BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

Tahap Implementasi sistem merupakan suatu proses penerapan rancangan program yang telah dibuat ke dalam sebuah aplikasi pemrograman sesuai dengan tujuan yang diharapkan dari program aplikasi tersebut.

4.1 Implementasi

Adapun tujuan dari implementasi sistem adalah sebagai berikut :

1. Mengkaji rangkaian sistem baik dari segi *software* maupun hardware sebagai sarana pengolahan data dan penyajian data .
2. Menyelesaikan rancangan sistem yang ada didalam dokumentasi sistem yang baru atau yang telah disetujui.
3. Memastikan bahwa pemakai dapat mengoperasikan dengan mudah sistem yang telah dibuat.
4. Memastikan bahwa sistem telah berjalan dengan lancar dengan mengontrol dan melakukan instalasi secara benar.
5. Memperhitungkan bahwa sistem telah memenuhi permintaan pemakai yaitu dengan menguji sistem secara menyeluruh

4.1.1 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan pada sistem komputer yang digunakan untuk membangun Aplikasi *Reporting Service Showcase* adalah sebagai berikut terlihat pada table 4.1 :

Tabel 4.1 Implementasi Perangkat Lunak

Perangkat Lunak	Spesifikasi
Sistem Operasi	*Windows 7, Android OS 2.2 Froyo*
Bahasa Pemrograman	*PHP,Java*
Web Server	*XMPP, Localhost*
Database server	*MySQL*

Web Browser	*Mozilla Firefox, Google Chrome*
Tools pembangun	*Eclipse, Android ADT*

4.1.2 Implementasi Perangkat Keras

Kebutuhan minimum perangkat keras (*hardware*) yang diperlukan untuk mengimplementasikan program aplikasi *reporting service* berbasis android yang dibuat adalah perangkat keras komputer PC dan *Mobile Phone* Android. Semakin tinggi spesifikasi komputer dan *Mobile Phone* yang digunakan untuk menjalankan aplikasi, akan semakin baik. Kebutuhan minimum perangkat keras untuk pembangun aplikasi *reporting service* terlihat pada table 4.2 :

Tabel 4.2 Implementasi Perangkat Keras

Perangkat Keras	Spesifikasi
Processor	*Intel Pentium Core 2 Duo 2.93Ghz*
Monitor	CRT Standar
RAM	2 GB
VGA	512 MB
Hardisk	250 GB
CPU	600 MHz ARMv6
Display	*TFT capacitive touchscreen*
Storage	160 MB *storage*
RAM	384MB RAM

4.1.3 Implementasi Basis Data

Implementasi basis data diambil berdasarkan perancangan basis data yang dibuat sebelumnya. Pembuatan *database* dilakukan dengan menggunakan DBMS MySQL. Implementasi *database* dalam bahasa *SQL* terlihat pada gambar 4.1 – 4.11 :

1. Implementasi Tabel Berita

```
CREATE TABLE IF NOT EXISTS `tbl_berita` (
  `no_berita` int(10) NOT NULL AUTO_INCREMENT,
```

```
  `nama_pengisi` varchar(250) NOT NULL,
  `id_user` int(10) NOT NULL,
  `nik` varchar(50) NOT NULL,
  `jabatan` varchar(200) NOT NULL,
  `hp_pengisi` varchar(50) NOT NULL,
  `no_kerusakan` int(10) NOT NULL,
  `merk_b` varchar(150) NOT NULL,
  `type_b` varchar(150) NOT NULL,
  `no_seri_b` varchar(150) NOT NULL,
  `nama_toko` varchar(250) NOT NULL,
  `id_jenis` int(10) NOT NULL,
  `keterangan` text NOT NULL,
  `tgl_berita` date NOT NULL,
  PRIMARY KEY (`no_berita`),  ADD CONSTRAINT tbl_berita_ibfk_2 FOREIGN KEY (id_jenis) REFERENCES tbl_jenis (id_jenis),
ADD CONSTRAINT tbl_berita_ibfk_1 FOREIGN KEY (no_kerusakan) REFERENCES tbl_kerusakan (no_kerusakan)
) ENGINE=InnoDB
```

Gambar 4.1 Implementasi Tabel Berita

2. Implementasi Tabel Detail

```
CREATE TABLE IF NOT EXISTS `tbl_detail` (
  `no_detail` int(10) NOT NULL AUTO_INCREMENT,
  `no_perbaikan` int(10) NOT NULL,
  `id_sparepart` int(10) NOT NULL,
  `qty` int(10) NOT NULL,
  `harga_sparepart` varchar(100) NOT NULL,
  `no_kerusakan` int(150) NOT NULL,
  PRIMARY KEY (`no_detail`),
  ADD CONSTRAINT tbl_detail_ibfk_3 FOREIGN KEY (no_kerusakan) REFERENCES tbl_kerusakan (no_kerusakan),
  ADD CONSTRAINT tbl_detail_ibfk_2 FOREIGN KEY (id_sparepart) REFERENCES tbl_sparepart (id_sparepart),
ADD CONSTRAINT tbl_detail_ibfk_1 FOREIGN KEY (no_perbaikan) REFERENCES tbl_perbaikan (no_perbaikan)
) ENGINE=InnoDB
```

Gambar 4.2 Implementasi tabel Detail

3. Implementasi Tabel Detail Surat

```
CREATE TABLE IF NOT EXISTS `tbl_detail_srt` (
```

```
  `no_detail` int(150) NOT NULL AUTO_INCREMENT,
  `no_surat` int(150) NOT NULL,
  `id_sparepart` int(50) NOT NULL,
  `qty` int(150) NOT NULL,
  `harga_sparepart` varchar(100) NOT NULL,
  PRIMARY KEY (`no_detail`),
ADD CONSTRAINT tbl_detail_srt_ibfk_2 FOREIGN KEY (id_sparepart)
REFERENCES tbl_sparepart (id_sparepart),
  ADD CONSTRAINT tbl_detail_srt_ibfk_1 FOREIGN KEY (no_surat)
REFERENCES tbl_surat (no_surat)
) ENGINE=InnoDB
```

Gambar 4.3 Implementasi Tabel Detail Surat

4. Implementasi Tabel Faktur

```
CREATE TABLE IF NOT EXISTS `tbl_faktur` (
  `no_faktur` int(10) NOT NULL AUTO_INCREMENT,
  `no_perbaikan` int(10) NOT NULL,
  `nama_toko` varchar(250) NOT NULL,
  `alamat_toko` text NOT NULL,
  `id_teknisi` int(10) NOT NULL,
  `tgl` date NOT NULL,
  PRIMARY KEY (`no_faktur`),
ADD CONSTRAINT tbl_faktur_ibfk_1 FOREIGN KEY (no_perbaikan)
REFERENCES tbl_perbaikan (no_perbaikan)
) ENGINE=InnoDB
```

Gambar 4.4 Implementasi Tabel Faktur

5. Impelementasi Tabel Kerusakan

```
CREATE TABLE IF NOT EXISTS `tbl_kerusakan` (
  `no_kerusakan` int(10) NOT NULL AUTO_INCREMENT,
  `kode_toko` int(10) NOT NULL,
  `id_user` int(10) NOT NULL,
  `keterangan` int(10) NOT NULL,
  `app` int(10) NOT NULL,
  `id_teknisi` int(10) NOT NULL,
  `tgl_kerusakan` date NOT NULL,
  `status` int(10) NOT NULL,
  PRIMARY KEY (`no_kerusakan`),
  ADD CONSTRAINT tbl_kerusakan_ibfk_1 FOREIGN KEY (kode_toko)
REFERENCES tbl_toko (kode_toko),
```

```
  ADD CONSTRAINT tbl_kerusakan_ibfk_2 FOREIGN KEY (id_user)
REFERENCES tbl_user (id_user)
) ENGINE=InnoDB
```

Gambar 4.5 Implementasi Tabel kerusakan

6. Implementasi Tabel Perbaikan

```
CREATE TABLE IF NOT EXISTS `tbl_perbaikan` (
  `no_perbaikan` int(10) NOT NULL AUTO_INCREMENT,
  `no_kerusakan` int(10) NOT NULL,
  `nama_toko` varchar(250) NOT NULL,
  `merk` varchar(150) NOT NULL,
  `type` varchar(150) NOT NULL,
  `no_seri` varchar(150) NOT NULL,
  `kondisi` varchar(150) NOT NULL,
  `tgl_perbaikan` date NOT NULL,
  `longitude` double NOT NULL,
  `latitude` double NOT NULL,
  PRIMARY KEY (`no_perbaikan`),
ALTER TABLE `tbl_perbaikan`
  ADD CONSTRAINT tbl_perbaikan_ibfk_1 FOREIGN KEY (no_kerusakan)
REFERENCES tbl_kerusakan (no_kerusakan)
) ENGINE=InnoDB
```

Gambar 4.6 Implementasi Tabel Perbaikan

7. Implementasi Tabel Sparepart

```
CREATE TABLE tbl_sparepart (
  id_sparepart int(10) NOT NULL AUTO_INCREMENT,
  nama_sparepart varchar(100) NOT NULL,
  harga_sparepart varchar(100) NOT NULL,
  PRIMARY KEY (id_sparepart)
) ENGINE=InnoDB
```

Gambar 4.7 Impelemntasi Tabel Sparepart

8. Implementasi Tabel Surat

```
CREATE TABLE IF NOT EXISTS `tbl_surat` (
  `no_surat` int(10) NOT NULL AUTO_INCREMENT,
  `no_kerusakan` int(10) NOT NULL,
  `id_buat` int(10) NOT NULL,
  `nama_toko` varchar(250) NOT NULL,
```

```
  `id_user` int(10) NOT NULL,
  `merk` varchar(150) NOT NULL,
  `type` varchar(150) NOT NULL,
  `no_seri` varchar(150) NOT NULL,
  `status` varchar(250) NOT NULL,
  `keterangan` varchar(25) NOT NULL,
  `tanggal` date NOT NULL,
  PRIMARY KEY (`no_surat`),
ALTER TABLE `tbl_surat`
  ADD CONSTRAINT tbl_surat_ibfk_1 FOREIGN KEY (no_kerusakan)
REFERENCES tbl_kerusakan (no_kerusakan)
) ENGINE=InnoDB
```

Gambar 4.8 Implementasi Tabel Surat

9. Implementasi Tabel Toko

```
CREATE TABLE IF NOT EXISTS `tbl_toko` (
  `kode_toko` int(10) NOT NULL AUTO_INCREMENT,
  `kategori` int(1) NOT NULL,
  `id_user` int(10) NOT NULL,
  `id_kepala` int(10) NOT NULL,
  `nama_toko` varchar(250) NOT NULL,
  `area_toko` varchar(150) NOT NULL,
  `tlp_toko` varchar(50) NOT NULL,
  `hp_toko` varchar(50) NOT NULL,
  `alamat_toko` text NOT NULL,
  `tgl_go` date NOT NULL,
  `kodepos_toko` varchar(50) NOT NULL,
  `longitude` double NOT NULL,
  `latitude` double NOT NULL,
  PRIMARY KEY (`kode_toko`),
  ALTER TABLE `tbl_toko`
  ADD CONSTRAINT tbl_toko_ibfk_1 FOREIGN KEY (id_user)
REFERENCES tbl_user (id_user) )
ENGINE=InnoDB
```

Gambar 4.9 Implementasi Tabel Toko

10. Implementasi Tabel *User*

```
CREATE TABLE IF NOT EXISTS `tbl_user` (
  `id_user` int(10) NOT NULL AUTO_INCREMENT,
  `nama` varchar(150) NOT NULL,
```

```
  `nip` int(10) NOT NULL,
  `alamat` text NOT NULL,
  `no_hp` varchar(50) NOT NULL,
  `level` varchar(50) NOT NULL,
  `foto` varchar(100) NOT NULL,
  `user`name` varchar(100) NOT NULL,
  `password` varchar(100) NOT NULL,
  PRIMARY KEY (`id_user`),
  KEY nama (nama)
) ENGINE=InnoDB
```

Gambar 4.10 Implementasi Tabel *User*

4.1.4 Implementasi Class

Implementasi *class* pada Aplikasi *Reporting Service* Berbasis Android ini merupakan implementasi dari perancangan *class* diagram yang terbentuk dari *Use Case* diagram kedalam bentuk fisik file-file dengan ekstensi *.java* maupun ekstensi *.php* yang digunakan dalam pembuatan dan jalannya proses dalam sistem. Berikut pada tabel 4.1 beberapa implementasi *class* yang telah terdefinisi.

Tabel 4.3 Implementasi Class

No	Nama Kelas	Nama File Fisik
1	Login	LoginActivity.java
2	Cperbaikan	LihatLokasiActivity.java
3	Ckerusakan	LaporanPerbaikan.java
4	Csparepart	BagianKerjaActivity.java
5	Main	DashboardActivity.java
6	C*user*	C*user*.php
7	Ckerusakan	Ckerusakan.php
8	Cperbaikan	Cperbaikan.php
9	Csurat	Csurat.php
10	Cberita	Cberita.php

11	Cnota	Cnota.php
12	Ccetak	Ccetak.php
13	Csparepart	Csparepart.php
14	Ctoko	Ctoko.php
15	Utama	Utama.php

4.1.5 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap halaman yang dibuat baik aplikasi *Frontend* yang dibangun di perangkat *mobile* maupun aplikasi *Backend* yang dibangun menggunakan *Web*. Berikut ini beberapa tampilan antar muka yang telah di implementasikan.

4.1.5.1 Implementasi Antarmuka *Backend*

Antar muka aplikasi *Backend* terdiri dari beberapa halaman utama yaitu halaman menu *login*, halaman menu utama, halaman pengolahan data *user* , data toko, data berita acara, surat persetujuan , data sparepart, laporan perbaikan , laporan kerusakan , cetak laporan , cetak nota halaman isian konten.

1. Halaman Login *Backend*

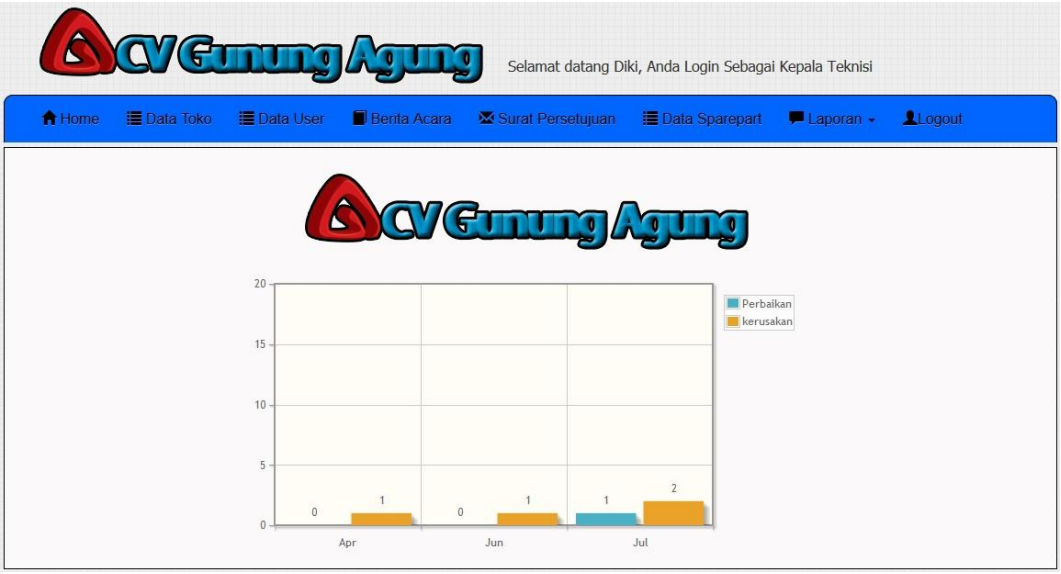
   Halaman Login *Backend* merupakan Halaman Login Dimana setiap *user* yang mengakses *Web* cvgunungagung.com langsung menuju ke halaman utama *Backend*.



Gambar 4.11 Halaman *Login Backend*

2. Halaman Utama

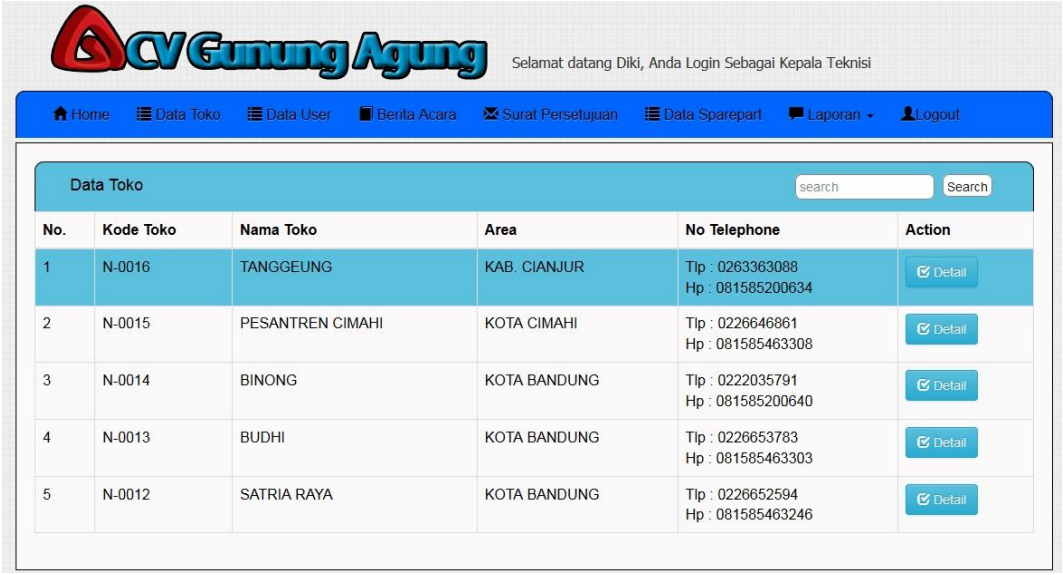
   Halaman utama *Backend* Kepala teknisi merupakan halam utama saat teknisi telah melakukan login.



Gambar 4.12 Halaman utama Kepala Teknisi

2. Menu Data Toko

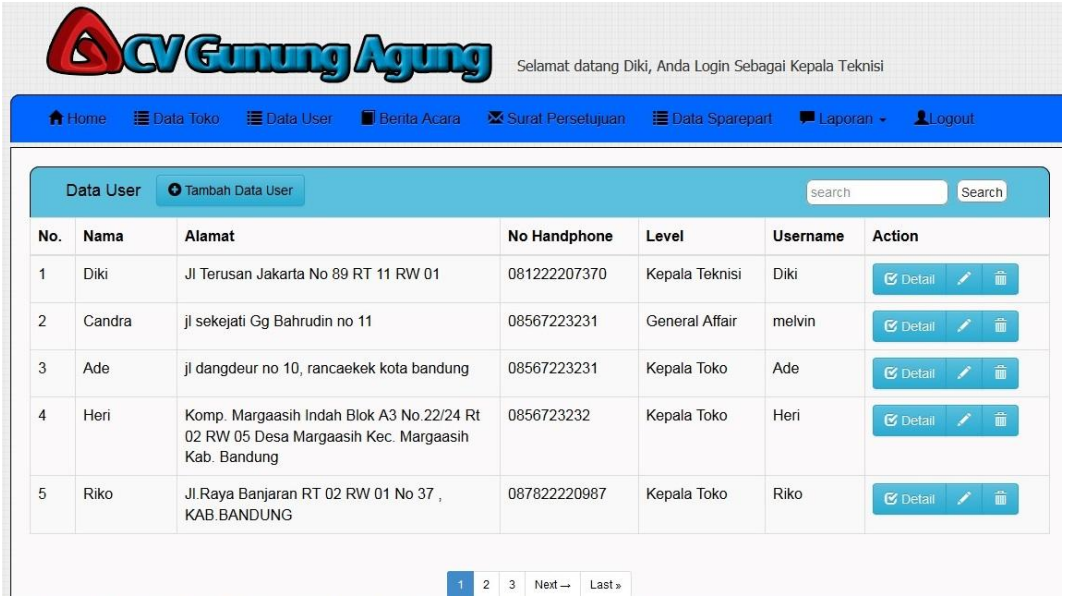
Tampilan Menu Data toko saat kepala teknisi memilih menu data toko.



Gambar 4.13 Tampilan Menu Data Toko

3. Menu Data *User*

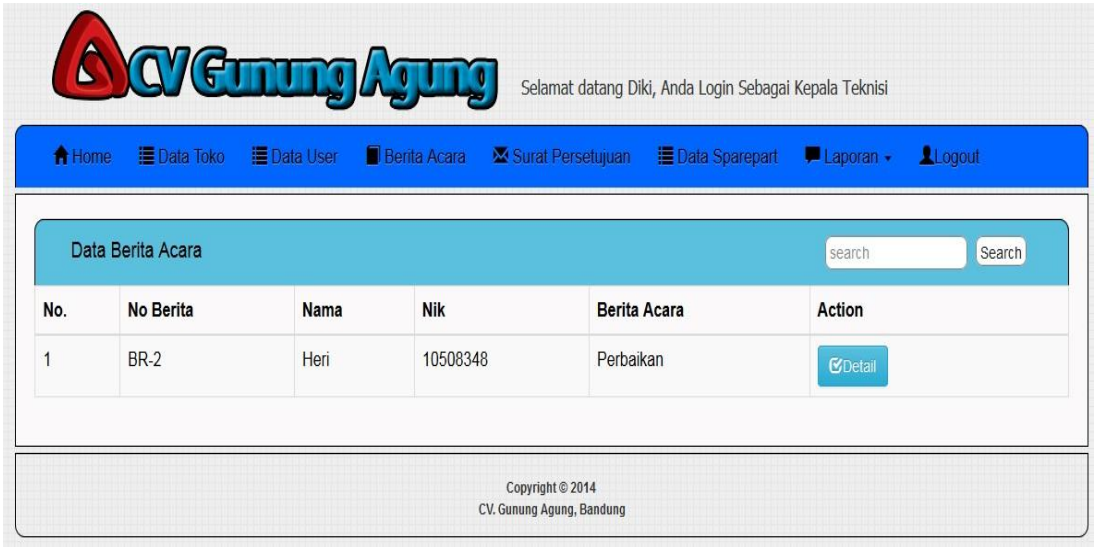
Tampilan data *user* dari aplikasi *Backend* jika seorang keala teknisi memilih menu data *user*.



Gambar 4.14 Tampilan Menu Data *User*

4. Menu Berita Acara

   Tampilan menu berita acara dari aplikasi *Backend* jika seorang kepala teknisi memilih menu berita acara.



Gambar 4.15 Tampilan menu berita acara

5. Menu Surat Persetujuan

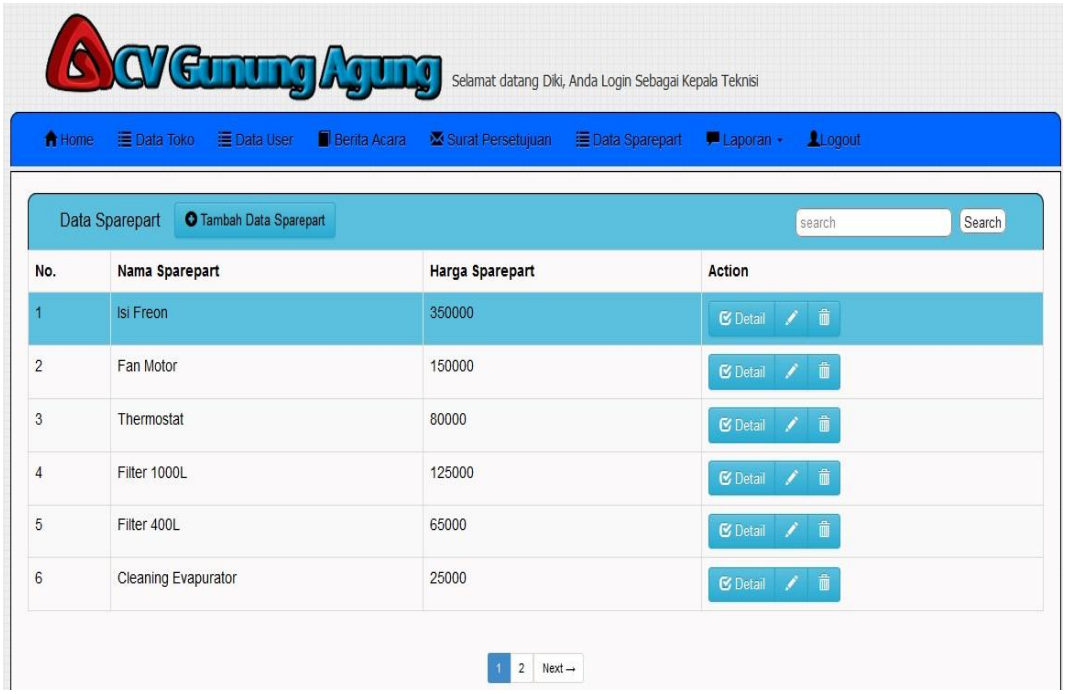
   Tampilan menu surat persetujuan dari aplikasi *Backend* jika seorang kepala teknisi memilih menu surat persetujuan.



Gambar 4.16 Tampilan menu surat persetujuan

6. Menu Data Saprepart

   Tampilan menu data sparepart jika seorang kepala teknisi memilih menu data sparepart.



Gambar 4.17 Tampilan Menu Data Saprepart

7. Tampilan Menu Laporan Perbaikan *Backend*

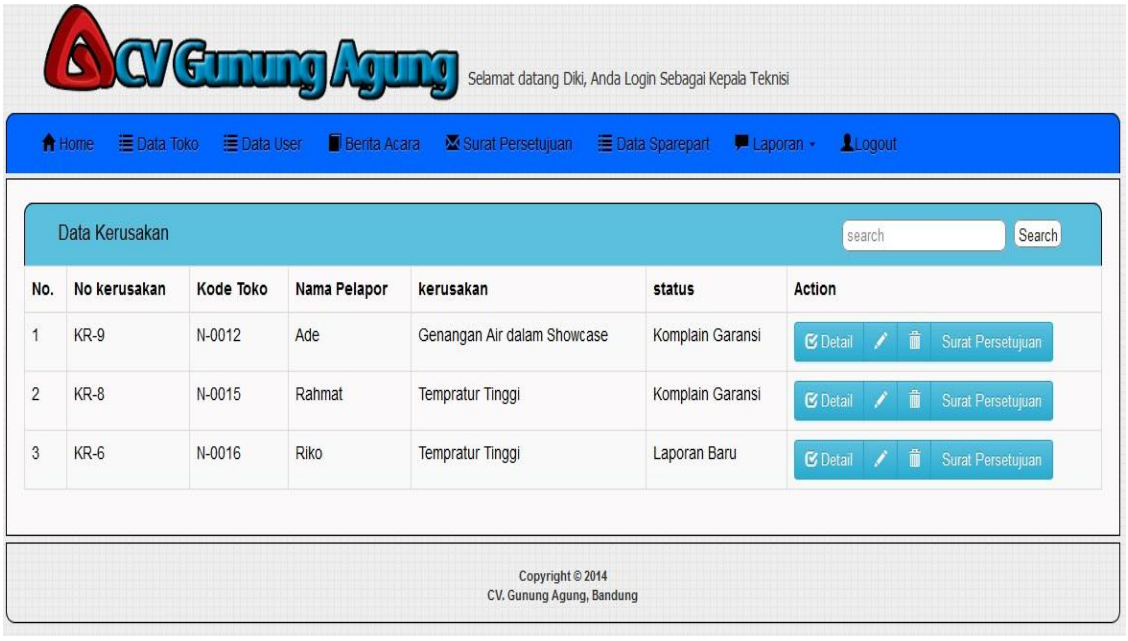
   Tampilan menu laporan perbaikan *Backend* akan tampil jika seorang kepala teknisi memilih menu laporan perbaikan.



Gambar 4.18 Tampilan Menu Laporan Perbaikan

8. Tampilan Menu Laporan Kerusakan

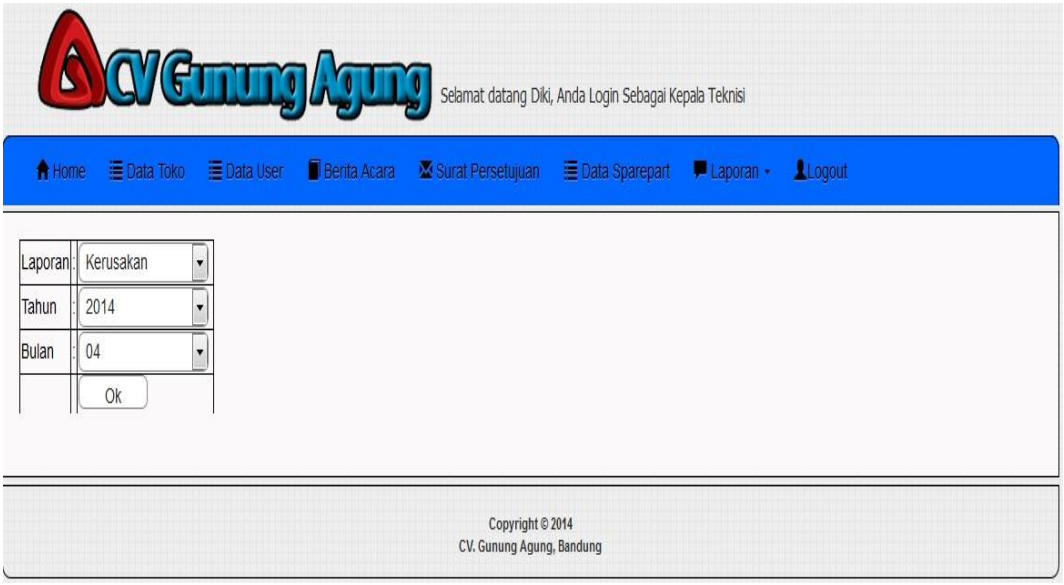
   Tampilan Menu Laporan Kerusakan *Backend* digunakan oleh kepala teknisi setelah memilihmenu laporan kerusakan.



Gambar 4.19 Tampilan Menu Laporan kerusakan

9. Tampilan Menu Cetak Laporan

   Tampilan Menu Cetak Laporan Untuk melihat data Laporan kerusakan dan laporan perbaikan.



Gambar 4.20 Tampilan Menu Cetak Laporan

10. Tampilan Menu Nota

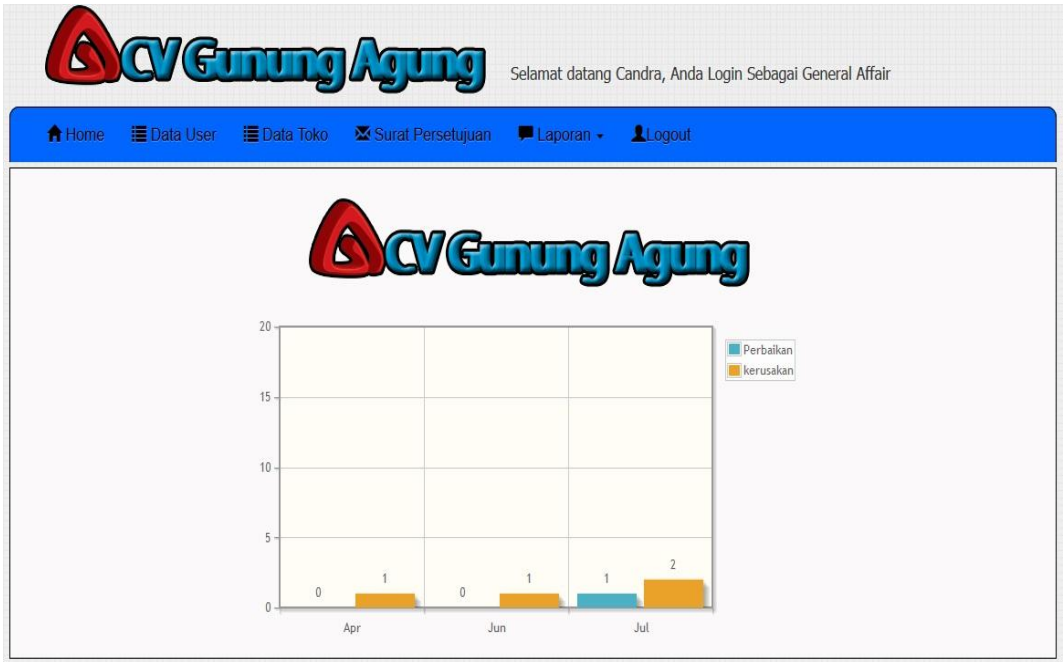
    Tampilan Menu Nota adalah halaman data nota yang telah dilakukan perbaikan



Gambar 4.21 Tampilan Menu Data Nota

11. Tampilan menu utama *Backend General Affair*

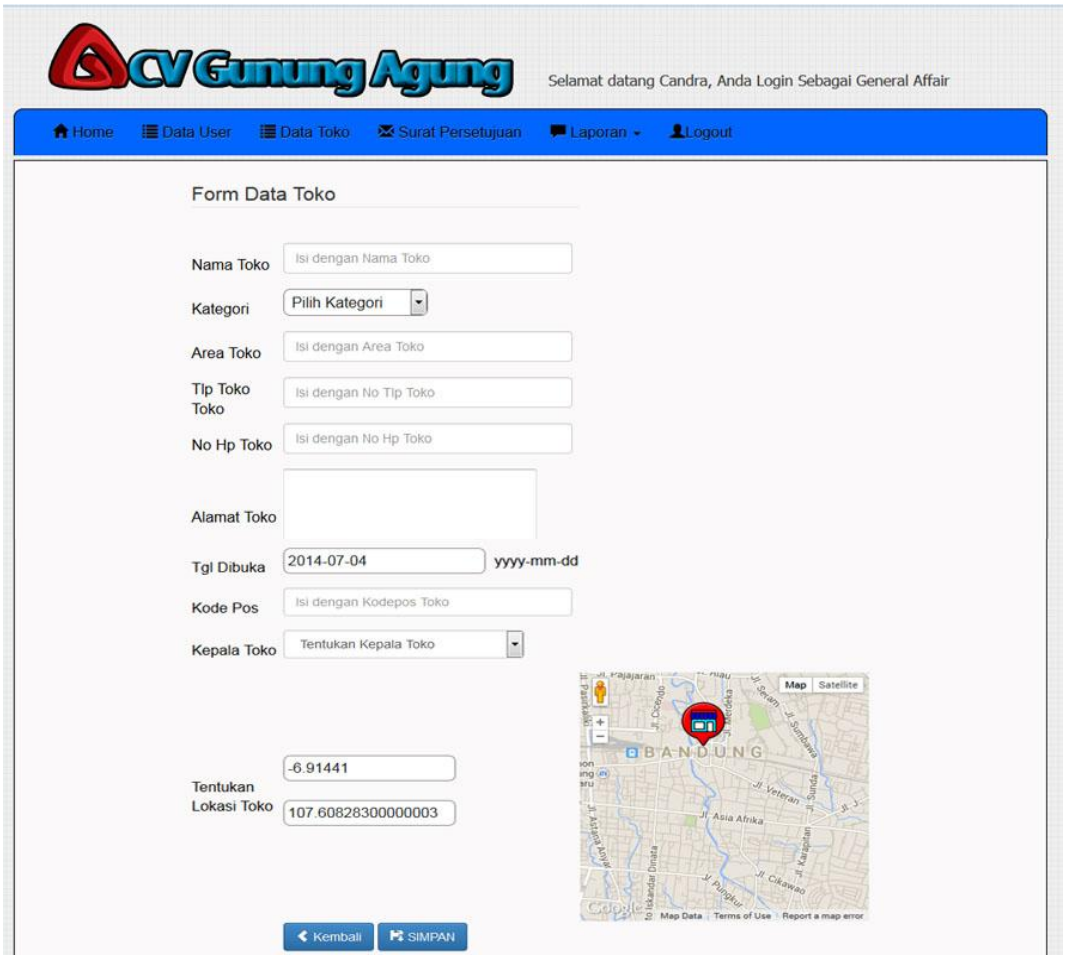
Pada Tampilan Utama *Backend General Affair* terdapat 4 menu utama.



Gambar 4.22 Tampilan menu utama *Backend General Affair*

12. Tampilan Menu Data Toko *General Affair*

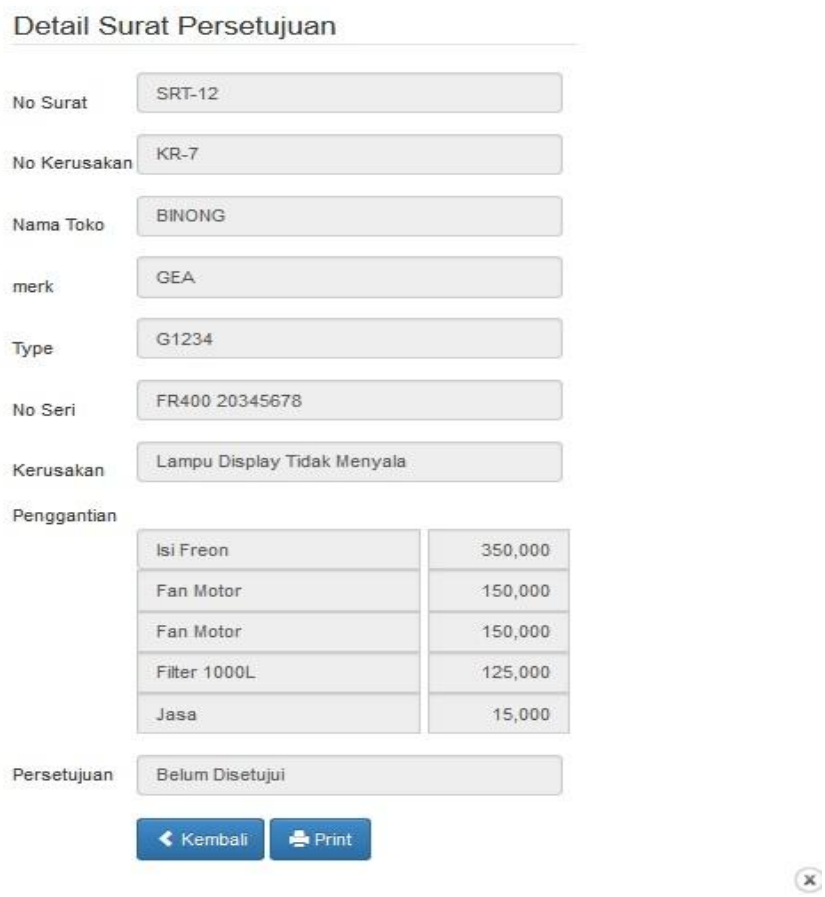
Pada Tampilan menu data toko seorang *General Affair* bisa menambahkan data toko.



Gambar 4.23 Tampilan Menu Form tambah data toko

13. Tampilan Form surat persetujuan

    Tampilan form surat persetujuan seorang *General Affair* hanya dapat merubah status disetujui atau tidak disetujui.



Gambar 4.24 Form Surat Persetujuan

14. Tampil Menu Halaman Utama Kepala Toko

    Tampilan Menu Utama Kepala Toko hanya terdapat 3 menu utama yaitu home, berita acara, dan laporan kerusakan.



Gambar 4.25 Menu utama Kepala Toko

15. Tampil Berita Acara

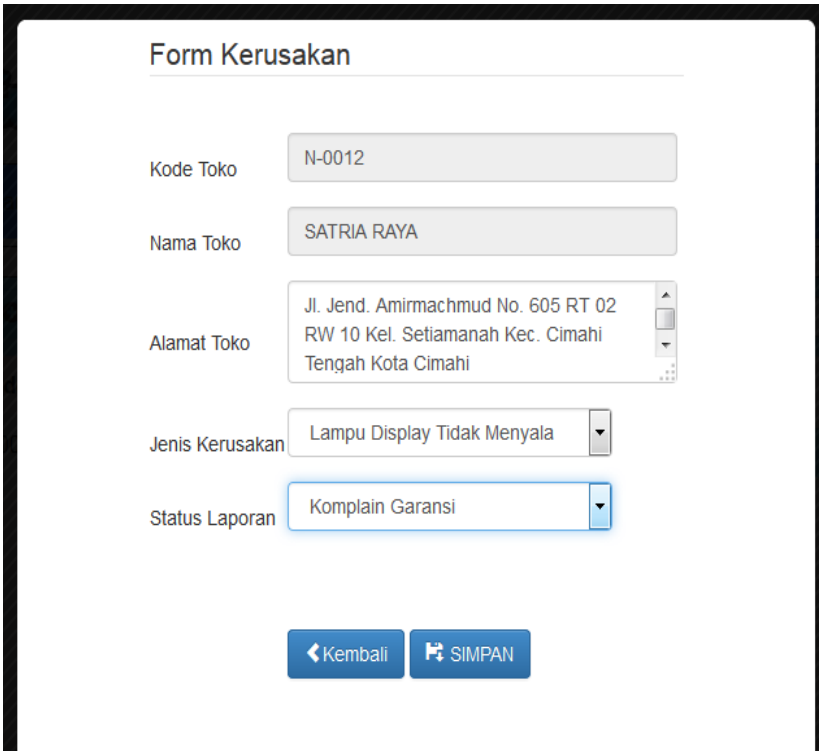
Tampil berita acara merupakan berita acara pemeriksaan baik berita acara perbaikan maupun berita acara kerusakan.



Gambar 4.26 Tampilan Menu Berita Acara

16. Tampil Form Laporan Kerusakan

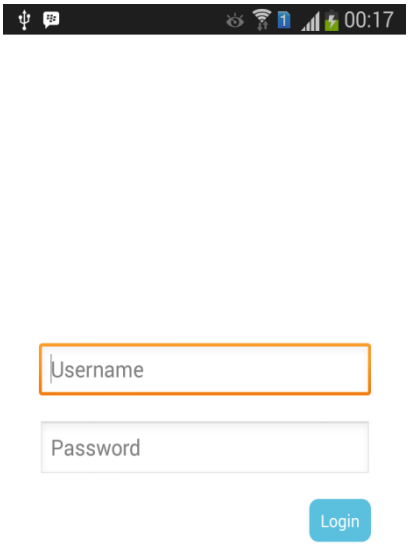
Tampil Form Laporan kerusakan jika kepala toko memilih menu tambah laporan kerusakan.



Gambar 4.27 Form Kerusakan

17. Tampilan Utama Login *Frontend*

    Tampil Halaman utama login *Backend* ketika menjalankan aplikasi *reporting service* android .



Gambar 4.28 Tampil Halaman Login aplikasi *Frontend*

18. Tampilan Utama *Frontend* setelah melakukan login

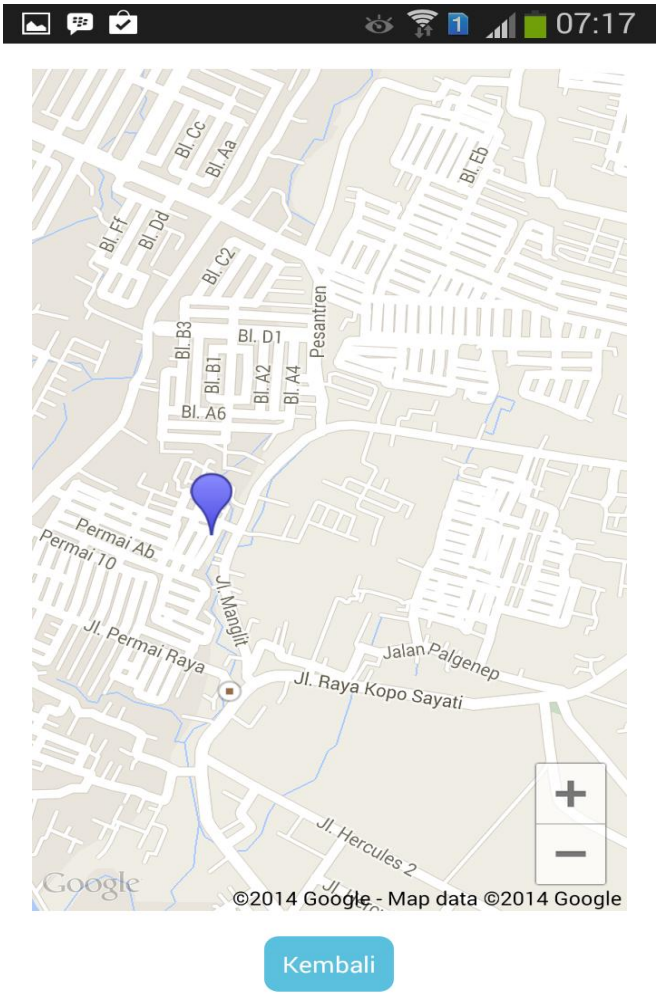
    Tampilan menu utama *Frontend* akan tampil jika *user* telah berhasil login kedalam aplikasi *Frontend*.



Gambar 4.29 Tampilan menu utama aplikasi *Frontend*

19. Tampilan Menu lokasi sekarang

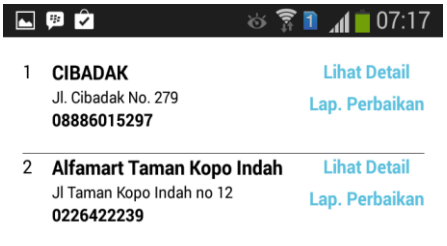
    Tampilan menu lokasi sekarang akan tampil ketika *user* yang telah login kedalam aplikasi *Frontend* memilih menu lokasi sekarang.



Gambar 4.30 Tampil Halaman Lokasi sekarang

20. Tampil menu bagian kerja

    Tampil menu bagian kerja akan tampil ketika *user* memilih menu bagian kerja.



Gambar 4.31 Tampilan menu bagian kerja

21. Tampilan Menu Lihat Detail

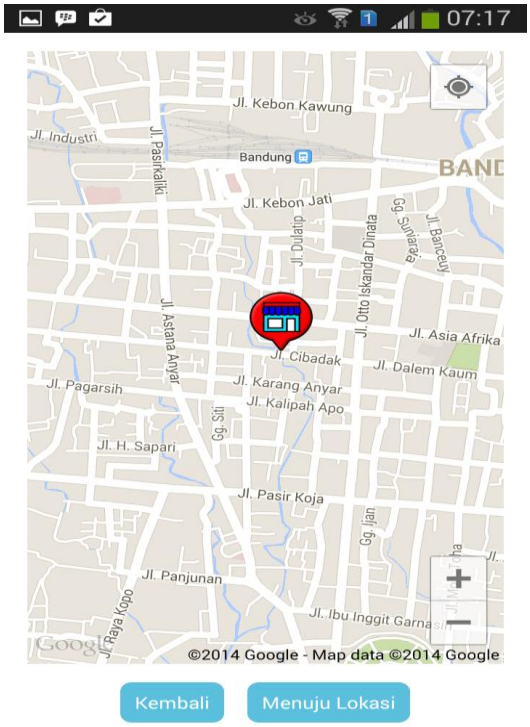
Menu Lihat detail akan tampil ketika *user* yang melakukan login memilih menu lihat detail.



Gambar 4.32 Tampil Menu Lihat detail

21. Tampil menu lihat Lokasi

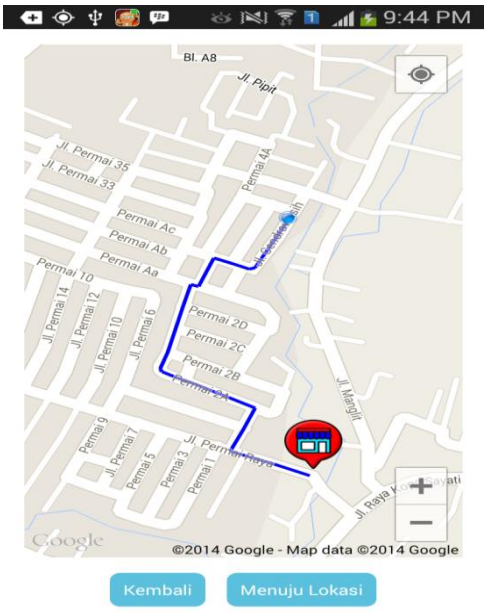
Menu lihat lokasi merupakan bagian dari menu *Frontend* yang berfungsi untuk menampilkan lokasi *user* dan lokasi toko yang mengalami kerusakan.



Gambar 4.33 Tampilan Menu Lihat Lokasi

22. Tampil Menu Menuju Lokasi

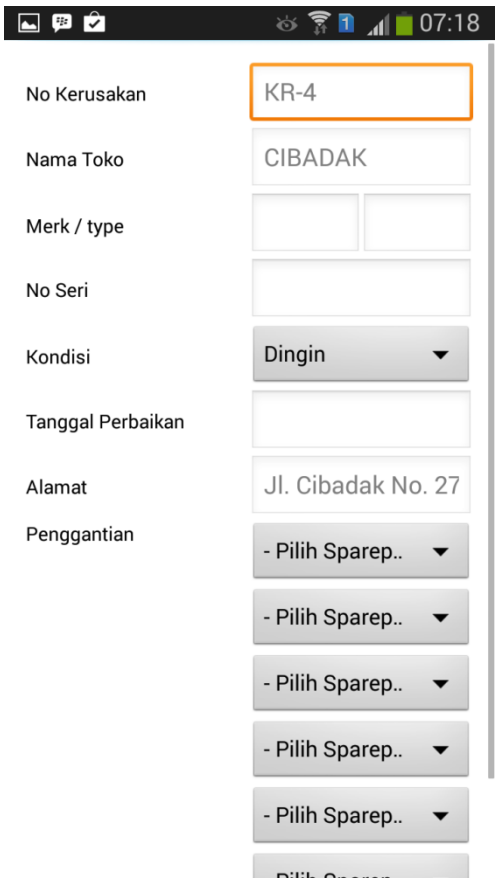
Menu menuju lokasi digunakan oleh teknisi untuk mendapatkan rute tercepat untuk menuju lokasi toko yang mengalami kerusakan.



Gambar 4.34 Rute Menuju Lokasi

23. Menu Laporan Perbaikan

Menu laporan perbaikan akan tampil ketika *user* memilih menu laporan perbaikan pada aplikasi *Frontend.*



Gambar 4.35 Form perbaikan

4.2 Pengujian

Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan kesalahan–kesalahan atau kekurangan–kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut. Adapun beberapa jenis strategi untuk menguji perangkat lunak diantaranya adalah sebagai berikut :

1. Pengujian Modul / Unit

   Pengujian modul / unit dilakukan antar pengembang untuk mencari sebuah kesalahan. pengujian Modul / Unit dalam strategi pengujian perangkat lunak menggunakan metode pengujian *white box.*

2. Pengujian Integrasi

   Pengujian Integrasi dilakukan antara pengembang dengan suatu kelompok independent untuk menguji hasil rancangan perangkat lunak, pengujian Integrasi menggunakan metode pengujian *black box dan white box.*

3. Pengujian Validasi

   Pengujian validasi dilakukan antara pengembang dengan konsumen untuk menguji kesesuaian antara sistem dan requirement , pada pengujian validasi menggunakan metode pengujian *black box.*

4. Pengujian Sistem

   Pengujian Sistem dilakukan untuk menguji perangkat lunak dan elemen sistem lain sebagai kesatuan sistem.

Dalam penelitian dan pembangunan aplikasi *Reporting Service Showcase* pada CV Gunung Agung digunakan pengujian validasi dengan metode pengujian *black box*.

4.2.1 Pengujian Alpha

Pengujian fungsional yang digunakan untuk menguji sistem yang baru adalah metode pengujian alpha. Pengujian alpha berfokus pada persyaratan fungsional perangkat lunak.

4.2.2 Rencana Pengujian

Pengujian dilakukan dengan menguji setiap proses *Use Case* diagram dan kemungkinan kesalahan yang terjadi untuk setiap proses. Pengujian ini dilakukan secara *Black Box,* yaitu dilakukan dengan memperhatikan masukan ke sistem dan keluaran dari sistem. Rencana pengujian yang akan dilakukan pada aplikasi ini selengkapnya terlihat pada tabel 4.4 :

Tabel 4.4 Rencana Pengujian

Menu yang diuji	Detail pengujian	Jenis Uji
Login Backend	Verifikasi data *login Backend user* ke *database* server	*Black box*
Data *user*	Membuat sebuah *user* baru pada sistem	*Black box*
Data Laporan	Menampilkan data laporan kerusakan	*Black box*
Data Nota	Menampilkan dan membuat data nota	*Black box*
Data Toko	Menambah data toko	*Black box*
Data Surat Persetujuan	Menampilkan data surat persetujuan dan menyetujui surat persetujuan	*Black box*
Data Berita Acara	Membuat data berita acara	*Black box*
Bagian Kerja	Mengelola bagian kerja setiap teknisi lapangan	*Black box*
Login *Frontend*	Verifikasi Login *Frontend user* ke *database* server	*Black box*
Lokasi Sekarang	Mengetahui Lokasi Sekarang	*Black box*
Laporan Perbaikan	Membuat data laporan perbaikan	*Black box*

4.2.3 Kasus dan Hasil pengujian

Berdasarkan rencana pengujian, maka dapat dilakukan pengujian alpha pada Aplikasi *Reporting Service Showcase* pada CV Gunung Agung sebagai berikut:

4.2.3.1 Pengujian *Login Backend*

Pengujian login merupakan pengujian fungsionalitas untuk masuk kedalam aplikasi *Backend Reporting Service Showcase* pada CV Gunung Agung

Tabel 4.5 Pengujian Login *Backend*

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Username : Diki *Password*: Diki12	Sistem akan menampilkan halaman utama *Backend*	Pengguna dapat masuk kedalam halaman utama *Backend*	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Username : Diki *Password* : Diki23	Sistem tidak akan menampilkan halaman utama dan sistem akan menampilkan pesan bahwa *username* atau *password* tidak sesuai	Pengguna tidak dapat masuk kedalam halaman utama dan sistem menampilkan pesan bahwa *username* atau *password* tidak sesuai	[√] Diterima [ ] Ditolak

4.2.3.2 Pengujian membuat *User* Baru

Pengujian membuat *user* baru merupakan pengujian fungsional untuk menambah data *user* yang berhak untuk mengakses aplikasi *Reporting Service Showcase* pada CV Gunung Agung.

Tabel 4.6 Pengujian Membuat *User* Baru

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih Menu Data *User* dan Memilih menu Tambah Data *User*	Sistem akan menampilkan halaman Data *User* dan Menampilkan Form *User*	Pengguna dapat masuk kedalam halaman Data *User* dan Form *User*	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**

Memilih Menu Data *User* dan Memilih menu Tambah Data *User*	Data dalam field form tidak akan tersimpan jika tidak terisi dan tidak terisi dengan benar	Muncul Pesan Kesalahan saat akan melakukan penyimpanan data	[√] Diterima [ ] Ditolak

4.2.3.3 Pengujian Data Laporan

Pengujian data laporan berfungsi untuk menguji fungsionalitas menampilkan data kerusakan dan tambah data laporan kerusakan

Tabel 4.7 pengujian Data Laporan

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih Menu laporan kerusakan	Sistem akan menampilkan data laporan kerusakan	Pengguna dapat masuk kedalam halaman Data kerusakan dan form laporan kerusakan	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih Menu laporan kerusakan	Data dalam field form tidak akan tersimpan jika tidak terisi dan tidak terisi dengan benar	Muncul Pesan Kesalahan saat akan melakukan penyimpanan data	[√] Diterima [ ] Ditolak

4.2.3.4 Pengujian Data Nota

Pengujian data nota merupakan pengujian fungsionalitas untuk menampilkan data nota dan pembuatan data nota.

Tabel 4.8 Pengujian Data Nota

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih Menu Nota	Sistem akan menampilkan data Nota	Pengguna dapat masuk kedalam halaman Data Nota dan Form Nota	[√] Diterima [ ] Ditolak

Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih Menu Nota	Jika kode perbaikan tidak di pilih maka rincian total perbaikan tidak dapat di cetak	Muncul Pesan Kesalahan saat akan melakukan penyimpanan data	[√] Diterima [ ] Ditolak

4.2.3.5 Pengujian Data Toko

Pengujian data toko merupakan pengujian fungsionalitas tampil data toko dan fungsionalitas tambah data toko.

Tabel 4.9 Pengujian Data Toko

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih Menu Data Toko	Sistem akan menampilkan data Toko dan form toko	Pengguna dapat masuk kedalam halaman Data Toko dan Form Toko serta menentukan lokasi berdasarkan langtitude dan longtitude	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan Kode Toko yang salah	Data dalam field form tidak akan tersimpan jika tidak terisi dan tidak terisi dengan benar	Muncul Pesan Kesalahan saat akan melakukan penyimpanan data	[√] Diterima [ ] Ditolak

4.2.3.6 Pengujian Surat Persetujuan

Pegnujian surat persetujuan merupakan pengujian fungsionalitas tampil data surat persetujuan dan tambah data surat persetujuan.

Tabel 4.10 Pengujian Surat Persetujuan

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih menu Surat Persetujuan	Sistem akan menampilkan data Surat persetujuan	Pengguna dapat masuk kedalam halaman surat perseujuan	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih menu surat persetujuan	Data dalam field form tidak akan tersimpan jika tidak terisi dan tidak terisi dengan benar	Muncul Pesan Kesalahan saat akan melakukan penyimpanan data	[√] Diterima [ ] Ditolak

4.2.3.7 Pengujian Berita Acara

Pengujian berita acara merupakan pengujian fungsionalitas untuk menampilkan data berita acara dan menambah data berita acara.

Tabel 4.11 Pengujian Berita Acara

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih menu Berita Acara	Sistem akan menampilkan data laporan Berita Acara	Pengguna dapat masuk kedalam halaman Data berita acara dan form laporan kerusakan	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih menu berita acara	Data dalam field form tidak akan tersimpan jika tidak terisi dan tidak terisi dengan benar	Muncul Pesan Kesalahan saat akan melakukan penyimpanan data	[√] Diterima [ ] Ditolak

4.2.3.8 Pengujain Bagian Kerja

Pengujian bagian kerja ini merupakan pengujian fungsionalitas dari tampil data bagian kerja dan menambah data bagian kerja.

Tabel 4.12 Pengujian Bagian Kerja

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih menu Bagian Kerja	Sistem akan menampilkan data laporan kerusakan sesuai *user* yang melakukan login	Pengguna dapat masuk kedalam halaman Detail laporan kerusakan	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih menu Bagian kerja	Data dalam field form tidak akan tersimpan jika tidak terisi dan tidak terisi dengan benar	Muncul Pesan Kesalahan saat akan melakukan penyimpanan data	[√] Diterima [ ] Ditolak

4.2.3.9 Login Frontend

Pengujian Login *Frontend* merupakan pengujian fungsionalitas untuk masuk kedalam sistem *Frontend*.

Tabel 4.13 Login *Frontend*

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Username : ade *Password*: adeade	Sistem akan menampilkan halaman utama *Frontend*	Pengguna dapat masuk kedalam halaman utama *Frontend*	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Username : ade *Password* :	Sistem tidak akan menampilkan halaman utama dan	Pengguna tidak dapat masuk kedalam halaman	[√] Diterima

adesarde	sistem akan menampilkan pesan bahwa *username* atau *password* tidak sesuai	utama dan sistem menampilkan pesan bahwa *username* atau *password* tidak sesuai	[ ] Ditolak

4.2.3.10 Pengujian Data Lokasi

Pengujian data lokasi meruapakan pengujian fungsionalitas untuk mendapatkan lokasi toko yang menagalami kerusakan.

Tabel 4.14 Pengujian Data Lokasi

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih menu lokasi sekarang	Sistem akan menampilkan peta dan posisi teknisi lapangan	Pengguna dapat mlihat data lokasi berada	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih menu lokasi sekarang	Muncul pesan kesalahan jika lokasi tidak ditemukan	Muncul Pesan Kesalahan saat lokasi tidak terjangkau oleh gps	[√] Diterima [ ] Ditolak

4.2.3.11 Pengujian Menu Laporan perbaikan

Pengujian laporan perbaikan merupakan pengujian fungsi dari tampil data laporan perbaikan dan pengolahan data laporan perbaikan.

Tabel 4.15 Pengujian Menu Laporan perbaikan

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih menu Laporan Perbaikan	Sistem akan menampilkan form laporan perbaikan	Pengguna dapat membuat laporan perbaikan	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**

Memilih menu Laporan Perbaikan	Sistem akan menampilkan formlaporan	Muncul Pesan Kesalahan saat akan melakukan penyimpanan data jika form tidak terisi semua.	[√] Diterima [ ] Ditolak

4.2.4 Kesimpulan Hasil Pengujian Alpha

Berdasarkan hasil pengujian alpha yang telah dilakukan dapat ditarik kesimpulan bahwa aplikasi *Reporting Service Showcase* pad acv Gunung Agung yang dibangun sudah berjalan sesuai dengan perancangan awal dan berjalan cukup maksimal, tetapi tidak menutup kemungkinan dapat terjadi kesalahan, pada saat aplikasi digunakan, baik itu kesalahan pada perangkat yang digunakan, kesalahan pengguna, maupun kesalahan-kesalahan lainnya. Sehingga membutuhkan proses perawatan dan pengecekan (*maintenance*) untuk menjaga agar aplikasi tetap berjalan sesuai dengan yang diharapkan.

4.2.5 Pengujian beta

Pengujian beta merupakan pengujian yang dilakukan dengan tujuan untuk mengetahui sejauh mana kualitas sistem pada aplikasi yang dibangun, apakah sudah memenuhi harapan atau belum. Untuk itu dalam pengujian beta dilakukan penelitian terhadap responden atau pengguna sistem dengan melakukan pengumpulan data. Metode yang digunakan dalam penelitian ini adalah metode kualitatif.

Pengujian beta dilakukan pada aplikasi *Frontend* dan *Backend*. Pengujian pada aplikasi *Backend dan Frontend* dilakukan dengan melakukan *interview* terhadap 4 orang yang ditunjuk sebagai pengguna aplikasi.

4.2.5.1 Pengujian Beta Aplikasi *Backend*

Pengujian pada aplikasi *Backend* dilakukan dengan melakukan *interview* / wawancara terhadap 3 orang yang ditunjuk sebagai pengguna. Wawancara ini dilakukan dengan menggunakan teknik kualitatif melalui pertanyaan terbuka. Wawancara mengenai aplikasi *Web* dilakukan terhadap 3 orang yaitu pengguna yang akan mengolah aplikasi *Backend*.

wawancara tersebut akan dilakukan pengambilan kesimpulan terhadap penilaian penerapan sistem yang baru. Wawancara terdiri dari pertanyaan terbuka yang akan ditanyakan kepada narasumber.

Dari hasil wawancara tersebut, menghasilkan jawaban dari narasumber Rincian hasil jawaban yang diperoleh dari wawancara kepada narasumber terdapat pada tabel sebagai berikut :

4.2.5.1.1 Pengujian beta oleh Kepala Teknisi

Kepala teknisi merupakan bagian dari *user* yang menggunakan aplikasi *Reporting Service Showcase* pada halaman *Backend*.

Tabel 4.16 Hasil Pertanyaan Pertama

1. Apakah aplikasi *Reporting Service Showcase* memudahkan bapak dalam menerima laporan kerusakan ?	
Kepala Teknisi CV Gunung Agung	Mudah karena dengan menggunakan aplikasi *Reporting Service Showcase* kita bisa mengetahui langsung setiap ada laporan kerusakan yang masuk.
Kesimpulan	Laporan kerusakan dapat dengan mudah diketahui dan lebih cepat tersampaikan.

Tabel 4.17 Hasil Pertanyaan Kedua

2. Apakah aplikasi *Reporting Service Showcase* memudahkan bapak dalam membagi bagian kerja setiap teknisi?	
Kepala Teknisi CV Gunung Agung	Mudah karena kita bisa menentukan teknisi langsung pada setiap form laporan kerusakan.
Kesimpulan	Mudah dalam pembagian wilayah kerja setiap teknisi

Tabel 4.18 Hasil Pertanyaan Ketiga

3. Apakah aplikasi *Reporting Service Showcase* pada CV Gunung Agung membantu dalam pengolahan data perbaikan?

Kepala Teknisi CV Gunung Agung	Sangat membantu karena kita bisa menerima laporan perbaikan dari teknisi lapangan secara cepat
Kesimpulan	Memudahkan dalam pengolahan data karena lebih efisien dan efektif sehingga lebih cepat dalam mengupdate data terbaru.

Tabel 4.19 Hasil Pertanyaan Keempat

4. Apakah aplikasi *reporting service* membantu bapak dalam pembuatan surat persetujuan perbaikan?	
Kepala Teknisi CV Gunung Agung	Membantu karena cukup dengan memilih menu dan mengisi beberapa kolom dan disimpan maka data sudah dapat dilihat oleh *General Affair* alfamart.
Kesimpulan	Surat persetujuan lebih mudah dibuat dengan hanya mengisi beberapa field

Tabel 4.20 Hasil Pertanyaan Kelima

5. Apakah dengan aplikasi *Reporting Service Showcase* membantu bapak dalam menyampaikan informasi surat persetujuan perbaikan ?	
Kepala Teknisi CV Gunung Agung	Sangat membantu karena data surat persetujuan dapat dilihat setelah data tersimpan dan kita dapat mengetahui status surat persetujuan dengan cepat.
Kesimpulan	Surat persetujuan dapat tersampaikan lebih cepat sehingga dapat ditindaklanjuti lebih cepat.

Tabel 4.21 Hasil Pertanyaan Keenam

6. Apakah dengan aplikasi *Reporting Service Showcase* membantu bapak dalam membuat nota perbaikan?	
Kepala Teknisi CV Gunung Agung	Sangat membantu karena nota perbaikan dapat dicetak berdasarkan laporan perbaikan dari teknisi lapangan.
Kesimpulan	Nota perbaikan dapat dicetak sesuai dengan laporan

	perbaikan.

Tabel 4.22 Hasil pertanyaan Ketujuh

7. Apakah aplikasi *Reporting Service Showcase* memudahkan dalam mengetahui jumlah laporan kerusakan yang sudah ditangani?	
Kepala Teknisi CV Gunung Agung	Mudah karena dengan memlih menu kita dapat mengetahui jumlah laporan kerusakan yang masuk dan jumlah laporan kerusakan yang sudah di perbaiki.
Kesimpulan	Dapat mengetahui laporan perbaikan dan kerusakan berdasarkan bulan.

4.2.5.1.2 Pengujian Beta Oleh Kepala Toko

Kepala Toko merupakan bagian dari *user* yang menggunakan aplikasi *Reporting Service Showcase* pada halaman *Backend*.

Tabel 4.23 Hasil pertanyaan pertama

1. Apakah aplikasi *Reporting Service Showcase* memudahkan dalam penyampaian laporan kerusakan kepada *General Affair*?	
Kepala Toko Alfamart	Mudah karena kita cukup mengisi form laporan kerusakan yang sudah disediakan pada aplikasi *reporting service Showcase*.
Kesimpulan	Informasi laporan kerusakan dapat mudah disampaikan.

Tabel 4.24 Hasil Pertanyaan ke dua

2. Apakah dengan aplikasi *reporting service* memudahkan dalam mengetahui informasi laporan kerusakan yang kita sampaikan disetujui atau tidak ?	
Teknisi Lapangan CV Gunung Agung	Ya sangat membantu, karena dengan aplikasi ini kami dapat mengetahui laporan kerusakan sudah di setujui atau belum.
Kesimpulan	Dapat mudah mengetahui status laporan kerusakan apakah sudah disetujui *General Affair* atau tidak disetujui.

Tabel 4.25 Hasil Pertanyaan ke tiga

3. Apakah aplikasi *Reporting Service Showcase* pada CV Gunung Agung membantu dalam pengolahan data berita acara?	
Teknisi Lapangan CV Gunung Agung	Ya sangat membantu, karena kita tidak bingung dalam format pembuatan berita acara perbaikan dan berita acara kerusakan.
Kesimpulan	Pembuatan berita acara mempunyai format yang sama.

4.2.5.1.3 Pengujian Beta Oleh *General Affair*

General Affair merupakan bagian dari *user* yang menggunakan aplikasi *Reporting Service Showcase* pada halaman *Backend*.

Tabel 4.26 Hasil pertanyaan pertama

1. Apakah aplikasi *Reporting Service Showcase* pada CV Gunung Agung memudahkan bapak dalam menyampaikan informasi kerusakan kepada CV Gunung Agung ?	
General Affair Alfamart	Cukup mudah karena saya bisa menerima laporan kerusakan dari toko dan tinggal merubah kolom status pada form kerusakan maka data akan tersampaikan kepada kepala teknisi CV Gunung Agung.
Kesimpulan	Informasi kerusakan dapat tersampaikan kepada kepala teknisi dengan cepat setelah data disetujui.

Tabel 4.27 Hasil Pertanyaan Kedua

2. Apakah aplikasi *Reporting Service Showcase* pada CV Gunung Agung membantu dalam mengetahui jumlah laporan yang sudah di tangani?	
General Affair Alfamart	Cukup mudah karena saya bisa menerima laporan kerusakan dari toko dan tinggal merubah kolom status pada form kerusakan maka data akan tersampaikan

	kepada kepala teknisi CV Gunung Agung.
Kesimpulan	Informasi hasil perbaikan dapat diketahui kepala teknisi dengan cepat setelah data disetujui.

Tabel 4.28 Hasil pertanyaan ketiga

3. Apakah aplikasi *Reporting Service Showcase* pada CV Gunung Agung membantu dalam mengelola surat persetujuan perbaikan ?	
General Affair Alfamart	Cukup membantu karena dengan aplikasi *Reporting Service Showcase* saya bisa mengetahui surat persetujuan yang masuk dan yang belum disetujui.
Kesimpulan	Informasi laporan kerusakan yang sudah ditangani diketaui berdasarkan data yang ada pada CV Gunung Agung.

Tabel 4.29 Hasil pertanyaan keempat

4. Apakah aplikasi *Reporting Service Showcase* pada CV Gunung Agung membantu bapak dalam melihat rincian perbaikan?	
General Affair Alfamart	Sangat membantu karena dengan memilih menu laporan perbaikan kita bisa melihat detail dari hasil perbaikan *Showcase* yang mengalami kerusakan.
Kesimpulan	Detail laporan hasil perbaikan dapat dilihat sesuai laporan perbaikan yang dipilih.

Tabel 4.30 Hasil pertanyaan ke lima

5. Apakah dengan menggunakan aplikasi *Reporting Service Showcase* pada CV Gunung Agung membantu dalam mengetahui biaya perbaikan?

General Affair Alfamart	Ya membantu karena dengan apikasi *reporting service* kita dapat dengan cepat mengetahui rinican biaya perbaikan berdasarkan masing masing laporan perbaikan.
Kesimpulan	Biaya perbaikan dapat dilihat sesuai laporan perbaikan yang dipilih.

4.2.5.2 Pengujian Beta Aplikasi *Frontend*

Pengujian pada aplikasi *Frontend* dilakukan dengan melakukan *interview* / wawancara terhadap salah seorang teknisi lapangan. Wawancara ini dilakukan dengan menggunakan teknik kualitatif melalui pertanyaan terbuka. Wawancara mengenai aplikasi android dilakukan terhadap 1 orang yaitu pengguna yang akan menggunakan aplikasi *Frontend*.

wawancara tersebut akan dilakukan pengambilan kesimpulan terhadap penilaian penerapan sistem yang baru. Wawancara terdiri dari pertanyaan terbuka yang akan ditanyakan kepada narasumber.

Dari hasil wawancara tersebut, menghasilkan jawaban dari narasumber Rincian hasil jawaban yang diperoleh dari wawancara kepada narasumber terdapat pada tabel sebagai berikut :

4.2.5.2.1 Pengujian Beta Oleh Teknisi Lapangan

Teknisi Lapangan merupakan bagian dari *user* yang menggunakan aplikasi *Reporting Service Showcase* pada halaman *Backend*.

Tabel 4.31 Hasil pertanyaan pertama

1. Apakah aplikasi *Reporting Service Showcase* memudahkan dalam mengetahui jumlah laporan kerusakan yang sudah ditangani?	
Teknisi Lapangan CV Gunung Agung	ya, karena aplikasi *Reporting Service Showcase* dapat menyimpan data laporan hasil perbaikan secara tepat dan cepat kepada kepala teknisi
Kesimpulan	Dapat mengetahui laporan perbaikan dan kerusakan

	berdasarkan bulan.

Tabel 4.32 Hasil Pertanyaan ke dua

2. Apakah aplikasi *Reporting Service Showcase* pada CV Gunung Agung memudahkan dalam pencarian lokasi toko yang mengalami kerusakan?	
Teknisi Lapangan CV Gunung Agung	Mudah karena setiap laporan kerusakan yang diterima terdapat menu untuk menuju lokasi kerusakan.
Kesimpulan	Dapat mengetahui laporan perbaikan dan kerusakan berdasarkan bulan.

4.3 Kesimpulan pengujian Beta

Berdasarkan hasil kesimpulan jawaban setiap narasumber terhadap pertanyaan yang diajukan pada pengujian beta dapat ditarik kesimpulan bahwa sistem yang dibangun sudah *user friendly*, mudah dioperasikan, dapat memberikan kemudahan bagi responden dalam mengatasi kesulitan membuat laporan kerusakan , membuat laporan perbaikan, membuat berita acara, membuat surat persetujuan, pencarian lokasi toko, mengelola data toko, mengelola laporan perbaikan serta pencarian lokasi toko.