BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Merancang suatu sistem maka terlebih dahulu dilakukan menganalisa sistem yang ada saat iniuntuk mengetahui permasalahan dan kebutuhan dari sistem tersebut.Dari hasil analisis dapat diketahui kelemahan atau kekurangan sistem yang lama dan dapat dirancang atau diperbaiki menjadi sebuah sistem yang lebih efektif dan efisien.

3.1.1 Analisis Masalah

Masalah yang ada saat ini yaitu bagaimana menginformasikan dengan cepat HCR untuk segera ditangani permasalahannya. Pemberitahuan dari tim*project* ke tim *quality* ataupun tim *acces* memberitahukannya melalui email dan data yang diberikan belum dipisahkan antar tim *quality* maupun tim *access*. Sehingga tidak efektif dan efisien sebagaimana diingikan untuk segera di perbaiki permasalahan *hardware* yang menjadi kendala.

Permasalahan lain perbedaan *format* permintaan dari tim*project*, yaitu 2G Nokia, 2G Ericsson dan 3G Ericsson. Sehingga menjadi masalah apabila ada pegawai baru membaca permintaan HCR yang berbeda tersebut. Selain itu lokasi permasalahan HCR di PT Indosat area *West Java* tidak hanya terjadi di satu *branch* saja sehingga perlu diinformasikan ke *branch* yang lain.

Data semua aktifitas HCR perlu direkap dan diinformasikan ke level *manager* tetapi dari sistem yang sekarang masih mengalami kendala waktu dan

*update*informasinya. Karena harus menunggu atau memintainformasi HCR yang sudah di *action*olehtim yang mengeksekusinya.

3.1.2 AnalisisSistem Berjalan

Analisis sistem berjalan adalah analisis yang dilakukan pada sistem yang sedang berjalan di PT. Indosat Tbk pada saat ini. Berdasarkan hasil observasi yang dilakukan dan wawancara dengan pihak yang terlibat,didapatkan prosedur yang sedang berjalan adalah sebagai berikut :

1. Proses permintaan HCR
2. Proses pengerjaan HCR
3. Proses pelaporan HCR

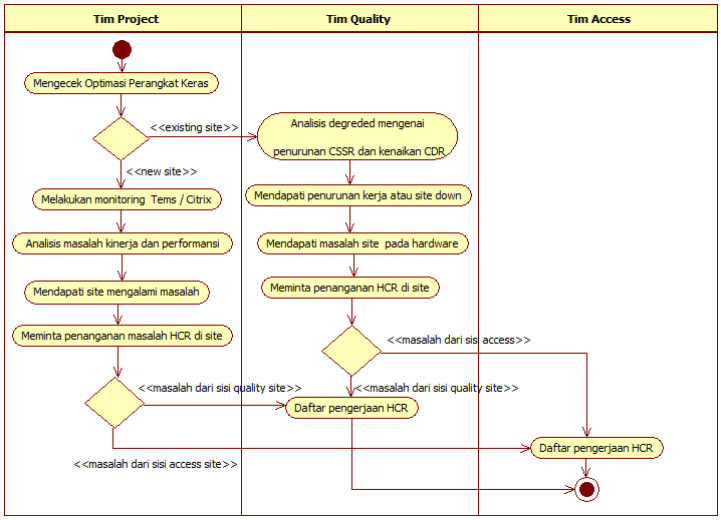
Proses-proses yang berjalan tersebut dilakukan antara tim yang meminta HCR, mengerjakannya dan merekapnya untuk dijadikan laporan ke *manager*. Penjelasan dari setiap prosedur adalah sebagai berikut :

3.1.2.1 ProsesPermintaan HCR

Proses permintaan HCR adalah prosedur dimana tim project membuat daftar hardware yang mesti harus diperbaiki. Permintaan ini dibuat berdasarkan analisis data, sehingga diketahui adanya permasalahan didalam *hardware*. Daftar permintaan ini terdiri dari 3 jenis yaitu 2G Nokia, 2G Ericsson dan 3G Ericsson. Pada gambar 3.1akan dijelaskan mengenai *Activity Diagram* proses permintaan HCR. Proses yang dilakukan pada prosedur permintaan HCR adalah sebagai berikut :

1. Tim *project* perlu optimasi perangkat keras supaya kestabilan penggunaan jaringan komunikasi berjalan dengan baik. Tetapi kenyataannya masih ada yang mengalami gangguan dari site baru maupun site yang*existing*.

2. Tim *project* memonitoringtems atau citrix yaitu perangkat lunak untuk melihat parameter site antara lain traffic, presentasi kinerja site, presentasi *call complete,* presentasi *blocking*, presentasi signal dan lain sebagainya.
3. Tim *quality*menganalisis *degraded*CSSR(*call setup success rate*), CDR(*call drop rate*). Apabila ditemukan penurunan kinerja atau site *down*pada site*existing*yang disebabkan perangkat keras berarti terjadi masalah pada site.
4. Maka didapatkan masalah pada hardware sehingga tim*project* maupun tim *quality* membuat permintaan penanganan HCR.
5. Permintaan penanganan HCR dipisahkanmenurut masalahnya untuk membedakan HCR ditangani oleh timmana.
6. Jika permasalahan yang dihadapi adalah masalah akses*hardware* maka daftar diberikan ke tim*access*. Dan apabila masalah dari quality *hardware* maka daftar diberikan ke tim*quality*.

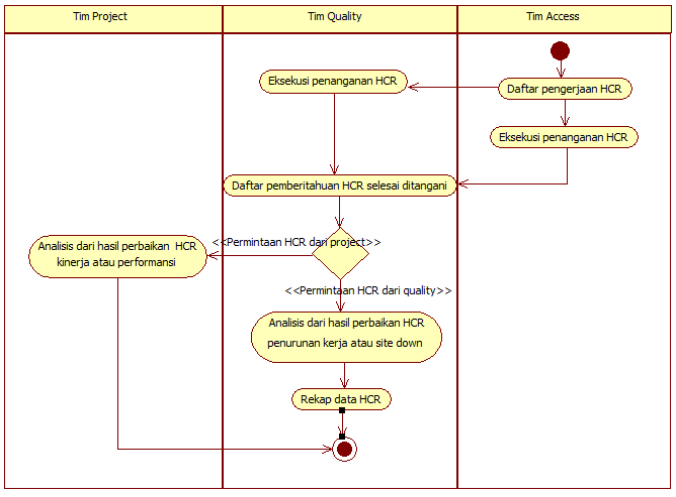


Gambar 3.1 *Activity Diagram* ProsesPermintaan HCR

3.1.2.2 Proses Pengerjaan HCR

Proses pengerjaan HCR bisa dilakukan oleh tim*quality* atau tim *access*tergantung *hardware* site yang mengalami masalah. Biasanya setelah di analisis oleh tim*project* atau tim *quality* diketahui mesti ditangani oleh tim mana. Pada gambar 3.2akan dijelaskan mengenai *Activity Diagram*proses pengerjaan HCR. Proses yang dilakukan pada prosedur pengerjaan HCR adalah sebagai berikut :

1. Tim *quality* dan tim*access* menerima daftar pengerjaan HCR.
2. Pengerjaan HCR dieksekusi, dantim yang mengerjakan HCR memberikan daftar pemberitahuan hasil pengerjaan yang telah selesai ditangani.
3. Daftar hasil pengerjaan HCR diberikan untuk diketahui dan dianalisa masalah yang ada sebelumnya.
4. Daftarnya dibedakan sesuai permintaan HCR dari tim*project* atau *quality*, selanjutanya hasil pengerjaan *hardware* di analisis kinerja, performansi dan keadaan site yang sekarang.
5. Analisis pengerjaan HCRkembali normal,permintaan HCR ditutup dan data HCR direkap tim*quality*.

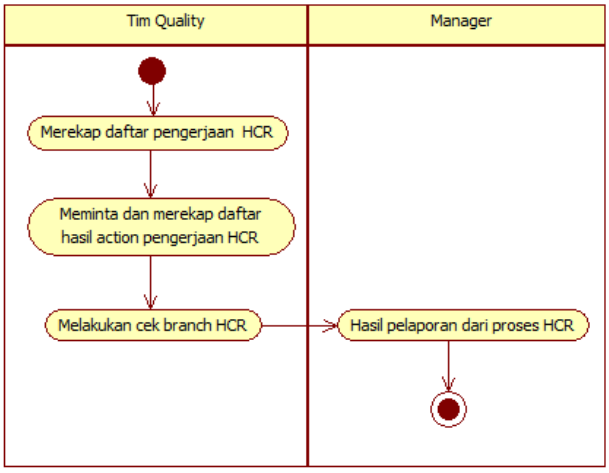


Gambar 3.2 *Activity Diagram*Proses Pengerjaan HCR

3.1.2.3 Proses Pelaporan HCR

Proses pelaporan HCR perlu dilakukan agar diketahui oleh level *manager* tingkat masalah HCR yang menganggu *quality* trafik area*west java*. Laporan yang dibutuhkan permingguan, untuk didapatkan perbandingan dari minggu sekarang ke minggu - minggu sebelumnya dan dibedakan tiap *branch* di *West Java*.Pada gambar 3.3akan dijelaskan mengenai *Diagram Activity* proses pelaporan HCR. Proses yang dilakukan pada prosedur pelaporan HCR adalah sebagai berikut :

1. Tim *quality*merekap daftar pengerjaan HCR di*Microsoft excel* .
2. Selanjutnyameminta dan merekap daftar hasil *action* pengerjaan dan perkembangannya ke tim yang mengerjakan HCR
3. Tim *quality* melakukan pengecekan di*branch* mana site yang *hardware*nya mengalami masalah.
4. Hasilnya semuanya digunakan untuk laporan HCR ke*manager*.



Gambar 3.3 *ActivityDiagram* Proses Pelaporan HCR

3.1.3 Analisis Kebutuhan Data Non Fungsional

Analisis kebutuhan non fungsional adalah suatu analisis untuk mengetahui elemen – elemen pendukung sistem yang akan dijalankan seperti spesifikasi kebutuhan

sistem. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan. Adapun kebutuhannya sebagai berikut:

3.1.3.1 Analisis Perangkat Keras

Perangkat keras adalah seluruh komponen atau unsur peralatan yang digunakan untuk menunjang penggunaan sistem.Di PT Indosat Tbk area *West Java* memiliki sejumlah komputer yang memiliki spesifikasi minimumperangkat komputer sebagai berikut:

1. Processor dengan kecepatan 2,8 GHz
2. RAM sebesar 2 GB
3. Kapasitas Hardisk 160 GB
4. Monitor VGA dengan revolusi 1024 x 768
5. DVD-RW 52x
6. Mouse
7. Keyboard
8. Lan Card

Kebutuhan minimal perangkat keras yang dibutuhkan spesifikasi sebagai berikut :

1. Processor dengan kecepatan 1,8 GHz
2. RAM sebesar 500 MB
3. Kapasitas *free* Hardisk 10 GB
4. Monitor VGA dengan revolusi 1024 x 768
5. Mouse
6. Keyboard

7. Lan Card

Setelah dilakukan analisis terhadap perangkat keras yang ada di perusahaan ternyata sudah memenuhi spesifikasi kebutuhan untuk menggunakan perangkat lunak yang akan dibangun.

3.1.3.2 Analisis Perangkat Lunak

Perangkat lunak yangumum digunakan PT Indosat Tbk area *West Java* adalah sebagai berikut :

1. Sistem Operasi Windows XP atau 7
2. Microsoft Excel 2003 dan 2007
3. Web browser Mozila Firefox dan Google Chrome

Kebutuhan perangkat lunak untuk membangun aplikasi ini adalah sebagai berikut:

1. Sistem Operasi Windows XP atau 7
2. Web Server Xampp
3. Microsoft Excel 2003 dan 2007
4. Web browser Mozila Firefox dan Google Chrome

Dari hasil perbandingan antara kebutuhan non fungsional dan hasil analisis maka pihak perusahaanhanya tinggal membutuhkan web serverxampp1.7.5 yang aplikasinya dapat diperoleh secara gratis dikarenakan merupakan aplikasi *open source.*

3.1.3.3 Analisis Pengguna

Analisis pengguna dimaksudkan untuk mengetahui siapa saja pengguna yang terlibat dalam proses HCR sehingga dapat diketahui tingkat pengalaman dan

pemahaman pengguna terhadap komputer. Dari hasil pengumpulan data, diketahui bahwa karakteristik pegawai PT Indosat Tbk adalah sebagai berikut :

Tabel 3.1 Karakteristik Pegawai PT Indosat Tbk

Pengguna	*Manager*	Tim *Project, Quality, Access*
Tanggung Jawab	Memeriksa laporan HCR dan melihat aktifitas HCR	Membuat ticket, mengeksekusi HCR, menginformasikan HCR
Tingkat Pendidikan	Lulus S1	Lulus D3 atau SMK
Tingkat Ketrampilan	Mengerti dan dapat menggunakan komputer dan aplikasi *webbase*	Mengerti dan dapat menggunakan komputer dan aplikasi *webbase*

Pengguna sistem yang akandibangun ini dibagi menjadi 3 kategori, yaitu *administrator*, *manager* dan *user* (tim *project*, tim *quality*, tim *access*). Spesfikasi pengguna dapat dideskripsikan sebagai berikut:

Tabel 3.2Karakteristik Pengguna

Pengguna	*Administrator*	*Manager*	*User* (Tim *Project*, *Quality*, *Access*)
Hak Akses	Mengelola *user*, mengelola data master, data *ticket* dan data action.	Memonitoring HCR, Dashboard dan kinerja user.	Membuat *ticket*, mengeksekusi HCR, menginformasikan HCR
Tingkat Pendidikan	Lulus S1	Lulus S1	Lulus D3 atau SMK
Tingkat Ketrampilan	Mengerti dan dapat menggunakan komputer	Mengerti dan dapat menggunakan komputer	Mengerti dan dapat menggunakan komputer
Pengalaman	Mengoperasikan komputer dan mengikuti pelatihan	Mengoperasikan komputer dan mengikuti pelatihan	Mengoperasikan komputer dan mengikuti pelatihan

Pengguna yang akan menggunakan aplikasi yang dibangun ini terdiri dari 3 kelompok pengguna, dengan asumsi pengguna sudah terbiasa menggunakansistem barbasis *webbase*.

1. *Administrator*

Administrator adalah bagian pengguna sistem yang menentukan siapapengguna lain yang boleh menggunakan sistem ini dan dapat merubah data master.*Administrator* berasal dari bagian tim*quality* yang memiliki kemampuan dan pengetahuan yang lebihtentang informasi HCR.

2. *Manager*

*Manager*adalah bagian pengguna sistem yang dapat melihat laporan aktifitas informasi keseluruhan data pada sistem.

3. *User* (Tim *Project, Quality, Access*)

User adalah bagian pengguna sistem yang memiliki akses hanya dapatmengelola data master, data ticket dan data action.

3.1.3.4 Analisis *Severity* dan Status *Ticket*

Analisis status ini merupakan penjelasan tingkatan *ticket* dan *severity*sesuai sistem yang berjalan sekarang.Dan penjelasannya sebagai berikut :

1. Severity*Ticket*

   Menjelaskan tentang estimasi terhadap suatu *ticket* sesuai batasan waktu yang ditetapkan perusahaan untuk mengerjakan HCR.*Severityticket* antara lain :

   a. *Critical* : severity*ticket* tertinggi, yaitu segera ditangani ketika ada *ticket* HCR *open*. Karena site mengalami permasalahan yang serius dan akan berdampak secara langsung ke jaringan pengguna.

   b. *Urgent* : severity *ticket* dibawah *critical*, yaitu segera ditangani dengan syarat *ticketcritical* sudah *closed* dan toleransi waktunya 3 sampai 6 hari waktu pengunduran perbaikan HCR.

c. *Normal* :severity *ticket* yang biasa, yaitu ada masalah di *hardware* tetapi tidak berdampak langsung ke jaringan. Maka pengerjaannya bisa dikerjakan setelah *ticket* yang severity *critical* dan *urgent* selesai perbaikan.

2. Status *Ticket*

Menjelaskan proses perkembangan dari *ticket*. Status *ticket* antara lain :

a. *Open* : status *ticket* pertama kali dibuat dan belum dikerjakan.

b. *Follow-up*: status *ticket* dikerjakan dan belum selesai keseluruhan.

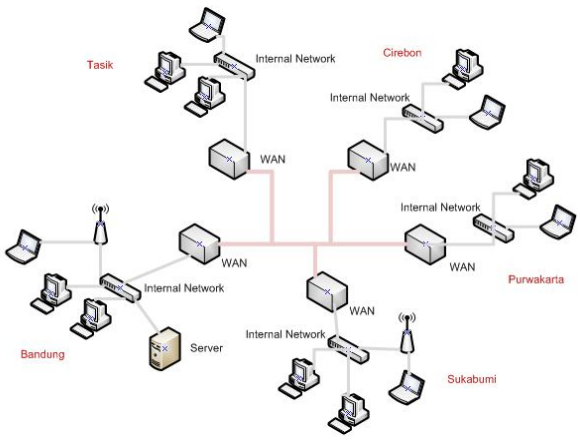
c. *Done* : status *ticket* sudah dikerjakan keseluruhan.

d. *Closed*: status *ticket* sudah selesai dan sudah dikonfirmasi oleh yang membuat permintaan HCR.

Untuk sistem yang akan dibangun untuk permberian status *open* diperjelas dengan diberi warna merah, *follow-up* dan *done* diberi warna biru dan *closed* diberi warna hijau.

3.1.3.5 Analisis Jaringan

Penggambaran analisis jaringan *intranet* di PT Indosat Tbk area *West Java*dijelaskanpada denahgambar 3.4.



Gambar 3.4 AnalisisStruktur Jaringan

3.1.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proseskegiatan yang terjadi di dalam sistem dan menjelaskan sesuai kebutuhan yang diperlukan agar sistem dapat berjalan dengan baik.

Analisis yang dilakukan dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahapan pemodelan dalam analisis antara lain mengidentifikasi aktor, pembuatan us*e case diagram, use case scenario, activity diagram, sequence diagram, class diagram, component diagram dan deployment diagram*.

Aktor yang telibat dalam sistem ini ada 5 aktor yaitu *administrator*, timp*roject*, tim *quality*, tim *access* dan *manager*. Deskripsi untuk setiap aktornya akan dijelaskan pada tabel 3.3.

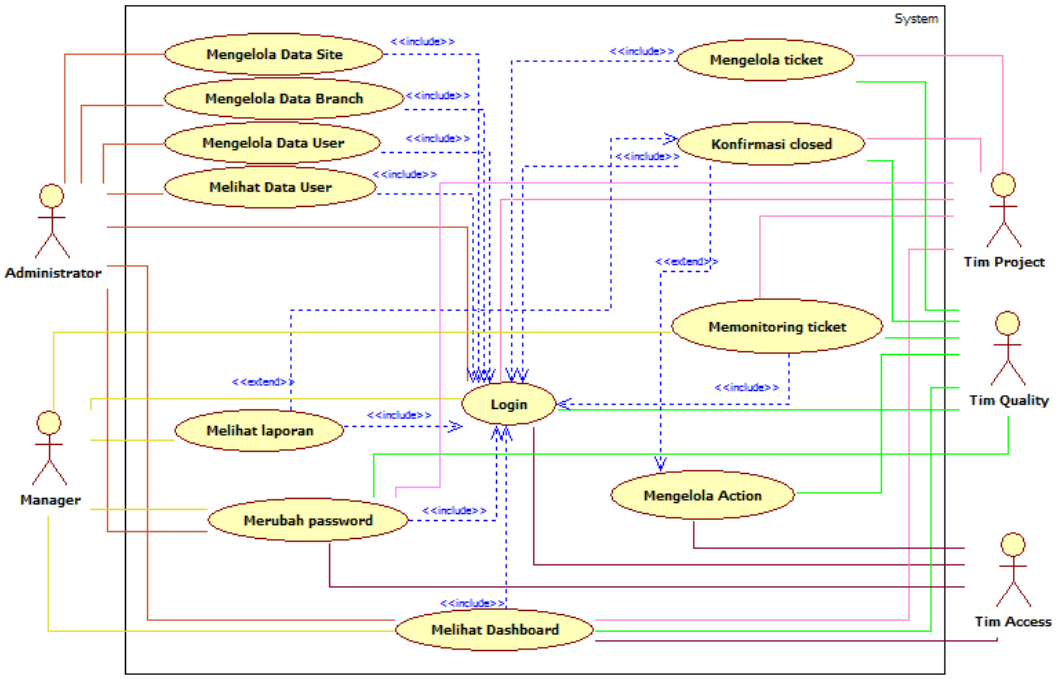
Tabel 3.3 Identifikasi Aktor

No	Aktor	Deskripsi
1	*Administrator*	Mengatur dan mengelola *user* masuk ke dalam sistem. Melakukan evaluasi keseluruhan data master.
2	Tim *Project*	Membuat daftar *ticket*HCR dan mengecek hasil penanganan HCR
3	Tim *Quality*	Membuat daftar *ticket*HCR, mengecek hasil penanganan HCR, dan menambahkan action perbaikan HCR
4	Tim *Access*	Menambahkan*action* perbaikan HCR
5	*Manager*	Mengawasi dan melakukan evaluasi penanganan kinerja HCR

3.1.4.1 *Use Case Diagram*

Use Case Diagram adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar dan menjelaskan sistem secara fungsional yang terlihat pengguna. Adalah konstruksi untuk mendeskripsikan bagaimana sistem terlihat dimata pengguna.

Dari identifikasi aktor yang terlibat di atas maka *Use Case Diagram* dapat dilihat pada gambar 3.5.



Gambar 3.5*Use Case* Diagram

3.1.4.2 *Use Case Scenario*

Use case scenario adalah deskripsi urutan langkah – langkah dalam proses kerja, baik yang dilakukan aktor terhadap sistem maupun sistem terhadap aktor. Berdasarkan *use case diagram* pada gambar 3.5 maka u*se case scenario*sebagai berikut :

1. ***Use Case Scenario* Login *User***

Interaksi antara aktor pengguna sistem dengan *use case login* dijelaskan dalam *use case scenario* sebagai berikut :

Tabel 3.4*Use Case Scenario* Login *User*

Identifikasi	
Nama	Login

Tujuan	Masuk kedalam sistem sebagai *user* / pengguna
Deskripsi	Proses login ini sebagai autentifikasi hak akses*user* sebagai user dalam sistem
Aktor	*Administrator*, *manager*, tim *project*, tim *quality* dan tim *access*
Skenario Utama	
Kondisi Awal	Halaman login ditampilkan

Aksi Aktor	Reaksi Sistem
1. Mengisi username dan *password* pada halaman login kemudian menekan tombol login	2. Sistem mengecek isian halaman login, jika masih ada yang kosong akan ditampilkan pesan *field* masih kosong, jika tidak sistem mengautentifikasi data login dengan data pengguna pada *database*
	3. Menampilkan halaman utama untuk pengguna
SkenarioAlternatif (Jika gagal)	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan pesan tidak terdaftar di *database*
	2. Menampilkan kembali halaman login
3. Mengisi username dan *password* di halaman login kemudian menekan tombol login	4. Mengautentifikasi data login dengan data pengguna pada *database*
	5. Menampilkan halaman utama untuk pengguna

Kondisi Akhir	*User* / pengguna dapat melakukan akses pada sistem sesuai hak akses masing – masing

2. *Use Case Scenario* Mengelola Data Site

Interaksi antara aktor, yaitu administrator dengan *use case* mengelola data site dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.5*Use Case Scenario* Mengelola data site

Identifikasi		
Nama	Mengelola data site	
Tujuan	Untuk mengelola data site	
Deskripsi	Proses mengelola data site meliputi menambah, mengubah, menghapus dan mencari data site	
Aktor	Administrator	
Usecase Yang Berkaitan	-	
Skenario Utama		
Kondisi Awal	Tampilan daftar data site	
Aksi Aktor		Reaksi Sistem
1. *Administrator* memilih tombol tambah site baru		2. Sistem akan menampilkan halaman isi tambah data site kepada administrator
3. *Administrator* dapat mengisikan data site baru kemudian menekan tombol simpan		4. Sistem memeriksa data yang diisikan *administrator*, jika ada isian wajib diisi dikosongkan, sistem akan memunculkan notifikasi isian tidak boleh kosong
		5. Jika pengecekan berhasil dan data valid lalu sistem akan menyimpan ke *database* dan menampilkan notifikasi berhasil menambah site
6. *Administrator* mencari data site yang ingin diubah		7. Sistem menampilkan hasil daftar pencarian
8. *Administrator* memilih tombol ubah pada data site yang diinginkan		9. Sistem akan menampilkan halaman ubah site sesuai data
10. *Administrator* mengisi data yang diubah dengan data yang benar dan valid		11. Sistem memeriksa data yang diubah*administrator*, jika ada ubahan wajib diisi dikosongkan, sistem akan memunculkan notifikasi isian tidak boleh kosong

	12. Jika pengecekan berhasil dan data valid lalu sistem akan menyimpan data yang diubah ke *database* dan menampilkan notifikasi berhasil mengubah site
13. *Administrator* mencari data site yang ingin dihapus	14. Sistem menampilkan hasil daftar pencarian
15. *Administrator* memilih data site yang ingin dihapus	16. Sistem akan menampilkan pesan, apakah data site ingin dihapus atau tidak
17. *Administrator* akan menekan tombol Ok	18. Sistem akan menghapus data yang diinginkan dan tampilan daftar site akan menghilangkan data tersebut
19. *Administrator* mengisikan data site yang dicari di *field* cari	20. Sistem akan mencari data site tersebut, jika ditemukan akan ditampilkan dalam daftar site tetapi jika tidak ditemukan sistem menampilkan pesan data tidak ditemukan
Skenario Alternatif (Jika gagal)	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan notifikasi data site gagal ditambah karena site-id yang ditambahkan sudah ada di data site
2. Mengisi kembali site-id di halaman tambah data site	
Kondisi Akhir	Data site disistem sudah mengalami proses pengelolaan

3. *Use Case Scenario* Mengelola Data *Branch*

Interaksi antara aktor, yaitu administrator dengan *use case* mengelola data *branch* dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.6*Use Case Scenario* Mengelola data *branch*

Identifikasi	
Nama	Mengelola data *branch*

Tujuan	Untuk mengelola data *branch*
Deskripsi	Proses mengelola data *branch*meliputi menambah, mengubah, menghapus dan mencari data *branch*
Aktor	Administrator
Usecase Yang Berkaitan	-
Skenario Utama	
Kondisi Awal	Tampilan daftar data *branch*

Aksi Aktor	Reaksi Sistem
1. *Administrator* memilih tombol tambah *branch*baru	2. Sistem akan menampilkan halaman isi tambah data *branch*kepada *administrator*
3. *Administrator* dapat mengisikan data *branch*baru kemudian menekan tombol simpan	4. Sistem memeriksa data yang diisikan *administrator*, jika ada isian wajib diisi dikosongkan, sistem akan memunculkan notifikasi isian tidak boleh kosong
	5. Jika pengecekan berhasil dan data valid lalu sistem akan menyimpan ke *database* dan menampilkan notifikasi berhasil
6. *Administrator* mencari data *branch* yang ingin diubah	7. Sistem menampilkan hasil daftar pencarian
8. *Administrator* memilih tombol ubah pada daftar *branch*yang diinginkan	9. Sistem akan menampilkan halaman ubah *branch*sesuai data yang diinginkan
10. *Administrator* mengisi data yang diubah dengan data *branch*yang benar dan valid	11. Sistem memeriksa data yang diubah*administrator*, jika ada ubahan wajib diisi dikosongkan, sistem akan memunculkan notifikasi isian tidak boleh kosong
	12. Jika pengecekan berhasil dan data valid lalu sistem akan menyimpan data yang diubah ke *database* dan menampilkan notifikasi berhasil

13. *Administrator*mencari data *branch* yang ingin dihapus	14. Sistem menampilkan hasil daftar pencarian
15. *Administrator* memilih data *branch* yang ingin dihapus	16. Sistem akan menampilkan pesan, apakah data *branch*ingin dihapus atau tidak
17. *Administrator* akan menekan tombol Ok	18. Sistem akan menghapus data yang diinginkan dan tampilan daftar *branch*akan menghilangkan data tersebut
19. *Administrator* mengisikan data branchyang dicari di *field* cari	20. Sistem akan mencari data *branch*tersebut, jika ditemukan akan ditampilkan dalam daftar *branch*tetapi jika tidak ditemukan sistem menampilkan pesan data tidak ditemukan
Kondisi Akhir	Data *branch*disistem sudah mengalami proses pengelolaan

4. *Use Case Scenario* Mengelola Data *User*

Interaksi antara aktor, yaitu *administrator* dengan *use case* mengelola data user dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.7*Use Case Scenario* Mengelola data *user*

Identifikasi	
Nama	Mengelola data *user*
Tujuan	Untuk mengelola data *user* / pengguna sistem
Deskripsi	Proses mengelola data *user*meliputi menambah, mengubah dan menghapus data *user*
Aktor	*Administrator*
Usecase Yang Berkaitan	-
Skenario Utama	
Kondisi Awal	Tampilan daftar data *user*
Aksi Aktor	Reaksi Sistem
1. *Administrator* memilih tombol tambah userbaru	2. Sistem akan menampilkan halaman isi tambah data userkepada *administrator*
3. *Administrator* dapat mengisikan	4. Sistem memeriksa data yang diisikan

data userbaru kemudian menekan tombol simpan	*administrator*, jika ada isian wajib diisi dikosongkan, sistem akan memunculkan notifikasi isian tidak boleh kosong
	5. Jika pengecekan berhasil dan data valid lalu sistem akan menyimpan ke *database* dan menampilkan notifikasi berhasil
6. *Administrator* mencari data user yang akan diubah	7. Sistem menampilkan hasil daftar pencarian
8. *Administrator* memilih data useryang ingin diubah	9. Sistem akan menampilkan halaman ubah *user*
10. *Administrator* mengisi data yang diubah dengan data *user*yang benar dan valid	11. Sistem memeriksa data yang diubah *administrator*, jika ada ubahan wajib diisi dikosongkan, sistem akan memunculkan notifikasi isian tidak boleh kosong
	12. Jika pengecekan berhasil dan data valid lalu sistem akan menyimpan data yang diubah ke *database* dan menampilkan notifikasi berhasil
13. *Administrator* mencari data *user* yang akan dihapus	14. Sistem menampilkan hasil daftar pencarian
15. *Administrator* memilih data *user* yang ingin dihapus	16. Sistem akan menampilkan pesan, apakah data useringin dihapus atau tidak
17. *Administrator* akan menekan tombol Ok	18. Sistem mengecek keaktifan login *user* tersebutjika sedang login akan menampilkan pesan *user* tidak bisa dihapus tetapi jika tidak user dapat dihapus dan halaman daftar userakan menghilangkan data tersebut
Kondisi Akhir	Data *user*disistem sudah mengalami proses pengelolaan

5. *Use Case Scenario*Melihat Data *User*

Interaksi antara aktor dengan *use case* melihat data user dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.8*Use Case Scenario* Melihat data *user*

Identifikasi		
Nama	Melihat data *user*	
Tujuan	Untuk melihat data *user* pada sistem	
Deskripsi	Proses untuk melihat dan mencari data *user* yang terdaftar pada sistem	
Aktor	*Administrator* dan *manager*	
Skenario Utama		
Kondisi Awal	Tampilan daftar *user*	
Aksi Aktor		Reaksi Sistem
1. Menentukan lihat data *user* yang dipilih		2. Sistem menampilkan rincian data *user* yang dipilih
3. *Administrator* mengisikan data user yang dicari di *field*cari		4. Sistem akan mencari data usertersebut, jika ditemukan akan ditampilkan dalam daftar usertetapi jika tidak ditemukan akan menampilkan pesan data tidak ditemukan
Kondisi Akhir	User dapat mengetahui informasi rincian user sesuai kebutuhannya	

6. *Use Case Scenario* Mengelola *Ticket*

Interaksi antara aktor, yaitu tim project dan tim quality dengan *use case* mengelola *ticket* dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.9*Use Case Scenario* Mengelola *ticket*

Identifikasi	
Nama	Mengelola *ticket*
Tujuan	Untuk mengelola *ticket*
Deskripsi	Proses mengelola *ticket*meliputi menambah, mengubah,

	menghapus, melihat dan mencari *ticket*	
Aktor	Tim *project* dan tim *quality*	
Usecase Yang Berkaitan	Memonitoring *ticket*, Konfirmasi *closed*, dan Mengelola action	
Skenario Utama		
Kondisi Awal	Tampilan daftar *ticket*	
Aksi Aktor		Reaksi Sistem
1. User memilih menu tambah ticketbaru sesuai jenis *ticket*		2. Sistem akan menampilkan halaman isi tambah *ticket*
3. User dapat mengisikan *ticket*baru kemudian menekan tombol simpan		4. Sistem memeriksa isian diisikan user, jika ada isian wajib diisi dikosongkan, sistem akan memunculkan notifikasi isian tidak boleh kosong
		5. Jika pengecekan berhasildan *ticket* valid lalu sistem akan menyimpan ke *database* dan menampilkan notifikasi berhasil
6. Administrator mencari data *ticket* yang akan diubah		7. Sistem menampilkan hasil daftar pencarian
8. User memilih tombol ubah pada daftar ticketyang diinginkan		9. Sistem akan menampilkan halaman ubah *ticket*sesuai *ticket* yang dipilih
10. User mengisi isian *ticket* yang diubah dengan data *ticket*yang benar dan valid		11. Sistem memeriksa isian yang diubahuser, jika ada ubahan wajib diisi dikosongkan, sistem akan memunculkan notifikasi isian tidak boleh kosong
		12. Jika pengecekan berhasi dan ticket valid lalu sistem akan menyimpan data yang diubah ke *database* dan menampilkan notifikasi berhasil
13. Administrator mencari data *ticket* yang akan dihapus		14. Sistem menampilkan hasil daftar pencarian
15. User memilih tombol hapus pada		16. Sistem akan menampilkan pesan, apakah

daftar *ticket* yang diinginkan	*ticket*ingin dihapus atau tidak
17. User akan menekan tombol Ok	18. Sistem akan menghapus data yang diinginkan di *database*dan tampilan daftar *ticket*akan menghilangkan *ticket* tersebut
19. User memilih tombol melihat rincian *ticket* pada daftar *ticket* yang diinginkan	20. Sistem akan menampilkan data rincian tersebut
21. User mengisikan *ticket*yang dicari di *field* cari	22. Sistem akan mencari *ticket*tersebut, jika ditemukan akan ditampilkan dalam daftar *ticket*tetapi jika tidak ditemukan akan menampilkan pesan data tidak ditemukan
Kondisi Akhir	Data *ticket*disistem sudah mengalami proses pengelolaan

7. *Use Case Scenario*Memonitoring *Ticket*

Interaksi antara aktor dengan *use case* memonitoring *ticket* dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.10*Use Case Scenario* Memonitoring *ticket*

Identifikasi	
Nama	Memonitoring *ticket*
Tujuan	Untuk melihat perkembangan pengerjaan ticket HCR
Deskripsi	Proses untuk memonitoringperkembangan pengerjaan data ticket HCR
Aktor	Administrator, manager, tim project, tim quality, dan tim access
Usecase Terkait	Konfirmasi closed dan mengelola *ticket*
Skenario Utama	
Kondisi Awal	Tampilan daftar *ticket*
Aksi Aktor	Reaksi Sistem
1. Memilih tombol kriteria monitoring	2. Sistem menampilkan form isian kriteria
3. Mengisikan kriteria monitoring	4. Sistem akan mencari *ticket*yang sesuai

yang dimaksud	kriteria tersebut, jika ditemukan akan ditampilkan dalam daftar *ticket*tetapi jika tidak ditemukan maka halaman daftar ticketakan menampilkan kosong
5. Menentukan lihat *ticket* yang dipilih	6. Sistem menampilkan rincian *ticket* yang dipilih
Kondisi Akhir	User dapat memonitoring ticket HCR sesuai kebutuhannya

8. *Use Case Scenario*Mengelola *Action*

Interaksi antara aktor, yaitu tim *quality* dan tim *acces* dengan *use case* mengelola *action* dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.11*Use Case Scenario* Mengelola *Action*

Identifikasi	
Nama	Mengelola action
Tujuan	Untuk mengelola action
Deskripsi	Proses mengelola actionmeliputi menambah, mengubah dan melihat
Aktor	Tim *project* dan tim *quality*
Usecase Yang Berkaitan	Memonitoring *ticket* dan Konfirmasi *closed*
Skenario Utama	
Kondisi Awal	Tampilan halaman daftar*ticket*yang belum diaction
Aksi Aktor	Reaksi Sistem
1. User memilih tombol *action* pada *ticket*yang dipilih	2. Sistem akan menampilkan halaman action*ticket*sesuai data yang dipilih
3. User mengisi halamanaction dan perkembangan pengerjaan *ticket* HCR	4. Sistem memeriksa *action*, jika ada isian*action* wajib diisi dikosongkan, sistem akan memunculkan notifikasi isian tidak boleh kosong
	5. Jika pengecekan berhasil, berarti *action* valid lalu sistem akan menyimpan *action* ke database dan menampilkan notifikasi

	berhasil
6. User memilih tombol ubah pada daftar action di *ticket* yang dipilih	7. Sistem akan menampilkan halaman ubahaction *ticket*sesuai *ticket* yang dipilih
8. User mengisi data ubah*action* dan perkembangan pengerjaanHCR yang benar dan valid	9. Sistem memeriksa data yang diubah user, jika ada ubahan wajib diisi dikosongkan, sistem akan memunculkan notifikasi isian tidak boleh kosong
	10. Jika pengecekan berhasil, berarti *action* valid lalu sistem akan menyimpan data yang diubah ke *database* dan menampilkan notifikasi berhasil
11. User memilih tombol melihat rincian*action ticket* pada daftar ticket yang diinginkan	12. Sistem akan menampilkan data rincian *action* pada *ticket* yang dipilih
Kondisi Akhir	Data *action ticket*disistem sudah mengalami proses pengelolaan

9. *Use Case Scenario*Konfirmasi *Closed*

Interaksi antara aktor dengan *use case* konfirmasi c*losedticket* dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.12*Use Case Scenario* Konfirmasi *Closed*

Identifikasi	
Nama	Konfirmasi *Closed*
Tujuan	Untuk konfirmasi pengecekan hasil *action* terakhir
Deskripsi	Proses untuk memberitahukan bahwa *ticket* yang di*action* sudah kembali ke performa semula / normal
Aktor	Tim *project* dan tim *quality*
Usecase Terkait	Mengelola *action*
Skenario Utama	
Kondisi Awal	Tampilan halaman daftar *ticket* yang sudah di*action*
Aksi Aktor	Reaksi Sistem

1. Memilih *ticket* yang akan di *closed* hasil analisis	2. Sistem akan menampilkan pesan, apakah ticketingin di*closed*
3. User akan menekan tombol Ok	4. Sistem akan merubah data *ticket* menjadi statusnya *closed* dan tampilan daftar *ticket*akan dihilangkan dari data daftar tersebut
Kondisi Akhir	*Ticket* berganti status dari *open* ke *closed*

10. *Use Case Scenario*Melihat Laporan

Interaksi antara aktor dengan *use case* melihat laporan dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.13*Use Case Scenario* Melihat laporan

Identifikasi	
Nama	Melihat Laporan
Tujuan	Untuk melihat laporan
Deskripsi	Proses untuk melihat dan mengekspor ke format excel data HCR
Aktor	Administrator, manager, tim project, tim quality dan tim access
Skenario Utama	
Kondisi Awal	Tampilan halaman laporan HCR
Aksi Aktor	Reaksi Sistem
1. Menentukan kriteria dan rentang waktu yang ingin ditampilkan	2. Sistem menampilkan data sesuai kriteria
3. Menekan tombol ekspor	4. Sistem akan menampilkan laporan dalam format excel
Kondisi Akhir	User / pengguna dapat mengolah laporan sesuai kebutuhan

11. *Use Case Scenario*Melihat Dashboard

Interaksi antara aktor dengan *use case* melihat *dashboard* dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.14*Use Case Scenario* Melihat *dashboard*

Identifikasi		
Nama	Melihat *Dashboard*	
Tujuan	Untuk melihat kesimpulan terbaruHCR dari sistem yang berjalan	
Deskripsi	Proses untuk menampilkan kesimpulan HCR menurut kriteria pengelompokan, penghitungan data HCR dan grafik sistem	
Aktor	Administrator, manager, tim project, tim quality dan tim access	
Skenario Utama		
Kondisi Awal	Tampilan halaman beranda dashboard	
Aksi Aktor		Reaksi Sistem
1. User memilih menu kriteria dashboard		2. Sistem akan menampilkan halaman dashboard dengandata terkait dengan kriteria tersebut
Kondisi Akhir	User / pengguna dapat mengetahui perkembangan HCR	

12. *Use Case Scenario*Merubah *Password*

Interaksi antara aktor dengan *use case* merubah *password* dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.15*Use Case Scenario* Merubah *password*

Identifikasi	
Nama	Merubah*Password*
Tujuan	Untuk merubah kata sandi / *password* masuk sistem
Deskripsi	Proses untuk merubah *password* lama menjadi *password* baru, dengan tujuan keamanan akses
Aktor	*Administrator*, *manager*, tim *project*, tim *quality* dan tim *access*

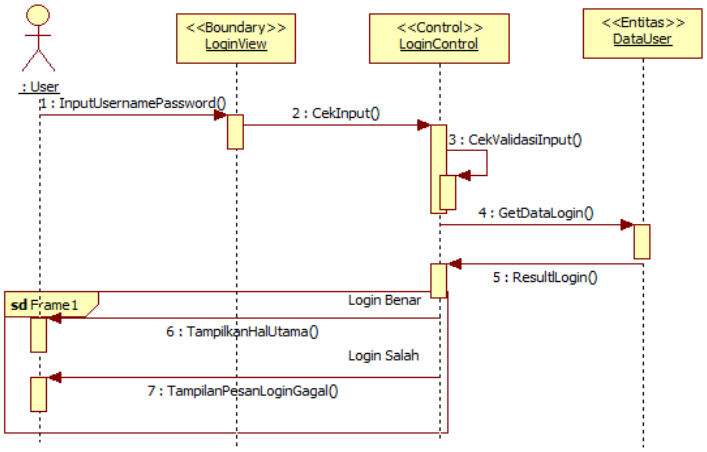
Skenario Utama	
Kondisi Awal	Setelah login sistem
Aksi Aktor	Reaksi Sistem
1. User memilih menu ganti *password*	2. Sistem akan menampilkan halaman ganti *password*
3. User mengisikan isian halaman, lalu tekan simpan	4. Sistem akan mengecek data isian kosong, kecocokan *password* lama, kesamaan pengisian *password* baru dan konfirmasi *password*, jika ada yang tidak sesuai sistem menampilkan pesan
	5. Sistem akan menyimpan data *password* baru ke database dan ganti *password* berhasil
Kondisi Akhir	Data *password* user / pengguna mengalami perubahan

3.1.4.3 *SequenceDiagram*

Sequence diagram menjelaskan secara detil urutan proses yangdilakukan dalam sistem untuk mencapai tujuan dari use case yaitu : interaksi yangterjadi antar class, operasi apa saja yang terlibat, urutan antar operasi, daninformasi yang diperlukan oleh masing-masing operasi.Berikut penjelasan dari masing-masing *sequence*diagram :

1. ***Sequence Diagram* Login**

Interaksi antara aktor penggunadengan *use case* login dijelaskan dalam *Sequence Diagram* gambar 3.6.



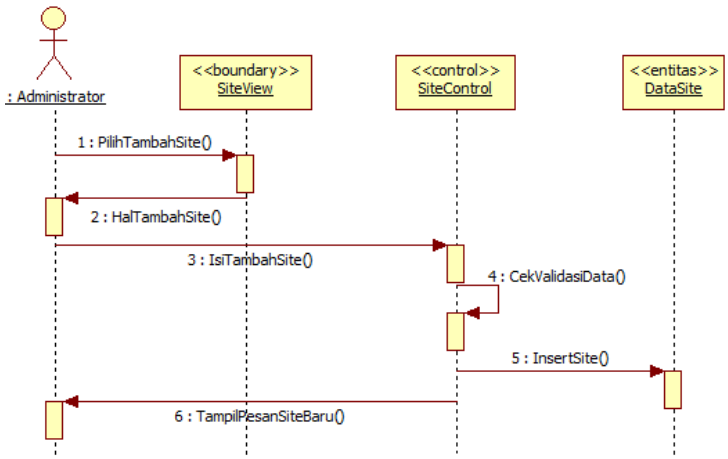
Gambar 3.6*Sequence Diagram* Login

2. Sequence Diagram Mengelola Data Site

Interaksi antara aktor pengguna dengan*use case* mengelola data site dijelaskan dalam *Sequence Diagram* tambah data site, ubah data site, hapus data site dan cari data site. Berikut penjelasan dari masing-masing *Sequence Diagram*:

2.a. *Sequence Diagram*Tambah Data Site

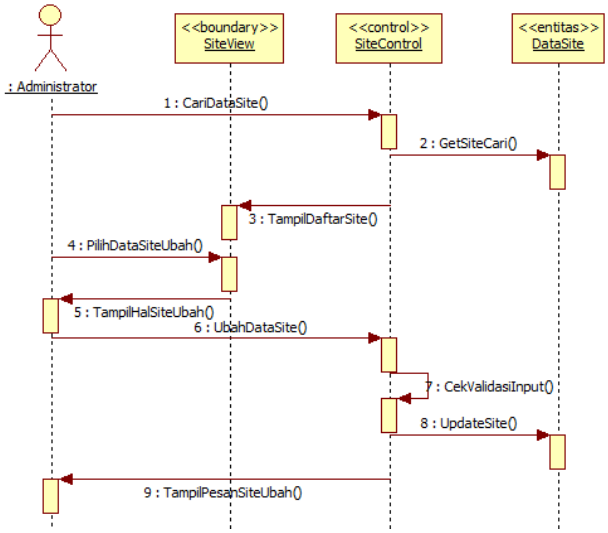
Interaksi antara aktor pengguna dengan *use case* tambah data site dijelaskan dalam *Sequence Diagram* gambar 3.7.



Gambar 3.7*Sequence Diagram* Tambah Data Site

2.b. *Sequence Diagram*Ubah Data Site

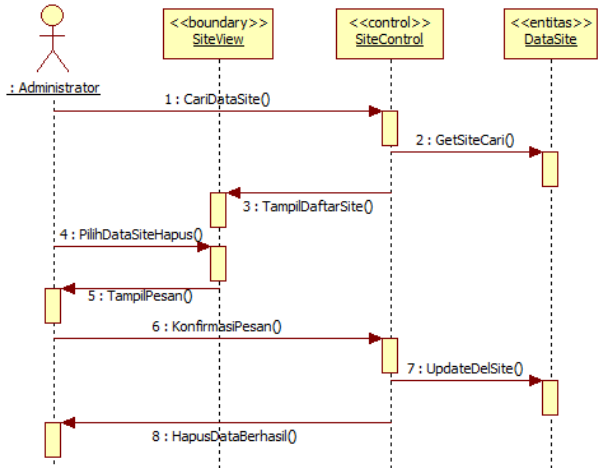
Interaksi antara aktor pengguna dengan *use case* ubah data site dijelaskan dalam *Sequence Diagram* gambar 3.8.



Gambar 3.8*Sequence Diagram* Ubah Data Site

2.c. *Sequence Diagram*Hapus Data Site

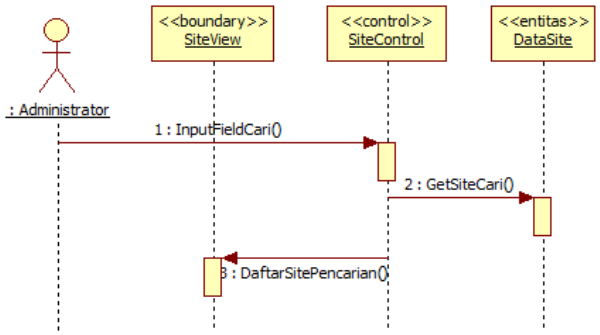
Interaksi antara aktor pengguna dengan *use case* hapus data site dijelaskan dalam *Sequence Diagram* gambar 3.9.



Gambar 3.9*Sequence Diagram* Hapus Data Site

2.d. *Sequence Diagram*Cari Data Site

Interaksi antara aktor penggunadengan *use case* cari data site dijelaskan dalam *Sequence Diagram* gambar 3.10.



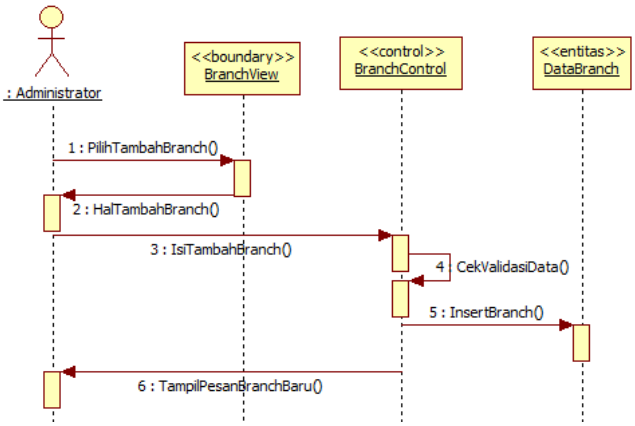
Gambar 3.10*Sequence Diagram* Cari Data Site

3. Sequence Diagram Mengelola Data Branch

Interaksi antara aktor penggunadengan *use case* mengelola data branch dijelaskan dalam *Sequence Diagram* tambah data branch, ubah data branch, hapus data branch dan cari data branch. Berikut penjelasan dari masing-masing *Sequence Diagram*:

3.a. *Sequence Diagram*Tambah Data Branch

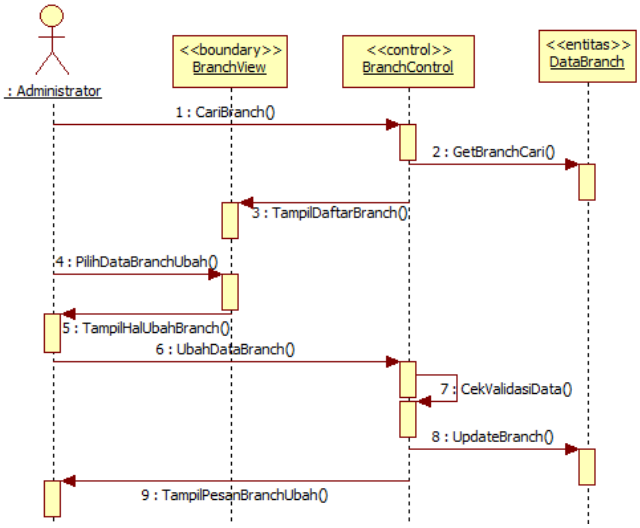
Interaksi antara aktor pengguna dengan *use case* tambah data branch dijelaskan dalam *Sequence Diagram* gambar 3.11.



Gambar 3.11*Sequence Diagram* Tambah Data Branch

3.b. *Sequence Diagram*Ubah Data Branch

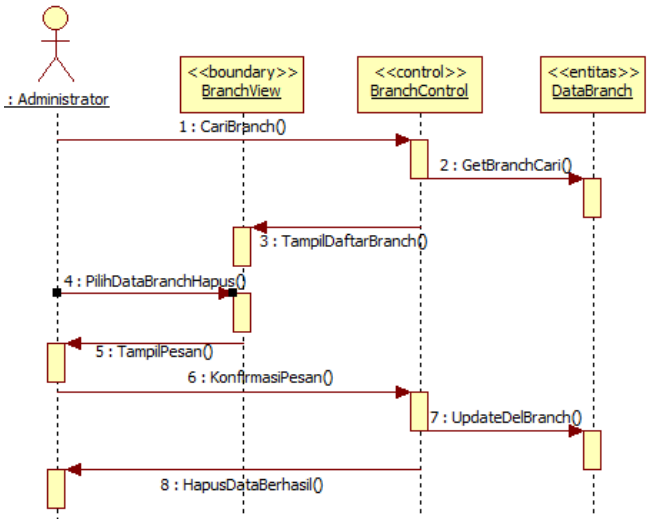
Interaksi antara aktor penggunadengan *use case* ubah data branch dijelaskan dalam *Sequence Diagram* gambar 3.12.



Gambar 3.12*Sequence Diagram* Ubah Data Branch

3.c. *Sequence Diagram*Hapus Data Branch

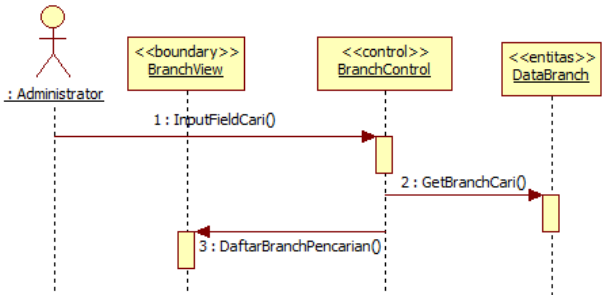
Interaksi antara aktor pengguna dengan *use case* hapus data branch dijelaskan dalam *Sequence Diagram* gambar 3.13.



Gambar 3.13*Sequence Diagram*Hapus Data Branch

3.d.*Sequence Diagram*Cari Data Branch

Interaksi antara aktor penggunadengan *use case* cari data branch dijelaskan dalam *Sequence Diagram* gambar 3.14.



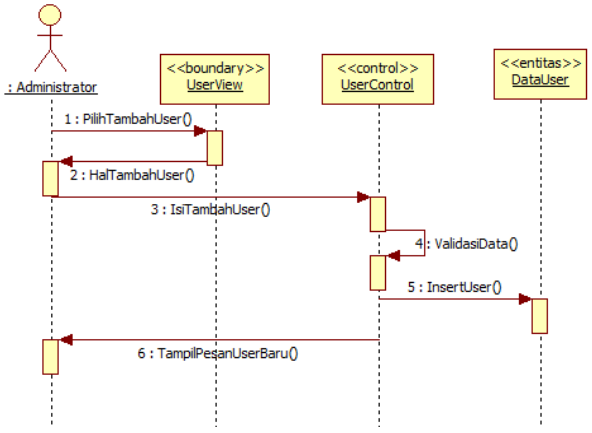
Gambar 3.14*Sequence Diagram* Cari Data Branch

4. *Sequence Diagram* Mengelola Data User

Interaksi antara aktor penggunadengan *use case* mengelola data user dijelaskan dalam *Sequence Diagram* tambah data user, ubah data user dan hapus data user. Berikut penjelasan dari masing-masing *Sequence Diagram*:

4.a.*Sequence Diagram*Tambah Data User

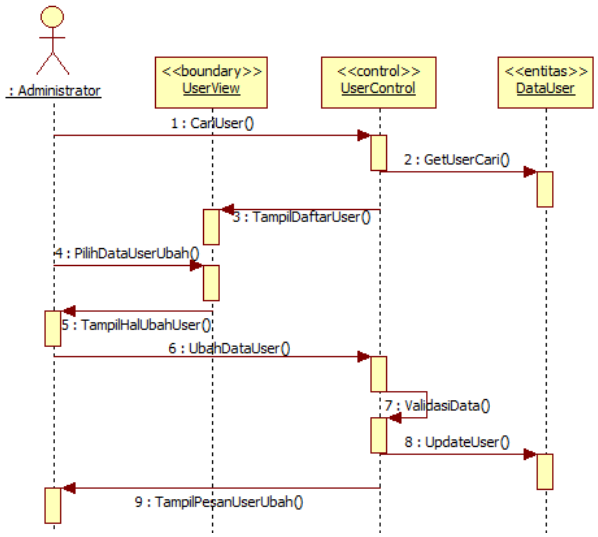
Interaksi antara aktor pengguna dengan *use case* tambah data user dijelaskan dalam *Sequence Diagram* gambar 3.15.



Gambar 3.15*Sequence Diagram* Tambah Data User

4.b.*Sequence Diagram*Ubah Data User

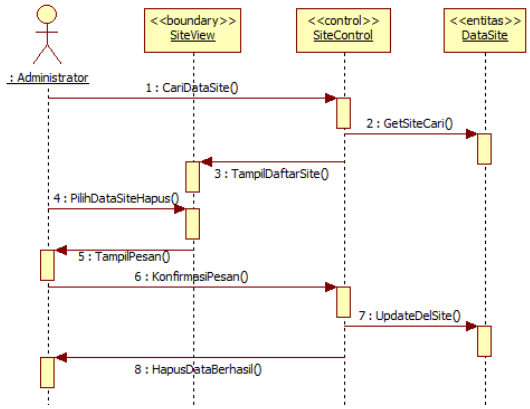
Interaksi antara aktor penggunadengan *use case* ubah data user dijelaskan dalam *Sequence Diagram* gambar 3.16.



Gambar 3.16*Sequence Diagram* Ubah Data User

4.c. *Sequence Diagram*Hapus Data User

Interaksi antara aktor pengguna dengan *use case* hapus data user dijelaskan dalam *Sequence Diagram* gambar 3.17.



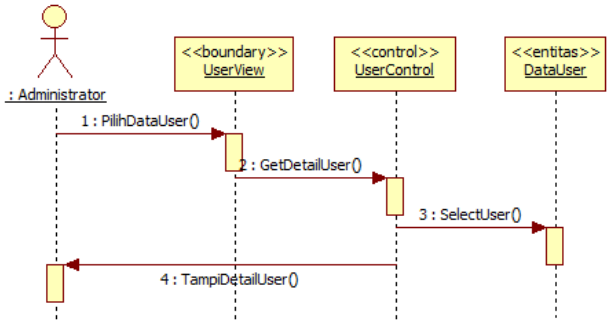
Gambar 3.17*Sequence Diagram* HapusData User

5. Sequence Diagram Melihat Data User

Interaksi antara aktor penggunadengan *use case* melihat data user dijelaskan dalam *Sequence Diagram* lihat data user dan cari data user. Berikut penjelasan dari masing-masing *Sequence Diagram*:

5.a.*Sequence Diagram*Lihat Data User

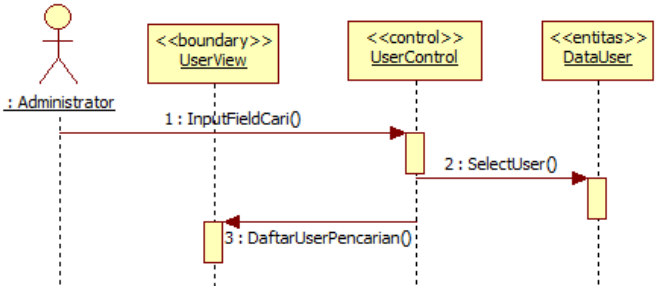
Interaksi antara aktor penggunadengan *use case* lihat data user dijelaskan dalam *Sequence Diagram* gambar 3.18.



Gambar 3.18*Sequence Diagram* Lihat Data User

5.b. *Sequence Diagram*Cari Data User

Interaksi antara aktor penggunadengan *use case* cari data user dijelaskan dalam *Sequence Diagram* gambar 3.19.



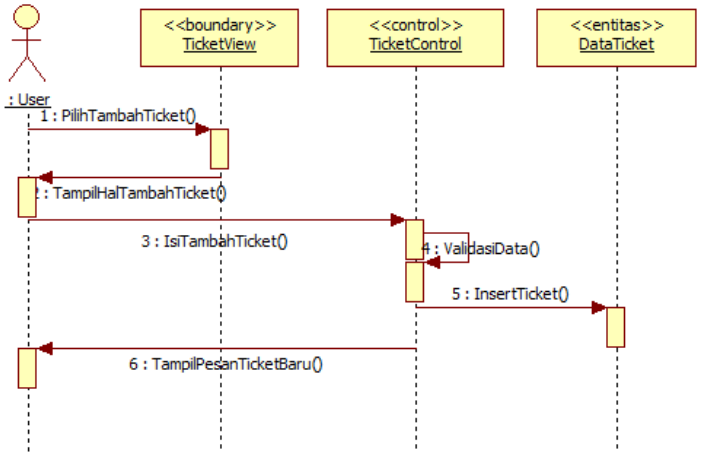
Gambar 3.19*Sequence Diagram*Cari Data User

6. *Sequence Diagram*Mengelola Ticket

Interaksi antara aktor penggunadengan *use case* mengelola ticket dijelaskan dalam *Sequence Diagram* tambah ticket, ubah ticket, hapus ticket, lihat ticket dan cari ticket. Berikut penjelasan dari masing-masing *Sequence Diagram*:

6.a. *Sequence Diagram*Tambah Ticket

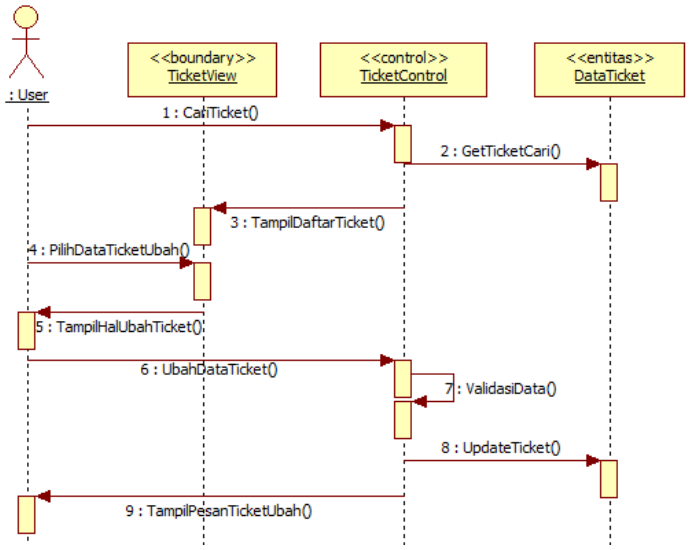
Interaksi antara aktor pengguna dengan *use case* tambahticket dijelaskan dalam *Sequence Diagram* gambar 3.20.



Gambar 3.20*Sequence Diagram* Tambah Ticket

6.b. *Sequence Diagram*Ubah Ticket

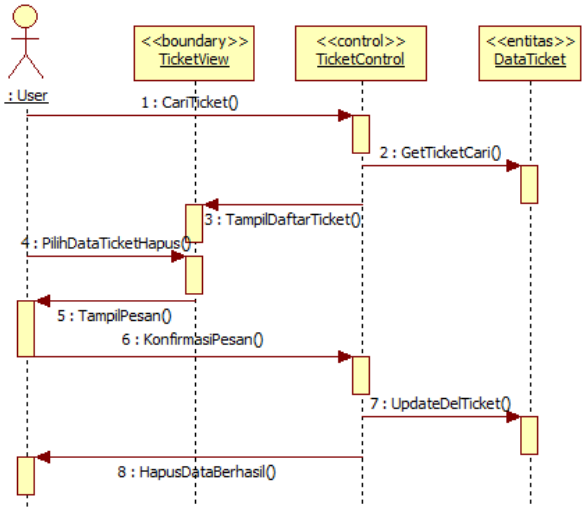
Interaksi antara aktor pengguna dengan *use case* ubah ticket dijelaskan dalam *Sequence Diagram* gambar 3.21.



Gambar 3.21*Sequence Diagram* Ubah Ticket

6.c. *Sequence Diagram*Hapus Ticket

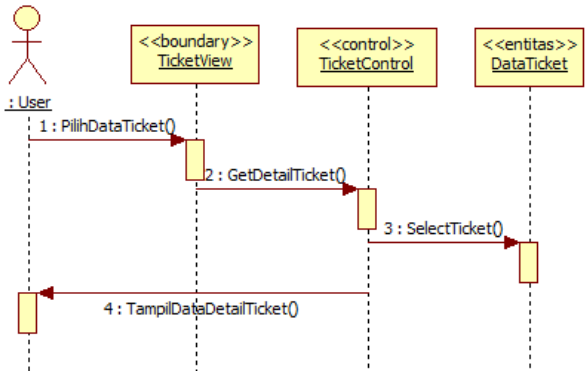
Interaksi antara aktor pengguna dengan *use case* hapus ticket dijelaskan dalam *Sequence Diagram* gambar 3.22.



Gambar 3.22*Sequence Diagram* Hapus Ticket

6.d. *Sequence Diagram*Lihat Ticket

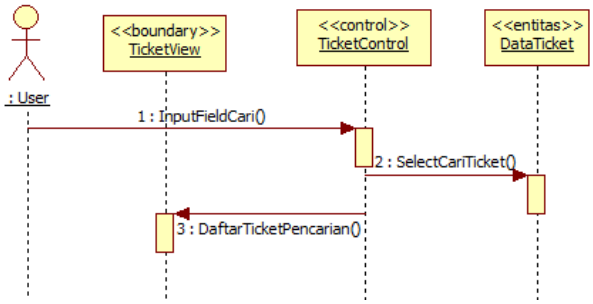
Interaksi antara aktor pengguna dengan *use case* lihat ticket dijelaskan dalam *Sequence Diagram* gambar 3.23.



Gambar 3.23*Sequence Diagram* Lihat Ticket

6.e. *Sequence Diagram*Cari Ticket

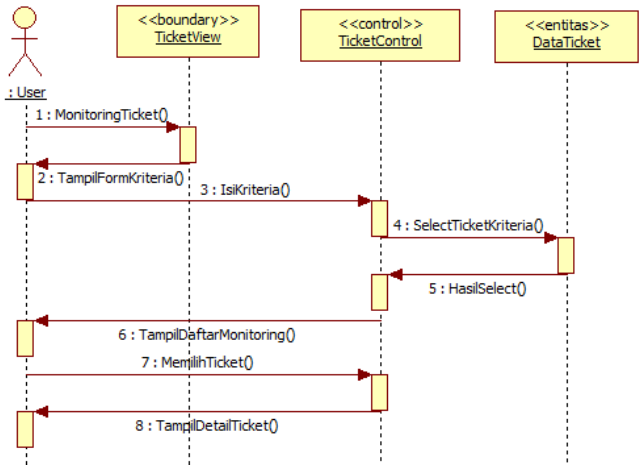
Interaksi antara aktor pengguna dengan *use case* cari data *ticket* dijelaskan dalam *Sequence Diagram* gambar 3.24.



Gambar 3.24*Sequence Diagram* Cari Ticket

7. *Sequence Diagram*Memonitoring Ticket

Interaksi antara aktor pengguna dengan *use case* monitoring ticket dijelaskan dalam *Sequence Diagram* gambar 3.25.



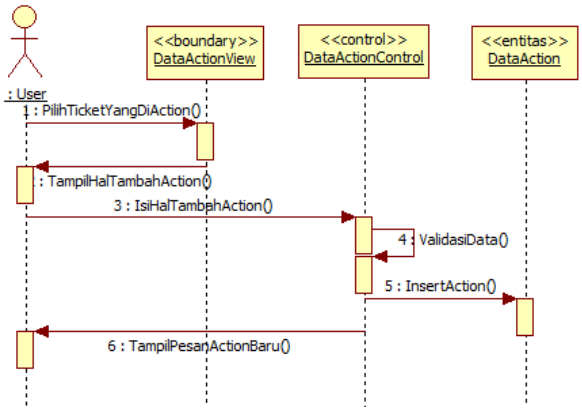
Gambar 3.25*Sequence Diagram* Monitoring Ticket

8. *Sequence Diagram*MengelolaAction

Interaksi antara aktor pengguna dengan *use case* mengelola action dijelaskan dalam *Sequence Diagram* tambah action, ubah action dan lihat action. Berikut penjelasan dari masing-masing *Sequence Diagram*:

8.a.*Sequence Diagram*TambahAction

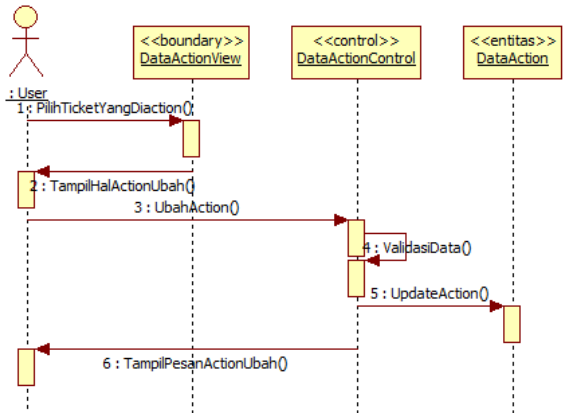
Interaksi antara aktor pengguna dengan *use case* tambahaction dijelaskan dalam *Sequence Diagram* gambar 3.26.



Gambar 3.26*Sequence Diagram* Tambah Action

8.b. *Sequence Diagram*Ubah Action

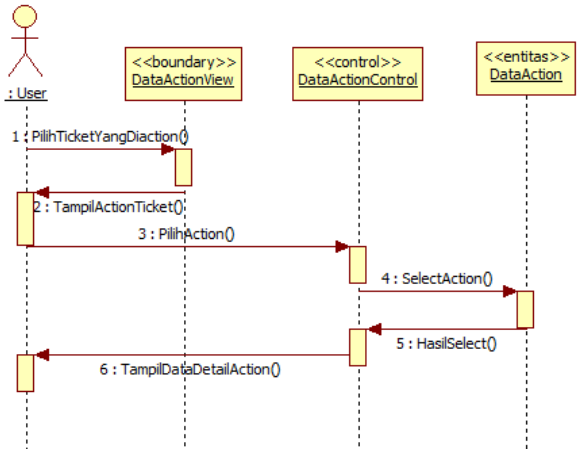
Interaksi antara aktor pengguna dengan *use case* ubah action dijelaskan dalam *Sequence Diagram* gambar 3.27.



Gambar 3.27*Sequence Diagram* Ubah Action

8.c.*Sequence Diagram*Lihat Action

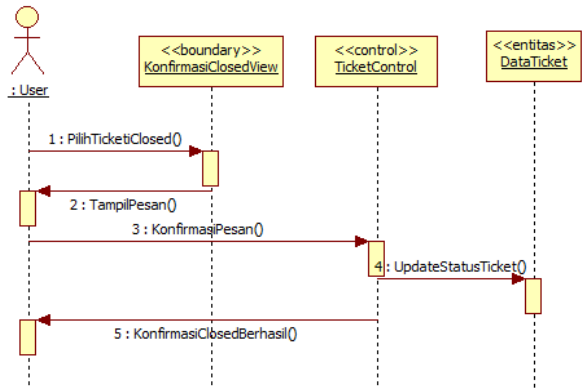
Interaksi antara aktor pengguna dengan *use case* lihat action dijelaskan dalam *Sequence Diagram* gambar 3.28.



Gambar 3.28*Sequence Diagram* Lihat Action

9. *Sequence Diagram*Konfirmasi *Closed*

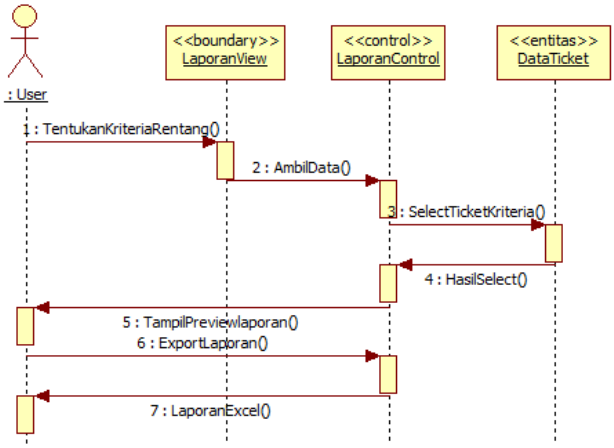
Interaksi antara aktor pengguna dengan *use case* konfirmasi*closed* dijelaskan dalam *Sequence Diagram* gambar 3.29.



Gambar 3.29*Sequence Diagram*Konfirmasi*Closed*

10. *Sequence Diagram*Lihat Laporan

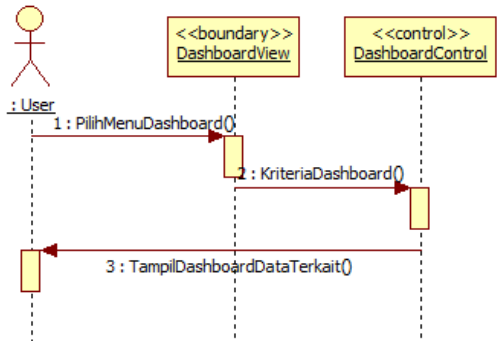
Interaksi antara aktor pengguna dengan *use case* lihat laporan dijelaskan dalam *Sequence Diagram* gambar 3.30.



Gambar 3.30*Sequence Diagram* Lihat Laporan

11. *Sequence Diagram*Melihat Dashboard

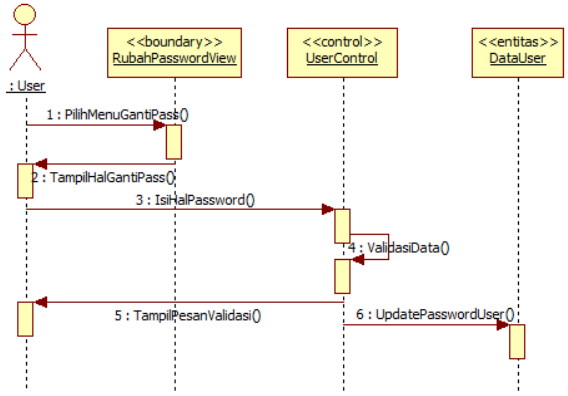
Interaksi antara aktor pengguna dengan *use case* melihat dashboard dijelaskan dalam *Sequence Diagram* gambar 3.31.



Gambar 3.31*Sequence Diagram* Melihat Dashboard

12. *Sequence Diagram*Merubah Password

Interaksi antara aktor pengguna dengan *use case* merubah password dijelaskan dalam *Sequence Diagram* gambar 3.32.



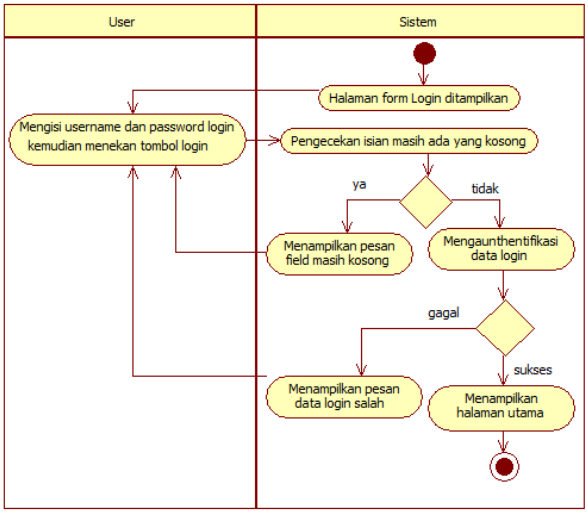
Gambar 3.32*Sequence Diagram* Merubah Password

3.1.4.4 *Activity Diagram*

Activity Diagram memodelkan aliran kerja atau *workflow* dari urutan aktifitas dalam suatu proses yang mengacu pada *Use Case Diagram* yang ada.Berikut penjelasan dengan *Activity Diagram* untuk penjelasan dari masing-masing *Use Case Scenario*:

1. *Activity Diagram* Login User

Interaksi antara aktor pengguna dengan *use case* login dijelaskan dalam *Activity Diagram* gambar 3.33.



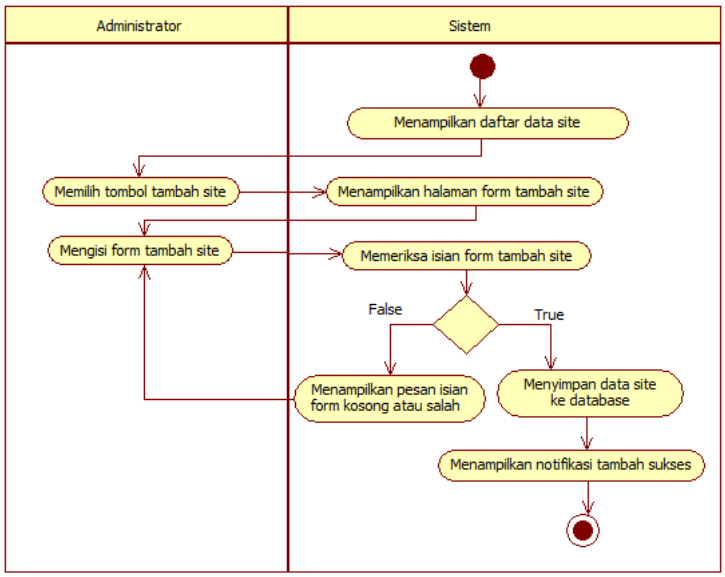
Gambar 3.33*Activity Diagram* Login User

2. *Activity Diagram*Mengelola Data Site

Interaksi antara aktor pengguna dengan *use case* mengelola data site dijelaskan dalam *Activity Diagram* tambah data site, ubah data site, hapus data site dan cari data site. Berikut penjelasan dari masing-masing *ActivityDiagram*:

2.a. *Activity Diagram*Tambah Data Site

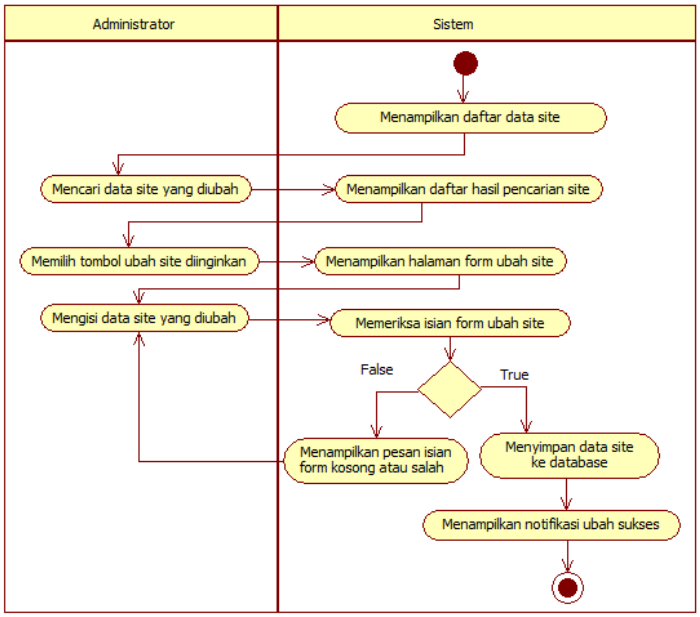
Interaksi antara aktor pengguna dengan *use case*tambah data site dijelaskan dalam *Activity Diagram* gambar 3.34.



Gambar 3.34*Activity Diagram*Tambah Data Site

2.b.*Activity Diagram*Ubah Data Site

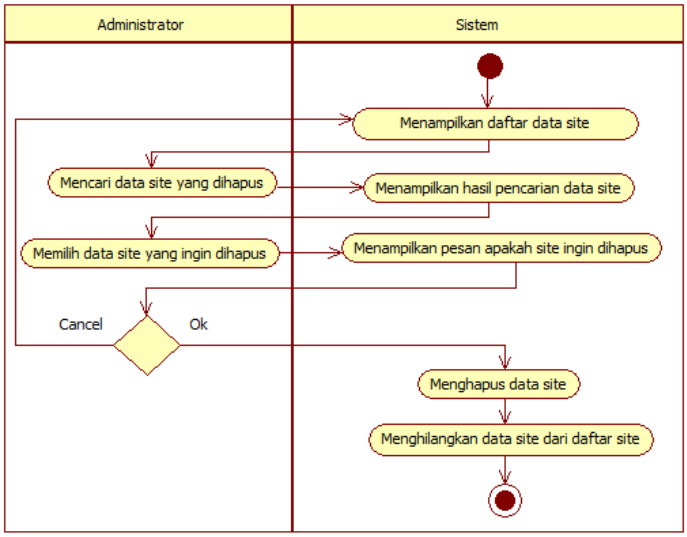
Interaksi antara aktor pengguna dengan *use case* ubah data site dijelaskan dalam *Activity Diagram* gambar 3.35.



Gambar 3.35*Activity Diagram* Ubah Data Site

2.c.*Activity Diagram*Hapus Data Site

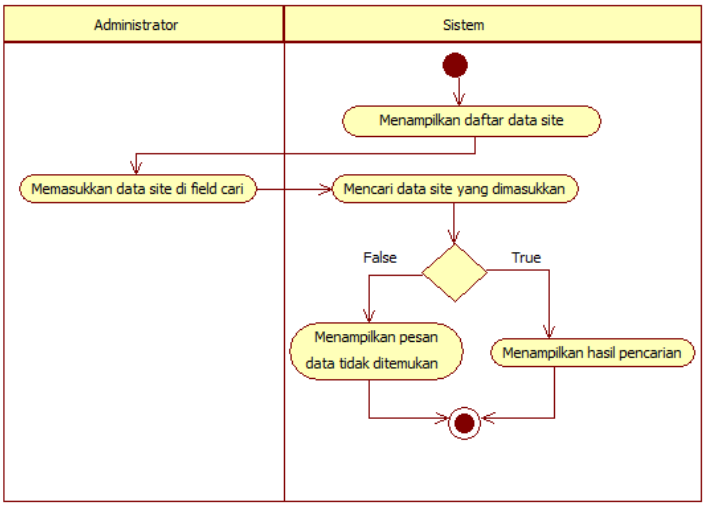
Interaksi antara aktor pengguna dengan *use case* hapus data site dijelaskan dalam *Activity Diagram* gambar 3.36.



Gambar 3.36*Activity Diagram* Hapus Data Site

2.d. *Activity Diagram*Cari Data Site

Interaksi antara aktor pengguna dengan *use case* cari data site dijelaskan dalam *Activity Diagram* gambar 3.37.



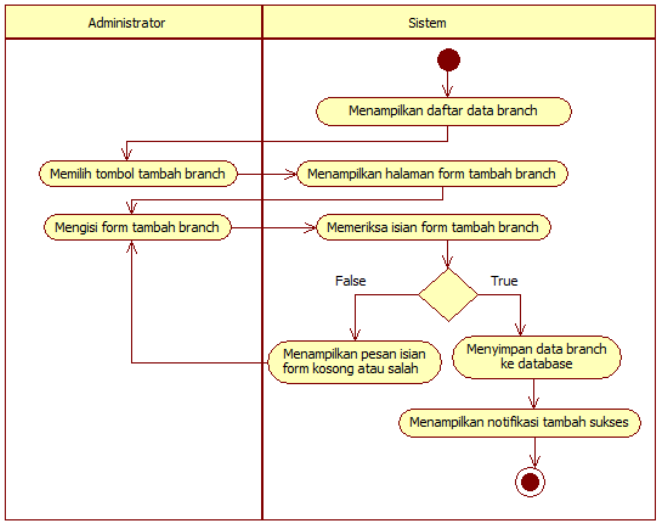
Gambar 3.37*Activity Diagram* Cari Data Site

3. *Activity Diagram*MengelolaData Branch

Interaksi antara aktor pengguna dengan *use case* mengelola data branch dijelaskan dalam *Activity Diagram* tambah data branch, ubah data branch, hapus data branch dan cari data branch. Berikut penjelasan dari masing-masing *Activity Diagram*:

3.a.*Activity Diagram* Tambah Data Branch

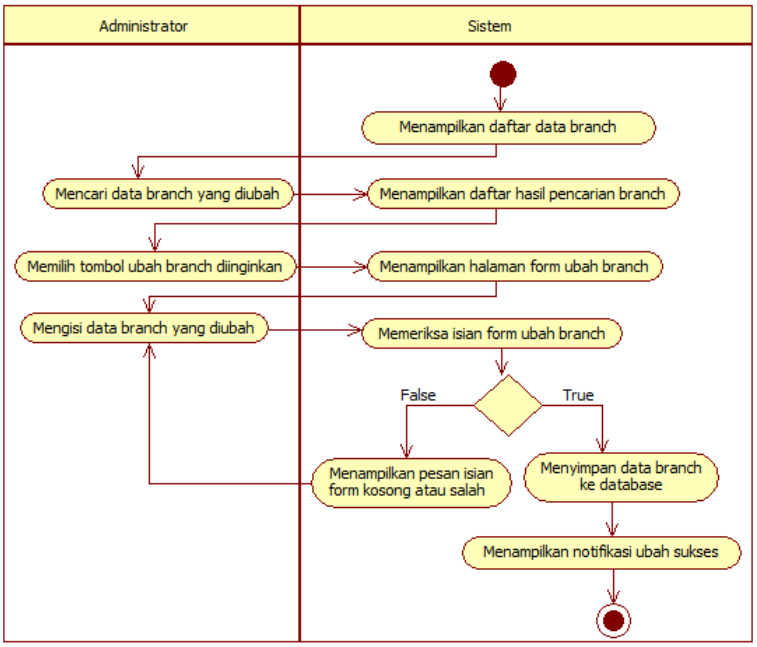
Interaksi antara aktor pengguna dengan *use case* tambah data branch dijelaskan dalam *Activity Diagram* gambar 3.38.



Gambar 3.38*Activity Diagram* Tambah Data Branch

3.b.*Activity Diagram* Ubah Data Branch

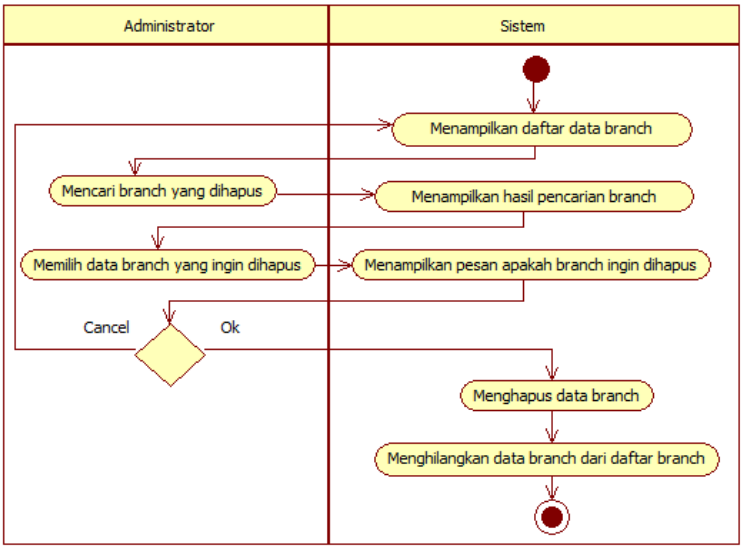
Interaksi antara aktor pengguna dengan *use case* ubah data branch dijelaskan dalam *Activity Diagram* gambar 3.39.



Gambar 3.39*Activity Diagram* Ubah Data Branch

3.c.*Activity Diagram* Hapus Data Branch

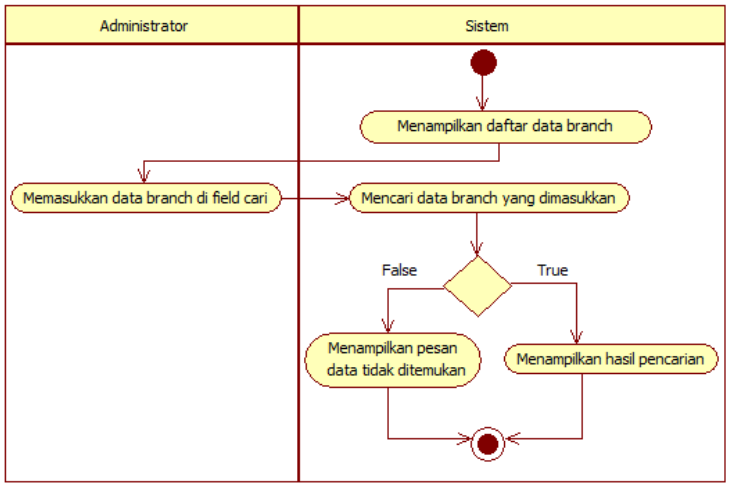
Interaksi antara aktor pengguna dengan *use case* hapus data branch dijelaskan dalam *Activity Diagram* gambar 3.40.



Gambar 3.40*Activity Diagram* Hapus Data Branch

3.d/*Activity Diagram* Cari Data Branch

Interaksi antara aktor pengguna dengan *use case* cari data branch dijelaskan dalam *Activity Diagram* gambar 3.41.



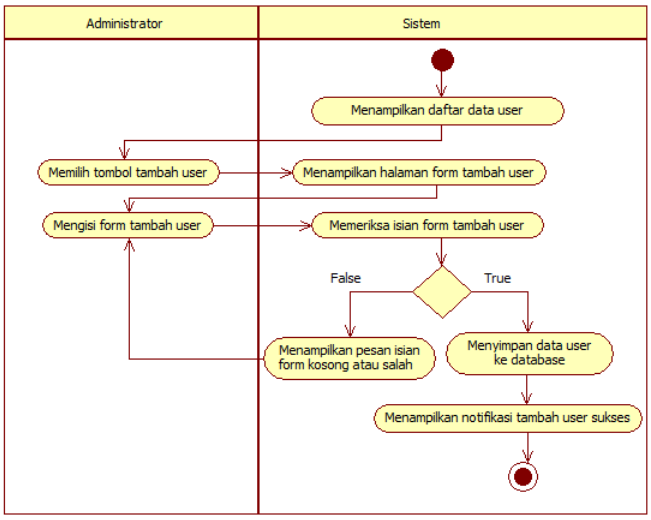
Gambar 3.41*Activity Diagram*Cari Data Branch

4. *Activity Diagram*TambahData User

Interaksi antara aktor pengguna dengan *use case* mengelola data user dijelaskan dalam *Activity Diagram* tambah data user, ubah data user, hapus data user dan cari data branch. Berikut penjelasan dari masing-masing *Activity Diagram*:

4.a.Activity Diagram Tambah Data User

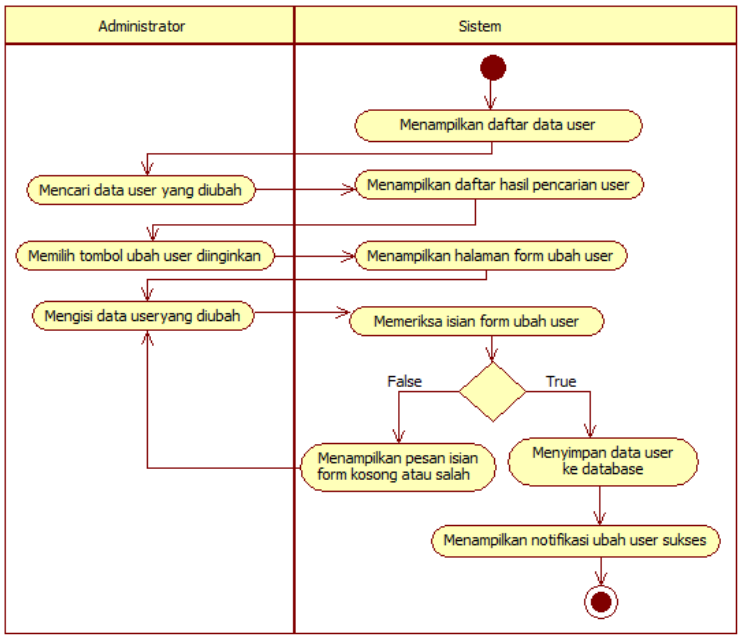
Interaksi antara aktor pengguna dengan *use case* tambah data user dijelaskan dalam *Activity Diagram* gambar 3.42.



Gambar 3.42*Activity Diagram* Tambah Data User

4.b. *Activity Diagram* Ubah Data User

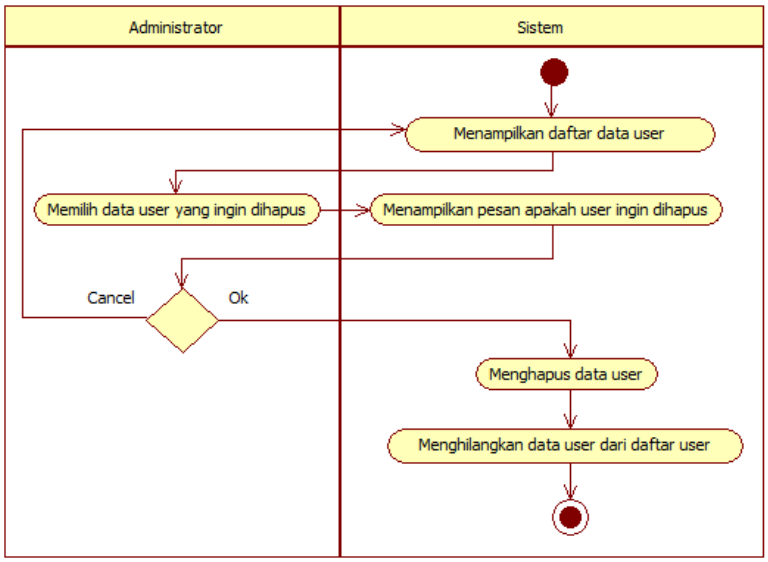
Interaksi antara aktor pengguna dengan *use case* ubah data user dijelaskan dalam *Activity Diagram* gambar 3.43.



Gambar 3.43*Activity Diagram* Ubah Data User

4.c. *Activity Diagram*Hapus DataUser

Interaksi antara aktor pengguna dengan *use case* hapus data user dijelaskan dalam *Activity Diagram* gambar 3.44.



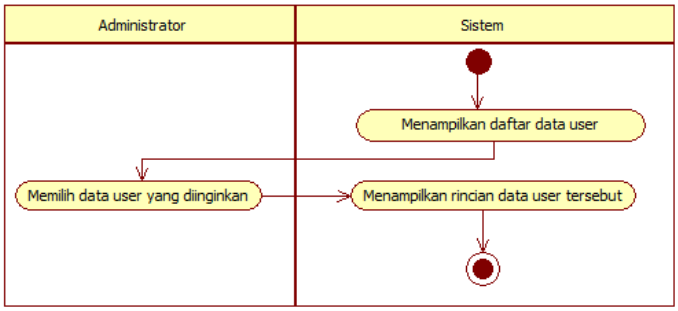
Gambar 3.44*Activity Diagram* Hapus Data User

5. *Activity Diagram*Melihat Data User

Interaksi antara aktor penggunadengan *use case* melihat data user dijelaskan dalam *Activity Diagram* lihat data user dan cari data user. Berikut penjelasan dari masing-masing *Activity Diagram*:

5.a. *Activity Diagram* Lihat Data User

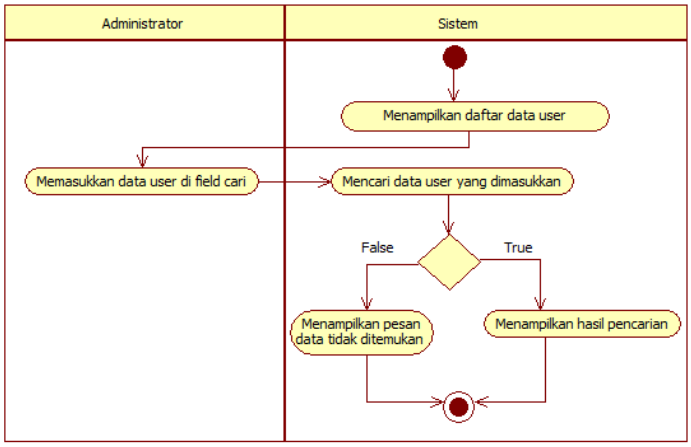
Interaksi antara aktor pengguna dengan *use case* lihat data user dijelaskan dalam *Activity Diagram* gambar 3.45.



Gambar 3.45*Activity Diagram* Lihat Data User

5.b. *Activity Diagram*Cari Data User

Interaksi antara aktor pengguna dengan *use case* cari data user dijelaskan dalam *Activity Diagram* gambar 3.46.



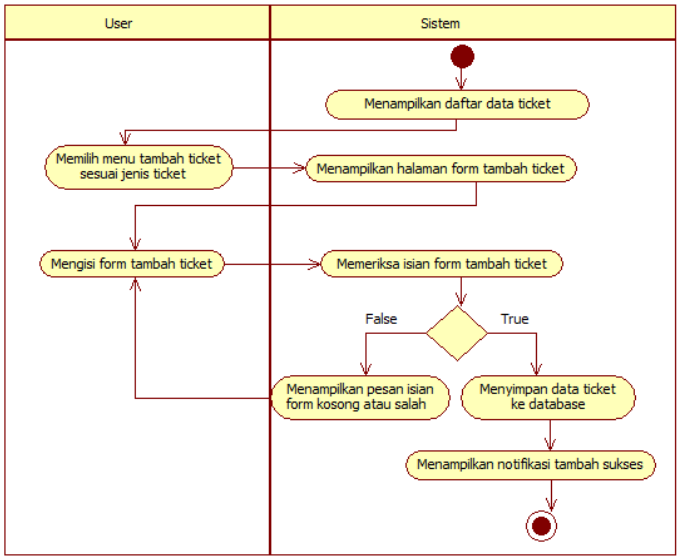
Gambar 3.46*Activity Diagram*Cari Data User

6. *Activity Diagram*Mengelola Ticket

Interaksi antara aktor pengguna dengan *use case* mengelola ticket dijelaskan dalam *Activity Diagram* tambah ticket, ubah ticket, hapus ticket, lihat ticket dan cari ticket. Berikut penjelasan dari masing-masing *Activity Diagram*:

6.a. *Activity Diagram* Tambah Ticket

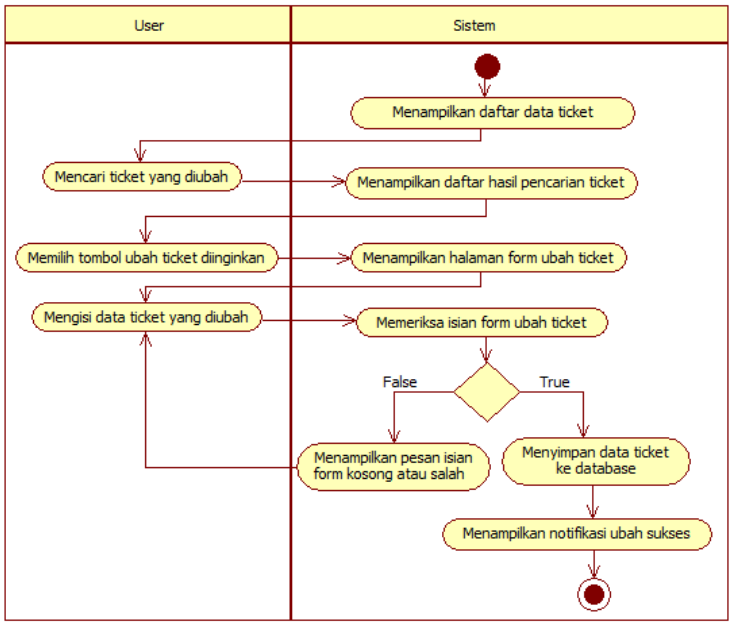
Interaksi antara aktor pengguna dengan *use case* tambahticket dijelaskan dalam *Activity Diagram* gambar 3.47.



Gambar 3.47*Activity Diagram* Tambah Ticket

6.b. *Activity Diagram* Ubah Ticket

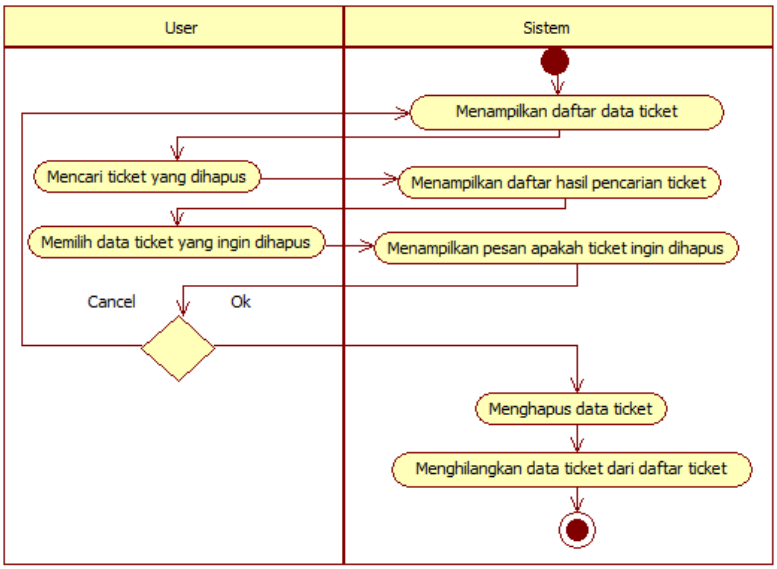
Interaksi antara aktor pengguna dengan *use case* ubahticket dijelaskan dalam *Activity Diagram* gambar 3.48.



Gambar 3.48*Activity Diagram* Ubah Ticket

6.c. *Activity Diagram*Hapus Ticket

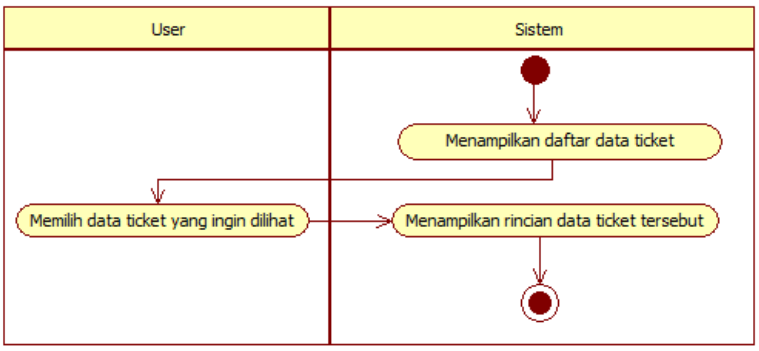
Interaksi antara aktor pengguna dengan *use case* hapusticket dijelaskan dalam *Activity Diagram* gambar 3.49.



Gambar 3.49*Activity Diagram* Hapus Ticket

6.d. *Activity Diagram*Lihat Ticket

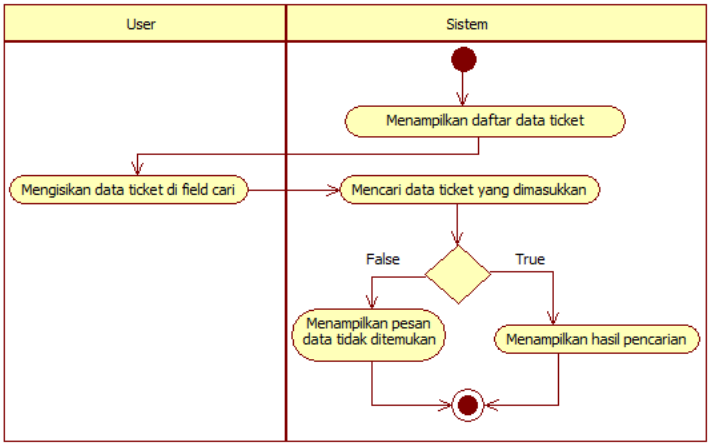
Interaksi antara aktor pengguna dengan *use case* lihat ticket dijelaskan dalam *Activity Diagram* gambar 3.50.



Gambar 3.50*Activity Diagram* Lihat Ticket

6.e. *Activity Diagram*Cari Ticket

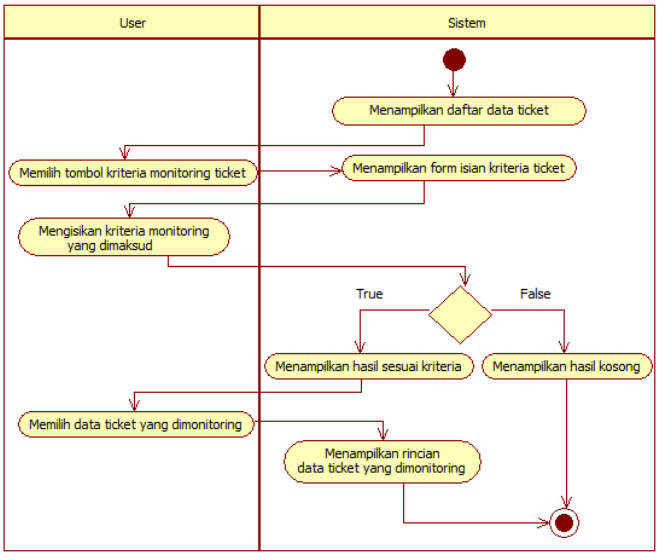
Interaksi antara aktor pengguna dengan *use case* cari ticket dijelaskan dalam *Activity Diagram* gambar 3.51.



Gambar 3.51*Activity Diagram* Cari Ticket

7. *Activity Diagram*Memonitoring Ticket

Interaksi antara aktor pengguna dengan *use case me*monitoring ticket dijelaskan dalam *Activity Diagram* gambar 3.52.



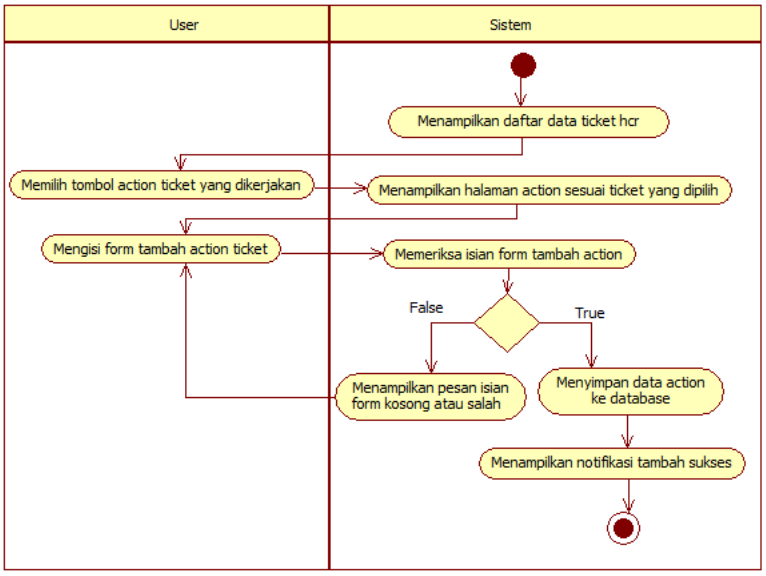
Gambar 3.52*Activity Diagram*Memonitoring Ticket

8. *Activity Diagram*MengelolaAction

Interaksi antara aktor pengguna dengan *use case* mengelola action dijelaskan dalam *Sequence Diagram* tambah action, ubah action dan lihat action. Berikut penjelasan dari masing-masing *Sequence Diagram*:

8.a. *Activity Diagram* Tambah Action

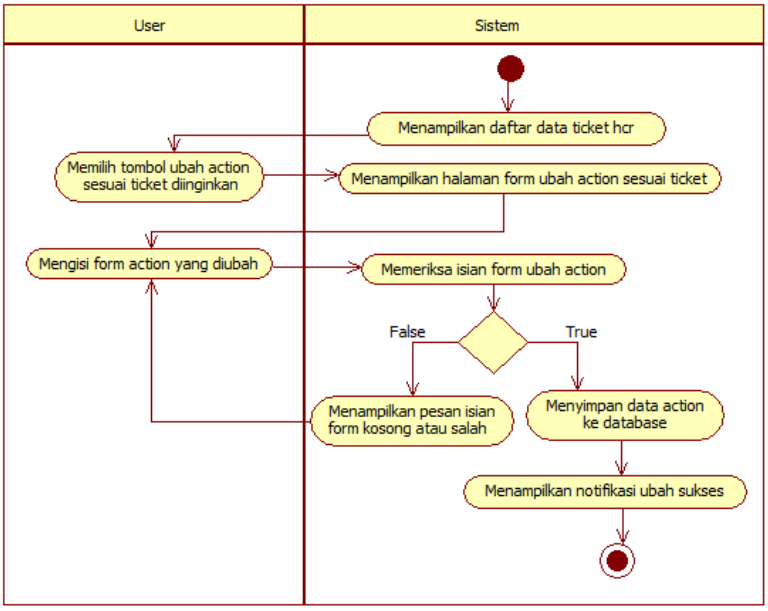
Interaksi antara aktor pengguna dengan *use case* tambah action dijelaskan dalam *Activity Diagram* gambar 3.53.



Gambar 3.53*Activity Diagram* Tambah Action

8.b. *Activity Diagram* Ubah Action

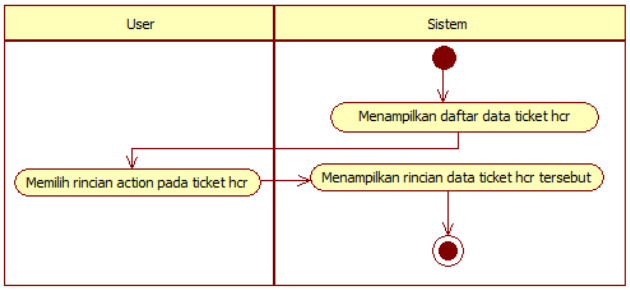
Interaksi antara aktor pengguna dengan *use case* ubah action dijelaskan dalam *Activity Diagram* gambar 3.54.



Gambar 3.54*Activity Diagram* Ubah Action

8.c.*Activity Diagram* Lihat Action

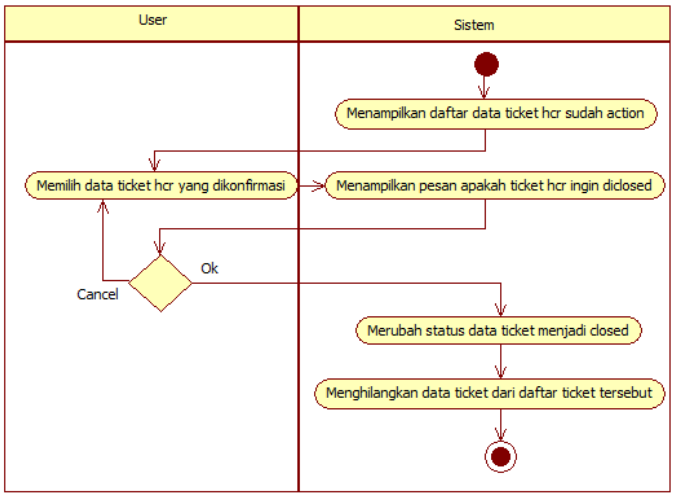
Interaksi antara aktor pengguna dengan *use case* lihat action dijelaskan dalam *Activity Diagram* gambar 3.55.



Gambar 3.55*Activity Diagram* Lihat Action

9. *Activity Diagram* Konfirmasi *Closed*

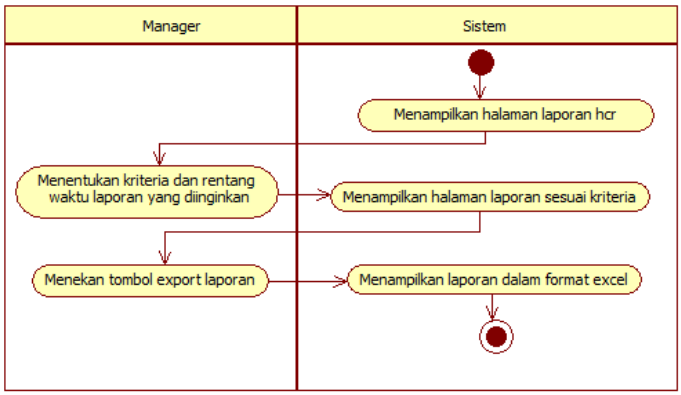
Interaksi antara aktor pengguna dengan *use case* konfirmasi *closed*dijelaskan dalam *Activity Diagram* gambar 3.56.



Gambar 3.56*Activity Diagram* Konfirmasi *Closed*

10. *Activity Diagram* Lihat Laporan

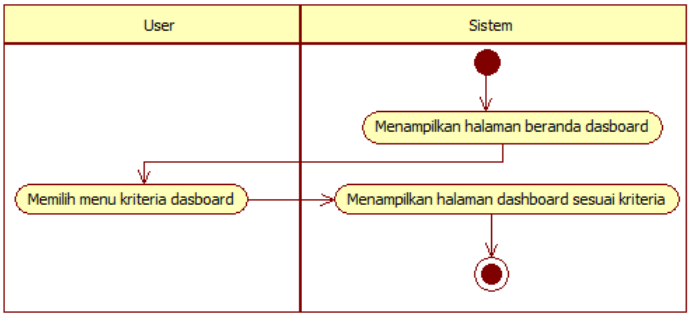
Interaksi antara aktor pengguna dengan *use case* lihat laporan dijelaskan dalam *Activity Diagram* gambar 3.57.



Gambar 3.57 Activity Diagram Lihat Laporan

11. *Activity Diagram* Lihat *Dashboard*

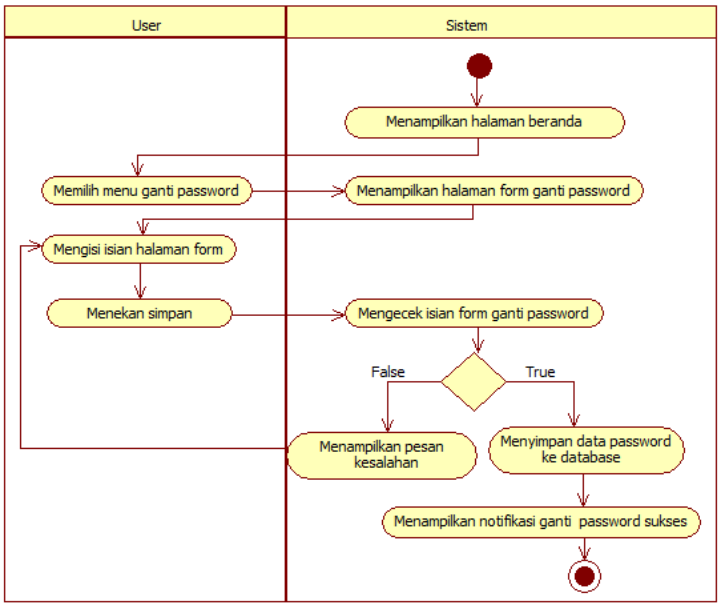
Interaksi antara aktor pengguna dengan *use case* lihat *dashboard*dijelaskan dalam *Activity Diagram* gambar 3.58.



Gambar 3.58*Activity Diagram* Lihat *Dashboard*

12. *Activity Diagram* Mengubah *Password*

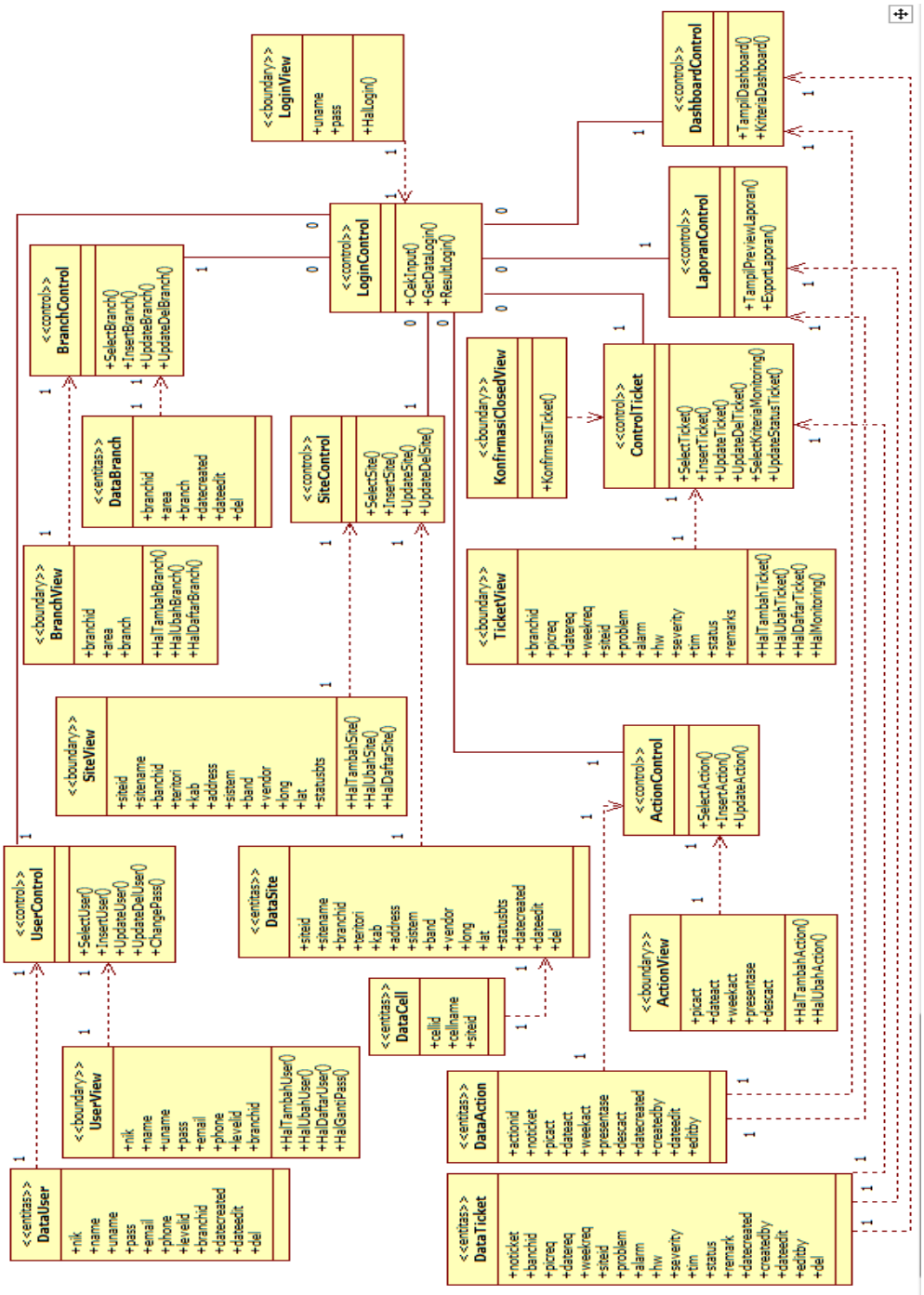
Interaksi antara aktor pengguna dengan *use case* mengubah *password*dijelaskan dalam *Activity Diagram* gambar 3.59.



Gambar 3.59*Activity Diagram* Mengubah *Password*

3.1.4.5 *Class Diagram*

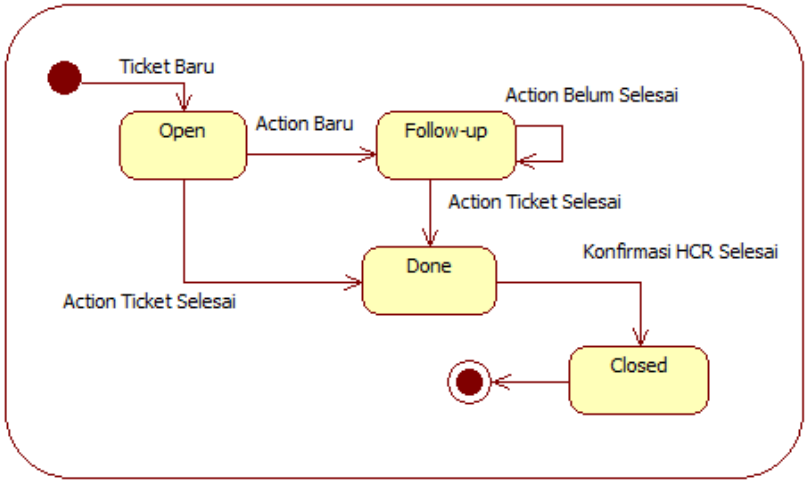
Class Diagram menjelaskan struktur dan hubungan antar objek-objek yang ada pada sistem. Class – class yang dirancang dalam sistem ini dapat dilihat pada gambar 3.60.



Gambar 3.60*Class Diagram*

3.1.4.6 *State Diagram*

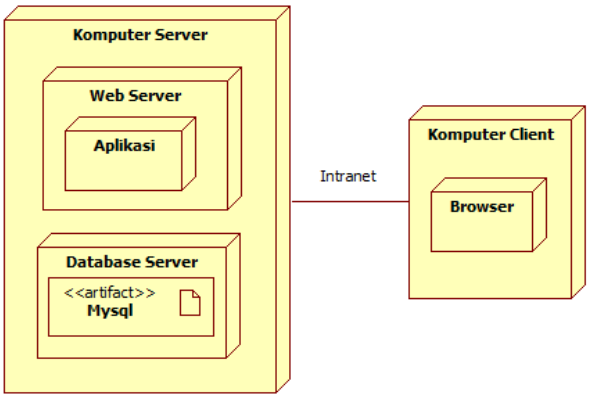
State diagram menggambarkan transisi perubahan keadaan (dari suatu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimulasi yang diterima. Pada gambar 3.61dijelaskan state diagram yang terjadi pada status ticket yang mengalami perubahan state setiap ada stimulasi yang diterima.



Gambar 3.61*State Diagram*

3.1.4.7 *Deployment Diagram*

Deployment Diagram yaitu diagram pendisitribusian,yang menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal. *Deployment Diagram* pada sistem aplikasi manajemen HCR di PT Indosat Tbk dapat dilihat di gambar 3.62.



Gambar 3.62*Deployment Diagram*

3.2 Perancangan Sistem

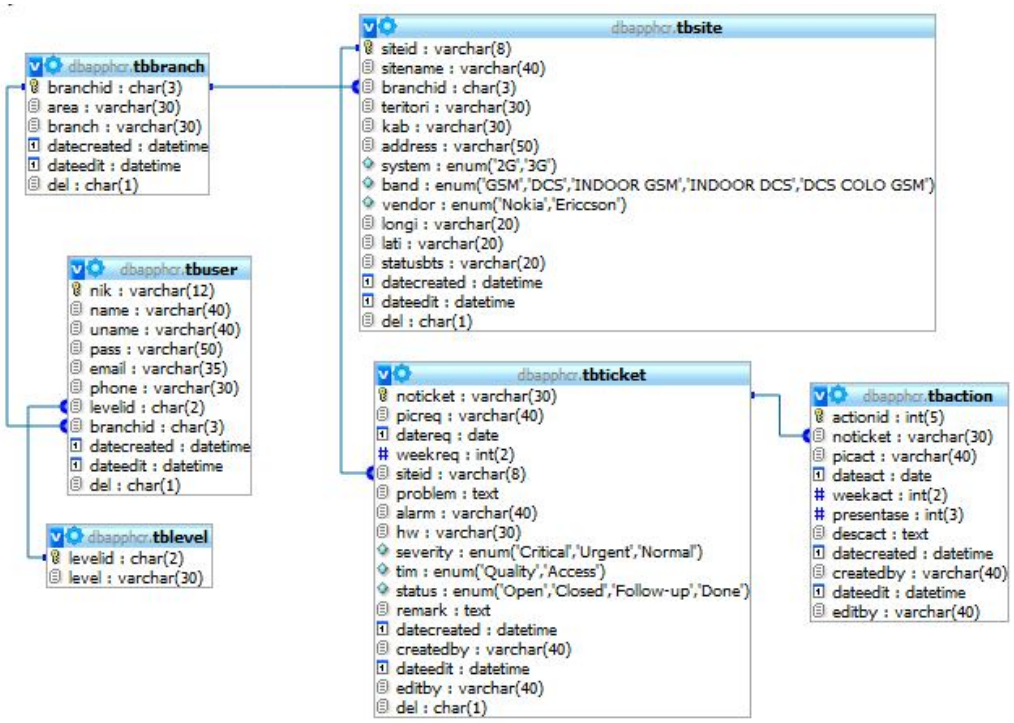
Perancangan sistem merupakan penggambaran, perencanaan, dan pembuatan pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh.Tahapan ini meliputi mengkonfigurasi komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem.Adapun perancangannya sebagai berikut.

3.2.1 Perancangan Basis Data

Perancangan data merupakan tahapan untuk memetakan model konseptual ke model basis data yang akan dipakai. Perancangan data terbagi menjadi skema relasi, diagram skema, dan perancangan struktur tabel.

3.2.1.1 Skema Relasi

Skema relasi merupakan keterkaitan antartabel yang dihubungkan dengan kunci utama (*primary key*) sehingga tabel-tabel tersebut menjadi satu-kesatuan.Untuk lebih jelasnya mengenai skema relasi dapat dilihat pada gambar 3.63.



Gambar 3.63 Skema Relasi

3.2.1.2 Struktur Tabel

Struktur tabel merupakan urutan isi atau data yang berada dalam suatu *record*.Struktur tabel Aplikasi Manajemen HCR (*Hardware Change Request*) adalah sebagai berikut:

1. Tabel Branch

Tabel 3.16 Tabel Branch

Field	Type	Length	Keterangan
branchid	CHAR	3	Primary Key
area	VARCHAR	30	Not Null
branch	VARCHAR	30	Not Null
datecreated	DATETIME		
dateedit	DATETIME		
del	CHAR	1	Not Null

2. Tabel Level

Tabel 3.17 Tabel Level

Field	Type	Length	Keterangan
levelid	CHAR	2	Primary Key
level	VARCHAR	30	Not Null

3. Tabel User

Tabel 3.18 Tabel User

Field	Type	Length	Keterangan
nik	VARCHAR	12	Primary Key
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	FK
branchid	CHAR	3	FK
datecreated	DATETIME		
dateedit	DATETIME		
del	CHAR	1	Not Null

4. Tabel Site

Tabel 3.19 Tabel Site

Field	Type	Length	Keterangan
siteid	VARCHAR	8	Primary Key
sitename	VARCHAR	40	Not Null
branchid	INTEGER	3	FK
teritori	VARCHAR	30	Not Null
kab	VARCHAR	30	Not Null
address	VARCHAR	50	Not Null
sistem	ENUM		(2G,3G)

band	VARCHAR	20	Not Null
vendor	ENUM		(Nokia,Ericcson)
long	VARCHAR	10	Not Null
lat	VARCHAR	10	Not Null
statusbts	VARCHAR	20	Not Null
datecreated	DATETIME		
dateedit	DATETIME		
del	CHAR	1	Not Null

5. Tabel Ticket

Tabel 3.20 Tabel Ticket

Field	Type	Length	Keterangan
noticket	VARCHAR	25	Primary Key
picreq	VARCHAR	40	Not Null
datereq	DATE		
weekreq	INTEGER	2	Not Null
siteid	VARCHAR	8	FK
problem	TEXT		
alarm	VARCHAR	40	Not Null
hw	VARCHAR	40	Not Null
severity	ENUM		(Critical,Urgent,Normal)
tim	ENUM		(Quality,Access)
status	ENUM		(Open,Closed,Follow)
remark	TEXT		
datecreated	DATETIME		
createdby	VARCHAR	40	Not Null
dateedit	DATETIME		
editby	VARCHAR	40	Not Null
del	CHAR	1	Not Null

6. Tabel Action

Tabel 3.21 Tabel Action

Field	Type	Length	Keterangan
actionid	INT	5	Primary Key
noticket	VARCHAR	25	FK
picact	VARCHAR	40	Not Null
dateact	DATE		
weekact	INT	2	
presentase	INTEGER	3	
descact	TEXT		
datecreated	DATETIME		
createdby	VARCHAR	40	Not Null
dateedit	DATETIME		
editby	VARCHAR	40	Not Null

3.2.2 Perancangan Kode

Berikut adalah perancangan kode yang dibuat pada sistem informasiyang akan dibangun :

1. **Branch Id**

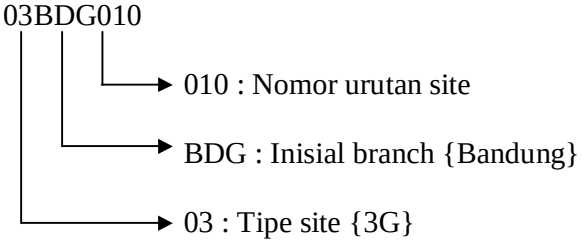
Penulisan branch id adalah sebagai inisial yang diambul dari 3 huruf nama branch yang memperpendek penulisan branch tanpa menghilangkan nama branch. Contoh formatnya sebagai berikut :

Bandung → BDG

Cirebon →CRB

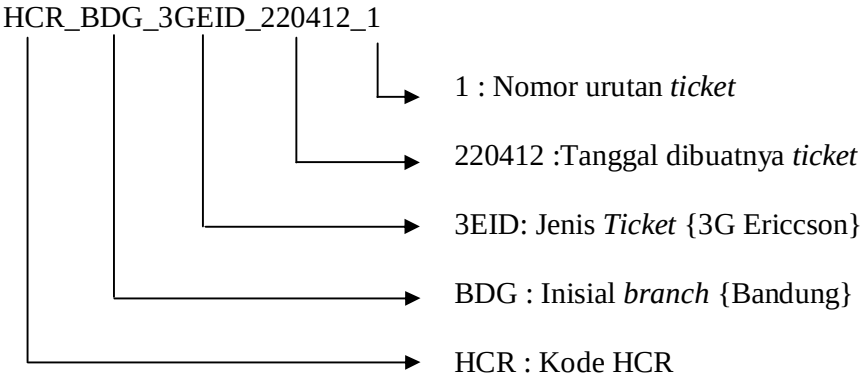
2. Site Id

Penulisan format site id disesuaikan berdasarkan tipe site, inisial branch, dan nomor urutan.Penulisan angka pertama sebagai tipe site, string kedua sebagai inisial branch dan angka ketiga sebagai nomor urutan pembeda atau unik. Contoh formatnya sebagai berikut :



3. Nomor*Ticket*

Format nomor ticket yang akan digunakan pada sistem disesuaikan berdasarkanidentitas ticket HCR, branch id, jenis *ticket*, tanggal dibuat dan nomor urutan *ticket*. Dimana string pertamasebagai inisial HCR, string kedua menunjukan inisial branch, string ketiga menunjukan jenis *ticket*, angka menunjukan tanggal dibuat HCR, angkaterakhir menunjukan nomor urutan dan semuanya dipisahkan dengan karakter _. Format nomor tersebutdapat digambarkan seperti di bawah ini :

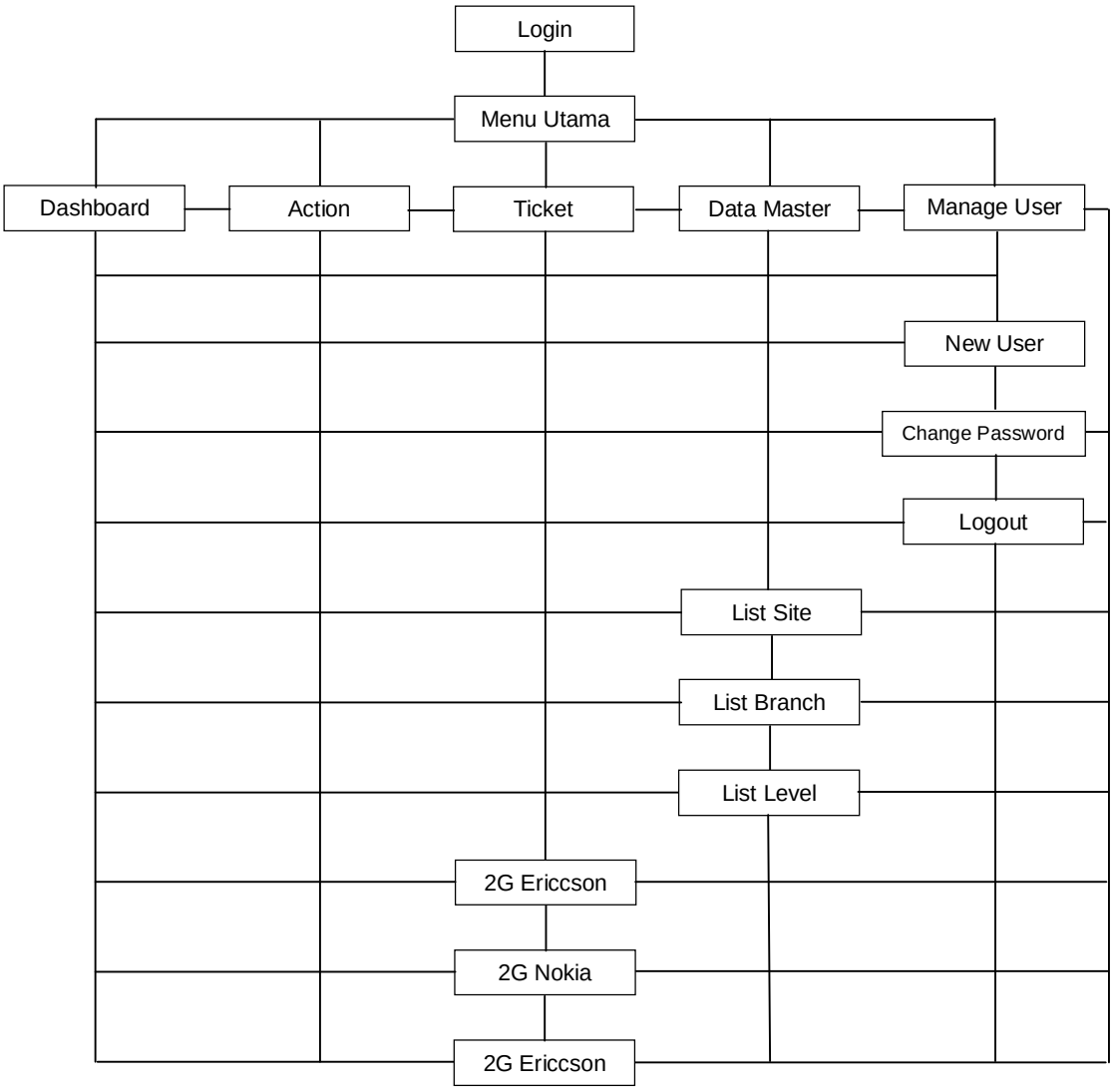


3.2.3 Perancangan Stuktur Menu

Struktur menu dibangun untuk menggambarkan perancangan menu proses yang dapat digunakan oleh pengguna. Struktur menu secara umum yang dibuat dalam sistem ini terdiri dari menu-menu dan sub menunya. Penggambaran stuktur menunya menggunakan diagram *network*.

3.2.3.1 Stuktur Menu *Administrator*

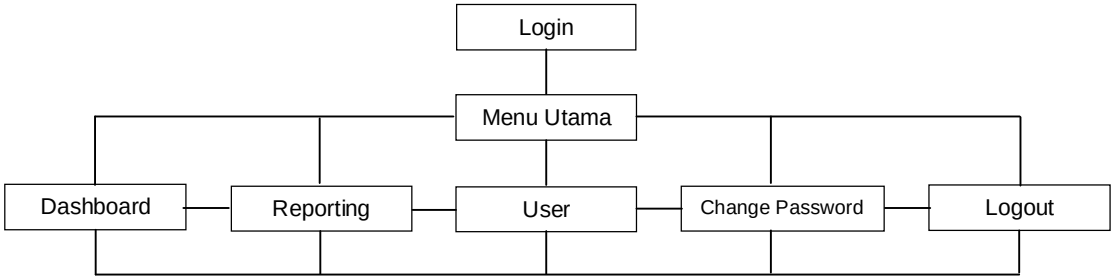
Pada gambar 3.64 digambarkan mengenai struktur menu pada sisi admin:



Gambar 3.64 Stuktur Menu *Administrator*

3.2.3.2 Stuktur Menu *Manager*

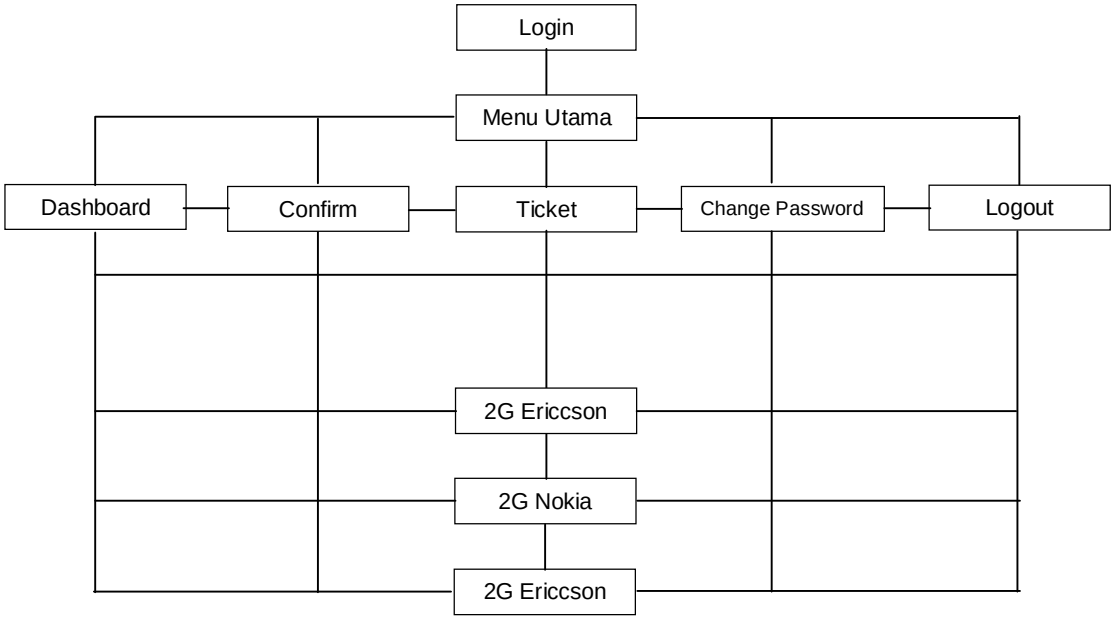
Pada gambar 3.65 digambarkan mengenai struktur menu pada sisi manager:



Gambar 3.65 Stuktur Menu *Manager*

3.2.3.3 Stuktur Menu Tim*Project*

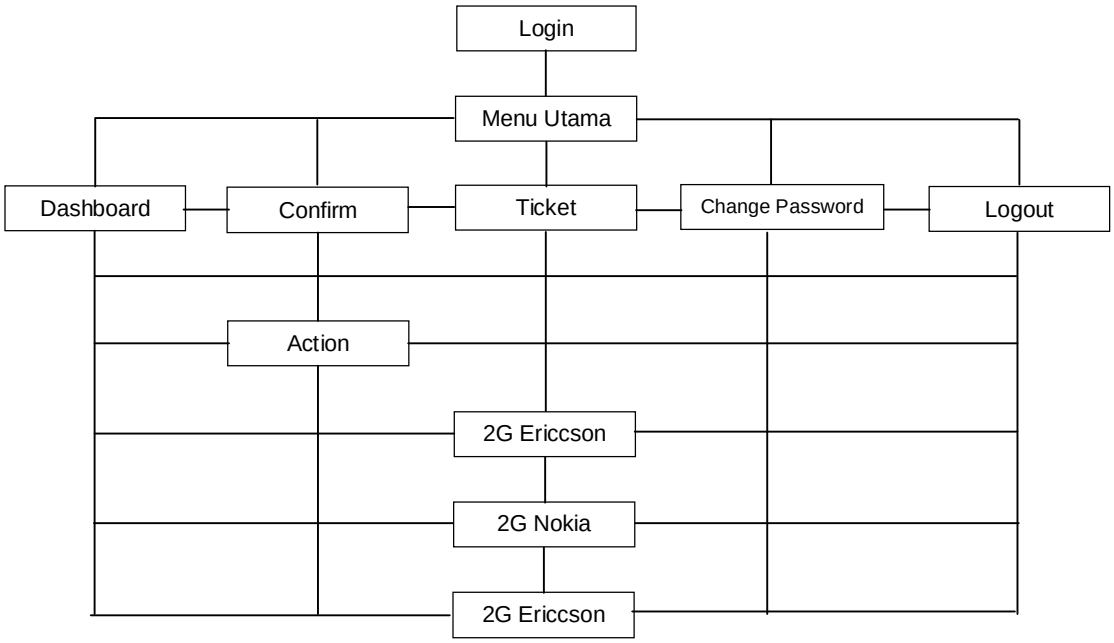
Pada gambar 3.66 digambarkan mengenai struktur menu pada sisi tim project:



Gambar 3.66 Stuktur Menu Tim *Project*

3.2.3.4 Stuktur Menu Tim*Quality*

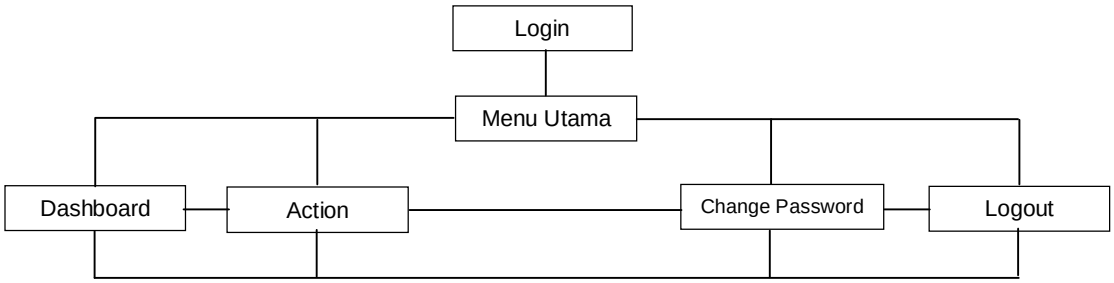
Pada gambar 3.67digambarkan mengenai struktur menu pada sisi tim*quality*:



Gambar 3.67 Stuktur Menu Tim *Quality*

3.2.3.5 Stuktur Menu Tim*Access*

Pada gambar 3.68 digambarkan mengenai struktur menu pada sisi tim*access*:



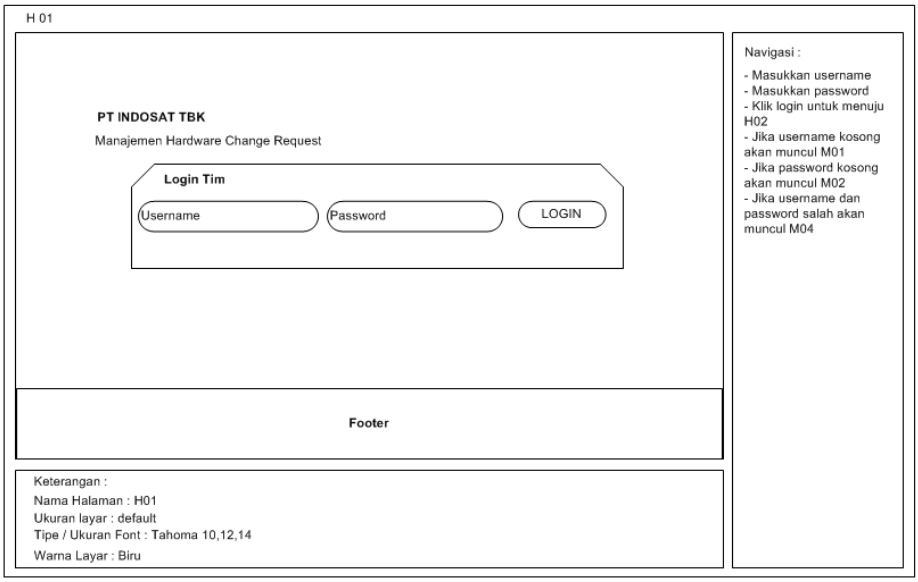
Gambar 3.68 Stuktur Menu Tim *Access*

3.2.4 Perancangan Antarmuka

Perancangan antarmuka merupakan tahap yang harus dikerjakan pada proses pembuatan sebuah program atau aplikasi. Antarmuka atau interface adalah suatu bagian yang berhubungan langsung dengan pengguna aplikasi. Perancangan antarmuka bertujuan agar program atau aplikasi yang dihasilkan terlihat lebih menarik dan mudah

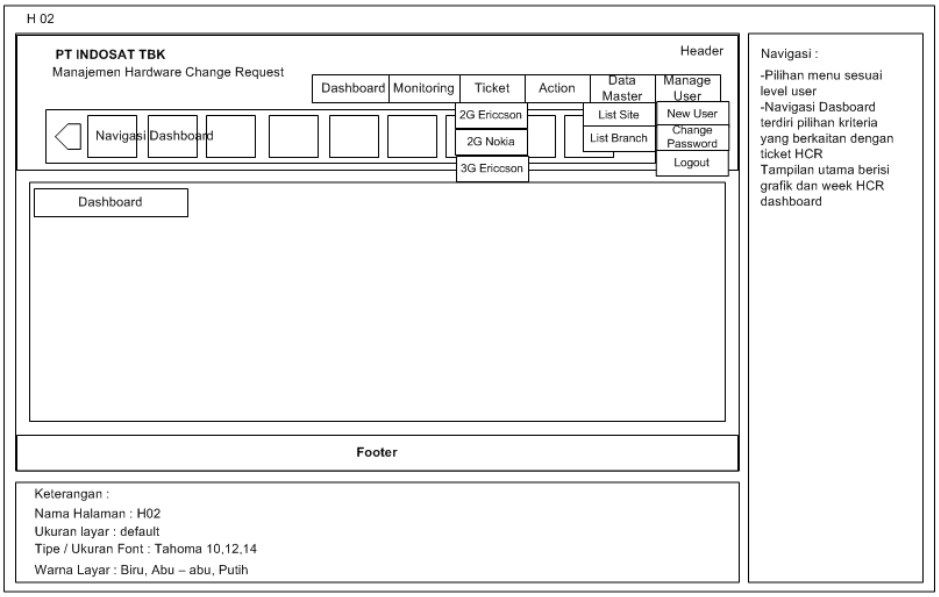
dimengerti pada saat dioperasikan.Perancangan tampilan antarmuka Aplikasi Manajemen HCR di PT Indosat Tbk dapat dilihat pada gambar 3.69 – 3.88.

1. Tampilan Halaman Login



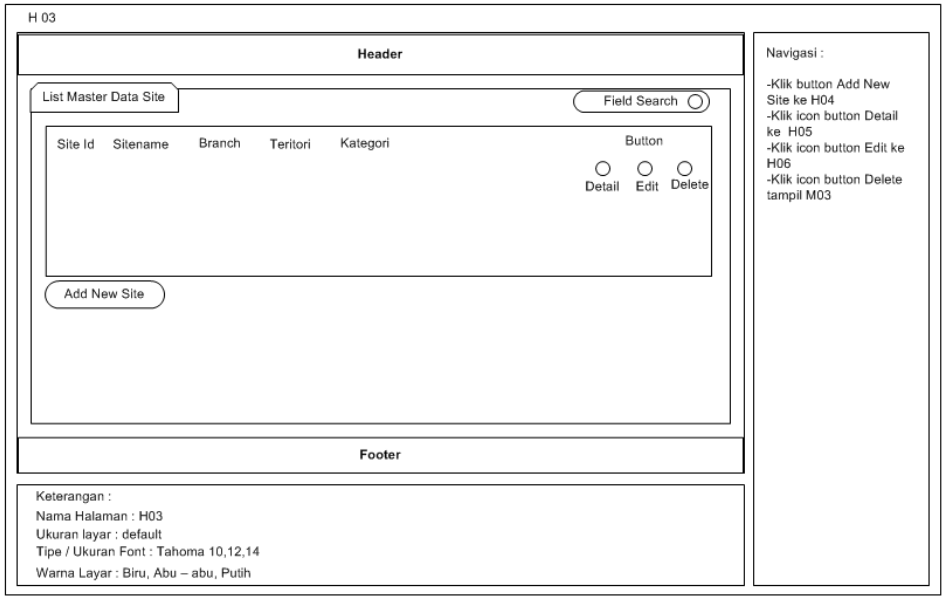
Gambar 3.69 Perancangan Halaman Login

2. Tampilan Halaman Utama / Beranda *Dashboard*



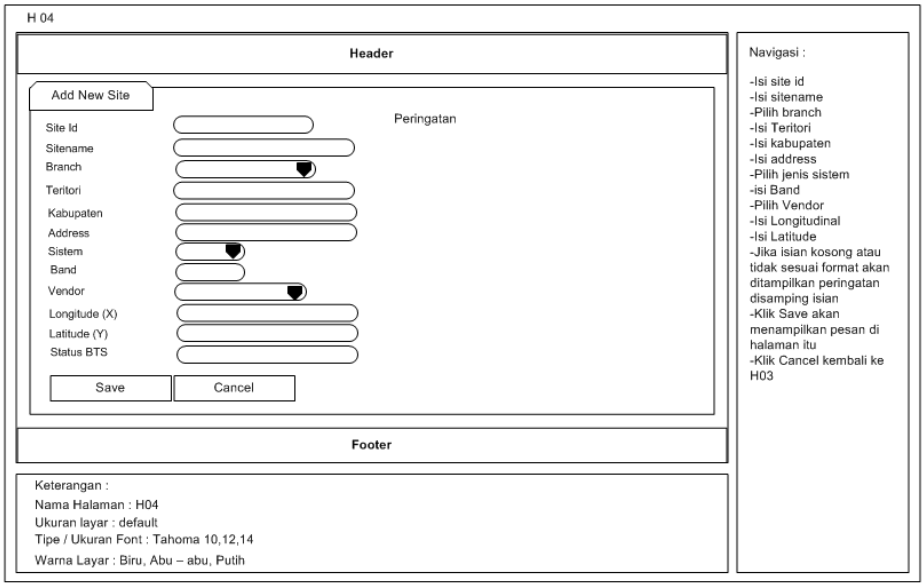
Gambar 3.70 Perancangan Halaman Utama

3. Tampilan HalamanList Data Site



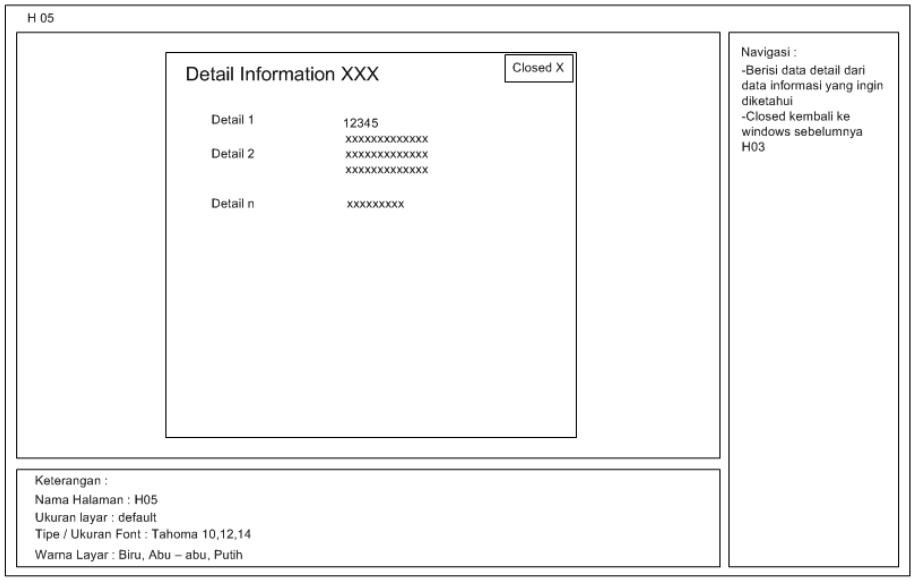
Gambar 3.71 Perancangan Halaman List Data Site

4. Tampilan Halaman Tambah Data Site



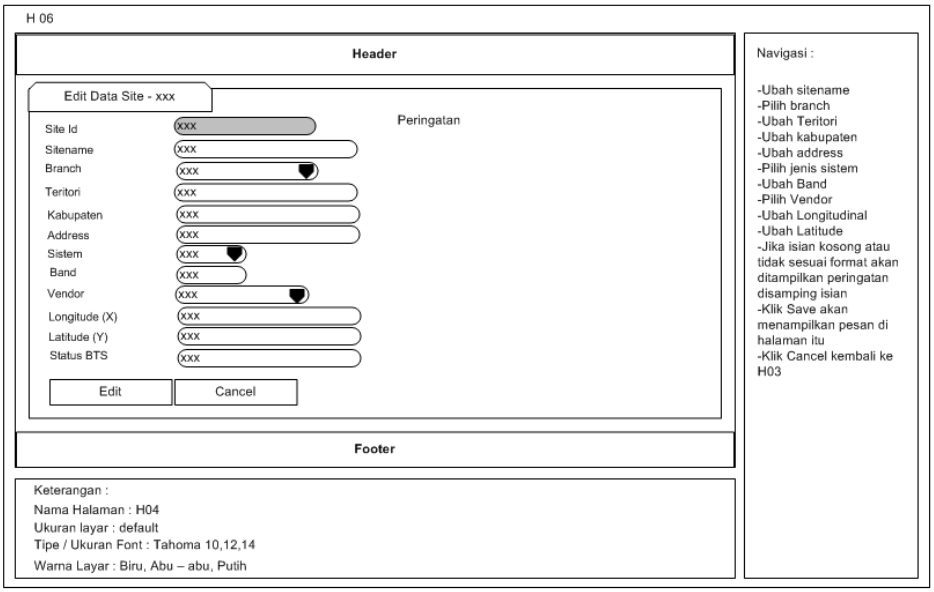
Gambar 3.72 Perancangan Halaman Tambah Data Site

5. Tampilan Halaman Detail Information Data



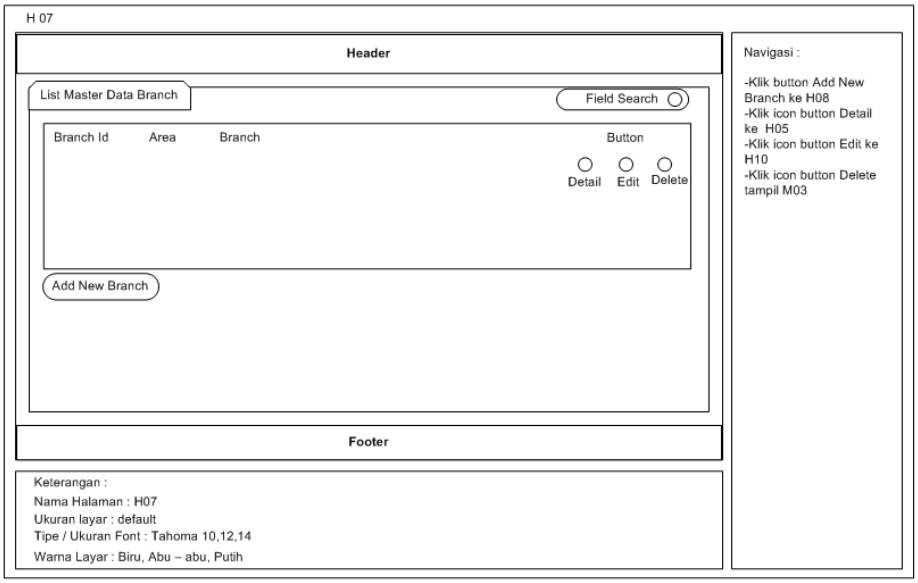
Gambar 3.73 Perancangan Halaman Detail Information Data

6. Tampilan Halaman Ubah Data Site



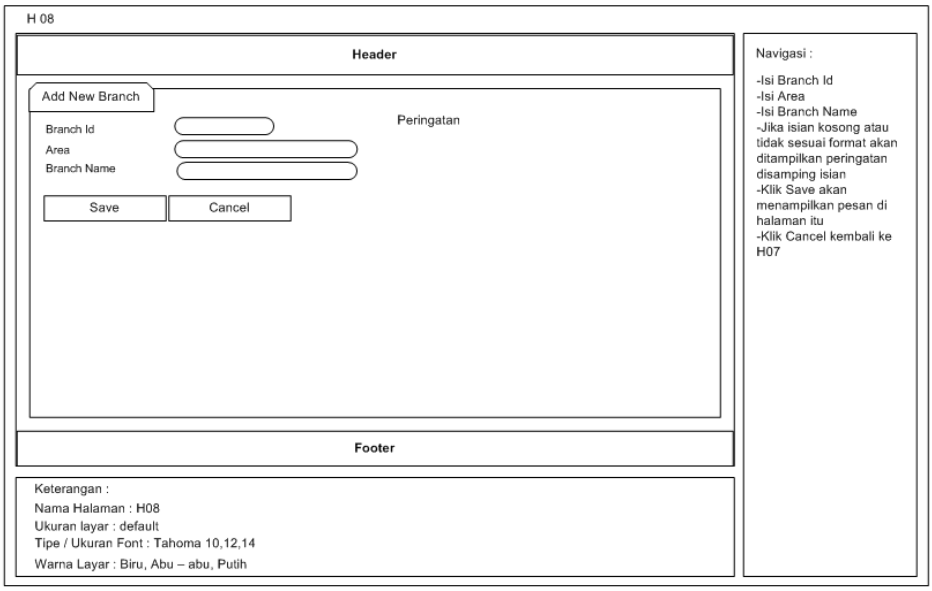
Gambar 3.74 Perancangan Halaman Ubah Data Site

7. Tampilan Halaman List Data Branch



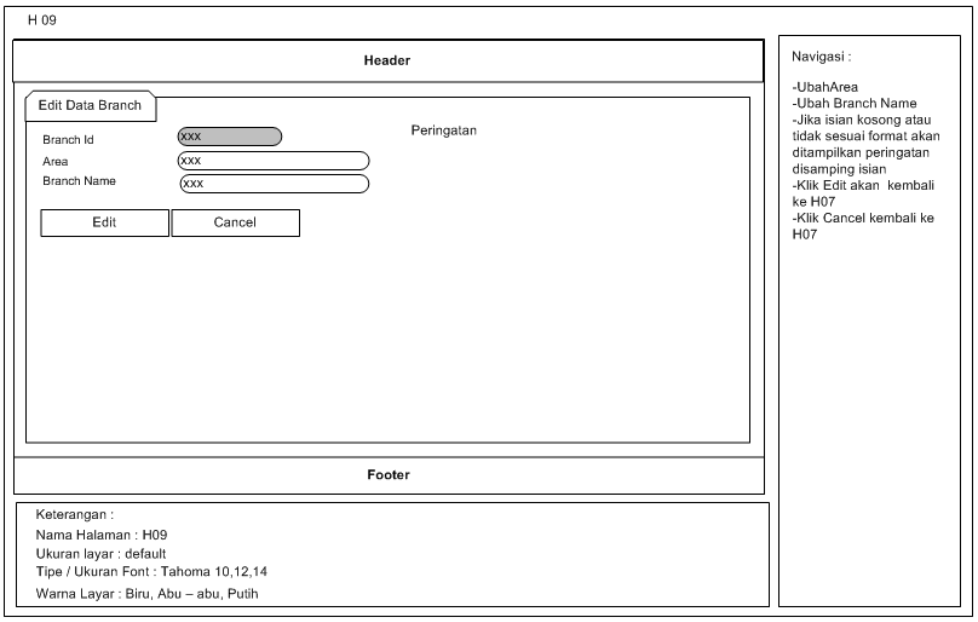
Gambar 3.75 Perancangan Halaman List Data Branch

8. Tampilan Halaman Tambah Data Branch



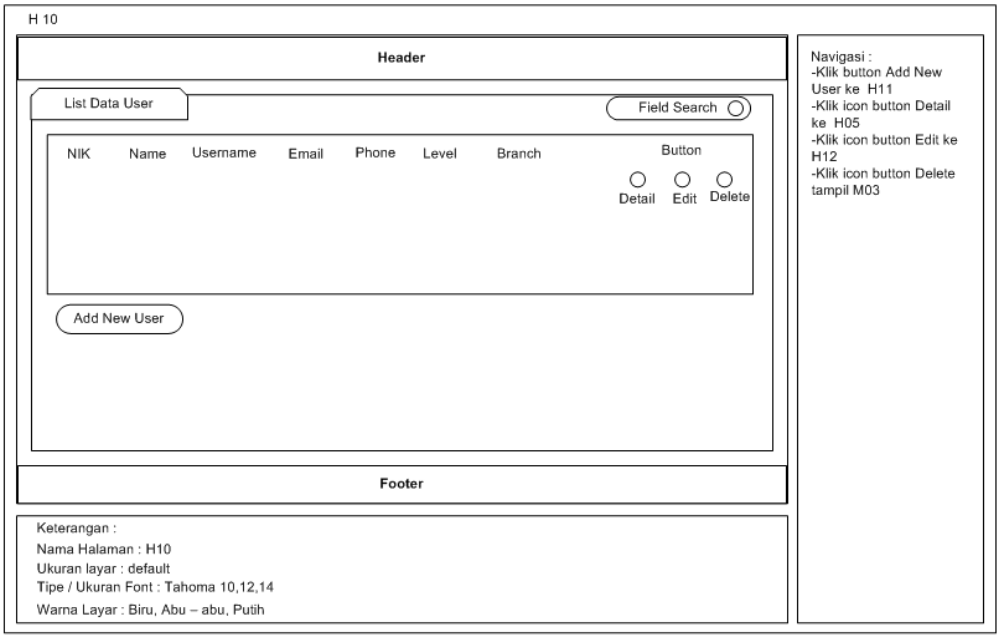
Gambar 3.76 Perancangan Halaman Tambah Data Branch

9. Tampilan Halaman Ubah Data Branch



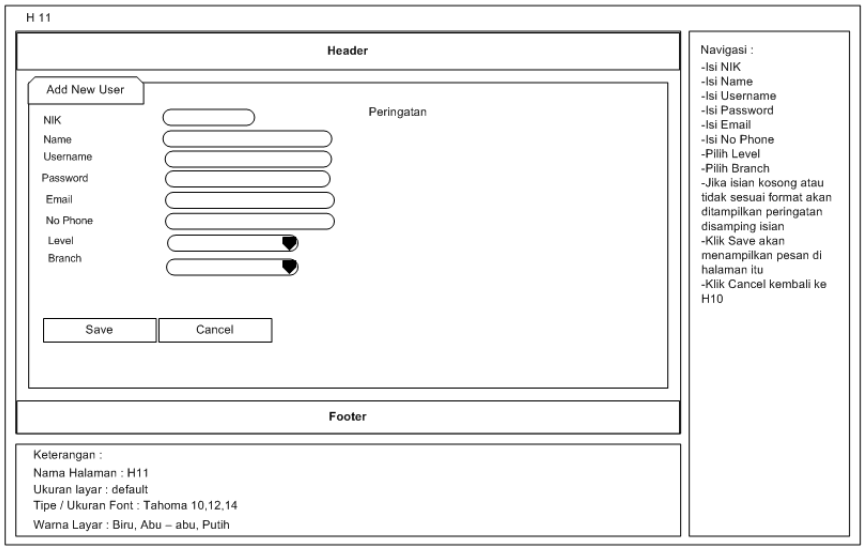
Gambar 3.77 Perancangan Halaman Ubah Data Branch

10. Tampilan Halaman List Data User



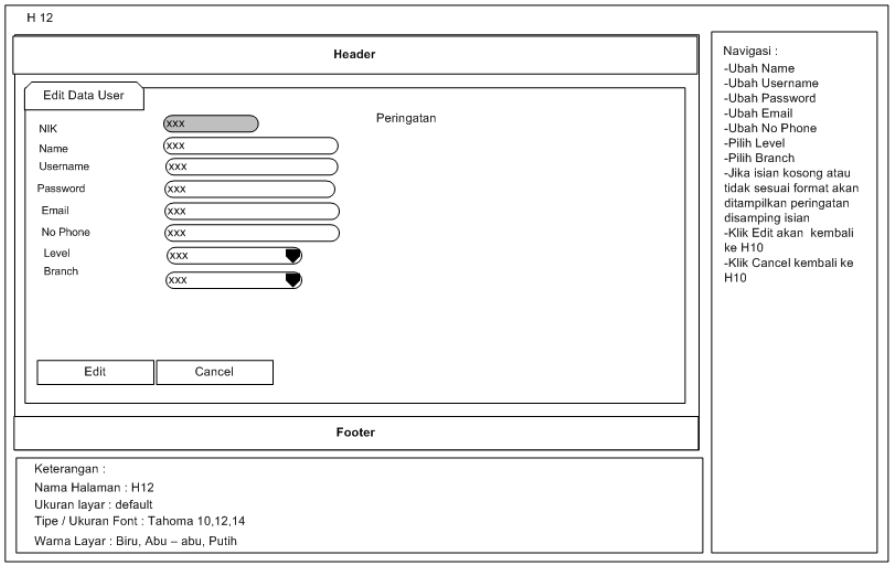
Gambar 3.78 Perancangan Halaman List Data User

11. Tampilan Halaman Tambah Data User



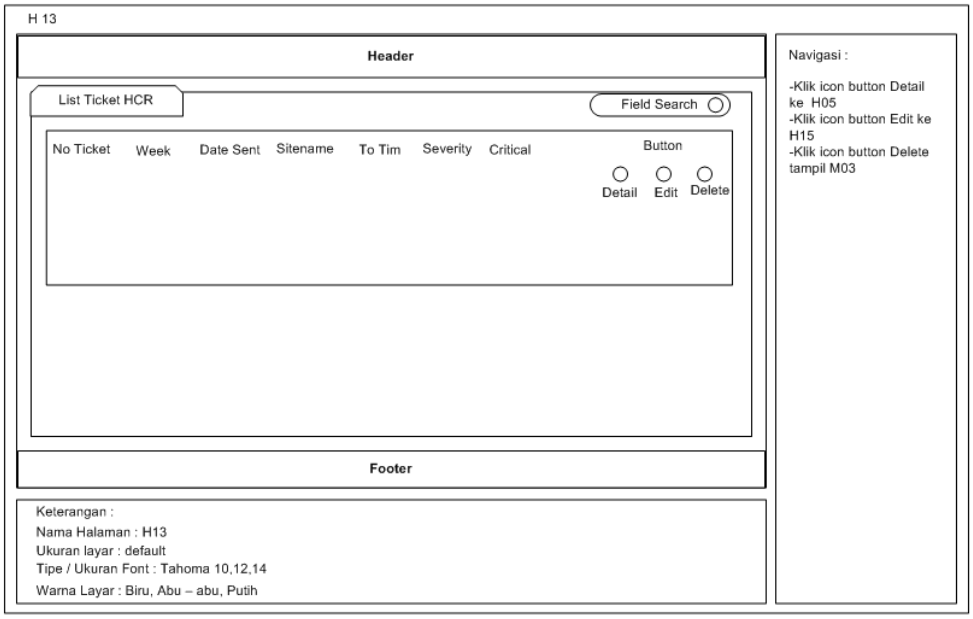
Gambar 3.79 Perancangan Halaman Tambah Data User

12. Tampilan Halaman Ubah Data User



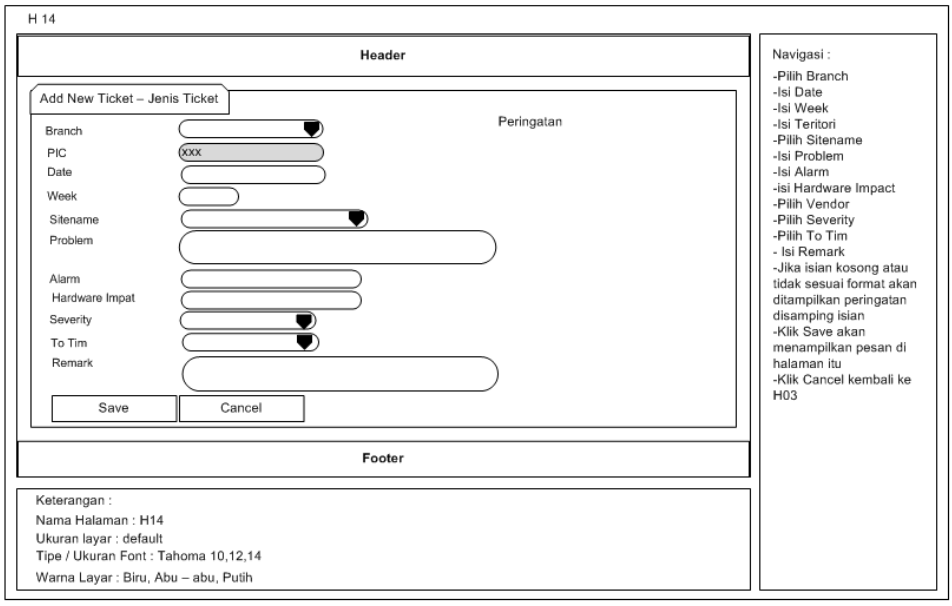
Gambar 3.80 Perancangan Halaman Ubah Data User

13. Tampilan Halaman List Ticket



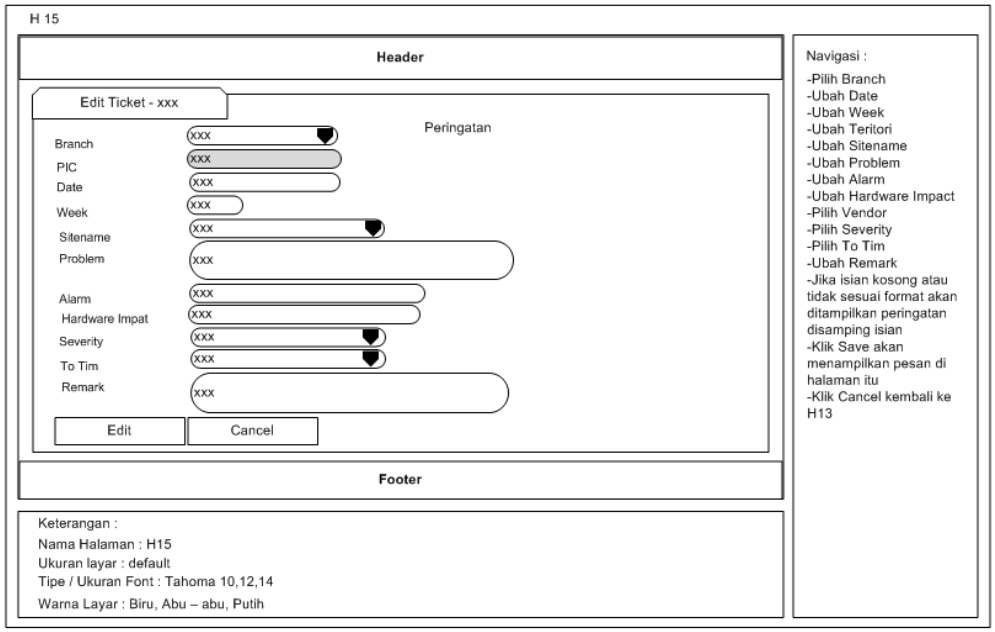
Gambar 3.81 Perancangan Halaman List Ticket

14. Tampilan Halaman Tambah Ticket



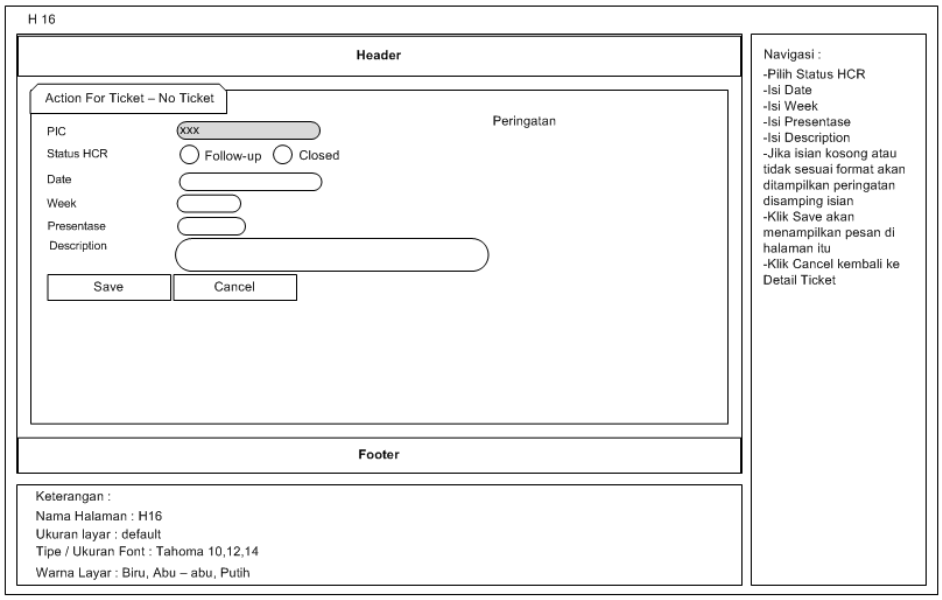
Gambar 3.82 Perancangan Halaman Tambah Ticket

15. Tampilan Halaman Ubah Ticket



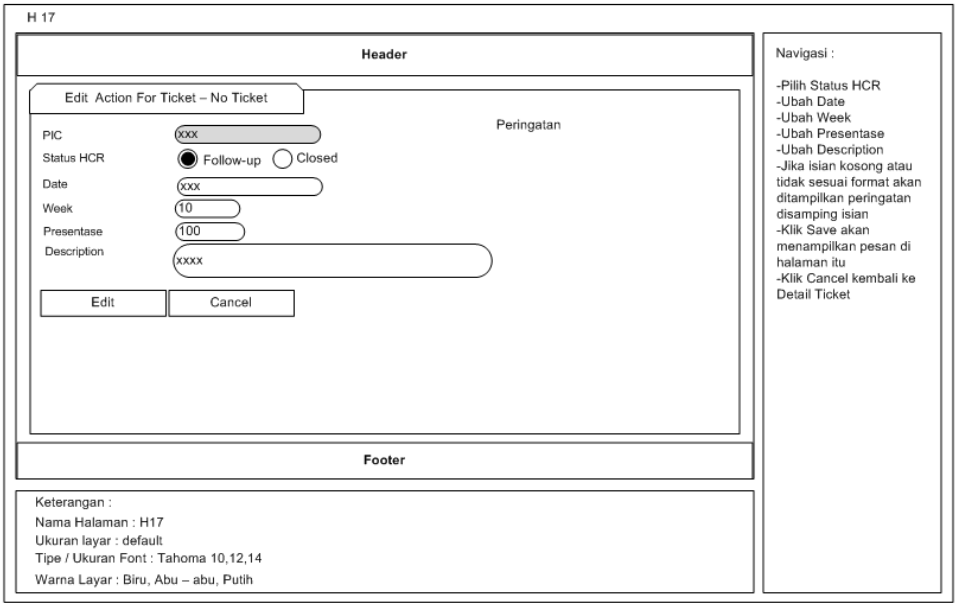
Gambar 3.83 Perancangan Halaman Ubah Ticket

16. Tampilan Halaman Tambah Action



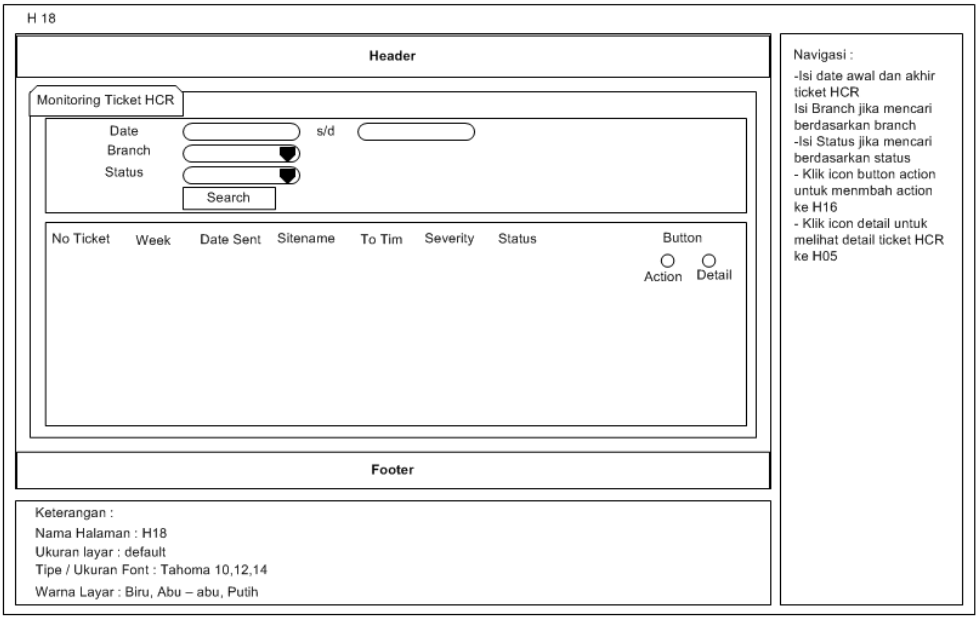
Gambar 3.84 Perancangan Halaman Tambah Action

17. Tampilan Halaman Ubah Action



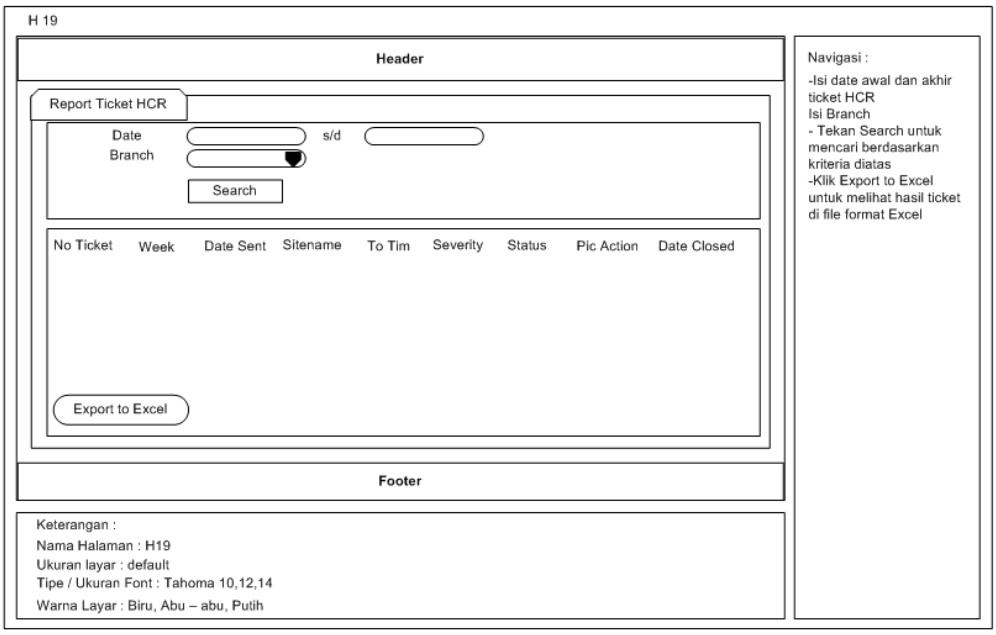
Gambar 3.85 Perancangan Halaman Ubah Action

18. Tampilan Halaman Monitoring Ticket



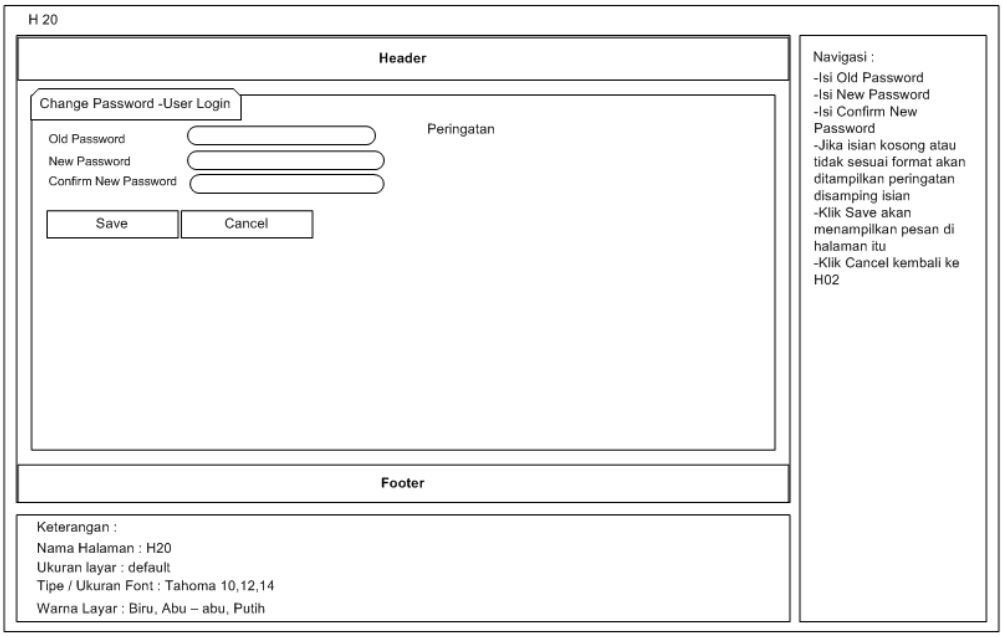
Gambar 3.86 Perancangan Halaman Monitoring Ticket

19. Tampilan Halaman Laporan



Gambar 3.87 Perancangan Halaman Laporan

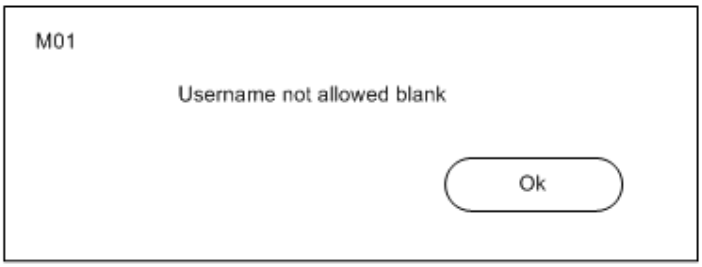
20. Tampilan Halaman Ganti *Password*



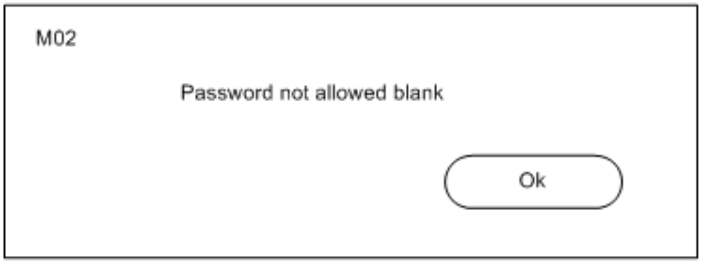
Gambar 3.88 Perancangan Halaman Ganti *Password*

3.2.5 Perancangan Pesan

Perancangan pesan Aplikasi Manajemen HCR di PT Indosat Tbk dapat dilihat pada gambar 3.89 – 3.116.



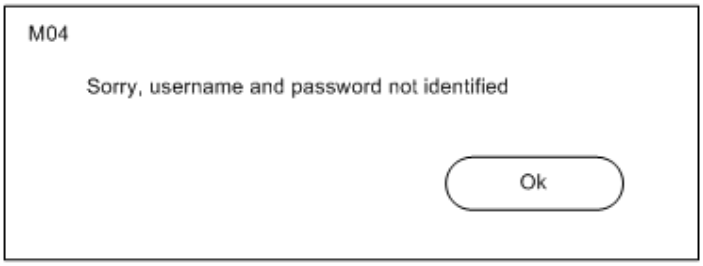
Gambar 3.89 Perancangan Pesan M01



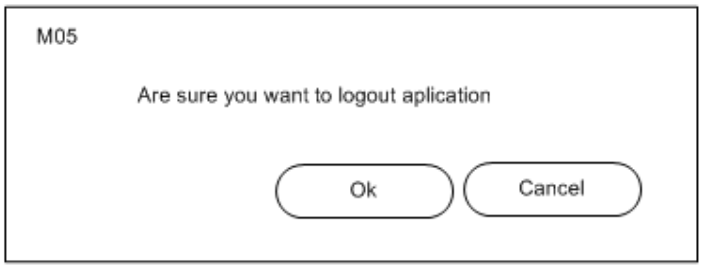
Gambar 3.90 Perancangan Pesan M02



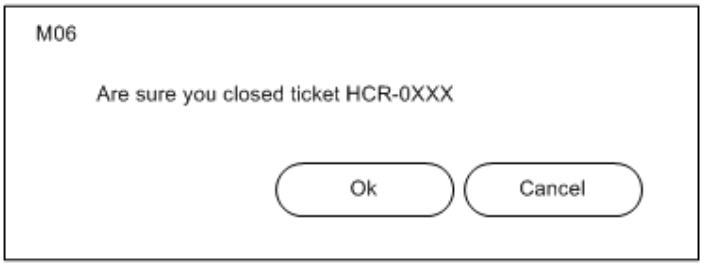
Gambar 3.91 Perancangan Pesan M03



Gambar 3.92 Perancangan Pesan M04



Gambar 3.93 Perancangan Pesan M05



Gambar 3.94 Perancangan Pesan M06



Gambar 3.95 Perancangan Pesan M07



Gambar 3.96 Perancangan Pesan M08



Gambar 3.97 Perancangan Pesan M09



Gambar 3.98 Perancangan Pesan M10



Gambar 3.99 Perancangan Pesan M11



Gambar 3.100 Perancangan Pesan M12



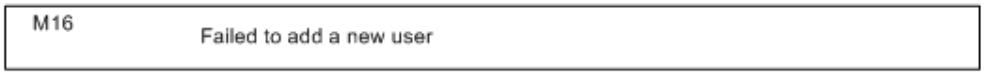
Gambar 3.101 Perancangan Pesan M13



Gambar 3.102 Perancangan Pesan M14



Gambar 3.103 PerancanganPesan M15



Gambar 3.104 Perancangan Pesan M16



Gambar 3.105 Perancangan Pesan M17



Gambar 3.106 Perancangan Pesan M18



Gambar 3.107 Perancangan Pesan M19



Gambar 3.108 Perancangan Pesan M20

M21	Successfully to add a new ticket XXX, with no XXX

Gambar 3.109 Perancangan Pesan M21

M22	Failed to add a new ticket

Gambar 3.110 Perancangan Pesan M22

M23	Successfully edit ticket

Gambar 3.111 Perancangan Pesan M23

M24	Failed to edit ticket

Gambar 3.112 Perancangan Pesan M24

M25	The new action to ticket number: XX successfully added

Gambar 3.113 Perancangan Pesan M25

M26	Failed to add a new action

Gambar 3.114 Perancangan Pesan M26

M27	Failed to add a new action, check the value presentase

Gambar 3.115 Perancangan Pesan M27

M28	Failed to add a new action, Date action >= date request ticket

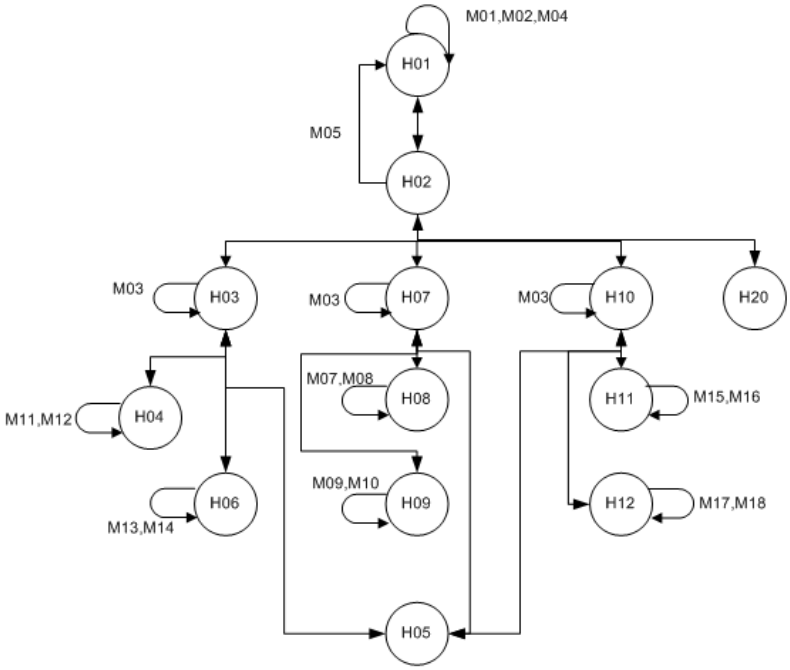
Gambar 3.116 Perancangan Pesan M28

3.2.6 Jaringan Sematik

Jaringan semantik pada aplikasi yang dibangun terdiri dari jaringan semantik pada login user *administrator*, *manajer*, tim*project*, tim *quality* dan tim *access* adalah sebagai berikut:

1. Jaringan Semantik *Administrator*

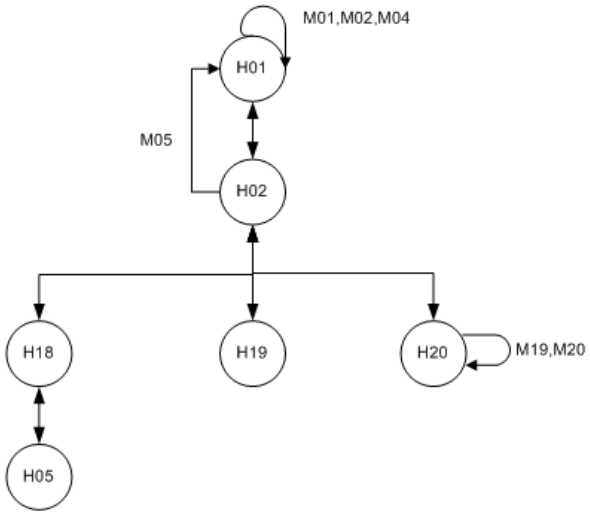
   Pada gambar 3.117 digambarkan mengenai jaringan semantik *administrator*:



Gambar 3.117Jaringan Semantik *Administrator*

2. Jaringan Semantik *Manajer*

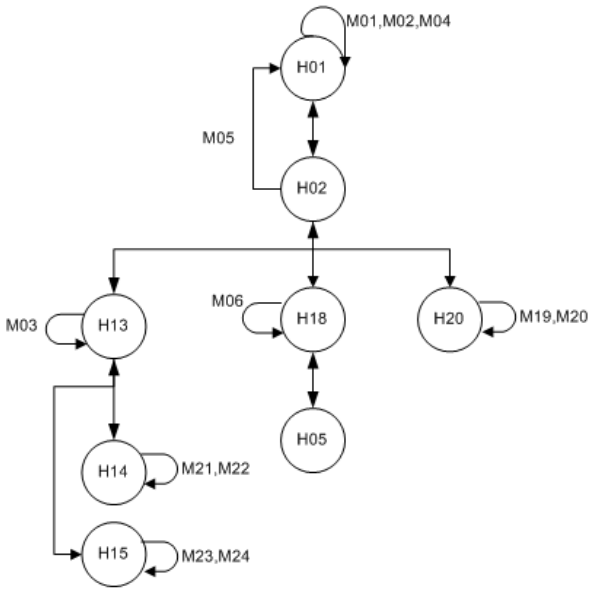
   Pada gambar 3.118 digambarkan mengenai jaringan semantik *manager*:



Gambar 3.118 Jaringan Semantik *Manager*

3. Jaringan Semantik Tim *Project*

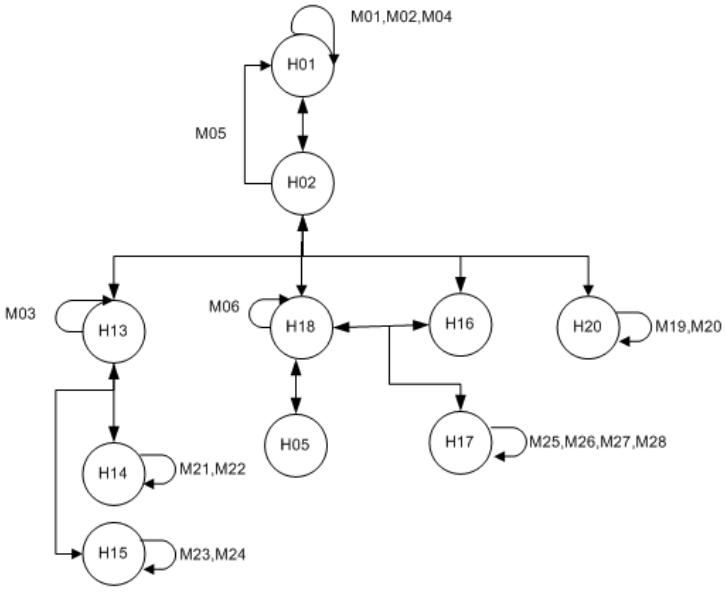
Pada gambar 3.119 digambarkan mengenai jaringan semantik tim*project*:



Gambar 3.119 Jaringan Semantik Tim *Project*

4. Jaringan Semantik Tim *Quality*

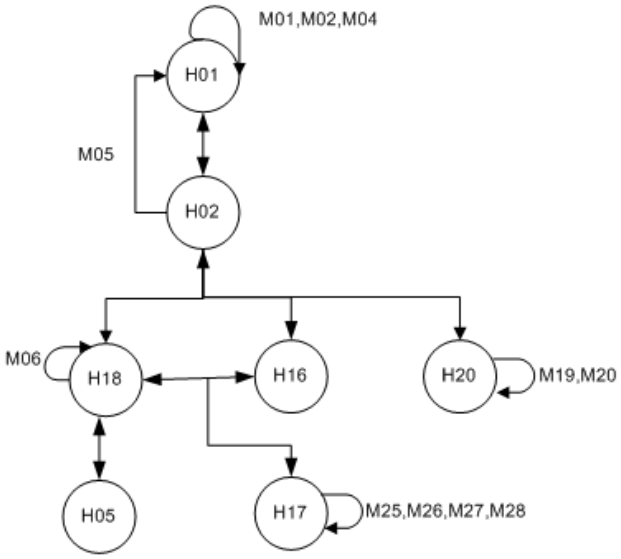
Pada gambar 3.120 digambarkan mengenai jaringan semantik tim*quality*:



Gambar 3.120 Jaringan Semantik Tim *Quality*

5. Jaringan Semantik Tim *Access*

Pada gambar 3.121 digambarkan mengenai jaringan semantik tim*access*:



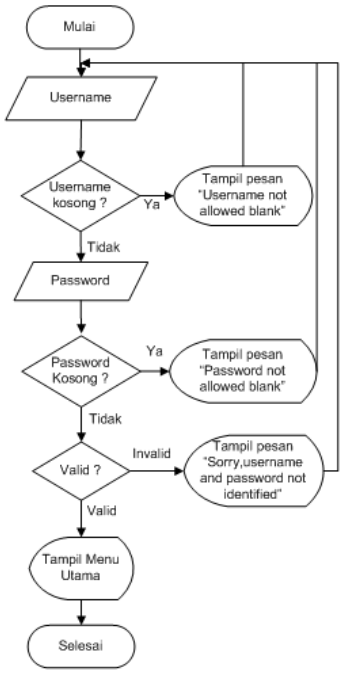
Gambar 3.121 Jaringan Semantik Tim Access

3.2.7 Perancangan Method

Method merupakan suatu operasi berupa fungsi-fungsi yang dapat di kerjakan oleh suatu objek. Method di definisikan pada class akantetapi dipanggil melalui object.*Method* yang digunakan pada class diagram.

1. *Method* Login

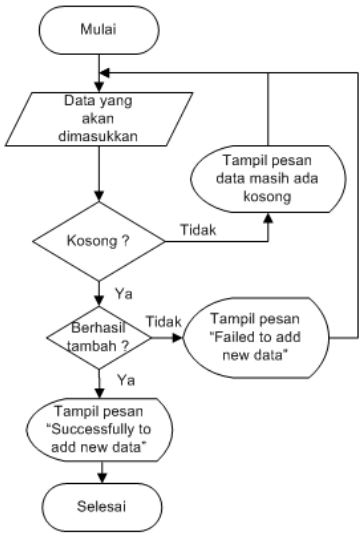
*Method*yang dipanggil ketika penggunaakan mengakases aplikasi.Untuk lebih jelasnya dapat dilihat pada*flowchart* gambar3.122.



Gambar 3.122*Flowchart Method* Login

2. *Method*Tambah Data

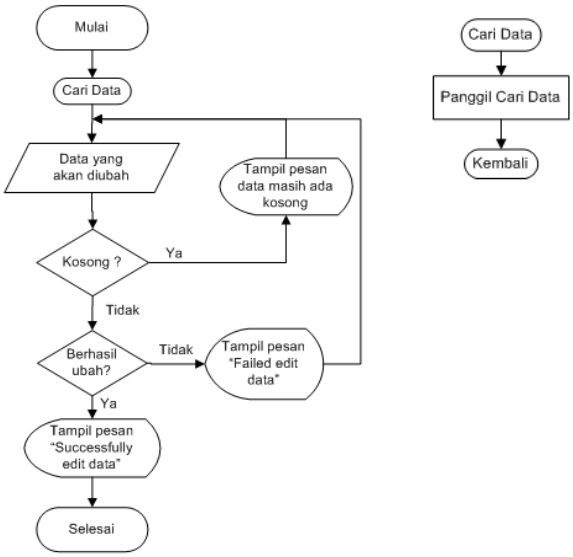
Method yang dipanggil ketika pengguna melakukan penambahan data.Untuk lebih jelasnya dapat dilihat pada *flowchart* gambar 3.123.



Gambar 3.123*Flowchart Method* Tambah Data

3. *Method*Ubah Data

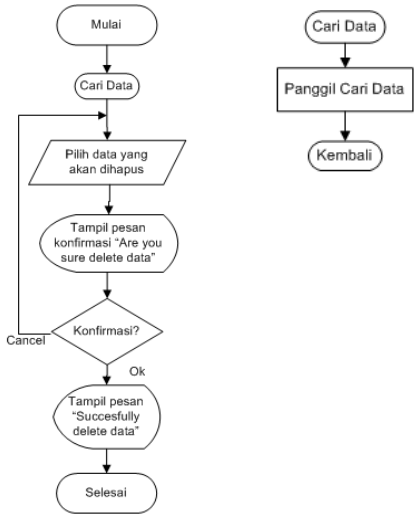
Method yang dipanggil ketika pengguna melakukan perubahan data.Untuk lebih jelasnya dapat dilihat pada *flowchart* gambar 3.124.



Gambar 3.124*Flowchart Method* Ubah Data

4. *Method*Hapus Data

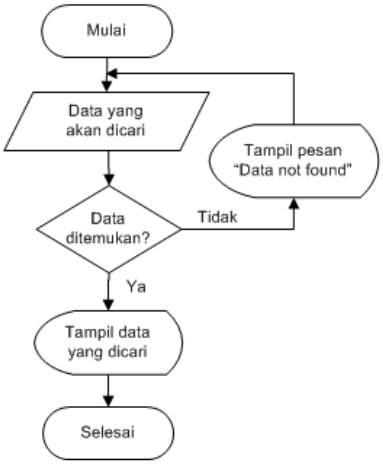
Method yang dipanggil ketika pengguna melakukan penghapus data.Untuk lebih jelasnya dapat dilihat pada *flowchart* gambar 3.125.



Gambar 3.125*Flowchart Method* Hapus Data

5. *Method*Cari Data

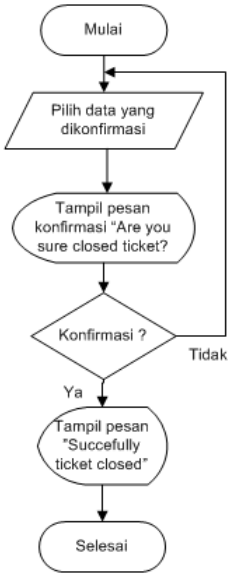
Method yang dipanggil ketika pengguna melakukan pencarian data.Untuk lebih jelasnya dapat dilihat pada *flowchart* gambar 3.126.



Gambar 3.126*Flowchart Method* Cari Data

6. *Method* Konfirmasi

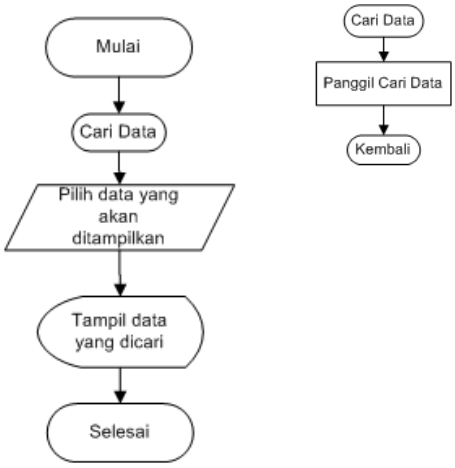
Method yang dipanggil ketika pengguna melakukan konfirmasi ticket.Untuk lebih jelasnya dapat dilihat pada *flowchart* gambar 3.127.



Gambar 3.127*Flowchart Method* Konfirmasi

7. *Method*Tampil Data

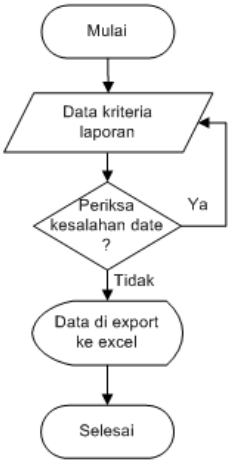
Method yang dipanggil ketika pengguna menampilkan rincian data.Untuk lebih jelasnya dapat dilihat pada *flowchart* gambar 3.128.



Gambar 3.128*Flowchart Method* Tampil Data

8. *Method*Export

Method yang dipanggil ketika pengguna meminta print laporan ke *excel*. Untuk lebih jelasnya dapat dilihat pada *flowchart* gambar 3.129.



Gambar 3.129*Flowchart Method* Export