabel.

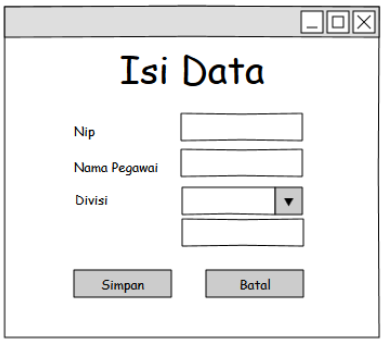
j. Tombol X yang ada di sudut kanan digunakan untuk kembali ke halaman utama untuk admin.

5. *Form* Pegawai

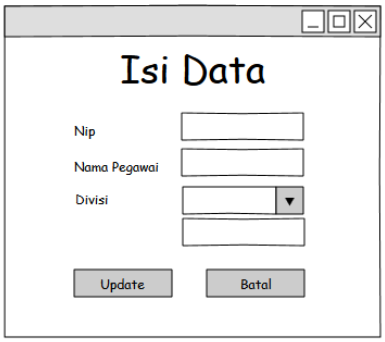
Rancangan *form* ini berfungsi untuk menginputkan data pegawai dari setiap divisi yang mengikuti kegiatan upacara. Di mana didalamnya terdapat *fields* untuk mengisikan kode pegawai, nama pegawai serta divisi dimana pegawai itu berada.



Gambar 4.30 Perancangan *Form* Pegawai



Gambar 4.31 Perancangan Dialog Tambah Data Pegawai



Gambar 4.32 Perancangan Dialog Edit Data Pegawai

Keterangan :

a. *Field* NIP di isi secara manual dari data divisi yang ada.
b. *Field* Nama Pegawai di isi secara manual dari data divisi yang ada.
c. *Combobox* Divisi menunjukkan keterangan di divisi mana pegawai yang akan di isi.
d. Tombol Tambah digunakan untuk menambah data pegawai, sebelum menekan tombol ini, dialog tambah data pegawai tidak akan muncul. Setelahnya, maka untuk menyimpan data yang telah ditambah, tekan tombol Simpan.
e. Tombol Edit digunakan untuk mengedit data nama pegawai dan divisi dimana pegawai ditempatkan yang dipilih dari table, setelah

menekan tombol edit maka akan muncul dialog edit data pegawai. Untuk menyimpan data yang telah diedit, tekan tombol Update.

f. Tombol Hapus digunakan untuk menghapus data pegawai yang dipilih dari tabel.

g. Tombol Batal digunakan bila ingin membatalkan pengolahan data.

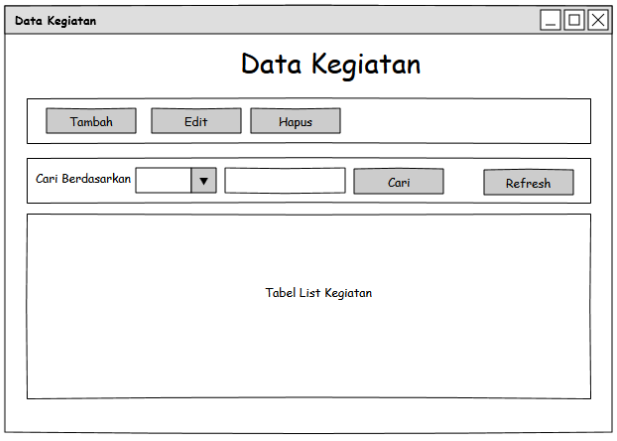
h. Tombol Cari digunakan untuk mencari data pegawai dengan cara mengisi *field* berdasarkan *combo box* yang dipilih.

i. Tombol Refresh digunakan untuk merefresh data pegawai yang ada pada tabel.

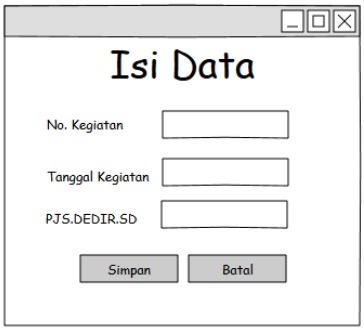
j. Tombol X yang ada di sudut kanan digunakan untuk kembali ke halaman utamauntuk admin.

6. *Form* Kegiatan

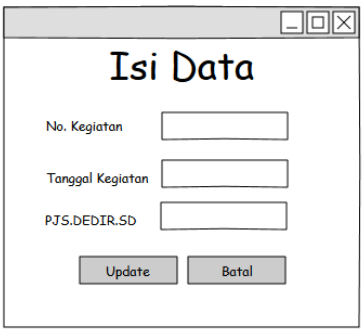
Rancangan *form* ini berfungsi untuk membuat waktu pelaksanaan kegiatan upacara dimana didalamnya terdapat *fields* No. Kegiatan yang secara otomatis akan terisi apabila menekan tombol Tambah, serta terdapat *fields* untuk mengisikan tanggal pelaksanaan kegiatan upacara.



Gambar 4.33 Perancangan *Form* Data Kegiatan



Gambar 4.34 Perancangan Dialog Tambah Data Kegiatan



Gambar 4.35 Perancangan Dialog Edit Data Kegiatan

Keterangan :

a. *Field* No. Kegiatan akan secara otomatis ditambahkan.
b. *Field* Tanggal disi dengan melihat kalender yang ada pada *field* tanggal.
c. Tombol Tambah digunakan untuk menambah data kegiatan, sebelum menekan tombol ini, dialog tambah data kegiatan tidak akan muncul. Setelahnya, maka untuk menyimpan data yang telah ditambah, tekan tombol Simpan.
d. Tombol Edit digunakan untuk mengedit data tanggal kegiatan dan yang dipilih dari table, setelah menekan tombol edit maka akan

muncul dialog edit data kegiatan. Untuk menyimpan data yang telah diedit, tekan tombol Update.

e. Tombol Hapus digunakan untuk menghapus data kegiatan yang dipilih dari tabel.

f. Tombol Batal digunakan bila ingin membatalkan pengolahan data.

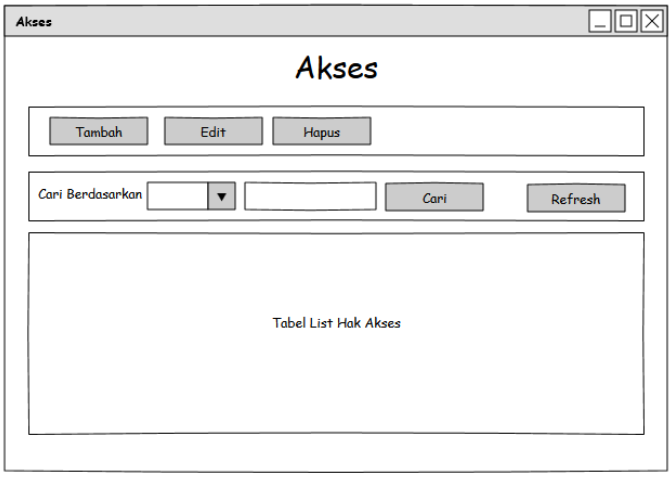
g. Tombol Cari digunakan untuk mencari data kegiatan dengan cara mengisi *field* berdasarkan *combo box* yang dipilih.

h. Tombol Refresh digunakan untuk merefresh data kegiatan yang ada pada tabel.

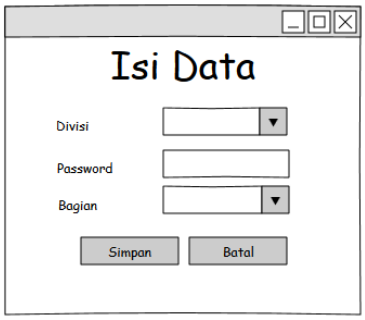
i. Tombol X yang ada di sudut kanan digunakan untuk kembali ke halaman utama untuk admin.

7. *Form* Akses

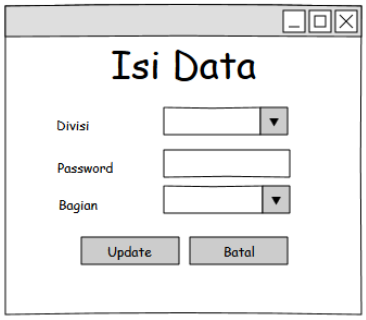
Rancangan *form* ini berfungsi untuk mengisi hak akses untuk setiap divisi yang pegawai nya akan menjadi anggota upacara.



Gambar 4.36 Perancangan *Form* Akses



Gambar 4.37 Perancangan Dialog Tambah Akses



Gambar 4.38 Perancangan Dialog Edit Akses

Keterangan :

a. *Combo box* Divisi untuk memilihi divisi yang akan diberi akses untuk menggunakan aplikasi penjadwalan kegiatan upacara.
b. *Field Password* di isi secara manual.
c. *Combobox* Bagian menunjukkan keterangan akses yang diberikan.
d. Tombol Tambah digunakan untuk menambah data akses, sebelum menekan tombol ini, dialog tambah data akses tidak akan muncul. Setelahnya, maka untuk menyimpan data yang telah ditambah, tekan tombol Simpan.
e. Tombol Edit digunakan untuk mengedit data *password* dan bagian akses yang diberikan yang dipilih dari table, setelah menekan tombol

edit maka akan muncul dialog edit data pegawai. Untuk menyimpan data yang telah diedit, tekan tombol Update.

f. Tombol Hapus digunakan untuk menghapus data akses yang dipilih dari tabel.

g. Tombol Batal digunakan bila ingin membatalkan pengolahan data.

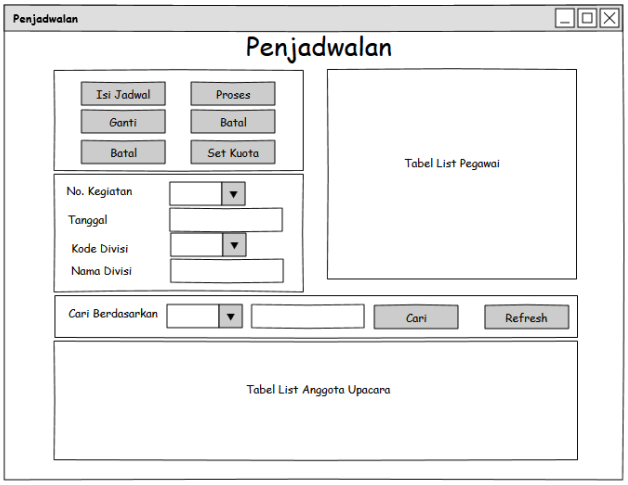
h. Tombol Cari digunakan untuk mencari data akses dengan cara mengisi *field* berdasarkan *combo box* yang dipilih.

i. Tombol Refresh digunakan untuk merefresh data akses yang ada pada tabel.

j. Tombol X yang ada di sudut kanan digunakan untuk kembali ke halaman utama untuk admin.

8. *Form* Penjadwalan

Rancangan *form* ini berfungsi untuk mengisi pegawai yang akan menjadi anggota upacara pada setiap divisi dari kegiatan yang telah diisi sebelumnya.



Gambar 4.39 Perancangan *Form* Penjadwalan

Keterangan :

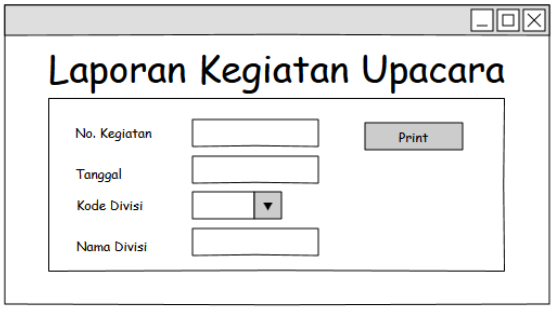
a. *Combobox* No. Kegiatan untuk memilih kegiatan upacara yang akan di ikuti oleh pegawai yang akan menjadi anggota upacara.
b. *Combobox* Divisi untuk memilih divisi yang pegawai nya akan menjadi anggota upacara.
c. Tabel list pegawai digunakan untuk memilih pegawai yang akan menjadi anggota upacara.
d. Tombol Isi Jadwal digunakan untuk mengisi data pegawai yang akan menjadi anggota upacara berdasarkan no. kegiatan dan divisi yang dipilih. Sebelum menekan tombol ini, *Combo box* pada *form* tidak akan bisa digunakan. Setelahnya, maka untuk menyimpan data yang telah ditambah, tekan tombol Proses.
e. Tombol Ganti digunakan untuk mengganti data pegawai yang telah menjadi anggota upacara dengan pegawai lain yang dipilih dari tabel anggota upacara, kemudian menekan tombol Update (Proses).
f. Tombol Hapus digunakan untuk menghapus data anggota upacara yang dipilih dari tabel.
g. Tombol Set Kuota digunakan untuk mengkondisikan berapa persen pegawai dari setiap divisi yang akan menjadi anggota upacara.
h. Tombol Batal digunakan bila ingin menbatalkan pengolahan data.
i. Tombol Cari digunakan untuk mencari data anggota upacara dengan cara mengisi *field* berdasarkan *combo box* yang dipilih.

j. Tombol Refresh digunakan untuk merefresh data anggota upacara yang ada pada tabel.

k. Tombol X yang ada di sudut kanan digunakan untuk kembali ke halaman utama untuk admin.

9. *Form* Laporan Kegiatan Upacara Admin

Rancangan *form* ini berfungsi untuk mencetak laporan kegiatan upacara.



Gambar 4.40 Perancangan *Form* Laporan Kegiatan Upacara Admin

Keterangan :

a. *Combobox* No. Kegiatan untuk memilih kegiatan upacara yang laporannya akan dicetak.

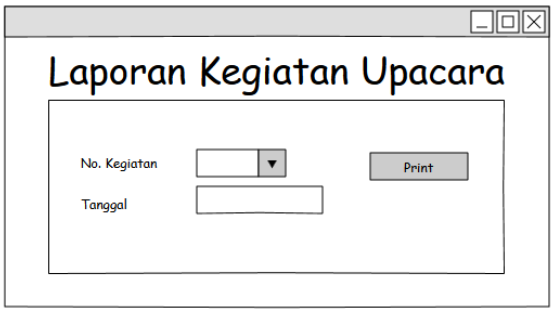
b. *Combobox* kode divisi untuk divisi mana yang laporannya akan dicetak.

c. Tombol Print digunakan untuk mencetak laporan data kegiatan dari *combo box* yang dipilih.

d. Tombol X yang ada di sudut kanan digunakan untuk kembali ke halaman utama untuk admin.

10. *Form* Laporan Kegiatan Upacara Divisi

Rancangan *form* ini berfungsi untuk mencetak laporan kegiatan upacara.



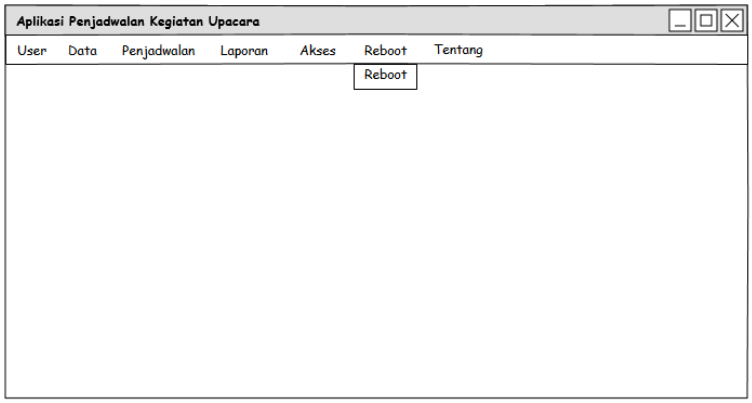
Gambar 4.41 Perancangan *Form* Laporan Kegiatan Upacara Divisi

Keterangan :

a. *Combobox* No. Kegiatan untuk memilih kegiatan upacara yang laporannya akan dicetak
b. Tombol Print digunakan untuk mencetak laporan data kegiatan dari *combo box* yang dipilih.
c. Tombol X yang ada di sudut kanan digunakan untuk kembali ke halaman utama untuk admin.

11. Sub Menu *Reboot*

Rancangan sub menu ini berfungsi untuk mengulang kembali status pegawai yang pernah menjadi anggota upacara.



Gambar 4.42 Perancangan Sub Menu *Reboot*

Keterangan :

a. Saat sub menu reboot di pilih maka status pegawai yang pernah menjadi anggota upacara akan terulang kembali.

4.2.3 Perancangan Pesan

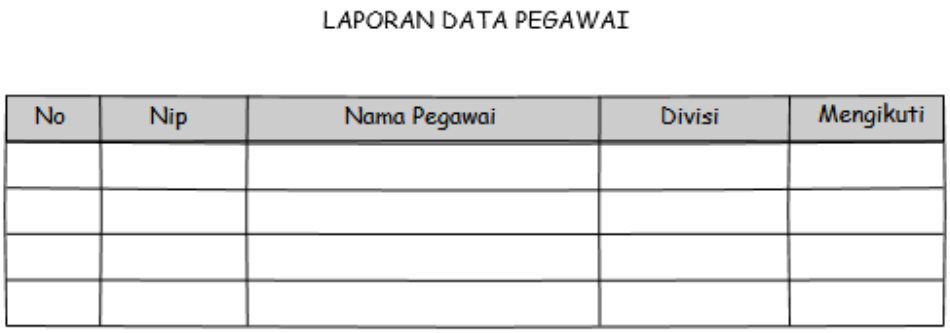
Perancangan pesan diperlukan untuk memberikan konfirmasi sebelum melakukan manipulasi data. Berikut *form* perancangan pesan :



Gambar 4.43 Perancangan Pesan

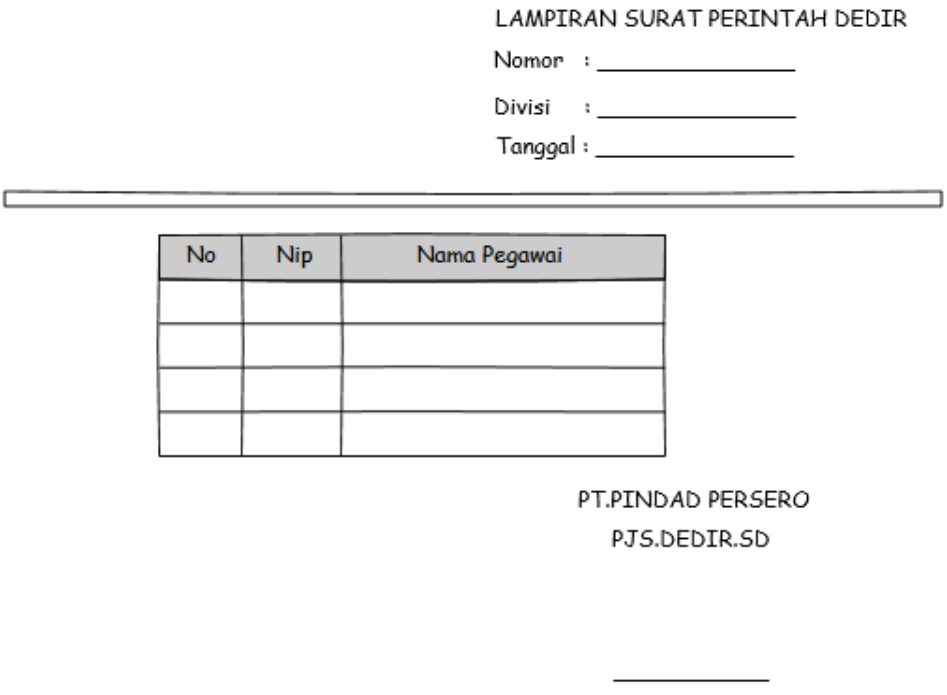
4.2.4 Perancangan Output

Perancangan *output* diperlukan untuk menentukan tampilan program yang berfungsi sebagai tempat menampilkan informasi dan data yang telahdi*input*kan. Dari aplikasi ini, output yang dihasilkan berupa jadwal kegiatan upacara.


LAPORAN DATA PEGAWAI

No	Nip	Nama Pegawai	Divisi	Mengikuti

Gambar 4.44 Perancangan Laporan Data Pegawai Untuk Admin


LAMPIRAN SURAT PERINTAH DEDIR

Nomor : _____

Divisi : _____

Tanggal : _____

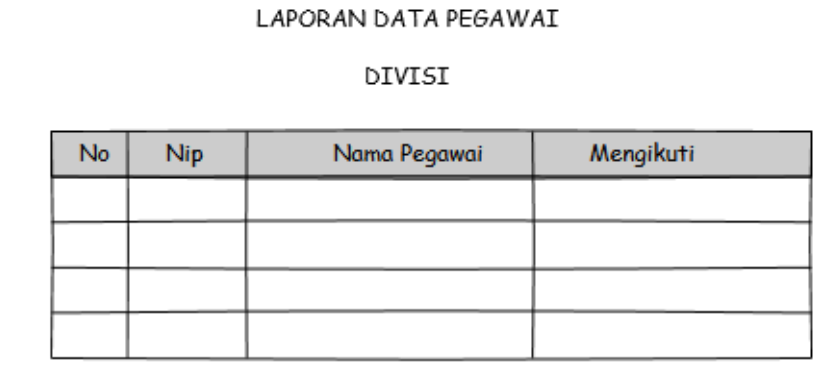
No	Nip	Nama Pegawai

PT.PINDAD PERSERO

PJS.DEDIR.SD

_____

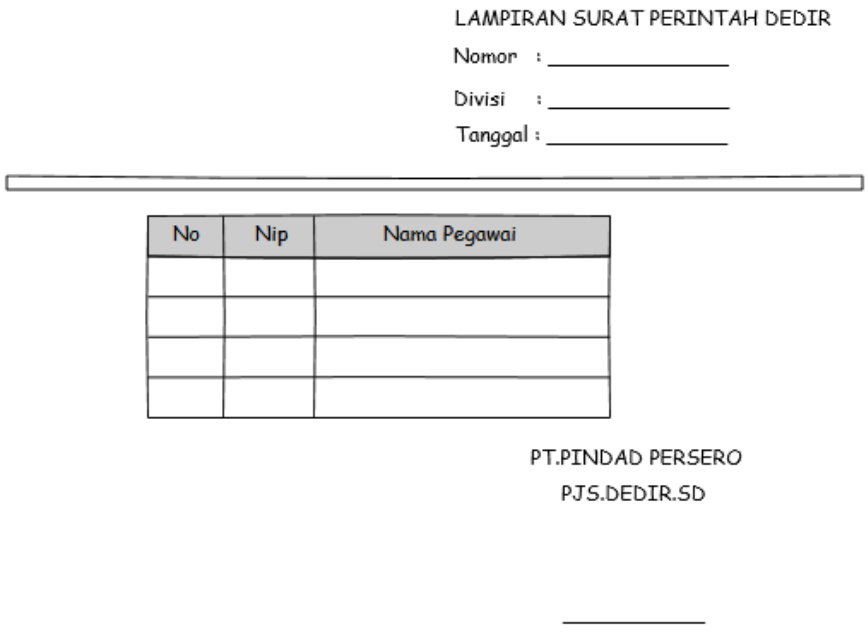
Gambar 4.45 Perancangan Laporan Jadwal Upacara Untuk Admin


LAPORAN DATA PEGAWAI

DIVISI

No	Nip	Nama Pegawai	Mengikuti

Gambar 4.46 Perancangan Laporan Data Pegawai Untuk Divisi


LAMPIRAN SURAT PERINTAH DEDIR

Nomor : _____

Divisi : _____

Tanggal : _____

No	Nip	Nama Pegawai

PT.PINDAD PERSERO

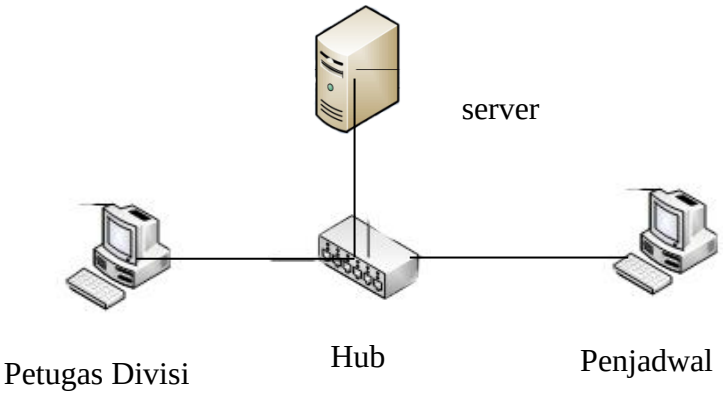
PJS.DEDIR.SD

_____

Gambar 4.47 Perancangan Laporan Jadwal Upacara Untuk Divisi

4.3 Perancangan Arsitektur Jaringan

Dalam pengimplementasian aplikasi penjadwalan kegiatan upacara pada PT.Pindad. Jaringan ini berfungsi untuk sebagai penghubung antara komputer antara satu dan yang lainnya.Berikut adalah skema dari arsitektur jaringan yang digunakan.



Gambar 4.48 Perancangan Arsitektur Jaringan

4.4 Implementasi

Implementasi merupakan pengaplikasian perangkat lunak, perangkat keras serta *database* yang digunakan. Pada tahapan ini akan dilakukan implementasi meliputi lingkungan implementasi, batasan implementasi, dan konfigurasi dari kebutuhan sistem untuk dapat menjalankan aplikasi perangkat lunak yang telah dibangun.

4.4.1 Batasan Implementasi

Berikut adalah hal-hal yang menjadi batasan dalam implementasi dari aplikasi ini :

1. Aplikasi ini dapat diakses oleh dua atau lebih pengguna dari tiap divisi yang ditentukan oleh pengguna utama (petugas penjadwalan).
2. Penginputan data hanya dengan tipe data standar yaitu *real*, integer dan string.

3. Konfigurasi basis data harus dilakukan secara manual dengan memanfaatkan *file Dump* SQL .
4. Aplikasi yang dibuat bertujuan untuk membuat penjadwalan kegiatan upacara berdasarkan ketersediaan pegawai yang akan menjadi anggota upacara.

4.4.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan program aplikasi penjadwalan ini adalah sebagai berikut :

1. Sistem operasi yang digunakan dalam pembangunan dan pengujian aplikasi ini adalah *Microsoft Windows 7*dan *Microsoft Windows 7.*
2. Bahasa pemrograman yang digunakan untuk membangun applikasi ini adalah bahasa pemrograman java dengan *Java Development Kit* 6.9 (JDK6.9).
3. Perangkat Pemograman Netbeans 7.3.1.
4. Basis data dengan MySQL 5.0.67.
5. Laporan menggunakan iReport-4.0.2 *for* Windows.
6. Java *Libraries.*

4.4.3 Implementasi Perangkat Keras

Berikut adalah perangkat keras yang digunakan dalam implementasi aplikasi penjadwalan ini dengan minimal spesifikasi adalah :

1. CPU Intel Dual Core 2.0 GHz.

2. Memory DDR RAM 512 MB.
3. Harddisk 40 GB.
4. Perangkat keluaran berupa *Monitor* 14”.
5. Perangkat masukan berupa *mouse* dan *keyboard*

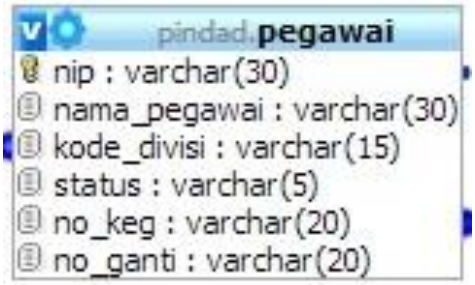
4.4.4 Implementasi Basis Data

Pada pembangunan aplikasi ini dibangun *database* agar tercipta sistem informasi yang terintegrasi dan dinamis. Selain itu pembuatan dan implementasi *database* merupakan langkah yang memperlihatkan kesesuaian perancangan pada tahapan analisis dan perancangan terhadap pembangunan aplikasi. Berikut adalah basis data yang dibuat menggunakan bahasa SQL (MySQL) dalam Aplikasi Penjadwalan Kegiatan Upacara:

Tabel 4.10 *DDL* Untuk Divisi

Nama Tabel	Hasil Generate
	CREATE TABLE `divisi` (`kode_divisi` varchar(30) NOT NULL,`nama_divisi` varchar(30) NOT NULL, `kuota` varchar(4) NOT NULL,`aktifasi` varchar(15) NOT NULL, PRIMARY KEY (`kode_divisi`), UNIQUE KEY `kode_divisi` (`kode_divisi`)) ENGINE=MyISAM DEFAULT CHARSET=latin1;

Tabel 4.11 *DDL* Untuk Pegawai

Nama Tabel	Hasil Generate
	CREATE TABLE `pegawai` (`nip` varchar(30) NOT NULL, `nama_pegawai` varchar(30) NOT NULL, `kode_divisi` varchar(15) NOT NULL, `status` varchar(5) NOT NULL, `no_keg` varchar(20) NOT NULL, `no_ganti` varchar(20) NOT NULL, PRIMARY KEY (`nip`), UNIQUE KEY `nip` (`nip`)) ENGINE=MyISAM DEFAULT CHARSET=latin1;

Tabel 4.12 *DDL* Untuk Kegiatan

Nama Tabel	Hasil Generate
	CREATE TABLE `kegiatan` (`no_keg` varchar(20) NOT NULL, `tanggal` date NOT NULL, `pjs` varchar(50) NOT NULL, PRIMARY KEY (`no_keg`), UNIQUE KEY `no_keg` (`no_keg`)) ENGINE=MyISAM DEFAULT CHARSET=latin1;

Tabel 4.13 *DDL* Untuk d_kegiatan

Nama Tabel	Hasil Generate
	CREATE TABLE `d_kegiatan` (`no_keg` varchar(20) NOT NULL, `nip` varchar(30) NOT NULL) ENGINE=MyISAM DEFAULT CHARSET=latin1;

Tabel 4.14 *DDL* Untuk User

Nama Tabel	Hasil Generate
	CREATE TABLE `user` (`username` varchar(15) NOT NULL, `pass` varchar(15) NOT NULL, `bagian` varchar(15) NOT NULL, PRIMARY KEY (`username`), UNIQUE KEY `username` (`username`)) ENGINE=MyISAM DEFAULT CHARSET=latin1;

4.4.5 Implementasi Antar Muka

Antarmuka dalam sebuah aplikasi adalah hal yang sangat penting, aplikasi *desktop* yang menggunakan bahasa JAVA ini, implementasi antarmuka dilakukan dengan sebuah halaman yang berekstensi .java. Implementasi antarmuka dari halaman yang dibuat dapat dilihat pada tabel di bawah ini.

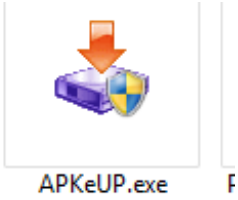
Tabel 4.15 Implementasi Antarmuka

Menu	Deskripsi	Tipe
Menu Utama Admin	Tampilan menu utama untuk admin	*Form*
DataDivisi	*Form* data divisi	*Form*
DataPegawai	*Form* data pegawai	*Form*
DataDivisi	*Form* data divisi	*Form*
Penjadwalan	*Form* data Penjadwalan	*Form*
LaporanPenjadwalan	Tampilan untuk cetak jadwal	*Form*
Menu UtamaDivisi	Tampilan menu utama untuk admin	*Form*

4.4.6 Implementasi Instalasi Program

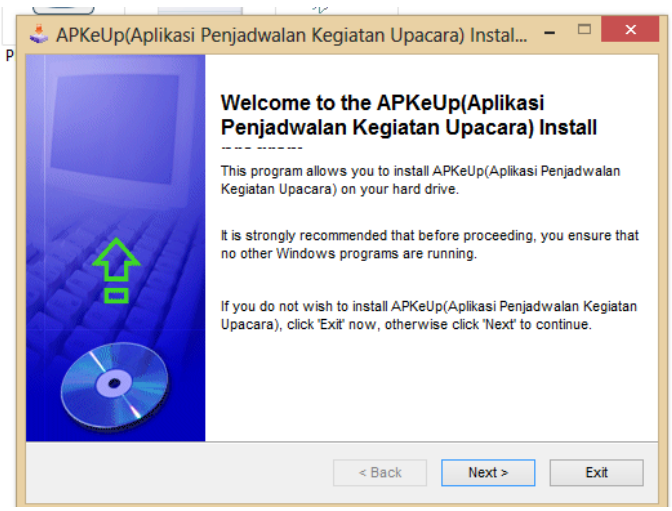
Dalam mengimplementasikan program ke dalam komputer, maka sebelumnya harus dipersiapkan segala kebutuhan yang mendukung agar perangkat lunak dapat berjalan dengan baik pada *platform* teknologi yang kita pakai. Berikut ini adalah beberapa hal yang harus dipersiapkan terlebih dahulu, yaitu :

1. Instalasi XAMPP.
2. Lakukan *Dump* SQL untuk mengimport *database* ke dalam XAMPP.
3. Lalu lakukan instalasi SetUp APKeUp PT.Pindad V1.0, sebagai berikut :



Gambar 4.49 Icon Installer Aplikasi Penjadwalan Kegiatan Upacara

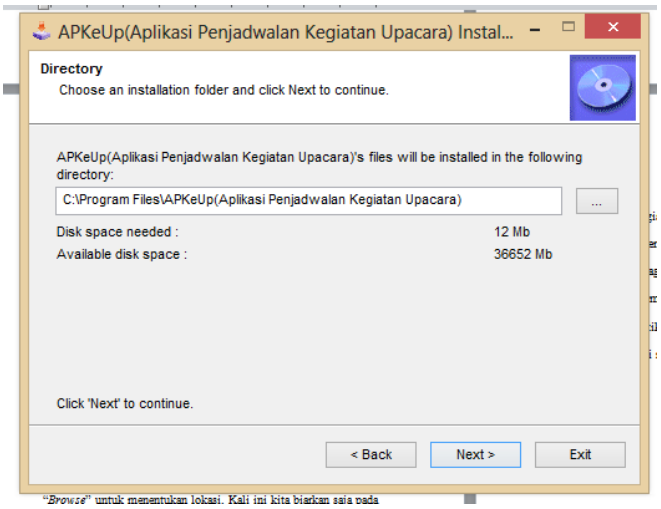
a. Klik ganda File Setup APKeUp PT.Pindad, maka akan tampak halaman "Welcome", seperti gambar dibawah ini.



Gambar 4.50 Tampilan *Welcome*

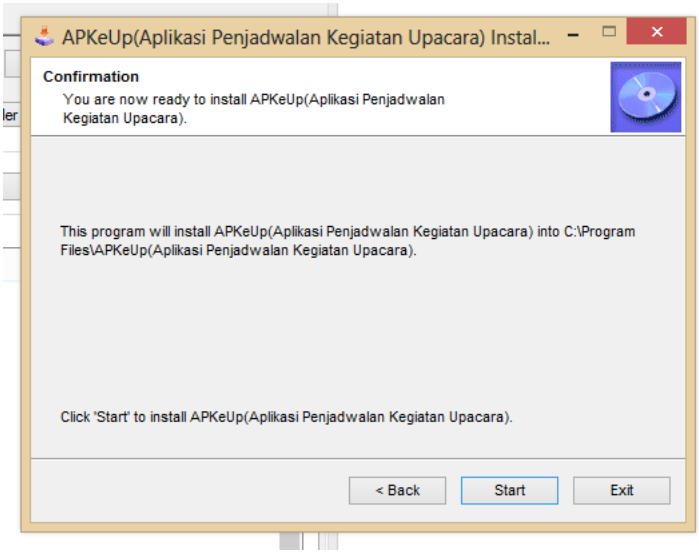
b. Kemudian klik *Next*.

c. Maka akan muncul tampilan pemilihan lokasi instalasi program, klik "*Browse*" untuk menentukan lokasi. Kali ini kita biarkan saja pada kondisi *default*. Kemudian klik *Next*



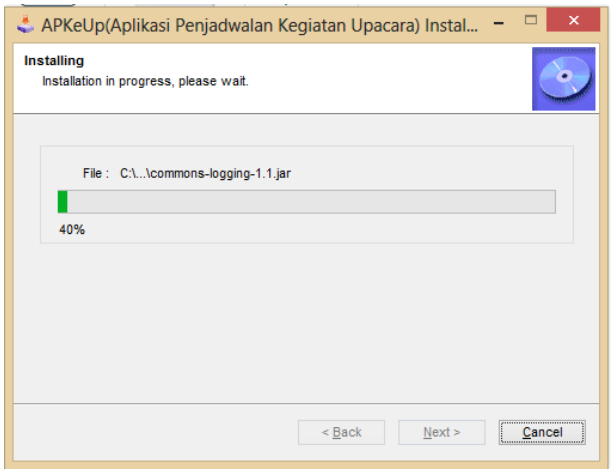
Gambar 4.51 Lokasi Instalasi

d. Jika tampilan berikut muncul, berarti Program telah siap untuk di install. Klik *install*.



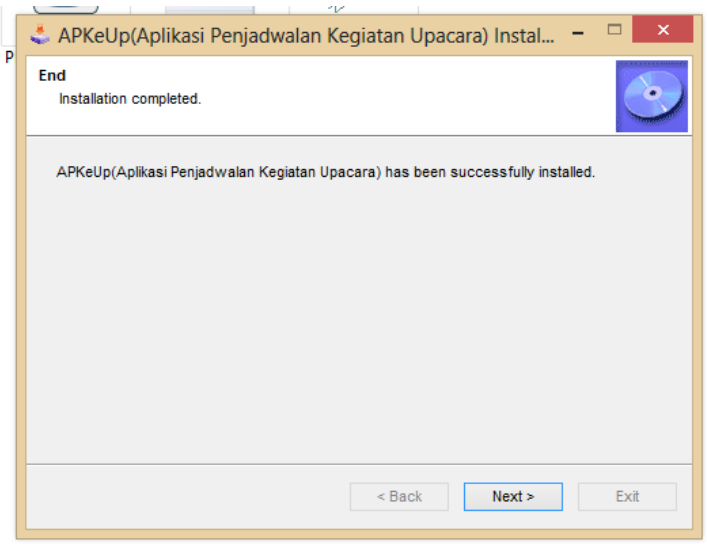
Gambar 4.52 *Ready to Install*

e. Tunggulah proses instalasi sampai selesai



Gambar 4.53 *Setup* APKeUp

f. Klik *Next.*



Gambar 4.54 *Finish*

4.4.7 Penggunaan Program

Kegiatan ini bertujuan untuk menerangkan secara singkat penggunaan program Pembuatan Jadwal KegiataUpacara. Adapun cara pengoperasiannya adalah sebagai berikut :

1. *Form Login*

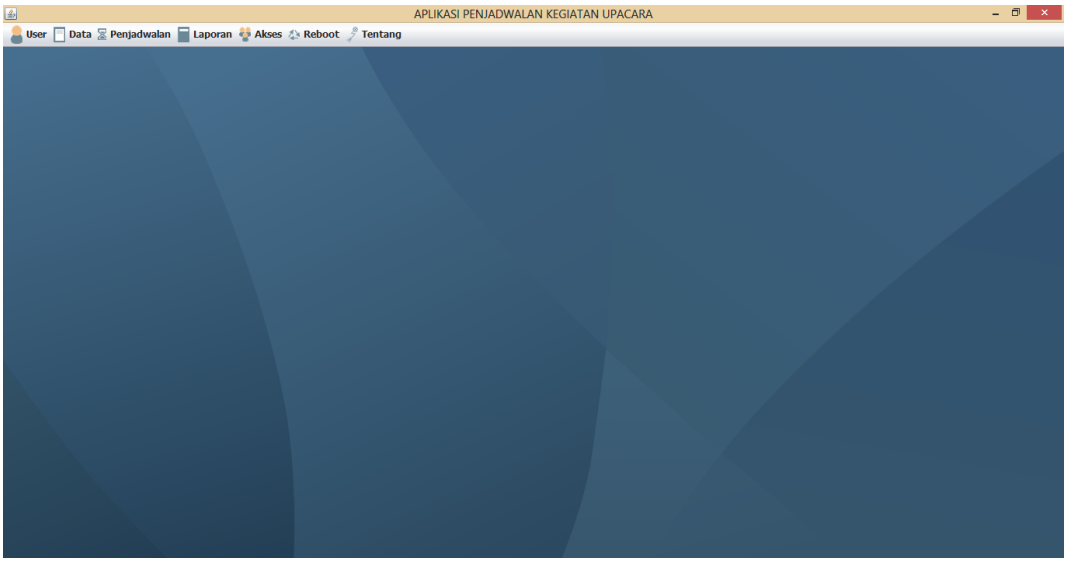
   Ketika pertama kali program aplikasi ini dijalankan maka yang pertama kali muncul adalah *form login* seperti di bawah ini.



Gambar 4.55 Tampilan *Login*

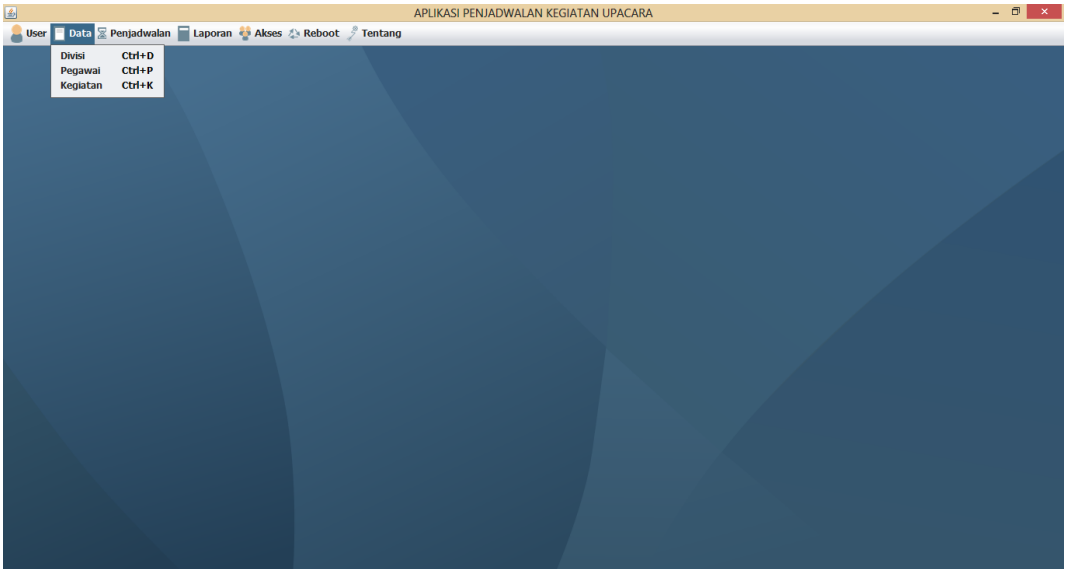
2. Menu Admin

Halaman ini merupakan halaman yang berisikan menu-menu yang dapat dilakukan oleh admin pada aplikasi penjadwalan kegiatan upacara ini.



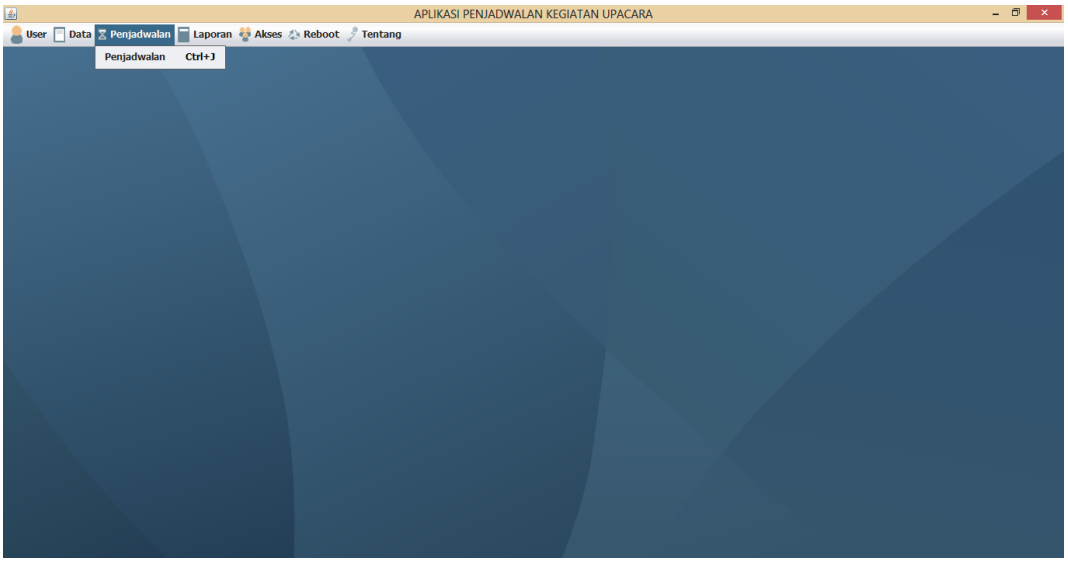
Gambar 4.56 Tampilan Menu Admin

3. Menu data merupakan menu pada halaman admin. Dimana didalamnya terdapat 3 sub menu, yaitu Divisi, Pegawai, Kegiatan.



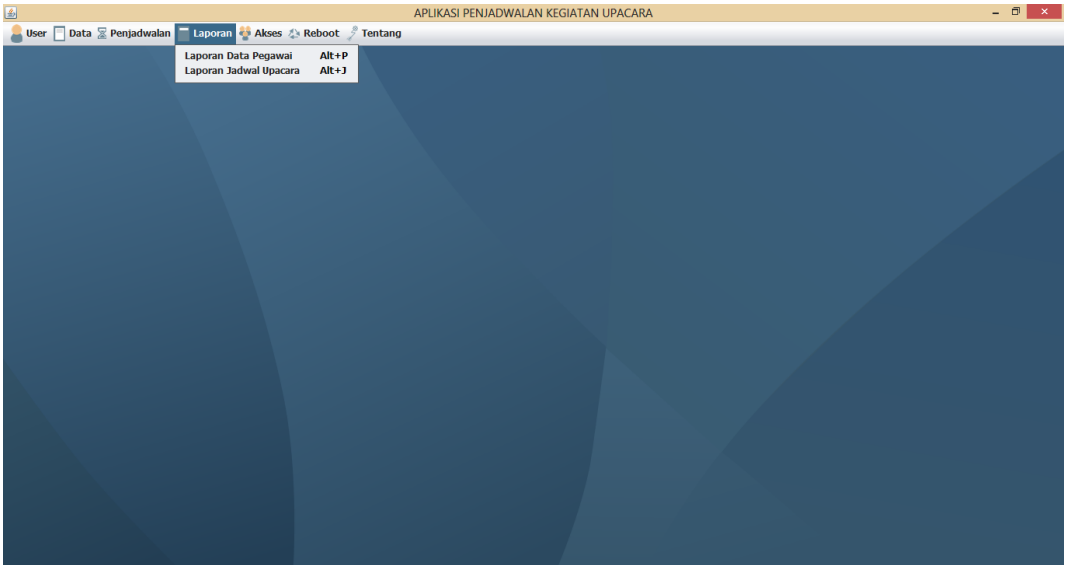
Gambar 4.57 Tampilan Menu Data

4. Menu penjadwalan merupakan menu pada halaman admin. Dimana didalamnya terdapat 1 sub menu, yaitu Penjadwalan.



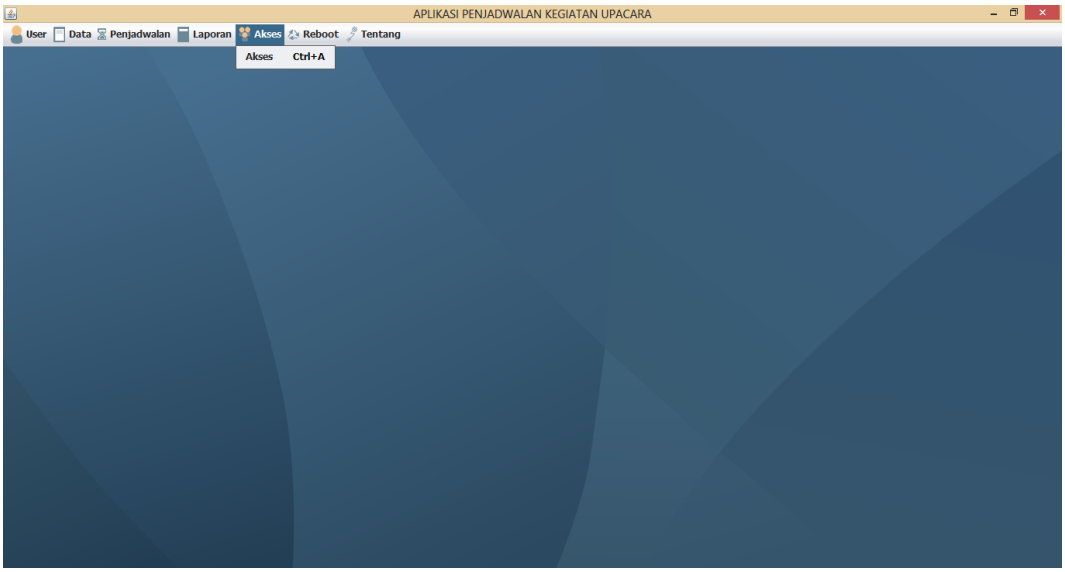
Gambar 4.58 Tampilan Menu Penjadwalan

5. Menu Laporan merupakan menu pada halaman admin. Dimana didalamnya terdapat 2 sub menu, yaitu Laporan Data Pegawai, dan Laporan Jadwal Upacara.



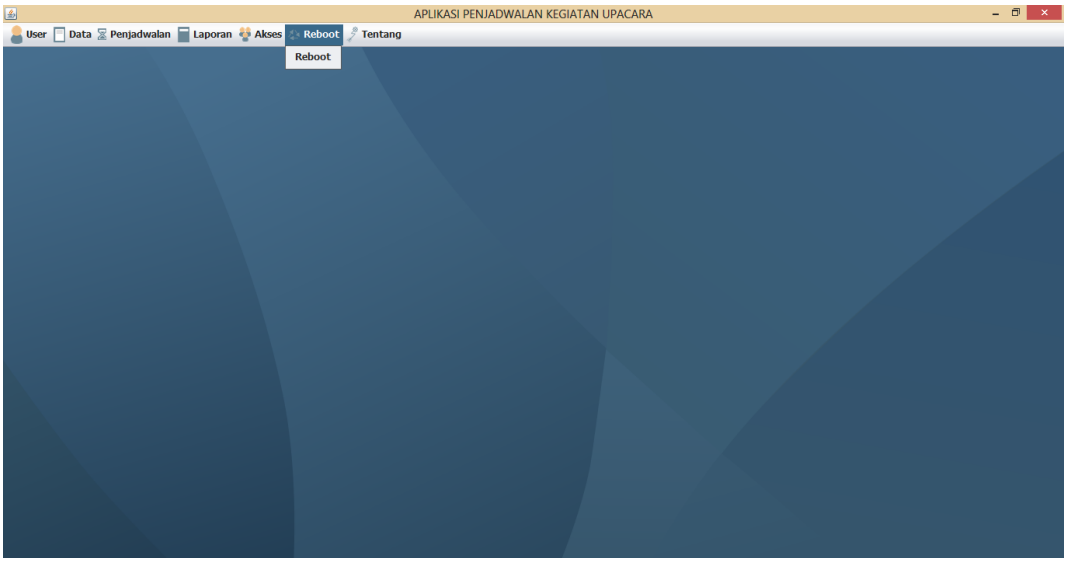
Gambar 4.59 Tampilan Menu Laporan

6. Menu Akses merupakan menu pada halaman admin. Dimana didalamnya terdapat 1 sub menu, yaitu Akses.



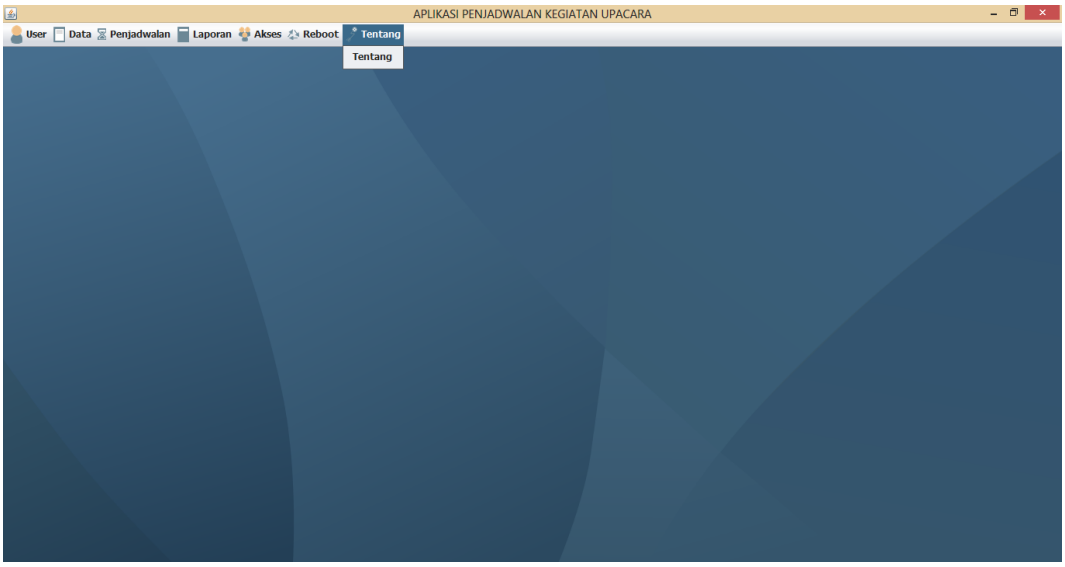
Gambar 4.60 Tampilan Menu Akses

7. Menu *reboot* merupakan menu pada halaman admin. Dimana didalamnya terdapat 1 sub menu, yaitu *Reboot*.



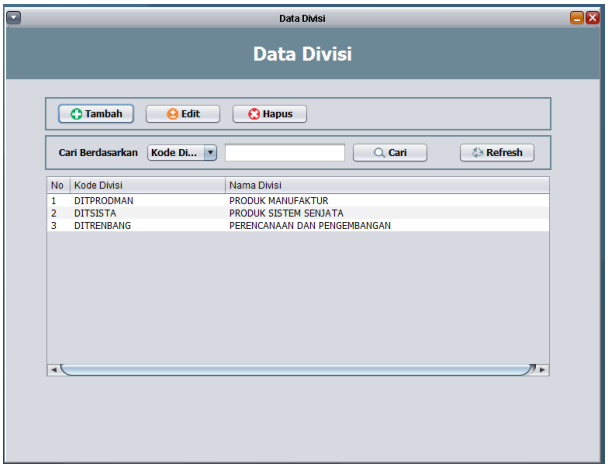
Gambar 4.61 Tampilan Menu *Reboot*

8. Menu Tentang merupakan menu pada halaman admin. Dimana didalamnya terdapat 1 sub menu, yaitu Tentang.



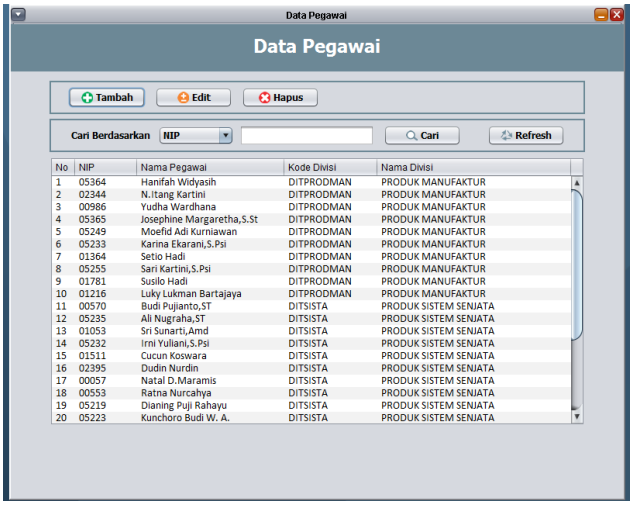
Gambar 4.62 Tampilan Menu Tentang

9. Sub menu divisi merupakan *form* pada halaman admin untuk menambah, menghapus, mengubah, menyimpan, serta melihat data divisi.



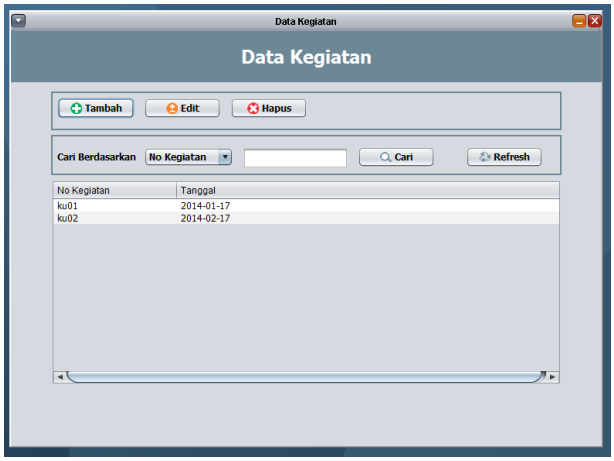
Gambar 4.63 *Form* Divisi

10. Sub menu pegawai merupakan *form* pada halaman admin untuk menambah, menghapus, mengubah, menyimpan, serta melihat data pegawai.



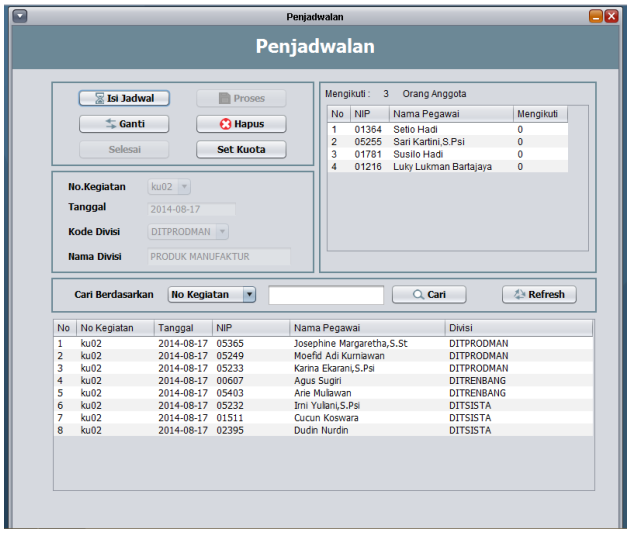
Gambar 4.64 *Form* Pegawai

11. Sub menu kegiatan merupakan *form* pada halaman admin untuk menambah, menghapus, mengubah, menyimpan, serta melihat data kegiatan.



Gambar 4.65 *Form* Kegiatan

12. Sub menu penjadwalan merupakan *form* pada halaman admin untuk menambah, menghapus, mengubah, menyimpan, serta melihat data penjadwalan.



Gambar 4.66 *Form* Penjadwalan

13. Sub laporan jadwal upacara merupakan *form* pada halaman admin untuk mencetak laporan berdasarkan kegiatan.



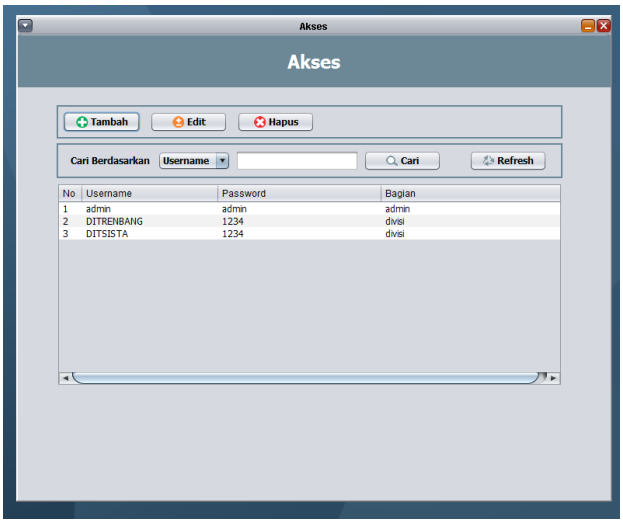
Gambar 4.67 *Form* Laporan Admin

14. Sub laporan jadwal upacara pada halaman setiap divisi untuk mencetak laporan berdasarkan kegiatan.



Gambar 4.68 *Form* Laporan divisi

15. Sub menu akses merupakan *form* pada halaman admin untuk menambah, menghapus, mengubah, menyimpan, serta melihat data akses.



Gambar 4.69 *Form* Akses

4.5 Pengujian

Proses pengujian berfokus pada logika internal *software*, memastikan bahwa semua pernyataan sudah diuji. Dan pada eksternal fungsional mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa masukan yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

4.5.1 Rencana Pengujian

Pengujian dilakukan melalui menggunakan data uji yang dimasukkan oleh admin. Berikut adalah rencana pengujian yang dilakukan:

Tabel 4.16 Rencana Pengujian

Item Uji	Deskripsi	*Form*	Jenis Pengujian
Login	Melakukan *login* ke dalam sistem	FrameLogin.java	*Black Box*
Olah Data	Melakukan pengolahan data	Divisi.java	*Black Box*
		Pegawai.java	
		Kegiatan.java	
Penjadwalan	Membuat jadwal kegiatan upacara	Penjadwalan.java	*Black Box*

4.5.2 Kasus dan Hasil Pengujian

Pada bagian ini, item-item yang telah ditentukan di atas akan diujikan. Kasus-kasus dalam pengujian akan dipaparkan berikut dengan hasil pengujian tersebut.

1. Pengujian Validalitas *Login*

   Pada pengujian *login*, data antara *username* dan *password* harus sesuai, untuk pengujian diasumsikan data *username* "admin" dan *password* inputan "rahasia" adalah benar, selain dari itu salah. Pada pengujian ini akan diperlihatkan pengujian pada tingkatan Validitas dari Black Box.

Tabel 4.17 Pengujian *Login*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Username: admin *Password*: rahasia	Menampilkan menu utama	Tampil menu utama	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Username: salah	Menampilkan	Tampil pesan	[X] Diterima

Password: salah	pesan peringatan	peringatan	[ ] Ditolak

2. Pengujian Divisi

Tabel 4.18 Pengujian Divisi

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukan lengkap	Menampilkan pesan data tersimpan dan data akan terlihat di table pada *form*.	Data tampil pada tabel	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukan tidak lengkap	Menampilkan pesan data tidak boleh kosong.	Tampil Pesan Peringatan	[X] Diterima [ ] Ditolak
Masukan data Sama dengan yang telah dimasukan	Menampilkan pesan data telah ada.	Tampil Pesan Peringatan	[X] Diterima [ ] Ditolak

3. Pengujian Pegawai

Tabel 4.19 Pengujian Pegawai

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukan lengkap	Menampilkan pesan data tersimpan dan data akan terlihat di table pada *form*.	Data tampil pada tabel	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukan tidak lengkap	Menampilkan pesan data tidak boleh kosong.	Tampil Pesan Peringatan	[X] Diterima [ ] Ditolak
Masukan data Sama dengan yang telah dimasukan	Menampilkan pesan data telah ada.	Tampil Pesan Peringatan	[X] Diterima [ ] Ditolak

4. Pengujian Kegiatan

Tabel 4.20 Pengujian Kegiatan

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukan lengkap	Menampilkan pesan data tersimpan dan data akan terlihat di table pada *form*.	Data tampil pada tabel	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukan tidak lengkap	Menampilkan pesan data tidak boleh kosong.	Tampil Pesan Peringatan	[X] Diterima [ ] Ditolak

5. Pengujian Penjadwalan

Tabel 4.21 Pengujian Penjadwalan

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukan lengkap	Menampilkan pesan data tersimpan dan data	Data tampil pada tabel	[X] Diterima [ ] Ditolak

	akan terlihat di table pada *form*.		
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukan lengkap tetapi kuota sudah terpenuhi	Menampilkan pesan kuota sudah terpenuhi.	Tampil Pesan Peringatan	[X] Diterima [ ] Ditolak

4.5.3 Kesimpulan dan Hasil Pengujian

Pengujian yang telah dilakukan merupakan beberapa proses yang terdapat dalam aplikasi. Berdasarkan hasil pengujian diatas dapat ditarik kesimpulan bahwa perangkat lunak bebas dari kesalahan sintaks dan secara fungsional mengeluarkan hasil yang diharapkan. Namun demikian pengujian tersebut dapat dikatakan belum sempurna, dikarenakan hanya dilakukan pada satu sisi pengujian. Dari semua yang telah dilakukan dalam pengujian ini diharapkan dapat mewakili pengujian fungsi yang lainnya. Secara garis besar dari pengujian dan hasil implementasi perangkat lunak yang dibangun sudah sesuai dengan spesifikasi kebutuhan dan hasil perancangan perangkat lunak