BAB 3
ANALISIS DAN PERANCANGAN SISTEM

Analisis dan perancangan sistem memerlukan tahapan yang sistematis untuk mendapatkan aplikasi yang baik dan bersesuaian dengan kegunaan dan tujuannya. Tahap awal dari analisis adalah menganalisis kebutuhan-kebutuhan sistem mulai dari kebutuhan pengguna, kebutuhan non fungsional, dan kebutuhan fungsional. Sedangkan untuk tahap perancangan aplikasi yaitu perancangan *Database*, dan perancangan antarmuka.

3.1 Analisis Sistem

Langkah pertama yang dilakukan dalam membangun sistem ini yaitu menganalisis sistem. Sistem yang akan dianalisis adalah sistem yang sedang berjalan (dalam hal ini sistem yang sudah ada) dan yang akan dibangun. Analisis harus dilakukan dengan seksama dan teliti untuk benar-benar mengetahui detail yang ada pada suatu sistem. Apabila dalam melakukan analisis dilakukan secara benar maka untuk lanjut ke tahap berikutnya tidak akan mengalami kesulitan, begitu juga sebaliknya.

3.1.1 Analisis Masalah

Pada umumnya sebuah perusahaan memerlukan informasi mengenai data – data yang bersifat penting untuk disampaikan kepada kliennya. Namun saat ini media yang umum digunakan oleh perusahaan perusahaan masih berupa buku yang memiliki fisik mudah rusak dan sering terjadi kehilangan. Hal ini tentu merepotkan dan menghambat kinerja perusahaan dalam penyampaian informasi mengenai data – data yang dibutuhkan kliennya. Oleh karena itu dibutuhkan suatu sistem yang dapat mengatasi kendala tersebut agar sebuah perusahaan dapat bekerja secara efektif dan efisien.

Permasalahan dalam penelitian ini terdapat di Perusahaan CV Gunung Agung yaitu pada Bagian *Service* mengenai sistem laporan kerusakan dari *General Affair* alfamart pusat yang masih berupa form manual yang di *fax* kepada

kepala teknisi yang terus berulang setiap saat maka akan menimbulkan penumpukan berkas laporan kerusakan, sehingga seringnya terjadi kehilangan data laporan kerusakan tersebut. Pembuatan beritaca acara kerusakan dan Laporan *Service* yang masih berupa form manual menyulitkan kepala teknisi dalam melakukan verifikasi pergantian data sparepart. Tidak adanya pemantauan terhadap kinerja teknisi lapangan yang melakukan *Service*, Serta sulitnya pencarian lokasi toko alfamart yang mengalami kerusakan.

Dari masalah yang telah di jelaskan maka di butuhkan suatu aplikasi *mobile* berbasis android dan *Web Service* yang dapat menunjang kinerja perusahaan CV Gunung Agung dalam pembuatan laporan *Service*, pembuatan laporan kerusakan , pembuatan berita acara serta dapat melihat kinerja dari setiap teknisi lapangan, dan dapat menyimpan lokasi toko alfamart yang mengalami kerusakan.

3.2 Analisis Sistem yang Sedang Berjalan

Analisis sistem yang berjalan bertujuan untuk mengetahui secara jelas tentang sistem yang sedang berjalan . Tahap ini dilakukan untuk mempermudah dalam mengetahui kelebihan dan kekurangan sistem dan dapat dijadikan acuan dalam usulan perancangan sistem agar dapat menghasilkan output yang diinginkan .

3.2.1 Prosedur *Service Show Case*

Prosedur *Service Showcase* pada CV Gunung Agung yang sedang berjalan saat ini memiliki 4 entitas diantaranya adalah sebagai berikut:

1. ***General Affair* alfamart pusat.**
    a. Membuat Surat Laporan Kerusakan
    b. Menyerahkan Surat Laporan Kerusakan kepada CV Gunung Agung.
    c. Menerima rincian kerusakan , rincian biaya penggantian *sparepart*, serta berita acara hasil pengecekan *Showcase* yang mengalami kerusakan dengan biaya perbaikan diatas satu jutarupiah untuk di setujui apakah di lakukan perbaikan dan penggantian *sparepart*,

jika setuju maka menyerahkan laporan pengecekan untuk *Showcase* yang mengalami kerusakan dengan biaya perbaikan diatas satu juta rupiah yang sudah di setujui untuk dilakukan perbaikan dan penggantian *sparepart*.

d. Menerima laporan dari kepala teknisi CV Gunung Agung perihal hasil perbaikan dari kepala teknisi yang di sertai berita acara perbaikan yang sudah di tanda tangani kepala toko alfamart dan nota biaya pembayaran yang di tandatangani kepala teknisi.

2. Kepala Teknisi

a. Menerima Surat Laporan Kerusakan dari *General Affair* alfamart pusat.
b. Mengecek Kelengkapan Surat Laporan Kerusakan jika lengkap maka laporan kerusakan akan di lanjutkan kepada teknisi lapangan, jika tidak maka akan di kembalikan ke *General Affair* alfamart pusat.
c. Mencetak formulir pemeriksaan kerusakan yang akan di berikan ke teknisi.
d. Mengarsipkan salinan kerusakan yang di terima dari *General Affair* alfamart pusat.
e. Menerima berita acara pengecekan dari teknisi lapangan yang telah di tanda tangani kepala toko untuk melaksanakan perbaikan dan penggantian *sparepart* apabila kerusakan dengan biaya diatas satu juta rupiah yang di alami toko alfamart .
f. Membuat rincian kerusakan dan rincian biaya penggantian *sparepart* untuk *Showcase* yang mengalami kerusakandengan biaya perbaikan diatas satu juta rupiah.
g. Memberikan rincian kerusakan,rincian biaya penggantian *sparepart* serta berita acara hasil pengecekan *Showcase* yang mengalami kerusakan dengan biaya perbaikan diatas satu juta rupiah kepada *General Affair* alfamart pusat.

h. Menerima rincian kerusakan dan rincian biaya penggantian *sparepart* dari *General Affair* alfamart pusat yang sudah di setujui untuk di lakukan perbaikan dan penggantian *sparepart*.
i. Memberikan rincian kerusakan dan rincian biaya penggantian *sparepart* dari *General Affair* alfamart pusat yang sudah di setujui untuk di lakukan perbaikan dan penggantian *sparepart* kepada teknisi lapangan.
j. Menerima laporan perbaikan dari teknisi yang di sertai berita acara perbaikan yang sudah di tandatangani kepala toko.
k. Memberikan laporan dan berita acara perbaikan yang sudah selesai di perbaiki dan membuat tagihan perbaikan sesuai dengan laporan yang diberikan ke *General Affair* alfamart pusat.
l. Mengarsipkan laporan perbaikan,form persetujuan penggantian *sparepart* dengan biaya diatas satu juta rupiah, berita acara perbaikan, dan nota biaya perbaikan untuk setiap toko alfamart.

3. Teknisi Lapangan

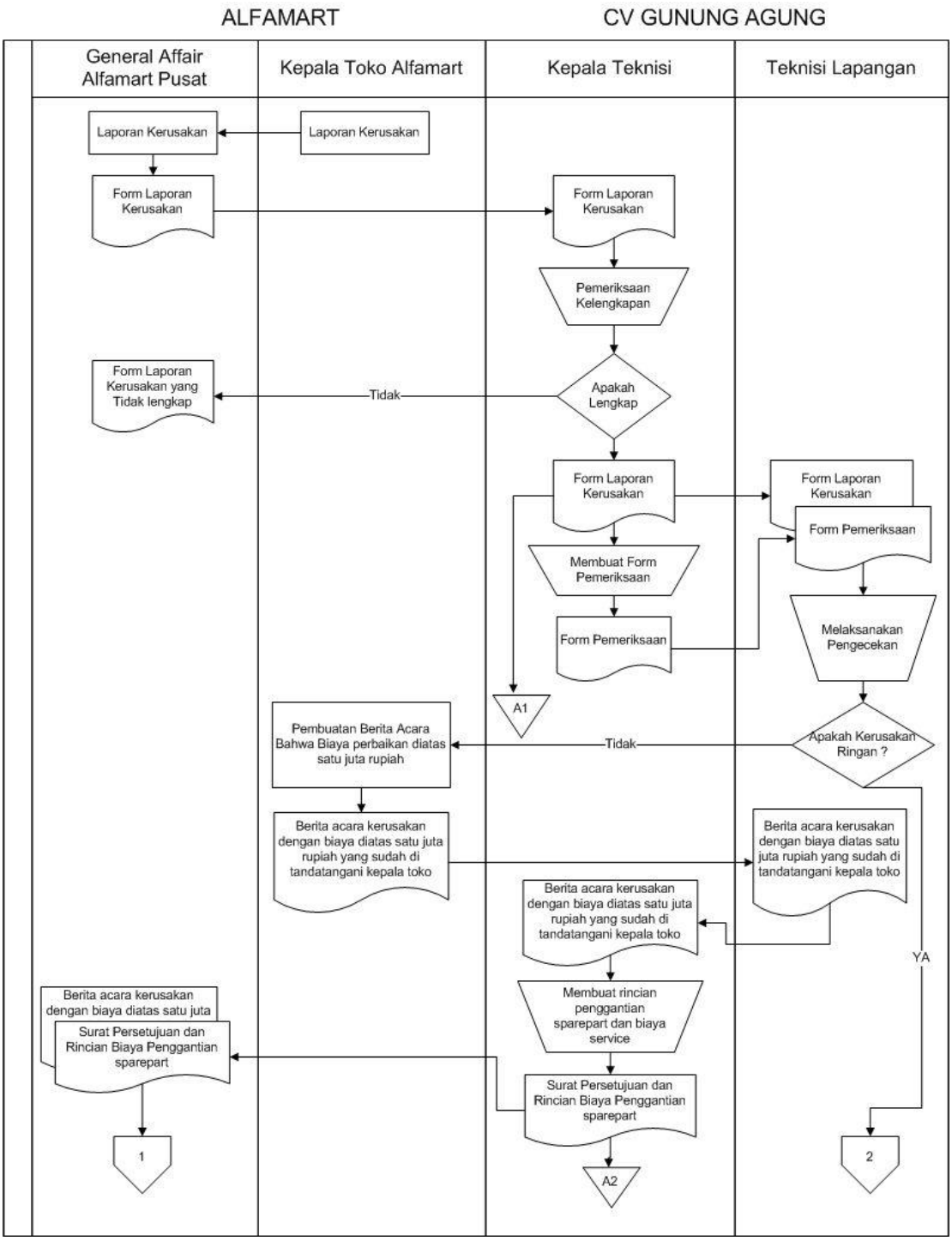
a. Menerima Laporan Kerusakan yang diberikan oleh kepala teknisi dan formulir pemeriksaan.
b. Melakukan Pengecekan *Showcase* di toko alfamart, jika kerusakan biaya perbaikan di bawah satu juta rupiah maka langsung diperbaiki , jika kerusakan *Showcase* memerlukan biaya perbaikan diatas satu juta rupiah maka teknisi lapangan meminta kepala toko alfamart agar membuat berita acara bahwa kerusakan *Showcase* memerlukan biaya diatas satu juta rupiah yang sudah di tandatangani kepala toko.
c. Memberikan berita acara pengecekan dari teknisi lapangan yang telah di tanda tangani kepala toko untuk melaksanakan perbaikan dan penggantian *sparepart* apabila kerusakan yang di alami toko alfamart memerlukan biaya perbaikan diatas satu juta rupiah kepada kepada kepala teknisi.

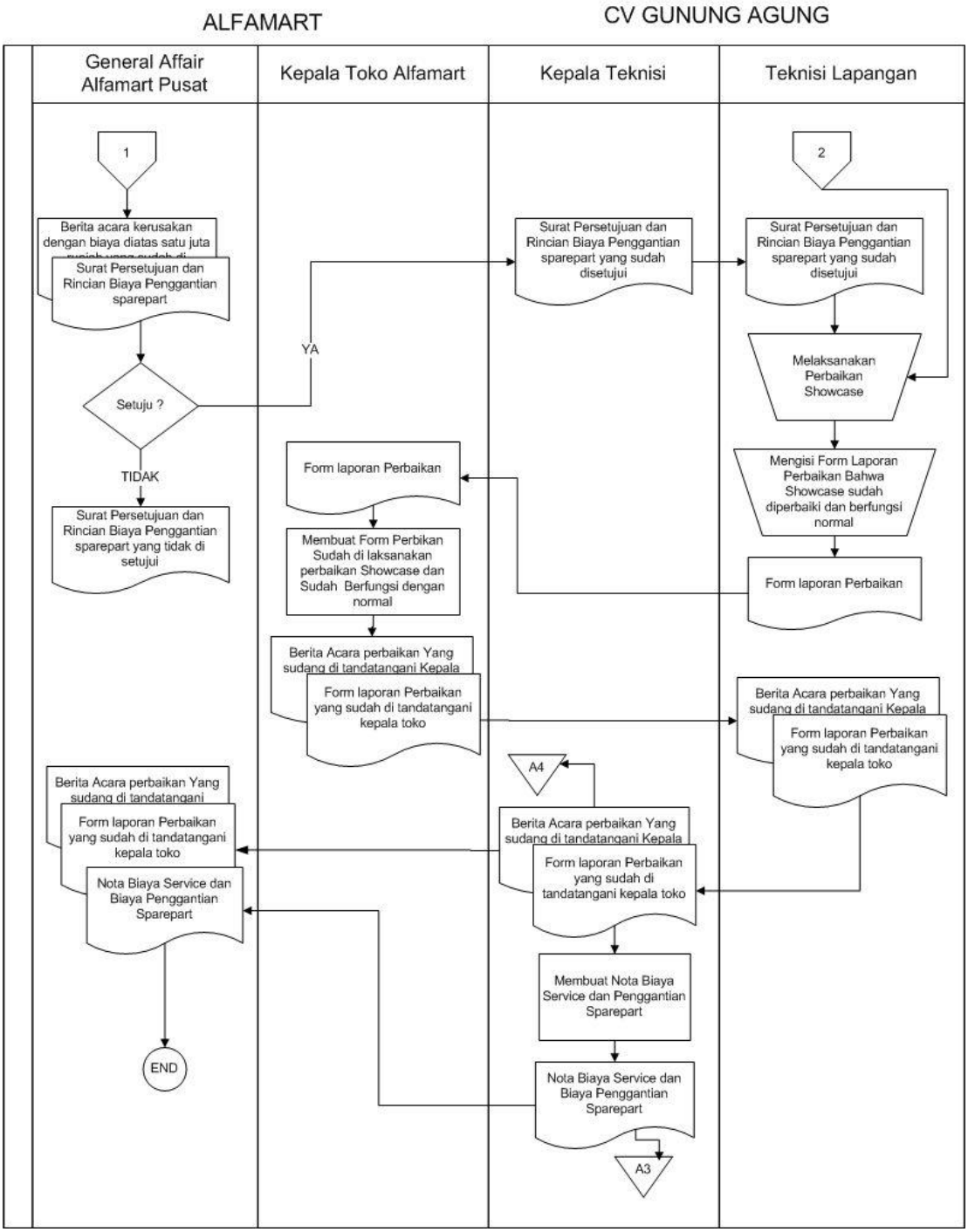
d. Menerima rincian kerusakan dan rincian biaya penggantian *sparepart* dari kepala teknisi yang sudah di setujui *General Affair* alfamart pusat untuk di lakukan perbaikan dan penggantian *sparepart*.
e. Teknisi memperbaiki *Showcase* yang mengalami kerusakan.
f. Membuat laporan perbaikan jika *Showcase* sudah berfungsi dengan normal.
g. Meminta menandatangani laporan perbaikan dan meminta membuat berita acara perbaikan kepada kepala toko apabila *Showcase* sudah berfungsi normal.
h. Teknisi lapangan memberikan laporan perbaikan beserta berita acara perbaikan yang sudah di tandatangani kepala toko untuk diberikan kepada kepala teknisi.

4. Kepala Toko Alfamart .

a. Membuat laporan kerusakan *Showcase* kepada *General Affair General Affair* alfamart pusat.
b. Membuat berita acara kerusakan *Showcase* yang memerlukan biaya perbaikan diatas satu juta rupiah.
c. Memberikan berita acara yang sudah ditandatangani kepala toko Kepada teknisi lapangan bahwa *Showcase* mengalami kerusakan dengan biaya perbaikan diatas satu juta rupiah.
d. Membuat berita acara *Showcase* yang sudah diperiksa, diperbaiki dan berfungsi dengan normal sesuai laporan perbaikan.
e. Memberikan berita acara yang sudah ditandatangani kepala toko Kepada teknisi lapangan bahwa *Showcase* sudah diperiksa, diperbaiki dan dingin.
f. Mengarsipkan laporan perbaikan dan berita acara perbaikan.

Flow map sistem yang berjalan pada CV Gunung Agung dapat di lihat pada gambar 3.1 :


ALFAMART
CV GUNUNG AGUNG
General Affair Alfamart Pusat
Kepala Toko Alfamart
Kepala Teknisi
Teknisi Lapangan
Laporan Kerusakan
Laporan Kerusakan
Form Laporan Kerusakan
Form Laporan Kerusakan
Pemeriksaan Kelengkapan
Form Laporan Kerusakan yang Tidak lengkap
Tidak
Apakah Lengkap
Form Laporan Kerusakan
Form Laporan Kerusakan
Form Pemeriksaan
Membuat Form Pemeriksaan
Form Pemeriksaan
Melaksanakan Pengecekan
A1
Pembuatan Berita Acara Bahwa Biaya perbaikan diatas satu juta rupiah
Tidak
Apakah Kerusakan Ringan ?
Berita acara kerusakan dengan biaya diatas satu juta rupiah yang sudah di tandatangani kepala toko
Berita acara kerusakan dengan biaya diatas satu juta rupiah yang sudah di tandatangani kepala toko
Berita acara kerusakan dengan biaya diatas satu juta rupiah yang sudah di tandatangani kepala toko
YA
Berita acara kerusakan dengan biaya diatas satu juta
Surat Persetujuan dan Rincian Biaya Penggantian sparepart
Membuat rincian penggantian sparepart dan biaya service
Surat Persetujuan dan Rincian Biaya Penggantian sparepart
1
A2
2



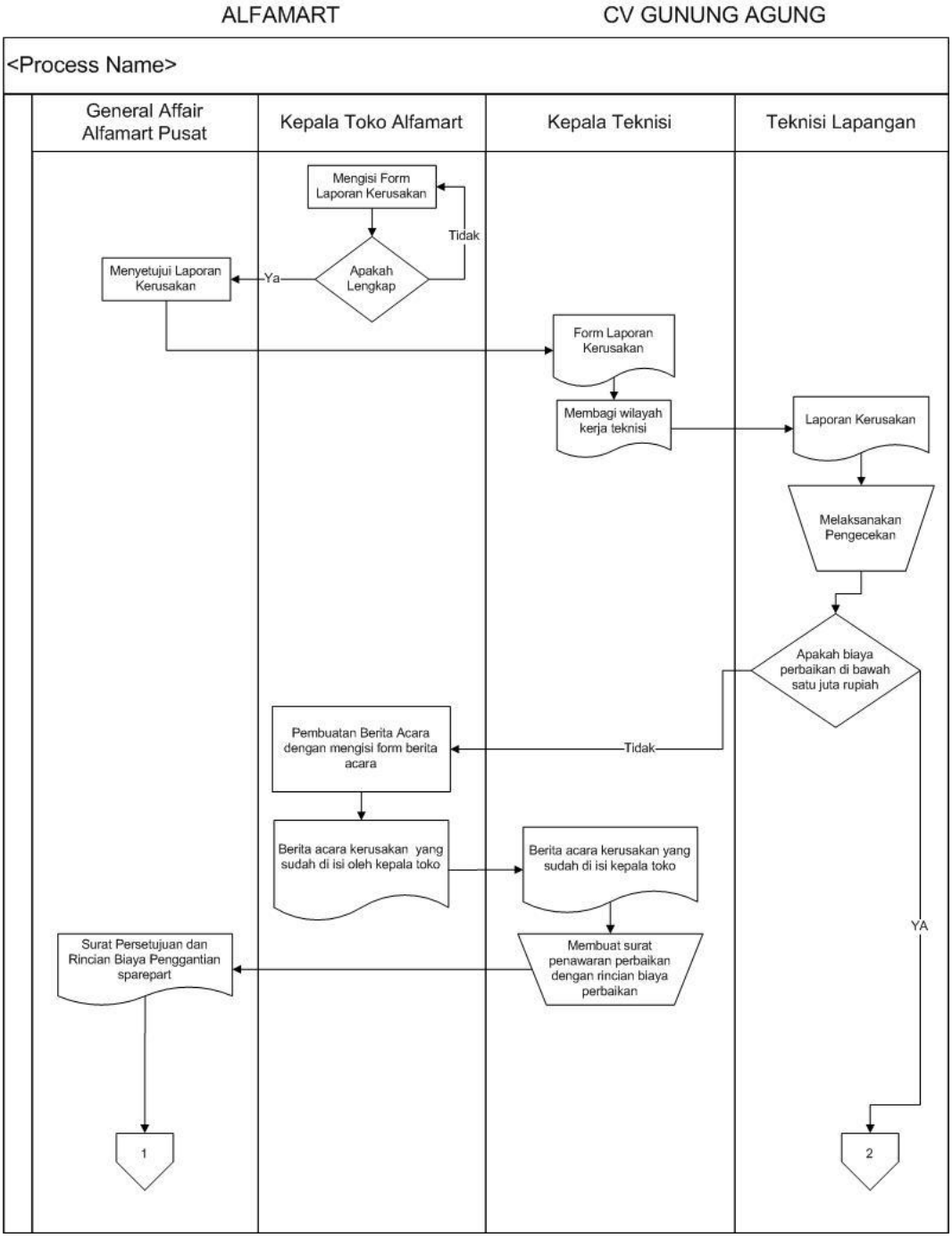
A1 = Arsip Laporan Kerusakan , A2 = Arsip Rincian Kerusakan

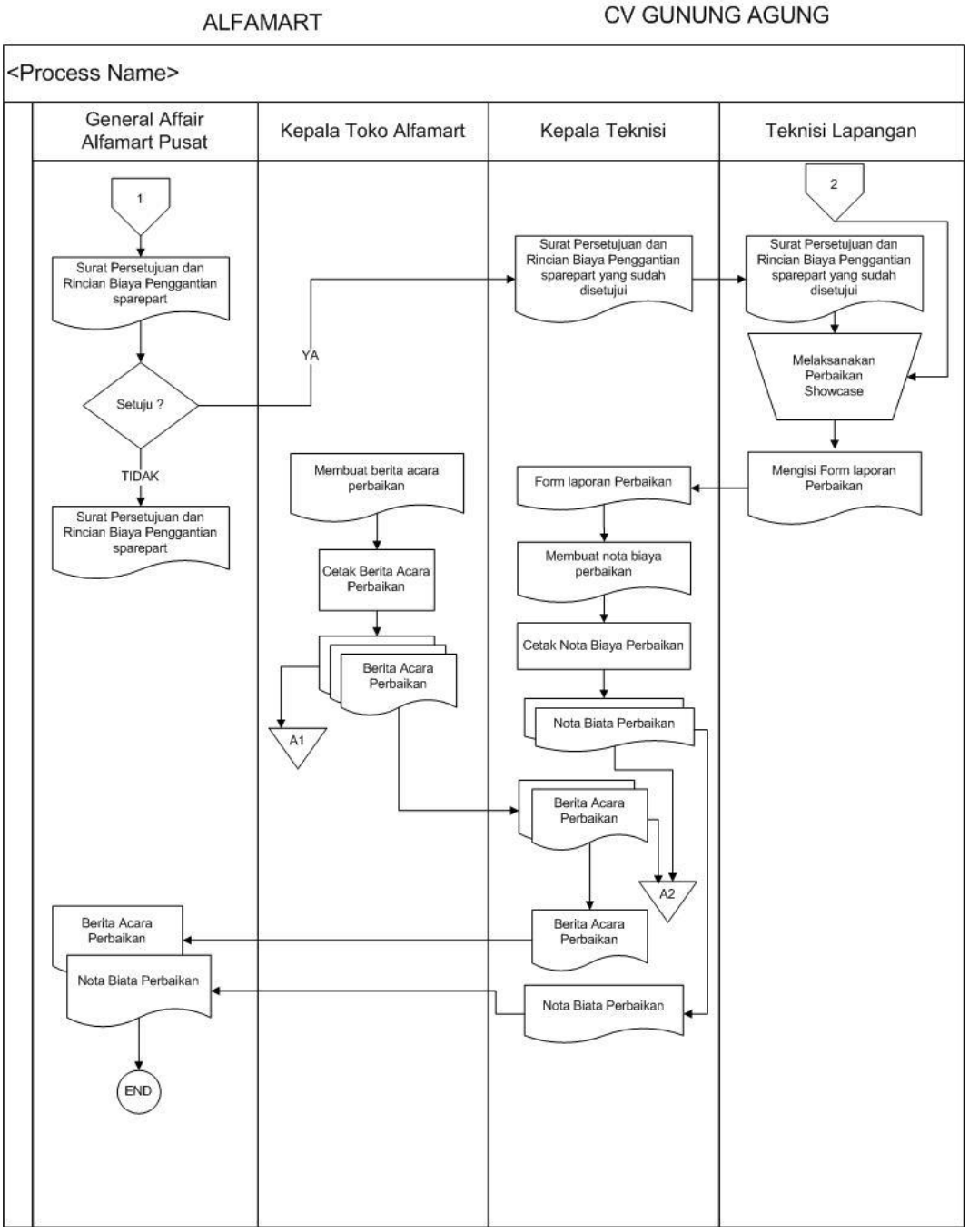
A3 = Arsip Nota Pembayaran , A4 = Arsip Berita Acara *Service*

Gambar 0.1 *Flow Map* sistem yang Berjalan pada CV Gunung Agung

3.2.2 Alur Sistem *service Showcase* yang dibangun

Alur sistem yang dibangun meruapakan alur sistem berdasarkan hasil analisis dan pengembangan dari sistem yang sudah berjalan. Dapat dilihat pada gambar 3.2



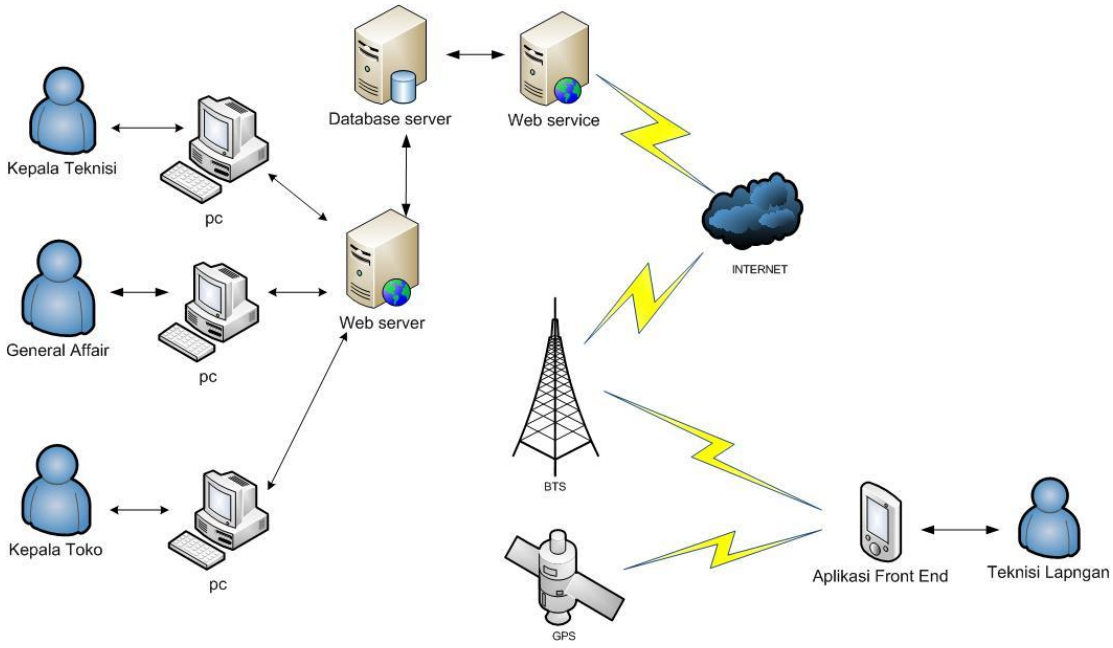


Gambar 0.2 Alur sistem yang dibangun

3.2.3 Deskripsi Sistem

Arsitektur fisik sistem yang dipakai dalam penelitian ini adalah menggunakan model arsitektur aplikasi *two tier* dimana arsitektur fisik sistem ini mempunyai empat komponen utama yaitu *Frontend* berbasis android *mobile* ,

Backend terdiri dari *Web*site untuk mengelola data, *middleware* yaitu *Web server* dan *database server*.



Gambar 0.3 Arsitektur Sistem

3.2.3.1 Prosedur *Service Showcase* yang di bangun

Prosedur *Service Showcase* pada CV Gunung Agung yang dibangun saat ini memiliki 4 entitas diantaranya adalah sebagai berikut:

1. ***General Affair*** **alfamart pusat.**
    a. Menerima Laporan Kerusakan dari kepala toko alfamart.
    b. Menyetujui Laporan Kerusakan dari kepala toko alfamart.
    c. Menerima rincian kerusakan , rincian biaya penggantian *sparepart*, serta berita acara hasil pengecekan *Showcase* yang mengalami kerusakan dengan biaya perbaikan diatas satu jutarupiah untuk di setujui apakah di lakukan perbaikan dan penggantian *sparepart*, jika setuju maka pilih opsi menyetujui laporan pengecekan untuk *Showcase* yang mengalami kerusakan dengan biaya perbaikan diatas satu juta rupiah yang sudah di setujui untuk dilakukan perbaikan dan penggantian *sparepart*.

d. Menerima laporan dari kepala teknisi CV Gunung Agung perihal hasil perbaikan dari kepala teknisi yang di sertai berita acara perbaikan , dan nota biaya pembayaran yang di tandatangani kepala teknisi.

2. Kepala Teknisi

a. Menerima Surat Laporan Kerusakan dari *General Affair* alfamart pusat.
b. Mengecek Kelengkapan Surat Laporan Kerusakan jika lengkap maka kepala teknisi menentukan teknisi lapangan yang diberi tugas untuk melakukan perbaikan .
c. Menerima berita acara pengecekan dari kepala toko yang telah di tanda tangani kepala toko untuk melaksanakan perbaikan dan penggantian *sparepart* apabila kerusakan dengan biaya diatas satu juta rupiah yang di alami toko alfamart .
d. Membuat rincian kerusakan dan rincian biaya penggantian *sparepart* untuk *Showcase* yang mengalami kerusakan dengan biaya perbaikan diatas satu juta rupiah.
e. Menerima rincian kerusakan dan rincian biaya penggantian *sparepart* dari *General Affair* alfamart pusat yang sudah di setujui untuk di lakukan perbaikan dan penggantian *sparepart*.
f. Menerima laporan perbaikan dari teknisi lapangan.

3. Teknisi Lapangan

a. Menerima Laporan Kerusakan yang diberikan oleh kepala teknisi.
b. Melakukan Pengecekan *Showcase* di toko alfamart, jika kerusakan biaya perbaikan di bawah satu juta rupiah maka langsung diperbaiki , jika kerusakan *Showcase* memerlukan biaya perbaikan diatas satu juta rupiah maka teknisi lapangan meminta kepala toko alfamart agar membuat berita acara bahwa kerusakan *Showcase* memerlukan biaya diatas satu juta rupiah.
c. Teknisi memperbaiki *Showcase* yang mengalami kerusakan.

d. Membuat laporan perbaikan jika *Showcase* sudah berfungsi dengan normal.
e. Teknisi lapangan memberikan laporan perbaikan kepada kepala teknisi.

4. Kepala Toko Alfamart .

a. Membuat laporan kerusakan *Showcase* kepada *General Affair General Affair* alfamart pusat.
b. Membuat berita acara kerusakan *Showcase* yang memerlukan biaya perbaikan diatas satu juta rupiah.
c. Membuat berita acara *Showcase* yang sudah diperiksa, diperbaiki dan berfungsi dengan normal sesuai laporan perbaikan.

Selain itu Pada gambar 3.2 menjelaskan alur sistem yang akan di bangun. Aplikasi *Backend* ini berupa aplikasi *Web* yang akan berfungsi mengolah semua data dan sebagai penghubung dengan alikasi *Frontend*. Pada aplikasi *Backend* ini diperlukan koneksi internet untuk proses komunikasi data. Aplikasi *Frontend* digunakan pada *Smartphone* yang memiliki sistem operasi Android, pada aplikasi *Frontend* ini terdapat semua informasi mengenai data laporan kerusakan, laporan data *Service* , selain itu pada aplikasi ini juga tersedia fitur peta lokasi yang akan menampilkan lokasi toko yang mengalami kerusakan serta lokasi terdekat yang dapat di tempuh untuk mencapai lokasi toko yang mengalami kerusakan. Untuk dapat menjalankan semua fitur yang ada pada aplikasi *Frontend* diperlukan koneksi internet untuk mengakses data dan diperlukan juga koneksi *gps satelite* untuk mendapatkan infromasi mengenai lokasi toko. Untuk lebih jelas mengnai sistem yang akan di bangun pada aplikasi *Frontend* dan aplikasi *Backend* di jelaskan pada bab 3.2.3.1 dan 3.2.3.2 .

3.2.3.2 Aplikasi *Frontend*

Aplikasi *Frontend* yaitu aplikasi yang digunakan oleh teknisi lapangan . aplikasi ini berfungsi sebagai antarmuka antara pengguna dan sistem.Pada aplikasi *Frontend* ini seorang teknisi lapangan dapat melakukan *login* dan data yang ditampilkan adalah data kerja teknisi lapangan yang melakukan *login* , membuat

data laporan *Service*. Pada aplikasi *Frontend* ini diperlukan koneksi internet untuk mengakses data dan *gps satellite* untuk mengetahui titik lokasi toko yang mengalami kerusakan.

3.2.3.2.1 Analisis Penentuan Lokasi Dengan GPS

Pada penelitian ini metode yang digunakan untuk menentukan posisi suatu lokasi adalah metode *Cell-Id* (*Cell Identification*) yaitu penentuan posisi didasarkan pada daerah geografis yang tercakup oleh sebuah cell berhubungan dengan daerah cakupan dari sinyal radio. Ketika sebuah *smartphone* tehubung secara aktif dengan sebuah base stasion, berarti *smartphone* tersebut diasumsikan berada dalam cell dari base station tersebut. Untuk mengukur jarak dan arah handset dari base station tidak dapat diketahui dengan pasti. Oleh karena itu , untuk lebih meningkatkan lagi akurasi hasil pencarian , metode *Cell ID* ini dapat dikombinasikan dengan metode *Timing Advanced (TA)* , dengan menggunakan TA ini, metode *Cell ID* akan ditambahkan sebuah fungsionalitas untuk menghitung *Round Trip Time (RTT),* yatu waktu transmisi sebuah frame (dari base station ke *smartphone* dan waktu penerimaan sebuah frame (dari *smartphone* ke base station). Dengan tambahan metode ini jarak antara *smartphone* dan *base station* dapat ditentukan dengan keakuratan 50 M.

Berikut ini salah satu rumus yang sering digunakan untuk menentukan lokasi jarak lokasi pada metode *Cell ID* dengan mengabaikan bentuk geografis baik bukit maupun lembah

Jarak	**=**	**Arc Cos(Cos(Lat1) x Cos(Lat2) x Cos(Long2-Long1) + Sin (Lat1) x Sin(Lat 2)) x R**

Ket :

Lat1 = Nilai Latitude dari Lokasi Awal.

Long1 = Nilai Longtitude dari Lokasi Awal.

Lat2 = Niai Latitude dari Lokasi Tujuan.

Long2 = Nilai Longtitude dari Lokasi Tujuan.

Dari hasil perhitungan Cell ID akan mendapatkan sebuah kordinat lokasi atau langtitude dan longtitude titik awal teknisi lapangan berada, sedangkan hasil perhitungan rumus jarak itu menentukan jarak antara teknisi lapangan dengan lokasi toko yang mengalami kerusakan, sehingga akan didapatkan jarak terdekat dari titik awal teknisi lapangan berada menuju titik lokasi toko yang mengalami kerusakan.

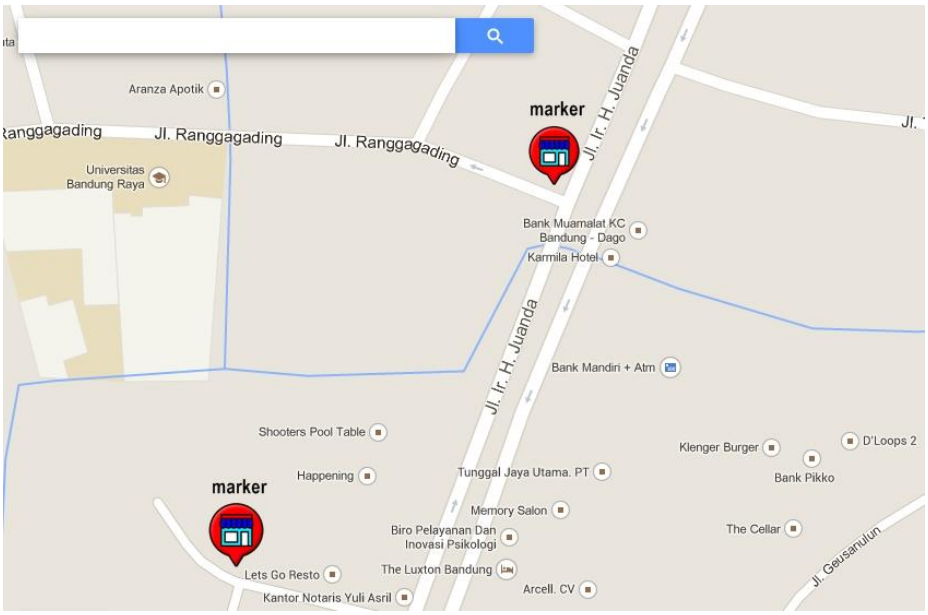
3.2.3.2.2 Elemen Google Maps API

Beberapa elem dan objek yang terdapat pada *Google Maps API* yang diterapkan dalam aplikasi adalah sebagai berikut:

1. *Markers*

   Marker adalah fungsionalitas symbol yang menandakan suatu titik kordinat lokasi. *Marker* digunakan untuk menandai lokasi seperti lokasi toko yang mengalami kerusakan pada peta *Google Maps.*

   Contoh *Marker* yang akan digunakan pada pembangunan aplikasi *reporting Service* berbasis android adalah seperti gambar 3.4



Gambar 0.4 Contoh Penggunaan *Marker*

2. *Directions*

   Directions adalah *utility* yang menangani masalah pencarian rute. *Directions* ini digunakan untuk memadu rute jalan dari teknisi lapangan menuju tempat lokasi toko yang mengalami kerusakan pada peta *Google Maps* seperti pada gambar 3.5



Gambar 0.5 Contoh penggunaan *Direction*

3. *Reverse GeoCoding*

   Reverse GeoCoding adalah *utility* untuk memproses konversi alamat latitude dan longtitude berguna untuk menempatkan penanda atau posisi sesuai dengan inputan dari longtitude dan latitude yang kita inputkan contoh seperti terlihat pada gambar 3.6



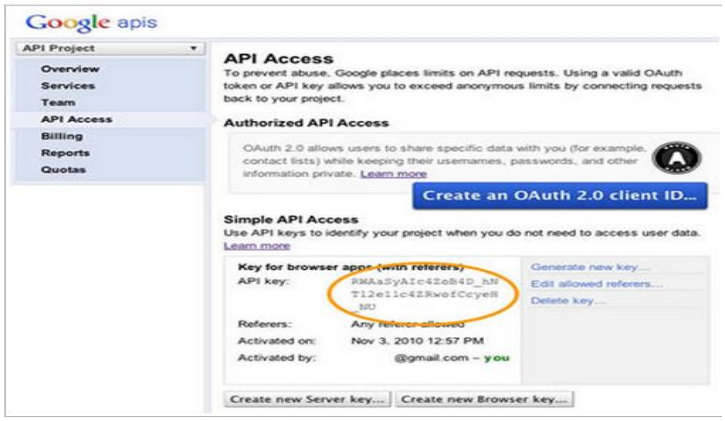
Gambar 0.6 Contoh penggunaan *Geocoding*

4. *Location Manager*

   Location Manager adalah API yang paling penting, untuk memberikan akses ke layanan sistem dengan lokasi pemetaan dan GPS. Dalam hal ini *location Manager* berfungsi untuk mendapatkan update periodik lokasi geografis perangkat dengan memetakan nilai latitude dan longtitude kedalam *Google Maps.*

5. *Loading Google Maps Api*

   Loading Google Maps Api adalah koneksi script yang dibuat keserver *google Maps API* dengan menggunakan *API key* yang didapat pada saat mendaftar ke *Google Maps API* contohnya pada gambar 3.7

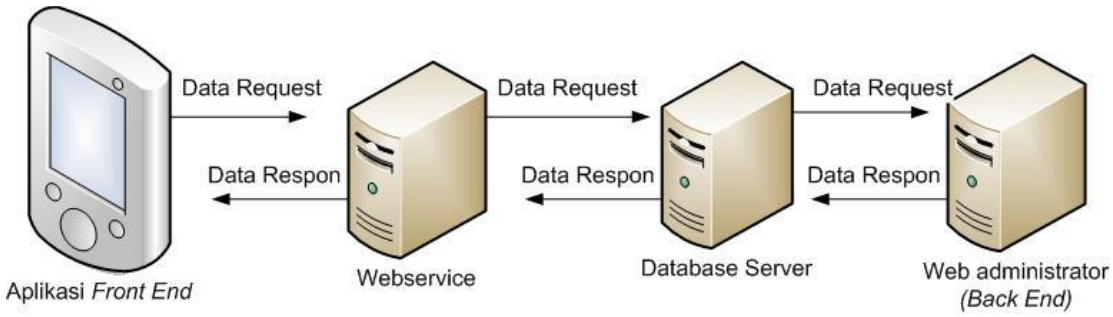


Gambar 0.7 Contoh Google *Maps API key*

3.2.3.3 Aplikasi Backend

Aplikasi *Backend* berfungsi sebgai penghubung antara aplikasi *Frontend* dengan *database* server. Pada aplikasi *Backend* ini terdapat dari tiga bagian yaitu *Web* administrator , *Web server* dan *database* server. *Web* administrator berfungsi untuk mengolah data yang dioprasikan oleh kepala teknisi , kepala toko alfamart, dan *General Affair* alfamart pusat. *Web Service* berfungsi untuk menghubungkan berbagai *platform* Android.*Web Service* yang digunkan untuk membangun sistem ini adalah JSON (*Javascript Object Nation*) , *Database server* berfungsi untuk menyimpan data – data yang telah diolah. Manajemen data yang digunakan pada aplikasi ini adalah MySQL, untuk aplikasi *Backend* terdapat tiga orang *user* yaitu kepala teknisi, kepala toko alfamart, *General Affair* alfamart pusat. Kepala teknisi bertugas untuk mengolah data pembagian wilayah kerja teknisi, membuat rincian biaya *Service*, Kepala Toko alfamart bertugas untuk mebuat berita acara baik berita acara *Service* dan berita acara kerusakan, sedangkan *General Affair* alfamart pusat memiliki akses untuk membuat laporan kerusakan, serta melihat laporan hasil *Service*.

3.2.4 Analisis Alur Kerja Sistem



Gambar 0.8 Alur Data Sistem

Analisi alur data yang terjadi pada sistem yang dibuat adalah sebagai berikut

1. Aplikasi *Frontend* berbasis android

   Aplikasi *Frontend* dalam hal ini merupakan analisis fungsionalitas aplikasi untuk mengakses data dari *database server* melalui *Web Service.* Aplikasi ini dibangun diatas *platform* android dan bekerja untuk *parsing* data dari

WebService yang bertipe XML untuk kemudian diolah pada *platform* android.

2. *Web Service*

   Web Service ini berfungsi untuk menghubungkan aplikasi *Frontend* dan aplikasi *Backend*. Cara kerja *Web Service* ini yaitu dengan mengambil data dari *database server,* dalam aplikasi *reporting Service* berbasis android *Web Service* digunakan untuk melakukan pertukaran data pembagian wilayah kerja teknisi lapngan, data laporan perbaikan,serta data laporan kerusakan yang berada di *database server* yang akan di konversikan kedalam format pertukaran data JSON. Dalam proses pengambilan data pada *Database server,* akan melalui beberapa proses diantaranya melakukan request kepada *server* melalui script PHP dimana koneksinya berada pada script php tersebut. Kemudian php akan membaca request dari sistem dan akan memprosesnya untuk melakukan pengambilan data dari *database server*.contoh : pertukaran data saat melakukan login.

```
<?php
        mysql_connect($DB_SERVER,$DB_USER,$DB_PASSWORD);
mysql_select_db($DB_NAME);
$u = $_POST['user'];
$p = $_POST['password'];
$query = mysql_query("SELECT username, password FROM tbl_user WHERE usename='.$u.'
and password='.$p.';

$r = array();
$c = mysql_num_rows($query);
If($query>0){
        $r('tbl_user') = array();
        While($a = mysql_fetch_array($query)){
                $b = array ();
                $b["username"] = $a["username"];
                $b["password"]= $a["password"];
                array_push($r["tbl_user"], $b );
        }
        $r["success"]=1;
        //fungsi encode merubah array kedalam objek
        echo json_encode($r);
}else{
        $r["success"]=0;
        echo json_encode($r);
}
Mysl_close();
?>
```

Gambar 0.9 Mekanisme perubahan data ke bentuk *Object* (JSON)

```
Try {  json Object = new JSONObejct( result );
        detail = new Login( );
            //Mengambil user dan password dari Object JSON
          detail.setUsername(json Object.getString("user"));
          detail.setPassword(json Object.getString("password"));
return detail;
}
```

Gambar 0.10 Script Parsing *Object* JSON

3. *Web Administrator*

   Web Administrator disini adalah berfungsi sebagai pengelola *Backend*. *Web Administrator* bertugas untuk mengolah data yang ada pada bagian *Service* CV Gunung Agung diantaranya yaitu tambah data,edit data, cari data, export data dan hapus data. *Web Administrator* disini melakukan konfigurasi *cross* antara *Web Service* dengan *Aplikasi Frontend.*

4. Protocol HTTP

   Protocol HTTP adalah sebuah protocol yang diciptakan untuk menyediakan perintah dalam aktifitas komunikasi antar jaringan, yakni komunikasi antara jaringan client dengan komupter server. Pada aplikasi *reporting Service* berbasis android Protocol HTTP digunakan untuk melakukan pertukaran data laporan kerusakan, data laporan perbaikan , serta data bagian kerja teknisi lapangan yang terdapat pada *database server* melalui koneksi internet.

3.2.5 Analisis Kebutuhan Non Fungsional

Analisis dan kebutuhan non fungsional meliputi analisis dan kebutuhan perangkat keras serta analisis dan kebutuhan perangkat lunak. Adapun kebutuhan non-fungsional untuk menjalankan aplikasi ini meliputi kebutuhan perangkat keras,kebutuhan perangkat lunak dan pengguna sistem yang akan memakai aplikasi. Analisis kebutuhan non-fungsional bertujuan agar aplikasi yang dibangun dapat digunakan sesuai dengan kebutuhan pengguna aplikasi dalam mencari informasi yang dibutuhkan.

3.2.5.1 Analisis Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan berdasarkan kebutuhan minimal yang harus terpenuhi untuk menggunakan sistem ini adalah sebagai berikut pada tabel 3.1

Tabel 0.1 Analisis Kebutuhan Perangkat Keras

Aplikasi *Frontend*	Aplikasi *Backend*
1. Telepon seluler (OS *Android* Minimal 2.2 *Froyo*)	1. PC/Laptop dengan spesifikasi: 1. *Processor Core 2 duo* 2.93GHz 2. RAM 2GB *3.* *Harddisk* 250GB
2. Memiliki koneksi internet EDGE,GPRS,2G dan 3G	2. Koneksi Internet

3.2.5.2 Analisis Kebutuhan Perangkat Lunak

Analisis perangkat lunak terdiri dari spesifikasi minimum perangkat lunak yang dibutuhkan untuk membangun aplikasi *Reporting Service* Berbasis Android pada tabel 3.2 :

Tabel 0.2 Analsis Kebutuhan Perangkat Lunak

Aplikasi *Frontend*	Aplikasi *Backend*
1. Eclipse Helios	*1. Dreamweaver 8*
2. *SQLite*	2. *MySQL*
3. ADT 21.1.0	*3. Mozilla Firefox 20.0.1 , Google Chrome, dan browser lainnya*
4. *Android SDK*	

Sedangkan perangkat lunak yang dibutuhkan untuk menjalankan aplikasi *reporting Service* berbasis Android pada tabel 3.3 :

Tabel 0.3 Perangkat Lunak Untuk menjalankan Aplikasi

Aplikasi *Frontend*	Aplikasi *Backend*
1. Sistem Operasi *Android* Minimal Versi 2.2 *Froyo*	1. Browser *Mozilla Firefox*, *Internet Explorer*, *Chrome* atau browser yang lain.

3.2.5.3 Analisis Pengguna Sistem *(User)*

Spesifikasi pengguna sistem dimaksudkan untuk mengetahui siapa saja aktor yang terlibat dalam menjalankan sistem. Berdasarkan hasil pengamatan, pengguna yang terlibat dalam sistem yang sedang berjalan ialah:

1. Kepala Teknisi : yaitu pengelola aplikasi *Reporting Service* berbasis android yang memiliki kapabilitas untuk memonitoring kinerja teknisi, memonitoring laporan kerusakan, membagi wilayah kerja setiap teknisi, membuat nota rincian biaya perbaikan yang kemudian di kirim kepada *General Affair* alfamart pusat, membuat surat pengajuan perbaikan yang mengalami kerusakan diatas satu juta rupiah.
2. Kepala Toko alfamart : yaitu pengguna pada bagian *Backend* yang memiliki kapabilitas untuk membuat berita acara kerusakan, dan berita acara perbaikan dan membuat laporan kerusakan.
3. *General Affair* alfamart pusat : yaitu pengguna pada bagian *Backend* yang memiliki kapabilitas untuk menyetujui laporan kerusakan, menerima berta acara , menyetujui atau tidaknya laporan kerusakan yang memerlukan biaya perbaikan diatas satu juta rupiah, serta menerima laporan perbaikan yang dikirim oleh kepala teknisi.
4. Teknisi lapangan : yaitu pengguna pada bagian *Frontend* yang memiliki kapabilitas untuk membuat laporan perbaikan, melihat wilayah kerja setiap teknisi, melakukan pencarian toko alfamart yang mengalami kerusakan.
5. Berikut karakteristik pengguna yang terlibat dalam sistem yang berjalan ditunjukkan oleh tabel 3.4 dan 3.5 :

Tabel 0.4 Karakteristik Pengguna *Backend*

Pengguna	Tanggung Jawab	Hak Akses	Tingkat Keterampilan
Kepala Teknisi	Melakukan Pemeliharaan terhadap sistem	Mengakses dan mengelola data laporan *Service*, Nota Pembayaran	Mengenal Internet dan terbiasa melakukan browsing.
Kepala Toko	Menjaga Kerahasiaan *ID* dan *Password*	Mengakses dan membuat berita acara , dan laporan kerusakan	Mengenal Internet dan terbiasa melakukan browsing.
***General Affair* alfamart pusat**	Menjaga Kerahasiaan *ID* dan *Password*	Mengakses , menyetujui laporan kerusakan, melihat jumlah kerusakan yang	Mengenal Internet dan terbiasa melakukan browsing.

		sudah di tangani , melihat nota rincian biaya *Service*	

Tabel 0.5 karakteristik Pengguna *Frontend*

Pengguna	Tanggung Jawab	Hak Akses	Tingkat Keterampilan
Teknisi Lapangan	Menjaga Kerahasiaan *ID* dan *Password*	Mengakses , membuat laporan *Service* , melihat lokasi toko alfamart yang mengalami kerusakan, melihat wilayah kerja.	Mengerti dan bisa menjalankan aplikasi berbasis *mobile*

3.2.6 Analisis Kebutuhan Fungsional

Kebutuhan fungsional berhubungan dengan fitur *software* yang akan dibuat atau dikembangkan. Berikut adalah tahapan analisis kebutuhan fungsional aplikasi *reporting Service* berbasis android.

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan.

Analisis yang dilakukan dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain identifikasi aktor, *Use Case diagram*, *skenario*, *Sequence diagram*, *activity diagram*, dan *class diagram*.

3.2.6.1 Analisis Kebutuhan Sistem

Sebelum melakukan perancangan sistem, terlebih dahulu dilakukan analisis kebutuhan sistem yaitu dengan cara melakukan wawancara terhadap pengguna. Hal ini dimaksudkan agar aplikasi yang dirancang sesuai dengan kebutuhan pengguna. Adapun kebutuhan sistem yang diperlukan antara lain :

1. Menampilkan informasi data laporan kerusakan, laporan *Service* , data berita acara.
2. Memberikan Informasi data toko dan konsumen yang mengalami kerusakan.

3. Memberikan informasi wilayah kerja setiap teknisi yang melakukan login.
4. Menampilkan data lokasi toko alfamart yang mengalami kerusakan.

3.2.6.2 Analisis Kebutuhan Data

Sistem yang akan dibangun merupakan sistem yang digunakan untuk mempermudah dalam membuat laporan kerusakan, mempermudah pembuatan berita acara, mempermudah pembuatan laporan perbaikan, serta mempermudah dalam pencarian lokasi toko alfamart. Untuk aplikasi *Backend* yang digunakan oleh *General Affair* alfamart pusat untuk menyetujui laporan kerusakan, serta melihat laporan hasil perbaikan. Sistem *Backend* yang digunakan oleh Kepala Toko Alfamart untuk mempermudah dalam pembuatan berita acara, pembuatan laporan kerusakan. Sistem *Backend* yang digunakan oleh kepala teknisi untuk melihat laporan kerusakan dari *General Affair* alfamart pusat , menentukan wilayah kerja teknisi, mengirim berita acara hasil pemeriksaan kepada *General Affair* alfamart pusat, melihat hasil laporan perbaikan, mengirim laporan perbaikan kepada *General Affair* alfamart pusat. dan untuk teknisi lapangan dapat melakukan proses pemanggilan method atau fungsi yang ada pada *Web Service* untuk mendapatkan data wilayah kerja, data laporan kerusakan serta pembuatan laporan perbaikan. Sehingga untuk membangun sistem di butuhkan data berupa form laporan kerusakan, laporan perbaikan, dan berita acara yang diperoleh dari Perusahaan CV Gunung Agung.

3.2.6.3 Spesifikasi Sistem

Dalam pembanguanan aplikasi *reporting Service* berbasis android diperukan spesifikasi sebagai berikut :

1. Sistem dikembangkan dengan model arsitektur *two tier.* Aplikasi *Frontend* sebagai aplikasi *mobile* dengan *platform* android. Sedangkan disisi server atau *Backend* akan dikembangkan aplikasi *Web* sebagai layanan untuk mengolah data dan penghubung antara aplikasi *mobile* dengan *database server*. *Database* yang digunakan untuk menyimpan data adalah MySQL.

2. Aplikasi *reporting Service* yang disediakan dibatasi pada informasi laporan kerusakan, wilayah kerja teknisi, pembuatan surat persetujuan , pengolahan data sparepart, pengolahan data laporan perbaikan, pembuatan data berita acara, pembuatan nota biaya perbaikan.
3. Spesifikasi sistem dalam sisi *Frontend* yaitu :
    I. Perangkat yang digunakan harus bisa digunakan dimana saja dan kapan serta memiliki akses internet.
    II. Aplikasi ini menyediakan tentang laporan kerusakan, laporan perbaikan, informasi berita acara, lokasi pada peta yang menggunakan *google maps* dan lokasi *realtime* pengguna dengan perangkat yang terintegrasi GPS.
    III. Untuk *user* yang sebelumnya melakukan registrasi pada aplikasi *Backend* serta telah melakukan kostomisasi data dapat melakukan login dan dapat melihat data sesuai hak akses *user* tersebut.
    IV. Aplikasi ini memiliki ukuran yang minimum sehingga mampu disimpan dalam perangkat *mobile.*
4. Spesifikasi sistem dalam sisi *Backend* yaitu :
    I. Sistem dapat memfasilitasi pengelolaan data perbaikan CV Gunung Agung.
    II. Sistem dapat memfasilitasi kepala teknisi untuk melakukan pengolahan data biaya perbaikan CV Gunung Agung
    III. Aplikasi *Backend* menyediakan fitur pembuatan berita acara, pembuatan laporan kerusakan,membagi wilayah kerja setiap teknisi, melihat laporan hasil perbaikan ,pembuatan surat persetujuan penggantian *sparepart* jika biaya perbaikan diatas satu juta rupiah, serta pembuatan nota rincian biaya perbaikan.

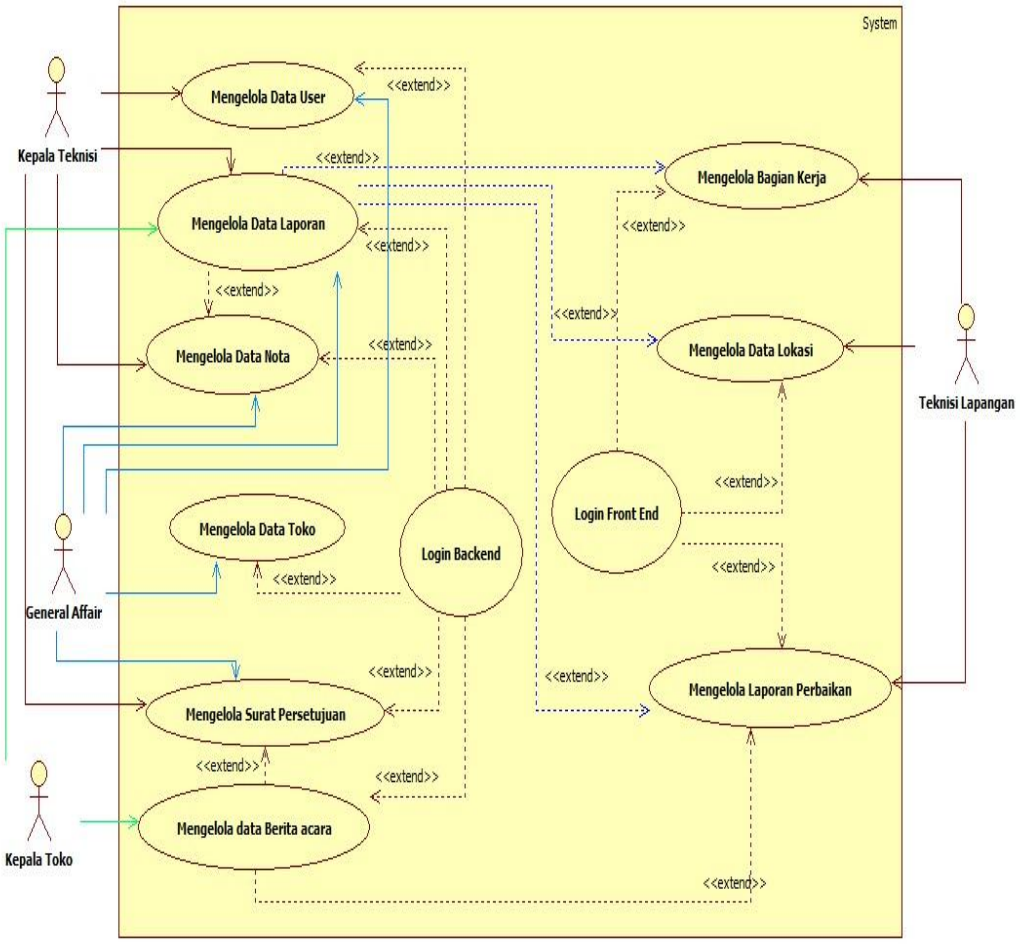
3.2.7 Pemodelan Sistem

Pemodelan sistem berfungsi untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem pada aplikasi serta menentukan kelas yang dibutuhkan

untuk realisasi fungsi - fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram.

3.2.7.1 Use Case Diagram

Use Case diagram Digunakan untuk memodelkan atau menggambarkan batasan sistem dan fungsi-fungsi utamanya. Mendiskripsikan fungsi dari sebuah sistem dari perspektif pengguna, *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebauh sistem dipakai. *Use Case* terdiri dari tiga bagian yaitu identifikasi *Use Case*, identifikasi *aktor*, dan scenario *Use Case*.



Gambar 0.11 *Use Case* Aplikasi *Reporting Service*

3.2.7.2 Identifikasi Aktor

Dalam aplikasi yang dibangun memiliki beberapa actor yang berperan untuk menjalankan sistem diantaranya adalah dapat dilihat pada tabel 3.6 :

Tabel 0.6 Identifikasi Aktor

No	Aktor	Deskripsi
A-01	Kepala Teknisi	Merupakan aktor yang dapat mengakses aplikasi *Backend* dan memiliki kapabilitas memonitoring kinerja teknisi, memonitoring laporan kerusakan, membagi wilayah kerja setiap teknisi, membuat nota rincian biaya *Service* yang kemudian di kirim kepada *General Affair* alfamart pusat, membuat surat pengajuan perbaikan yang mengalami kerusakan diatas satu juta rupiah.
A-02	Teknisi Lapangan	Merupakan aktor yang dapat mengakses aplikasi *Frontend* dan memiliki kapabilitas untuk membuat laporan *Service*, melihat wilayah kerja setiap teknisi, melakukan pencarian toko alfamart yang mengalami kerusakan.
A-03	*General Affair* Alfamart Pusat	Merupakan aktor yang dapat mengakses aplikasi *Backend* dan memiliki kapabilitas untuk menyetujui laporan kerusakan, menerima berita acara , menyetujui atau tidaknya laporan kerusakan yang memerlukan biaya perbaikan diatas satu juta rupiah, serta menerima laporan perbaikan yang sudah ditangani.
A-04	Kepala Toko Alfamart	Merupakan aktor yang dapat mengakses aplikasi *Backend* dan memiliki kapabilitas untuk membuat laporan kerusakan , membuat berita acara kerusakan, dan berita acara perbaikan.

3.2.7.3 Identifikasi *Use Case*

Dari *Use Case* yang terdapat pada gambar 3.6 dapat di Identifikasikan bahwa pada sistem yang dibangun terdapat beberapa *Use Case* yang bekerja dalam sistem diantaranya dapat di lihat pada tabel 3.7 :

Tabel 0.7 Identifikasi Aktor pada *Use Case*

No	Nama *Use Case*	Deskripsi
UC-01	Login *Backend*	Fungsionalitas ini berguna untuk masuk ke dalam sistem *Backend*

UC-02	Mengelola Data *User*	Fungsionalitas ini berguna untuk mengelola data *User* seperti teknisi lapangan, kepala toko, *General Affair*.
UC-03	Mengelola Data laporan	Fungsionalitas ini berguna untuk mengelola data laporan baik hasil laporan perbaikan dari teknisi lapangan dan laporan kerusakan dari *General Affair* yang akan di laporkan kepada kepala teknisi.
UC-04	Mengelola Data Nota	Fungsionalitas ini berguna untuk mengelola data biaya penagihan hasil perbaikan yang akan di tagihkan kepada *General Affair* .
UC-05	Mengelola Data Toko	Fungsionalitas ini berguna untuk pengelolaan data toko yang di inputkan oleh *General Affair*.
UC-06	Mengelola Surat Persetujuan	Fungsionalitas ini berguna untuk membuat surat persetujuan, melihat surat persetujuan yang dibuat oleh Kepala Teknisi dan di tujukan kepada *General Affair*.
UC-07	Mengelola Data berita acara	Fungsionalitas ini berguna untuk mengelola data berita acara yang di gunakan oleh kepala toko dalam pembuatan berita acara.
UC-08	Login *Frontend*	Fungsionalitas ini berguna untuk masuk ke dalam sistem *Frontend*
UC-09	Mengelola Bagian Kerja	Fungsionalitas ini berguna untuk mengelola Bagian Kerja agar teknisi lapangan dapat melihat wilayah kerja atau bagian kerja yang sudah di tentukan oleh kepala teknisi.
UC-10	Mengelola Data Lokasi	Fungsionalitas ini berguna untuk mengelola data lokasi dimana teknisi lapangan berada dan petunjuk arah teknisi lapangan dalam mencari lokasi toko yang mengalami kerusakan.
UC-11	Mengelola Data *Service*	Fungsionalitas ini berguna untuk mengelola hasil *Service* perbaikan yang sudah di lakukan oleh teknisi lapangan.

3.2.7.4 Skenario *Use Case*

Skenario *Use Case* adalah alur cerita pada setiap *Use Case* yang berjalan dalam sistem *Use Case* dapat dilihat pada Tabel 3.8 Skenario *Use Case* – Tabel 3.18 Skenario *Use Case*

Tabel 0.8 Skenario *Use Case* UC-01

Identifikasi			
Nomor *Use Case*	UC-01		
Nama *Use Case*	Login *Backend*		
Deskripsi :	berguna untuk masuk ke dalam sistem *Backend*		
Aktor :	Kepala Teknisi, *General Affair* Alfamart Pusat , Kepala Toko Alfamart		
Kondisi Awal :	*Web* CV Gunung Agung		
Kondisi Ahir :	Berhasil Login / Gagal Login		
Skenario Utama			
No	**Aksi Aktor**	No	**Respon Sistem**
1	Masuk Kedalam *Web* CV Gunung Agung		
		2	Tampil Halaman Login *Backend*
3	Memasukan *User* Name Dan Password		
		4	Memvalidasi *User* Name dan Password
	Kondisi Ahir		Berhasil Login / Gagal Login

Tabel 0.9 Skenario *Use Case* UC-02

Identifikasi			
Nomor *Use Case*	UC-02		
Nama *Use Case*	Mengelola Data *User*		
Deskripsi :	Berguna untuk mengelola data *User* seperti teknisi lapangan, kepala toko, dan *General Affair*		
Aktor :	Kepala Teknisi		
Kondisi Awal :	Tampil Halaman Menu Kepala Teknisi		
Kondisi Ahir :	Tampil Halaman Data *User*		
Skenario Utama			
No	**Aksi Aktor**	No	**Respon Sistem**
1	Memilih Menu Data *User*		
		2	Tampil Halaman Data *User* (*General Affair*, Teknisi Lapangan , Kepala Toko)
3	Memilih Menu tambah data *user*		
		4	Tampil Halaman form data *user*

5	Mengisi *Field* Form Konsumen		
6	Klik Tombol Simpan		
		7	Tampil Halaman Data *User* (*General Affair*, Teknisi Lapangan , Kepala Toko)
8	Memilih Lihat Detail		
		9	Tampil Detail dari data yang sudah di pilih
10	Pilih button Kembali		
	Kondisi Ahir		Tampil Halaman Data *User*

Tabel 0.10 Skenario *Use Case* UC-03

Identifikasi	
Nomor *Use Case*	UC-03
Nama *Use Case*	Mengelola data laporan
Deskripsi :	Berguna untuk mengelola data laporan baik laporan hasil perbaikan maupun laporan kerusakan
Aktor :	Kepala Teknisi, *General Affair* , Kepala Toko
Kondisi Awal :	Tampil halaman laporan
Kondisi Ahir :	Tampil Halaman Data Laporan

Skenario Utama			
No	**Aksi Aktor**	No	**Respon Sistem**
1	Memilih Menu Laporan Kerusakan		
		2	Tampil Halaman Data Laporan Kerusakan
3	Memilih Menu Tambah Laporan Kerusakan		
		4	Tampil Form Laporan Kerusakan
5	Mengisi Form Laporan Kerusakan yang sudah tersedia		
6	Klik button Simpan		
		7	Tampil Halaman Data laporan Kerusakan
8	Pilih Menu Lihat Detail		
		9	Tampil Detail Dari Laporan Kerusakan
10	Pilih Menu Print		
		11	Print laporan kerusakan yang dipilih oleh *user*
12	Pilih Menu Kembali		

		13	Tampil Halaman Data laporan Kerusakan
14	Pilih Menu Laporan Perbaikan		
		15	Tampil Halaman Data Laporan Perbaikan
16	Memilih Menu Tambah Laporan Perbaikan		
		17	Tampil Form Laporan Perbaikan
18	Mengisi Form Laporan Perbaikan yang sudah tersedia		
19	Klik Button Simpan		
		20	Tampil Halaman Data Laporan Perbaikan jika data berhasil disimpan
21	Pilih Button Lihat Detail		
		22	Tampil Detail Laporan Perbaikan yang dipilih
23	Pilih button Kembali		
	Kondisi Ahir		Tampil Halaman Data Laporan

Tabel 0.11 Skenario *Use Case* UC-04

Identifikasi			
Nomor *Use Case*	UC-04		
Nama *Use Case*	Mengelola Data Nota		
Deskripsi :	Berguna untuk membuat nota biaya perbaikan sesuai hasil laporan perbaikan yang akan di tunjukan kepada *General Affair*		
Aktor :	Kepala Teknisi dan *General Affair*		
Kondisi Awal :	Tampil Halaman Menu *Backend*		
Kondisi Ahir :	Tampil Halaman Data Nota Biaya Perbaikan		
Skenario Utama			
No	**Aksi Aktor**	No	**Respon Sistem**
1	Memilih Menu Nota		
		2	Tampil Halaman Nota Biaya Perbaikan
3	Klik Tambah Data Nota		
		4	Tampil Form Nota Biaya
5	Mengisi Form Nota Biaya Pembayaran		
6	Pilih Button Simpan		
		7	Tampil Halaman Data Nota Biaya Perbaikan jika data berhasil disimpan

8	Pilih Menu Lihat Detail		
		9	Tampil Detail Data Nota Biaya yang dipilih
10	Pilih Menu Print Nota		
		11	Print Nota yang dipilih oleh *user*
12	Pilih Menu Kembali		
	Kondisi Ahir		Tampil Halaman Data Nota Biaya Perbaikan

Tabel 0.12 Skenario *Use Case* UC-05

Identifikasi			
Nomor *Use Case*	UC-05		
Nama *Use Case*	Mengelola Data Toko		
Deskripsi :	Berguna untuk menampilkan data toko dan mencari data toko yang mengalami kerusakan		
Aktor :	*General Affair*		
Kondisi Awal :	Tampil Halaman Menu *General Affair*		
Kondisi Ahir :	Tampil Halaman Data Toko		
Skenario Utama			
No	**Aksi Aktor**	No	**Respon Sistem**
1	Memilih Menu Data Toko		
		2	Tampil Halaman Data Toko
3	Klik Tambah Data Toko		
		4	Tampil Form Data Toko
5	Mengisi Form Data Toko		
6	Klik Tombol Simpan		
		7	Tampil Halaman Data Toko
8	Pilih Menu Detail		
		9	Tampil Detail Data Toko
10	Klik Button Kembali		
	Kondisi Ahir		Tampil Halaman Data Toko

Tabel 0.13 Skenario *Use Case* UC-06

Identifikasi	
Nomor *Use Case*	UC-06
Nama *Use Case*	Membuat Surat Persetujuan
Deskripsi :	berguna untuk melihat data surat persetujuan , membuat surat persetujuan
Aktor :	Kepala Teknisi, *General Affair*

Kondisi Awal :	Tampil Halaman Menu Surat Persetujuan		
Kondisi Ahir :	Tampil Data Surat Persetujuan		
Skenario Utama			
No	**Aksi Aktor**	No	**Respon Sistem**
1	Memilih Menu Surat Persetujuan		
		2	Tampil Data Surat Persetujuan
3	Pilih Menu Tambah Data Surat Persetujuan		
		4	Tampil Form Surat Perseujuan
5	Mengsi *Field* Form Surat Persetujuan		
6	Klik Tombol Simpan		
		7	Tampil Data Surat Persetujuan jika data berhasil disimpan
8	Pilih Menu Lihat Detail		
		9	Tampil Detail Data Surat Persetujuan
10	Pilih Button Print		
		11	Mencetak Surat Persetujuan
12	Pilih Button Kembali		
	Kondisi Ahir		Tampil Data Surat Persetujuan

Tabel 0.14 Skenario *Use Case* UC-07

Identifikasi			
Nomor *Use Case*	UC-07		
Nama *Use Case*	Mengelola Berita Acara		
Deskripsi :	Berguna untuk mengelola data berita acara		
Aktor :	Kepala Toko, *General Affair*, Kepala Teknisi		
Kondisi Awal :	Tampil Halaman Menu Berita Acara		
Kondisi Ahir	Tampil Data Berita Acara		
Skenario Utama			
No	**Aksi Aktor**	No	**Respon Sistem**
1	Memilih Menu Berita Acara		
		2	Tampil Data Berita Acara
3	Pilih Button Tambah Data Berita Acara		
		4	Tampil Form Data Berita Acara
5	Mengisi Form Data Berita Acara		
6	Pilih Button Simpan Jika form sudah terisi semua		
		7	Tampil Data Berita Acara Jika Data Berhasil Disimpan
8	Pilih Button Lihat Detail		
		9	Tampil Data Detail Berita Acara
10	Pilih Button Print		

		11	Mencetak Berita Acara Sesuai Data yang dipilih
12	Pilih Button Kembali		
	Kondisi Ahir		Tampil Data Berita Acara

Tabel 0.15 Skenario *Use Case* UC-08

Identifikasi			
Nomor *Use Case*	UC-08		
Nama *Use Case*	Login *Frontend*		
Deskripsi :	berguna untuk masuk ke dalam sistem *Frontend*		
Aktor :	Teknisi lapangan		
Kondisi Awal :	Aplikasi *Reporting Service* CV Gunung Agung		
Kondisi Ahir :	Berhasil Login / Gagal login		
Skenario Utama			
No	**Aksi Aktor**	No	**Respon Sistem**
1	Masuk Kedalam Aplikasi *Reporting Service* CV Gunung Agung		
		2	Tampil Halaman Login
3	Memasukan *User* Name Dan Password		
		4	Memvalidasi *User* Name dan Password
	Kondisi Ahir		Berhasil login / Gagal login

Tabel 0.16 Skenario *Use Case* UC-09

Identifikasi			
Nomor *Use Case*	UC-09		
Nama *Use Case*	Mengelola Bagain Kerja		
Deskripsi :	Berguna Untuk Mengelola Bagian kerja teknisi lapangan		
Aktor :	Teknisi Lapangan		
Kondisi Awal :	Tampil Halaman Utama Teknisi Lapangan		
Kondisi Ahir :	Tampil Data Bagian Kerja Teknisi		
Skenario Utama			
No	**Aksi Aktor**	No	**Respon Sistem**
1	Memilih Menu Bagian Kerja		
		2	Tampil Data Bagian Kerja Teknisi
3	Memilih button Lihat Detail		
		4	Tampil Data Detail Laporan Bagian Kerja
5	Pillih Button Lihat Lokasi		
		6	Tampil Lokasi Toko dalam Google Maps
7	Pilih Button Kembali		

		8	Tampil Data Detail Laporan
9	Pilih Button Kembali		
	Kondisi Ahir		Tampil Data Bagian Kerja Teknisi

Tabel 0.17 Skenario *Use Case* UC-10

Identifikasi			
Nomor *Use Case*	UC-10		
Nama *Use Case*	Mengelola Data Lokasi		
Deskripsi :	Berguna untuk memberikan informasi mengenai lokasi teknisi lapangan dan lokasi tujuan dari toko yang mengalami kerusakan		
Aktor :	Teknisi Lapangan		
Kondisi Awal :	Tampil Halaman Menu Utama Teknisi Lapangan		
Kondisi Ahir :	Tampil Halaman Data Lokasi Dalam Google Maps		
Skenario Utama			
No	**Aksi Aktor**	No	**Respon Sistem**
1	Memilih Menu Lokasi Sekarang		
		2	Tampil Data Lokasi Dalam Google Maps
3	Cari Lokasi		
		4	Tampil Form Lokasi Tujuan
5	Klik Tombol Cari		
		6	Menampilkan Lokasi Yang Dituju jika lokasi ditemukan
7	Pilih Button Kembali		
	Kondisi Ahir		Tampil Halaman Data Lokasi Dalam Google Maps

Tabel 0.18 Skenario *Use Case* UC-11

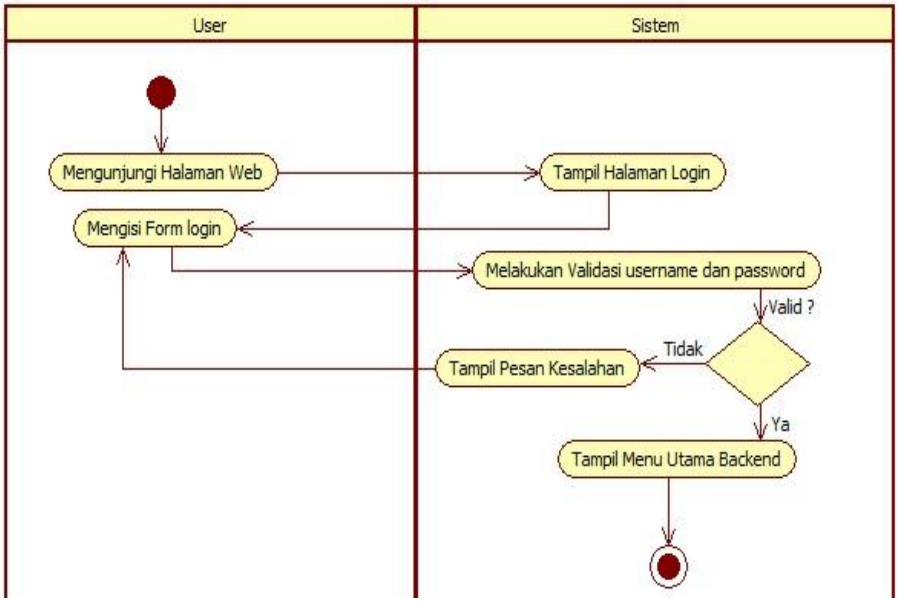
Identifikasi			
Nomor *Use Case*	UC-11		
Nama *Use Case*	Mengelola Data *Service*		
Deskripsi :	Berguna untuk mengelola data hasil *Service* yang dilakukan teknisi lapangan		
Aktor :	Teknisi Lapangan		
Kondisi Awal :	Tampil Halaman Teknisi Lapangan		
Kondisi Ahir :	Tampil Data Laporan Perbaikan		
Skenario Utama			
No	**Aksi Aktor**	No	**Respon Sistem**
1	Memilih Menu laporan kerusakan		
		2	Tampil Data Laporan kerusakan
3	Pilih Menu Tambah Data Laporan		
		4	Tampil Form Perbaikan

5	Mengisi Form Perbaikan		
6	Pilih Button Simpan		
	Kondisi Ahir		Tampil Data Laporan kerusakan

3.2.7.5 Activity Diagram

Activity Diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *Use Case* atau interaksi. Dapat di lihat dari gambar 3.12 – gambar 3.22 :

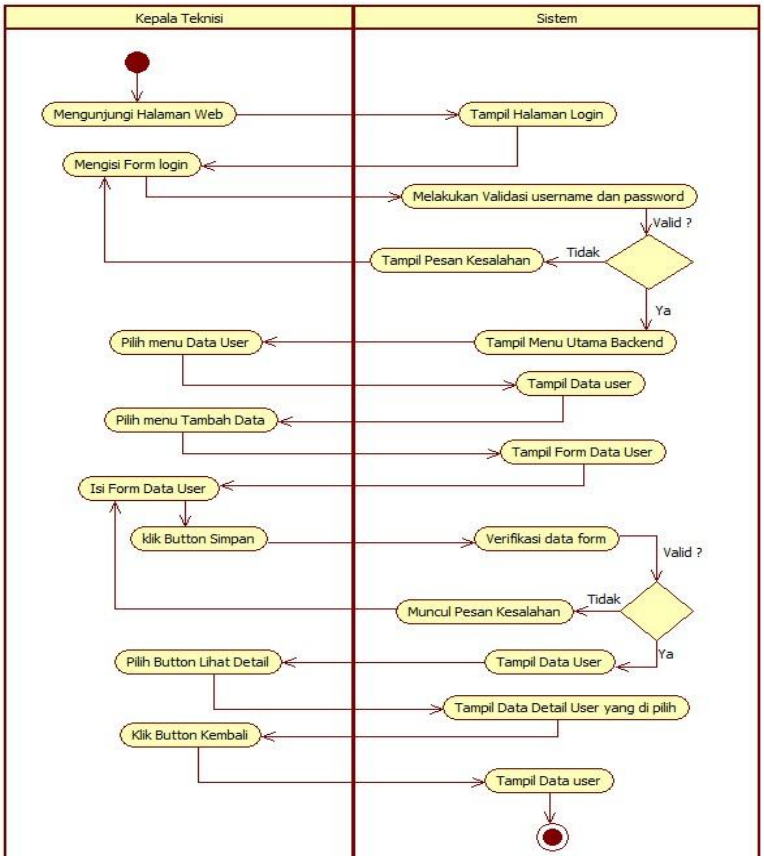
g. *Activity Diagram* Login *Backend*



Gambar 0.12 *Activity Diagram* Login *Backend*

Pada Gambar 3.12 Menjelaskan tentang aktifitas yang terjadi saat seorang *user* yang akan masuk kedalam sistem aplikasi *Backend* harus terlebih dahulu teridentifikasi agar *user* name dan password yang digunakan untuk melakukan login valid.

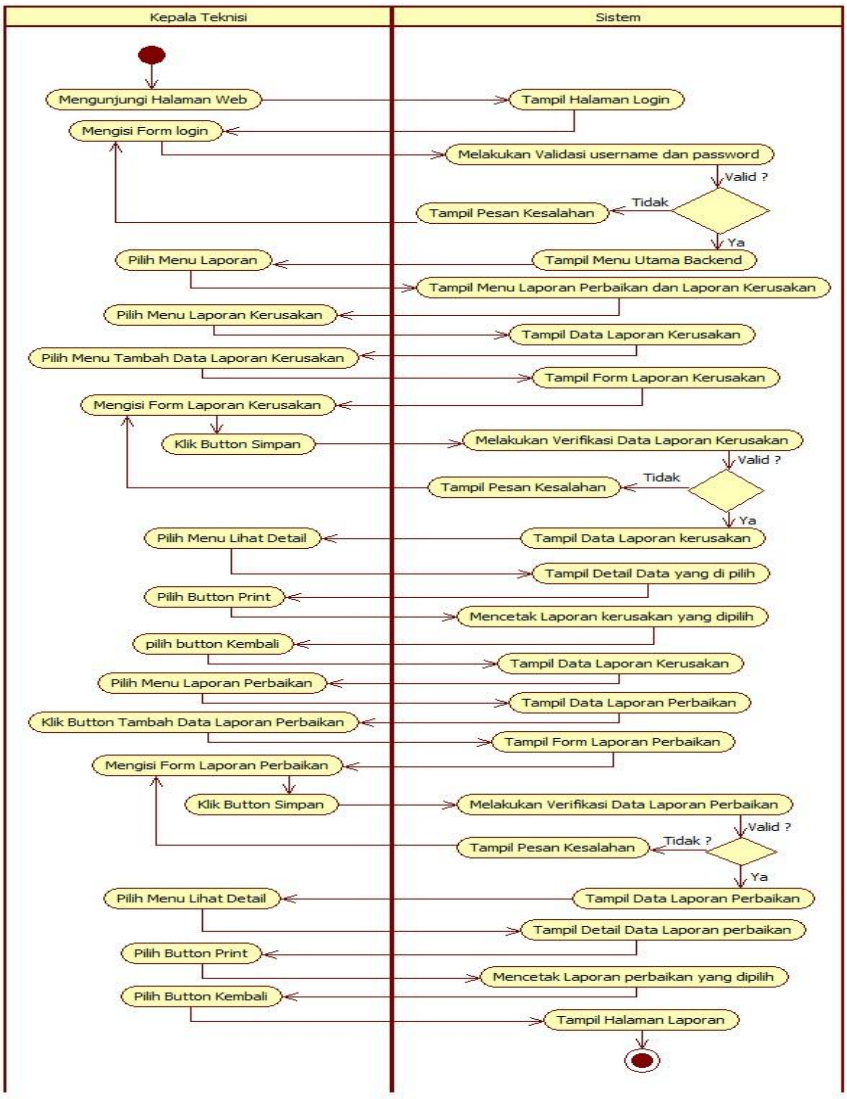
h. *Activity Diagram* Mengelola Data *User*



Gambar 0.13 *Activity Diagram* Mengelola Data *User*

Pada Gambar 3.13 Menjelaskan tentang aktifitas yang terjadi saat seorang kepala teknisi akan melakukan penambahan *user* baik *General Affair* ataupun teknisi lapangan. Pada awalnya kepala teknisi melakukan login kedalam *Web* CV Gunung Agung, kemudian akan menampilkan menu halaman utama untuk kepala teknisi, kemudian memilih menu data *user*, setelah dipilih maka akan tampil data *user* dari CV Gunung Agung dan kemudian untuk menambahnya kepala teknisi harus memilih button tambah data *user*. Setalah itu form data *user* akan muncul.isi form data *user* kemudian pilih simpan jika data field dalam form terisi semua maka data akan tersimpan jika terdapat field kosong maka data gagal tersimpan.

i. Activity Mengelola Data Laporan

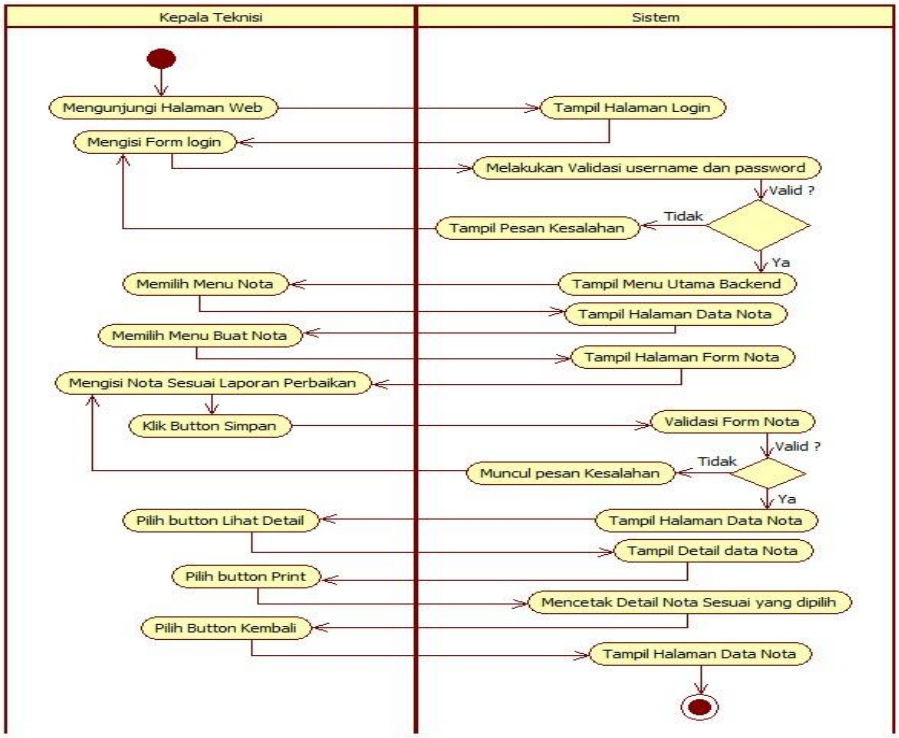


Gambar 0.14 *Activity Diagram* Mengelola Data Laporan

Pada Gambar 3.14 Menjelaskan tentang aktifitas yang terjadi saat seorang kepala teknisi akan melihat data laporan kerusakan dan *General Affair* Membuat laporan kerusakan. Pada awalnya kepala teknisi melakukan login kedalam *Web* CV Gunung Agung, kemudian akan menampilkan menu halaman utama untuk kepala teknisi, kemudian memilih menu lihat form kerusakan, kemudian sistem akan meload dari *database* untuk kemudian di tampilkan dalam *Web* berupa data kerusakan yang diterima oleh kepala teknisi dari *General Affair*, kemudian untuk

melihat detail rincian maka kepala teknisi memilih menu lihat detail maka akan menampilkan detail data laporan kerusakan.dan untuk laporan kerusakan hanya diisi oleh *General Affair* yang menerima laporan kerusakan oleh pihak toko.Sementara Untuk mengisi form laporan perbaikan di lakukan oleh teknisi lapangan.

j. *Activity Diagram* Mengelola Data Nota

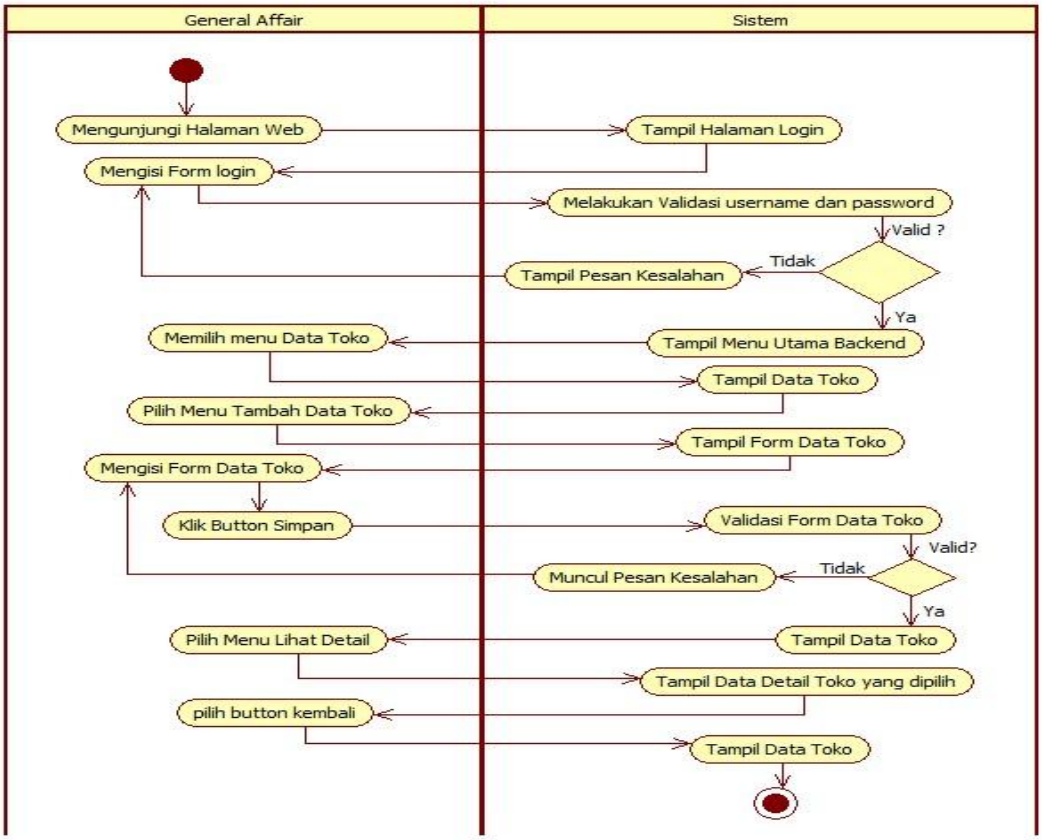


Gambar 0.15 Mengelola Data Nota

Pada Gambar 3.15 Menjelaskan tentang aktifitas yang terjadi saat seorang kepala teknisi akan membuat nota biaya perbaikan. Pada awalnya kepala teknisi melakukan login kedalam *Web* CV Gunung Agung, kemudian akan menampilkan menu halaman utama untuk kepala teknisi, kemudian memilih menu Nota, kemudian sistem akan meload dari *database* untuk menampilkan data Nota pembayaran, untuk menambah data data pilih menu buat nota isi form sesuai dengan laporan perbaikan. Setelah itu maka akan tampil kembali di halaman utama laporan kerusakan . pilih menu lihat detail untuk melihat detail data yang dipilih, pilih button

print jika data laporan kerusakan di butuhkan, pilih button kembali untuk kembali ke menu utama laporan.

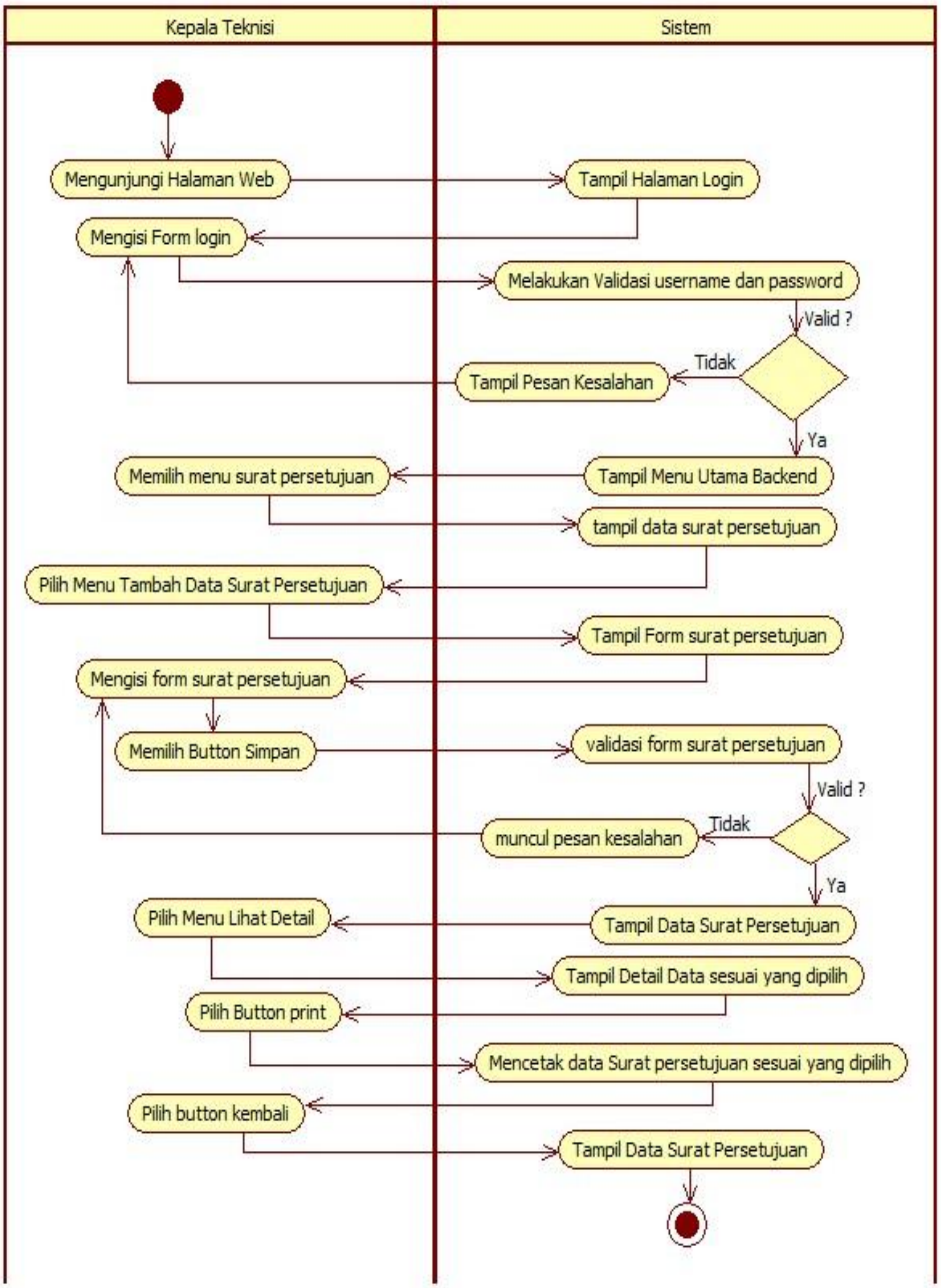
k. *Activity Diagram* Mengelola Data Toko



Gambar 0.16 *Activity Diagram* Mengelola data toko

Pada Gambar 3.16 Menjelaskan tentang aktifitas yang terjadi saat seorang *General Affair*. Pada awalnya *General Affair* mengakses halaman *Web* CV Gunung Agung, kemudian memilih menu data toko maka akan tampil data toko di halaman utama, untuk menambah data toko maka pilih button tambah data toko. Untuk melihat secara rinci data toko yang di pilih maka pilih menu lihat detail.

l. *Activity Diagram* Mengelola Surat Persetujuan

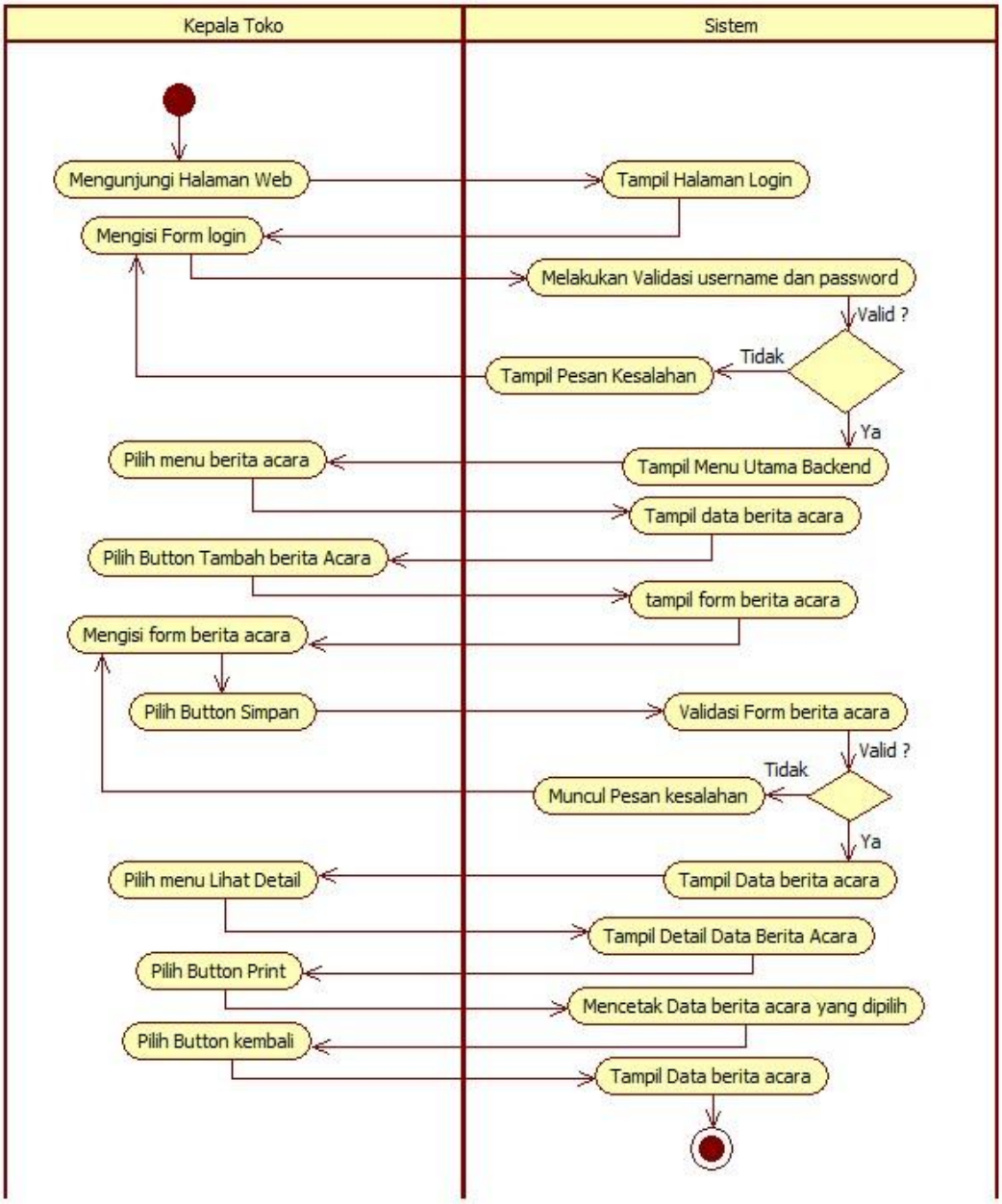


Gambar 0.17 *Activity Diagram* Mengelola Surat Persetujuan

Pada Gambar 3.17 Menjelaskan tentang aktifitas yang terjadi saat seorang kepala teknisi akan membuat surat persetujuan yang di tunjukan kepada *General*

Affair. Surat persetujuan di buat untuk toko yang mengalami kerusakan dengan biaya perbaikan diatas satujuta rupiah.

m. *Activity Diagram* Mengelola berita Acara

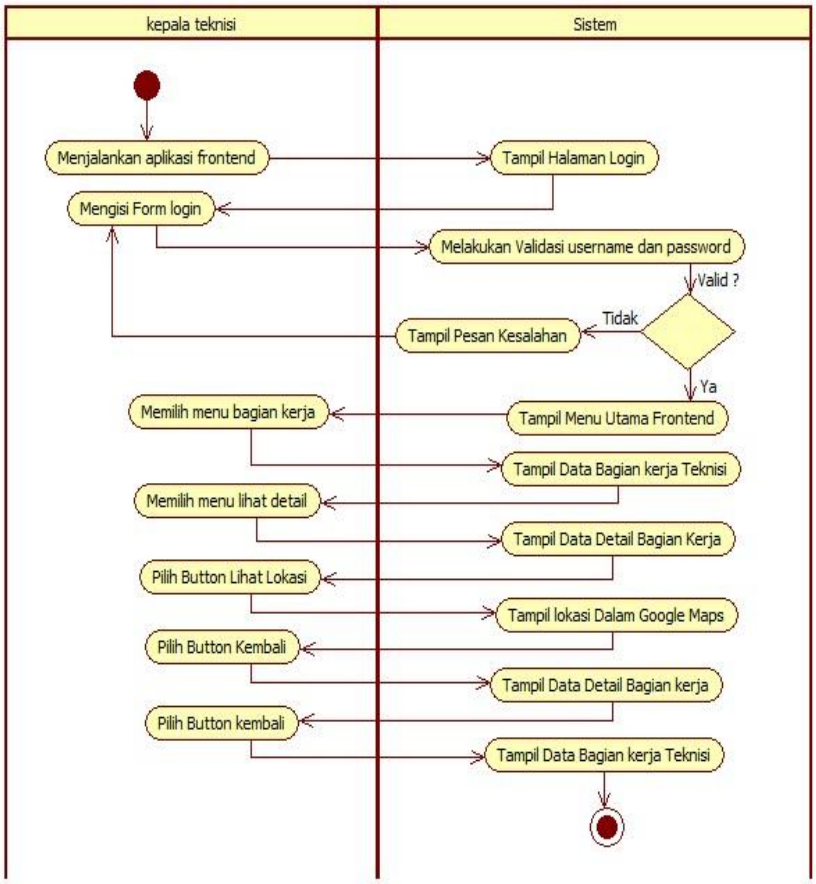


Gambar 0.18 *Activity Diagram* Mengelola Berita Acara

Pada Gambar 3.18 Menjelaskan tentang aktifitas yang terjadi saat seorang kepala Toko membuat berita acara, berita acara dibuat dimaksudkan untuk bukti

bahwa telah dilakukan pengecekan atau perbaikan unit mesin *Showcase* yang mengalami kerusakan. Berita acara akan di lampirkan pada pembuatan Surat persetujuan.

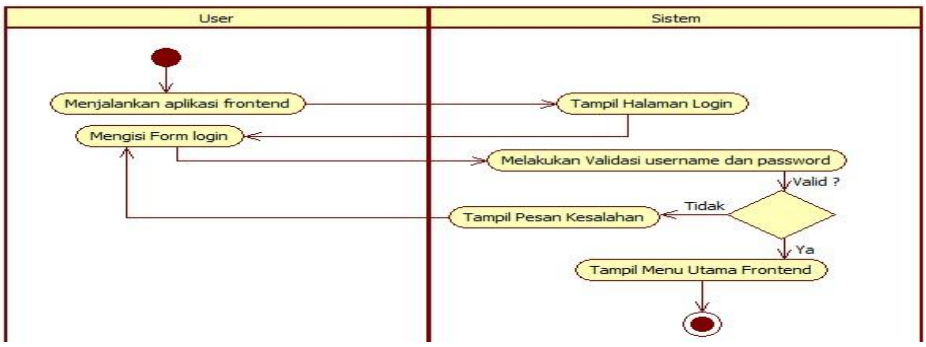
n. *Activity Diagram* Mengelola Bagian Kerja



Gambar 0.19 *Activity Diagram* Mengelola Bagian Kerja

Pada Gambar 3.19 Menjelaskan tentang aktifitas yang terjadi saat seorang teknisi lapangan yang berguna untuk mengetahui wilayah kerja atau bagian kerja teknisi tersebut, agar mudah di control oleh kepala teknisi, pada diagram mengelola bagian kerja teknisi lapangan juga dapat mencari lokasi toko yang mengalami kerusakan dengan bantuan data latitude dan langtitude toko yang terdapat pada laporan kerusakan.

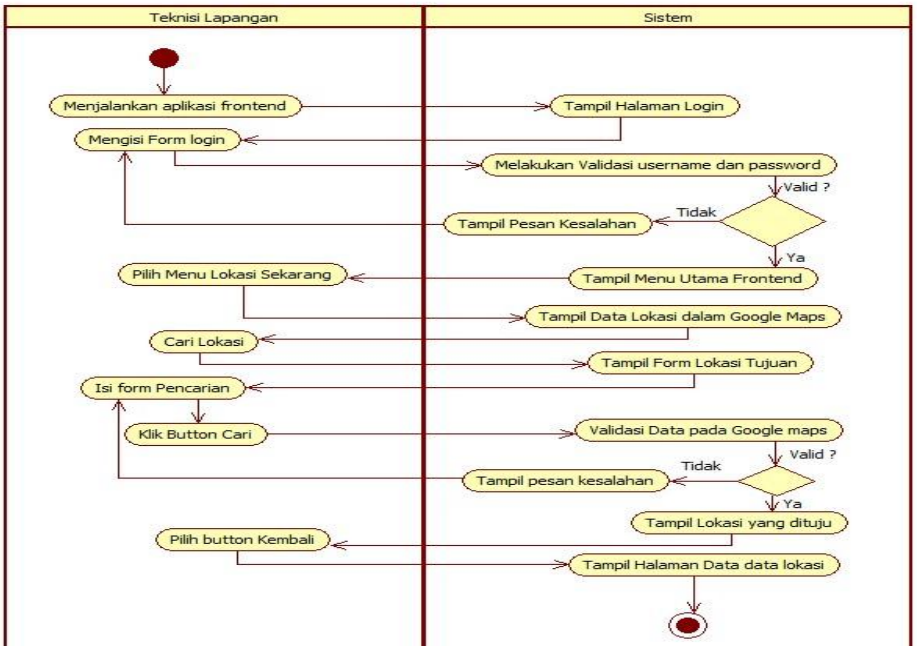
o. *Activity Diagram* Login *Frontend*



Gambar 0.20 *Activity Diagram* Mengelola Login *Frontend*

Pada Gambar 3.20 Menjelaskan tentang aktifitas yang terjadi saat seorang *user* yang akan mengakses aplikasi *Frontend* harus teridentikasi terlebih dahulu agar dapat melakukan login dan masuk kedalam halaman utama aplikasi *Frontend.*

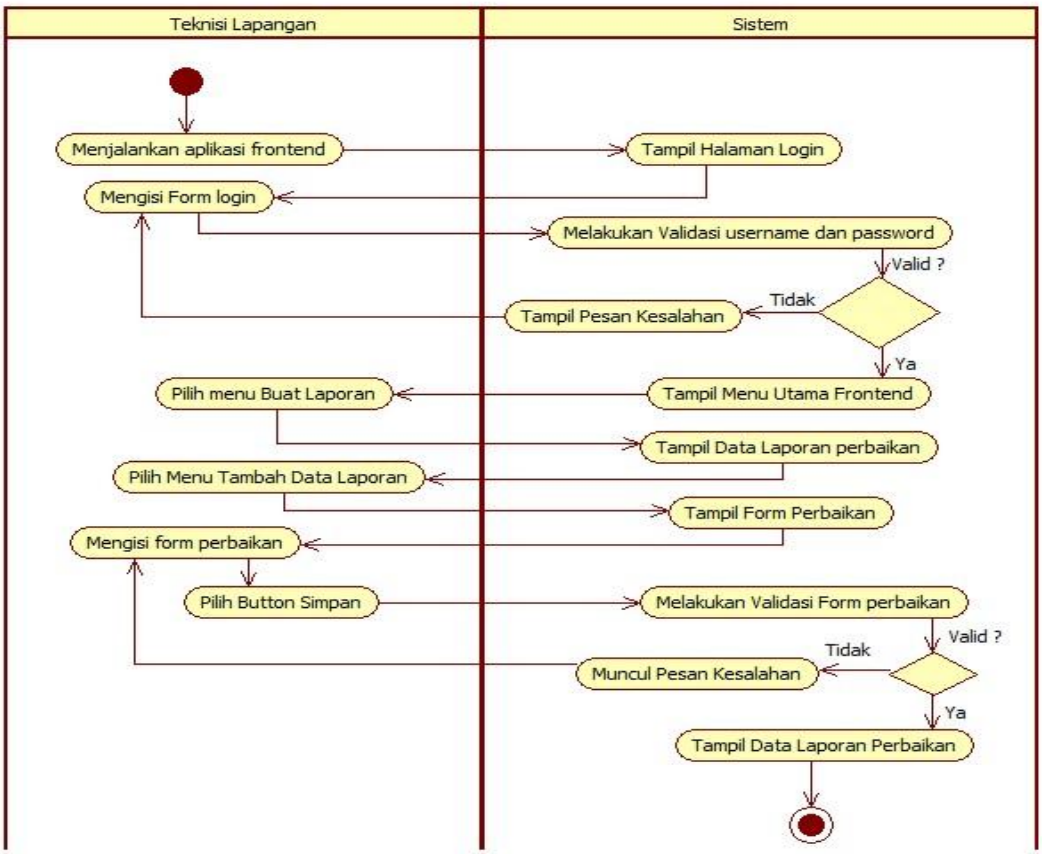
p. *Activity Diagram* Mengelola Data Lokasi



Gambar 0.21 *Activity Diagram* Mengelola Data Lokasi

Pada Gambar 3.21 Menjelaskan tentang aktifitas yang terjadi saat seorang teknisi lapangan yang akan melakukan perbaikan untuk mencari letak dan alamat toko serta memberikan informasi mengenai lokasi toko yang mengalami kerusakan serta informasi lokasi teknisi lapangan berada.

q. *Activity Diagram* Mengelola Data *Service*



Gambar 0.22 *Activity Diagram* Mengelola Data *Service*

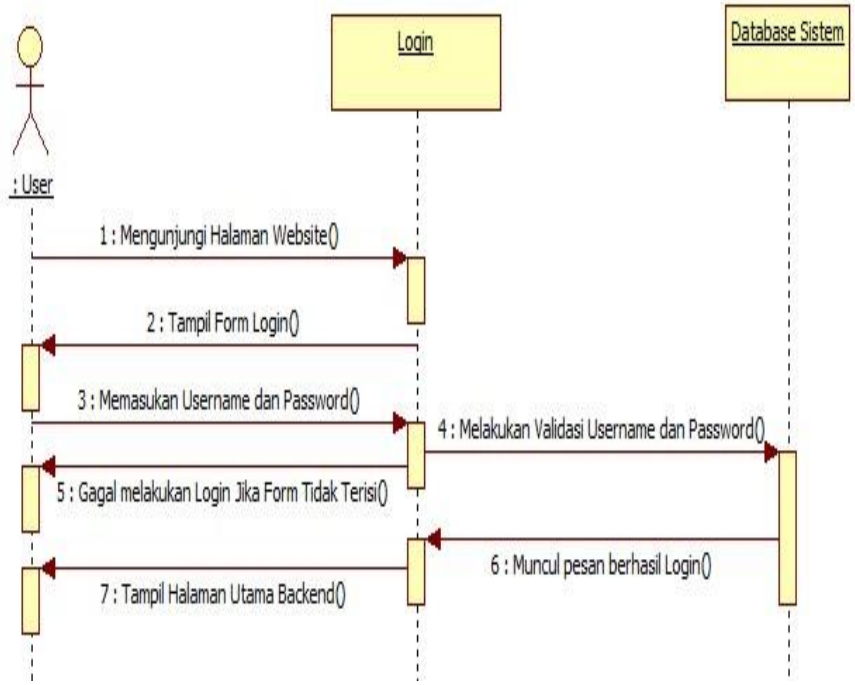
Pada Gambar 3.22 Menjelaskan tentang aktifitas yang terjadi saat seorang teknisi lapangan dalam pembuatan laporan perbaikan kerusakan apabila laporan kerusakan sudah ditindak lanjuti dan unit *Showcase* berfungsi dengan normal.

3.2.7.6 Sequence diagram

Diagram sequence menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan

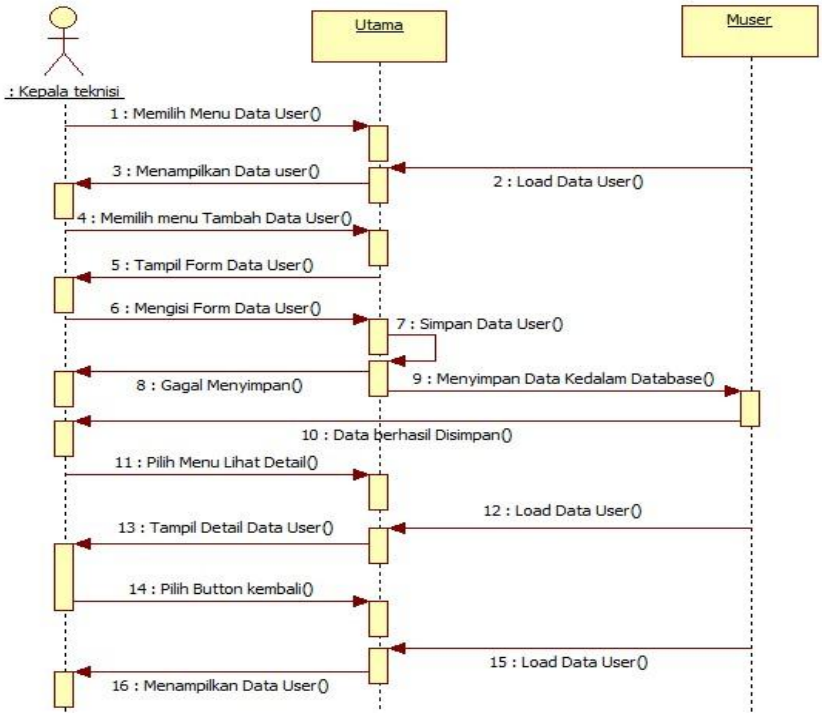
diterima antarobjek. Oleh karena itu untuk menggambar diagram sequence maka harus diketahui objek-objek yang terlibat dalam sebuah *Use Case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Berikut ini adalah *Sequence diagram* yang ada pada sistem *reporting Service* berbasis android yang akan di bangun. Seperti dapat dilihat dari gambar 3.23 – gambar 3.33.

a. *Sequence diagram* Login *Backend*



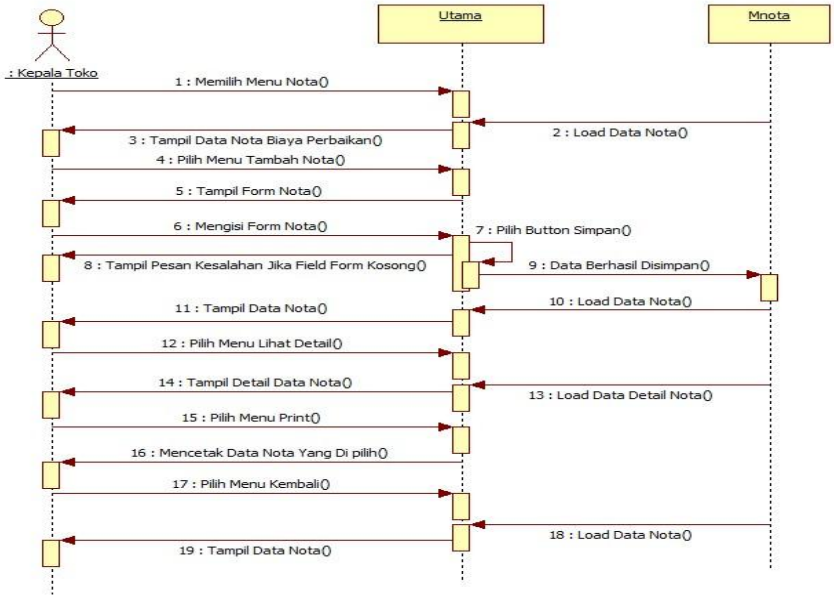
Gambar 0.23 *Sequence diagram* Login *Backend*

b. *Sequence diagram* mengelola Data *User*



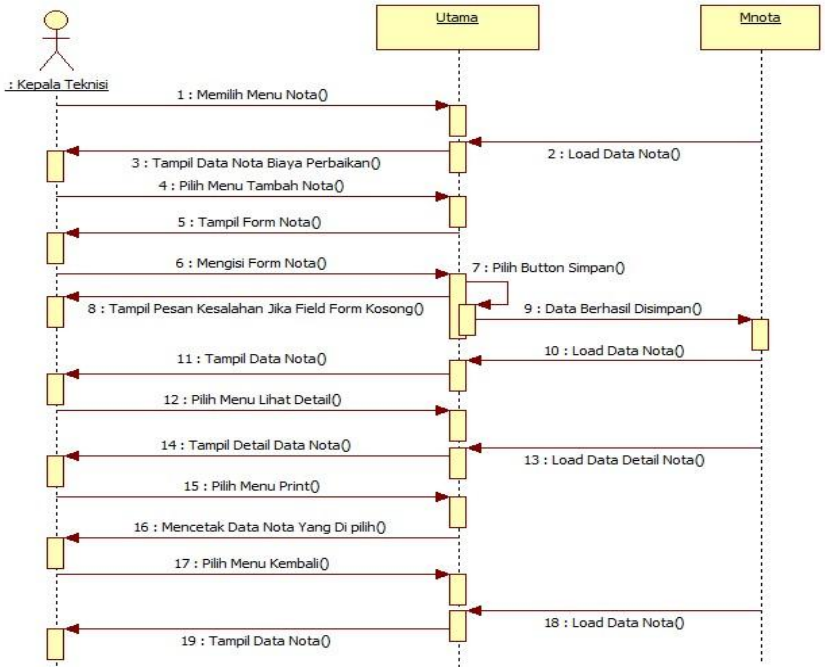
Gambar 0.24 *Sequence diagram* Mengelola Data *User*

c. *Sequence diagram* Mengelola Data Laporan



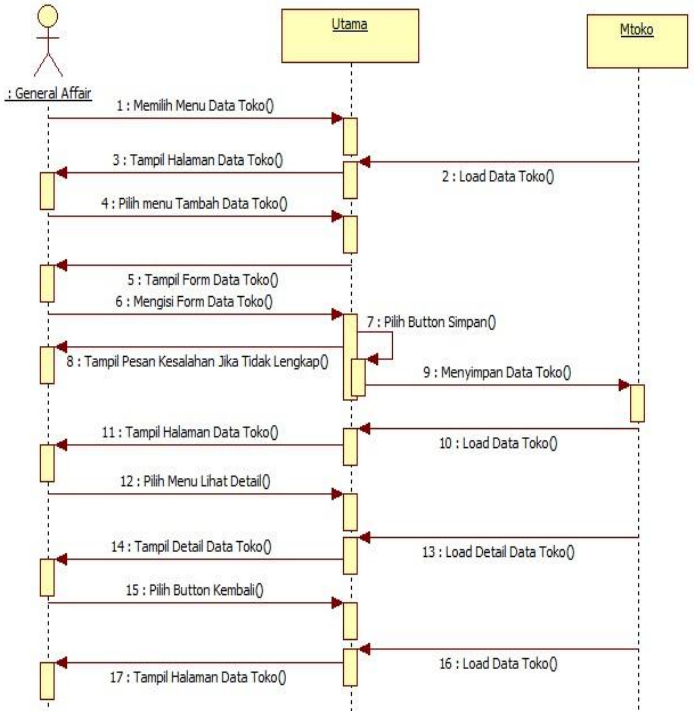
Gambar 0.25 *Sequence diagram* Mengelola Data Laporan

d. *Sequence diagram* Mengelola Data Nota



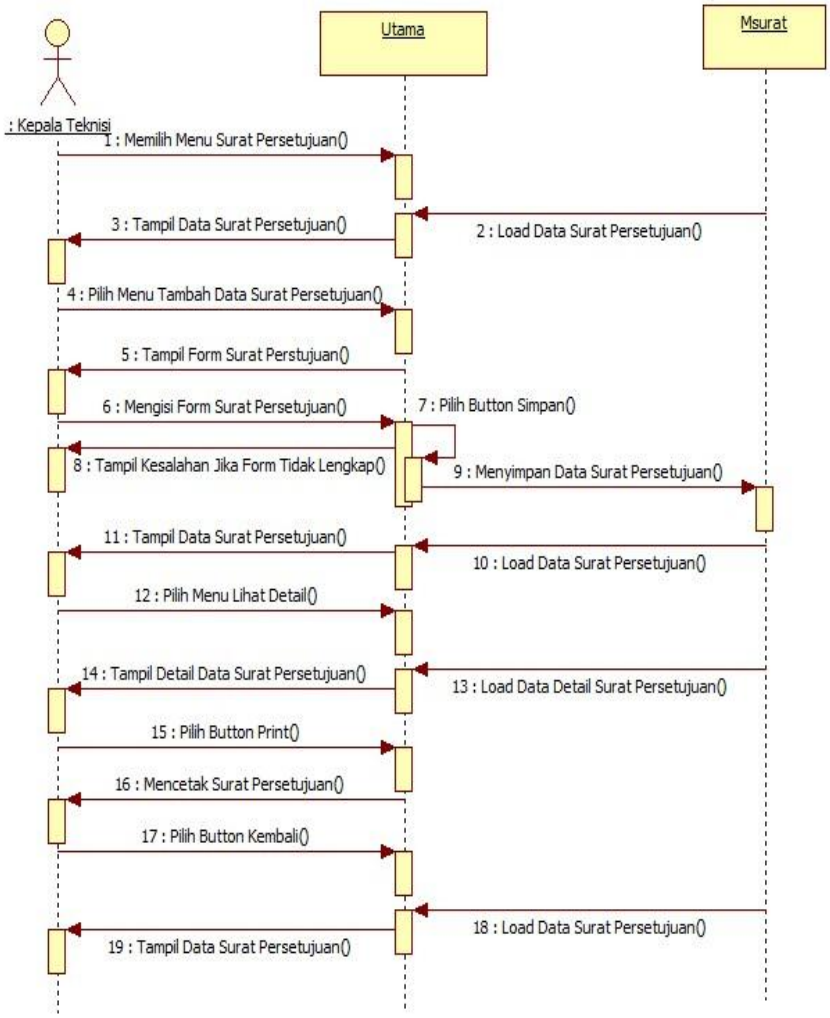
Gambar 0.26 *Sequence diagram* Mengelola Data Nota

e. *Sequence diagram* Mengelola Data Toko



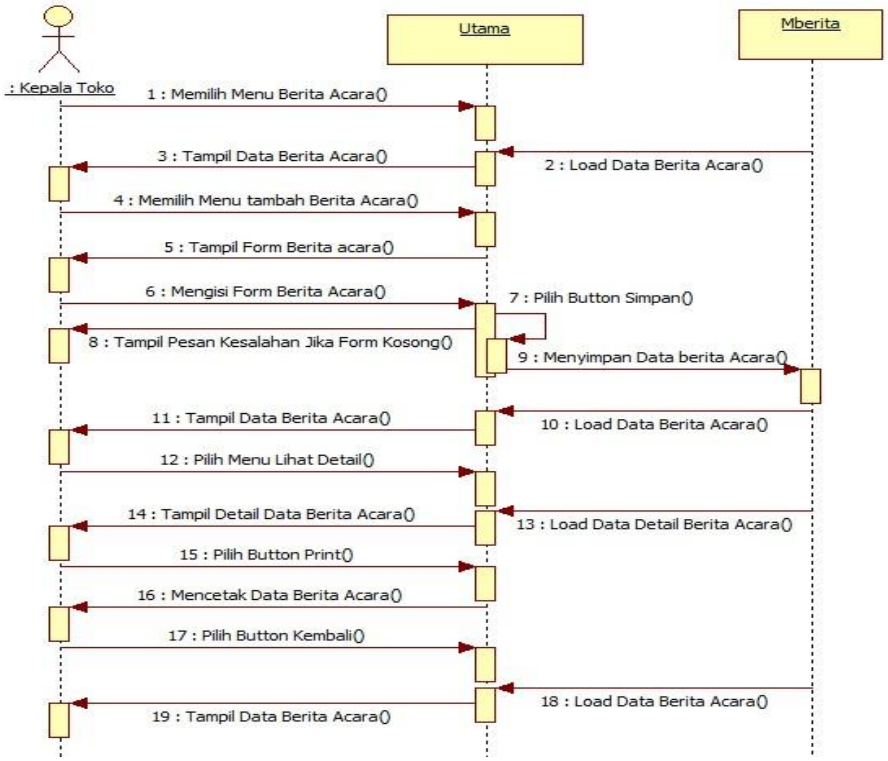
Gambar 0.27 Gambar *Sequence diagram* Mengelola Data Toko

f. *Sequence diagram* Mengelola Data Surat Persetujuan



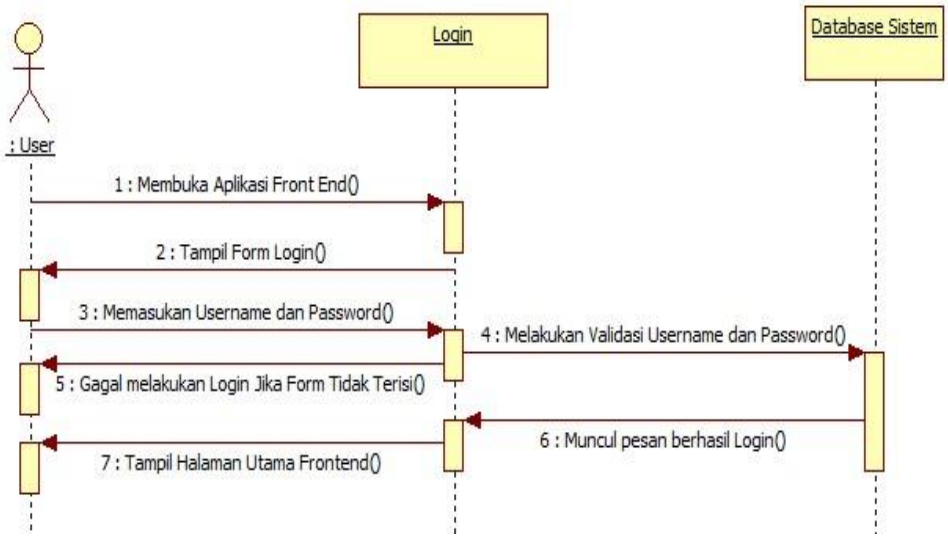
Gambar 0.28 *Sequence diagram* Mengelola Surat Persetujuan

g. *Sequence diagram* Mengelola Berita Acara



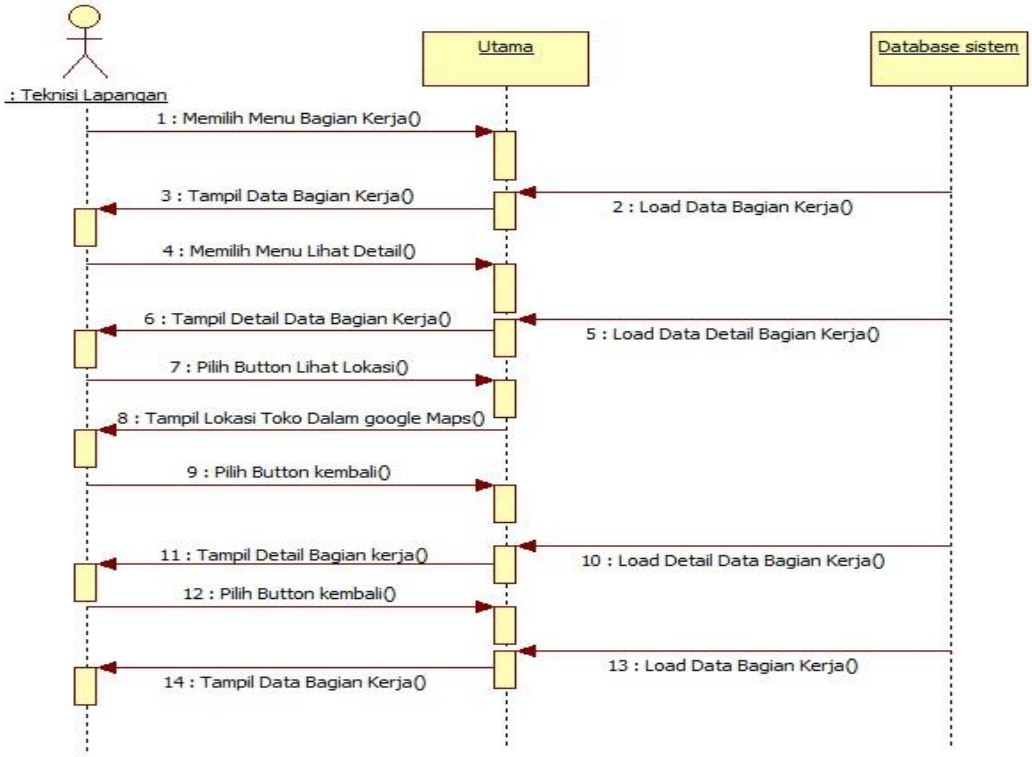
Gambar 0.29 *Sequence diagram* Mengelola Berita Acara

h. *Sequence diagram* Login *Frontend*



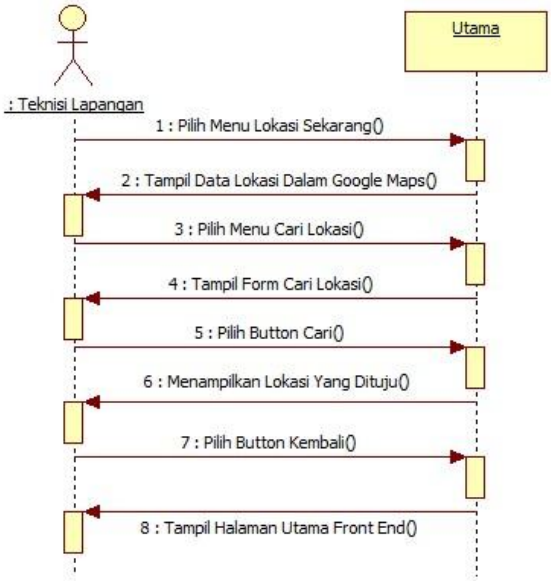
Gambar 0.30 *Sequence diagram* Login *Frontend*

i. *Sequence diagram* Mengelola Bagian Kerja



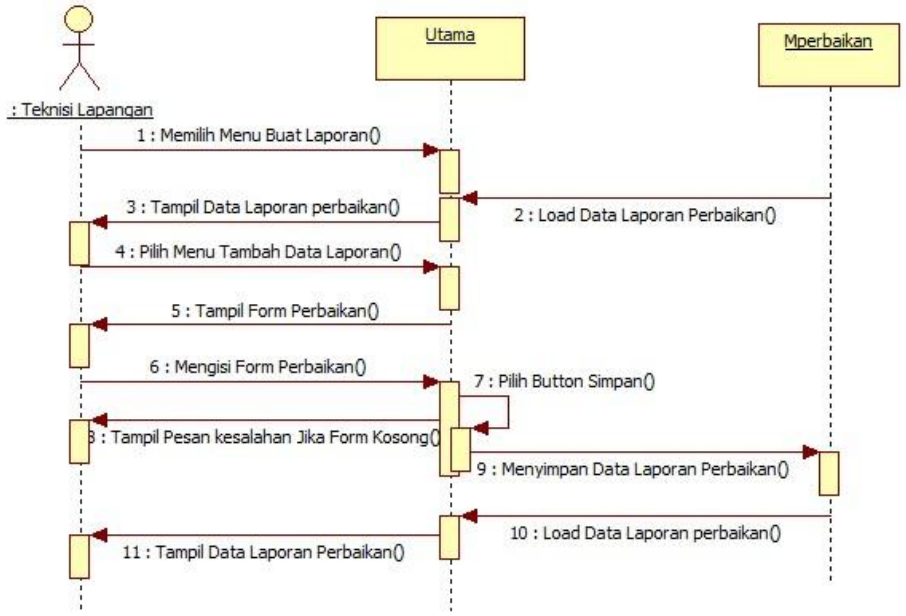
Gambar 0.31 *Sequence diagram* Mengelola Bagian Kerja

j. *Sequence diagram* mengelola Data Lokasi



Gambar 0.32 *Sequence diagram* Mengelola Data Lokasi

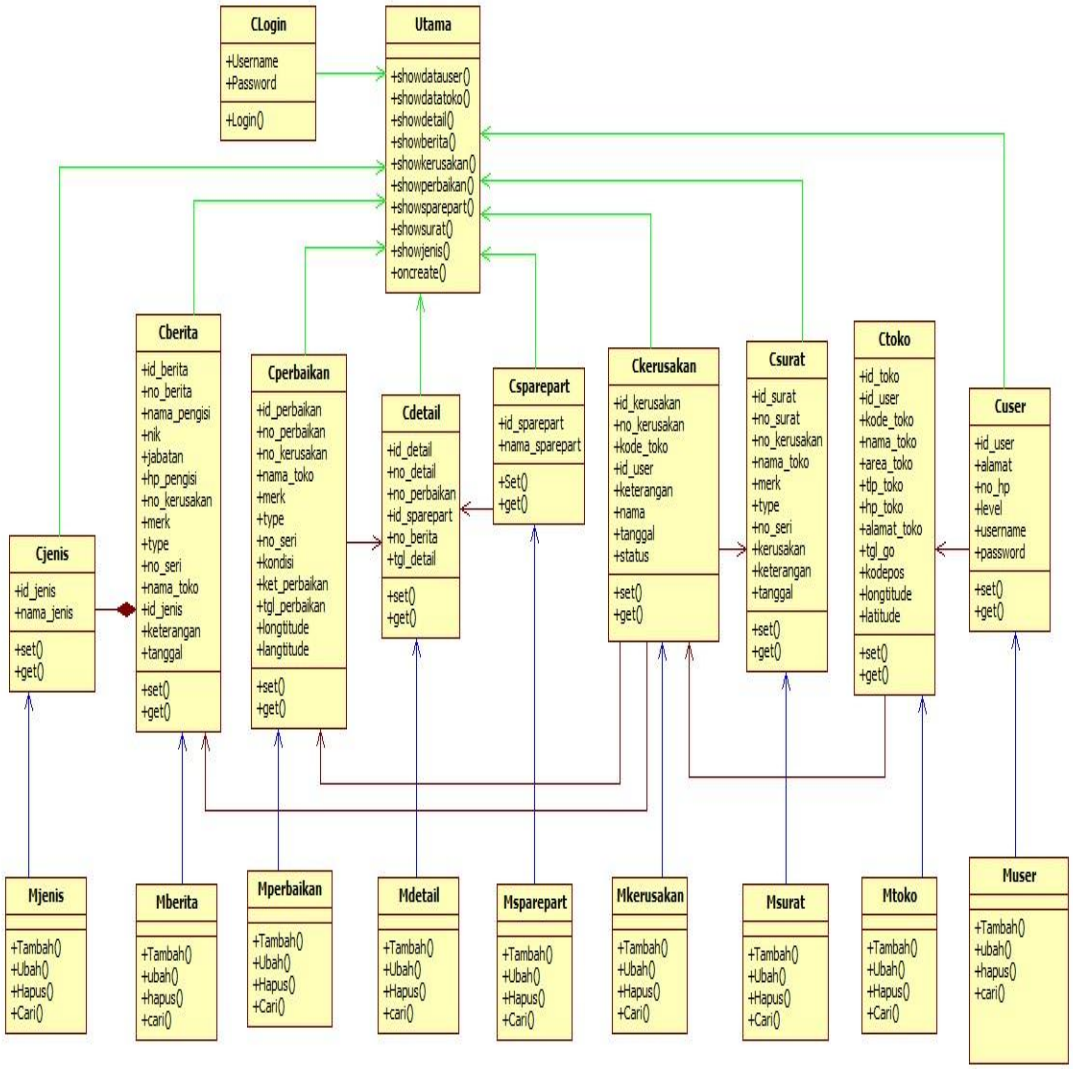
k. *Sequence diagram* Mengelola Data *Service*



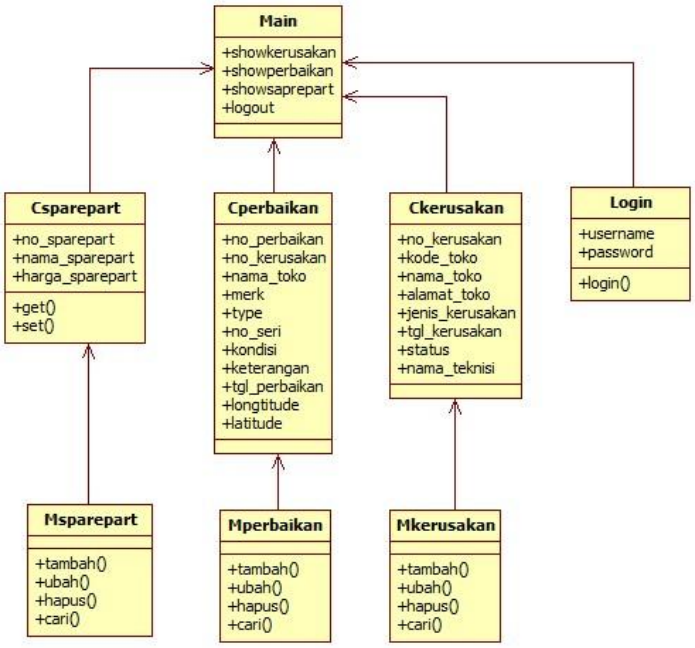
Gambar 0.33 *Sequence diagram* Mengelola Data *Service*

3.2.7.7 Class Diagram

Class Diagram menggambarkan struktur dan hubungan antar objek- objek yang ada pada sistem. Struktur itu meliputi atribut-atribut dan metode- metode yang ada pada masing-masing *class*. Hubungannya dari masing- masing *class* yang ada tersebut digambarkan pada gambar 3.34 *Backend* dan gambar 3.35 *Frontend*



Gambar 0.34 Class Diagram *Backend*



Gambar 0.35 Class diagram *Frontend*

Tabel 0.19 Deskripsi Class Diagram *Backend*

Class	Jenis Class	Deskripsi
Clogin	Controller	Merupakan class yang berisi method method untuk mengelola sistem login
C*user*	Controller	Merupakan class yang berisi method method untuk mengelola data *user*
Ctoko	Controller	Merupakan class yang berisi method method untuk mengelola data toko
Ckerusakan	Controller	Merupakan class yang berisi method method untuk mengelola data kerusakan
Cperbaikan	Controller	Merupakan class yang berisi method method untuk mengelola data perbaikan
Csurat	Controller	Merupakan class yang

		berisi method method untuk mengelola data surat persetujuan
Cberita	Controller	Merupakan class yang berisi method method untuk mengelola data berita acara
Csparepart	Controller	Merupakan class yang berisi method method untuk mengelola data sparepart
Cdetail	Controller	Merupakan class yang berisi method method untuk mengelola data detail perbaikan
M*user*	Model	Merupakan Clas yang digunakan untuk mengelola *database user*
Mtoko	Model	Merupakan Clas yang digunakan untuk mengelola *database* Toko
Mkerusakan	Model	Merupakan Clas yang digunakan untuk mengelola *database* Kerusakan
Mperbaikan	Model	Merupakan Clas yang digunakan untuk mengelola *database* Perbaikan
Msurat	Model	Merupakan Clas yang digunakan untuk mengelola *database* Surat Persetujuan
Mberita	Model	Merupakan Clas yang digunakan untuk mengelola *database* Berita Acara
Msparepart	Model	Merupakan Clas yang digunakan untuk mengelola *database* Sparepart
Mdetail	Model	Merupakan Clas yang digunakan untuk mengelola *database* detail
Utama	View	Merupakan Class yang

		digunakan untuk mengontrol class interface

Tabel 0.20 Deskripsi Class Diagram *Frontend*

Class	Jenis Class	Deskripsi
Clogin	Controller	Merupakan class yang berisi method method untuk mengelola data lokasi
Ckerusakan	Controller	Merupakan class yang berisi method method untuk mengelola data bagian kerja
Cperbaikan	Controller	Merupakan class yang berisi method method untuk mengelola data laporan perbaikan
Csparepart	Controller	Merupakan class yang berisi method method untuk mengelola data sparepart
Mlogin	Model	Merupakan Class yang digunakan untuk mengelola *database* lokasi
Mkerusakan	Model	Merupakan Class yang digunakan untuk mengelola *database* bagian kerja
Mperbaikan	Model	Merupakan Class yang digunakan untuk mengelola *database* laporan perbaikan
Msparepart	Model	Merupakan Class yang digunakan untuk mengelola *database* sparepart
Main	View	Merupakan Class yang digunakan untuk mengontrol interface

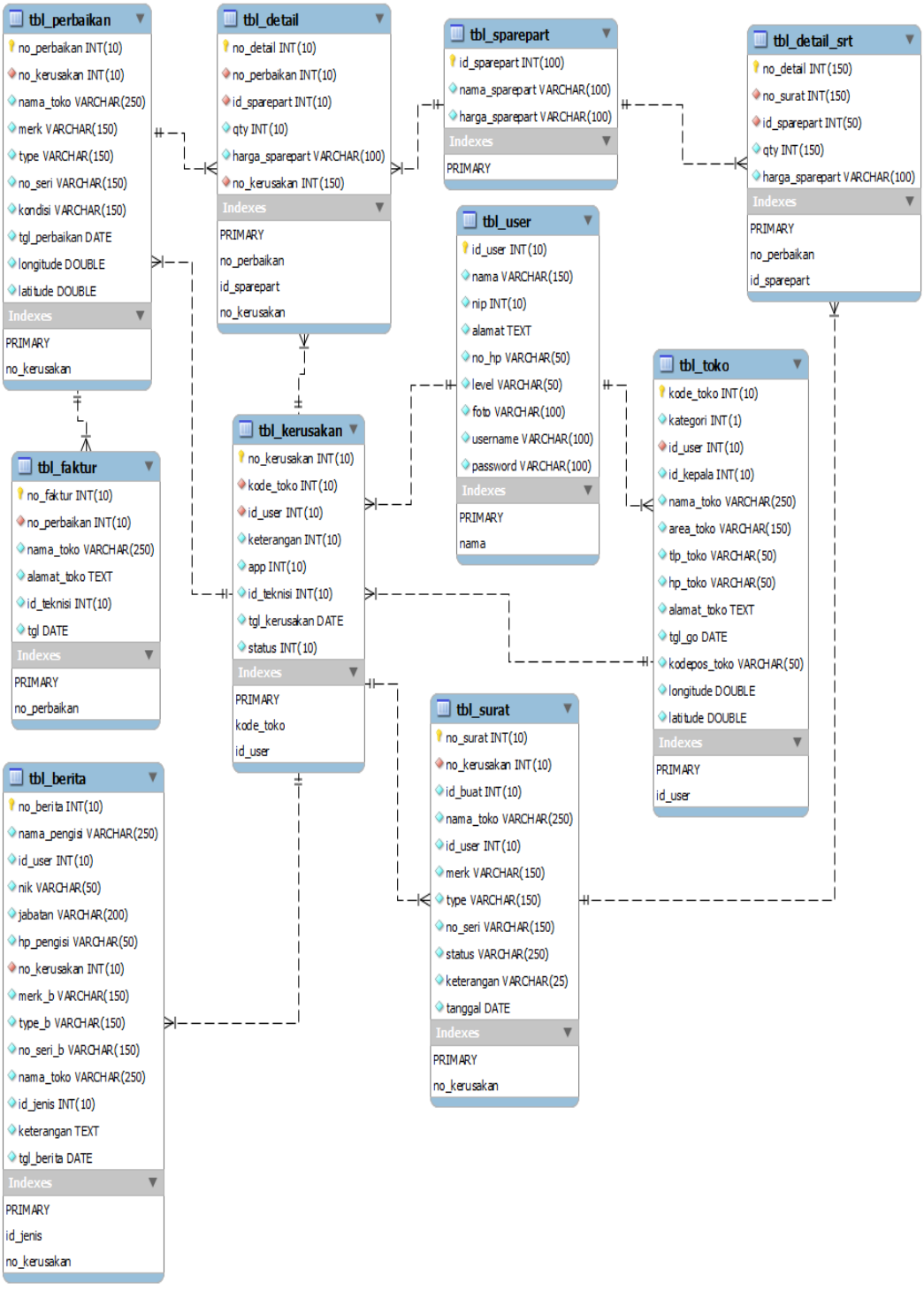
3.2.7.8 Perancangan Sistem

Perancangan sistem bertujuan untuk menspesifikasikan aspek – aspek teknik yang menjadi solusi dalam perencanaan. Pada tahap ini perancangan akan

di definisikan secara detail untuk mengatasi masalah – masalah yang lebih teknis, berkaitan dengan kegiatan implementasi seperti perancangan databse, dan perancangan antarmuka.

3.2.7.8.1 Skema Relasi

Skema Relasi merupakan gabungan antar *file* yang mempunyai kunci utama yang sama, sehingga *file – file* tersebut menjadi satu kesatuan yang dihubungkan oleh *field* kunci *(Primary Key)* . Pada proses ini elemen – elemen data dikelompokan menjadi satu *file database* beserta entitas dan hubungannya. Skemarelasi yang dibangun dapat dilihat pada gambar 3.36



Gambar 0.36 Skema Relasi

3.2.7.8.2 Struktur Tabel

Struktur table digunakan untuk menjelaskan field field yang dibutuhkan dalam *database* untuk pembangunan aplikasi *reporting Showcase* pada CV Gunung Agung.

Tabel 0.21 Struktur Tabel *User*

Nama Field	Tipe	Panjang	Keterangan
id user	*Integer*	10	*Primary key*
nama	*Varchar*	150	
nip	*Int*	10	
alamat	*Text*		
no_hp	*Varchar*	50	
level	*Varchar*	50	
Foto	*Varchar*	150	
Username	*Varchar*	100	
Password	*Varchar*	100	

Tabel 0.22 Struktur Tabel Toko

Nama Field	Tipe	Panjang	Keterangan
kode_toko	*Integer*	10	*Primary key*
kategori	*Interger*	1	
id_user	*Integer*	10	*Foreign key reference tbl_user* (*id_user*)
id_kepala	*Integer*	150	
nama_toko	*Varchar*	250	

area_toko	*Varchar*	150	
tlp_toko	*Varchar*	50	
hp_toko	*Varchar*	50	
alamat_toko	*Text*		
tgl_go	*Date*		
kodepos_toko	*Varchar*	50	
longtitude	*Double*		
latitude	*Double*		

Tabel 0.23 struktur tabel kerusakan

Nama Field	**Tipe**	**Panjang**	**Keterangan**
No_kerusakan	*Integer*	10	*Primary key*
Kode_toko	*Integer*	150	*Foreign key reference tbl_toko* (*kode_toko*)
Id_*user*	*Integer*	150	*Foreign key reference tbl_user*(*id_user*)
App	*Integer*	10	
Id_teknisi	*Varchar*	150	
Tgl_kerusakan	*Date*		
status	*Varchar*	100	

Tabel 0.24 Stuktur Tabel Perbaikan

Nama Field	Tipe	Panjang	Keterangan
No_perbaikan	*Integer*	10	*Primary key*
No_kerusakan	*Integer*	10	*Foreign key reference tbl_kerusakan* (*no_kerusakan*)
Nama_toko	*Varchar*	250	
Merk	*Varchar*	150	
Type	*Varchar*	150	
No_seri	*Varchar*	150	
Kondisi	*Varchar*	150	
Tgl_perbaikan	*Date*		
longtitude	*Double*		
latitude	*Double*		

Tabel 0.25 Struktur Tabel berita

Nama Field	Tipe	Panjang	Keterangan
No_berita	*Integer*	10	*Primary key*
Nama_pengisi	*Varchar*	250	
Id_*user*	*Varchar*	10	
Nik	*Varchar*	50	
Jabatan	*Varchar*	200	

hp_pengisi	*Varchar*	50	
No_kerusakan	*Integer*	10	*Foreign key reference tbl_kerusakan (no_kerusakan)*
Merk_b	*Varchar*	150	
Type_b	*Varchar*	150	
No_seri_b	*Varchar*	150	
Nama_toko	*Varchar*	250	
Id_jenis	*Integer*	50	*Foreign key reference tbl_jenis (id_jenis)*
Keterangan	*Text*		
Tgl_berita	*Date*		

Tabel 0.26 Struktur Tabel Surat

Nama Field	Tipe	Panjang	Keterangan
no_surat	*Integer*	10	*Primary key*
no_kerusakan	*Integer*	10	*Foreign key reference tbl_kerusakan (no_kerusakan)*
Id_buat	*Integer*	10	
nama_toko	*Varchar*	250	
Id_*user*	*Varchar*	10	
merk	*Varchar*	150	

type	*Varchar*	150	
no_seri	*Integer*	150	
status	*Varchar*	250	
keterangan	*Text*		
tanggal	*Date*		

Tabel 0.27 Struktur Tabel detail srt

Nama Field	Tipe	Panjang	Keterangan
no_detail	*Integer*	10	*Primary key*
no_surat	*Integer*	10	*Foreign key reference tbl_surat(no_surat)*
id_sparepart	*Varchar*	250	*Foreign key reference tbl_sparepart(id_sparepart)*
Qty	*Varchar*	150	
harga_sparepart	*Varchar*	150	

Tabel 0.28 Struktur Tabel Sparepart

Nama Field	Tipe	Panjang	Keterangan
id_sparepart	*Integer*	10	*Primary key*
nama_sparepart	*Varchar*	100	
harga_sparepart	*Varchar*	100	

Tabel 0.29 Struktur Tabel detail

Nama Field	Tipe	Panjang	Keterangan
no_detail	*Integer*	10	*Primary key*
no_perbaikan	*Integer*	10	*Foreign key reference tbl_surat(no_surat)*
id_sparepart	*Varchar*	250	*Foreign key reference tbl_sparepart(id_sparepart)*
Qty	*Integer*	10	
harga_sparepart	*Varchar*	150	
no_kerusakan	*Integer*	10	*Foreign key reference tbl_kerusakan(id_sparepart)*

Tabel 0.30 Struktur Tabel Faktur

Nama Field	Tipe	Panjang	Keterangan
no_faktur	*Integer*	10	*Primary key*
no_perbaikan	*Integer*	10	*Foreign key reference tbl_surat(no_surat)*
nama_toko	*Varchar*	250	*Foreign key reference tbl_sparepart(id_sparepart)*
alamat_toko	*Text*		
id_teknisi	*Varchar*	150	
tgl	*Date*		

3.2.7.9 Perancangan Kode

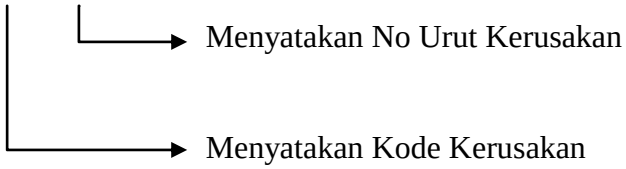
3.2.7.9.1 Kodevikasi

Kodifikasi atau sistem pengkodean digunakan sebagai alat bantu pengolahan data yang dapat mempermudah dalam mengidentifikasi objek dan dapat juga untuk mengelompokan data. Tujuan Pengkodean adalah untuk mempermudah proses pencarian data guna penyajian informasi yang dibutuhkan.

Adapun pengkodean dalam aplikasi *reporting Service* yang digunakan pada aplikasi *Backend* adalah sebagai berikut :

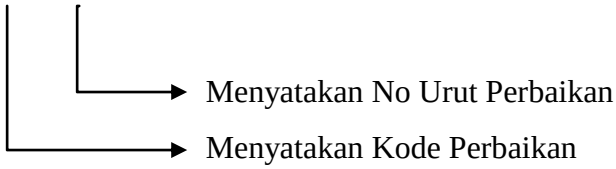
a. Kode Laporan Kerusakan

Format : KR – 1



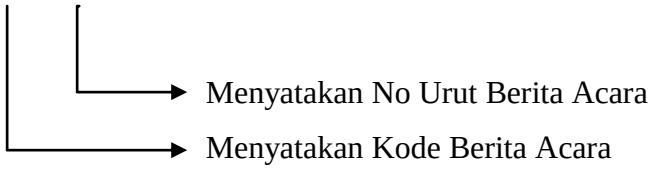
b. Kode Laporan Perbaikan

Format : PR –1



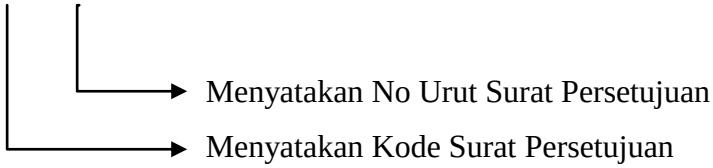
c. Kode Laporan Berita Acara

Format : BR –1



d. Kode Surat Persetujuan

Format : SRT –1



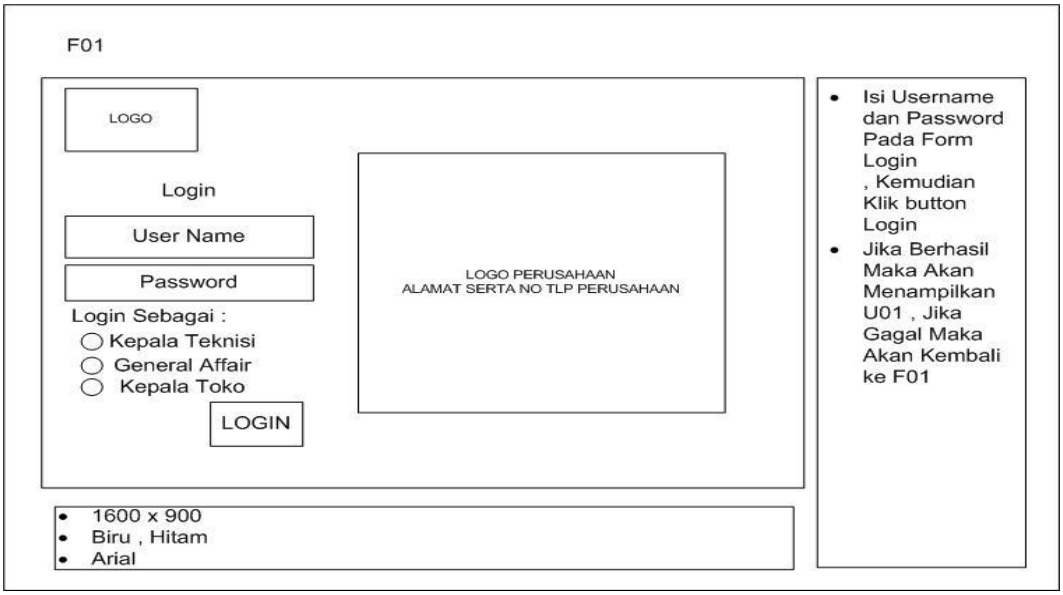
3.2.8 Perancangan Antar Muka

Pada perancangan antar muka atau *interface* ini akan menggambarkan rancangan tampilan apa saja yang ada dalam sistem ini dengan pertimbangan factor – factor permasalahan dan kebutuhan yang ada pada sistem seperti yang telah ditetapkan pada tahap analisis. Dalam tahap ini upaya yang dilakukan yaitu dengan cara mengkombinasikan penggunaan teknologi perangkat keras dan perangkat lunak yang tepat sehingga diperoleh hasil yang optimal dan mudah untuk diimplementasikan.

Adapun yang akan dibahas yaitu struktur menu yang terdiri dari dua bagian utama yaitu *interface bac kend ,* dan *interface Frontend.*

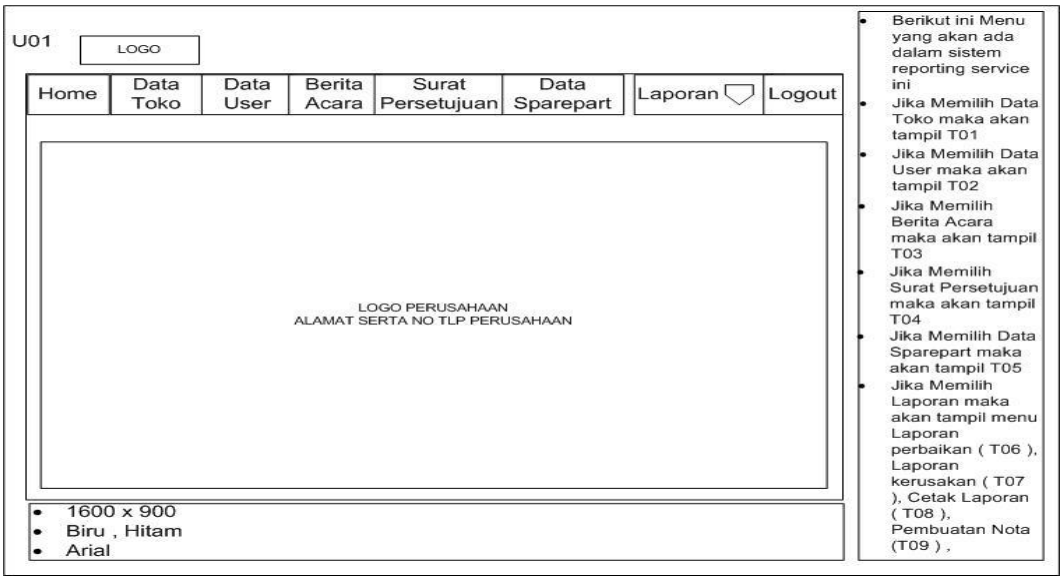
3.2.8.1 Perancangan antar muka *Backend*

1. Perancangan Form Login



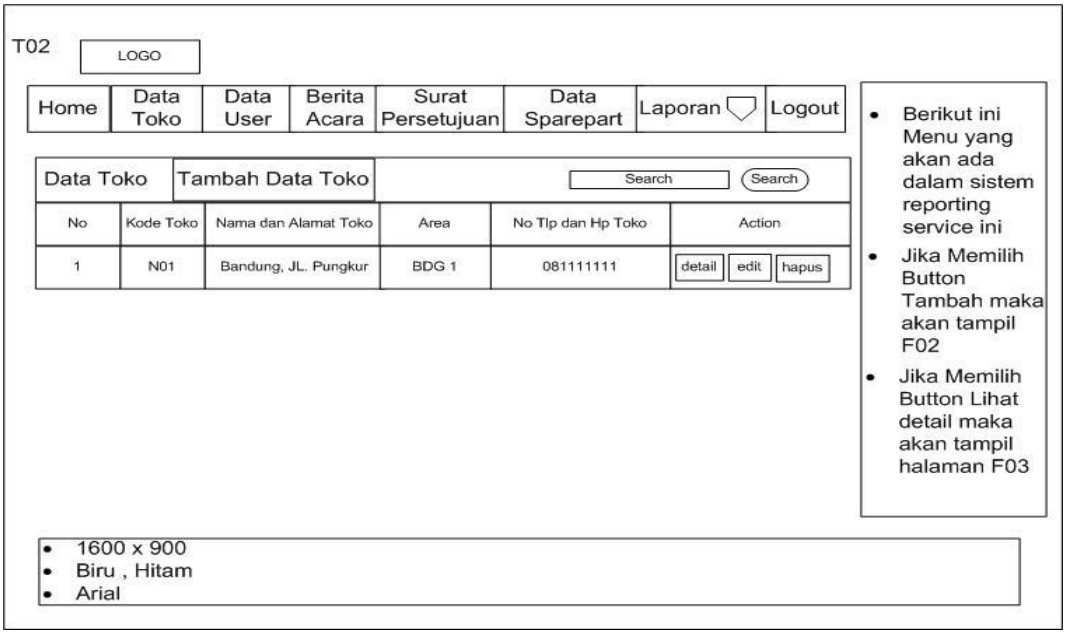
Gambar 0.37 Perancangan Tampilan Login *Backend*

2. Perancangan Tampilan Utama *Backend*



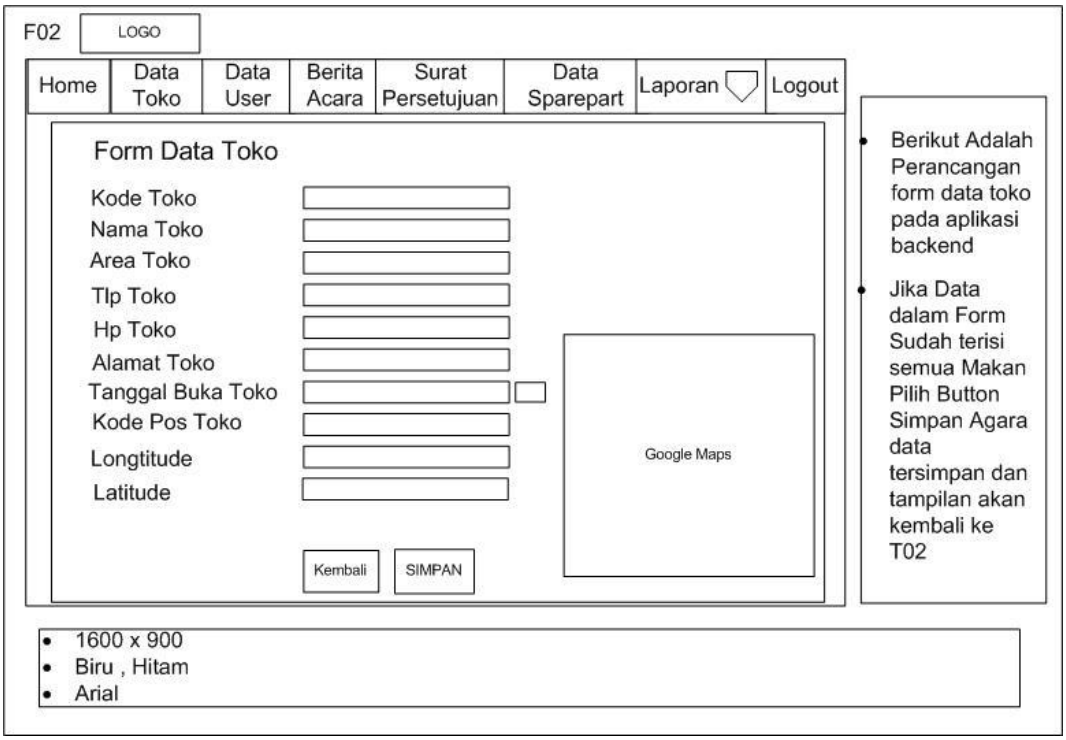
Gambar 0.38 Tampilan Utama Menu *Backend*

3. Perancangan Tampilan Data Toko



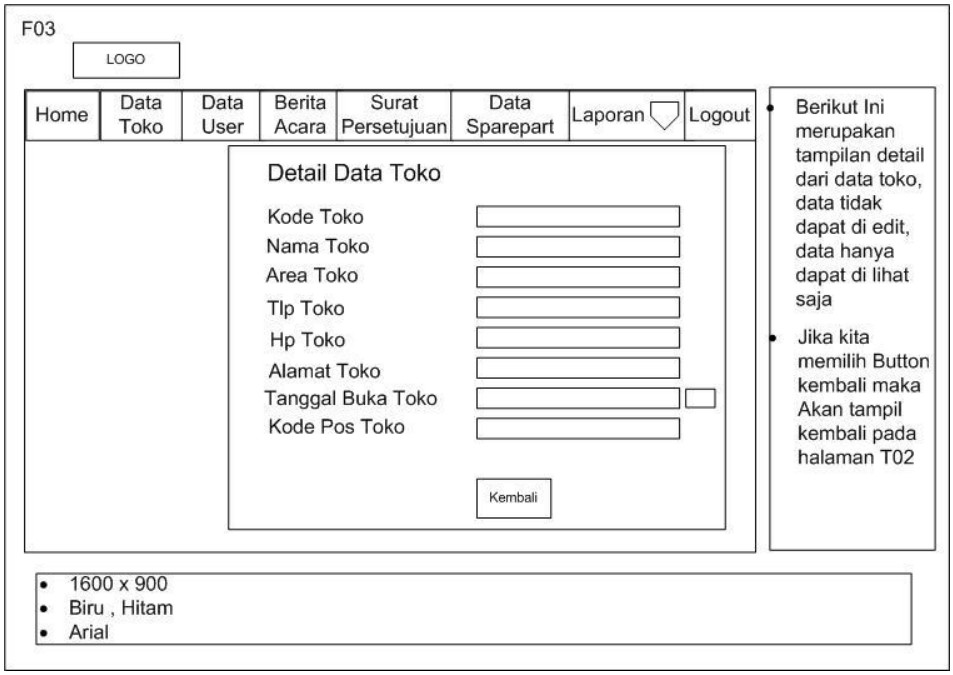
Gambar 0.39 Tampilan Data Toko

4. Perancangan Tampilan Form Tambah Data Toko



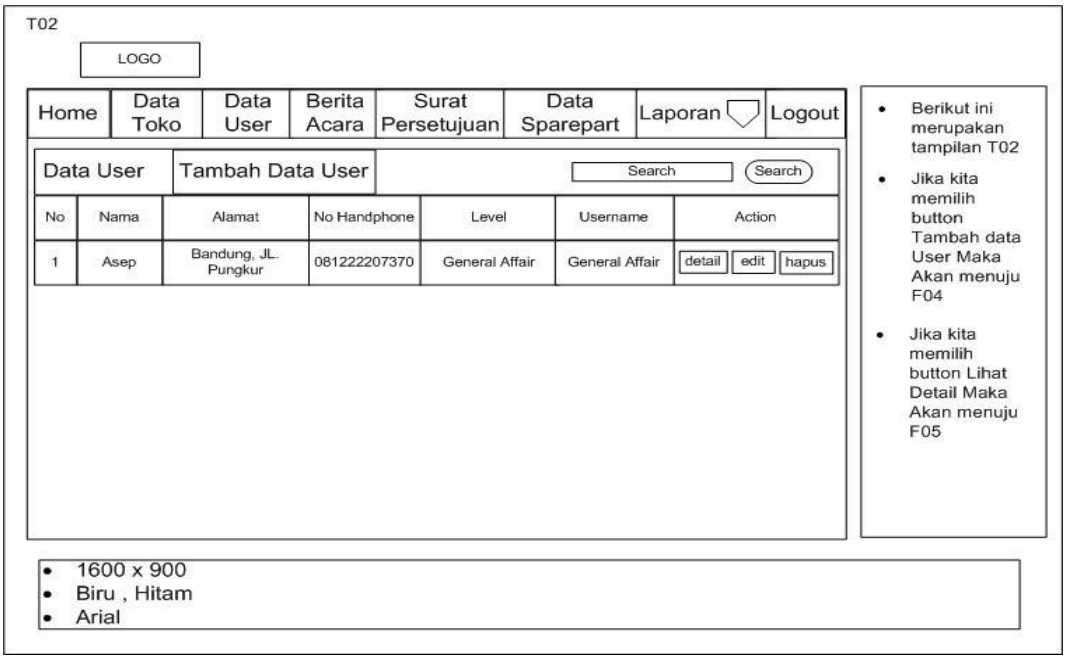
Gambar 0.40 Perancangan Form Data Toko

5. Perancangan Tampilan Detail Data Toko



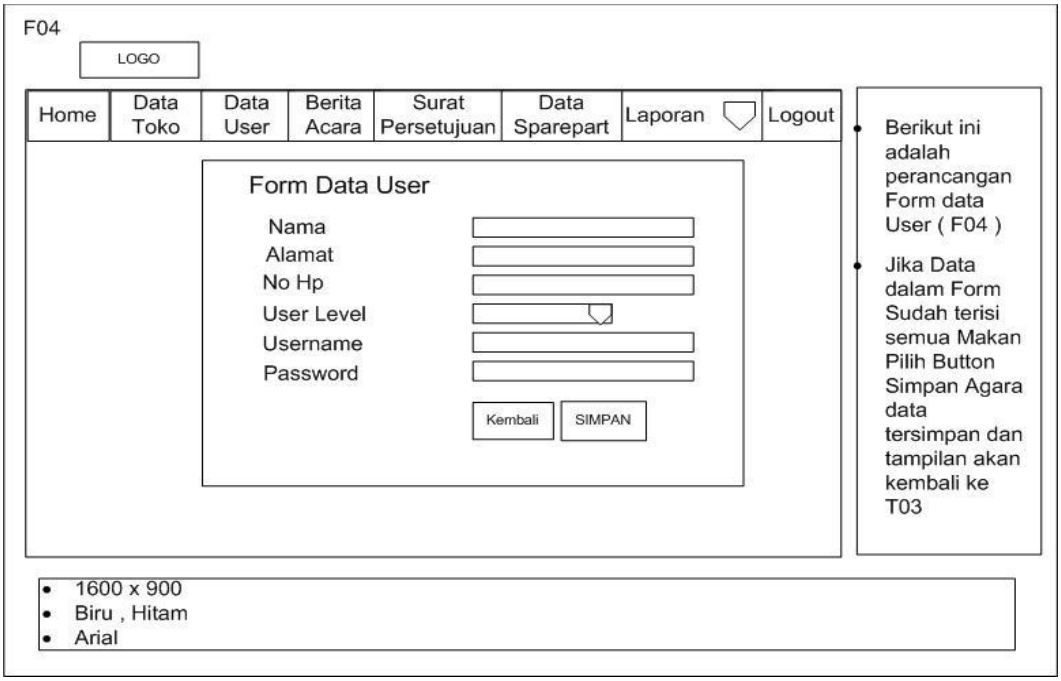
Gambar 0.41 Perancangan Detail Data Toko

6. Perancangan Tampilan Data *user*



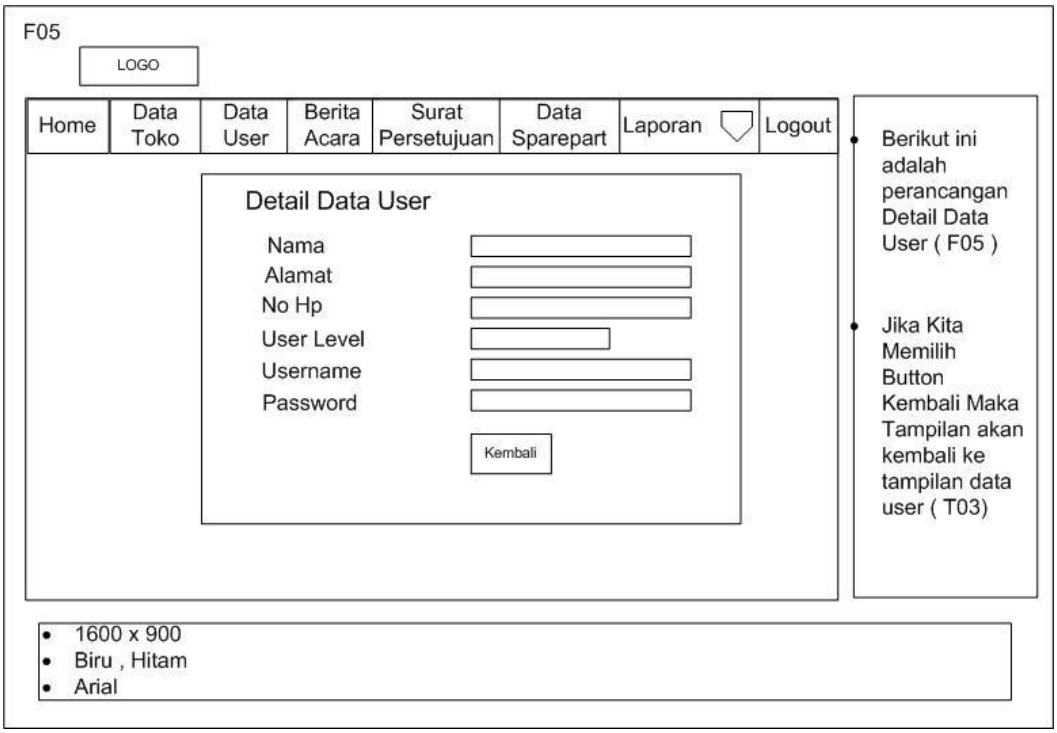
Gambar 0.42 Perancangan Tampilan Data *User*

7. Perancangan Tampilan Form Data *User*



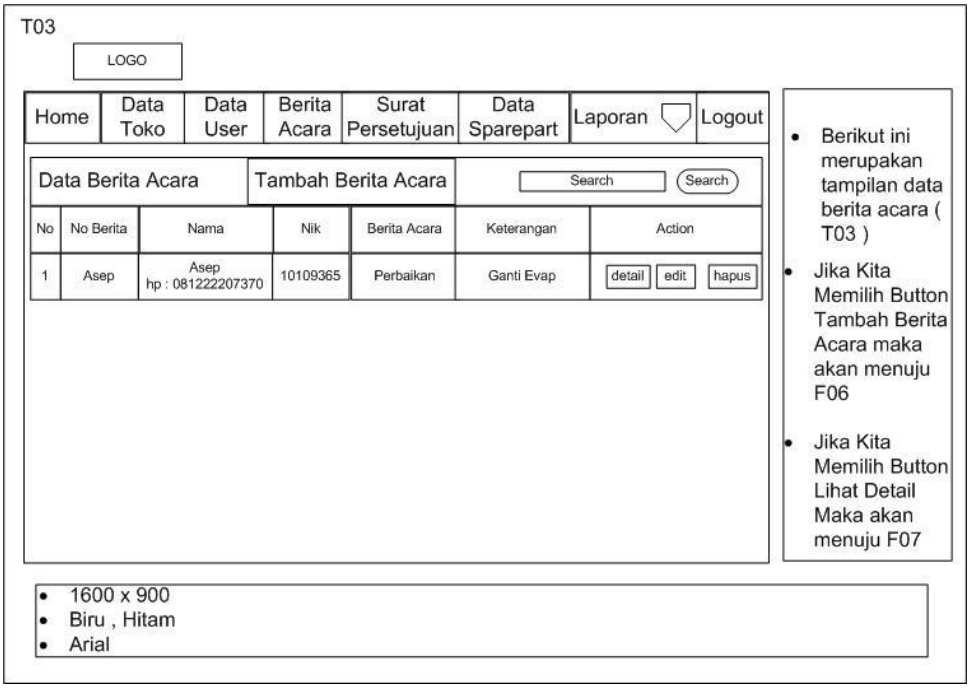
Gambar 0.43 Perancangan Form data *user*

8. Perancangan Detail Data *User*



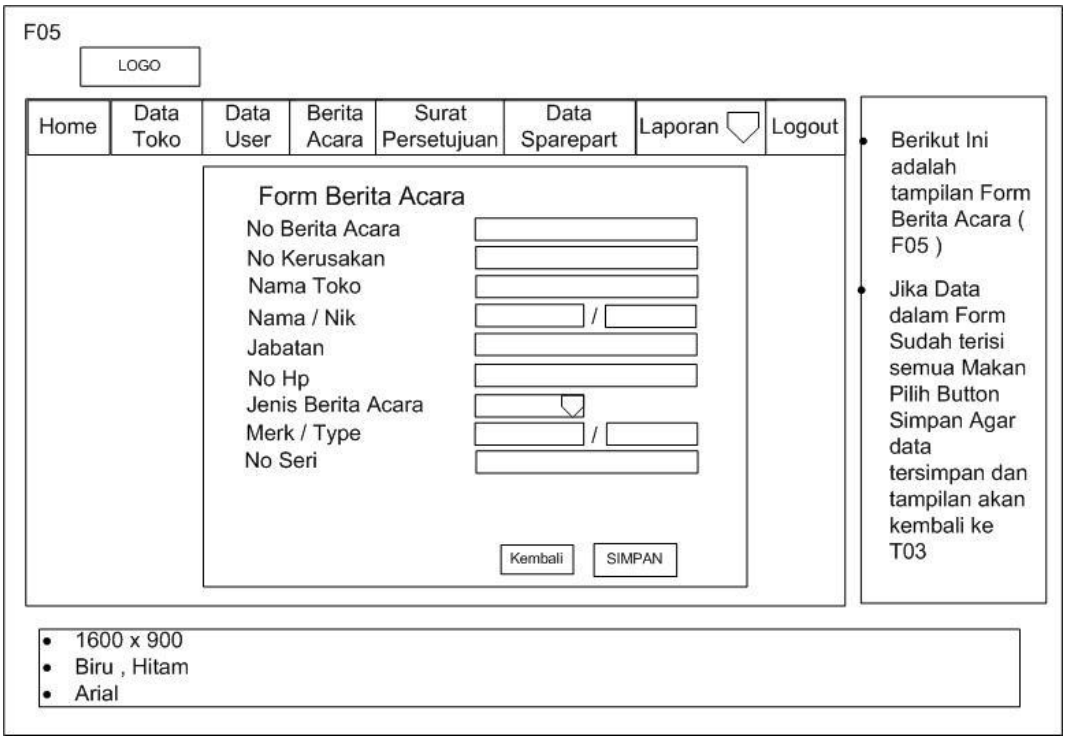
Gambar 0.44 Perancangan Detail *User*

9. Perancangan Data Berita Acara



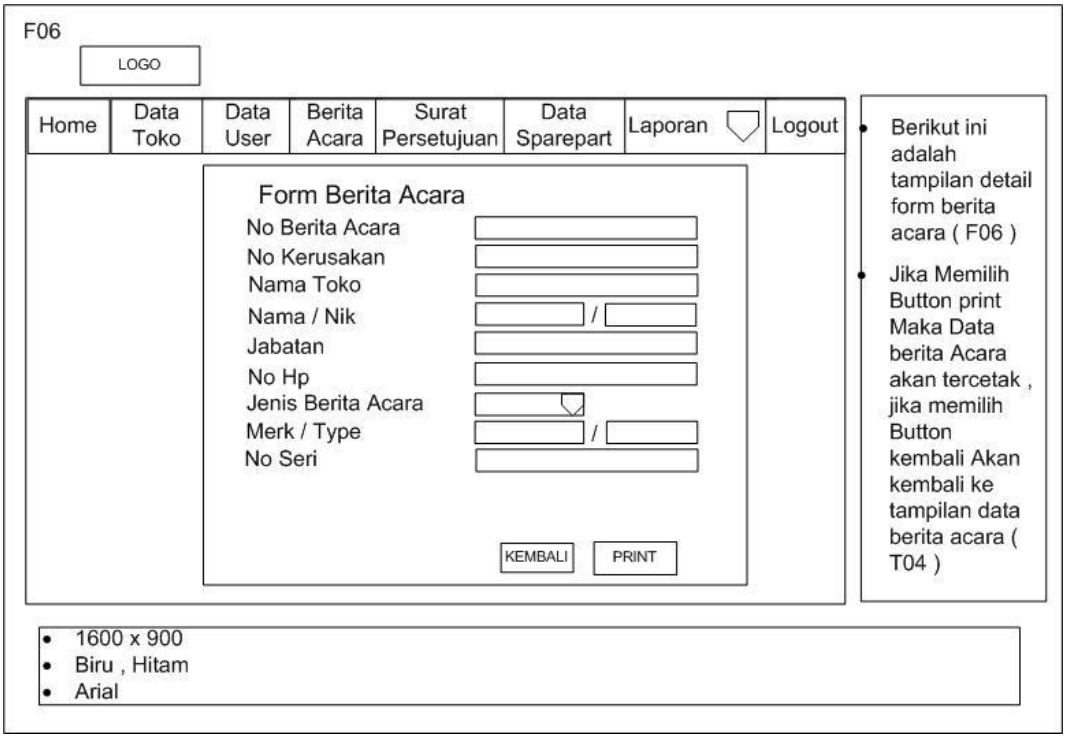
Gambar 0.45 Perancangan Data Berita Acara

10. Perangcangan Form Berita Acara



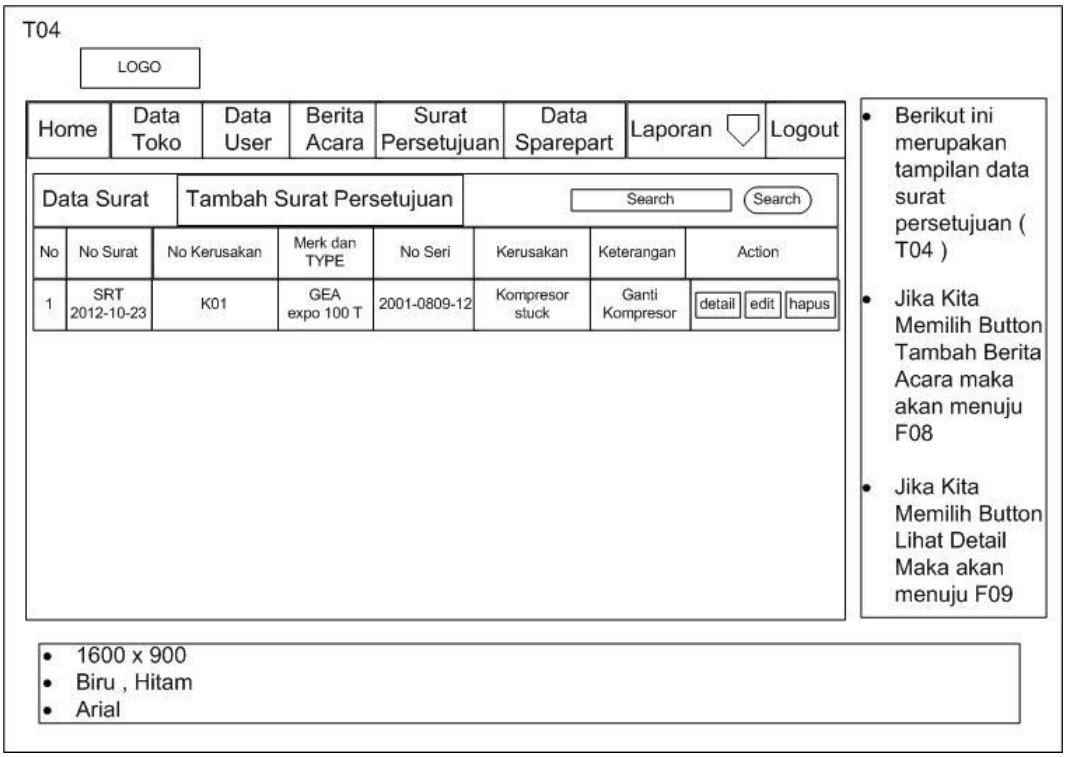
Gambar 0.46 Perancangan Form Berita Acara

11. Perancangan Detail Data Berita Acara



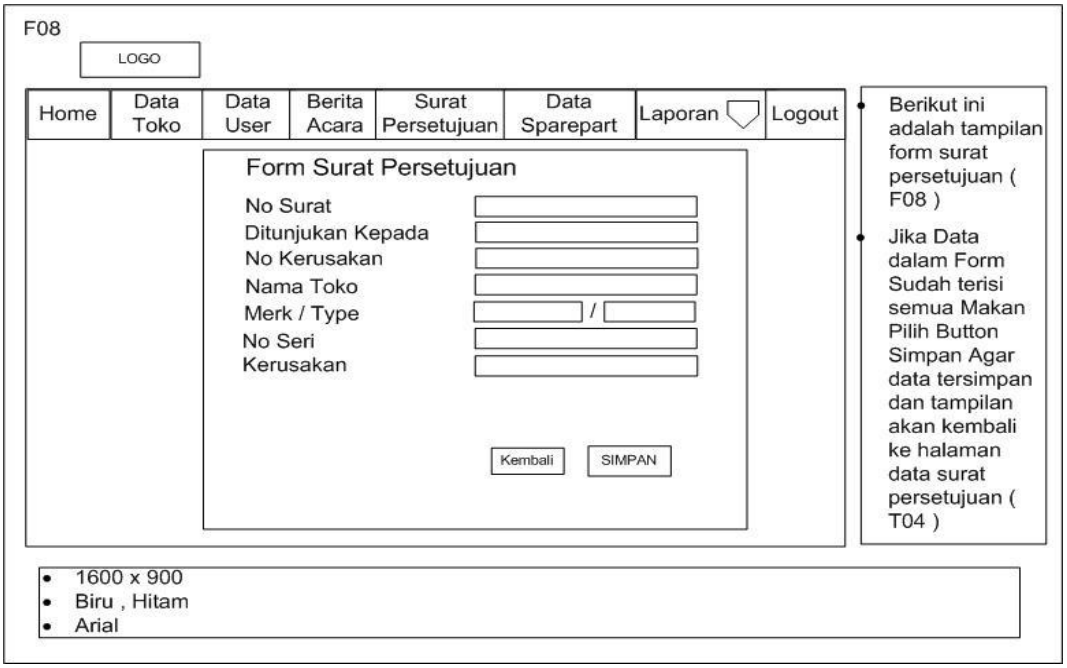
Gambar 0.47 Perancangan Detail berita Acara

12. Perancangan Tampilan Data Surat Persetujuan



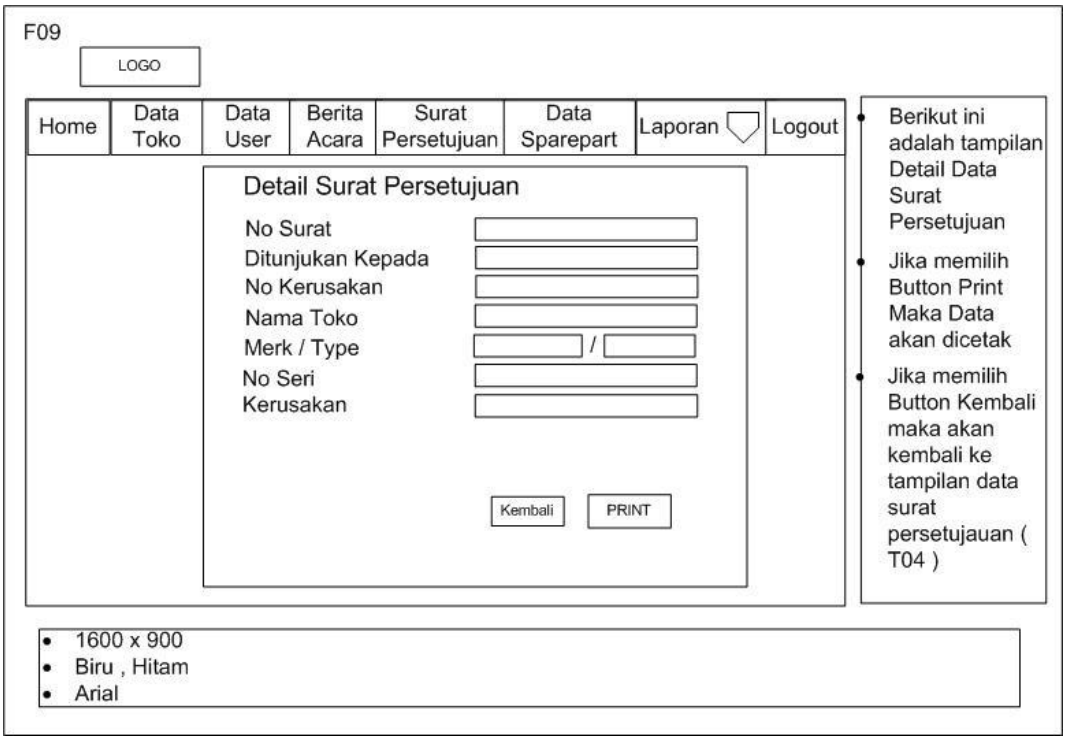
Gambar 0.48 Perancangan Tampilan Data Surat Persetujuan

13. Perancangan Form Surat Persetujuan



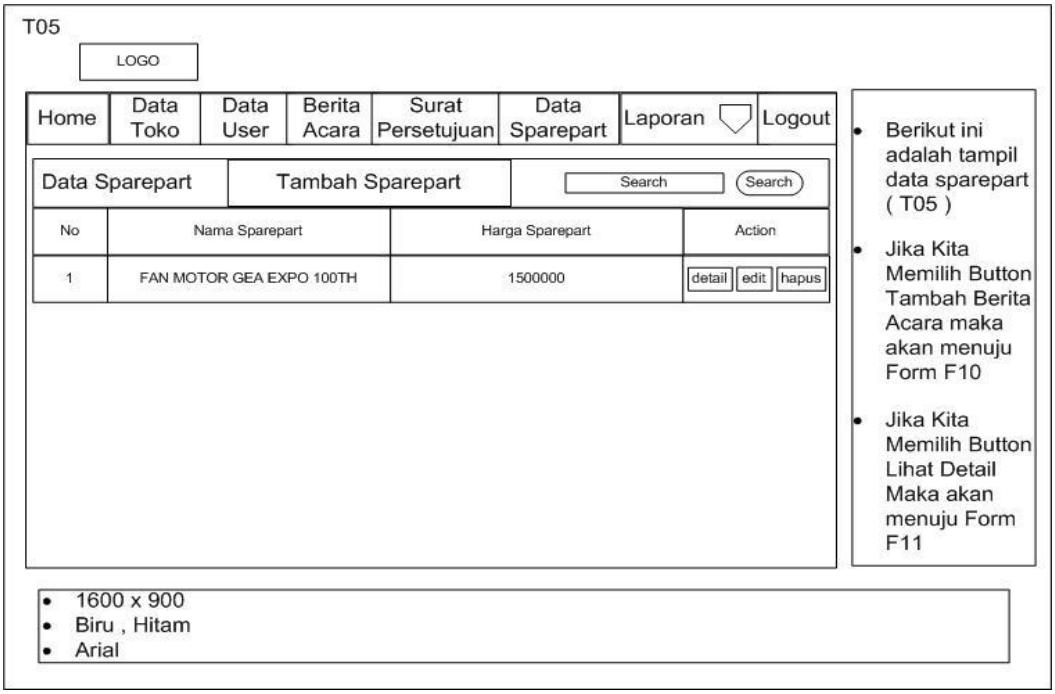
Gambar 0.49 Perancangan Form Surat Persetujuan

14. Perancangan Detail data surat persetujuan



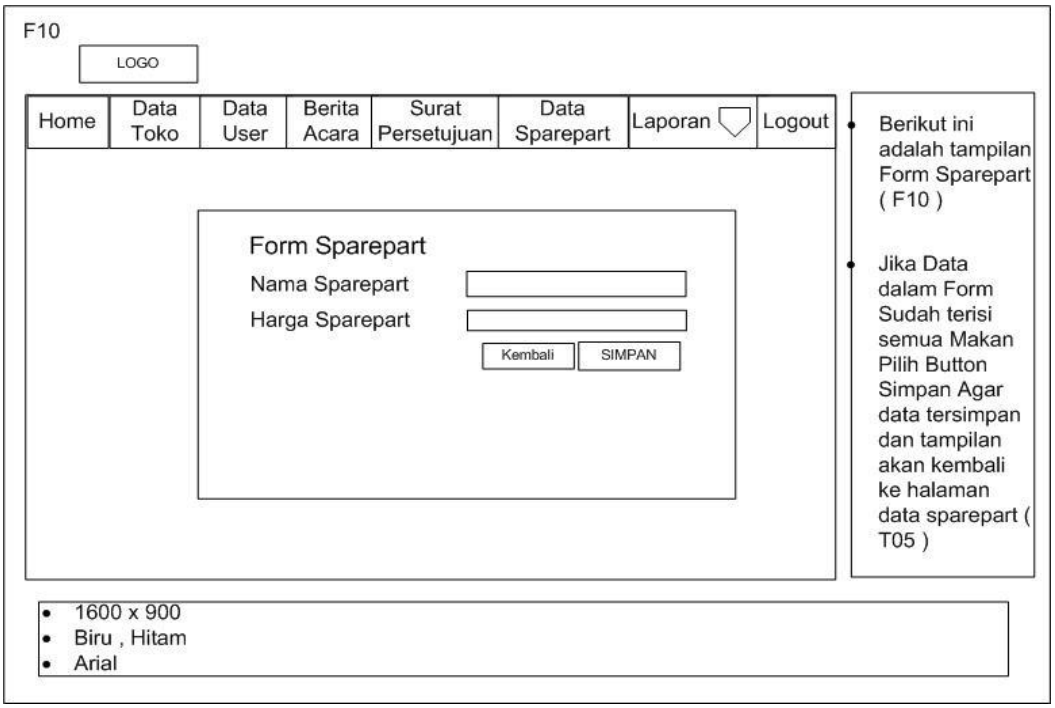
Gambar 0.50 Perancangan Detail Surat Persetujuan

15. Perancangan Tampilan data sparepart



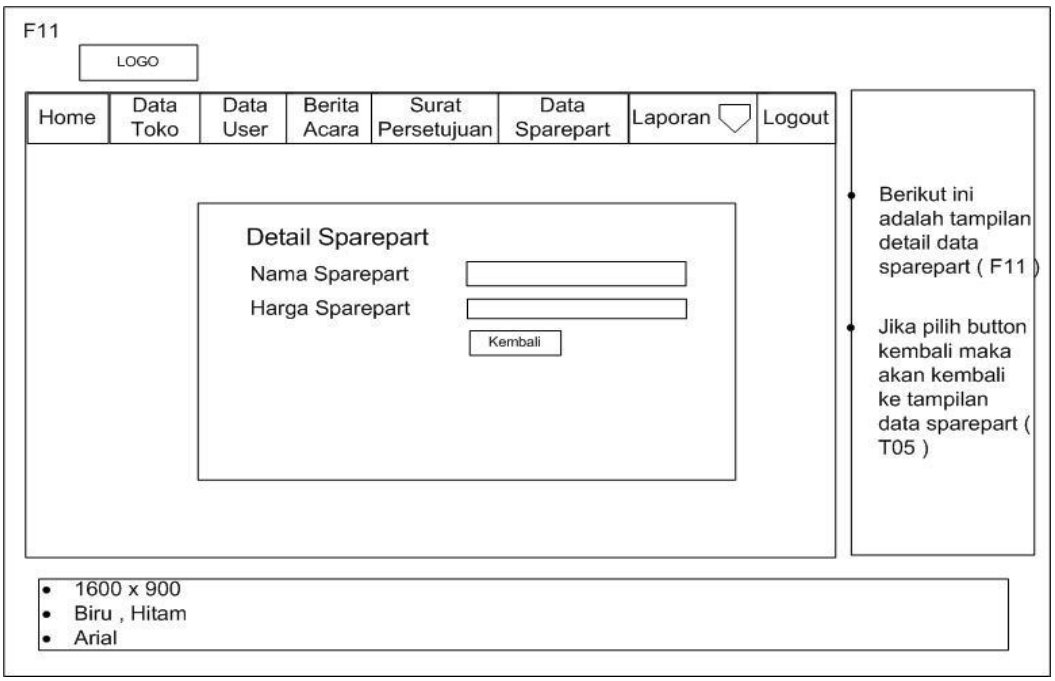
Gambar 0.51 Tampilan perancangan data sparepart

16. Tampilan perancangan form sparepart



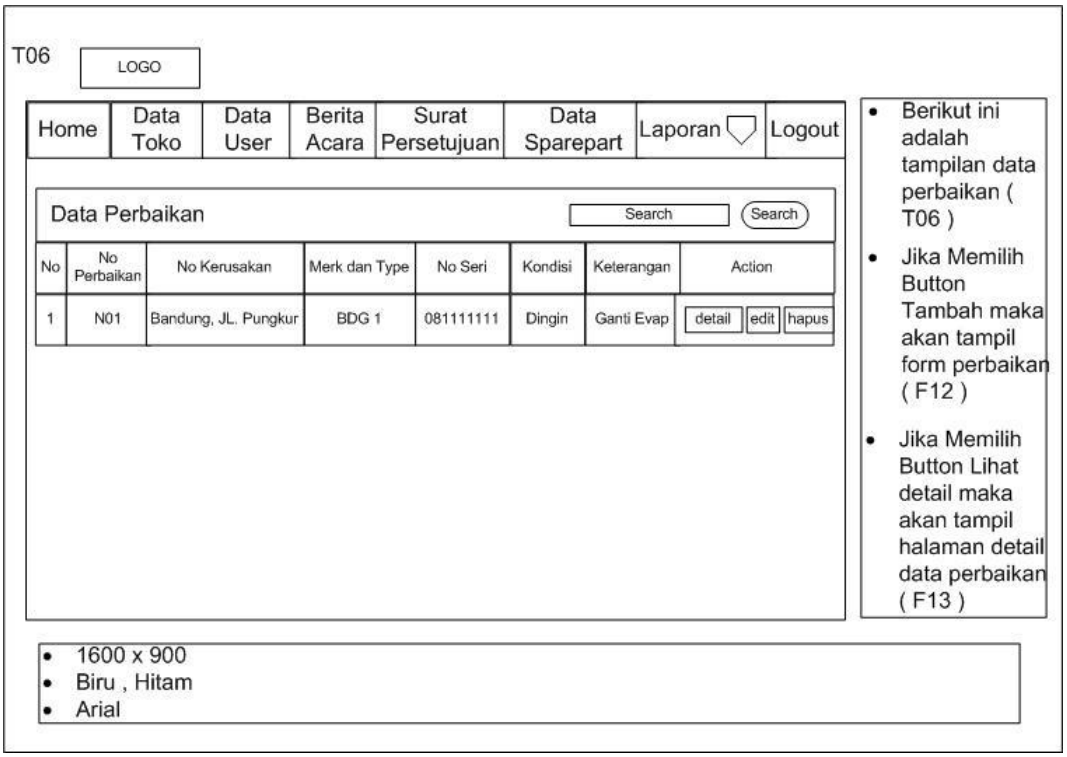
Gambar 0.52 Tampilan Perancangan form saprepart

17. Tampilan Perancangan Detail Data Sparepart



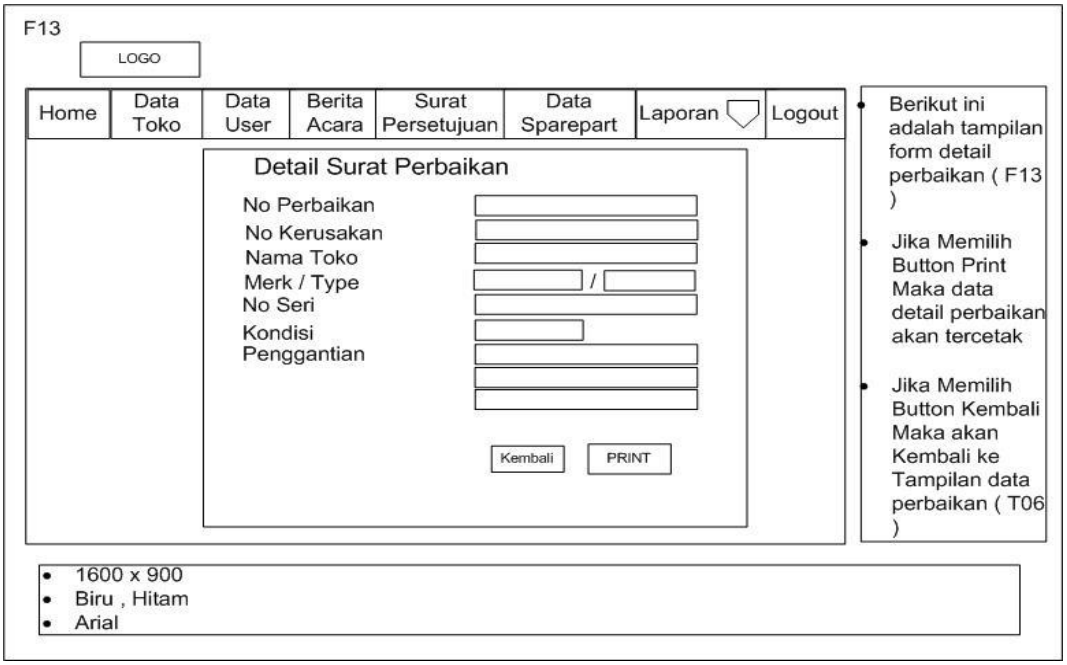
Gambar 0.53Tampilan perancangan detail data sparepart

18. Perancangan Tampilan data laporan perbaikan



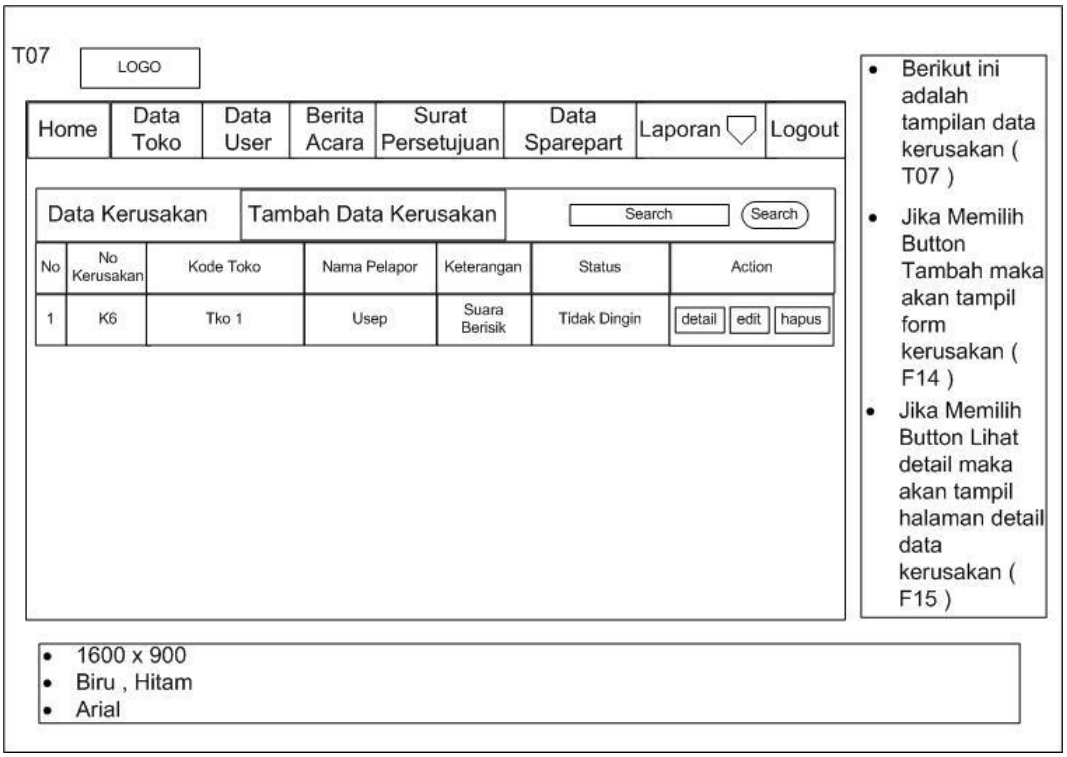
Gambar 0.54 perancangan tampilan data perbaikan

19. Perangcan Tampilan Detail Perbaikan



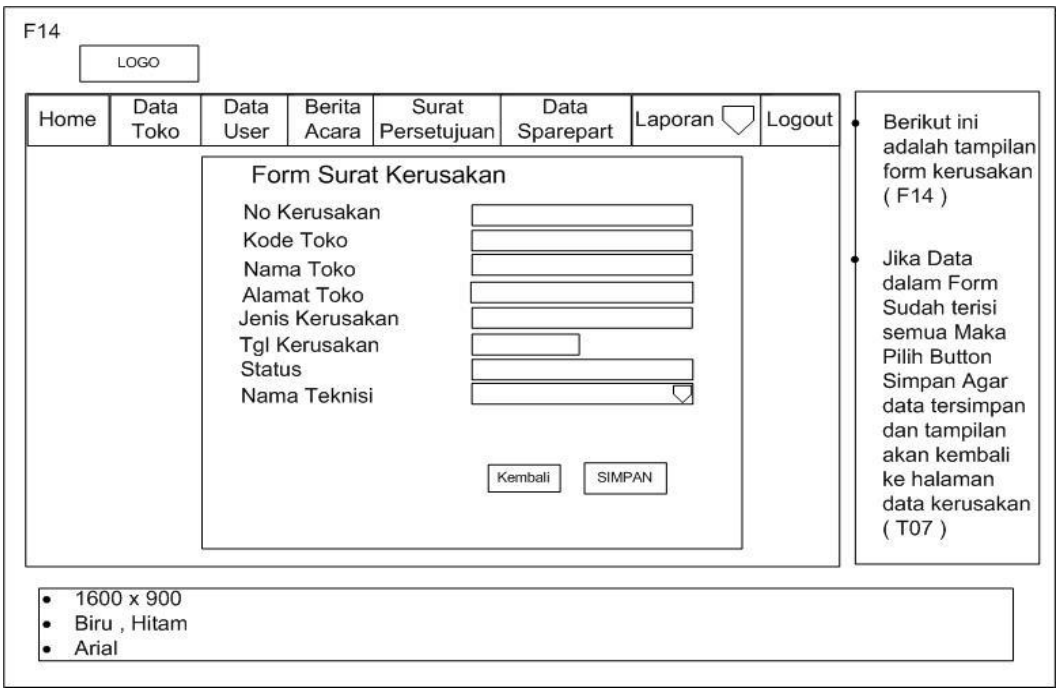
Gambar 0.55 Perancangan Tampilan Detail Perbaikan

20. Perancangan Tampilan Data Kerusakan



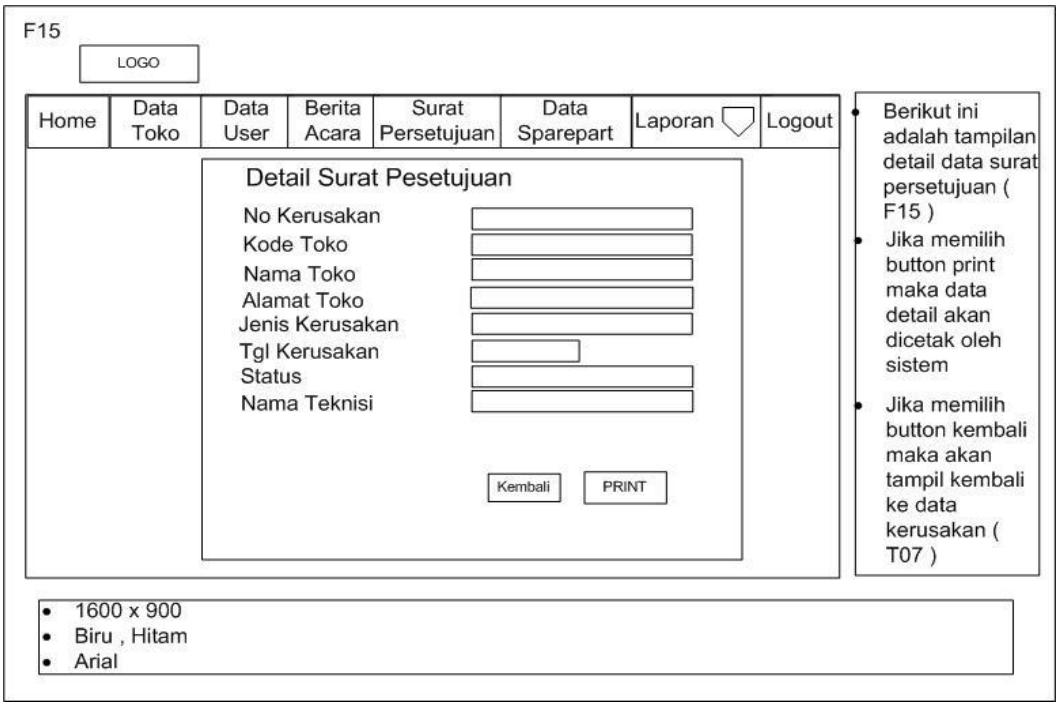
Gambar 0.56 Perancangan tampilan data kerusakan

21. Perancangan tampilan form kerusakan



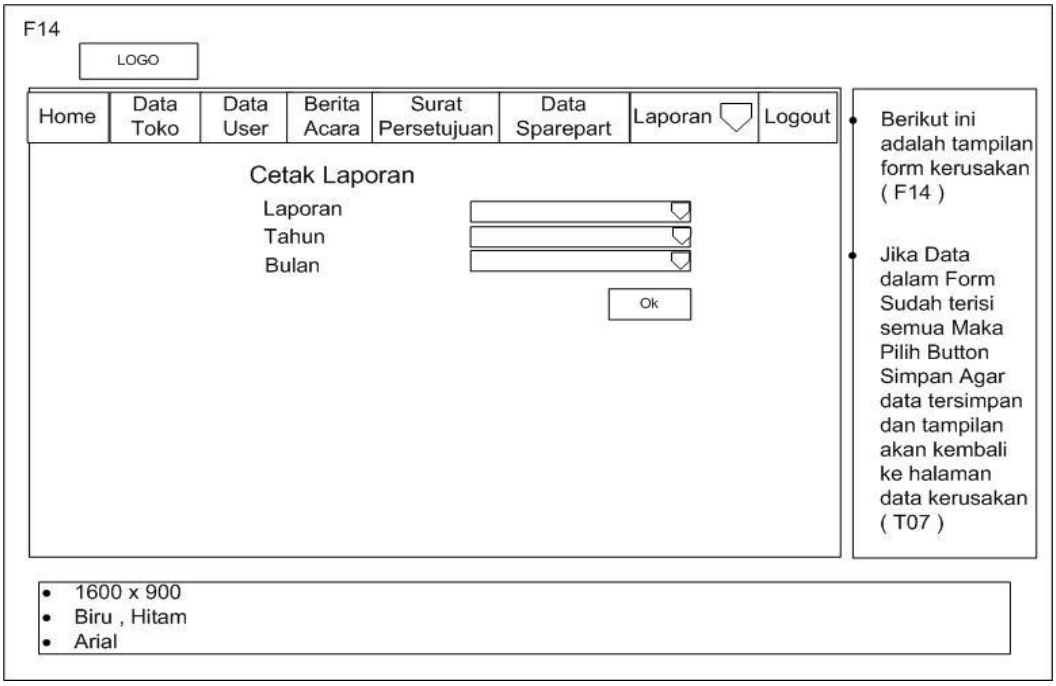
Gambar 0.57 Tampilan perancangan form kerusakan

22. Perancangan tampilan form detail kerusakan



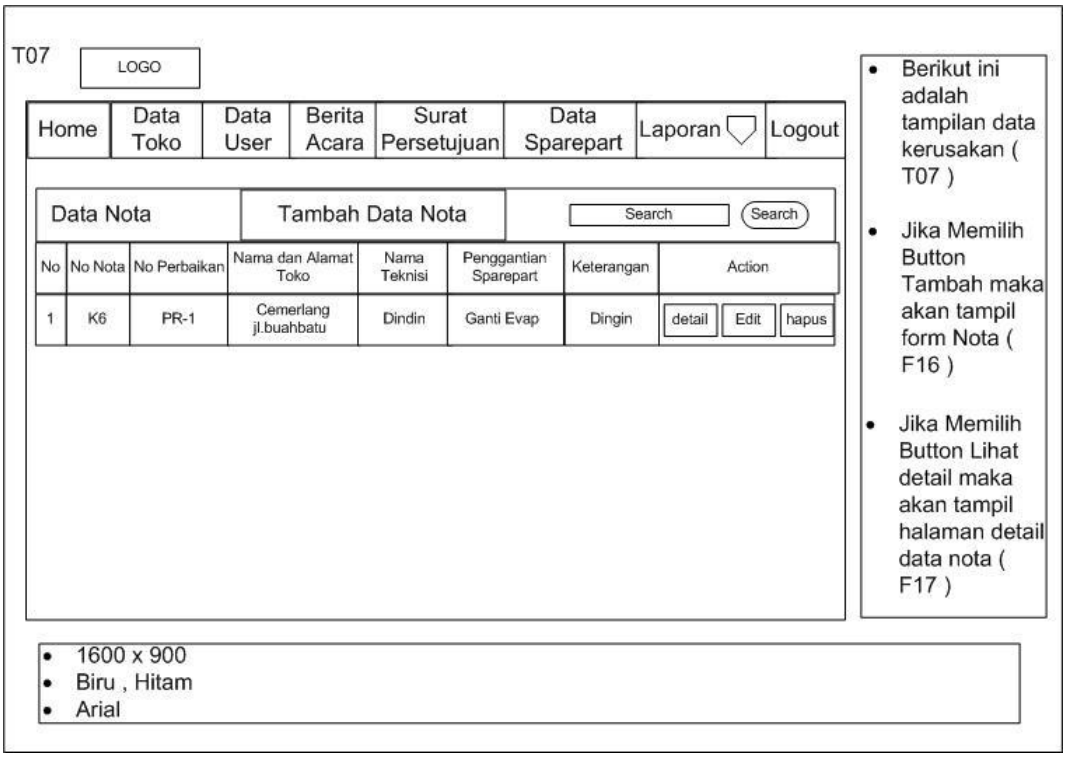
Gambar 0.58 Perancangan tampilan detail kerusakan

23. Perancangan tampilan Cetak Laporan



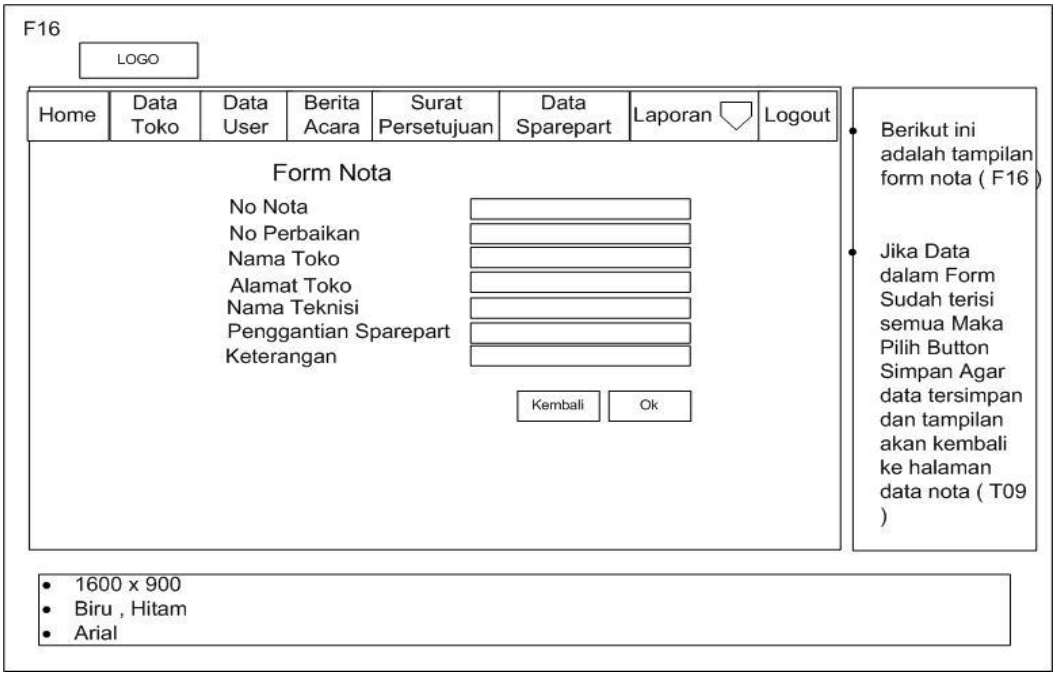
Gambar 0.59 Perancangan tampilan cetak laporan

24. Perancangan Tampilan data nota



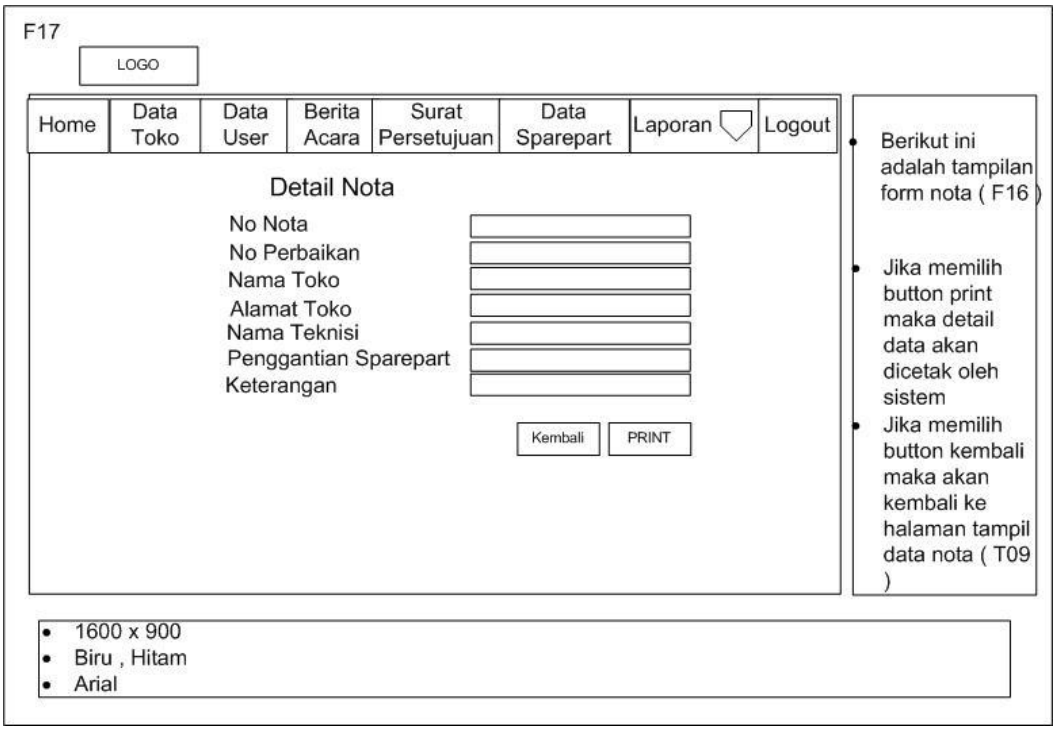
Gambar 0.60 Perancangan Tampilan Data Nota

25. Tampilan perancangan form nota



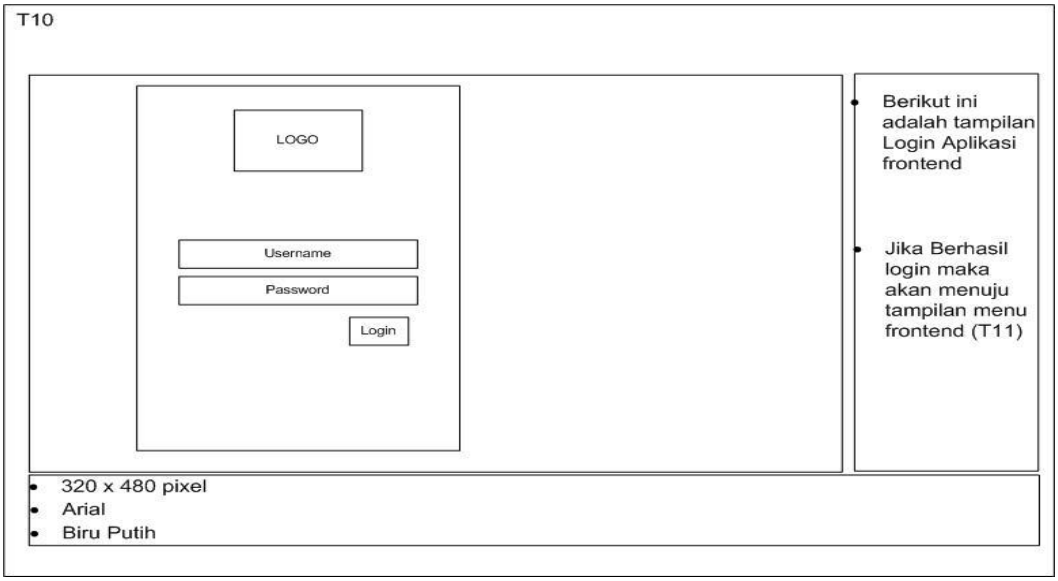
Gambar 0.61 Perancangan Tamppilan form nota

26. Perancangan Tampilan detail Nota



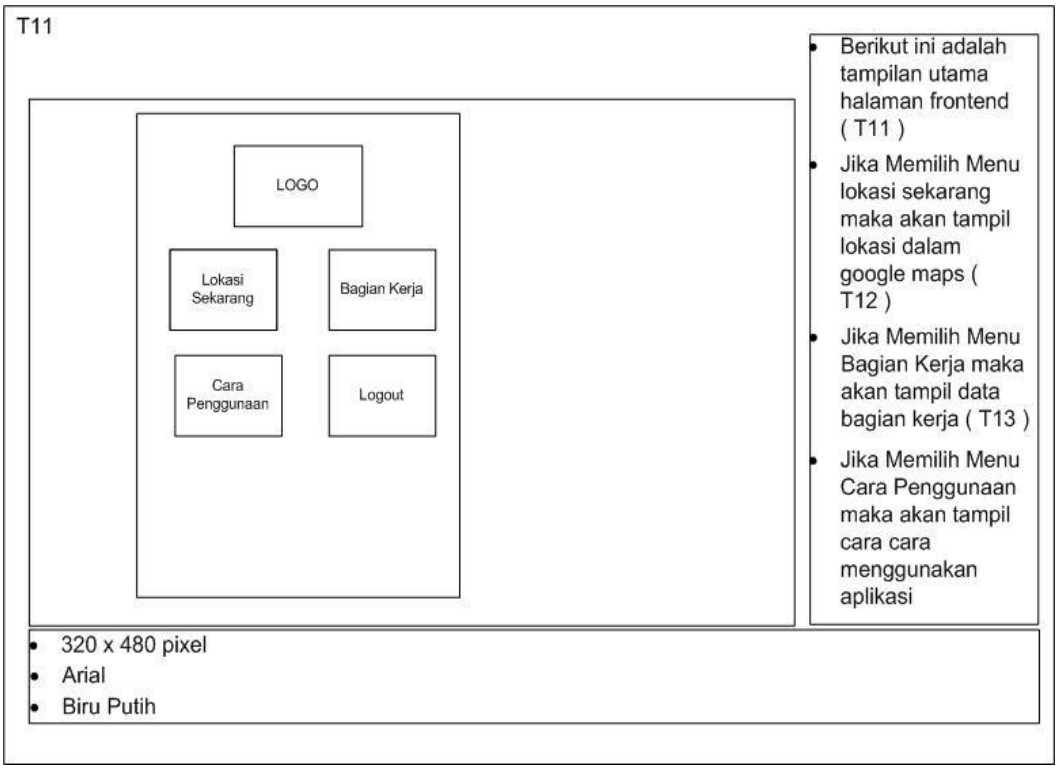
Gambar 0.62 Perancangan tampilan data nota

27. Perancangan Tampilan login *Frontend*



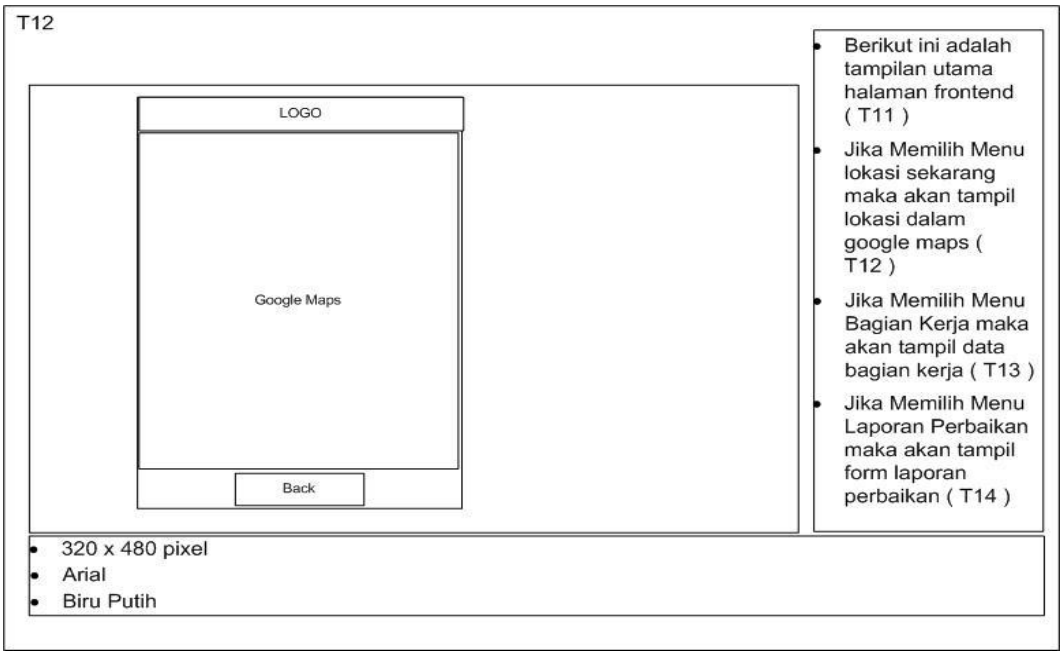
Gambar 0.63 Perancangan Tampilan login *Frontend*

28. Perancangan Tampilan Menu *Frontend*



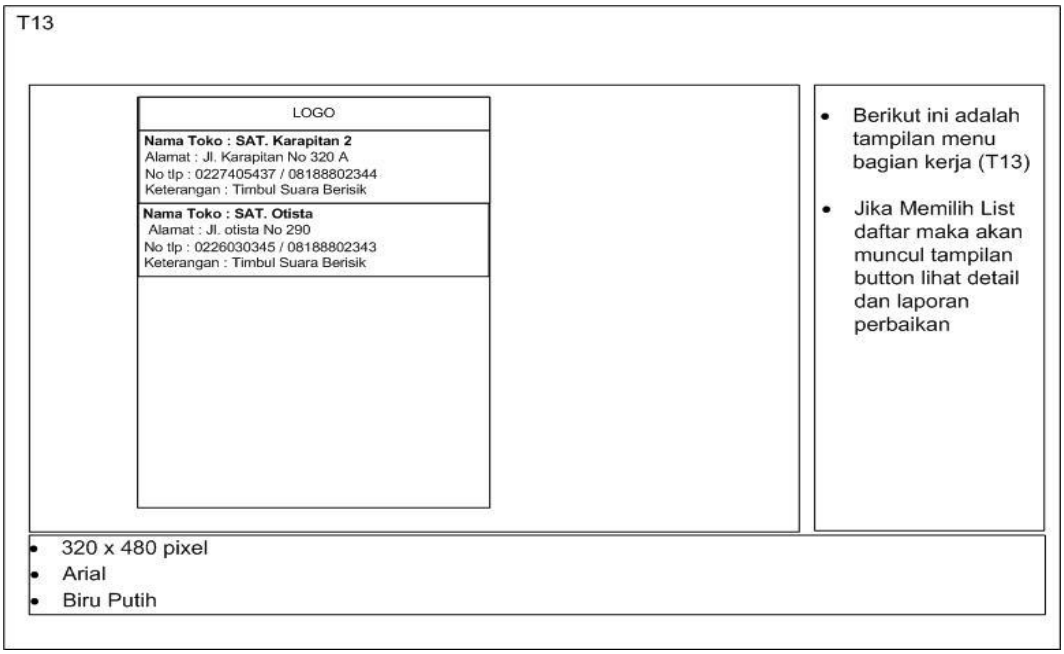
Gambar 0.64 Perancangan Tampilan Menu *Frontend*

29. Perancangan Tampilan Menu Lokasi Sekarang



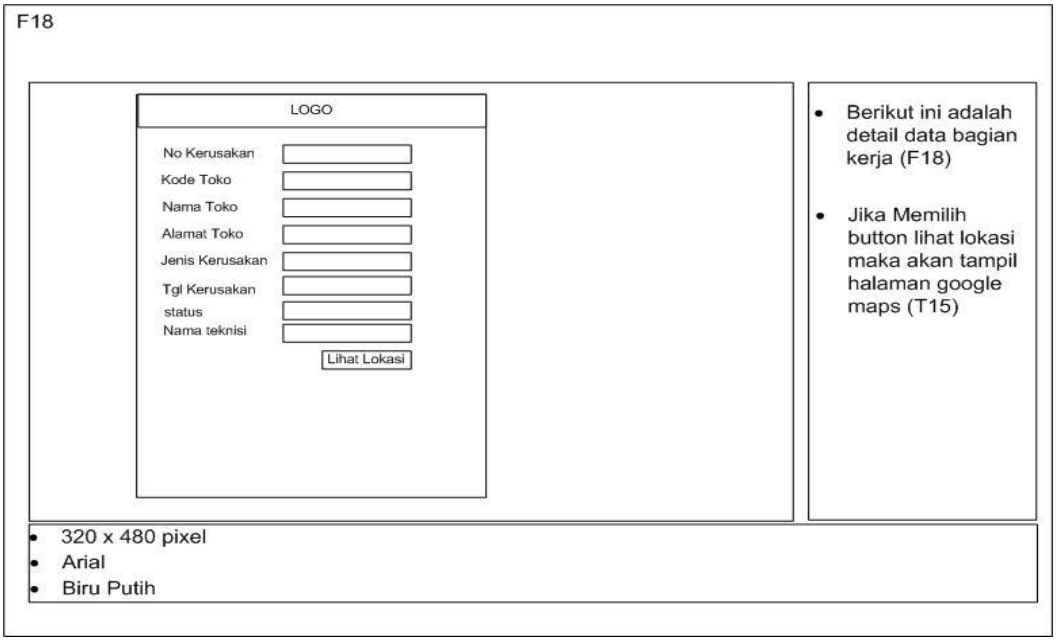
Gambar 0.65 Perancangan tampilan menu lihat lokasi sekarang

30. Perancangan Tampilan Menu Bagian kerja



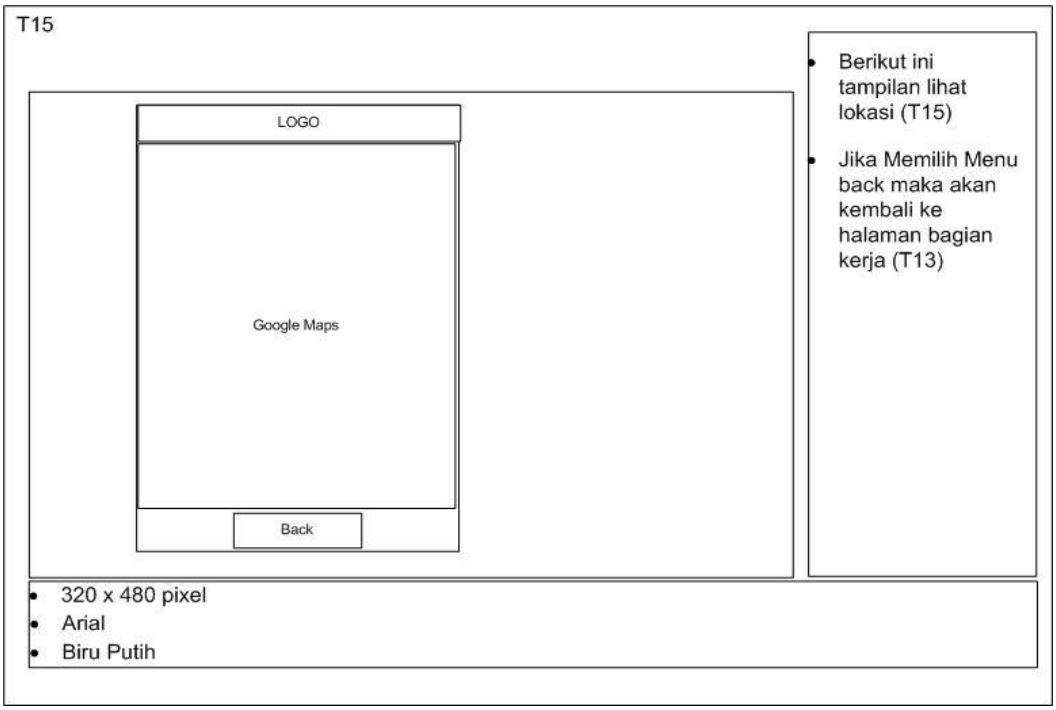
Gambar 0.66 Perancangan tampilan menu bagian kerja

31. Perancangan tampilan menu lihat detail bagian kerja



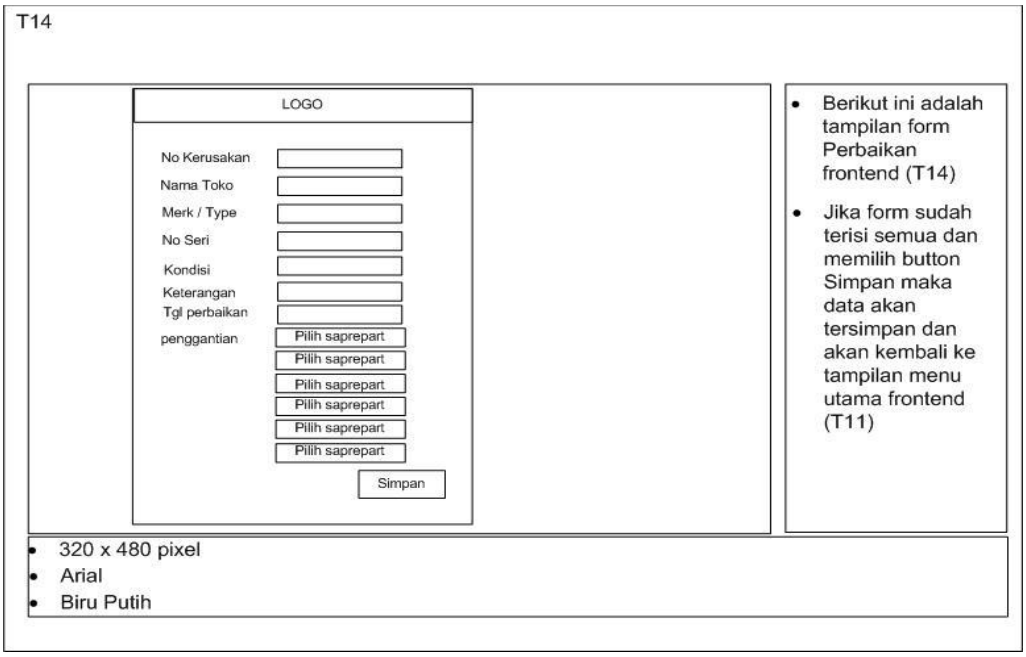
Gambar 0.67 Perancangan menu lihat detail bagian kerja

32. Perancangan Tampilan lihat lokasi



Gambar 0.68 Perancangan menu lihat lokasi pada bagian kerja

33. Perancangan menu laporan perbaikan



Gambar 0.69 Perancangan menu laporan perbaikan *Frontend*