BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Tahap implementasi sistem merupakan tahap menterjemahkan perancangan berdasarkan hasil analisis ke dalam bahasa yang dapat dimengerti oleh mesin, serta penerapan perangkat lunak pada keadaan yang sesungguhnya. Sesuai kode program yang digunakan dalam pembuatan Aplikasi Manajemen *Hardware Change Request* (HCR) di PT Indosat Tbk.

4.1.1 Perangkat Lunak Pendukung

Perangkat lunak yang digunakan pada sistem komputer dalam membangun Aplikasi Manajemen *Hardware Change Request* (HCR) di PT. Indosat Tbk ini adalah sebagai berikut:

1. Sistem Operasi *Windows 7*.
2. *Adobe Dreamweaver CS3* dan *Notepad++* untuk penulisan kode program.
3. *Xampp* untuk *local* internet dan penyimpanan *database*.
4. *Mozilla firefox atau Google Chrome* sebagai *browser*.

4.1.2 Kebutuhan Perangkat Keras

Kebutuhan minimum perangkat keras (*hardware*) yang diperlukan untuk mengimplementasikan program aplikasi yang dibuat adalah perangkat keras komputer PC dengan spesifikasi yang akan dijabarkan berikut. Semakin tinggi

spesifikasi komputer yang digunakan untuk menjalankan aplikasi, akan semakin baik. Kebutuhan spesifikasi minimumnya yaitu :

1. Prosessor Pentium IV 3.0 Ghz
2. RAM 1 GB
3. Harddisk dengan ruang kosong 2 GB
4. Monitor dengan resolusi 1280 x 800 pixels

4.1.3 Implementasi Basis Data

Implementasi basis data diambil berdasarkan perancangan basis data yang dibuat sebelumnya, Secara fisik, implementasi basis data diimplementasikan menggunakan perangkat lunak MySQL. Implementasi *database* dalam bahasa SQL adalah sebagai berikut:

1. Pembuatan Database

Tabel 4.1 Database Dbapphcr

Pembuatan Database

`DROP DATABASE IF EXISTS `dbapphcr`;`
`  CREATE DATABASE IF NOT EXISTS `dbapphcr`;`
`USE `dbapphcr`;`

Tabel 4.2 Tabel Branch

Tabel branch

`CREATE TABLE `tbbranch` (`
`  `branchid` char(3) NOT NULL,`
`  `area` varchar(30) NOT NULL,`
`  `branch` varchar(30) NOT NULL,`
`  `datecreated` datetime NOT NULL,`
`  `dateedit` datetime NOT NULL,`
`  `del` char(1) NOT NULL,`

```
  PRIMARY KEY (`branchid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabel 4.3 Tabel Level

Tabel level

```
CREATE TABLE `tblevel` (
  `levelid` char(2) NOT NULL,
  `level` varchar(30) NOT NULL,
  PRIMARY KEY (`levelid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabel 4.4 Tabel User

Tabel user

```
CREATE TABLE `tbuser` (
  `nik` varchar(12) NOT NULL,
  `name` varchar(40) NOT NULL,
  `uname` varchar(40) NOT NULL,
  `pass` varchar(50) NOT NULL,
  `email` varchar(35) NOT NULL,
  `phone` varchar(30) NOT NULL,
  `levelid` char(2) NOT NULL,
  `branchid` char(3) NOT NULL,
  `datecreated` datetime NOT NULL,
  `dateedit` datetime NOT NULL,
  `del` char(1) NOT NULL,
  PRIMARY KEY (`nik`),
  CONSTRAINT `FK_tbuser1`
  FOREIGN KEY (`levelid`) REFERENCES
 `tblevel` (`levelid `) ON DELETE CASCADE
  ON UPDATE CASCADE,
  CONSTRAINT `FK_tbuser2`
  FOREIGN KEY (`branchid`) REFERENCES
 `tbbranch` (`branchid`) ON DELETE CASCADE
```

```
    ON UPDATE CASCADE
 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabel 4.5 Tabel Site

Tabel site
CREATE TABLE `tbsite` (`siteid` varchar(8) NOT NULL, `sitename` varchar(40) NOT NULL, `branchid` char(3) NOT NULL, `teritori` varchar(30) NOT NULL, `kab` varchar(30) NOT NULL, `address` varchar(50) NOT NULL, `system` enum('2G','3G') NOT NULL, `band` enum('GSM','DCS','INDOOR GSM','INDOOR DCS', 'DCS COLO GSM') NOT NULL, `vendor` enum('Nokia','Ericcson') NOT NULL, `longi` varchar(20) NOT NULL, `lati` varchar(20) NOT NULL, `statusbts` varchar(20) NOT NULL, `datecreated` datetime NOT NULL, `dateedit` datetime NOT NULL, `del` char(1) NOT NULL, PRIMARY KEY (`siteid`), CONSTRAINT `FK_tbsite1` FOREIGN KEY (`branchid`) REFERENCES `tbbranch` (`branchid`) ON DELETE CASCADE ON UPDATE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=latin1;

Tabel 4.6 Tabel Ticket

Tabel ticket
CREATE TABLE `tbticket` (

```
  `noticket` varchar(30) NOT NULL,
  `picreq` varchar(40) NOT NULL,
  `datereq` date NOT NULL,
  `weekreq` int(2) NOT NULL,
  `siteid` varchar(8) NOT NULL,
  `problem` text NOT NULL,
  `alarm` varchar(40) NOT NULL,
  `hw` varchar(30) NOT NULL,
  `severity` enum('Critical','Urgent','Normal') NOT
NULL,
  `tim` enum('Quality','Access') NOT NULL,
  `status` enum('Open','Closed','Follow-up','Done')
NOT NULL,
  `remark` text NOT NULL,
  `datecreated` datetime NOT NULL,
  `createdby` varchar(40) NOT NULL,
  `dateedit` datetime NOT NULL,
  `editby` varchar(40) NOT NULL,
  `del` char(1) NOT NULL,
  PRIMARY KEY (`noticket`),
 CONSTRAINT `FK_tbticket1 `
 FOREIGN KEY (`siteid`) REFERENCES
 `tbsite` (`siteid`) ON DELETE CASCADE
 ON UPDATE CASCADE
 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabel 4.7 Tabel Action

Tabel action

```
CREATE TABLE `tbaction` (
  `actionid` int(5) NOT NULL AUTO_INCREMENT,
  `noticket` varchar(25) NOT NULL,
  `picact` varchar(40) NOT NULL,
  `dateact` date NOT NULL,
```

```
  `weekact` int(2) NOT NULL,
  `presentase` int(3) NOT NULL,
  `descact` text NOT NULL,
  `datecreated` datetime NOT NULL,
  `createdby` varchar(40) NOT NULL,
  `dateedit` datetime NOT NULL,
  `editby` varchar(40) NOT NULL,
  PRIMARY KEY (`actionid`),
  CONSTRAINT `FK_tbaction1`
  FOREIGN KEY (`noticket`) REFERENCES
  `tbticket` (`noticket`) ON DELETE CASCADE
  ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
AUTO_INCREMENT=10;
```

Tabel 4.8 View Vticket

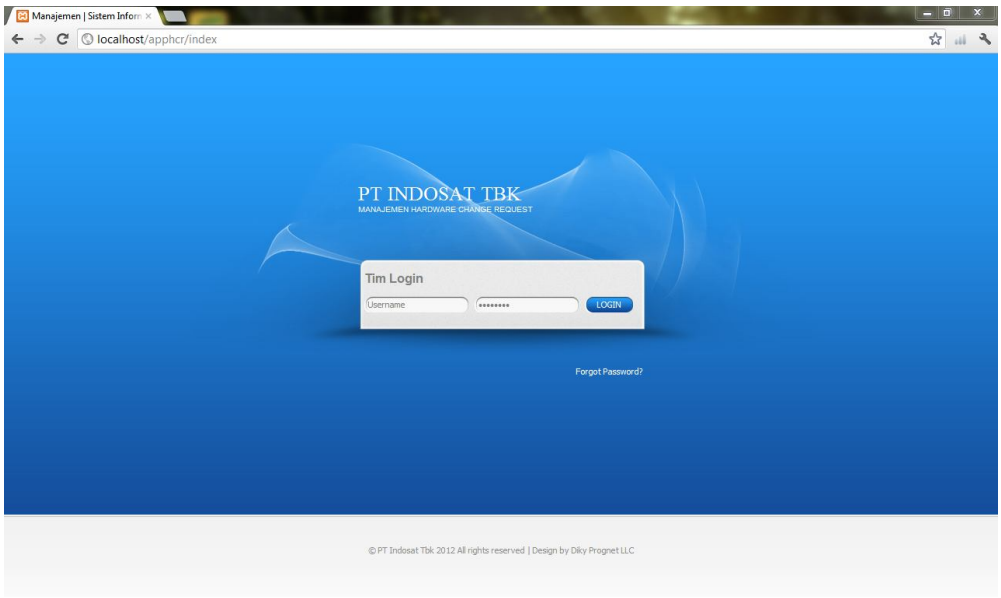
View vticket

```
CREATE OR REPLACE VIEW `vticket` (
 SELECT `tbticket`.`noticket` AS `noticket`,
 (SELECT (SELECT `tbbranch`.`branch`
 AS `branch` from `tbbranch` WHERE
 (`tbbranch`.`branchid` = `tbsite`.`branchid`)) AS
 `branch` FROM `tbsite`  where (`tbsite`.`siteid` =
 `tbticket`.`siteid`)) AS `branch`,
 date_format(`tbticket`.`datereq`,'%d/%m/%y') AS `dq`,
 (select `tbsite`.`sitename` AS `sitename`
 FROM `tbsite` WHERE (`tbsite`.`siteid` =
 `tbticket`.`siteid`)) AS `sitename`,
 `tbticket`.`severity` AS `severity`,
 `tbticket`.`tim` AS `tim`,
 `tbticket`.`status` AS `status`
 FROM `tbticket` WHERE (`tbticket`.`del` = 1)
 ORDER BY datereq DESC );
```

4.1.4 Implementasi Antarmuka

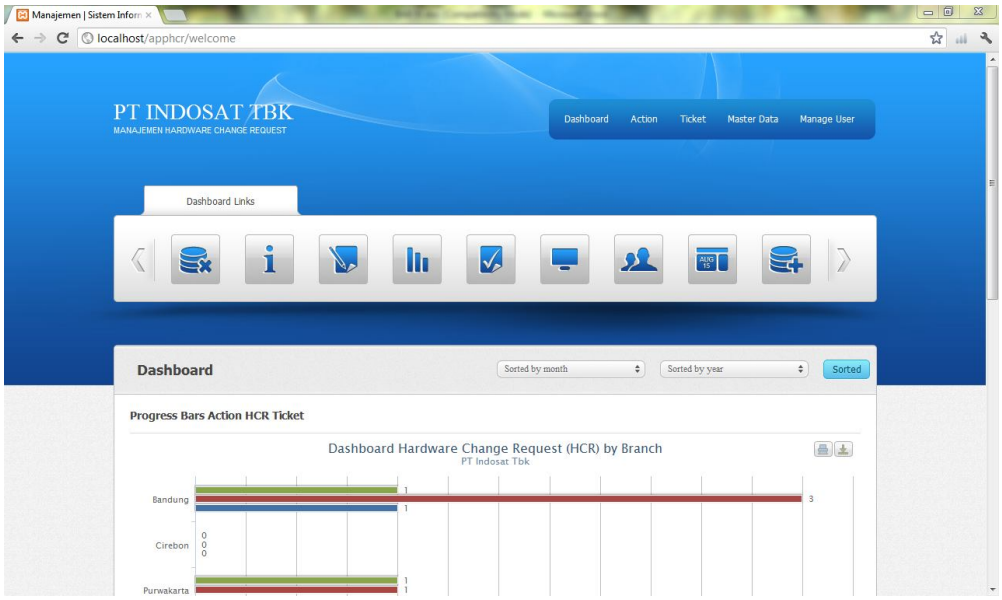
Implementasi antarmuka dilakukan dengan setiap halaman web yang dibuat. Implementasi antarmuka yang dibuat untuk semua pengguna, yaitu *administrator*, *manager*, tim *project*, tim *quality*, dan tim *access* (dapat dilihat pada gambar 4.1 - 4.8). Untuk sisanya dapat dilihat dilampiran antar muka.

1. Halaman Login



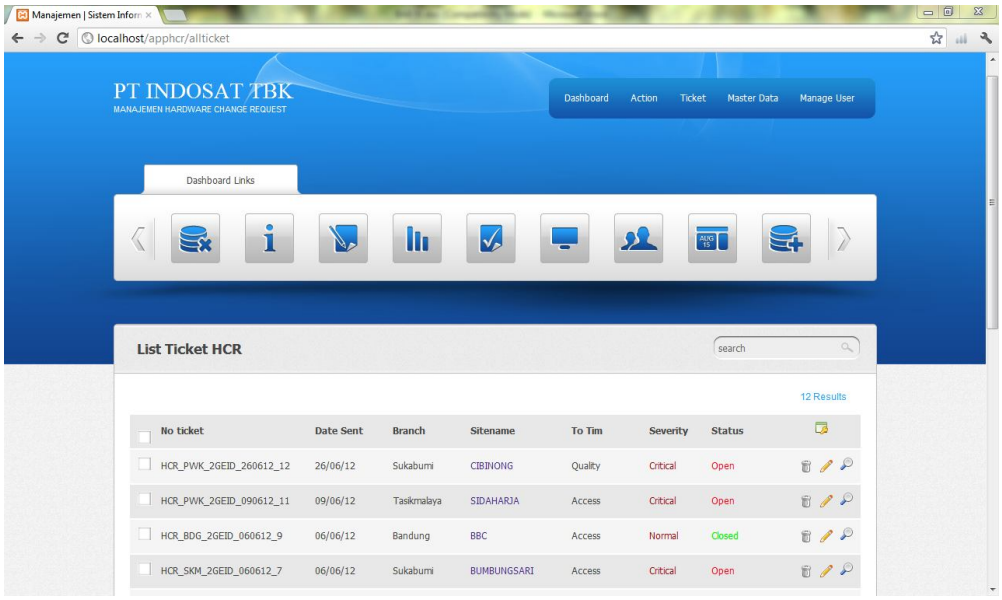
Gambar 4.1 Halaman Login

2. Halaman Utama



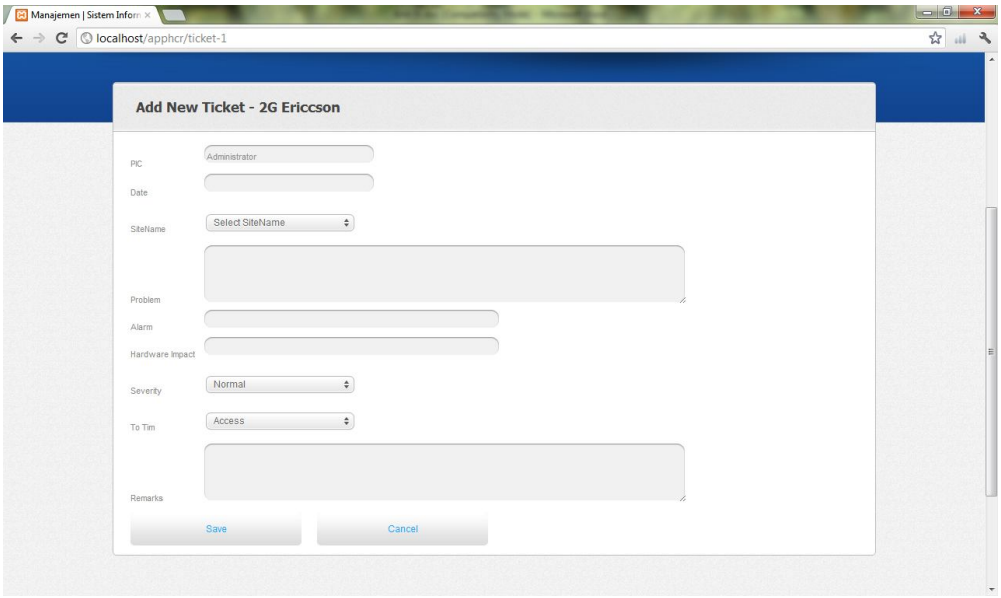
Gambar 4.2 Halaman Utama

3. Halaman List *Ticket*



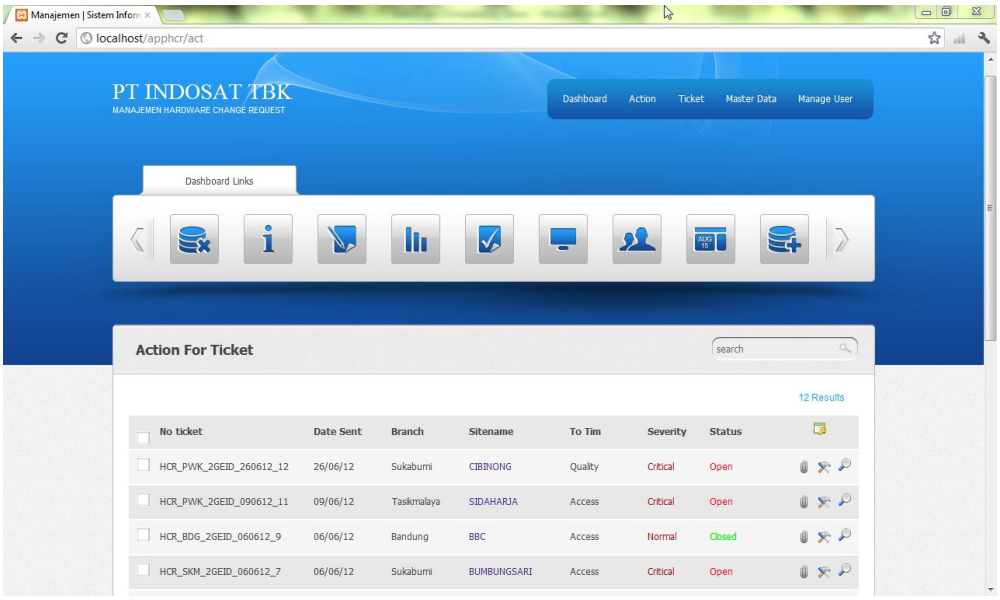
Gambar 4.3 Halaman List *Ticket*

4. Halaman Tambah *Ticket*



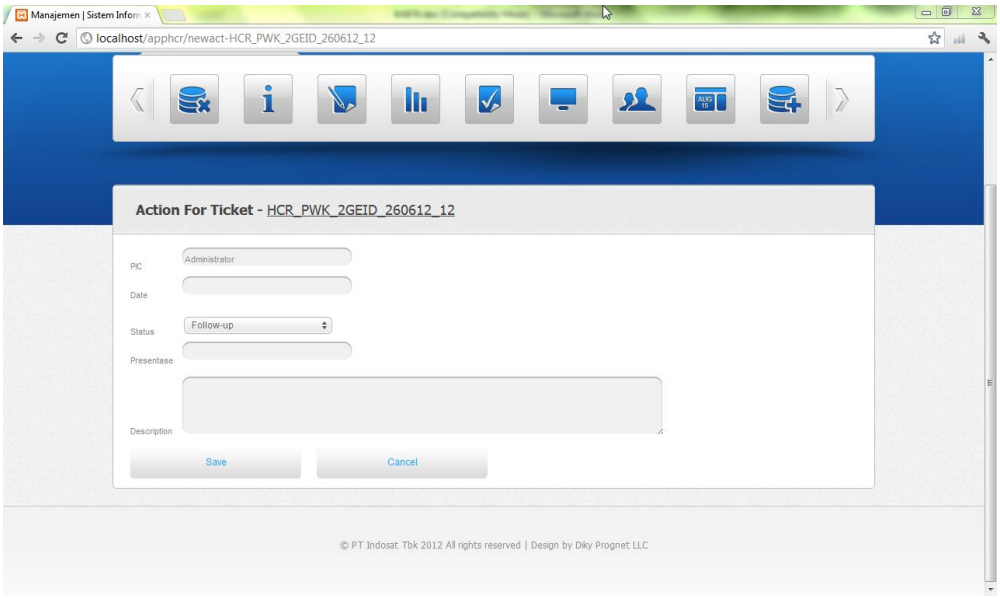
Gambar 4.4 Halaman Tambah *Ticket*

5. Halaman List *Action*



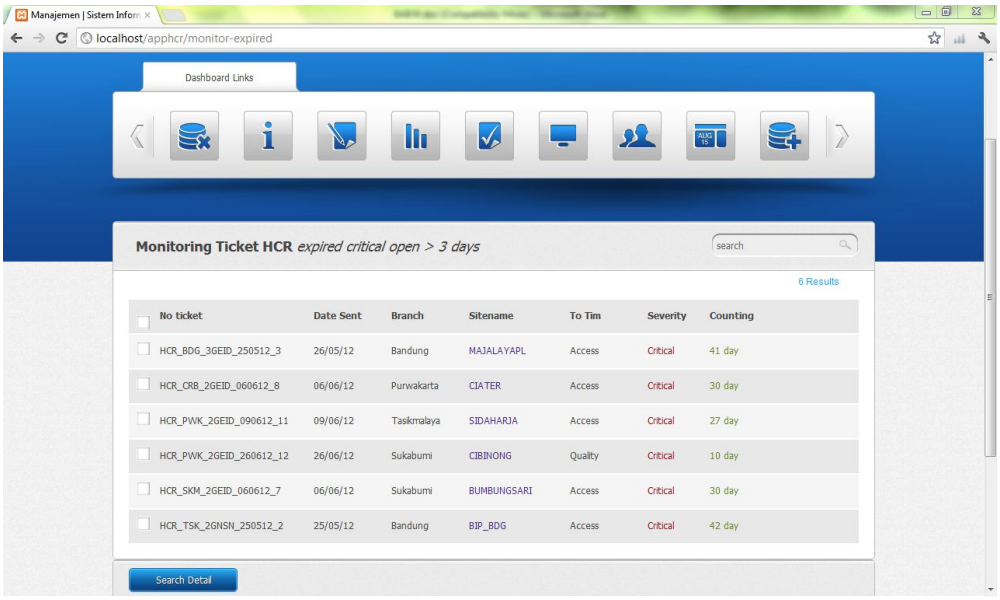
Gambar 4.5 Halaman List *Action*

6. Halaman Tambah *Action*



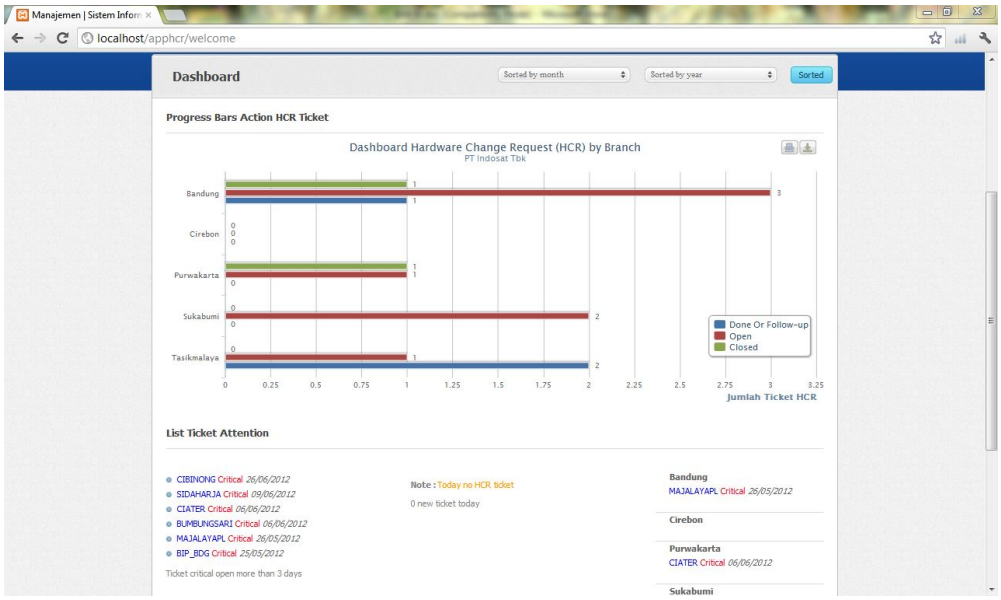
Gambar 4.6 Halaman Tambah *Action*

7. Halaman Monitoring Ticket



Gambar 4.7 Halaman Monitoring Ticket

8. Halaman Dashboard



Gambar 4.8 Halaman Dashboard

4.2 Pengujian Sistem

Tahap pengujian sistem (*testing*) merupakan tahap yang menentukan apakah hasil akhir sebuah sistem memenuhi *requirement* yang telah ditentukan. Pengujian dibuat berdasarkan analisa dan perancangan yang telah dilakukan sebelumnya. Dalam tahap Pengujian ini akan ditentukan apakah sistem telah memenuhi *requirement-requirement* yang dijadikan sebagai tolak ukur dalam menentukan keberhasilan dalam pengerjaan sebuah sistem perangkat lunak. Untuk membangun aplikasi manajemen *Hardware Change Request* (HCR) di PT Indosat Tbk akan ditentukan proses pengujian.

Pengujian yang dilakukan yaitu pengujian secara fungsional *(alpha)* dan beta. Metode pengujian *alpha* yang digunakan adalah pengujian *blackbox* yang berfokus pada persyaratan fungsional dari sistem yang dibangun. Pengujian ini

memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya semua persyaratan fungsional untuk suatu program.

4.2.1 Rencana Pengujian Alpha

Rencana pengujian yang akan dilakukan dengan menguji sistem secara alpha dan beta.

Tabel 4.9 Tabel Rencana Pengujian Alpha

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Verifikasi Login	*Black box*
Pengolahan Data User	Tambah Data User	*Black box*
	Ubah Data User	*Black box*
	Hapus Data User	*Black Box*
	Cari Data User	*Black Box*
Pengolahan Data Branch	Tambah Data Branch	*Black Box*
	Ubah Data Branch	*Black box*
	Hapus Data Branch	*Black Box*
	Cari Data Branch	*Black Box*
Pengolahan Data Site	Tambah Data Branch	*Black Box*
	Ubah Data Branch	*Black box*
	Hapus Data Branch	*Black Box*
	Cari Data Branch	*Black Box*
Pengolahan Ticket	Tambah Ticket	*Black Box*
	Ubah Ticket	*Black box*
	Hapus Ticket	*Black Box*
	Cari Ticket	*Black Box*
Pengolahan Action	Tambah Action	*Black Box*
	Ubah Action	*Black box*

Konfirmasi Ticket	Konfirmasi Closed Ticket	*Black Box*
Pencarian Monitoring	Berdasarkan Date	*Black Box*
	Berdasarkan Tim	*Black Box*
	Berdasarkan Kriteria	*Black Box*
Laporan	Laporan HCR	*Black Box*
Pencarian Dashboard	Bulan atau Tahun Dashboard	*Black Box*
Perubahan Password	Ubah Password	*Black Box*

4.2.2 Kasus Dan Hasil Pengujian *Alpha*

Pengujian *alpha* merupakan pengujian yang menitikberatkan pada hasil *output* dari kendali *input* yang dimasukkan pada tampilan kendali *input form* tampilan, dimana pengujian dikatakan berhasil apabila *output* sesuai dengan kendali *input* yang dimasukkan pada setiap tampilan. Berdasarakan rencana pengujian yang telah disusun, maka dapat dilakukan pengujian sebagai berikut.

1. Login

Tabel 4.10 Pengujian Verifikasi Login

Kasus dan Hasil Uji (Data Normal)	
Data Masukan	*Username :* admin dan p*assword :* admin
Yang diharapkan	*Username* dan *password* yang dimasukkan benar sehingga dapat masuk ke halaman utama sesuai hak akses user
Pengamatan	*Username* dan *password* yang dimasukkan diterima, tampilan halaman utama sesuai data login
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	*Username :* salah dan *password :* salah
Yang diharapkan	Setelah klik login maka login gagal, tidak masuk ke sistem

	dan muncul pesan *username, password* tidak terindentifikasi
Pengamatan	Login ke sistem gagal dan pesan muncul sesuai dengan yang diharapkan
Kesimpulan	Diterima
Kasus dan Hasil Uji (Username atau Password Kosong)	
Data Masukan	*Username :* dikosongkan atau *Password :* dikosongkan
Yang diharapkan	Login ke sistem gagal dan muncul pesan *username* atau *password* tidak diperbolehkan kosong
Pengamatan	Login ke sistem gagal dan muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima

2. Pengolahan Data *User*

Tabel 4.11 Pengujian Pengolahan Data *User*

Tambah Data *User*	
Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Mengisi atribut data *user* yang disediakan dengan data yang benar. Contoh : *NIK*: 70943880, *Name*: Deni Sopiana, *Username*: sopiana, *Password*: manager123, *Confrim Password:* manager123, *Phone*: 0815720048001, *Email*: deni.sopiana@indosat.com, *Level*: Manager, B*ranch*: Bandung
Yang diharapkan	Semua data yang dimasukkan benar, data tersimpan dan muncul informasi penambahan data berhasil ditambahkan. *User* yang ditambahkan dapat melakukan login ke sistem sesuai hak akses
Pengamatan	Semua data yang dimasukkan diterima, penambahan *user* berhasil. *User* baru dapat melakukan login sesuai data *user*
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	

Data Masukan	Mengisi atribut data *user* yang disediakan dengan data yang salah atau sama dengan data sebelumnya. Contoh: NIK : 70943880 sama dengan nik yang sebelumnya
Yang diharapkan	Proses penambahan data gagal, muncul informasi penambahan data *user* gagal
Pengamatan	Muncul informasi penambahan data *user* gagal, data *user* tidak jadi ditambahkan
Kesimpulan	Diterima
Ubah Data *User*	
Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Mengubah atribut data *user* dengan data yang benar. Contoh: *Name* : Deni Sopiana, *Username* : deni, *Password* : manager123, *Confrim Password :* manager123, *Phone* : 081572000500, *Email* : deni.sopiana@indosat.com, *Level* : Tim Project, B*ranch* : Cirebon
Yang diharapkan	Semua data yang dimasukkan benar, data ubah tersimpan dan muncul pesan merubah data *user* berhasil. Data login baru sesuai perubahan data yang dilakukan
Pengamatan	Semua data yang dimasukkan diterima, data di *database* berubah, ditampilkan pesan sesuai yang diharapkan
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	Mengubah atribut data *user* yang disediakan dengan data kosong, Contoh : *Name*: dikosongkan
Yang diharapkan	Proses perubahan data gagal, ditampilkan pesan kesalahan data yang dikosongkan
Pengamatan	Data *user* tidak berubah, muncul pesan kesalahan sesuai yang diharapkan
Kesimpulan	Diterima
Hapus Data *User*	

Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Pilih hapus user atau pilih beberapa user lalu klik tombol *delete checked* user. Contoh : *NIK* : 70943880
Yang diharapkan	Muncul pesan konfirmasi, pilih yakin hapus data *user*. Dan dilakukan pengecekan user sedang aktif login tidak jika masih login penghapusan gagal, tetapi jika tidak *user* dapat dirubah statusnya menjadi terhapus di *database* dan tidak dapat login lagi kesistem
Pengamatan	Muncul pesan konfirmasi, pilih yakin hapus data *user*. Dan dilakukan pengecekan user sedang aktif login tidak jika masih login penghapusan gagal, tetapi jika tidak *user* dapat dirubah statusnya menjadi terhapus di *database* dan tidak dapat login lagi ke sistem
Kesimpulan	Diterima
Cari Data *User*	
Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Masukkan kata kunci pencarian, Contoh : Bandung
Yang diharapkan	Pencarian sesuai kata kunci, kemudian data hasil pencarian ditampilkan
Pengamatan	Pencarian sesuai kata kunci pada *database* telah berhasil ditemukan, kemudian ditampilkan hasilnya
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	Masukkan kata kunci yang tidak sesuai, Contoh : Edi
Yang diharapkan	Mencari data sesuai kata kunci, jika data hasil pencarian tidak ditemukan maka tidak ada yang ditampilkan
Pengamatan	Data yang dengan kata kunci : Edi tidak ditemukan, data yang ditampilkan tidak ada
Kesimpulan	Diterima

3. Pengolahan Data *Branch*

Tabel 4.12 Pengujian Pengolahan Data *Branch*

Tambah Data *Branch*	
Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Mengisi atribut data branch yang disediakan dengan data yang benar. Contoh: *Branch id* : BDG, *Area* : West java, *Branch* : Bandung
Yang diharapkan	Semua data yang dimasukkan benar, data *branch* yang baru tersimpan di *database,* dan muncul pesan data branch berhasil ditambahkan
Pengamatan	Semua data yang dimasukkan benar, penambahan *branch* berhasil dan tersimpan di *database,* dan muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	Mengisi atribut data branch yang disediakan dengan data *branch id* sama. Contoh: *Branch id* : BDG dimana data dengan *branch id* tersebut sudah ada
Yang diharapkan	Muncul pesan proses penambahan *branch* gagal, data tidak tersimpan di *database* dan *branch* tidak ada penambahan
Pengamatan	Muncul pesan sesuai yang diharapkan, data tidak tersimpan di *database*, dan *branch* tidak ada penambahan
Kesimpulan	Diterima
Ubah Data *Branch*	
Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Mengubah atribut data *user* yang sudah ada dalam *database* dengan data yang benar Contoh : *Area* : *West Java*, *Branch* : Sukabumi
Yang diharapkan	Semua data yang dimasukkan benar, data ubah *branch* tersimpan dan muncul pesan merubah data *branch* berhasil

Pengamatan	Perubahan data *branch* tersimpan dan muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	Mengosongkan atribut data *branch* yang disediakan. Contoh: *Branch* : dikosongkan
Yang diharapkan	Data *branch* tidak mengalami perubahan dan muncul pesan sesuai atribut yang dikosongkan
Pengamatan	Data *branch* tidak berubah dan muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima
Hapus Data *Branch*	
Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Pilih hapus *branch* atau pilih beberapa *branch* lalu klik *delete checked branch.* Contoh: *Branch:* Bandung dengan *Branchid* : BDG
Yang diharapkan	Melakukan pengecekan keterhubungan data *branch* dengan data yang lain semisal *user* dan site. Jika masih ada keterhubungan penghapusan tidak bisa dilakukan. Muncul pesan hapus *branch* gagal. Tetapi jika data tidak ada keterhubungan makan *branch* dapat dirubah statusnya menjadi terhapus pada *database*
Pengamatan	Muncul pesan konfirmasi, klik yakin hapus branch. Melakukan pengecekan keterhubungan data *branch* dengan data yang lain semisal *user* dan site. Jika masih ada keterhubungan penghapusan tidak bisa dilakukan. Muncul pesan hapus *branch* gagal. Tetapi sebaliknya *branch* berhasil dirubah statusnya menjadi terhapus
Kesimpulan	Diterima
Cari Data *Branch*	

Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Masukkan kata kunci pencarian Contoh : CRB
Yang diharapkan	Mencari data sesuai kata kunci, kemudian data yang sesuai ditampilkan
Pengamatan	Data pencarian sesuai kata kunci ditampilkan
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	Masukkan kata kunci yang tidak sesuai Contoh : SMD
Yang diharapkan	Mencari data yang sesuai kata kunci, jika data yang dimaksud tidak ditemukan maka tidak akan ada yang ditampilkan
Pengamatan	Data dengan kata kunci tidak ditemukan yang sesuai, hasil pencarian data tidak ada yang ditampilkan
Kesimpulan	Diterima

4. Pengolahan Data Site

Tabel 4.13 Pengujian Pengolahan Data Site

Tambah Data Site	
Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Mengisi atribut data site yang disediakan dengan data yang benar. Contoh: *SiteId*: 101PWK02, *SiteName*: Cianting_Timur, *Branch* : Purwakarta, *Teritori* : Purwakarta Utara, *Kab* : Purwakarta, *Address* : Desa Cibodas Kec Sepatani, *System* : 2G, *Band* : GSM, *Vendor* : Nokia, *Longitude* : 107.415, *Latitude* : -6.6344, *Status* : New Existing
Yang diharapkan	Dilakukan pengecekan adakah kesamaan dengan *site id* dengan yang sudah ada. Jika tidak maka site data bisa ditambahkan. Pemasukan data berhasil, data yang baru

	tersimpan di *database.* Muncul pesan site berhasil ditambahkan
Pengamatan	Semua data yang dimasukkan diterima, penambahan site berhasil dan tersimpan di *database,* dan muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	Mengisi atribut data site yang disediakan dengan data yang salah atau sama dengan data sebelumnya. Contoh: *Siteid*: 101PWK02 digunakan kembali
Yang diharapkan	Penambahan site tidak bisa ditambahkan karena siteid baru sudah terdaftar di *database*, muncul pesan penambahan data site gagal
Pengamatan	Penambahan site gagal, dan muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima
Ubah Data Site	
Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Mengubah atribut data site yang sudah ada dalam *database* dengan data yang benar. Contoh : *SiteName* : Cianting Utara, *Branch* : Purwakarta, *Teritori* : Purwakarta Utara, *Kab* : Purwakarta, *Address* : Desa Cibodas Kec Sepatani, *System* : 2G, *Band* : GSM, *Vendor* : Nokia, *Longitude* : 107.415, *Latitude* : -6.6344, *Status* : After ATP
Yang diharapkan	Semua data yang dimasukkan benar, data ubah tersimpan dan muncul pesan merubah data site berhasil
Pengamatan	Data site yang dirubah berhasil dan muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	

Data Masukan	Mengosongkan atribut data site yang disediakan. Contoh : Sitename : dikosongkan
Yang diharapkan	Data site tidak mengalami perubahan, dan muncul perubahan data site gagal
Pengamatan	Data site gagal dirubah dan muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima
Hapus Data Site	
Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Pilih hapus site atau pilih beberapa site lalu *delete checked* site. Contoh : *Site* : Cianting dengan *SiteId* : 101PWK02
Yang diharapkan	Muncul pesan konfirmasi, yakin hapus data site, data site tersebut statusnya berubah menjadi terhapus dan data tidak ditampilkan kembali
Pengamatan	Data site terhapus dan data tidak ditampilkan kembali
Kesimpulan	Diterima
Cari Data Site	
Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Masukkan kata kunci pencarian. Contoh : Purwakarta
Yang diharapkan	Pencarian sesuai kata kunci pada *database*. Jika diketemukan Semua data yang sesuai kata kunci ditampilkan.
Pengamatan	Data sesuai kata kunci pencarian ditampilkan
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	Masukkan kata kunci yang tidak sesuai. Contoh : Banten
Yang diharapkan	Mencari data sesuai kata kunci, data pencarian tidak ada yang diketemukan dan tidak ada yang ditampilkan
Pengamatan	Data yang sesuai kata kunci tidak diketemukan karena tidak ada yang ditampilkan
Kesimpulan	Diterima

5. Pengolahan *Ticket*

Tabel 4.14 Pengujian Pengolahan *Ticket*

Tambah *Ticket*	
Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Mengisi atribut *ticket* yang disediakan dengan data yang benar. Contoh: *Date*: 2012-07-01, *SiteName*: BBC, *Problem* : Transmition Interminttent, *Alarm*: 7606, *Hardware Impact* : Transmition, *Severit* : Critical, *To Tim*: Access, *Remark*: Check site and change transmition
Yang diharapkan	Pengecekan apakah site tersebut masih open atau tidak jika tidak. Proses penambahan data *ticket* berhasil, *ticket* yang baru tersimpan di *database* dan grafik *open* dashboard HCR mengalami penambahan data. Dan muncul pesan ticket sesuai *no ticket* berhasil ditambahkan
Pengamatan	Semua data yang dimasukkan diterima, penambahan *ticket* berhasil, muncul pesan sesuai yang diharapkan. Dan data grafik dashboard *ticket open* mengalami penambahan data
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	Mengosongkan atribut *ticket* yang disediakan Contoh: *Date* : dikosongkan, *Site Name* : dikosongkan
Yang diharapkan	Proses penambahan *ticket* gagal. Muncul pesan penambahan *ticket* gagal
Pengamatan	*Ticket* gagal ditambahkan dan muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima
Ubah *Ticket*	
Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Mengubah atribut *ticket* yang sudah ada dalam *database* dengan data yang benar. Contoh: *Date*: 2012-07-01, *Site*

	Name: CIBINONG, *Problem*: Transmition Intermintent, *Alarm*: 7606, *Hardware Impact*: Transmition, *Severity*: Critical, *To Tim*: Quality, *Remark*: Check site and change transmition
Yang diharapkan	Semua data yang dimasukkan benar, ubah *ticket* tersimpan didatabase dan muncul pesan merubah *ticket* berhasil
Pengamatan	Merubah *ticket* berhasil dan muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	Mengosongkan atribut *ticket* yang disediakan
Yang diharapkan	Proses perubahan data gagal, ubah *ticket* tidak terjadi. Muncul pesan perubahan *ticket* gagal.
Pengamatan	Data *ticket* tidak mengalami perubahan dan muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima
Hapus *Ticket*	
Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Pilih hapus *ticket* atau pilih beberapa *ticket* lalu pilih *delete checked ticket* Contoh : *NoTicket* : HCR_PWK_2GEID_090612_11
Yang diharapkan	Muncul pesan konfirmasi, klik yakin hapus *ticket*, dan data akan berubah statusnya di *database* dan dashboard HCR mengalami sesuai data *ticket* yang dihapus tidak lagi dihitung. Data *ticket* tersebut tidak ditampilkan kembali
Pengamatan	Muncul pesan konfirmasi, klik yakin hapus *ticket*. Data *ticket* berhasil dihapus dan tidak ditampilkan kembali
Kesimpulan	Diterima
Cari *Ticket*	
Kasus dan Hasil Uji (Data Normal)	

Data Masukan	Masukkan kata kunci pencarian Contoh : Open, Quality atau Bandung
Yang diharapkan	Mencari data sesuai kata kunci di *database*, data berhasil ditemukan kemudian data ditampilkan
Pengamatan	Data sesuai kata kunci diketemukan dan ditampilkan
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	Masukkan kata kunci yang tidak sesuai Contoh : Jakarta
Yang diharapkan	Mencari data sesuai kata kunci di *database*, data pencarian tidak ditemukan maka tidak ada data ditampilkan
Pengamatan	Data pencarian tidak ditemukan, maka tidak ada data yang ditampilkan
Kesimpulan	Diterima

6. Pengolahan *Action*

Tabel 4.15 Pengujian Pengolahan *Action*

Tambah *Action*	
Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Mengisi atribut *action ticket* yang disediakan dengan data yang benar Contoh : *Date*: 2012-07-02, *Status*: Follow-up, *Presentase*: 100, *Desciption*: Change TRX
Yang diharapkan	Proses penambahan action sesuai no *ticket* yang dipilih, data baru tersimpan di *database*. *Action ticket* baru berhasil ditambahkan, *ticket* yang *action* berubah statusnya dan data dashboard HCR mengalami perubahan sesuai status *action*. Muncul pesan *action* berhasil ditambahkan
Pengamatan	Semua data yang dimasukkan diterima, penambahan *action ticket* berhasil sesuai *no ticket*. Muncul pesan sesuai yang diharapkan. *Ticket* yang di *action* berubah statusnya dan data

	dashboard HCR mengalami perubahan sesuai status *action*
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	Mengisi atribut *action ticket* yang disediakan dengan data yang salah Contoh : *Date* : 2012-07-02, *Status* : Follow-up, *Presentase* : 200, *Desciption* : Change TRX
Yang diharapkan	Proses penambahan *action* gagal, karena presentase terbesar 100, jika nilai presentase melebihi, *action* gagal disimpan. Muncul pesan periksa nilai presentase *action*
Pengamatan	Proses penambahan *action* gagal dan muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima
Ubah *Action*	
Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Mengubah atribut *action ticket* yang sudah ada dalam *database* dengan data yang benar. Contoh: *Date*: 2012-07-03, *Status*: Done, *Presentase*: 100, *Desciption*: *Change TRX dan BB2X*
Yang diharapkan	Semua data yang dimasukkan benar, data ubah action tersimpan dan muncul pesan merubah *action ticket* berhasil
Pengamatan	Merubah *action ticket* berhasil, muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	Mengisi atribut *action ticket* yang disediakan dengan data salah Contoh : perubahan data *Presentase* : 150
Yang diharapkan	Proses perubahan data *action ticket* gagal karena presentase *ticket* melebihi 100. Muncul pesan periksa presentase nilai *action*
Pengamatan	*Action ticket* tidak mengalami perubahan data, muncul pesan

	sesuai yang diharapkan
Kesimpulan	Diterima

7. Konfirmasi *Ticket*

Tabel 4.16 Pengujian Konfirmasi *Ticket*

Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Pilih konfirmasi *ticket* atau pilih beberapa *ticket* lalu pilih *confirm checked ticket* Contoh: *NoTicket* : HCR_TSK_3GEID_060612_10
Yang diharapkan	Muncul pesan konfirmasi, yakin konfirmasi *ticket*, dan data akan terubah statusnya menjadi *closed* dan *dashboard* HCR mengalami perubahan yaitu *closed* HCR menjadi bertambah karena *ticket* ini telah dikonfirmasi
Pengamatan	Muncul pesan konfirmasi, yakin konfirmasi *ticket* akan berubah status menjadi *closed* dan *dashboard* HCR mengalami perubahan yaitu *closed* HCR menjadi bertambah karena *ticket* ini telah dikonfirmasi
Kesimpulan	Diterima

8. Pencarian *Monitoring*

Tabel 4.17 Pengujian Pencarian *Monitoring*

Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Memilih kriteria *monitoring ticket* HCR dengan data yang benar. Contoh : *Date Start*: 2012-06-20 s/d *Date End*: 2012-07-02 atau *To Tim*: Access
Yang diharapkan	Mencari data pencarian sesuai kriteria monitoring, Data pencarian ditemukan di *database*, hasil pencarian akan ditampilkan
Pengamatan	Mencari data sesuai kriteria monitoring dan menampilkan

	hasilnya sesuai yang diharapkan
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	Memilih *monitoring ticket* HCR dengan data salah. Contoh: *Date Start* : 2012-07-03 s/d *Date End* : 2012-07-02
Yang diharapkan	Pencarian data monitoring salah dan muncul pesan tanggal awal lebih besar daripada tanggal akhir
Pengamatan	Pencarian data monitoring salah dan muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima

9. Laporan

Tabel 4.18 Pengujian Laporan

Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Memilih kriteria laporan dengan data yang benar Contoh : *Branch* : Bandung atau *To Tim* : Access *Date Start* : 2012-06-01 s/d *Date End* : 2012-07-01
Yang diharapkan	Mencari data sesuai kriteria laporan dan menampilkan hasil laporan dalam bentuk *excel*
Pengamatan	Menampilkan data hasil laporan sesuai kriteria pilihan dalam bentuk *excel*
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	Memilih kriteria laporan dengan data salah Contoh: *Start date* diisi lebih besar daripada *end date.* Contoh: *Date Start* : 2012-07-04 s/d *Date End* : 2012-07-02
Yang diharapkan	Laporan dibatalkan dan muncul pesan tanggal awal lebih besar daripada tanggal akhir
Pengamatan	Laporan gagal dan muncul pesan sesuai yang diharapkan

Kesimpulan	Diterima

10. Pencarian *Dashboard*

Tabel 4.19 Pengujian Pencarian D*ashboard*

Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Memilih bulan dan tahun *dashboard* HCR. Contoh : June 2012
Yang diharapkan	Data sesuai bulan dan tahun pilihan ditemukan dan tampilkan grafik dashboard sesuai bulan dan tahun
Pengamatan	Ditampilkan grafik dashboard sesuai yang diharapkan
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	
Data Masukan	Memilih bulan dan tahun *dashboard* HCR dengan data salah Contoh : June 2014
Yang diharapkan	Data sesuai bulan dan tahun pilihan tidak ditemukan dan grafik dashboard kosong karena tidak ada data yang dihitung
Pengamatan	Ditampilkan grafik dashboard sesuai yang diharapkan
Kesimpulan	Diterima

11. Perubahan *Password*

Tabel 4.20 Pengujian Perubahan *Password*

Kasus dan Hasil Uji (Data Normal)	
Data Masukan	Mengubah atribut ubah *password* dengan data yang benar
Yang diharapkan	Semua data yang dimasukkan benar, data ubah password tersimpan dan muncul pesan merubah *password* berhasil
Pengamatan	Semua data yang dimasukkan diterima, password berhasil dirubah dan muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima
Kasus dan Hasil Uji (Data Gagal)	

Data Masukan	Mengisi atribut ubah *password* dengan data salah. Contoh : *Old Password* : Salah atau *New Password* tidak sama dengan *Re-type Password*
Yang diharapkan	Proses perubahan *password* gagal dan muncul pesan password lama salah atau password baru dan penulisan ulang password baru harus sama.
Pengamatan	Perubahan *password* gagal dan muncul pesan sesuai yang diharapkan
Kesimpulan	Diterima

4.2.3 Kesimpulan Pengujian *Alpha*

Berdasarkan hasil pengujian *alpha* yang telah dilakukan dapat ditarik kesimpulan bahwa aplikasi sudah berjalan cukup maksimal, tetapi tidak menutup kemungkinan dapat terjadi kesalahan pada saat aplikasi digunakan, sehingga membutuhkan proses *maintenance* untuk lebih mengetahui kekurangan dari aplikasi.

4.2.4 Pengujian Beta

Pengujian *beta* merupakan pengujian yang dilakukan secara langsung pada perusahaan yaitu untuk mengetahui kualitas dari *aplikasi* yang telah dibuat, apakah sudah sesuai dengan yang diharapkan atau belum, pengujian dilakukan dengan melakukan kuesioner kepada pegawai yang nantinya sebagai pengguna aplikasi. Dari kuesioner tersebut akan dilakukan perhitungan untuk dapat mengambil kesimpulan terhadap penilaian dari aplikasi yang baru dibuat.

Berdasarkan data hasil kuesioner, dapat dicari presentase masing – masing jawaban dengan menggunakan rumus : $Y = P/Q * 100\%$

Keterangan :

P : Banyaknya jawaban responde tiap soal

Q : Jumlah responden

Y : Nilai presentase

4.2.5 Pertanyaan dan Hasil Pengujian Beta

Berikut ini adalah hasil presentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas. Kuesioner ini diujikan kepada 10 orang *user* (3 tim *project*, 3 tim *quality*, 3 tim *access*, 1 *manager*) yang menggunakan aplikasi ini dan merupakan pegawai PT Indosat Tbk. Pertanyaan kuesioner yang diberikan ada 4 pertanyaan umum dan sisanya adalah pertanyaan sesuai fungsi *user*. (contoh hasil kuesionernya dilampirkan)

1. Apakah dengan urutan *ticket* HCR di aplikasi ini mudah dimengerti? (Semua user)

Tabel 4.21 Hasil Pengujian Soal no 1

No	Keterangan	Responden	Presentase (%)
1	Sangat Mudah	0	0 %
2	Mudah	6	70 %
3	Biasa – Biasa Saja	2	20 %
4	Cukup Sulit	1	10 %
5	Sulit	1	10 %
6	Sangat Sulit	0	0 %

Berdasarkan hasil presentase pada Tabel 4.21 maka dapat disimpulkan bahwa 60% responden menyatakan aplikasi ini mudah dimengerti dengan bentuk permintaan urutan ticket, 20% responden menyatakan biasa – biasa saja, 10% menyatakan cukup sulit dan 10% menyatakan sulit.

2. Apakah aplikasi ini memudahkan dalam penyampaian informasi proses HCR dari pusat ke cabang atau sebaliknya? (Semua *user*)

Tabel 4.22 Hasil Pengujian Soal no 2

No	Keterangan	Responden	Presentase (%)
1	Sangat Mudah	0	0 %
2	Mudah	8	80 %
3	Biasa – Biasa Saja	2	20 %
4	Cukup Sulit	0	0 %
5	Sulit	0	0 %
6	Sangat Sulit	0	0 %

Berdasarkan hasil presentase pada Tabel 4.22 maka dapat disimpulkan bahwa 80% responden menyatakan aplikasi ini memudahkan dalam penyampaian informasi proses HCR dari pusat ke cabang atau sebaliknya dan sisanya 20 % menyatakan biasa – biasa saja.

3. Apakah sudah cukup jelas untuk melihat *history* pengerjaan HCR dengan memonitoring status *ticket* pada aplikasi ini ? (Semua *user*)

Tabel 4.23 Hasil Pengujian Soal no 3

No	Keterangan	Responden	Presentase (%)

1	Sangat Jelas	2	20 %
2	Jelas	0	0 %
3	Biasa – Biasa Saja	4	40 %
4	Cukup Jelas	4	40 %
5	Tidak Jelas	0	0 %
6	Sangat Tidak Jelas	0	0 %

Berdasarkan hasil presentase pada Tabel 4.23 maka dapat disimpulkan bahwa 20% responden menyatakan aplikasi ini sangat jelas untuk melihat history pengerjaan HCR, 40% responden menyatakan biasa – biasa saja dan sisanya 10 % menyatakan cukup jelas.

4. Apakah dalam mengelola action HCR pada aplikasi ini mudah? (Tim *Access*)

Tabel 4.24 Hasil Pengujian Soal no4

No	Keterangan	Responden	Presentase (%)
1	Sangat Mudah	0	0 %
2	Mudah	2	66,67 %
3	Biasa – Biasa Saja	1	33,33 %
4	Cukup Sulit	0	0 %
5	Sulit	0	0 %
6	Sangat Sulit	0	0 %

Berdasarkan hasil presentase pada Tabel 4.24 maka dapat disimpulkan bahwa 66,67% responden menyatakan aplikasi ini memudahkan mengelola action HCR dan sisanya 33,37 % menyatakan biasa – biasa saja.

5. Apakah cukup jelas dengan melihat dashboard HCR pada aplikasi ini mengerti adanya perubahan *ticket* HCR? (Semua user)

Tabel 4.25 Hasil Pengujian Soal no 5

No	Keterangan	Responden	Presentase (%)
1	Sangat Jelas	0	0 %
2	Jelas	6	60 %
3	Biasa – Biasa Saja	2	20 %
4	Cukup Jelas	2	20 %
5	Tidak Jelas	0	0 %
6	Sangat Tidak Jelas	0	0 %

Berdasarkan hasil presentase pada Tabel 4.25 maka dapat disimpulkan bahwa 60% responden menyatakan dengan dashboard aplikasi ini cukup jelas untuk melihat perubahan *ticket* HCR, 20% menyatakan biasa – biasa saja dan 20% responden menyatakan cukup jelas.

6. Apakah dalam pembuatan laporan HCR pada aplikasi ini mudah? (Tim Quality)

Tabel 4.26 Hasil Pengujian Soal no 6

No	Keterangan	Responden	Presentase (%)
1	Sangat Mudah	3	100 %
2	Mudah	0	0 %
3	Biasa – Biasa Saja	0	0 %
4	Cukup Sulit	0	0 %

5	Sulit	0	0 %
6	Sangat Sulit	0	0 %

Berdasarkan hasil presentase pada Tabel 4.26 maka dapat disimpulkan bahwa 100% responden menyatakan aplikasi ini sangat mudah untuk pembuatan laporan.

7. Apakah dalam mengkonfirmasi *ticket* yang telah di *action* HCR pada aplikasi ini mudah? (Tim *Quality* dan Tim *Project*)

Tabel 4.27 Hasil Pengujian Soal no 7

No	Keterangan	Responden	Presentase (%)
1	Sangat Mudah	0	0 %
2	Mudah	4	66,66 %
3	Biasa – Biasa Saja	1	16,67%
4	Cukup Sulit	1	16,67 %
5	Sulit	0	0 %
6	Sangat Sulit	0	0 %

Berdasarkan hasil presentase pada Tabel 4.27 maka dapat disimpulkan bahwa 66,66% responden menyatakan mudah untuk mengkonfirmasi ticket yang sudah di action, 16,67% responden menyatakan biasa – biasa saja dan sisanya 16,67% responden menyatakan cukup sulit.

8. Apakah dalam proses permintaan HCR pada aplikasi ini mudah? (Tim *Project*)

Tabel 4.28 Hasil Pengujian Soal no 8

No	Keterangan	Responden	Presentase (%)
1	Sangat Mudah	0	0 %
2	Mudah	2	66,67 %
3	Biasa – Biasa Saja	1	33,33 %
4	Cukup Sulit	0	0 %
5	Sulit	0	0 %
6	Sangat Sulit	0	0 %

Berdasarkan hasil presentase pada Tabel 4.28 maka dapat disimpulkan bahwa 66,67% responden menyatakan aplikasi ini memudahkan untuk permintaan HCR dan 33,33% responden menyatakan biasa – biasa saja.

4.2.6 Kesimpulan Pengujian Beta

Berdasarkan dari hasil presentase dari kuesioner pengguna maka dapat di tarik kesimpulan aplikasi ini siap untuk dipakai oleh pengguna karena sebagian besar dari hasil menyebutkan fakta bahwa aplikasi telah memenuhi kebutuhan dan sesuai dengan yang di harapkan pengguna. Sebagian besar pengguna yang telah mencoba aplikasi ini menyebutkan bahwa telah puas dengan fitur – fitur yang ada. Apabila muncul kebutuhan lain dikemudian hari, maka akan dimasukkan ke tahap *maintenance.*