

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

Pada analisis sistem akan dijelaskan secara keseluruhan tentang tahap analisis permasalahan yang ada pada proses penjadwalan PT. NCI cabang Surabaya. Kemudian setelah melakukan analisis sistem dilanjutkan dengan analisis permasalahan yang menerangkan tentang masalah yang ada pada proses penjadwalan penanganan *complain* pada PT. NCI khususnya cabang Surabaya.

3.1.1 Analisis Permasalahan

Pada tahap analisis permasalahan ini langkah-langkah yang dikerjakan seperti melakukan survei pada PT. NCI cabang Surabaya, melakukan wawancara pada bagian terkait penjadwalan penanganan *complain*. Dari beberapa tahap yang dikerjakan tersebut akan diketahui permasalahan yang ada terdapat penjadwalan *complain* PT. NCI cabang Surabaya saat ini.

Dari permasalahan tersebut diketahui sering terlambatnya penanganan suatu *complain* karena tidak memperhatikan kriteria *complain* atau masalah yang ada dan berdampak menumpuknya list *complain* yang di tangani oleh pihak *sofdev*. Dan jika masalah tersebut tidak segera di atasi maka akan mengakibatkan hilangnya kepercayaan pihak rumah sakit pada PT. NCI dikarenakan lambatnya proses pengerjaan *complain*.

3.1.2 Identifikasi Permasalahan

Dari analisis permasalahan yang telah dilakukan maka ditemukan identifikasi dari permasalahan yang ada pada proses penjadwalan penanganan *complain* yang ada pada PT. NCI khususnya pada cabang Surabaya. Proses penjadwalan masih menggunakan cara manual, koordinator *sofdev* langsung memberikan *complain* yang datang kepada *sofdev* dengan tidak melihat kriteria

complain mulai dari jenis bug modul, status rumah sakit, nilai kontrak rumah sakit, tanggal *complain* masuk dan batas waktu penyelesaian *complain*.

Dari kondisi seperti inilah akan terjadi keterlambatan karena tidak adanya prioritas kerja ada penjadwalan yang sekarang diterapkan pada PT. NCI khususnya pada cabang Surabaya, dan jika kondisi ini terjadi secara berkelanjutan maka akan berdampak pada hilangnya kepercayaan *customer* kepada PT. NCI.

Pada tahap identifikasi masalah ini terdapat langkah-langkah yang dikerjakan seperti melakukan survei pada PT. NCI cabang Surabaya, melakukan wawancara pada bagian terkait penjadwalan penanganan *complain*, kemudian mendokumentasikan data-data yang diperoleh dari hasil survei dan wawancara. Dari beberapa tahap yang dikerjakan tersebut akan diketahui permasalahan yang ada terdapat penjadwalan *complain* PT. NCI cabang Surabaya saat ini. Proses-proses yang di dapat dari identifikasi masalah ini adalah mengetahui secara keseluruhan proses penjadwalan penanganan *complain* yang ada di PT. NCI cabang Surabaya, mulai dari proses pengiriman *complain*, proses penerimaan *complain*, hingga penjadwalan penanganan *complain*.

3.1.3 Analisis Kebutuhan Sistem

Berdasarkan hasil survei dan wawancara diperoleh hasil dari kebutuhan sistem penjadwalan penanganan *complain* sebagai berikut :

- a. Sistem yang di rancang harus dapat menghubungkan proses yang ada pada pihak *onsite* dengan pihak *sofdev*.
- b. Pihak *onsite* terdapat pada setiap rumah sakit yang bekerja sama dengan PT. NCI, karena itu sistem yang di buat berbasis *website*.
- c. Output yang dihasilkan oleh sistem adalah jadwal dari penanganan *complain* oleh pihak *sofdev*.
- d. Sistem Operasi yang disarankan adalah Windows XP, Vista, Seven, 8, 10.
- e. Database yang digunakan ada Mysql.

3.1.4 Analisis Kebutuhan Pengguna

Kebutuhan pengguna (*user requirement*) yang telah diperoleh dari hasil survei dan wawancara digambarkan pada table 3.1 sebagai berikut :

Tabel 3.1 *User Requirement*

| No | User | Tugas | User Requirement |
|----|------------------------------|--|--|
| 1 | <i>Onsite</i> | a. Input data <i>complain</i> yang di dapat dari rumah sakit. | a. Data <i>complain</i> yang di inputkan harus lengkap dan valid, disertakan dengan kerangan error modul. |
| 2 | Koordinator <i>Onsite</i> | a. Menerima dan mengecek <i>complain</i> dari <i>onsite</i> . b. Memilih <i>complain</i> dari <i>onsite</i> yang layak diteruskan ke <i>sofdev</i> . c. Mengoreksi dan merevisi jika terdapat deskripsi error modul yang kurang tepat dari <i>onsite</i> . | a. Data <i>complain</i> yang di terima oleh koordinator <i>sofdev</i> harus bener bernar di cek apakah <i>complain</i> tersebut layak di lanjutkan ke <i>sofdev</i> atau tidak. b. Memilih <i>complain</i> sesuai dengan ketentuan yang ada. c. Cek deskripsi <i>error</i> modul harus benar-benar dilakukan dengan teliti karena akan mempengaruhi penentuan kesulitan <i>complain</i> yang akan ditangani. |
| 2 | Sekretaris | a. Input data rumah sakit b. Input data modul c. Input data modul rumah sakit d. Input data kategori modul e. Input data status rumah sakit | a. Data rumah sakit hanya yang ditangani oleh PT. NCI Cabang Surabaya b. Data modul SIM RS Medismart c. Data modul pada rumah sakit |

| No | User | Tugas | Tanggung Jawab |
|----|--------------------|---|---|
| | | f. Input data user g. Input data jabatan Update status pada user | d. harus sesuai dengan kontrak kerjasama. e. Kategori modul adalah mudah, sedang, susah. (<i>easy, medium, hard</i>) Status rumah sakit adalah sesuai dengan kontrak (baru berjalan, sedang berjalan, akan berakhir.) f. Data user sesuai dengan yang akan menggunakan aplikasi g. Jabatan yang di inputkan harus sesuai dengan jabatan user yang ada di PT. NCI Update status pada user digunakan untuk fasilitas login pada website. |
| 3 | Koordinator Sofdev | a. Mengecek kategori complain yang masuk dan mengupdate kategori complain sesuai tingkat kesulitan (Easy, Medium, Hard). b. Mengecek ketepatan waktu penanganan <i>complain</i> . c. Update Status pada jadwal penanganan complain menjadi selesai (<i>Done</i>) Setelah mengecek complain. d. Menerima laporan <i>complain</i> per <i>sofdev</i> . e. Menerima laporan modul | a. Kategori penentuan kategori complain harus benar-benar sesuai dengan tingkat kesulitan yang ada. b. Waktu penanganan complain harus sesuai dengan jadwal, c. Complain harus benar-benar sesuai dengan alur yang diminta oleh rumah sakit sebelum mengupdate status pada jadwal. d. Laporan complain per sofdev berisi catatan data waktu penanganan complain yang dilakukan oleh sofdev. |

| | | | |
|---|------------------|--|--|
| | | sering terdapat complain. | e. Laporan modul sering terdapat complain berisi data modul yang paling sering di <i>complain</i> kan oleh rumah sakit selama satu periode. |
| 4 | Sovdef | <p>a. Menerima dan mengecek jadwal penanganan complain.</p> <p>b. Mengupdate status dari pending menjadi <i>on progress</i>.</p> | <p>a. Jadwal yang di buat akan menampilkan nama sofdef dan <i>complain</i> apa yang dikerjakan, karena itu jadwal harus di cek oleh <i>sofdev</i></p> <p>b. Setelah sofdev mengetahui dan menerima complain mana yang akan dikerjakan maka <i>sofdev</i> wajib mengupdate status pada jadwal dari pending menjadi <i>on progress</i></p> |
| 5 | Manajer Regional | <p>a. Menerima laporan rs berdasarkan masa kontrak per tahun.</p> <p>b. Menerima laporan rs berdasarkan jumlah complain.</p> | <p>a. Laporan rs berdasarkan masa kontrak pertahun berisi data semua rumah sakit waktu kontrak, di urutkan dari waktu kontrak yang akan habis hingga waktu kontrak yang paling awal.</p> <p>b. Laporan rs berdasarkan jumlah <i>complain</i> berisi data rumah sakit yang paling sering melakukan <i>complain</i> selama satu periode.</p> |

Pada aplikasi sistem informasi penjadwalan *complain* ini terdapat empat user yaitu *onsite*, sekretaris, koordinator *sofdev*, *sofdev* yang mempunyai tugas dan tanggung jawab yang berbeda pada aplikasi yang ditujukan agar aplikasi dapat berjalan dengan baik dan sesuai dengan alur yang dirancang.

3.2. Analisis Kebutuhan Data

Dari analisis kebutuhan pengguna telah dibuat, maka dibutuhkan data-data yang nantinya akan menunjang aplikasi yang akan di rancang, data-data tersebut meliputi:

1. Data Jabatan.

Data jabatan diperoleh peneliti dari pihak sekretaris PT. NCI cabang Surabaya. Dalam aplikasi penjadwalan penanganan complain ini data jabatan digunakan sebagai data master yang fungsinya adalah sebagai data yang mendukung untuk melengkapi data user.

2. Data Provinsi.

Data provinsi adalah data yang diinputkan sama dengan data provinsi seluruh Indonesia, data provinsi berfungsi sebagai data yang mendukung data kota.

3. Data Kota.

Data kota adalah data yang diinputkan sesuai dengan kota-kota yang ada di seluruh Indonesia. Sebelum menginputkan kota, harus memilih provinsi yang sesuai dengan kota tersebut.

4. Data User.

Data user adalah data yang nantinya akan di inputkan sendiri oleh user kedalam sistem. Sebelumnya user memperoleh *username* dan *password* dari sekretaris. Username menggunakan NIK yang sudah digunakan sehari-hari oleh karyawan PT. NCI, sedangkan *password* akan diberikan random dan nantinya dapat diganti sesuai dengan keinginan. Data yang di inputkan adalah data diri (nama, TTL, alamat, email dan jabatan.) Jabatan sendiri berguna dalam *privilege* dalam aplikasi penjadwalan complain.

5. Data Kategori Modul.

Data kategori modul adalah data yang diinputkan untuk mengkategorikan modul menjadi tiga bagian, yaitu *front office*, *back office* dan pununjang.

6. Data Modul.

Data modul berisi data nama-nama modul yang ada pada SIM RS Medismart. Terdapat kurang lebih 25 modul yang akan di inputkan ke dalam sistem.

7. Data Rumah Sakit.

Data rumah sakit adalah data customer dari PT. NCI khususnya dari cabang Surabaya. Data rumah sakit akan di lengkapi oleh data modul, modul yang diinputkan adalah modul yang digunakan oleh rumah sakit yang menjadi costumer dari PT. NCI cabang Surabaya.

8. Data Modul Rumah Sakit.

Data modul rumah sakit adalah data dari modul yang digunakan oleh rumah sakit. Modul yang dicatat adalah modul berdasarkan modul SIM RS Medismart.

9. Data Status Rumah Sakit.

Data status rumah sakit adalah data yang di inputkan untuk mendukung proses penentuan kriteria dari *complain*. Status yang di inputkan adalah status rumah sakit kontrak baru, kontrak sedang berjalan dan kontrak akan habis.

10. Data Onsite Rumah Sakit.

Data onsite rumah sakit adalah data yang digunakan untuk menunjukkan onsite yang berada di rumah sakit.

11. Data *Complain*.

Data *complain* adalah data yang diperoleh dari rumah sakit dan diberikan kepada pihak *onsite* untuk diinputkan kedalam sistem. Data

yang di inputkan antara lain nama rumah sakit, tanggal, modul, deskripsi *error* modul.

12. Data Status *List Complain*.

Data status *list complain* adalah data yang di inputkan untuk mendukung proses penjadwalan dari penanganan *complain*. Status yang di inputkan adalah *pending*, diterima dan ditolak.

13. Data Kategori *Complain*.

Data Kategori complain adalah data yang digunakan untuk menentukan *complain* berdasarkan tingkat kesulitannya. Complain dikelompokkan menjadi kategori *easy*, *medium*, *hard* dan *very hard*.

14. Data Status Penjadwalan.

Data status penjadwalan ada data yang digunakan untuk memberikan status pada complain yang sudah terjadwal, status yang digunakan adalah *pending*, *queue*, *progress*, *done*.

3.3. Analisis Kebutuhan Fungsi

Tahap selanjutnya setelah mendeskripsikan kebutuhan pengguna dan kebutuhan data maka dapat dirancang sebuah kebutuhan *functional* dari aplikasi yang akan di rancang. Tahap Fungsional adalah tahap yang digunakan untuk menjabarkan seluruh fungsi-fungsi yang didapat dari kebutuhan pengguna.

Tabel 3.2 Kebutuhan Fungsi Input Master Jabatan

| Nama Fungsi | Fungsi Menginputkan Data Master Jabatan | |
|--------------------|--|--|
| <i>Stakeholder</i> | Bagian Sekretaris | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menginputkan data jabatan. | |
| Kondisi Awal | 1. Data User (<i>Username</i> dan <i>Password</i>) 2. Data Jabatan | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memasukkan Data Jabatan Pegawai | |
| | 1. Sekretaris melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 2. Sekretaris memilih menu master jabatan. | Sistem menampilkan menu <i>home</i> untuk sekretaris, setelah memilih master jabatan maka sistem akan menampilkan form master jabatan. |
| | 3. Sekretaris menginputkan data jabatan dan menyimpan data jabatan, | a. Sistem akan mengecek kolom nama jabatan, jika saat mengisi kolom dan menekan tombol <i>save</i> dan nama jabatan sudah pernah menggunakan nama yang sama maka sistem akan menampilkan “Nama Jabatan Sudah Terdaftar”. b. Sistem akan menyimpan data jabatan kedalam table jabatan. |

Tabel 3.3 Kebutuhan Fungsi Input Master Provinsi

| Nama Fungsi | Fungsi Menginputkan Data Master Provinsi | |
|--------------------|---|---|
| <i>Stakeholder</i> | Bagian Sekretaris | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menginputkan data provinsi. | |
| Kondisi Awal | 1. Data User (<i>Username</i> dan <i>Password</i>) 2. Data provinsi | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memasukkan Data Jabatan Pegawai | |
| | 1. Sekretaris melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 2. Sekretaris memilih menu master provinsi. | Sistem menampilkan menu <i>home</i> untuk sekretaris, setelah memilih master provinsi maka sistem akan menampilkan form master provinsi. |
| | 3. Sekretaris menginputkan data provinsi dan menyimpan data jabatan, | a. Sistem akan mengecek kolom nama provinsi, jika saat mengisi kolom dan menekan tombol <i>save</i> dan nama provinsi sudah pernah menggunakan nama yang sama maka sistem akan menampilkan “Provinsi Sudah Terdaftar”. b. Sistem akan menyimpan data jabatan kedalam table provinsi. |

Tabel 3.4 Kebutuhan Fungsi Input Master Kota

| | | |
|---|---|--|
| Nama Fungsi | Fungsi Menginputkan Data Master Kota | |
| Stakeholder | Bagian Sekretaris | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menginputkan data kota. | |
| Kondisi Awal | 1. Data user (<i>Username</i> dan <i>Password</i>) 2. Data provinsi 3. Data kota | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memasukkan Data Jabatan Pegawai | |
| | 1. Sekretaris melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 2. Sekretaris memilih menu master kota. | Sistem menampilkan menu <i>home</i> untuk sekretaris, setelah memilih master kota maka sistem akan menampilkan form master kota. |
| 3. Sekretaris menginputkan data provinsi dan menyimpan data jabatan | a. Sistem akan mengecek kolom nama provinsi, jika saat mengisi kolom dan menekan tombol <i>save</i> dan nama kota sudah pernah menggunakan nama yang sama maka sistem akan menampilkan “kota Sudah Terdaftar”. b. Sistem akan menyimpan data jabatan kedalam table kota. | |

Tabel 3.5 Kebutuhan Fungsi Input Master User

| Nama Fungsi | Fungsi Menginputkan Data Master User | |
|--|---|--|
| <i>Stakeholder</i> | <i>Onsite/ Koordinator Onsite, Koordinator Sofdev, Sofdev</i> | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menginputkan data jabatan | |
| Kondisi Awal | 1. Data User (<i>Username</i> dan <i>Password</i>) 2. Data Profile 3. Data Jabatan | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memasukkan Data <i>Profile</i> User dan <i>Jabatan</i> User | |
| | 1. User melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 2. User memilih menu <i>profile</i> . | Sistem menampilkan menu <i>home</i> untuk user, setelah memilih menu <i>profile</i> maka sistem akan menampilkan form <i>profile</i> yang wajib dilengkapi oleh user. |
| 3. User menginputkan data <i>profile</i> , data jabatan sesuai dengan jabatan di kantor dan menyimpan data user. | a. Sistem akan mengecek kolom yang masih kosong, jika saat menekan tombol <i>save</i> terdapat kolom yang masih kosong maka sistem akan menampilkan “Data Kosong Harap Isi Data”. b. Sistem akan menyimpan data user kedalam table user. | |

Tabel 3.6 Kebutuhan Fungsi Input Master Kategori Modul

| Nama Fungsi | Fungsi Menginputkan Data Master Kategori Modul | |
|---|---|--|
| <i>Stakeholder</i> | Bagian Sekretaris | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menginputkan data Kategori Modul. | |
| Kondisi Awal | c. Data User (<i>Username</i> dan <i>Password</i>) d. Data Kategori Modul | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memasukkan Data Master Jabatan | |
| | 1. Sekretaris melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 2. Sekretaris memilih menu master Modul. | Sistem menampilkan menu <i>home</i> untuk sekretaris, setelah memilih master kategori modul maka sistem akan menampilkan form master kategori modul. |
| 3. Sekretaris menginputkan data Modul dan menyimpan data Modul. | e. Sistem akan mengecek kolom nama modul, jika saat mengisi kolom dan menekan tombol <i>save</i> dan nama kategori modul sudah pernah menggunakan nama yang sama maka sistem akan menampilkan “Nama Kateogri Modul Sudah Terdaftar”. a. Sistem akan menyimpan data jabatan kedalam table <i>ktgr_modul</i> . | |

Tabel 3.7 Kebutuhan Fungsi Input Master Modul

| Nama Fungsi | Fungsi Menginputkan Data Master Modul | |
|--------------------|--|--|
| <i>Stakeholder</i> | Bagian Sekretaris | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menginputkan data Modul. | |
| Kondisi Awal | f. Data User (<i>Username</i> dan <i>Password</i>) g. Data Modul | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memasukkan Data Master Jabatan | |
| | 4. Sekretaris melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 5. Sekretaris memilih menu master Modul. | Sistem menampilkan menu <i>home</i> untuk sekretaris, setelah memilih master modul maka sistem akan menampilkan form master modul. |
| | 6. Sekretaris menginputkan data Modul dan menyimpan data Modul. | h. Sistem akan mengecek kolom nama modul, jika saat mengisi kolom dan menekan tombol <i>save</i> dan nama modul sudah pernah menggunakan nama yang sama maka sistem akan menampilkan “Nama Modul Sudah Terdaftar”. b. Sistem akan menyimpan data jabatan kedalam table modul. |

Tabel 3.8 Kebutuhan Fungsi Input Master Status Rumah Sakit

| Nama Fungsi | Fungsi Menginputkan Data Master Status Rumah Sakit | |
|---|---|--|
| <i>Stakeholder</i> | Bagian Sekretaris | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menginputkan data Status Rumah Sakit. | |
| Kondisi Awal | 1. Data User (<i>Username</i> dan <i>Password</i>) 2. Data Status Rumah Sakit | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memasukkan Data Status Rumah Sakit | |
| | 1. Sekretaris melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 2. Sekretaris memilih menu master Status Rumah Sakit. | Sistem menampilkan menu <i>home</i> untuk sekretaris, setelah memilih master status rumah sakit maka sistem akan menampilkan form master status rumah sakit. |
| 3. Sekretaris menginputkan data Status Rumah Sakit dan menyimpan data Status Rumah Sakit. | <p>a. Sistem akan mengecek kolom status rumah sakit, jika saat mengisi kolom dan menekan tombol <i>save</i> dan nama modul sudah pernah menggunakan nama yang sama maka sistem akan menampilkan “Nama Status RS Sudah Terdaftar”.</p> <p>c. Sistem akan menyimpan data status rumah sakit kedalam table <i>status_RS</i>.</p> | |

Tabel 3.9 Kebutuhan Fungsi Input Master Rumah Sakit

| Nama Fungsi | Fungsi Menginputkan Data Master Rumah Sakit | |
|---|---|--|
| <i>Stakeholder</i> | Bagian Sekretaris | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menginputkan data Rumah Sakit. | |
| Kondisi Awal | 1. Data User (<i>Username</i> dan <i>Password</i>) 2. Data Modul 3. Data Status Rumah Sakit 4. Data Rumah sakit | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memasukkan Data Rumah Sakit | |
| | 1. Sekretaris melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 2. Sekretaris memilih menu master Rumah Sakit. | Sistem menampilkan menu <i>home</i> untuk sekretaris, setelah memilih master Rumah Sakit maka sistem akan menampilkan form master Status Rumah Sakit. |
| 3. Sekretaris menginputkan data Rumah Sakit yang di dalam nya terdapat data modul dan data status rumah sakit kemudian menyimpan data Status Rumah Sakit. | a. Sistem Sistem akan mengecek kolom yang masih kosong, jika saat menekan tombol <i>save</i> terdapat kolom yang masih kosong maka sistem akan menampilkan “Data Kosong Harap Isi Data”. b. Sistem akan mengecek kolom nama rumah sakit, jika saat mengisi kolom dan menekan tombol <i>save</i> dan nama | |

| | | |
|--|--|---|
| | | <p>rumah sakit sudah pernah digunakan maka sistem akan menampilkan “Nama Rumah Sakit Sudah Terdaftar”.</p> <p>c. Sistem akan menyimpan data rumah sakit kedalam table rumah_sakit</p> |
|--|--|---|

Tabel 3.10 Kebutuhan Fungsi Input Master Modul Rumah Sakit

| Nama Fungsi | Fungsi Menginputkan Data Master Modul Rumah Sakit | |
|---------------------|--|--|
| Stakeholder | Bagian Sekretaris | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menginputkan data Modul Rumah Sakit. | |
| Kondisi Awal | 1. Data User (<i>Username</i> dan <i>Password</i>) 2. Data Modul 3. Data Rumah sakit | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memasukkan Data Modul Rumah Sakit | |
| | 1. Sekretaris melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 2. Sekretaris memilih menu master Rumah Sakit. | Sistem menampilkan menu <i>home</i> untuk sekretaris, setelah memilih master Modul Rumah Sakit maka sistem akan menampilkan form master Status Rumah Sakit. |
| | 3. Sekretaris menginputkan data Modul Rumah Sakit menyimpan data modul | a. Sistem Sistem akan mengecek kolom yang masih kosong, jika saat menekan tombol <i>save</i> terdapat |

| | | |
|--|--------------|---|
| | Rumah Sakit. | <p>kolom yang masih kosong maka sistem akan menampilkan “Data Kosong Harap Isi Data”.</p> <p>b. Sistem akan mengecek kolom nama modul rumah sakit, jika saat mengisi kolom dan menekan tombol <i>save</i> dan nama rumah sakit sudah pernah digunakan maka sistem akan menampilkan “Nama Modul Rumah Sakit Sudah Terdaftar”.</p> <p>c. Sistem akan menyimpan data rumah sakit kedalam table <i>rumah_sakit</i> dan <i>modul_rs</i>.</p> |
|--|--------------|---|

Tabel 3.11 Kebutuhan Fungsi Input Master *Onsite* Rumah Sakit

| | | |
|---------------------|---|--|
| Nama Fungsi | Fungsi Menginputkan Data Master Onsite Rumah Sakit | |
| Stakeholder | Bagian Sekretaris | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menginputkan data Onsite Rumah Sakit. | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Data User (<i>Username</i> dan <i>Password</i>) 2. Data User (<i>Onsite</i>) 3. Data Rumah sakit | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memasukkan Data Onsite Rumah Sakit | |
| | 1. Sekretaris melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |

| | | |
|--|--|--|
| | 2. Sekretaris memilih menu master Onsite Rumah Sakit. | Sistem menampilkan menu <i>home</i> untuk sekretaris, setelah memilih master Onsite Rumah Sakit maka sistem akan menampilkan form master Onsite RS. |
| | 3. Sekretaris menginputkan data Onsite Rumah Sakit sesuai dengan keadaan yang ada dan menyimpan data tersebut. | <p>a. Sistem akan mengecek kolom yang masih kosong, jika saat menekan tombol <i>save</i> terdapat kolom yang masih kosong maka sistem akan menampilkan “Data Kosong Harap Isi Data”.</p> <p>b. Sistem akan mengecek kolom nama onsite rumah sakit, jika saat mengisi kolom dan menekan tombol <i>save</i> dan nama rumah sakit sudah pernah digunakan maka sistem akan menampilkan “Nama Onsite Sudah Terdaftar”.</p> <p>c. Sistem akan menyimpan data rumah sakit kedalam table <code>rumah_sakit</code> dan <code>onsite</code>.</p> |

Tabel 3.12 Kebutuhan Fungsi Input Master Kategori *Complain*

| | | |
|--|---|--|
| Nama Fungsi | Fungsi Menginputkan Data Master Kategori <i>Complain</i> | |
| Stakeholder | Bagian Koordinator <i>Sofdev</i> | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menginputkan data kategori <i>Complain</i> . | |
| Kondisi Awal | 1. Data User (<i>Username</i> dan <i>Password</i>) 2. Data Kateogri <i>Complain</i> | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memasukkan Data Kategori <i>Complain</i> | |
| | 1. Koordonator <i>sofdev</i> melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 2. Koordonator <i>sofdev</i> memilih menu master kategori <i>complain</i> . | Sistem menampilkan menu <i>home</i> untuk Koordonator <i>sofdev</i> , setelah memilih master kategori <i>complain</i> maka sistem akan menampilkan form master kategori <i>complain</i> . |
| 3. Koordonator <i>sofdev</i> menginputkan data Status Rumah Sakit dan menyimpan data Status Rumah Sakit. | a. Sistem akan mengecek kolom kategori, jika saat mengisi kolom dan menekan tombol <i>save</i> dan kategori sudah pernah menggunakan nama yang sama maka sistem akan menampilkan “Kategori Sudah Terdaftar”. b. Sistem akan menyimpan data kategori <i>complain</i> kedalam table <i>ktgr_complain</i> . | |

Tabel 3.13 Kebutuhan Fungsi Input Data Master Status *List Complain*

| | | |
|---------------------|---|---|
| Nama Fungsi | Fungsi Menginputkan Data Master Status <i>List Complain</i> | |
| Stakeholder | Bagian Sekretaris | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menginputkan data master <i>list complain</i> . | |
| Kondisi Awal | 1. Data User (<i>Username</i> dan <i>Password</i>) 2. Data Status <i>List Complain</i> | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memasukkan Data Status <i>List Complain</i> | |
| | 1. Sekretaris melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 2. Sekretaris memilih menu master status <i>list complain</i> . | Sistem menampilkan menu <i>home</i> untuk sekretaris, setelah memilih master status <i>list complain</i> maka sistem akan menampilkan form master status <i>list complain</i> |
| | 3. Sekretaris menginputkan data status <i>list complain</i> dan menyimpan data Status Rumah Sakit. | a. Sistem akan mengecek kolom nama status <i>list complain</i> , jika saat mengisi kolom dan menekan tombol <i>save</i> dan nama status <i>list complain</i> sudah pernah menggunakan nama yang sama maka sistem akan menampilkan “Nama Status Sudah Terdaftar”. b. Sistem akan menyimpan data status <i>list complain</i> kedalam table <i>status_list_complain</i> . |

Tabel 3.14 Kebutuhan Fungsi Input Data Master Status Penjadwalan

| Nama Fungsi | Fungsi Menginputkan Data Master Status Penjadwalan | |
|---|---|--|
| <i>Stakeholder</i> | Bagian Sekretaris | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menginputkan data master status penjadwalan. | |
| Kondisi Awal | 1. Data User (<i>Username</i> dan <i>Password</i>) 2. Data status penjadwalan | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memasukkan Data Status Penjadwalan | |
| | 1. Sekretaris melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 2. Sekretaris memilih menu master status penjadwalan. | Sistem menampilkan menu <i>home</i> untuk sekretaris, setelah memilih master status penjadwalan maka sistem akan menampilkan form master status penjadwalan. |
| 3. Sekretaris menginputkan data status penjadwalan dan menyimpan data Status Rumah Sakit. | a. Sistem akan mengecek kolom nama status penjadwalan, jika saat mengisi kolom dan menekan tombol <i>save</i> dan nama status penjadwalan sudah pernah menggunakan nama yang sama maka sistem akan menampilkan “Nama Status Sudah Terdaftar”. b. Sistem akan menyimpan data status <i>list complain</i> kedalam table <i>sts_penjadwalan</i> . | |

Tabel 3.15 Kebutuhan Fungsi Input Data *List Complain*

| | | |
|--|---|--|
| Nama Fungsi | Fungsi Menginputkan Data <i>List Complain</i> | |
| Stakeholder | Bagian <i>Onsite</i> | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menginputkan data <i>List Complain</i> . | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Data User (<i>Username</i> dan <i>Password</i>) 2. Data Rumah Sakit 3. Data Status Rumah Sakit 4. Data Modul 5. Data Keterangan <i>Error Modul</i> | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memasukkan Data <i>List Complain</i> | |
| | 1. <i>Onsite</i> melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 2. <i>Onsite</i> memilih menu <i>Input Complain</i> . | Sistem menampilkan menu <i>home</i> untuk <i>Onsite</i> , setelah memilih menu <i>input complain</i> maka sistem akan menampilkan form <i>input complain</i> . |
| 3. <i>Onsite</i> menginputkan data <i>list complain</i> dan menyimpan <i>List Complain</i> . | <ol style="list-style-type: none"> a. Sistem Sistem akan mengecek kolom yang masih kosong, jika saat menekan tombol <i>save</i> terdapat kolom yang masih kosong maka sistem akan menampilkan “Data Kosong Harap Isi Data”. b. Sistem akan menyimpan data <i>List Complain</i> kedalam table <i>complain</i>. | |

Tabel 3.16 Kebutuhan Fungsi Pemilihan Complain yang Layak Ditangani

| Nama Fungsi | Fungsi Pemilihan Complain yang Layak Ditangani | |
|---|--|--|
| Stakeholder | Bagian Koordinator <i>Onsite</i> | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menginputkan complain yang benar-benar layak ditangani oleh pihak sofdev atau kah hanya di tangani oleh pihak onsite saja. | |
| Kondisi Awal | 1. Data User (<i>Username</i> dan <i>Password</i>) 2. Data <i>List Complain</i> 3. Data Modul 4. Data Keterangan <i>Error Modul</i> | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memilih Complain yang Layak Ditangani | |
| | 1. Koordinator <i>Onsite</i> melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> . |
| | 2. Koordinator <i>Onsite</i> memilih menu <i>Input Complain</i> . | Sistem menampilkan menu <i>home</i> untuk Koordinator <i>Onsite</i> , setelah memilih menu <i>list complain</i> maka sistem akan menampilkan <i>list complain</i> yang di <i>inputkan</i> oleh seluruh onsite yang ada di rumah sakit. |
| 3. Koordinator <i>Onsite</i> mengecek <i>list complain</i> dengan melihat keterangan <i>error modul</i> , memilih <i>complain</i> yang layak dan mengupdate status <i>List Complain</i> . | a. Sistem akan mengupdate status pada data <i>List Complain</i> kedalam table <i>complain</i> . | |

Tabel 3.17 Kebutuhan Fungsi Penambahan Kategori Pada Complain

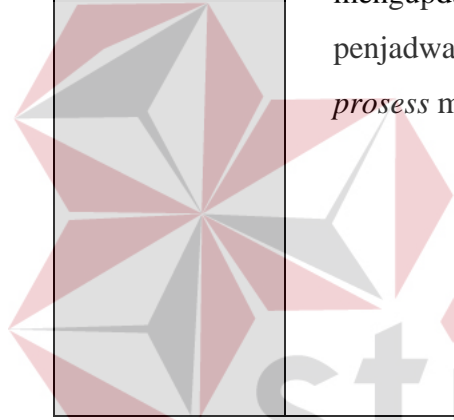
| Nama Fungsi | Fungsi Penambahan Kategori Pada Complain | |
|--|--|--|
| <i>Stakeholder</i> | Bagian Koordinator <i>Sofdev</i> | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menambahkan kategori pada <i>complain</i> yang sudah terverifikasi oleh koordinator <i>onsite</i> . | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Data User (<i>Username</i> dan <i>Password</i>) 2. Data <i>List Complain</i> terverifikasi 3. Data Modul 4. Data Keterangan <i>Error</i> Modul | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memilih Complain yang Layak Ditangani | |
| | 1. Koordinator <i>sofdev</i> melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 2. Koordinator <i>sofdev</i> memilih detail <i>complain</i> . | Sistem menampilkan menu <i>home</i> untuk Koordinator <i>sofdev</i> , setelah memilih detail <i>complain</i> maka sistem akan menampilkan data <i>complain</i> yang sudah disetujui oleh Koordinator <i>onsite</i> . |
| 3. Koordinator <i>Sofdev</i> mengecek <i>list complain</i> dengan melihat keterangan <i>error</i> modul, kemudian memberi kategori pada <i>complain</i> sesuai dengan tingkat kesulitan yang | a. Sistem akan mengupdate data kategori pada table <i>complain</i> | |

| | | |
|--|--------------------------------------|--|
| | ada pada deskripsi <i>complain</i> . | |
|--|--------------------------------------|--|

Tabel 3.18 Kebutuhan Fungsi *Update* Status Penjadwalan

| | | |
|---------------------|---|---|
| Nama Fungsi | Fungsi <i>Update</i> Status Penjadwalan | |
| Stakeholder | Bagian <i>Sofdev</i> Koordinator <i>Sofdev</i> | |
| Deskripsi | Fungsi ini merupakan suatu alur proses yang dilakukan untuk menambahkan kategori pada <i>complain</i> yang sudah terverifikasi oleh koordinator <i>onsite</i> . | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. Data User (<i>Username</i> dan <i>Password</i>) 2. Jadwal Penanganan <i>Complain</i> 3. Data Status Penjadwalan 4. Data <i>Complain</i> | |
| Alur Normal | Aksi Stake Holder | Respon Sistem |
| | Memilih <i>Complain</i> yang Layak Ditangani | |
| | 1. <i>Sofdev</i> melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 2. <i>Sofdev</i> memilih detail <i>jadwal</i> dari jadwal penanganan <i>complain</i> yang tampil di <i>Home</i> dan memilih <i>complain</i> yang akan ditangani dan mengupdate status penjadwalan. | Sistem menampilkan menu <i>home</i> untuk <i>Sofdev</i> yang berisi jadwal penanganan <i>complain</i> , kemudian memilih detail <i>jadwal</i> maka sistem akan list <i>complain</i> yang sudah terjadwal setelah itu <i>sofdev</i> akan memilih <i>complain</i> yang paling atas untuk ditangani terlebih dahulu, setelah milih <i>complain</i> yang ditangani maka otomatis akan mengupdate status |

| | | |
|--|---|---|
| | | penjadwalan menjadi <i>on progress</i> dan tanggal pengerjaan pada table <i>complain</i> dan <i>detail_complain</i> |
| | 3. Koordinator <i>sofdev</i> melakukan proses login pada sistem. | Sistem akan menampilkan form login sebagai yang berisikan <i>username</i> dan <i>password</i> . <i>Username</i> menggunakan NIK dari pegawai, kemudian <i>password</i> dapat diganti sesuai dengan keinginan pengguna. |
| | 4. Koordinator <i>Sofdev</i> mengupdate status penjadwalan dari <i>on process</i> menjadi <i>done</i> | Sistem menampilkan menu <i>home</i> untuk Koordinator <i>sofdev</i> , setelah memilih detail jadwal, mengecek <i>complain</i> yang sudah dikerjakan ketika <i>complain</i> sesuai alur maka koordinator <i>sofdev</i> akan mengupdate status penjadwalan menjadi <i>done</i> dan tanggal selesai ada tabel <i>complain</i> dan tabel <i>detail_complain</i> . |



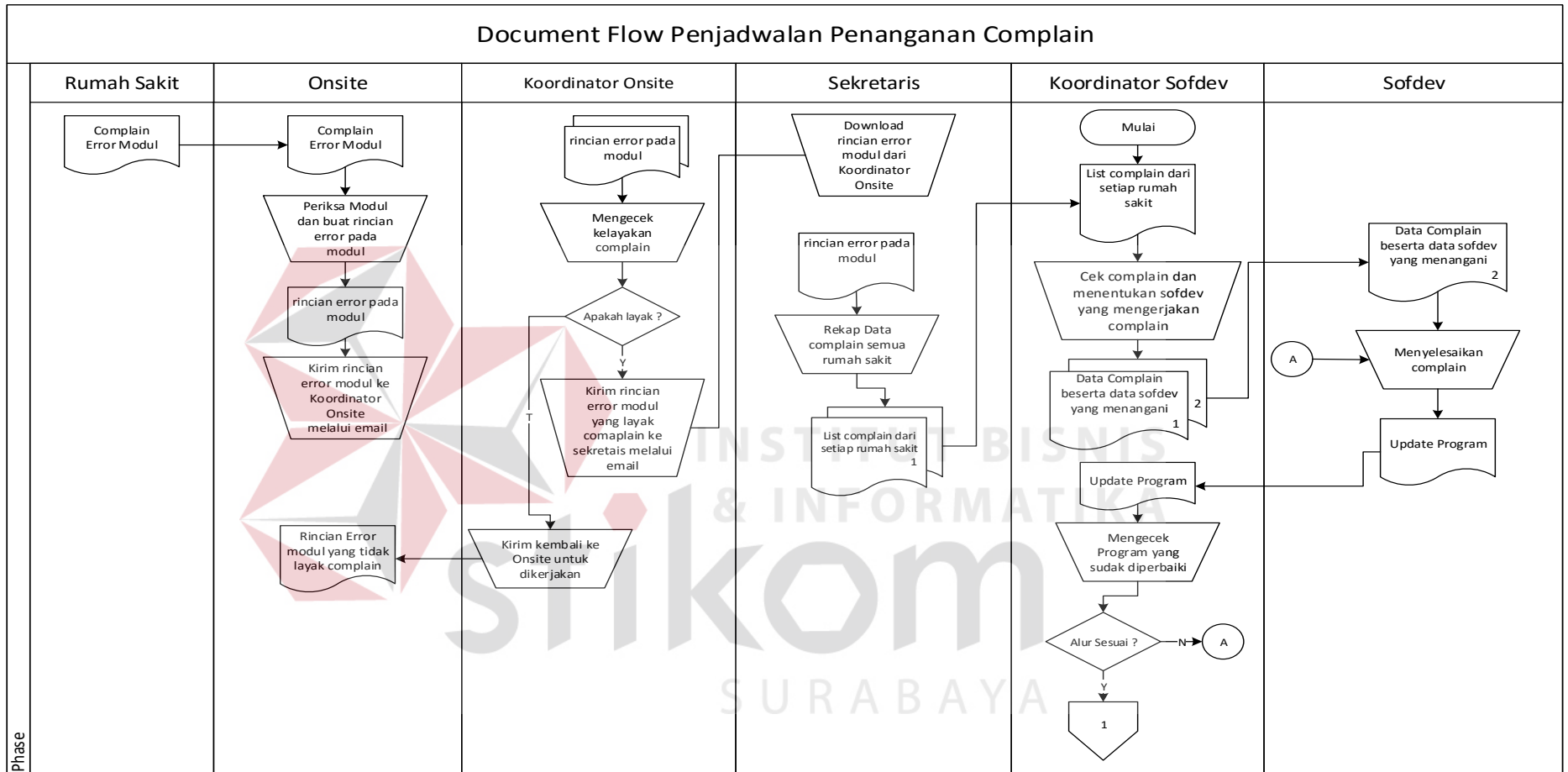
3.4. Perancangan Sistem

3.4.1. *Document Flow* Penjadwalan *Complain*

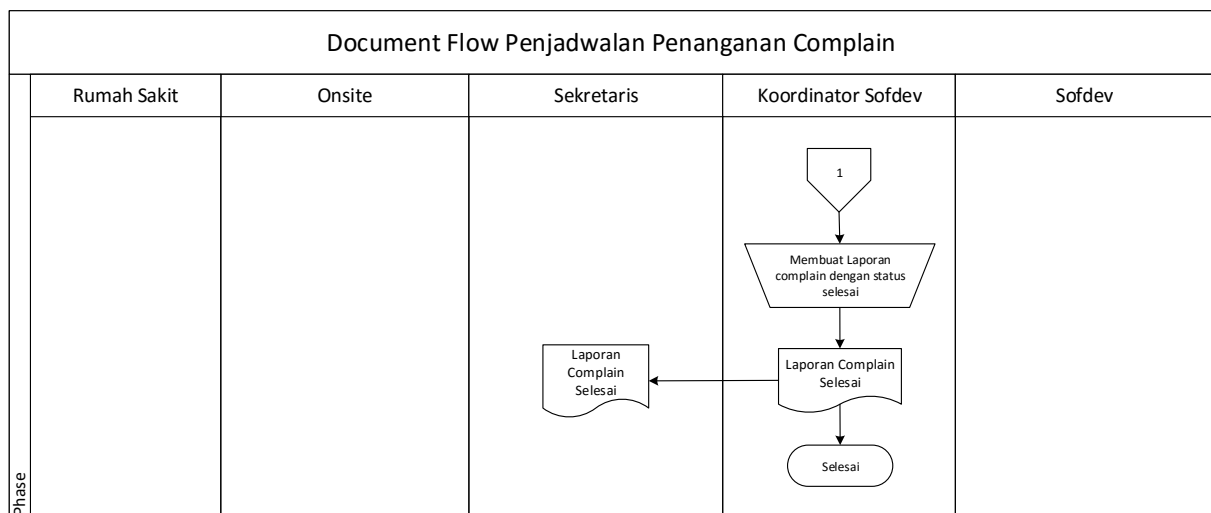
Dari hasil survei yang dilakukan pada PT. NCI cabang Surabaya, maka dilakukan sebuah analisa sistem yang menggambarkan proses penjadwalan *complain* yang berjalan selama ini.

Proses penjadwalan *complain* yang ada PT. NCI cabang Surabaya digambarkan dalam bentuk *document flow* pada gambar 3.1 dan gambar 3.2





Gambar 3.1 Document Flow Penjadwalan Complain



Gambar 3.2 Document Flow Penjadwalan Complain

Gambar 3.1 dan gambar 3.2 menjelaskan tentang document flow penjadwalan penanganan complain. Proses manual ini dimulai dari rumah sakit yang memberikan data *complain* kepada pihak *onsite*. Pihak *onsite* memeriksa modul yang *error* dan membuat rincian tentang modul yang *error* tersebut. Rincian *error* modul akan di kirimkan ke koordinator *onsite*. Koordinator *onsite* akan menentukan apakah complain tersebut layak untuk dilanjutkan ke pihak *sofdev* atau akan dikerjakan oleh *onsite* saja. Setelah menentukan kelayakan *complain* maka akan dirim ke sekretaris melalui email.

Sekretaris akan *mendownload* rincian modul *error* yang dikirim oleh koordinator *onsite* yang masuk ke email, rincian modul error akan di rekap oleh sekretaris kedalam Ms. Excel kemudian di cetak dan diberikan kepada koodinator *sofdev*.

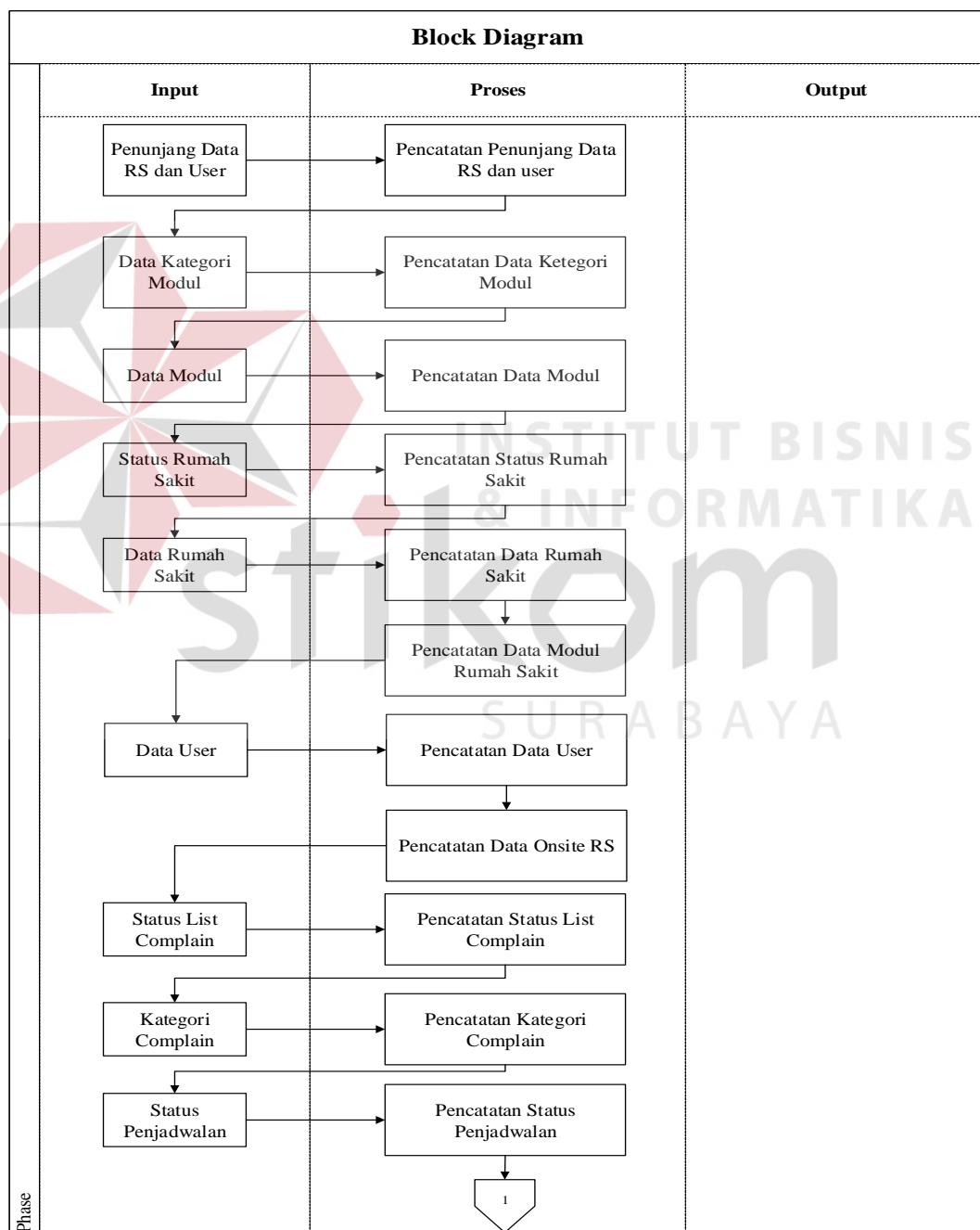
Koordinator Sofdev akan mengecek *error* tersebut dan menentukan complain yang akan dikerjakan dan sofdev siapa yang akan mengerjakan complain tersebut.

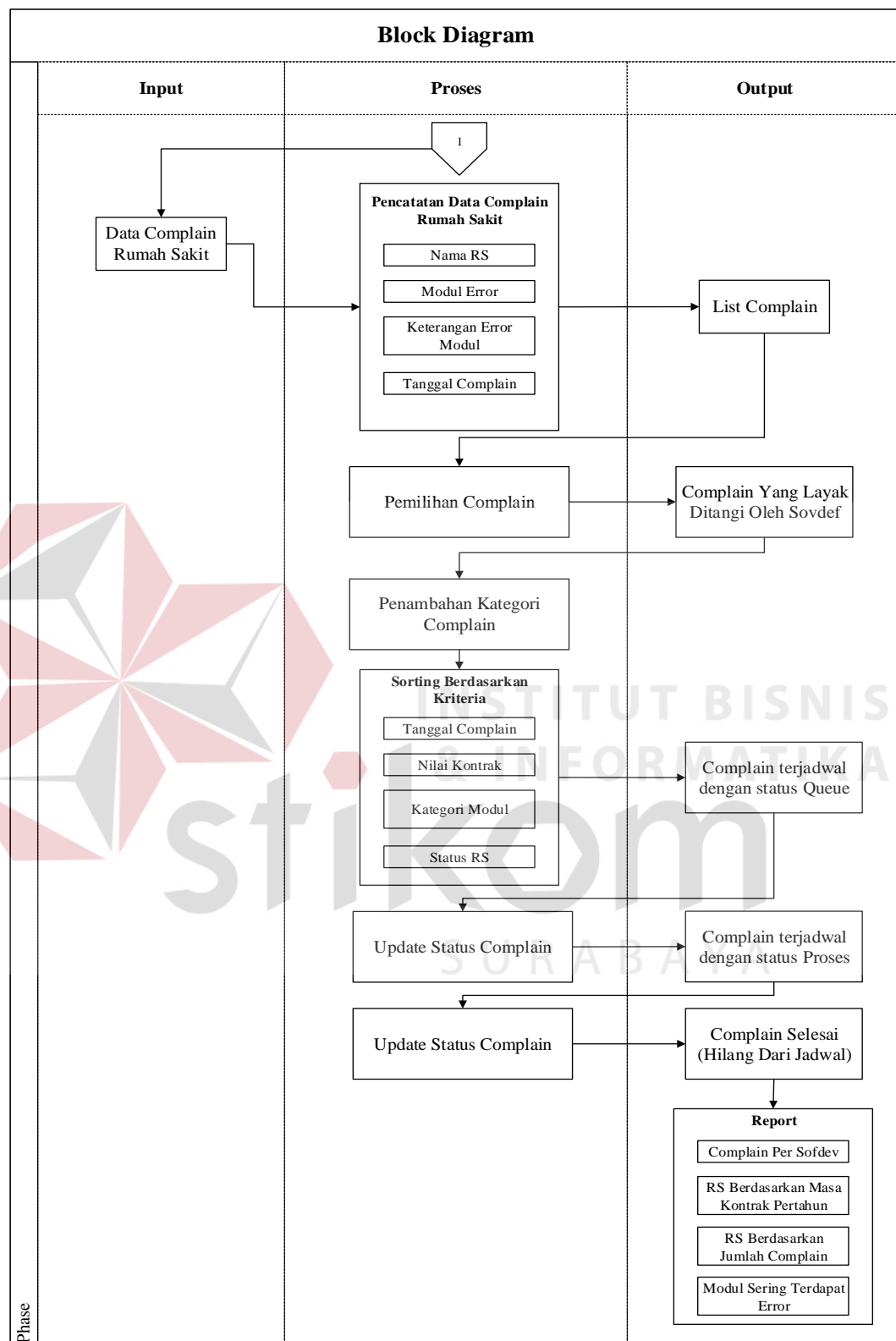
Setelah *sofdev* menyelesaikan complain dan memberikan update program yang baru maka koordinator sofdev akan mengecek program tersebut apakah sudah benar-benar terselesaikan masalah yang ada. Jika masih dimukan error yang sama maka akan di kembalikan ke *sofdev* yang bersangkutan untuk dikerjakan ulang, jika program sudah sesuai maka koordinator *sofdev* akan membuat sebuah

laporan tentang complain yang sudah di selesaikan yang di tujukan kepada sekretaris.

3.4.2 Alur Sistem

Alur Sistem pada aplikasi penjadwalan penanganan complain pada PT. Nuansa Cerah Informasi ini digambarkan pada blok diagram pada gambar 3.3





Gambar 3.3 *Block Diagram* Aplikasi Penjadwalan *Complain* Berbasis Web Pada PT. Nuansa Cerah Informasi Cabang Surabaya

Block diagram di atas dapat menggambarkan informasi yang dihasilkan yaitu laporan penjadwalan dan jadwal penjadwalan. Jadwal penjadwalan penanganan *complain* berupa laporan pada halaman home yang memberikan informasi mengenai jadwal *complain* yang terjadwal dalam sehari. Penjelasan *block diagram* di atas adalah sebagai berikut:

A. *Input*

1. Penunjang Data RS

Penunjang data RS (Rumah Sakit) dan user berisi data jabatan, provinsi, kota, modul rumah sakit, onsite RS. Data-data tersebut digunakan untuk menunjang dan melengkapi data-data RS dan user.

2. Data Kategori Modul

Data kategori modul digunakan untuk mengkategorikan modul pada tiga kategori, yaitu kategori *front office*, *penunjang*, *back office*. Dari kategori modul akan digunakan dalam proses *sorting complain* yang masuk sebelum dijadwalkan.

3. Data Modul

Data modul adalah data yang berisi modul modul yang ada di SIM RS Medismart. Sebelum mengisi data modul terlebih dulu harus memilih kategori modul sesuai dengan modul yang akan di inputkan.

4. Data Status Rumah Sakit

Data status rumah sakit berisi data status untuk rumah sakit. Status rumah sakit digolongkan menjadi tiga golongan, yaitu kontrak akan habis, kontrak baru, kontrak berjalan. Status rumah sakit ini juga akan dijadikan sebagai kriteria dalam proses *sorting complain* sebelum dijadwalkan.

5. Data Rumah Sakit

Data rumah sakit berisi data-data yang bersangkutan dengan rumah sakit, mulai dari nama rumah sakit, alamat status rumah sakit, hingga nilai kontrak rumah sakit. Nilai kontrak rumah sakit juga akan

dijadikan sebagai kriteria dalam proses *sorting* complain sebelum dijadwalkan.

6. Data *User*

Data user berisikan data-data yang berhubungan dengan *user* dan data yang berguna untuk user dalam menjalankan aplikasi. Data *user* berisikan NIK, *password*, jabatan sesuai dengan yang dimiliki oleh *user*, alamat, hingga email. NIK dan *password* digunakan oleh *user* dalam proses *login* pada aplikasi.

7. Data Status List Complain

Data status list complain berisikan data yang digunakan dalam mendukung proses transaksi input list complain. Status list complain berisi status pending, diterima dan ditolak.

8. Data Kategori Complain

Data kategori complain berisikan data yang digunakan dalam mendukung proses transaksi penambahan kategori complain. Kategori complain berisi kategori *easy*, *medium*, *hard*, *very hard*.

9. Data Status Penjadwalan

Data status penjadwalan adalah status yang nantinya akan tampil pada jadwal penanganan complain. Terdapat empat status yaitu *pending*, *queue*, *progress*, *done*.

10. Data Complain Rumah Sakit

Data complain rumah sakit adalah data yang di inputkan oleh *onsite* saat terjadi *error* pada modul yang ada di rumah sakit.

B. Proses

1. Proses Pencatatan Data Penunjang RS dan user

Proses pencatatan data penunjang RS dan user dilakukan agar sistem dapat menyimpan penunjang yang berguna dalam melengkapi data rumah sakit dan user.

2. Proses Pencatatan Data Kategori Modul

Proses pencatatan data kategori modul dilakukan agar sistem dapat menyimpan data kategori modul dan juga berguna dalam mendukung data modul. Data kategori modul digunakan juga untuk proses *sorting complain* sebelum dijadwalkan.

3. Proses Pencatatan Data Modul

Proses pencatatan data modul dilakukan agar sistem dapat menyimpan data modul, modul yang di inputkan harus sesuai dengan jumlah yang ada pada modul SIM RS Medismart. Sebelum menginputkan modul terlebih dulu harus memilih kategori modul sesuai dengan modul yang diinputkan.

4. Proses Pencatatan Status Rumah Sakit

Proses pencatatan status rumah sakit dilakukan untuk mendukung proses *sorting complain* dan juga untuk melengkapi data rumah sakit.

5. Proses Pencatatan Data Rumah Sakit

Proses pencatatan data rumah sakit dilakukan untuk menyimpan data-data yang berhubungan dengan rumah sakit. Mulai dari data rumah sakit hingga nilai kontrak. Nilai kontrak rumah sakit juga berfungsi dalam kategori *sorting complain*.

6. Proses Pencatatan Modul Rumah Sakit

Proses pencatatan modul rumah sakit digunakan untuk menyimpan data modul yang ada pada setiap rumah sakit. Modul yang diinputkan harus sesuai dengan modul yang ada pada setiap rumah sakit.

7. Proses Pencatatan Data User

Proses pencatatan data user berfungsi untuk menyimpan data user. Data yang disimpan mulai dari NIK, nama user, alamat, *password*, hingga *email*. NIK dan *password* berguna dalam proses *login*.

8. Proses Pencatatan Data Onsite RS

Proses pencatatan data onsite RS berfungsi untuk menyimpan data onsite yang ada di rumah sakit. Dalam menginputkan data onsite rs harus sesuai dengan siapa *onsite* yang ada pada rumah sakit tersebut.

9. Proses Pencatatan Status List Complain

Proses pencatatan status list complain berfungsi untuk menyimpan status list complain yaitu *pending*, diterima dan ditolak. Status list complain digunakan dalam proses transaksi pemilihan complain oleh koordinator onsite.

10. Proses Pencatatan Kategori Complain

Proses pencatatan kategori complain berfungsi untuk menyimpan kategori complain, yaitu *easy*, *medium*, *hard*, *very hard*. Pada setiap kategori terdapat waktu estimasi dan *deadline* yang berfungsi menampilkan peringatan pada jadwal jika complain yang jadwalkan belum terselesaikan hingga mendekati *deadline*.

11. Proses Pencatatan Status Penjadwalan

Proses pencatatan status penjadwalan berfungsi untuk menyimpan status penjadwalan, yaitu *pending*, *queue*, *progress*, *done*. Status ini yang akan di tampilkan dalam jadwal penanganan complain.

12. Proses Pencatatan Data *Complain* Rumah Sakit

Proses pencatatan data *complain* rumah sakit dilakukan oleh bagian onsite. Proses ini dilakukan saat terjadi error pada modul yang ada pada rumah sakit tempat onsite ditempatkan. Onsite akan membuka form pada aplikasi web dan menginputkan modul apa yang tersjadi *error* dan juga keterangan error yang terjadi.

13. Proses Pemilihan *Complain*

Proses pemilihan *complain* dilakukan oleh koordinator sofdev. Proses ini terjadi saat *complain* pada satu hari kerja dikirimkan oleh beberapa onsite yang mengalami *error* modul pada rumah sakit. *Complain* akan

diterima oleh koordinator *onsite*, koordinator *onsite* akan mengecek keterangan *complain* jika menurut koordinator *onsite* berdasarkan keterangan *complain* layak untuk dilanjutkan dan ditangani oleh *sofdev*, maka koordinator *onsite* akan menerima *complain* tersebut, jika tidak layak akan ditolak.

14. Proses Penambahan Kategori Complain

Proses penambahan kategori *complain* dilakukan oleh koordinator *sofdev*. Setelah di terima oleh koordinator *onsite* maka selanjutnya *complain* akan diserahkan kepada koordinator *sofdev*. Koordinator *sofdev* menerima *complain* tersebut dan selanjutnya akan diberukan kategori pada setiap *complain* berdasarkan keterangan *complain* yang ada.

15. Proses *Sorting* Berdasarkan Kriteria

Setelah proses pemberian kategori pada setiap *complain* oleh koordinator *sofdev* maka *complain* akan di *sorting* bedasarkan kriteria yang telah di tentukan yaitu tanggal masuk *complain*, nilai kontrak rumah sakit, kategori modul dan status rumah sakit.

16. Update Status Penjadwalan (*Sofdev*)

Setelah *complain* tersorting berdasarkan kriteria yang di tentukan, maka akan ditampilkan ke bagian *sofdev*. List *complain* yang di tampilkan sudah tersorting menurut kriteria. *Sofdev* memilih list *complain* yang paling atas dengan menekan tombol pilih.

17. Update Status Penjadwalan (Koordinator *Sofdev*)

Setelah *complain* benear-benar selesai dikerjakan dan di cek oleh koordinator *sofdev*, maka koordinator *sofdev* akan menekan tombol centang atau selesai yang akan mengupdate status dari *progress* ke *done*. Jika sudah menekan tombol centang maka *complain* dengan status *done* akan hilang dari jadwal penanganan *complain*.

C. Output

1. List Complain

List complain adalah *output* yang dihasilkan dari proses pencatatan data *complain* oleh onsite yang mengalami modul *error* di setiap rumah sakit.

2. List Complain Yang Layak Di Tangani Oleh Sofdev

List Complain Yang Layak Di Tangani Oleh *Sofdev* adalah output yang dihasilkan dari proses pemilihan *complain* yang dilakukan oleh koordinator *onsite*.

3. Complain Terjadwal Dengan Status Queue

Complain terjadwal dengan status *Queue* adalah *output* yang dihasilkan dari proses sorting dari empat kategori yaitu tanggal *complain*, nilai kontrak, kategori modul, status RS. Karena *complain* belum *approve* dan dikerjakan oleh *sofdev* maka status awal adalah *Queue*.

4. Complain Terjadwal Dengan Status Progress

Complain terjadwal dengan status *progress* adalah *output* yang dihasilkan setelah pihak *sofdev* memilih *complain*. Setelah proses tersebut maka pada jadwal list *complain* yang sudah di pilih oleh pihak *sofdev* akan berubah status nya menjadi *progress*.

5. Complain Selesai

Saat *complain* selesai maka *complain* terlebih dulu di cek oleh koordinator *sofdev*, ketika benar-benar selesai maka koordinator *sofdev* akan menekan tombol centang atau selesai dan mengupdate status dari *progress* menjadi *done*. *Complain* dengan status *done* akan hilang dari jadwal penanganan *complain*.

6. Report

Report yang dihasilkan dari aplikasi penjadwalan penanganan *complain* ini adalah *report complain* per *sofdev*, *report* RS berdasarkan

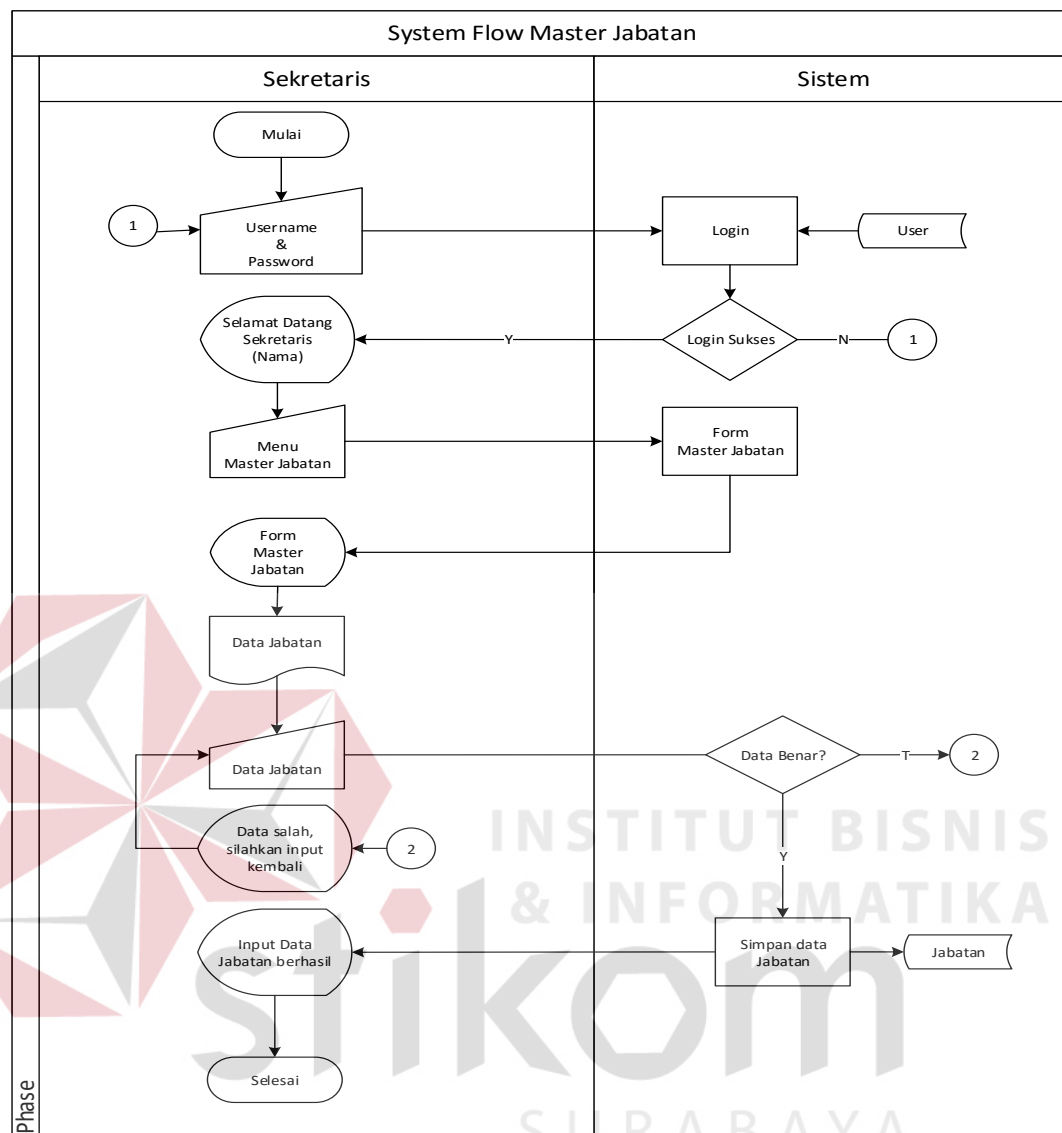
masa kontrak pertahun, *report* RS berdasarkan jumlah *complain*, *report* modul yang sering terdapat *error*.

3.5. Desain Sistem

Desain sistem dibuat setelah mengetahui kebutuhan sistem yang ada pada PT. NCI cabang Surabaya melalui hasil analisa sistem yang sudah dilakukan. Desain sistem dimulai dengan membuat *system flow* dengan berdasar pada analisa serta desain sistem yang akan dibuat, kemudian dilanjutkan dengan membuat *context diagram* dari program yang akan dibuat. *Context diagram* tersebut akan di *decompose* untuk proses selanjutnya yaitu membuat Data Flow Diagram (DFD). Dalam DFD digambarkan alur data dari user hingga tersimpa pada table. Selanjutnya akan dilakukan perancangan *Database* sesuai dengan kebutuhan yang ada pada sistem. Pada perancangan *database* akan dilakukan dengan menggunakan Conceptual Data Model (CDM) yang kemudian digenerate ke dalam Physical Data Model (PDM).

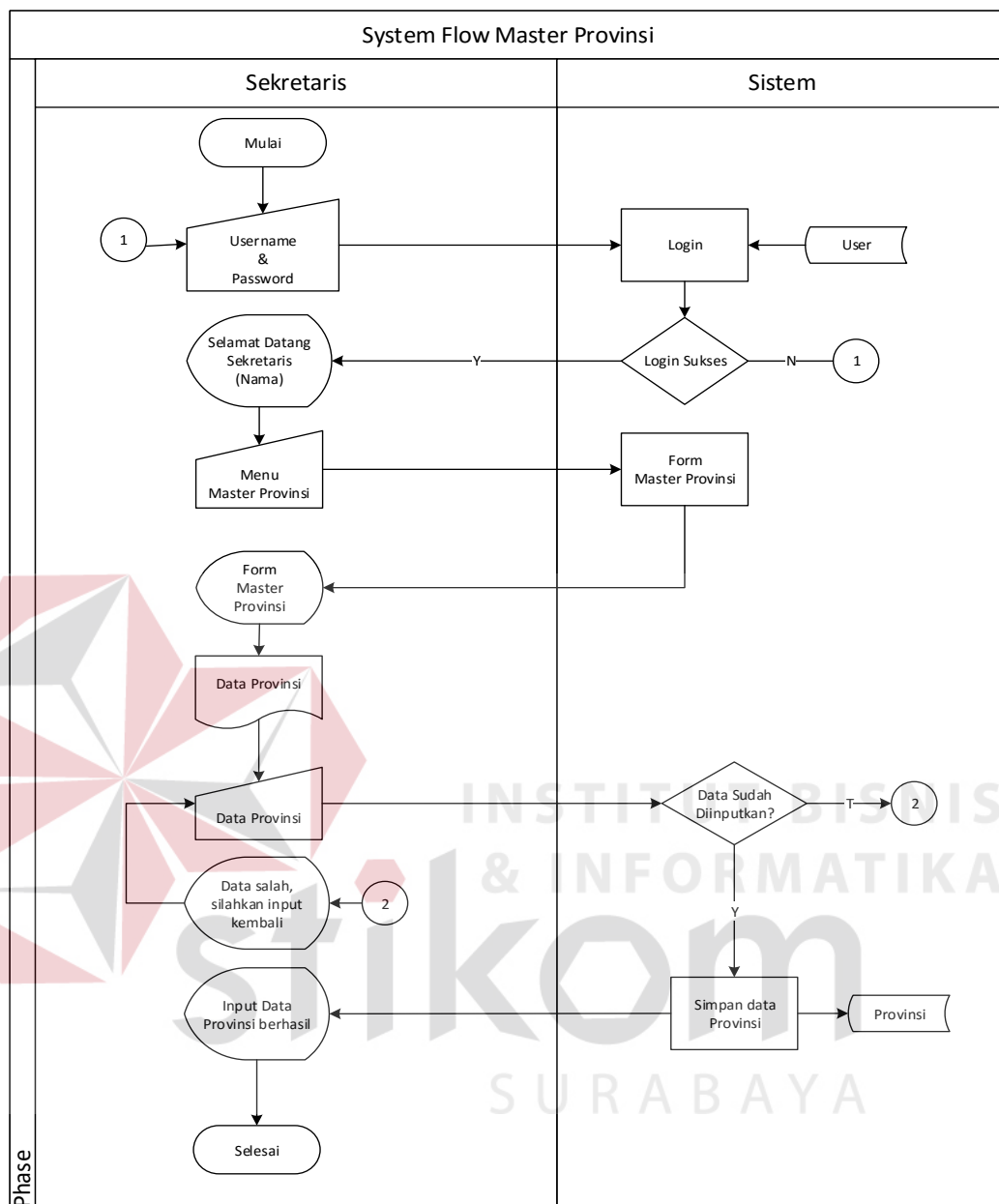
3.5.1. System Flow

System flow adalah desain sistem yang digambarkan dalam bentuk alur sistem dimana didalamnya arah dan tujuan dari suatu sistem secara keseluruhan. Alur sistem pada aplikasi penjadwalan penanganan *complain* berbasis web adalah sebagai berikut :



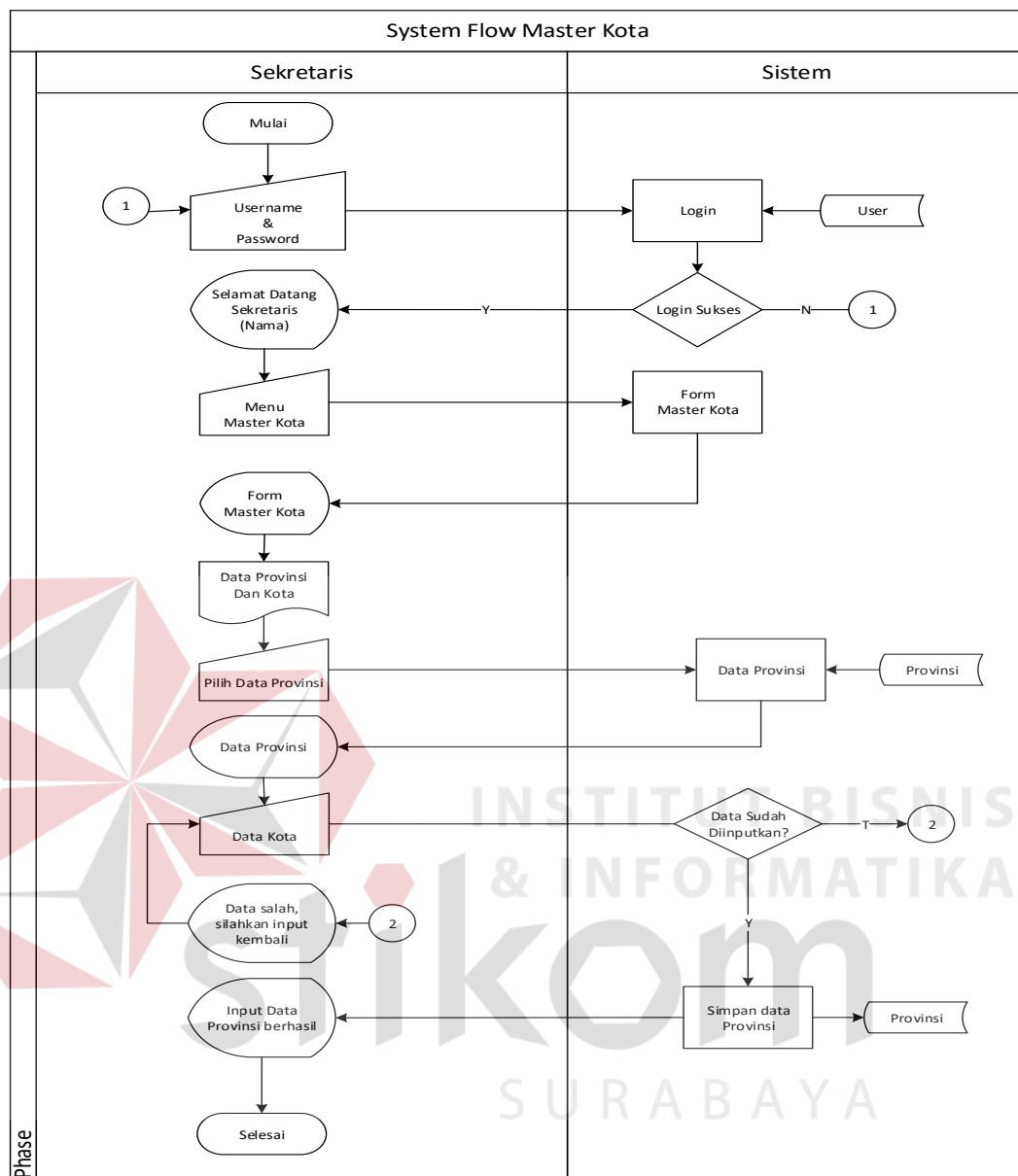
Gambar 3.4 *System Flow Master Jabatan*

Pada gambar 3.4 menjelaskan tentang proses input data pada master jabatan yang dilakukan oleh sekretaris. Dimulai dari proses login pada website dengan memasukkan *username* dan *password* pada website kemudian sekretaris memilih menu master jabatan. Setelah menu tampil maka sekretaris akan memasukkan data jabatan sesuai dengan modul yang sesuai pada PT. NCI dan menyimpan data tersebut kedalam *database*.



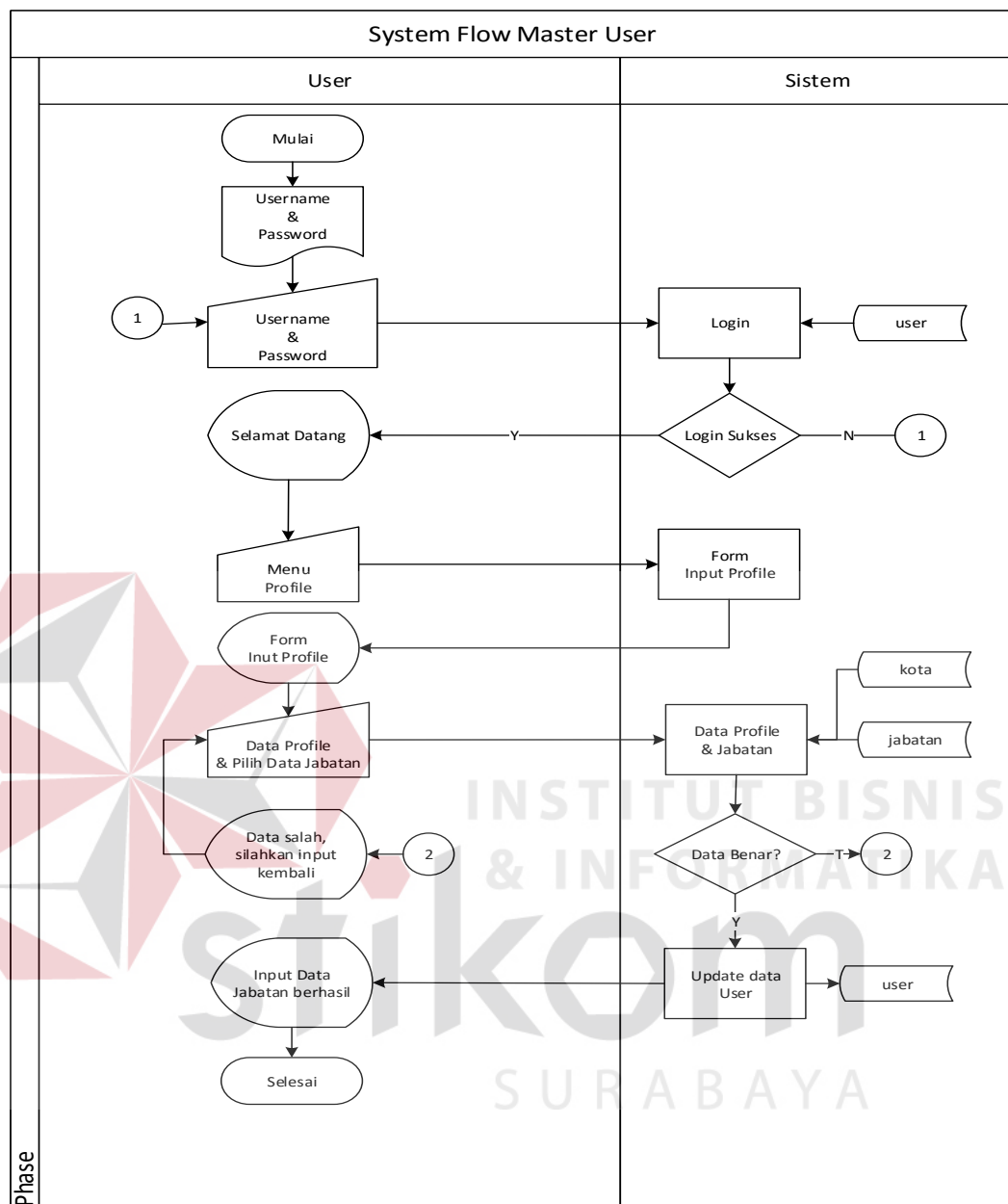
Gambar 3.5 *System Flow* Master Provinsi

Pada gambar 3.5 menjelaskan tentang proses input data pada master provinsi yang dilakukan oleh sekretaris. Dimulai dari proses login pada website dengan memasukkan *username* dan *password* pada website kemudian sekretaris memilih menu master provinsi. Setelah menu tampil maka sekretaris akan memasukkan data provinsi sesuai dengan provinsi yang ada di Indonesia dan menyimpan data tersebut kedalam *database*.



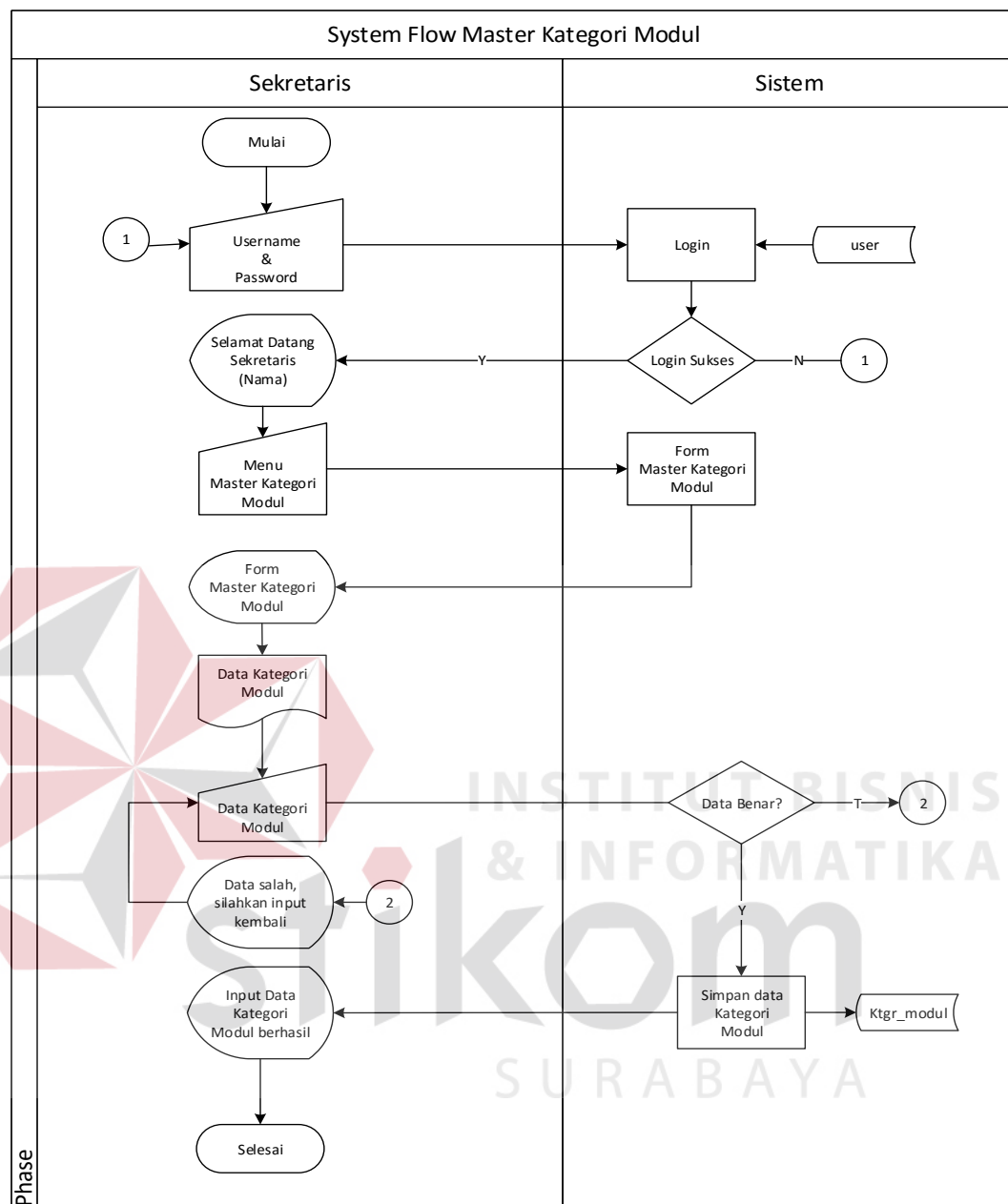
Gambar 3.6 *System Flow* Master Kota

Pada gambar 3.6 menjelaskan tentang proses input data pada master kota yang dilakukan oleh sekretaris. Dimulai dari proses login pada website dengan memasukkan *username* dan *password* pada website kemudian sekretaris memilih menu master kota. Setelah menu tampil maka sekretaris akan memasukkan data kota sesuai dengan data provinsinya dan menyimpan data tersebut kedalam *database*.

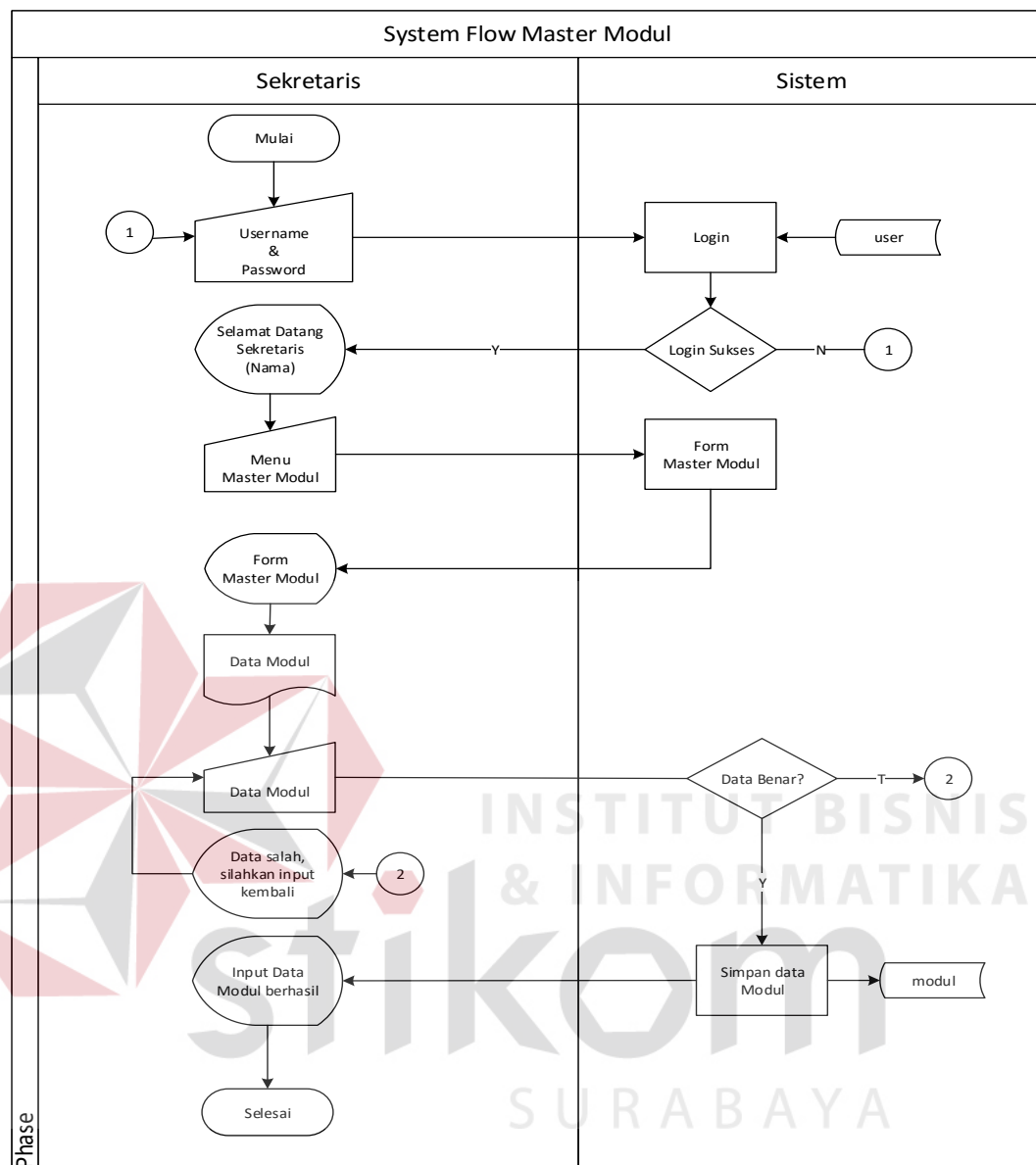


Gambar 3.7 *System Flow Master User*

Pada gambar 3.7 menjelaskan tentang proses input data pada master user yang dilakukan oleh seluruh user. Dimulai dari proses login pada website dengan memasukkan *username* dan *password* pada website kemudian user diharuskan untuk melengkapi data user pada profile masing-masing dengan menyertakan jabatan masing-masing. Setelah selesai maka data tersebut akan disimpan kedalam *database*.

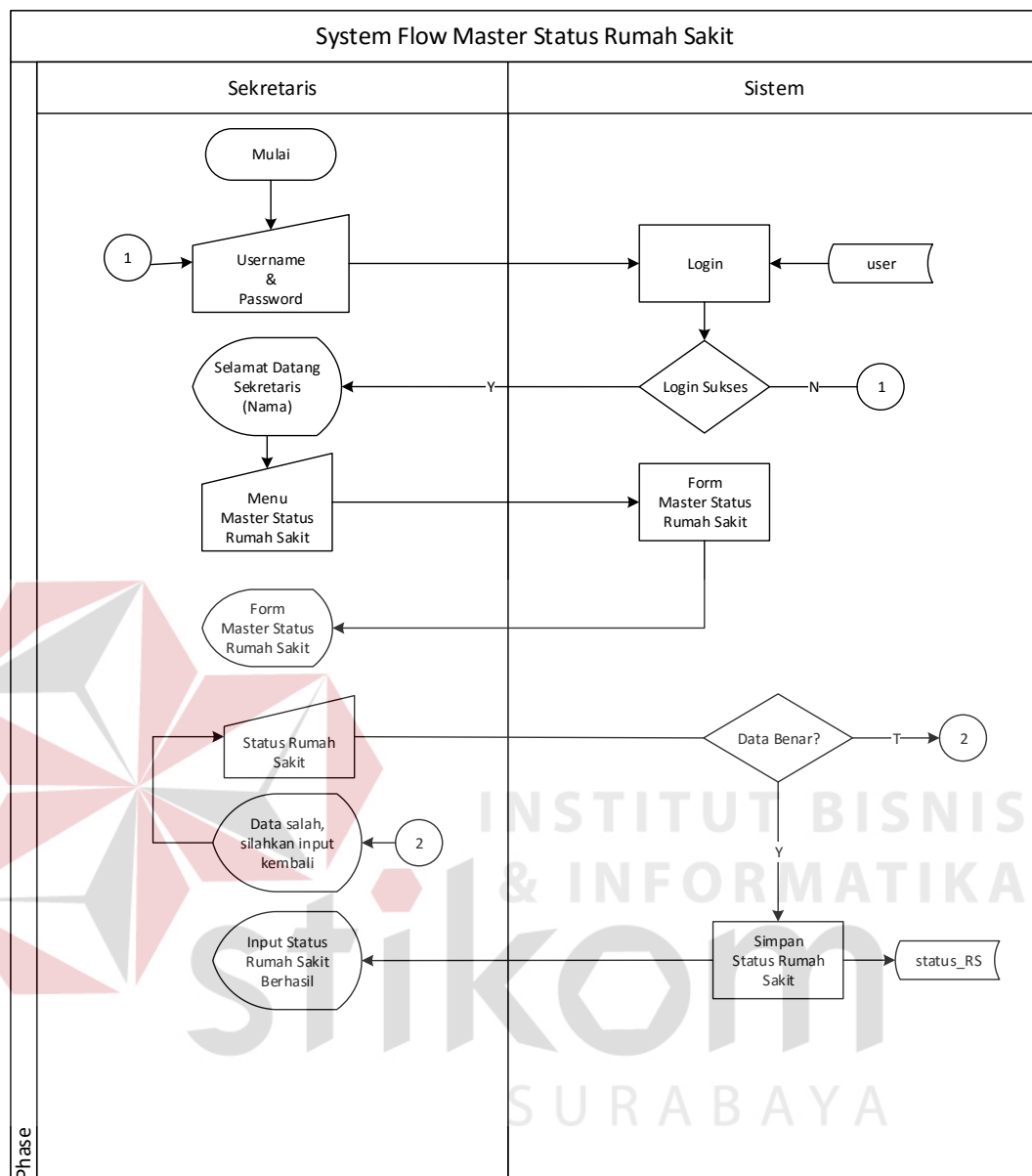


Gambar 3.8 *System Flow* Master Kategori Modul



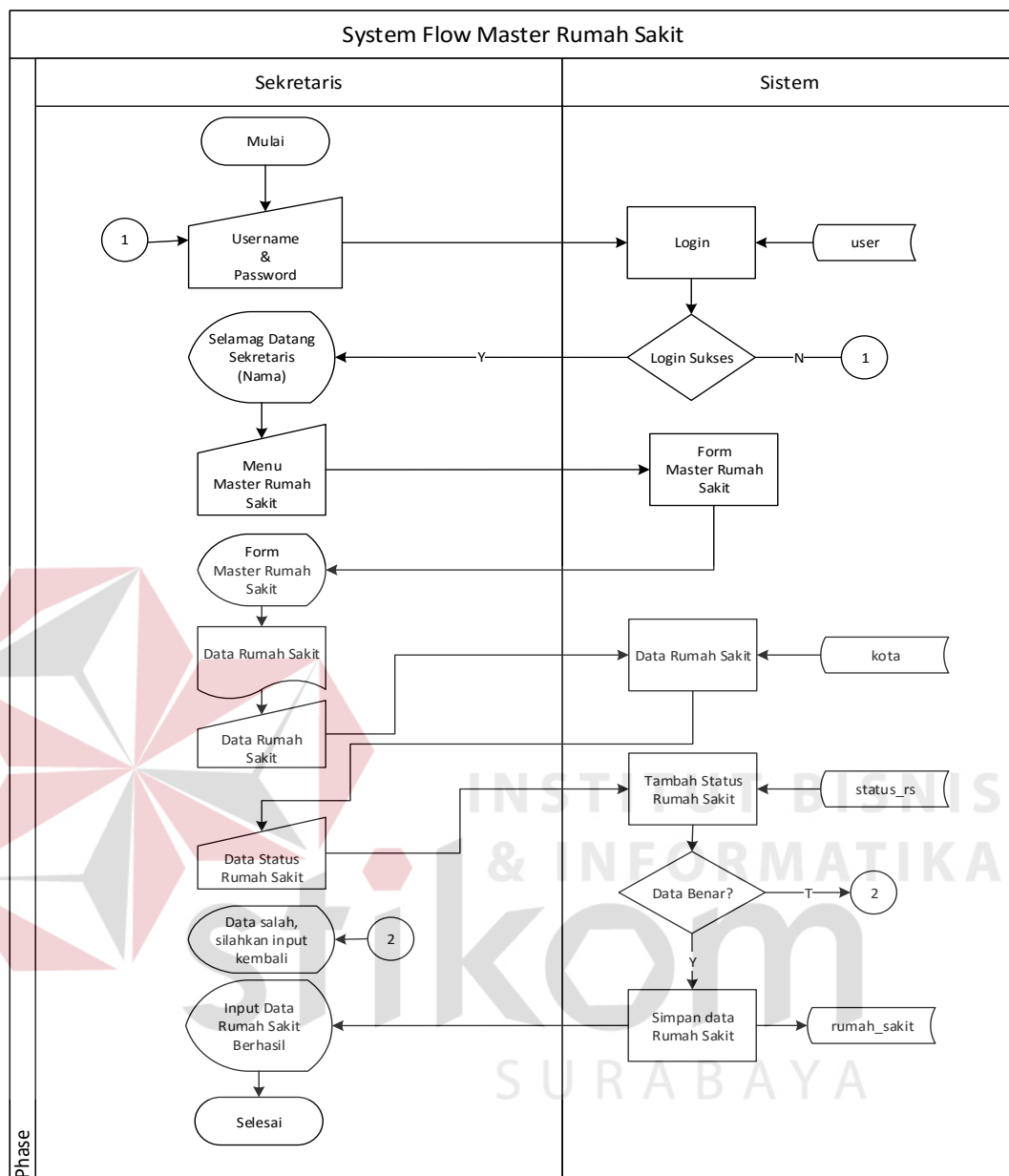
Gambar 3.9 *System Flow* Master Modul

Pada gambar 3.9 menjelaskan tentang proses input data pada master modul yang dilakukan oleh sekretaris. Dimulai dari proses login pada website dengan memasukkan *username* dan *password* pada website kemudian sekretaris memilih menu master modul. Setelah menu tampil maka sekretaris akan memasukkan data modul sesuai dengan modul yang ada SIM RS Medismart dan menyimpan data tersebut kedalam *database*.



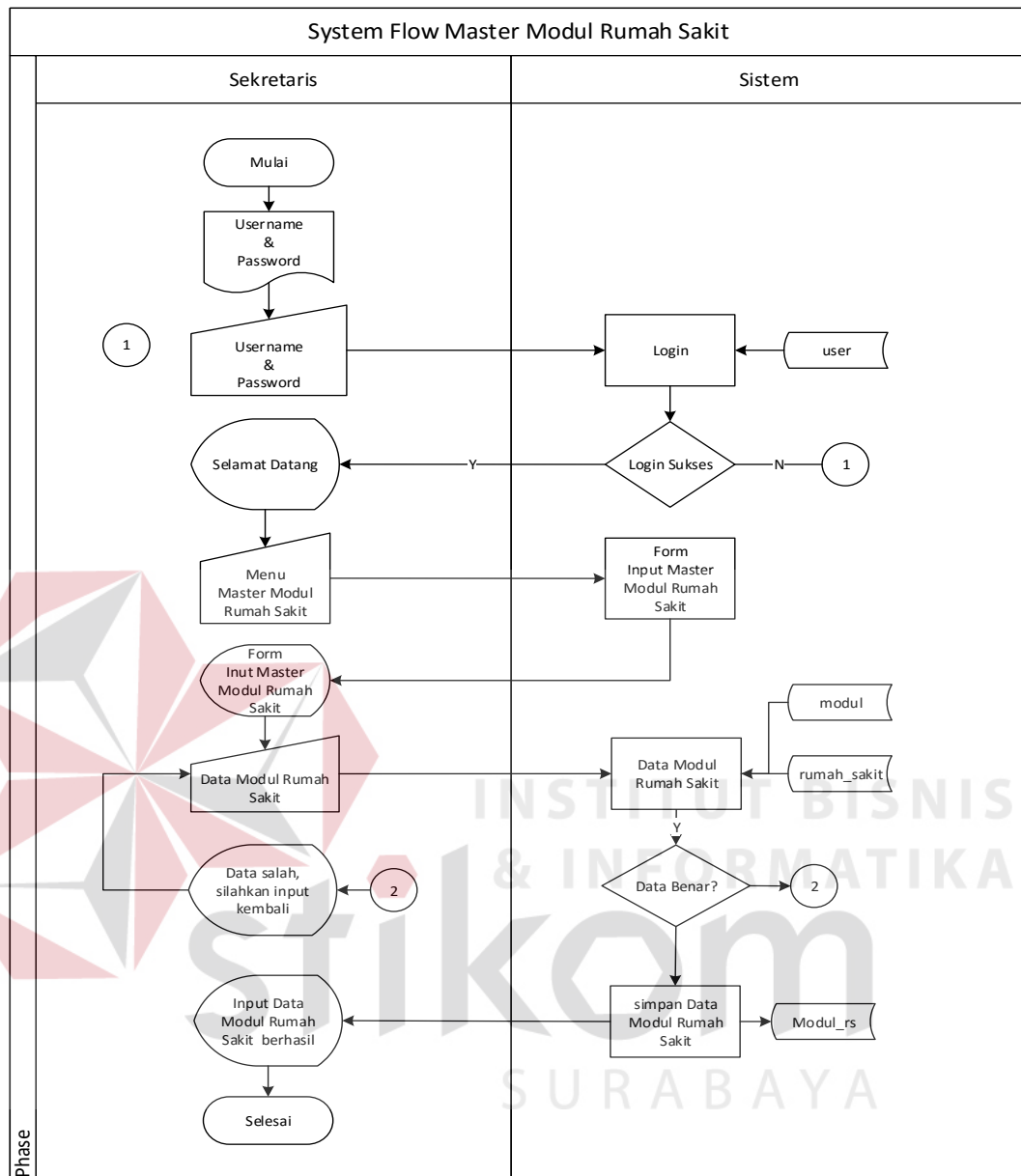
Gambar 3.10 System Flow Master Status Rumah Sakit

Pada gambar 3.10 proses input data pada master status rumah sakit yang dilakukan oleh sekretaris. Dimulai dari proses login pada website dengan memasukan *username* dan *password* pada website kemudian sekretaris memilih menu master status rumah sakit. Setelah menu status rumah sakit tampil maka sekretaris akan menginputkan status rumah sakit sesuai dengan data yang ada.

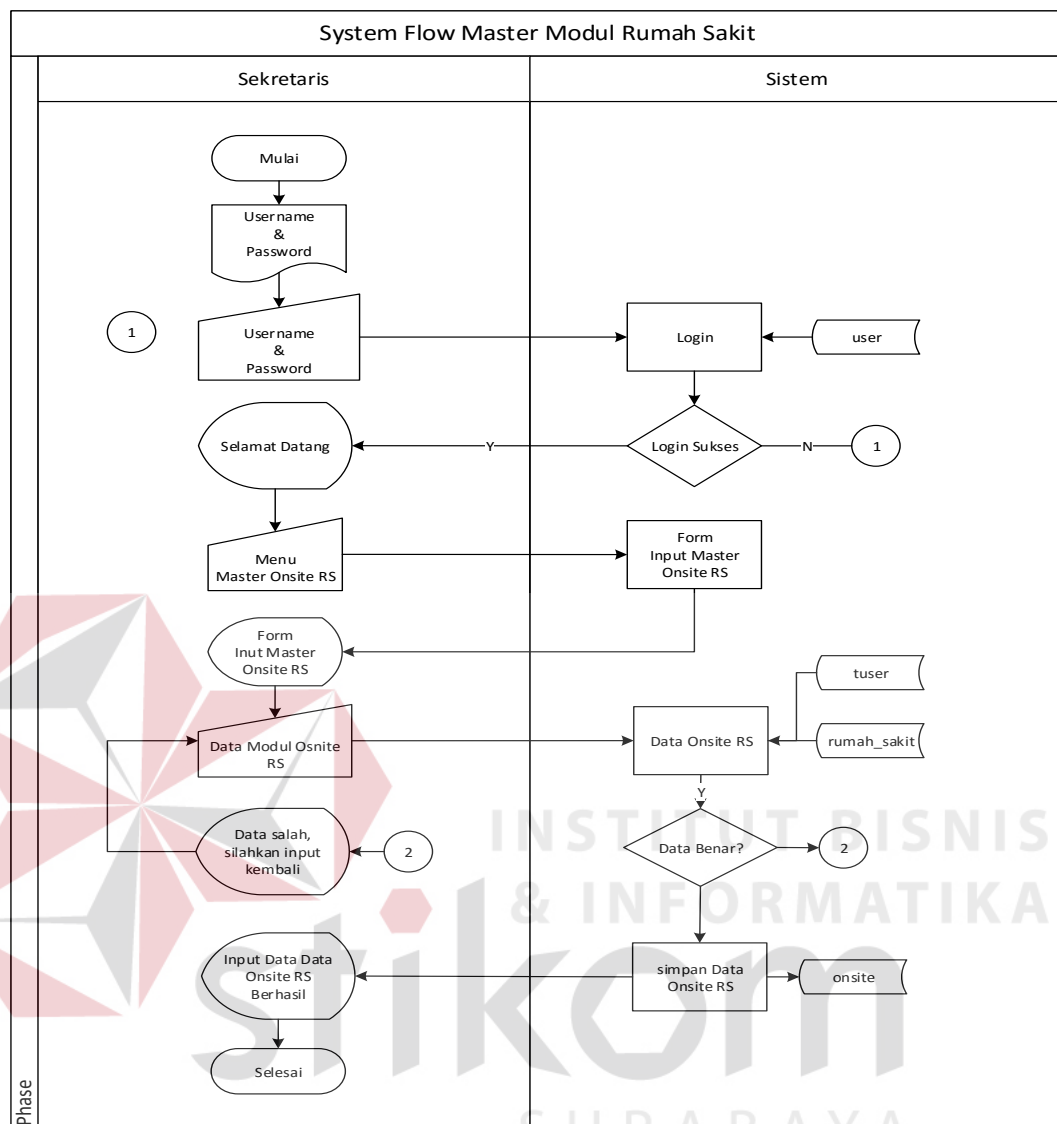


Gambar 3.11 *System Flow* Master Rumah Sakit

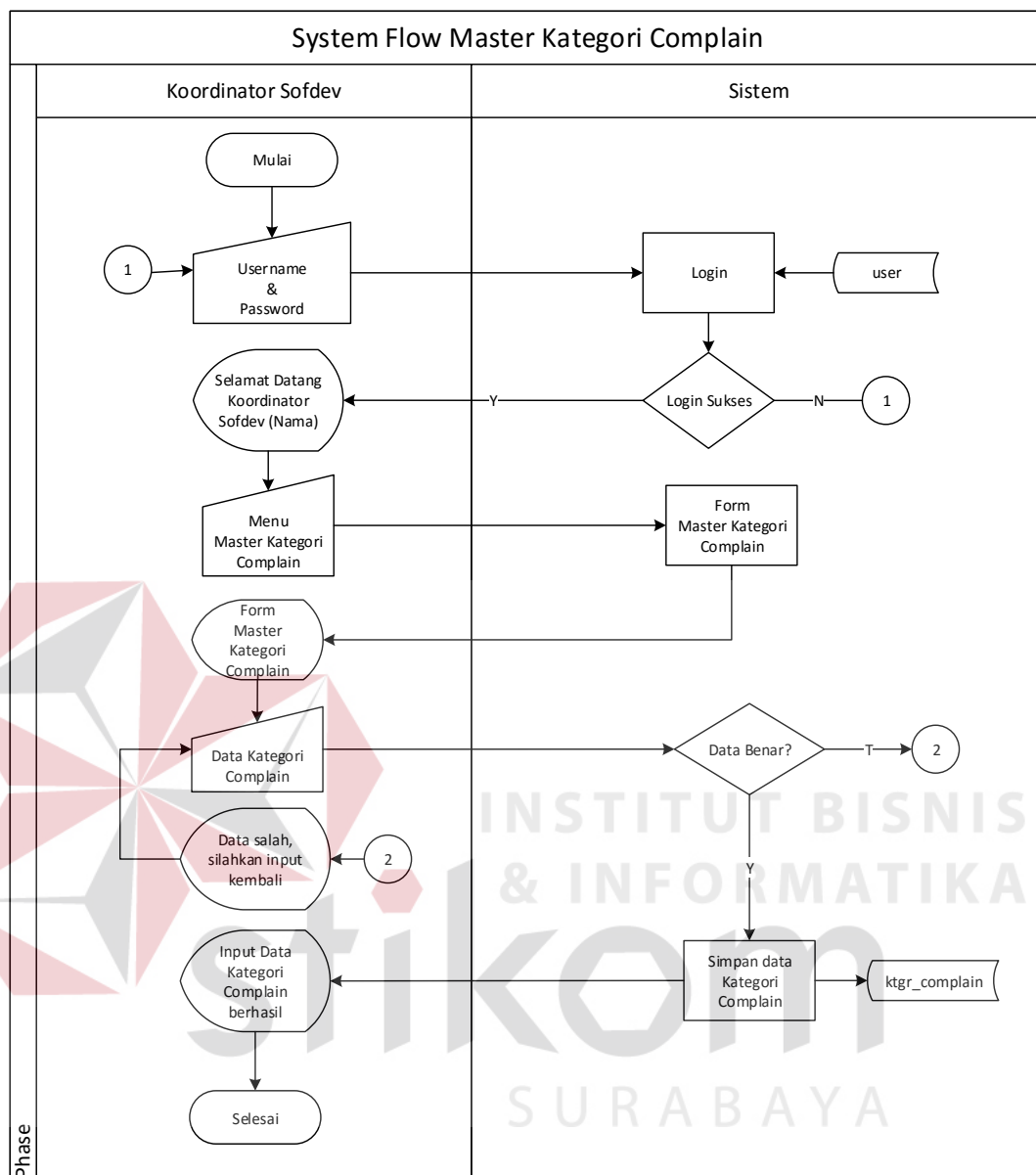
Pada gambar 3.11 menjelaskan tentang proses input data pada master rumah sakit yang dilakukan oleh sekretaris. Dimulai dari proses login pada website dengan memasukkan *username* dan *password* pada website kemudian sekretaris memilih menu master rumah sakit. Setelah menu master rumah sakit tampil, sekretaris akan menginputkan data rumah sakit dan status rumah sakit sesuai dengan data rumah sakit yang menjadi customer dari PT. NCI cabang Surabaya dan menyimpannya kedalam *database*.



Gambar 3.12 *System Flow* Master Modul Rumah Sakit

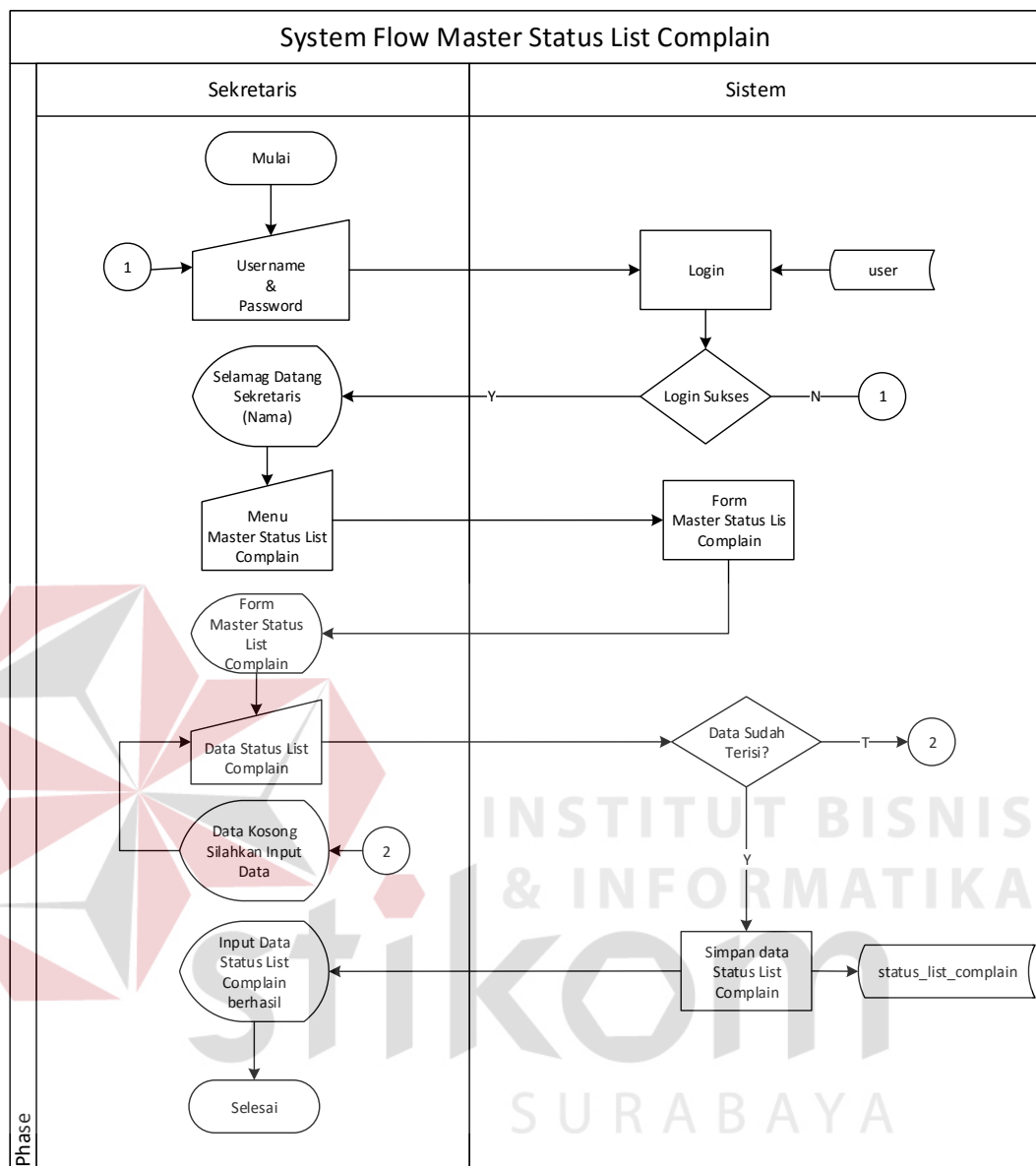


Gambar 3.13 *System Flow* Master Onsite RS



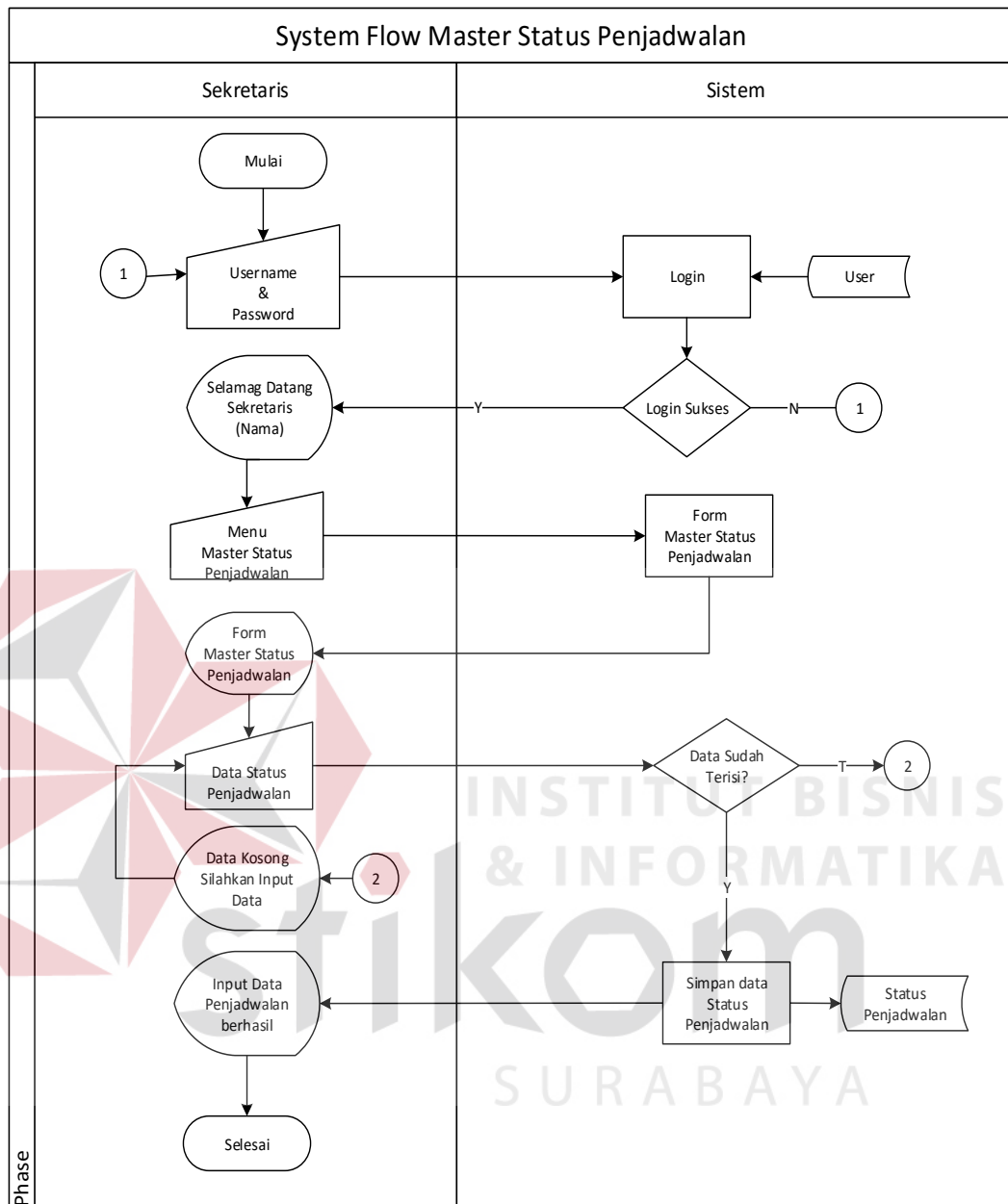
Gambar 3.14 *System Flow* Master Kategori *Complain*

Pada gambar 3.14 menjelaskan tentang input data pada master kategori *complain* yang dilakukan oleh koordinator *sofdev*. Dimulai dari proses login pada website dengan memasukan *username* dan *password* pada website kemudian koordinator *sofdev* memilih menu master kategori *complain*. Setelah menu master kategori *complain* tampil, koordinator *sofdev* akan menginputkan data kategori *complain* yaitu kategori mudah, sedang, sulit, sangat sulit (*easy, medium, hard, very hard*) dan menyimpannya kedalam database.



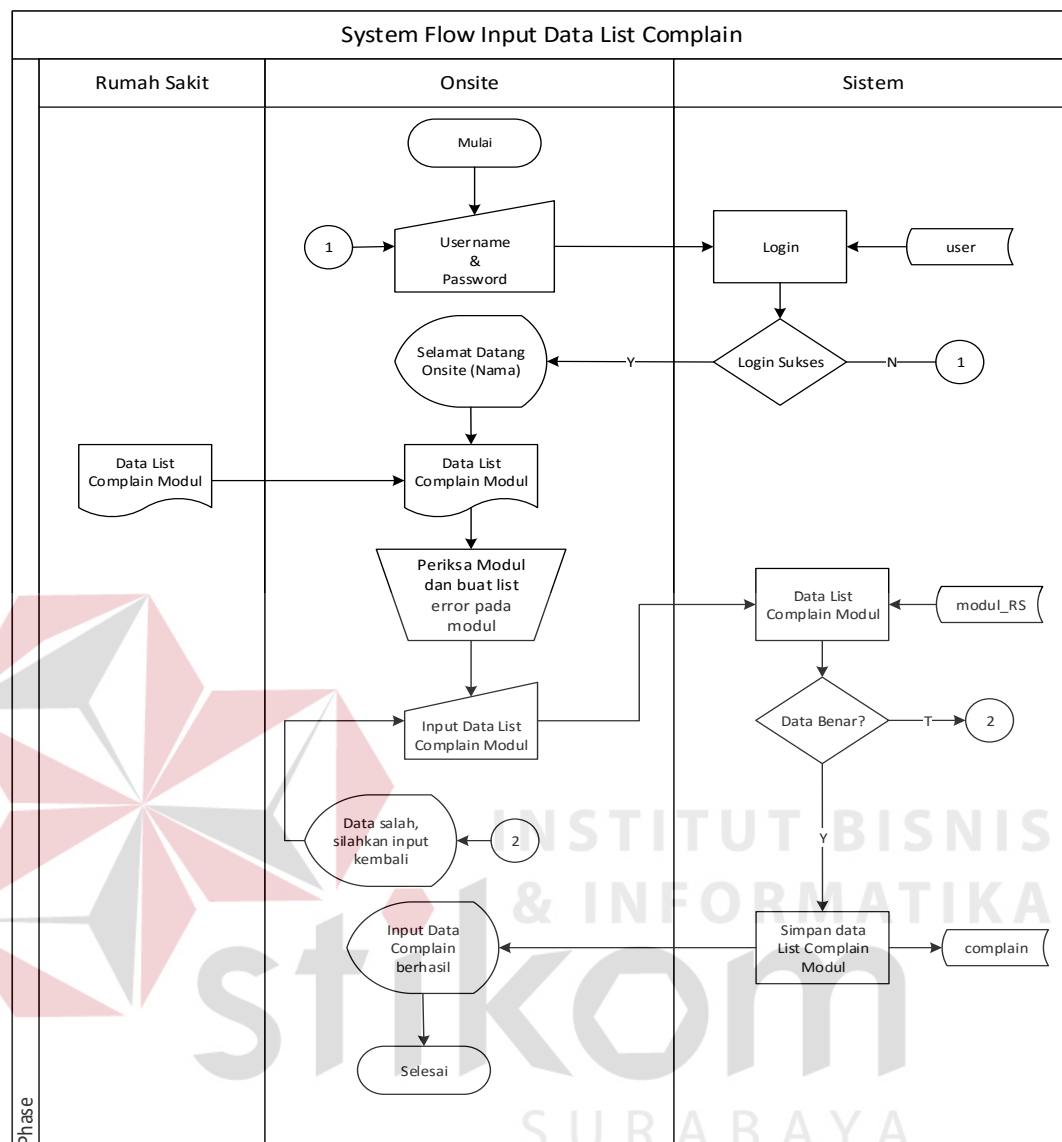
Gambar 3.15 *System Flow Master Status List Complain*

Pada gambar 3.15 menjelaskan tentang proses input data pada master status rumah sakit yang dilakukan oleh sekretaris. Dimulai dari proses login pada website dengan memasukkan *username* dan *password* pada website kemudian sekretaris memilih menu master status list complain. Setelah menu master status list complain tampil maka sekretaris akan menginputkan status rumah sakit sesuai dengan data yang ada.



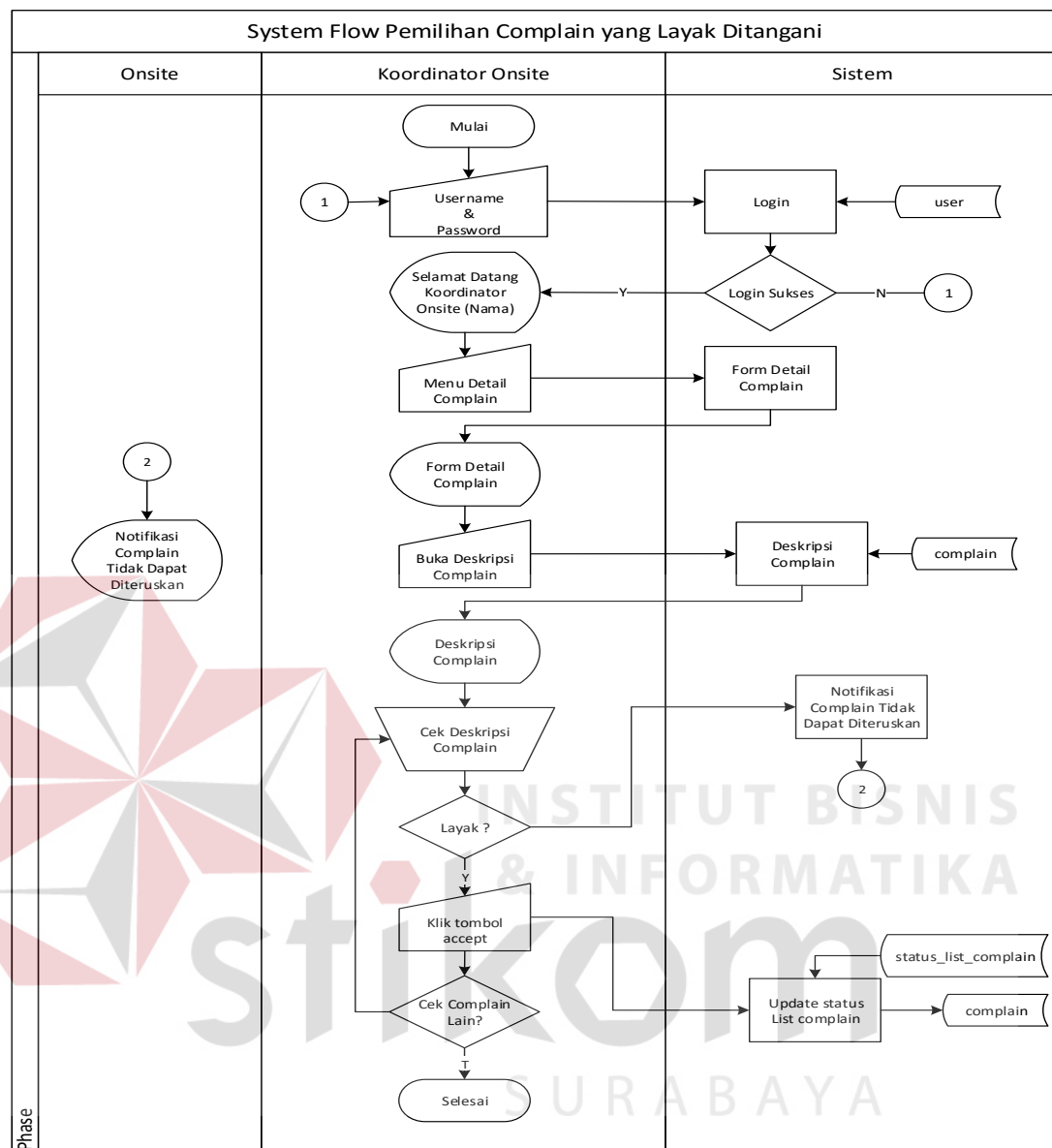
Gambar 3.16 *System Flow* Master Status Penjadwalan

Pada gambar 3.16 menjelaskan proses input data pada master status rumah sakit yang dilakukan oleh sekretaris. Dimulai dari proses login pada website dengan memasukkan *username* dan *password* pada website kemudian sekretaris memilih menu master status penjadwalan. Setelah menu master status penjadwalan tampil maka sekretaris akan menginputkan status rumah sakit sesuai dengan data yang ada yaitu *pending*, *on progress* dan *done*.



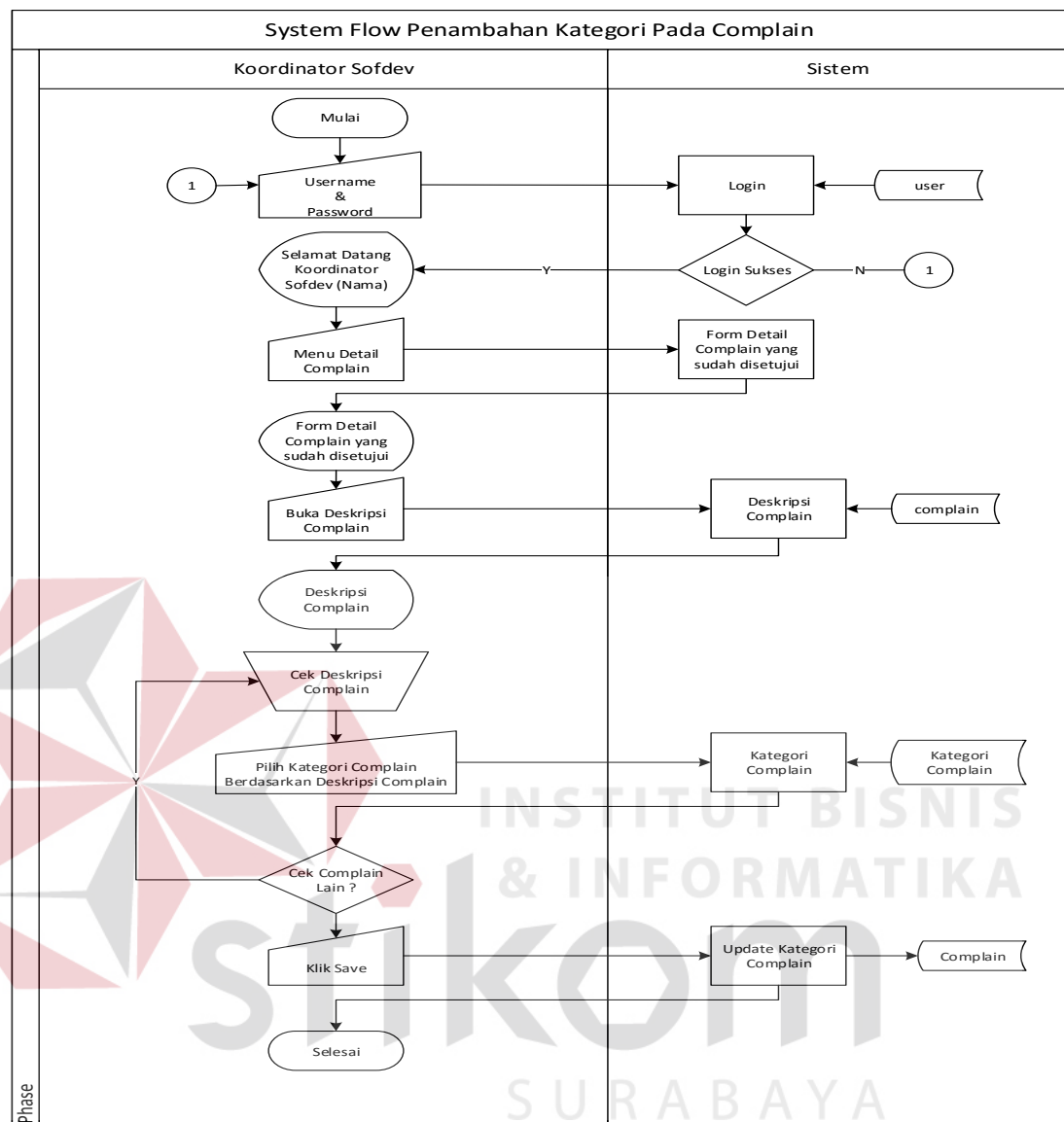
Gambar 3.17 *System Flow Input Data List Complain*

Pada gambar 3.17 input data list *complain* yang dilakukan oleh *oniste*. Dimulai dari *onsite* menerima list complain dari rumah sakit. *Onsite* melakukan proses login pada website dengan memasukkan *username* dan *password* setelah berhasil login, *onsite* memilih menu input data list complain, *onsite* akan menginputkan data sesuai dengan error yang ada pada modul tersebut. Ketika data yang di inputkan tidak sesuai dengan persyaratan sistem maka akan muncul peringatan untuk menginputkan kembali data yang salah tersebut. Ketika data yang di inputkan benar maka data tersebut akan tersimpan dalam database.



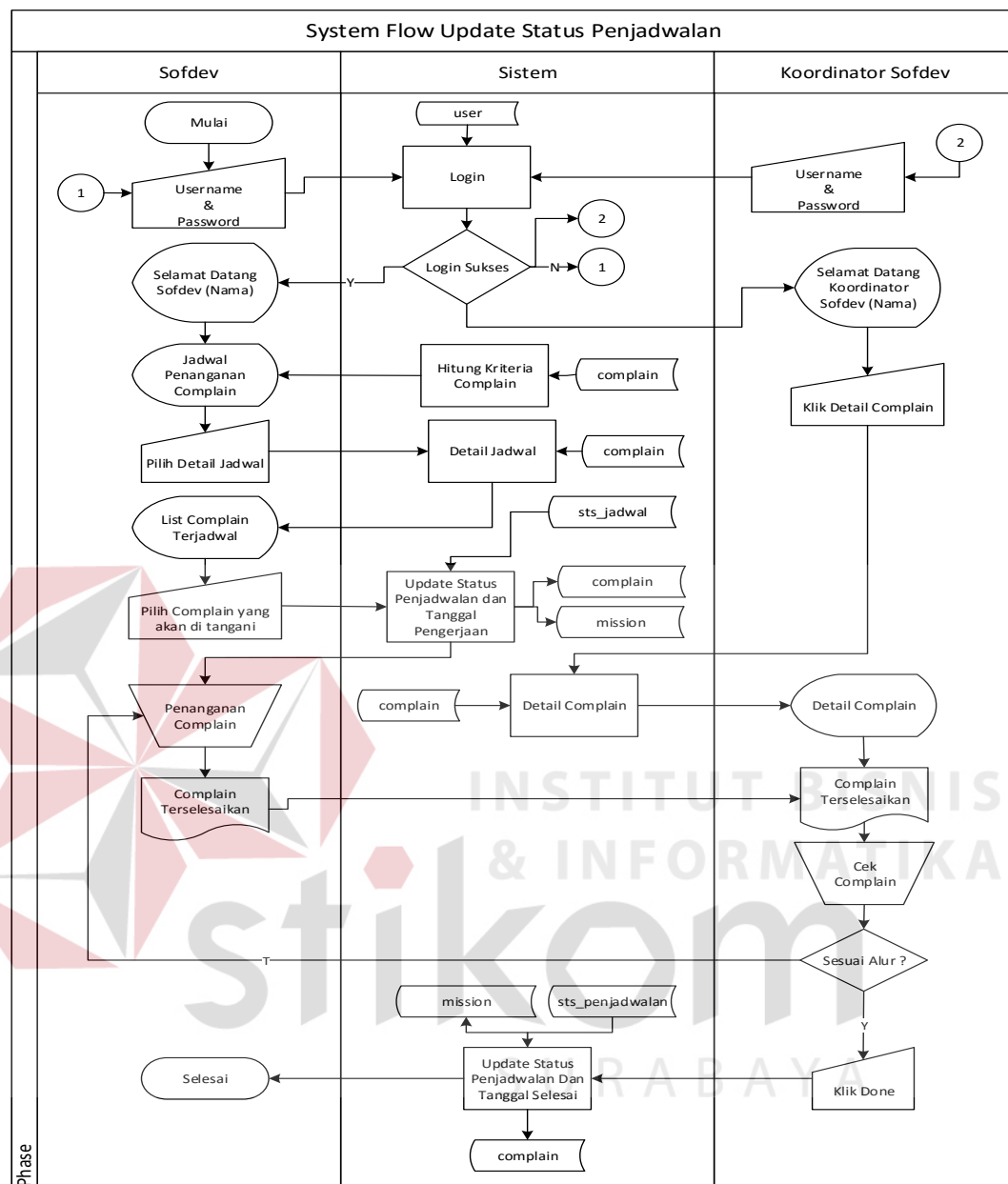
Gambar 3.18 *System Flow* Pemilihan *Complain* yang Layak Ditangani

Pada gambar 3.18 pemilihan *complain* yang layak ditangani yang dilakukan oleh koordinator *onsite*. Dimulai dari koordinator *onsite* menerima detail *complain* dari *onsite*, kemudian akan dicek deskripsi *complain* pada masing-masing *complain* yang masuk. Jika memang *complain* layak untuk di lanjutkan untuk dikerjakan oleh *sofdev* maka koordinator *onsite* akan menekan tombol *accept* dan akan mengupdate status pada tabel *complain*. Jika *complain* memang tidak layak diteruskan maka hanya dikerjakan oleh *onsite* saja maka koordinator *onsite* akan mengirimkan notifikasi ke *onsite*.



Gambar 3.19 *System Flow* Penambahan Kategori Pada *Complain*

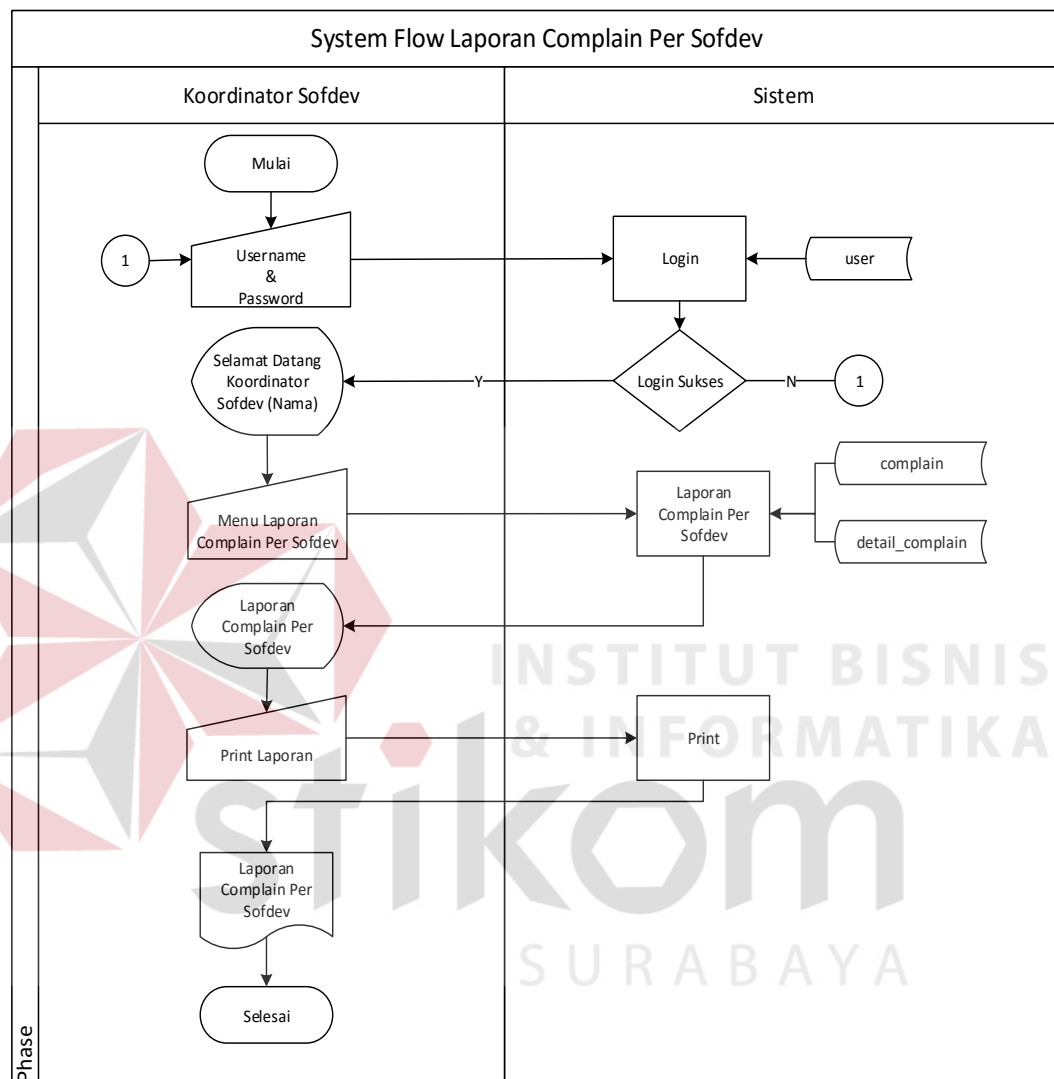
Pada gambar 3.19 menjelaskan tentang penambahan kategori pada complain yang dilakukan oleh koordinator *sofdev*. Dimulai dari koordinator *sofdev* menerima data detail complain yang sudah disetujui, kemudian akan di cek deskripsi complain, setelah itu koordinator *sofdev* akan memberikan kategori *complain* pada setiap complain yang masuk berdasarkan kesulitan yang ditulis pada deskripsi complain.



Gambar 3.20 System Flow Update Status Penjadwalan

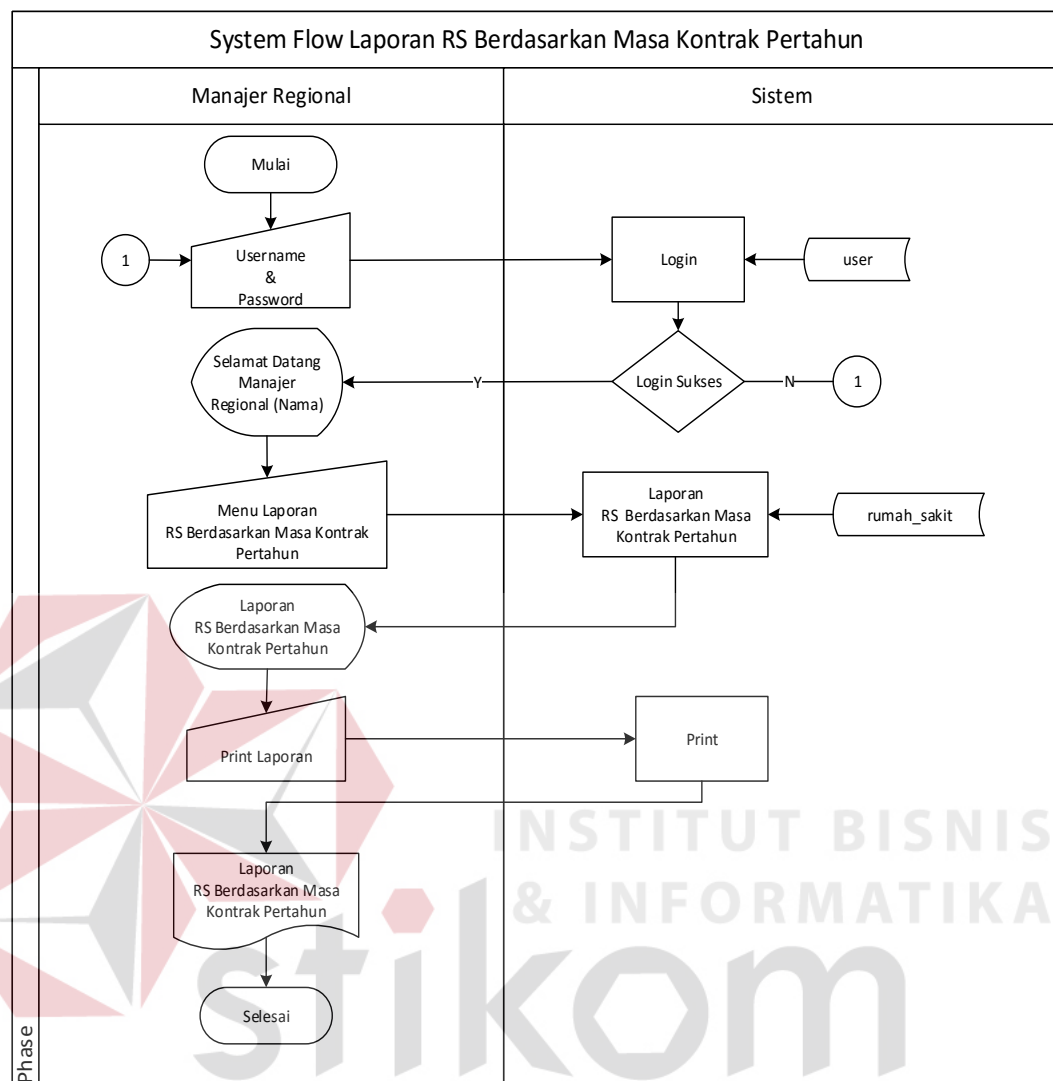
Pada gambar 3.20 menjelaskan tentang update status penjadwalan dilakukan oleh *sofdev* dan *koordinator sofdev*. Dimulai dari *sofdev* menerima jadwal penanganan complain kemudian masuk ke detail jadwal, kemudian memilih complain yang paling atas semuai urutan, setelah memilih complain yang akan ditangani maka otomatis *sofdev* akan mengupdate status jadwal menjadi *on progress* dan mengupdate tanggal pengerjaan pada tabel *complain* dan detail *complain*. Setelah complain dikerjakan maka *koordinator sofdev* akan mengecek

complain jika complain sudah sesuai alur maka koordinator *sofdev* akan mengupdate status jadwal menjadi *done* dan mengupdate tanggal selesai pada tabel *complain* dan detail *complain*



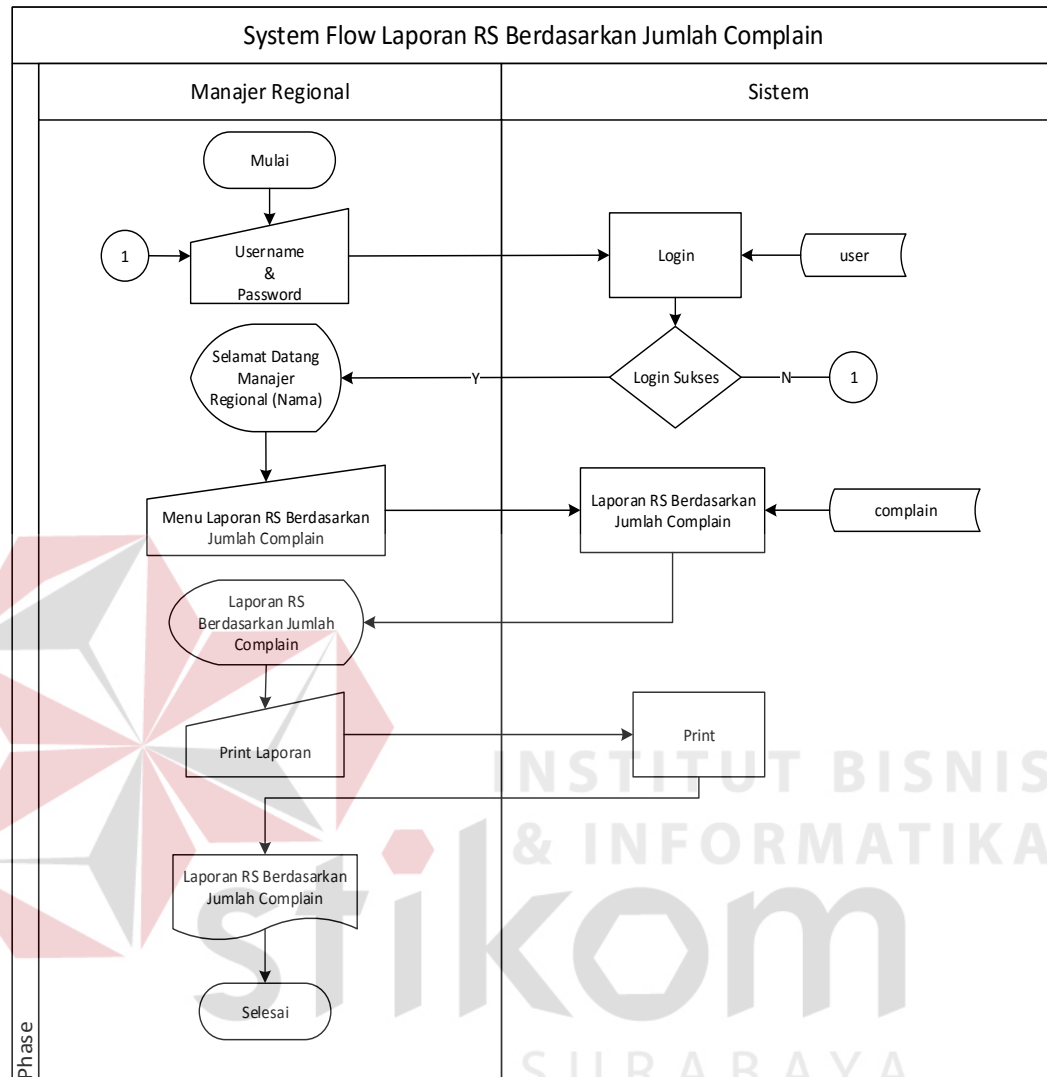
Gambar 3.21 *System Flow* Laporan Complain Per Sofdev

Pada gambar 3.21 menjelaskan tentang laporan complain per *sofdev* yang diterima oleh koordinator *sofdev*. Dimulai dari proses login pada website dengan memasukkan *username* dan *password* pada website kemudian sekretaris memilih menu laporan complain per *sofdev*. Setelah itu koordinator *sofdev* dapat memilih laporan per *sofdev* dengan memilih nama *sofdev*, setelah nama *sofdev* dipilih maka akan tampil data *record complain* apa saja yang di tangani oleh *sofdev* dengan nama terpilih tersebut.



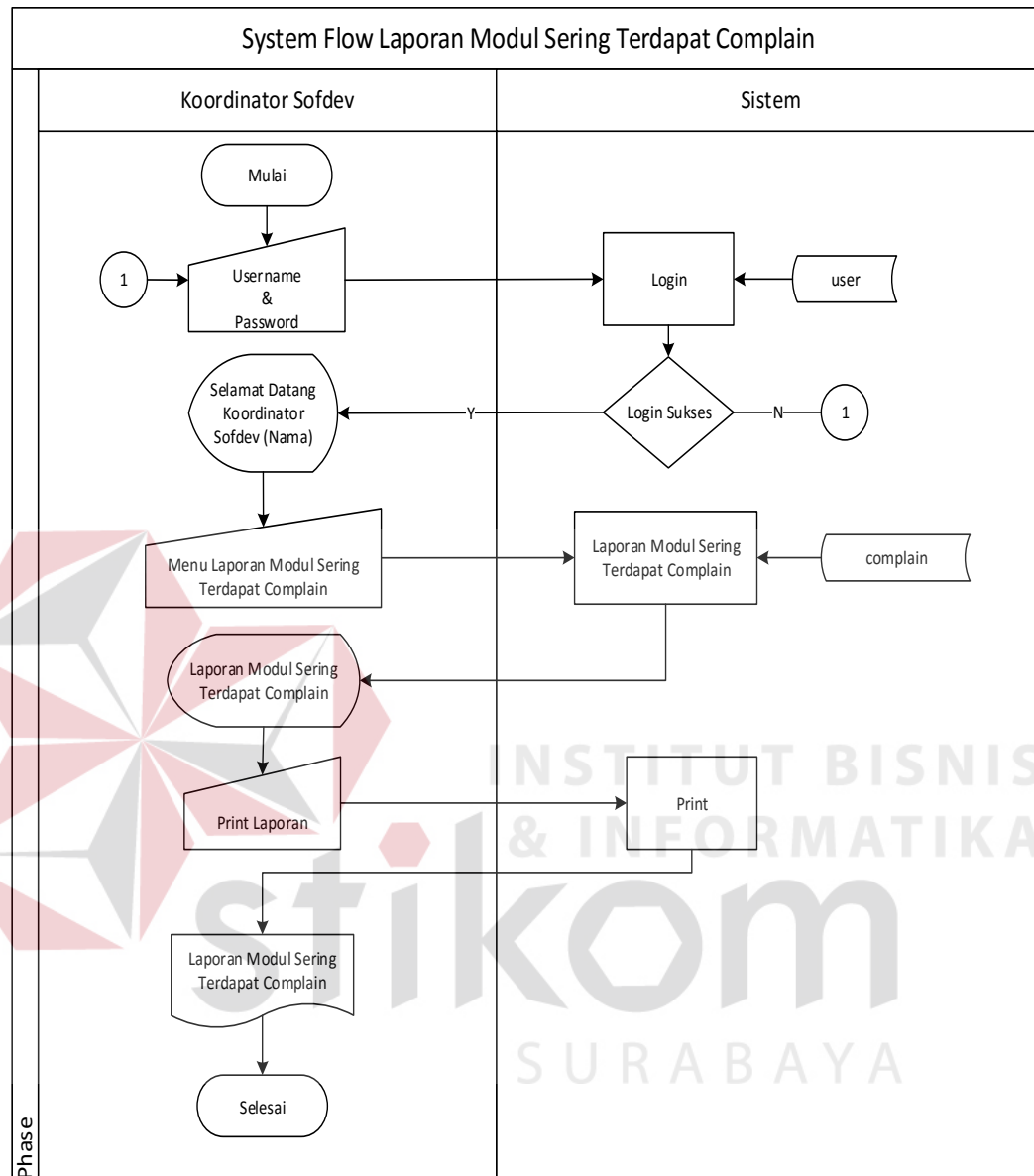
Gambar 3.22 *System Flow Update* Laporan RS Berdasarkan Masa Kontrak
Pertahun

Pada gambar 3.22 menjelaskan Laporan RS berdasarkan masa kontrak pertahun yang diterima oleh manajer regional. Dimulai dari proses login pada website dengan memasukkan *username* dan *password* pada website kemudian sekretaris memilih menu laporan masa kontrak RS. Setelah itu manajer regional dapat memilih laporan rumah sakit kontrak rumah sakit dengan memilih nama rumah sakit, setelah rumah sakit dipilih maka akan tampil data *record* detail data rumah sakit dengan kontrak rumah sakit di urutkan dari kontrak yang akan habis hingga yang memiliki kontrak baru berjalan.



Gambar 3.23 System Flow Laporan RS Berdasarkan Jumlah Complain

Pada gambar 3.23 menjelaskan tentang laporan RS berdasarkan jumlah *complain* yang diterima oleh manajer regional. Dimulai dari proses login pada website dengan memasukkan *username* dan *password* pada website kemudian sekretaris memilih menu laporan RS berdasarkan jumlah *complain*. Setelah itu manajer regional dapat memilih laporan per sovdef dengan memilih nama nama rumah sakit dan bulan, setelah nama rumah sakit dan bulan di pilih maka akan tampil data *record complain* apa saja yang pernah diberikan oleh rumah sakit dan bulan yang dipilih tersebut.

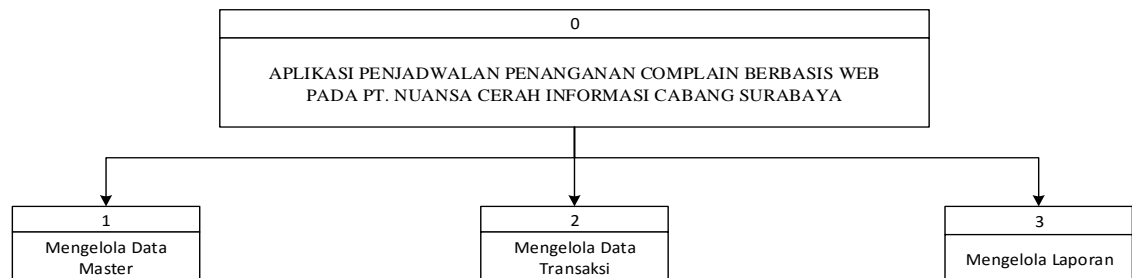


Gambar 3.24 *System Flow Update* Laporan Modul Sering Terdapat Complain

Pada gambar 3.24 menjelaskan tentang laporan modul sering terdapat complain yang diterima oleh koordinator *sofdev*. Dimulai dari proses login pada website dengan memasukkan *username* dan *password* pada website kemudian koordinator *sofdev* memilih menu laporan modul sering terdapat complain. Setelah itu koordinaor *sofdev* melihat laporan modul yang sering terdapat complain.

3.5.2 Data Flow Diagram (DFD)

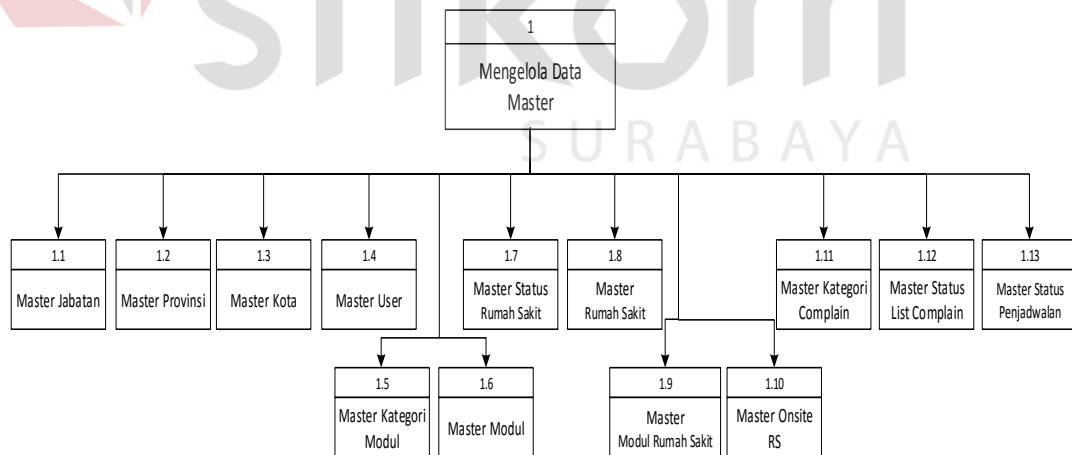
A. Diagram Berjenjang Aplikasi Penjadwalan Penanganan Complain



Gambar 3.25 Diagram Berjenjang Aplikasi Penjadwalan Penanganan Complain

Gambar 3.25 menjelaskan tentang diagram jenjang aplikasi penjadwalan penanganan complain. Dari gambar tersebut terdapat tiga proses yaitu mengelola data master, mengelola transaksi dan mengelola laporan.

B. Diagram Berjenjang Aplikasi Penjadwalan Penanganan Complain Mengelola Data Master

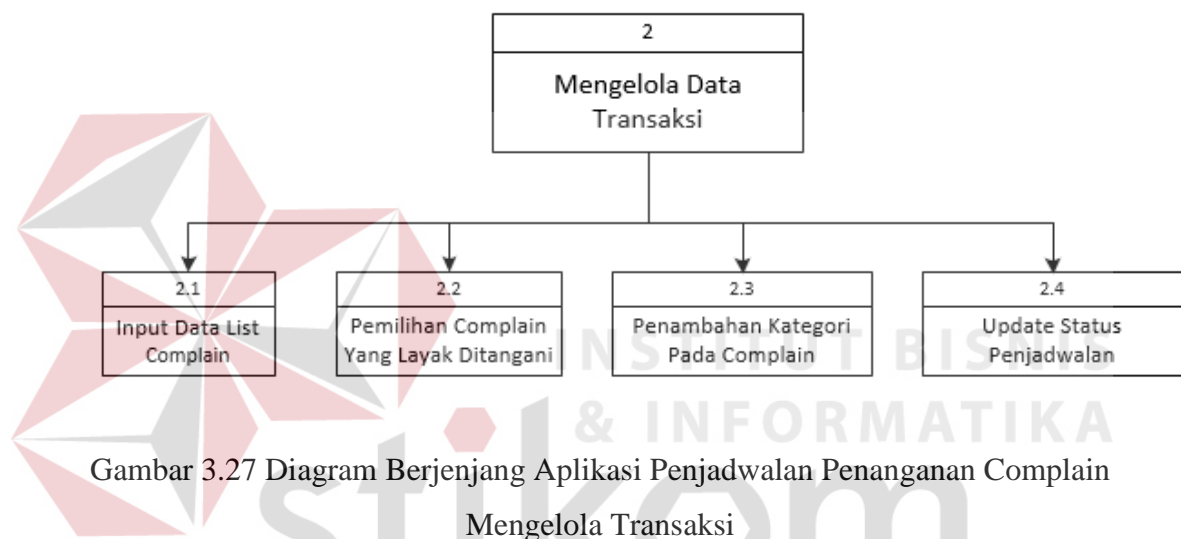


Gambar 3.26 Diagram Berjenjang Aplikasi Penjadwalan Penanganan Complain Mengelola Data Master

Gambar 3.26 menjelaskan tentang proses lanjutan dari mengelola data master. Dari proses mengelola data master dapat diturunkan lagi menjadi sepuluh

proses, yaitu master jabatan, master provinsi, master kota, master user, master modul, master status rumah sakit, master rumah sakit, master rumah sakit, master kategori complain, master status list complain, master status penjadwalan. Data master digunakan untuk menunjang berjalannya proses penjadwalan hingga menghasilkan sebuah laporan.

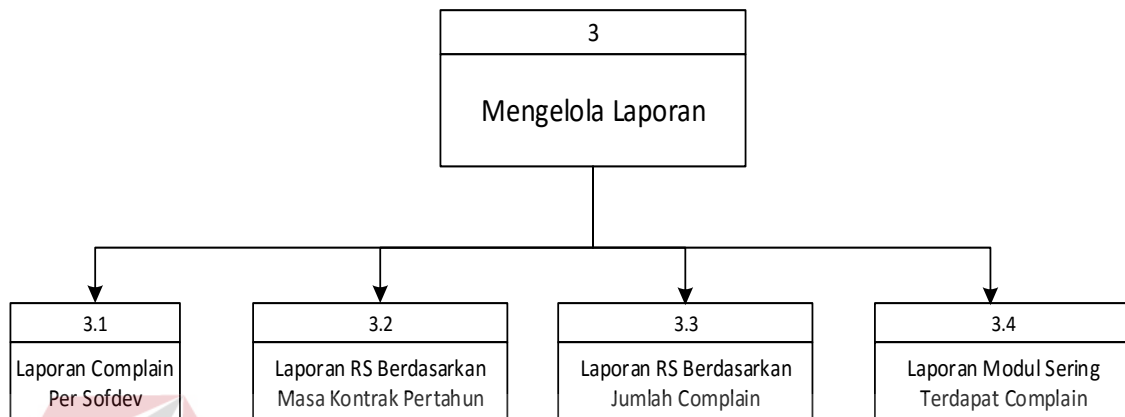
C. Diagram Berjenjang Aplikasi Penjadwalan Penanganan Complain Mengelola Transaksi



Gambar 3.27 Diagram Berjenjang Aplikasi Penjadwalan Penanganan Complain Mengelola Transaksi

Gambar 3.27 menjelaskan tentang proses lanjutan dari mengelola transaksi. Dari proses mengelola transaksi dapat diturunkan lagi menjadi empat proses yaitu input data list complain, pemilihan complain yang layak ditangani, penambahan kategori pada complain dan update status penjadwalan.

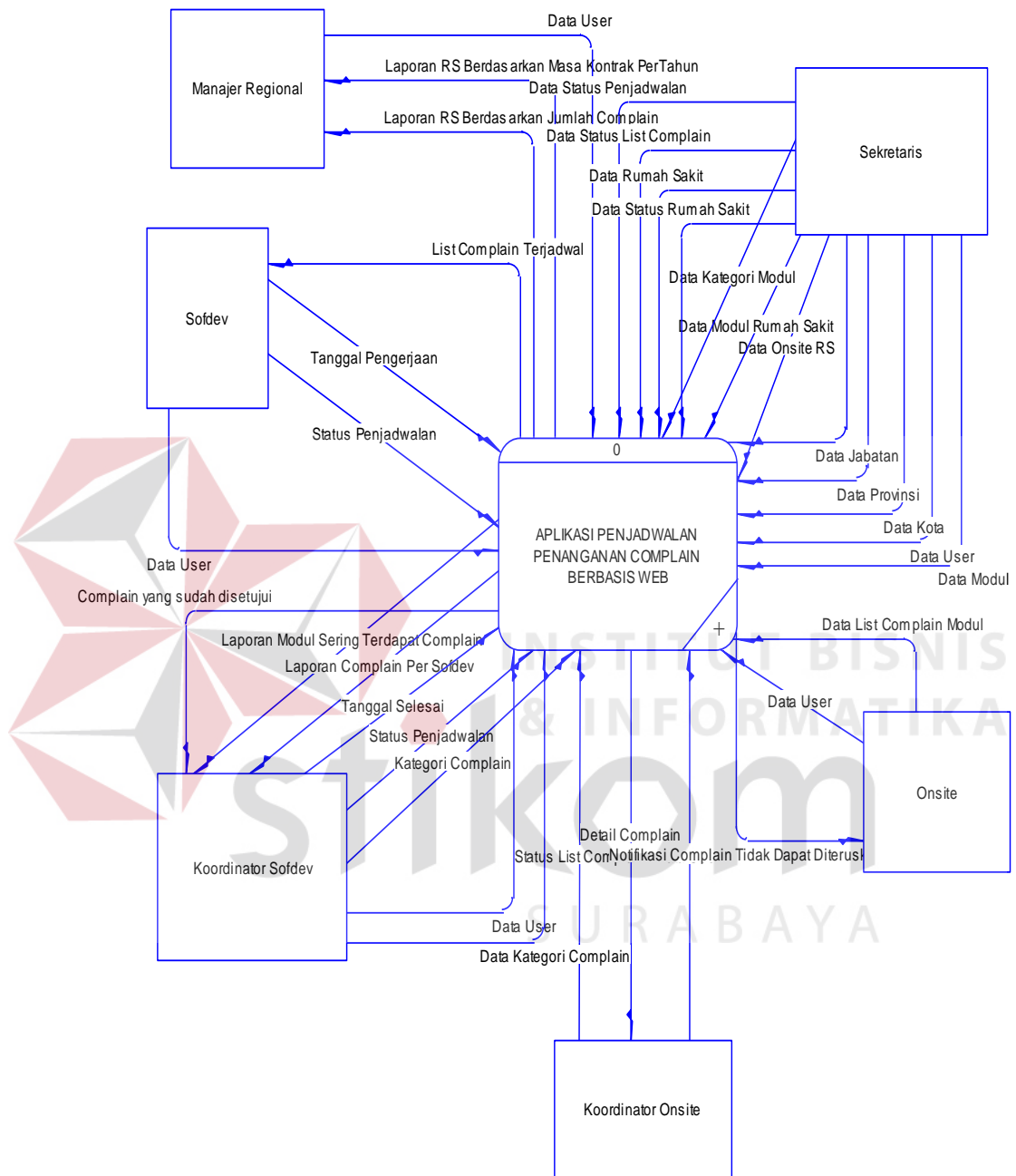
D. Diagram Berjenjang Aplikasi Penjadwalan Penanganan Complain Mengelola Laporan



Gambar 3.28 Diagram Berjenjang Aplikasi Penjadwalan Penanganan Complain Mengelola Laporan

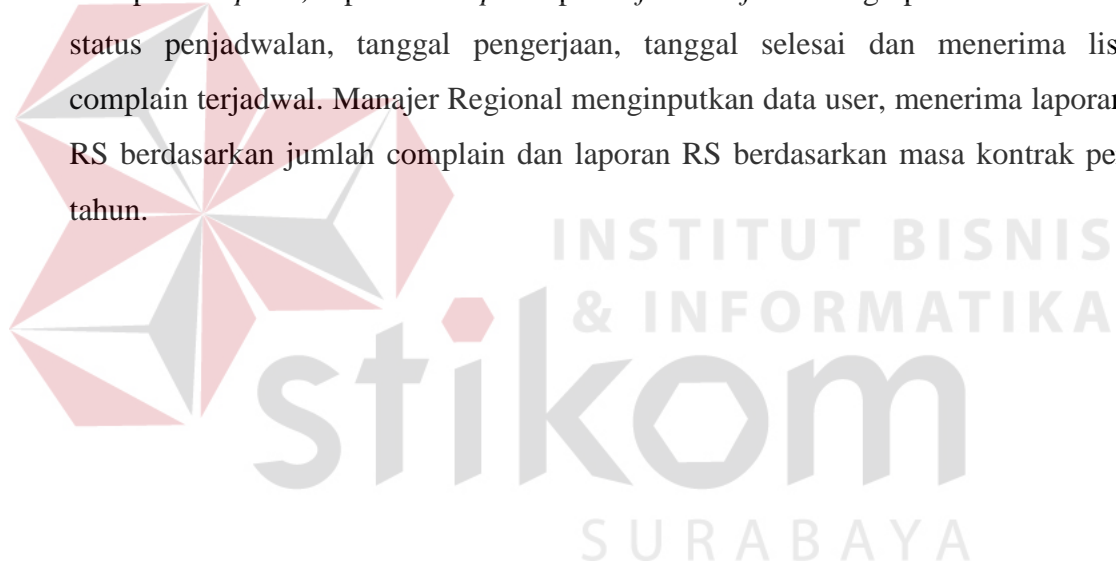
Gambar 3.28 menjelaskan tentang proses lanjutan dari mengelola laporan. Dari proses mengelola laporan dapat diturunkan lagi menjadi empat proses yaitu laporan complain per *sofdev*, laporan RS berdasarkan masa kontrak pertahun, laporan RS berdasarkan jumlah complain, laporan modul sering terdapat complain. Laporan tersebut diperoleh dari hasil proses mengelola data master dan mengelola transaksi.

E. Context Diagram Aplikasi Penjadwalan Penanganan Complain

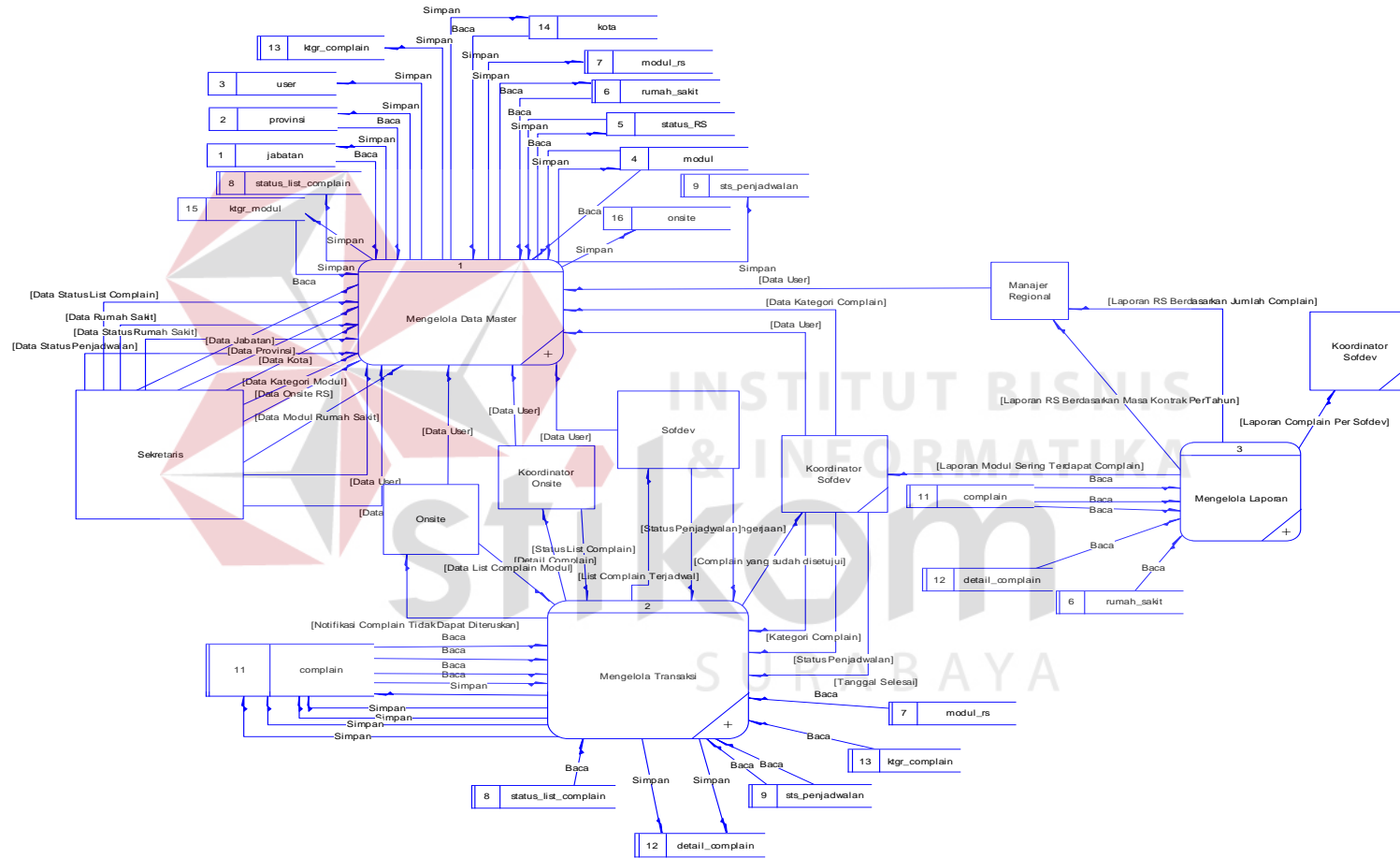


Gambar 3.29 Context Diagram Aplikasi Penjadwalan Penanganan Complain

Gambar 3.29 menjelaskan tentang *context diagram* aplikasi penjadwalan penanganan complain. Terdapat enam *eksternal entity* yaitu sekretaris yang menginputkan data master kecuali master kategori *complain* dikarenakan master kategori *complain* terdapat tanggal yang hanya diketahui oleh koordinator *sofdev* yang ada pada aplikasi penjadwalan kecuali data kategori *complain*. *Onsite* menginputkan data user, data list complain modul dan mendapatkan data notifikasi tidak dapat diteruskan. Koordinator *onsite* menginputkan data user, status *list complain* dan menerima detail *complain*. Koordinator *sofdev* menginputkan data user data kategori complain, kategori complain, status penjadwalan, dan menerima data user yang disetujui, laporan modul sering terdapat *complain*, laporan *complain* per *sofdev*. *Sofdev* menginputkan data user, status penjadwalan, tanggal pengerjaan, tanggal selesai dan menerima list complain terjadwal. Manajer Regional menginputkan data user, menerima laporan RS berdasarkan jumlah complain dan laporan RS berdasarkan masa kontrak per tahun.



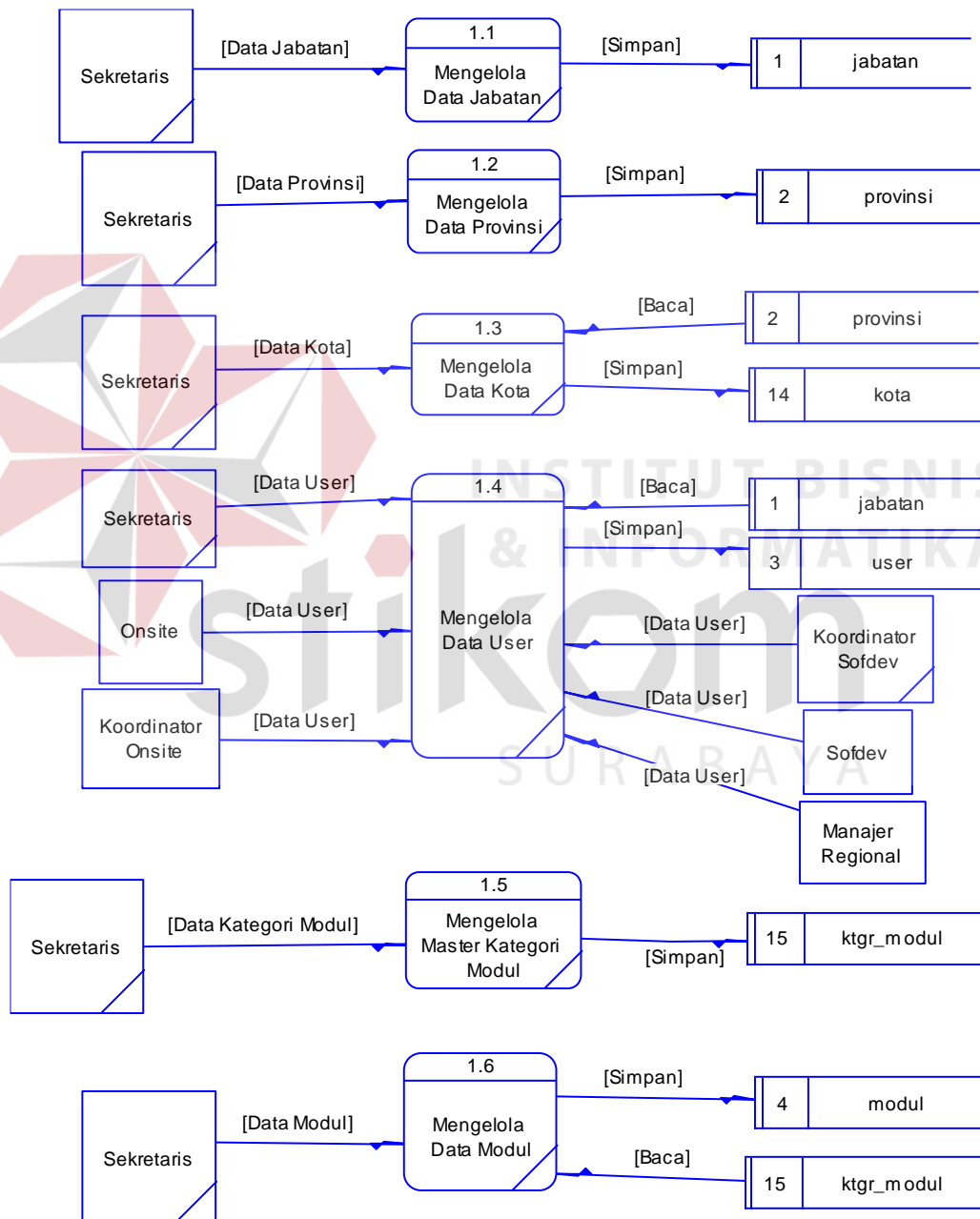
F. DFD Level 0 Aplikasi Penjadwalan Penanganan Complain

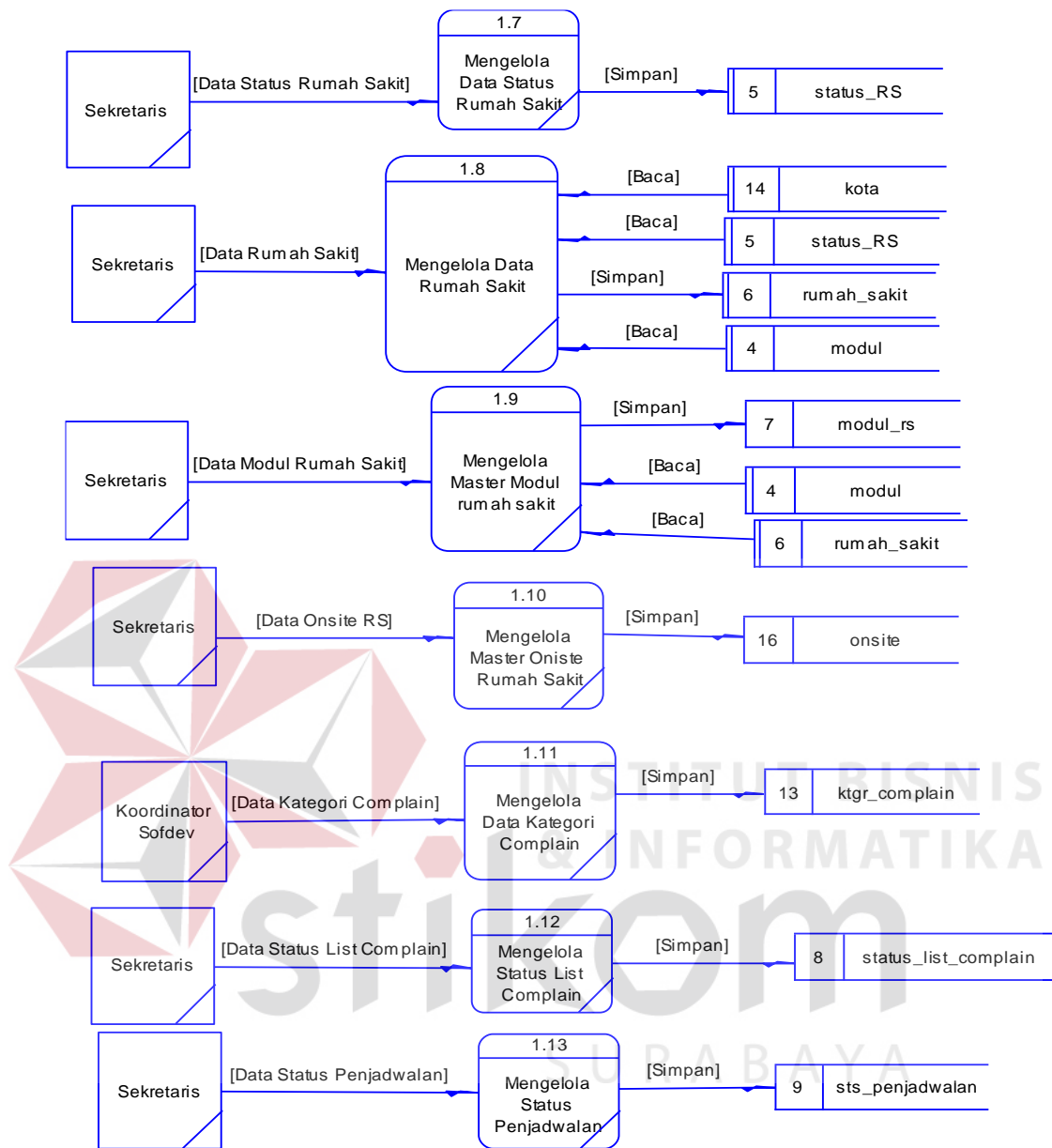


Gambar 3.30 DFD Level 0 Aplikasi Penjadwalan Penanganan Complain

Gambar 3.30 menjelaskan tentang proses DFD level 0 dimana terdapat tiga *proses* yaitu mengelola data master, mengelola transaksi dan mengelola laporan secara umum. Untuk proses yang lebih detail akan dijelaskan pada DFD lv 1 masing-masing proses yang telah di *brigdown*.

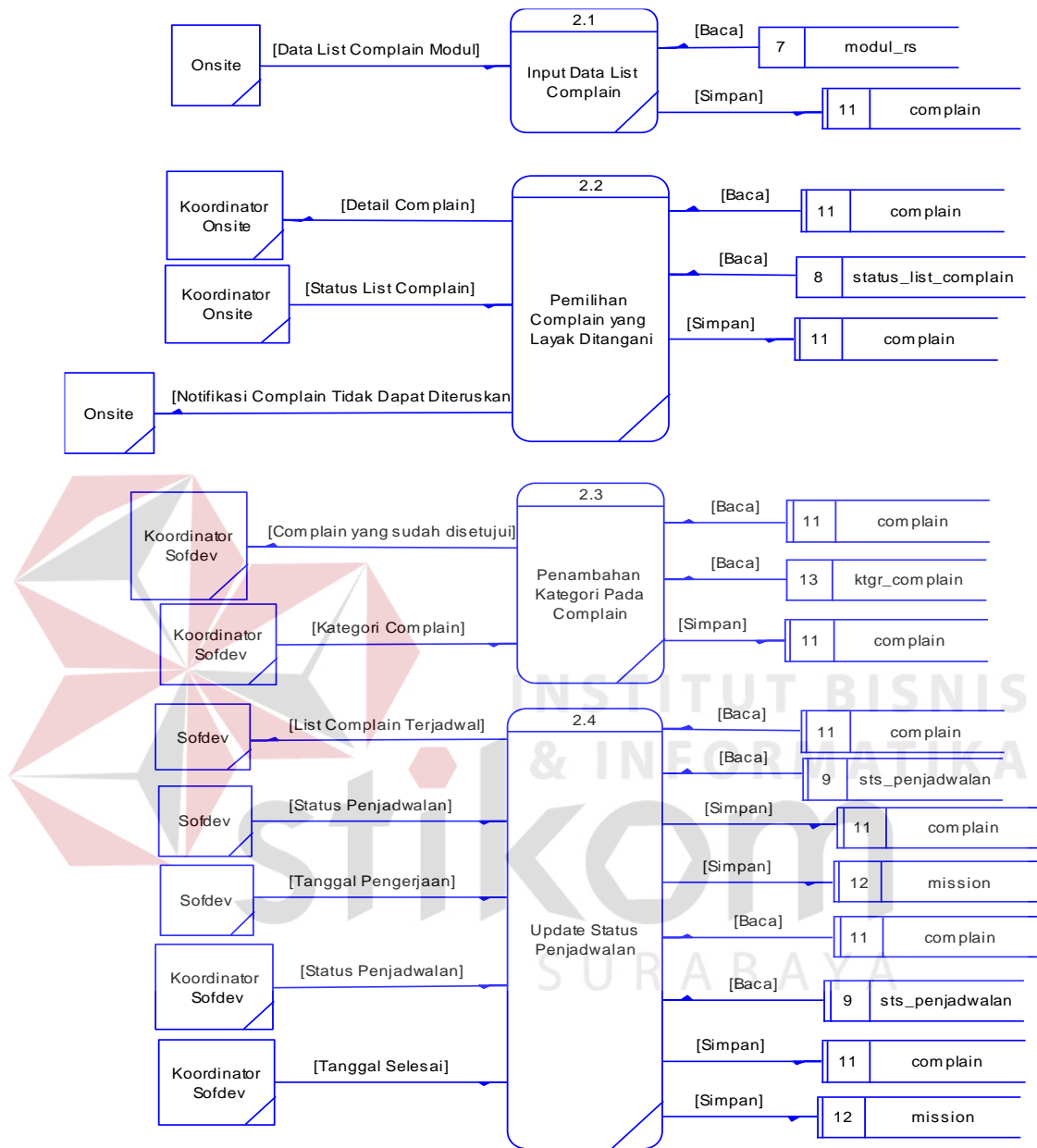
G. DFD Level 1 Mengelola Data Master





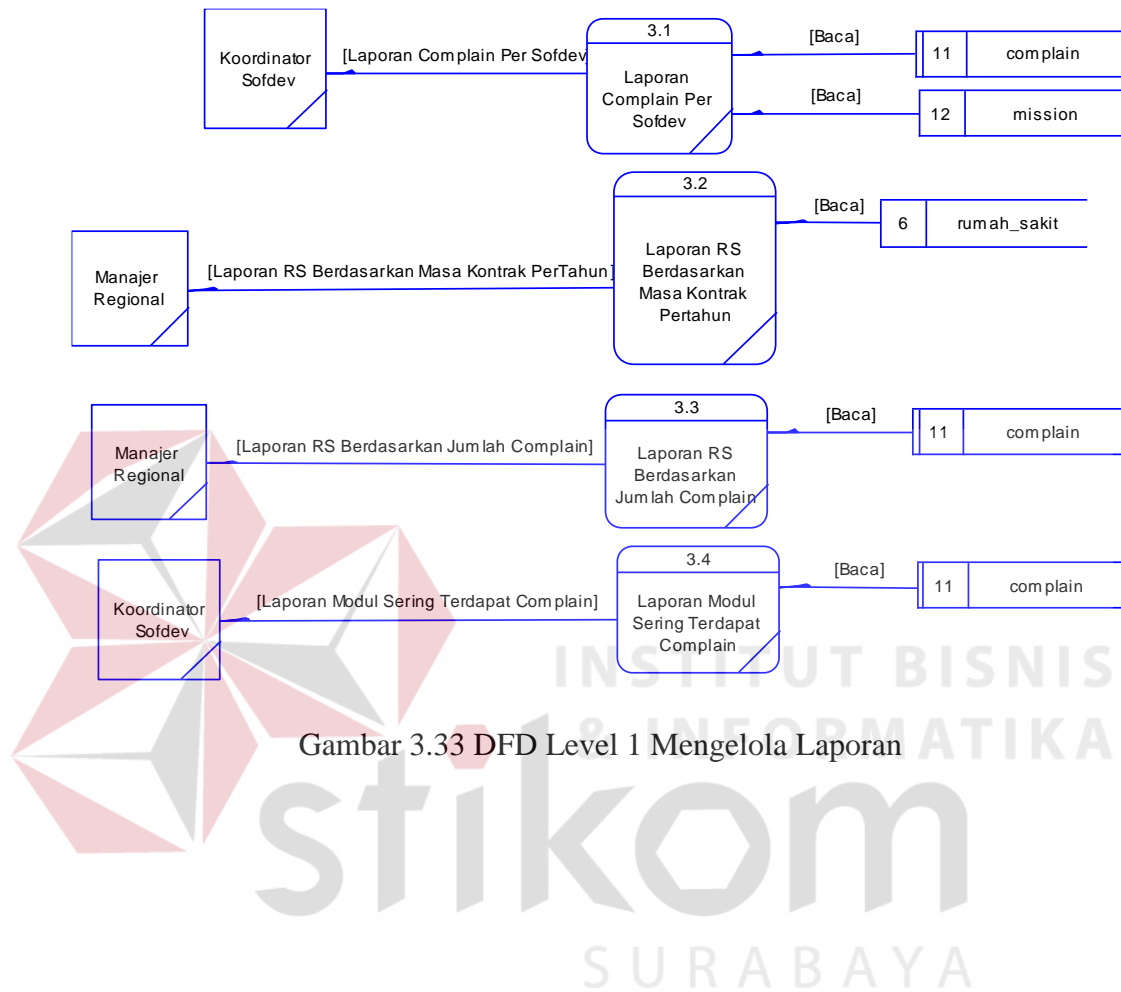
Gambar 3.31 DFD Level 1 Mengelola Data Master

H. DFD Level 1 Mengelola Transaksi



Gambar 3.32 DFD Level 1 Mengelola Transaksi

I. DFD Level 1 Mengelola Laporan

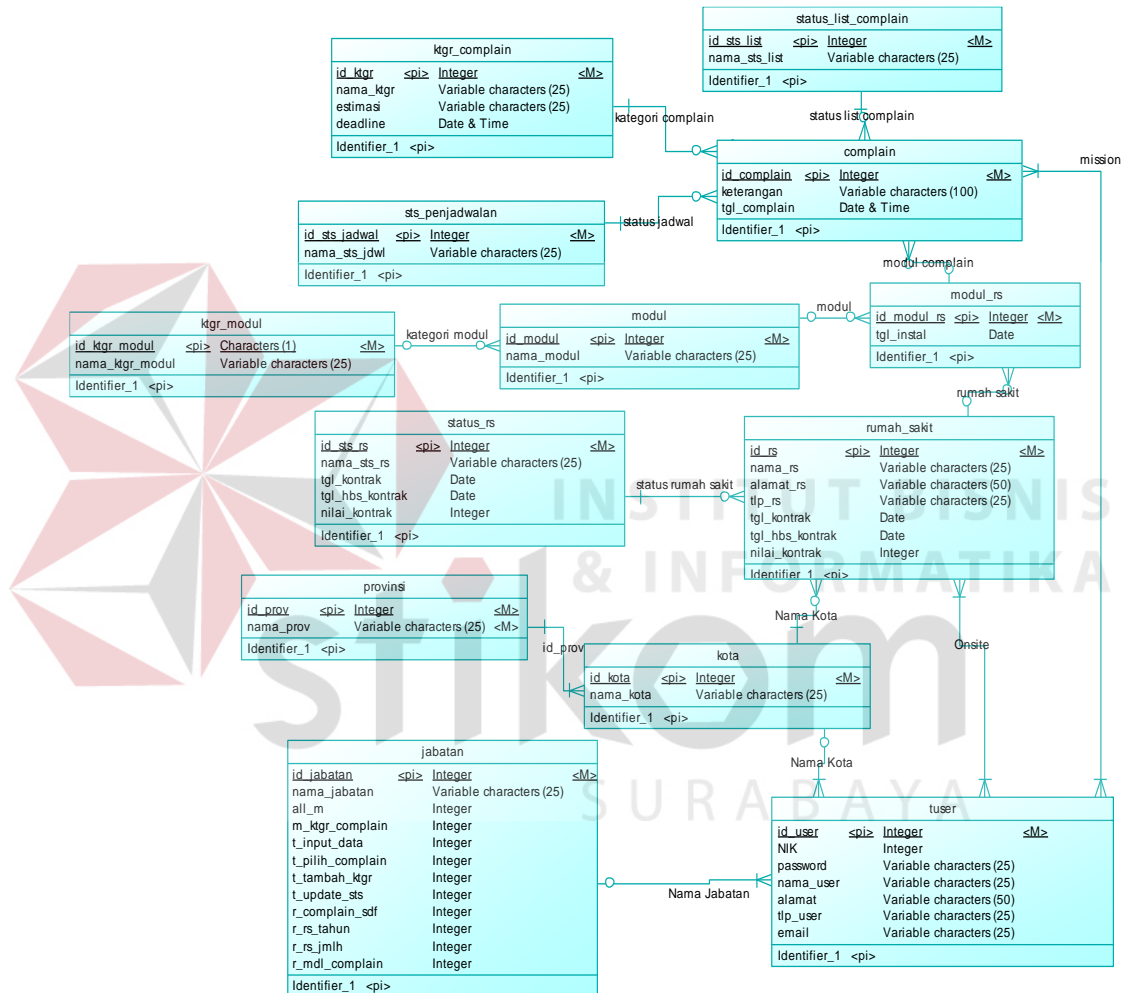


Gambar 3.33 DFD Level 1 Mengelola Laporan

3.5.3 ERD (Entity Relationship Diagram)

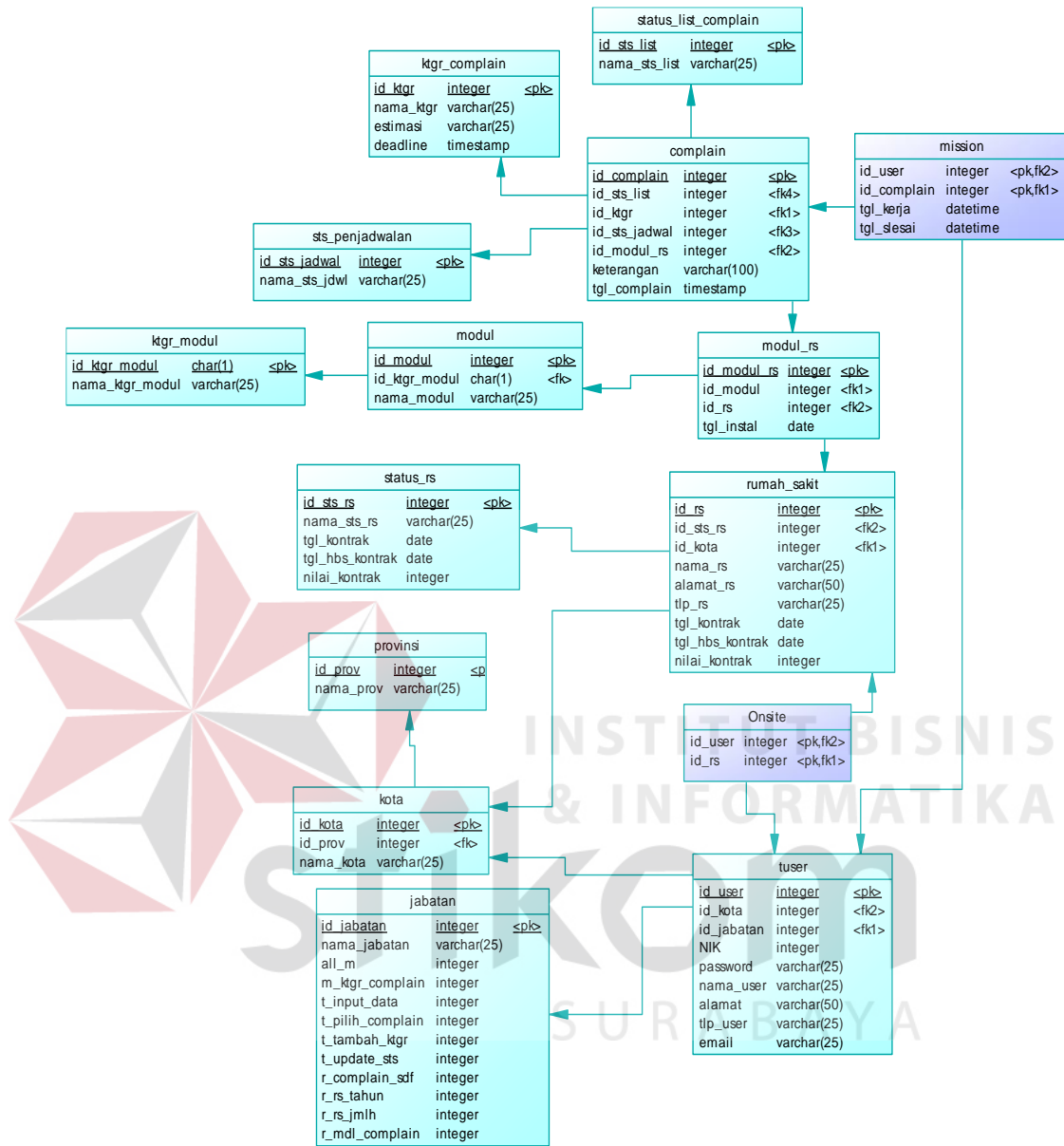
ERD menggambarkan data sebagai entitas, relasi dan *attribute* dari database yang telah dirancang. ERD dapat dibagi menjadi dua, yaitu *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM).

A. CDM (Conceptual Data Model)



Gambar 3.34 Conceptual Data Model (CDM)

B. PDM (Physical Data Model)



Gambar 3.35 Physical Data Model (PDM)

3.5.4 Struktur Tabel

Pada aplikasi penjadwalan complain pada PT. NCI cabang Surabaya ini terdapat lima belas tabel. Struktur tabel pada aplikasi penjadwalan complain pada PT. NCI cabang Surabaya sebagai berikut:

a. Tabel jabatan

Primary Key : id_jabatan

Foreign Key : -

Fungsi : Menyimpan data jabatan karyawan

Tabel 3.19 Struktur Tabel jabatan

| <i>Field</i> | <i>Type Data</i> | <i>Length</i> | <i>Keterangan</i> |
|--------------|------------------|---------------|--------------------|
| id_jabatan | Char | 2 | <i>Primary Key</i> |
| nama_jabatan | Varchar | 25 | |

b. Tabel provinsi

Primary Key : id_provinsi

Foreign Key : -

Fungsi : Menyimpan data provinsi

Tabel 3.20 Struktur Tabel provinsi

| <i>Field</i> | <i>Type Data</i> | <i>Length</i> | <i>Keterangan</i> |
|--------------|------------------|---------------|--------------------|
| id_prov | Char | 2 | <i>Foreign key</i> |
| nama_prov | Varchar | 25 | |

c. Tabel kota

Primary Key : id_kota

Foreign Key : id_provinsi

Fungsi : Menyimpan data kota

Tabel 3.21 Struktur Tabel kota

| <i>Field</i> | <i>Type Data</i> | <i>Length</i> | <i>Keterangan</i> |
|--------------|------------------|---------------|--------------------|
| id_kota | Char | 4 | <i>Primary Key</i> |
| id_prov | Char | 2 | <i>Foreign key</i> |
| nama_kota | Varchar | 25 | |

d. Tabel status_rs

Primary Key : id_sts_rs

Foreign Key : -

Fungsi : Menyimpan data status rumah sakit

Tabel 3.22 Struktur Tabel status_rs

| <i>Field</i> | <i>Type Data</i> | <i>Length</i> | <i>Keterangan</i> |
|--------------|------------------|---------------|--------------------|
| id_sts_rs | Char | 1 | <i>Primary Key</i> |
| nama_sts_rs | Varchar | 25 | |

e. Tabel ktgr_modul

Primary Key : id_ktgr_modul

Foreign Key : -

Fungsi : Menyimpan data kategori modul

Tabel 3.23 Struktur Tabel ktgr_modul

| <i>Field</i> | <i>Type Data</i> | <i>Length</i> | <i>Keterangan</i> |
|-----------------|------------------|---------------|--------------------|
| id_ktgr_modul | Char | 1 | <i>Primary Key</i> |
| nama_ktgr_modul | Varchar | 25 | |

f. Tabel Modul

Primary Key : id_modul

Foreign Key : id_ktgr_modul

Fungsi : Menyimpan data modul

Tabel 3.24 Struktur Tabel modul

| <i>Field</i> | <i>Type Data</i> | <i>Length</i> | <i>Keterangan</i> |
|---------------|------------------|---------------|--------------------|
| id_modul | Integer | 11 | <i>Primary Key</i> |
| id_ktgr_modul | Char | 1 | <i>Foreign key</i> |
| nama_modul | Varchar | 25 | |

g. Tabel rumah_sakit

Primary Key : id_rs

Foreign Key : id_sts_rs, id_kota

Fungsi : Menyimpan data rumah sakit

Tabel 3.25 Struktur Tabel rumah_sakit

| <i>Field</i> | <i>Type Data</i> | <i>Length</i> | <i>Keterangan</i> |
|-----------------|------------------|---------------|--------------------|
| id_rs | Integer | 11 | <i>Primary Key</i> |
| id_sts_rs | Char | 1 | <i>Foreign key</i> |
| id_kota | Char | 4 | <i>Foreign key</i> |
| nama_rs | Varchar | 25 | |
| alamat_rs | Varchar | 50 | |
| tlp_rs | Varchar | 25 | |
| tgl_kontrak | Date | | |
| tgl_hbs_kontrak | Date | | |
| nilai_kontrak | integer | 11 | |

h. Tabel modul_rs

Primary Key : id_modul_rs

Foreign Key : id_modul, id_rs

Fungsi : Menyimpan data modul yang sudah terinstal di rumah sakit

Tabel 3.26 Struktur Tabel modul_rs

| <i>Field</i> | <i>Type Data</i> | <i>Length</i> | <i>Keterangan</i> |
|--------------|------------------|---------------|--------------------|
| id_modul_rs | Integer | 11 | <i>Primary Key</i> |
| id_modul | Integer | 11 | <i>Foreign key</i> |
| id_rs | Integer | 11 | <i>Foreign key</i> |
| tgl_instal | Date | | |

i. Tabel tuser

Primary Key : NIK

Foreign Key : id_kota, id_jabatan

Fungsi : Menyimpan data user

Tabel 3.27 Struktur Tabel tuser

| <i>Field</i> | <i>Type Data</i> | <i>Length</i> | <i>Keterangan</i> |
|--------------|------------------|---------------|--------------------|
| NIK | Varchar | 25 | <i>Primary Key</i> |
| id_kota | Char | 4 | <i>Foreign key</i> |
| id_jabatan | Char | 2 | <i>Foreign key</i> |
| password | Varchar | 50 | |
| nama_user | Varchar | 25 | |
| alamat | Text | | |
| tlp_user | Varchar | 25 | |
| email | Varchar | 25 | |

j. Tabel onsite

Primary Key : NIK, id_rs

Foreign Key : NIK, id_rs

Fungsi : Menyimpan data pegawai onsite yang ada di rumah sakit

Tabel 3.28 Struktur Tabel onsite

| <i>Field</i> | <i>Type Data</i> | <i>Length</i> | <i>Keterangan</i> |
|--------------|------------------|---------------|---------------------------------|
| NIK | Varchar | 25 | <i>Primary Key, Foreign key</i> |
| id_rs | Integer | 11 | <i>Primary Key, Foreign key</i> |

k. Tabel status_list_complain

Primary Key : id_sts_list

Foreign Key : -

Fungsi : Menyimpan data status list complain

Tabel 3.29 Struktur Tabel status_list_complain

| <i>Field</i> | <i>Type Data</i> | <i>Length</i> | <i>Keterangan</i> |
|---------------|------------------|---------------|--------------------|
| id_sts_list | Char | 1 | <i>Primary Key</i> |
| nama_sts_list | Varchar | 25 | |

1. Tabel ktgr_complain

Primary Key : id_ktgr

Foreign Key : -

Fungsi : Menyimpan data kategori complain

Tabel 3.30 Struktur Tabel ktgr_complain

| <i>Field</i> | <i>Type Data</i> | <i>Length</i> | <i>Keterangan</i> |
|--------------|------------------|---------------|--------------------|
| id_ktgr | Char | 1 | <i>Primary Key</i> |
| nama_ktgr | Varchar | 25 | |
| estimasi | Varchar | 25 | |
| deadline | Varchar | 25 | |

m. Tabel sts_penjadwalan

Primary Key : id_sts_jadwal

Foreign Key : -

Fungsi : Menyimpan data status penjadwalan

Tabel 3.31 Struktur Tabel sts_penjadwalan

| <i>Field</i> | <i>Type Data</i> | <i>Length</i> | <i>Keterangan</i> |
|-----------------|------------------|---------------|--------------------|
| id_sts_jadwal | Char | 1 | <i>Primary Key</i> |
| nama_sts_jadwal | Varchar | 25 | |

n. Tabel complain

Primary Key : id_complain

Foreign Key : id_sts_list, id_ktgr, id_sts_jadwal, id_modul_rs

Fungsi : Menyimpan data yang berhubungan dengan complain

Tabel 3.32 Struktur Tabel complain

| <i>Field</i> | <i>Type Data</i> | <i>Length</i> | <i>Keterangan</i> |
|---------------|------------------|---------------|--------------------|
| id_complain | Varchar | 10 | <i>Primary Key</i> |
| id_sts_list | Char | 1 | <i>Foreign key</i> |
| id_ktgr | Char | 1 | <i>Foreign key</i> |
| id_sts_jadwal | Char | 1 | <i>Foreign key</i> |
| id_modul_rs | Integer | 11 | <i>Foreign key</i> |
| keterangan | Text | | |
| tgl_complain | Datetime | | |

o. Tabel mission

Primary Key : NIK, id_complain

Foreign Key : NIK, id_complain

Fungsi : Menyimpan data waktu *record* kerja *sofdev*

Tabel 3.33 Struktur Tabel mission

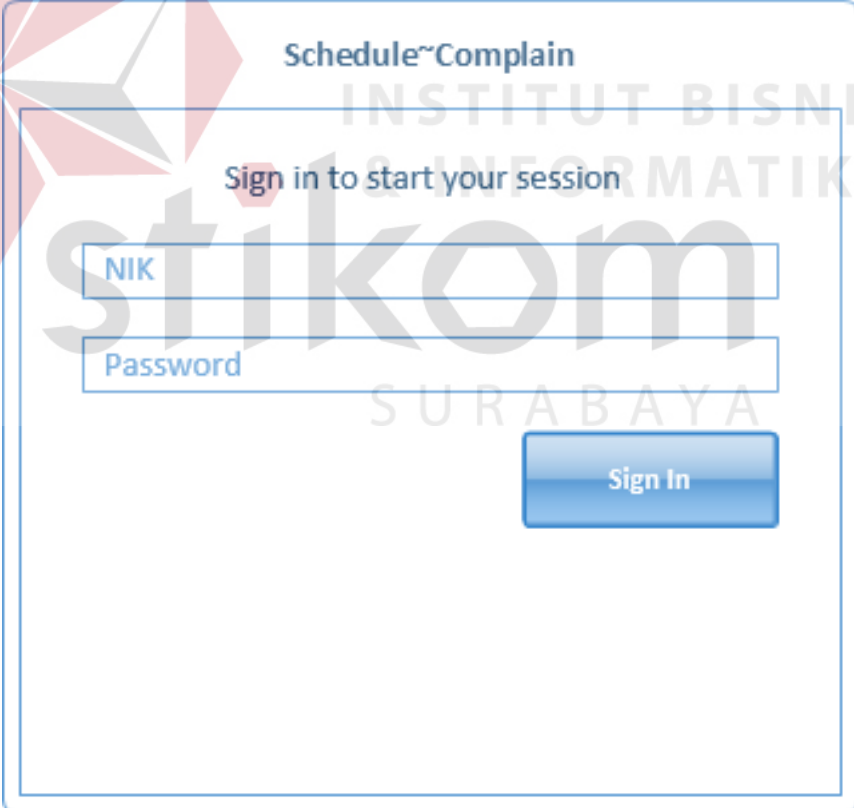
| <i>Field</i> | <i>Type Data</i> | <i>Length</i> | <i>Keterangan</i> |
|--------------|------------------|---------------|----------------------|
| id_mission | Char | 1 | <i>Primery Key</i> |
| NIK | Varchar | 25 | <i>Foreign key</i> |
| id_complain | Varchar | 10 | <i>, Foreign key</i> |
| tgl_kerja | Datetime | | |
| tgl_selesai | Datetime | | |

3.5.5 Desain *Input Output*

Desain *input output* digunakan untuk memberikan gambaran antarmuka aplikasi yang akan dibuat. Berikut ini adalah rancangan antarmuka dari aplikasi penjadwalan penanganan *complain* pada PT Nuansa Cerah Informasi Cabang Surabaya.

a. Tampilan *Login*

Tampilan login digunakan oleh *user* untuk mengakses aplikasi penjadwalan penanganan *complain*. *User* yang berhak mengakses aplikasi ini adalah sekretaris, *onsite*, koordinator *onsite*, koordinator *sofdev*, *sofdev* dan manajer regional. Untuk melakukan proses *login*, *user* menginputkan NIK dan Password. Gambar 3.36 menggambarkan tampilan desain dari menu login.



The image shows a login window titled "Schedule~Complain". The window contains the following elements:

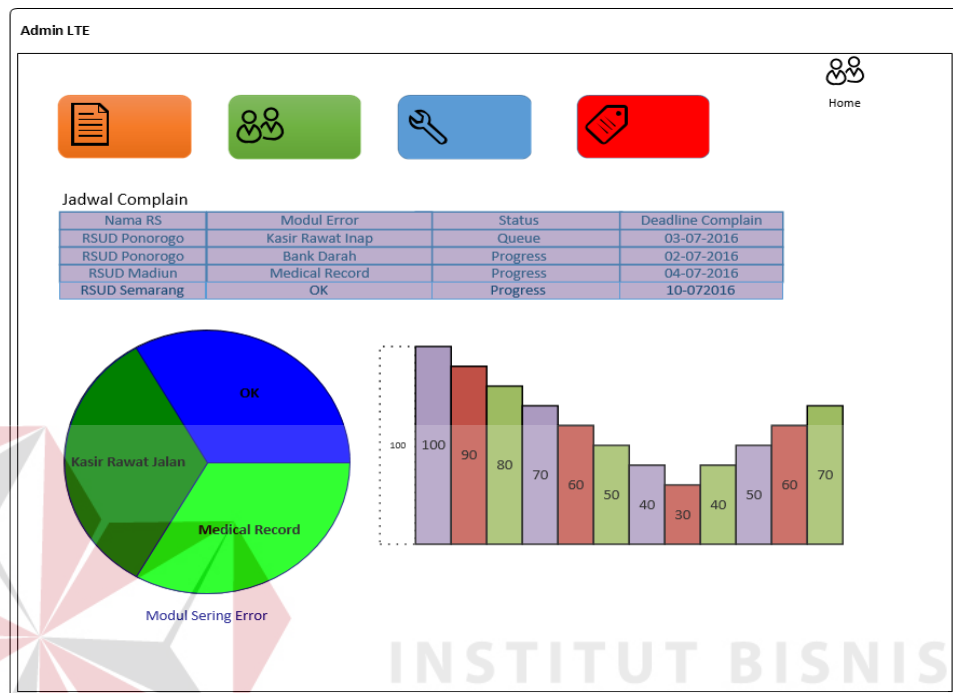
- Header: "Schedule~Complain"
- Text: "Sign in to start your session"
- Input field: "NIK"
- Input field: "Password"
- Button: "Sign In"

The background of the window features a large, stylized logo of STIKOM SURABAYA, which is a red and white geometric design.

Gambar 3.36 Tampilan Login

b. Tampilan Home

Tampilan *home* adalah tampilan informasi yang dibutuhkan oleh user yang di letakkan pada halaman home



Gambar 3.37 Tampilan Home

c. Tampilan Master Jabatan

Tampilan master jabatan digunakan oleh sekretaris untuk menginputkan data jabatan yang berguna dalam proses pada aplikasi penjadwalan penanganan complain. Dari jabatan ini dapat ditentukan juga menu apa saja yang dapat di gunakan oleh jabatan tertentu. Gambar 3.38 menunjukkan tampilan dari master jabatan.

Admin LTE Sekretaris

Home > Master > Jabatan

ID Jabatan

Nama Jabatan

All Master Tanpa Kategori Complain (Menu1) (Master) Kategori Complain (Menu2)
 (Transaksi) Input Data List Complain (Menu3) (Transaksi) Pemilihan Complain (Menu4)
 (Transaksi) Penambahan Kategori (Menu5) (Transaksi) Update Status Penjadwalan (Menu6)
 (Report) Complain Per Sofdev (Menu7) (Report) RS Berdasarkan Masa Kontrak Pertahun (Menu8)
 (Report) RS Berdasarkan Jumlah Complain (Menu9) (Report) Modul Sering Terdapat Complain (Menu10)

Gambar 3.38 Tampilan Master Jabatan

d. Tampilan Master Provinsi

Tampilan master provinsi berfungsi untuk menginputkan data provinsi oleh sekretaris. Sekretaris terlebih dulu masuk pada menu master dan provinsi untuk mengakses halaman master provinsi. Gambar 3.39 menunjukkan tampilan dari master provinsi.

Admin LTE Sekretaris

Home > Master > Provinsi

ID Provinsi

Nama Provinsi

| ID Provinsi | Nama Provinsi |
|-------------|---------------|
| | |
| | |
| | |

Gambar 3.39 Tampilan Master Provinsi

e. Tampilan Master Kota

Tampilan master kota berfungsi untuk menginputkan data kota oleh sekretaris. Sekretaris terlebih dulu masuk pada menu master dan kota untuk mengakses halaman master kota. Sebelum menginputkan data kota sekretaris harus memilih provinsi yang sesuai dengan kota yang akan diinputkan. Gambar 3.40 menunjukkan tampilan dari master kota.

The screenshot shows a web interface for 'Admin LTE' with the user role 'Sekretaris'. The breadcrumb trail is 'Home > Master > Kota'. The form contains three text input fields: 'ID Kota', 'Provinsi', and 'Nama Kota'. The 'Provinsi' field has a dropdown arrow on the right. Below the inputs are 'Save' and 'Cancel' buttons. At the bottom, there is a table with the following structure:

| ID Kota | Nama Provinsi | Nama Kota |
|---------|---------------|-----------|
| | | |
| | | |
| | | |

Gambar 3.40 Tampilan Master Kota

f. Tampilan Master Status Rumah Sakit

Tampilan master kota berfungsi untuk menginputkan data status rumah sakit oleh sekretaris. Sekretaris terlebih dulu masuk pada menu master dan Status RS untuk mengakses halaman master Status RS. Gambar 3.41 menunjukkan tampilan dari master status rumah sakit.

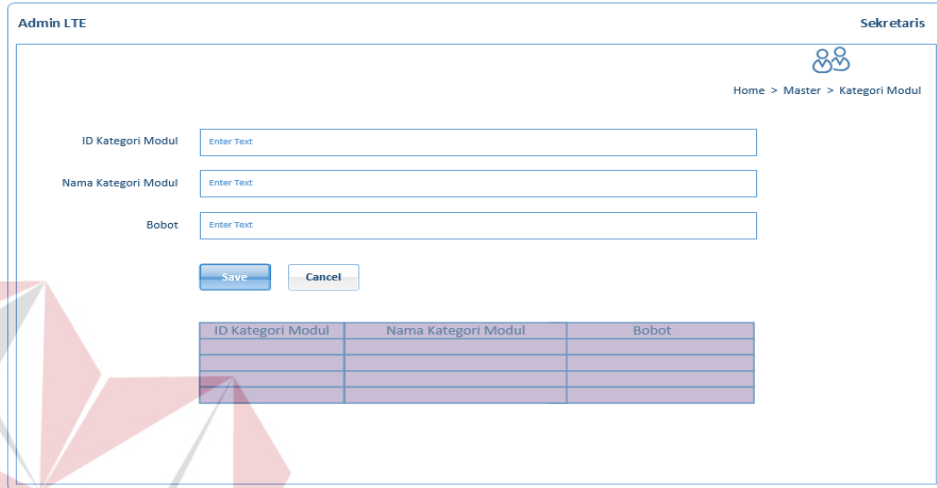
The screenshot shows a web interface for 'Admin LTE' with the user role 'Sekretaris'. The breadcrumb trail is 'Home > Master > Status RS'. The form contains three text input fields: 'ID Status RS', 'Nama Status RS', and 'Bobot'. Below the inputs are 'Save' and 'Cancel' buttons. At the bottom, there is a table with the following structure:

| ID Status RS | Nama Status RS | Bobot |
|--------------|----------------|-------|
| | | |
| | | |
| | | |

Gambar 3.41 Tampilan Master Status Rumah Sakit

g. Tampilan Master Kategori Modul

Tampilan master kategori modul berfungsi untuk menginputkan data kategori modul oleh sekretaris. Sekretaris terlebih dulu masuk pada menu master dan kategori modul untuk mengakses halaman master kategori modul. Gambar 3.42 menunjukkan tampilan dari master kategori modul.



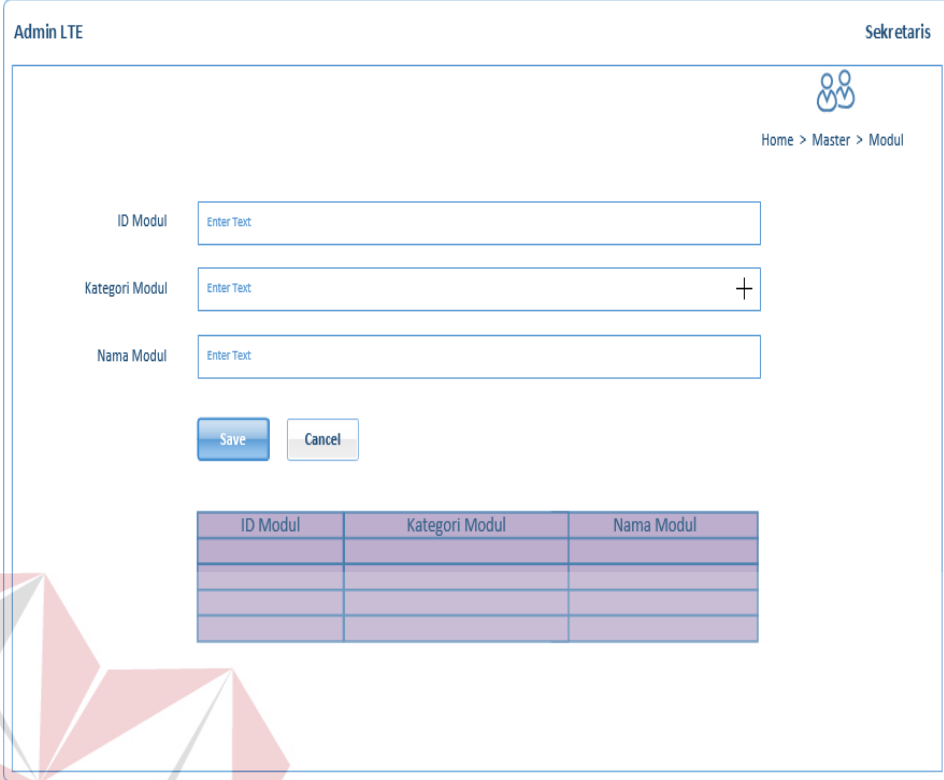
The screenshot displays a web interface for managing module categories. At the top, it shows 'Admin LTE' on the left and 'Sekretaris' on the right. A breadcrumb trail reads 'Home > Master > Kategori Modul'. The main area contains three text input fields labeled 'ID Kategori Modul', 'Nama Kategori Modul', and 'Bobot', each with a placeholder 'Enter Text'. Below these fields are 'Save' and 'Cancel' buttons. At the bottom, there is a table with three columns: 'ID Kategori Modul', 'Nama Kategori Modul', and 'Bobot'. The table currently contains no data rows.

| ID Kategori Modul | Nama Kategori Modul | Bobot |
|-------------------|---------------------|-------|
| | | |
| | | |
| | | |

Gambar 3.42 Tampilan Master Kategori Modul

h. Tampilan Master Modul

Tampilan master modul berfungsi untuk menginputkan data modul oleh sekretaris. Sekretaris terlebih dulu masuk pada menu master dan modul untuk mengakses halaman master modul. Saat proses *input* data modul terlebih dulu harus memilih kategori modul yang sesuai dengan modul yang akan di inputkan. Gambar 3.43 menunjukkan tampilan dari master modul.



The screenshot displays a web application interface for 'Admin LTE' with the user role 'Sekretaris'. The breadcrumb trail is 'Home > Master > Modul'. The form contains three text input fields: 'ID Modul', 'Kategori Modul', and 'Nama Modul', each with a placeholder 'Enter Text'. The 'Kategori Modul' field includes a '+' icon. Below the inputs are 'Save' and 'Cancel' buttons. A table below the buttons has the following structure:

| ID Modul | Kategori Modul | Nama Modul |
|----------|----------------|------------|
| | | |
| | | |
| | | |

Gambar 3.43 Tampilan Master Modul

i. Tampilan Master Rumah Sakit

Tampilan master rumah sakit berfungsi untuk menginputkan data rumah sakit oleh sekretaris. Sekretaris terlebih dulu masuk pada menu master dan rumah sakit untuk mengakses halaman master rumah sakit. Saat proses input data rumah sakit terlebih dulu harus memilih kota dan status rumah sakit yang sesuai dengan data rumah sakit yang akan diinputkan. Gambar 3.44 menunjukkan tampilan dari master rumah sakit.

The screenshot shows a web application interface for 'Admin LTE' with the user role 'Sekretaris'. The breadcrumb trail is 'Home > Master > Rumah Sakit'. The form contains the following fields:

- ID RS: Enter Text
- Rumah Sakit: Enter Text
- Alamat: Enter Text
- Kota: Enter Text +
- Telepon: Enter Text
- Status RS: Enter Text +
- Tanggal Kontrak: Enter Text
- Tanggal Habis Kontrak: Enter Text
- Nilai Kontrak: Enter Text

Buttons: Save, Cancel

Gambar 3.44 Tampilan Master Rumah Sakit

j. Tampilan Master Modul Rumah Sakit

Tampilan master modul rumah sakit berfungsi untuk menginputkan data modul yang terinstal pada setiap rumah sakit oleh sekretaris. Sekretaris terlebih dulu masuk pada menu master dan modul rumah sakit untuk mengakses halaman master modul rumah sakit. Terdapat fasilitas *search* untuk mencari modul yang terinstal di setiap rumah sakit. Gambar 3.45 menunjukkan tampilan dari master modul rumah sakit.

Admin LTE Sekretaris

Home > Master > Modul RS

Rumah Sakit

Input Data

ID Modul RS

Modul

Tanggal Instal +

| ID Modul RS | Rumah Sakit | Modul | Tanggal Instal |
|-------------|-------------|-------|----------------|
| | | | |
| | | | |
| | | | |

Gambar 3.45 Tampilan Master Modul Rumah Sakit

k. Tampilan Master *User*

Tampilan master *user* berfungsi untuk menginputkan data user oleh sekretaris. Sekretaris terlebih dulu masuk pada menu master dan user untuk mengakses halaman master user. Gambar 3.46 menunjukkan tampilan dari master modul rumah sakit.

Admin LTE Sekretaris

Home > Master > User

NIK

Nama User

Jabatan +

Alamat

Kota +

Telepon

Password

Email

Gambar 3.46 Tampilan Master User

l. Tampilan Master *Onsite* RS

Tampilan master *onsite* RS berfungsi untuk menginputkan data *user* yang mempunyai jabatan sebagai *onsite* dan rumah sakit tempat dimana *onsite* ditempatkan. Sekretaris terlebih dulu masuk pada menu master dan *onsite* rs untuk mengakses halaman master *onsite* rs. Gambar 3.47 menunjukkan tampilan dari master modul rumah sakit.

The screenshot shows a web application interface for 'Admin LTE' with the user role 'Sekretaris'. The breadcrumb trail is 'Home > Master > Onsite RS'. The form contains two text input fields: 'Onsite' and 'Rumah Sakit', both with a placeholder 'Enter Text'. Below the inputs are 'Save' and 'Cancel' buttons. At the bottom, there is a table with two columns: 'Nama Onsite' and 'Rumah Sakit'. The table has three empty rows for data entry.

Gambar 3.47 Tampilan Master Onsite RS

m. Tampilan Master Status List *Complain*

Tampilan master list complain berfungsi untuk menginputkan data status dari list complain yang akan di kirim oleh pihak *onsite*. Sekretaris terlebih dulu masuk pada menu master dan status list *complain* untuk mengakses halaman master status *list complain*. Gambar 3.48 menunjukkan tampilan dari master status *list complain*.

The screenshot shows a web application interface for 'Admin LTE' with the user role 'Sekretaris'. The breadcrumb trail is 'Home > Master > Status List Complain'. The form contains two text input fields: 'ID Status List Complain' and 'Status List Complain', both with a placeholder 'Enter Text'. Below the inputs are 'Save' and 'Cancel' buttons. At the bottom, there is a table with two columns: 'ID Status List Complain' and 'Status List Complain'. The table has three empty rows for data entry.

Gambar 3.48 Tampilan Master Status List Complain

n. Tampilan Master Kategori *Complain*

Tampilan master kategori *complain* berfungsi untuk menginputkan data kategori *complain* oleh koordinator *sofdev*. Koordinator *sofdev* terlebih dulu masuk pada menu master dan kategori *complain* untuk mengakses halaman master kategori *complain*. Gambar 3.49 menunjukkan tampilan dari master kategori *complain*.

| ID Kategori | Kategori | Estimasi | Deadline |
|-------------|----------|----------|----------|
| | | | |
| | | | |

Gambar 3.49 Tampilan Master Kategori *Complain*

o. Tampilan Master Status Penjadwalan

Tampilan master status penjadwalan berfungsi untuk menginputkan data status penjadwalan oleh sekretaris. Sekretaris terlebih dulu masuk pada menu master dan status penjadwalan untuk mengakses halaman master status penjadwalan. Gambar 3.50 menunjukkan tampilan dari master status penjadwalan.

| ID Status Penjadwalan | Status Penjadwalan |
|-----------------------|--------------------|
| | |
| | |

Gambar 3.50 Tampilan Master Status Penjadwalan

p. Tampilan Master Status *Mission*

Tampilan master status *mission* berfungsi untuk menginputkan status mission oleh sekretaris. Sekretaris terlebih dulu masuk pada menu master dan status penjadwalan untuk mengakses halaman master status *mission*. Gambar 3.51 menunjukkan tampilan dari master status *mission*.

The screenshot shows a web application interface for 'Admin LTE' with the user role 'Sekretaris'. The breadcrumb trail is 'Home > Master > Status Mission'. The form contains two text input fields: 'ID Status Mission' and 'Status Mission', both with 'Enter Text' placeholder. Below the inputs are 'Save' and 'Cancel' buttons. At the bottom, there is a table with two columns: 'ID Status Mission' and 'Status Mission', with three empty rows.

Gambar 3.51 Tampilan Master Status *Mission*

q. Tampilan Transaksi Input Data Complain

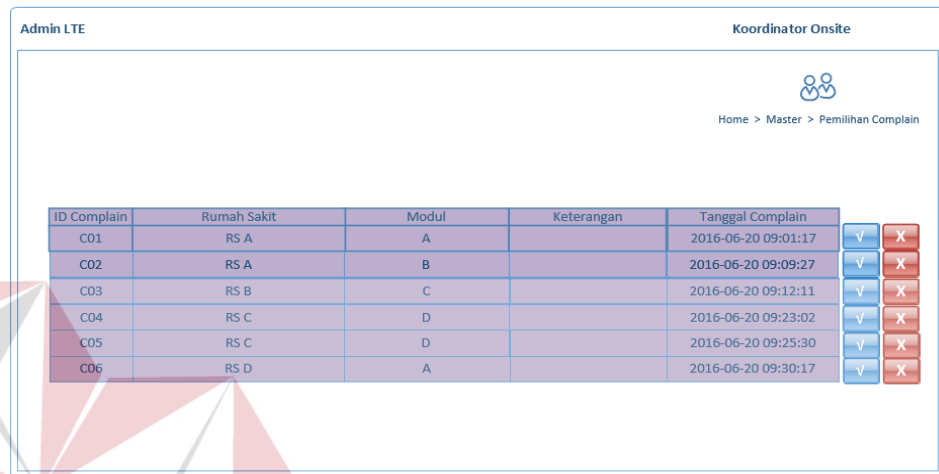
Tampilan transaksi input data complain berfungsi untuk menginputkan data complain yang di berikan oleh rumah sakit kepada pihak onsite. Pihak onsite masuk pada menu transaksi dan input data complain. Complain yang di masukan harus sesuai dengan modul dan keterangan *error*. Gambar 3.52 menunjukkan tampilan dari transaksi input data *complain*.

The screenshot shows a web application interface for 'Admin LTE' with the user role 'Onsite'. The breadcrumb trail is 'Home > Master > Kategori Complain'. The form contains five text input fields: 'ID Complain', 'Rumah Sakit', 'Modul', 'Keterangan', and 'Tanggal Complain', all with 'Enter Text' placeholder. Below the inputs are 'Save' and 'Cancel' buttons. At the bottom, there is a table with five columns: 'ID Complain', 'Rumah Sakit', 'Modul', 'Keterangan', and 'Tanggal Complain', with three empty rows.

Gambar 3.52 Tampilan Transaksi Input Data *Complain*

r. Tampilan Transaksi Pemilihan Complain

Tampilan transaksi pemilihan complain berfungsi untuk memilih complain yang di inputkan oleh sofdev. Proses ini dikerjakan oleh korrdinator *onsite*, complain di pilih berdasarkan keterangan complain. Gambar 3.53 menunjukkan tampilan dari transaksi pemilihan *complain*.

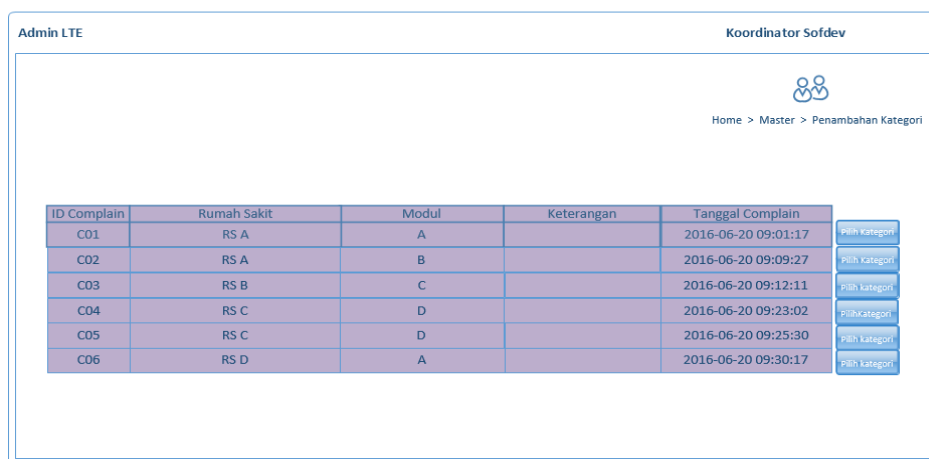


| ID Complain | Rumah Sakit | Modul | Keterangan | Tanggal Complain | | |
|-------------|-------------|-------|------------|---------------------|---|---|
| C01 | RS A | A | | 2016-06-20 09:01:17 | ✓ | X |
| C02 | RS A | B | | 2016-06-20 09:09:27 | ✓ | X |
| C03 | RS B | C | | 2016-06-20 09:12:11 | ✓ | X |
| C04 | RS C | D | | 2016-06-20 09:23:02 | ✓ | X |
| C05 | RS C | D | | 2016-06-20 09:25:30 | ✓ | X |
| C06 | RS D | A | | 2016-06-20 09:30:17 | ✓ | X |

Gambar 3.53 Tampilan Transaksi Pemilihan Complain

s. Tampilan Transaksi Penambahan Kategori

Tampilan transaksi penambahan kategori berfungsi untuk menambahkan kategori pada complain yang telah di plih oleh koordinator *onsite* dan dikirim ke koordinator *sofdev*. Gambar 3.54 menunjukkan tampilan dari transaksi penambahan kategori.

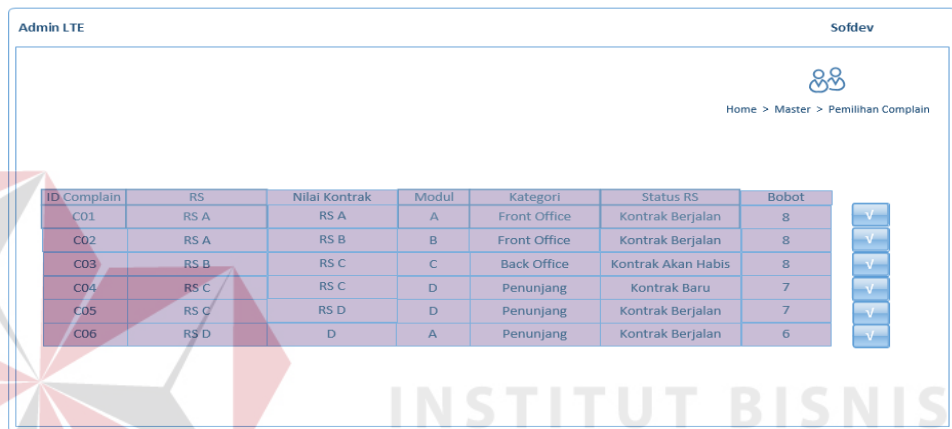


| ID Complain | Rumah Sakit | Modul | Keterangan | Tanggal Complain | |
|-------------|-------------|-------|------------|---------------------|----------------|
| C01 | RS A | A | | 2016-06-20 09:01:17 | Pilih Kategori |
| C02 | RS A | B | | 2016-06-20 09:09:27 | Pilih Kategori |
| C03 | RS B | C | | 2016-06-20 09:12:11 | Pilih Kategori |
| C04 | RS C | D | | 2016-06-20 09:23:02 | Pilih Kategori |
| C05 | RS C | D | | 2016-06-20 09:25:30 | Pilih Kategori |
| C06 | RS D | A | | 2016-06-20 09:30:17 | Pilih Kategori |

Gambar 3.54 Tampilan Transaksi Penambahan Kategori

t. Tampilan Transaksi Update Status Penjadwalan (*Sofdev*)

Tampilan transaksi update status penjadwalan untuk *sofdev* berfungsi untuk memilih complain yang akan dikerjakan. Complain yang tampil sudah terjadwal sesuai dengan bobot yang telah dihitung. *Sofdev* harus memilih complain yang paling atas untuk dikerjakan terlebih dahulu. Gambar 3.55 menunjukkan tampilan dari transaksi update status penjadwalan (*sofdev*).

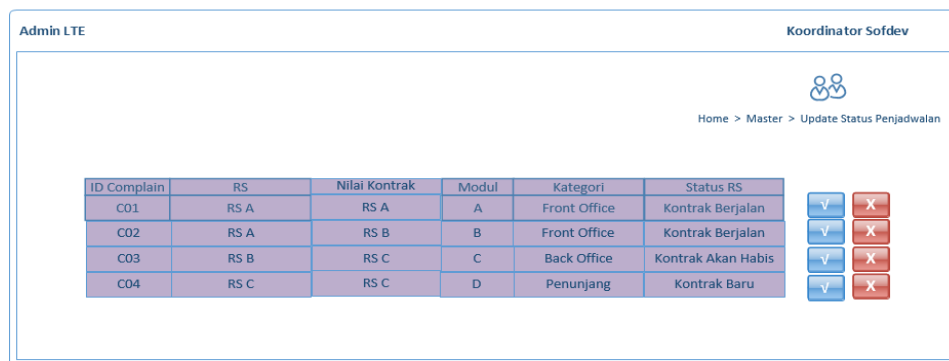


| ID Complain | RS | Nilai Kontrak | Modul | Kategori | Status RS | Bobot |
|-------------|------|---------------|-------|--------------|--------------------|-------|
| C01 | RS A | RS A | A | Front Office | Kontrak Berjalan | 8 |
| C02 | RS A | RS B | B | Front Office | Kontrak Berjalan | 8 |
| C03 | RS B | RS C | C | Back Office | Kontrak Akan Habis | 8 |
| C04 | RS C | RS C | D | Penunjang | Kontrak Baru | 7 |
| C05 | RS C | RS D | D | Penunjang | Kontrak Berjalan | 7 |
| C06 | RS D | D | A | Penunjang | Kontrak Berjalan | 6 |

Gambar 3.55 Tampilan Transaksi Update Status Penjadwalan (*sofdev*)

u. Tampilan Transaksi Update Status Penjadwalan (Koordinator *Sofdev*)

Tampilan transaksi update status penjadwalan untuk koordinator *sofdev* berfungsi untuk mengupdate status selesai (*done*) pada complain yang telah di kerjakan oleh *sofdev*. 3.56 menunjukkan tampilan dari transaksi update status penjadwalan (koordinator *sofdev*).

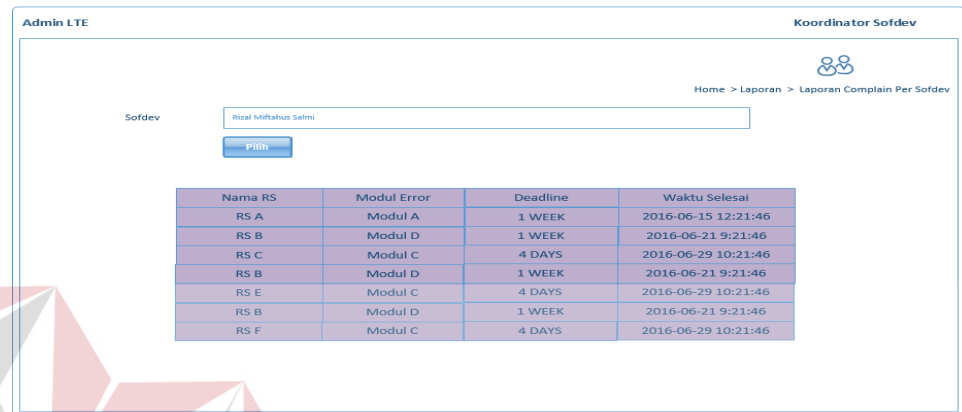


| ID Complain | RS | Nilai Kontrak | Modul | Kategori | Status RS |
|-------------|------|---------------|-------|--------------|--------------------|
| C01 | RS A | RS A | A | Front Office | Kontrak Berjalan |
| C02 | RS A | RS B | B | Front Office | Kontrak Berjalan |
| C03 | RS B | RS C | C | Back Office | Kontrak Akan Habis |
| C04 | RS C | RS C | D | Penunjang | Kontrak Baru |

Gambar 3.56 Tampilan Transaksi Update Status Penjadwalan (koordinator *sofdev*)

v. Laporan Complain Per *Sofdev*

Tampilan laporan *complain* per *sofdev* adalah sebuah tampilan yang berfungsi untuk menampilkan sebuah laporan yang berisi data complain yang sudah diselesaikan oleh sofdev. Gambar 3.57 menunjukkan tampilan dari laporan complain per *sofdev*.

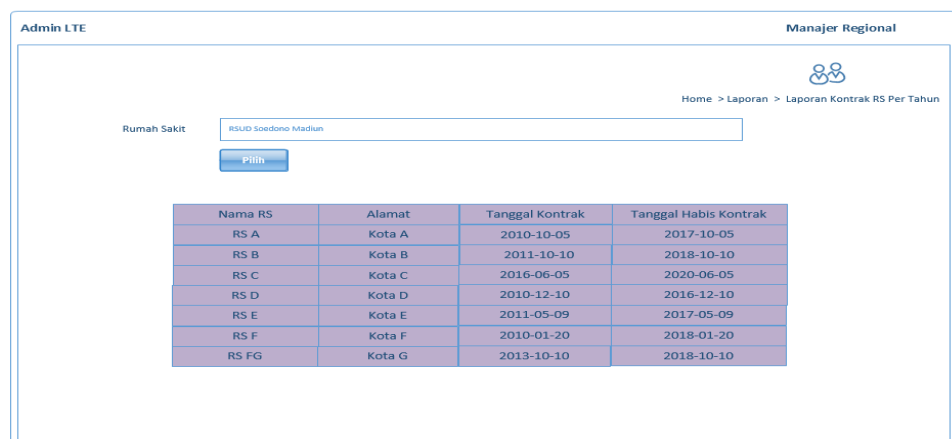


| Nama RS | Modul Error | Deadline | Waktu Selesai |
|---------|-------------|----------|---------------------|
| RS A | Modul A | 1 WEEK | 2016-06-15 12:21:46 |
| RS B | Modul D | 1 WEEK | 2016-06-21 9:21:46 |
| RS C | Modul C | 4 DAYS | 2016-06-29 10:21:46 |
| RS B | Modul D | 1 WEEK | 2016-06-21 9:21:46 |
| RS E | Modul C | 4 DAYS | 2016-06-29 10:21:46 |
| RS B | Modul D | 1 WEEK | 2016-06-21 9:21:46 |
| RS F | Modul C | 4 DAYS | 2016-06-29 10:21:46 |

Gambar 3.57 Tampilan Laporan Complain Per *Sofdev*

w. Laporan RS Berdasarkan Masa Kontrak Tahunan

Tampilan RS berdasarkan masa kontrak tahunan adalah sebuah tampilan yang berfungsi untuk menampilkan sebuah laporan yang berisi data rumah sakit berdasarkan tanggal kontrak dan tanggal habis kontrak. Gambar 3.58 menunjukkan tampilan dari laporan rumah sakit berdasarkan kontrak tahunan.



| Nama RS | Alamat | Tanggal Kontrak | Tanggal Habis Kontrak |
|---------|--------|-----------------|-----------------------|
| RS A | Kota A | 2010-10-05 | 2017-10-05 |
| RS B | Kota B | 2011-10-10 | 2018-10-10 |
| RS C | Kota C | 2016-06-05 | 2020-06-05 |
| RS D | Kota D | 2010-12-10 | 2016-12-10 |
| RS E | Kota E | 2011-05-09 | 2017-05-09 |
| RS F | Kota F | 2010-01-20 | 2018-01-20 |
| RS FG | Kota G | 2013-10-10 | 2018-10-10 |

Gambar 3.58 Tampilan Laporan RS Berdasarkan Masa Kontak Per Tahun

x. Laporan RS Berdasarkan Jumlah *Complain*

Tampilan RS berdasarkan jumlah *complain* pertahun adalah sebuah tampilan yang berfungsi untuk menampilkan sebuah laporan yang berisi data rumah sakit dan jumlah *complain* yang sudah diberikan. Gambar 3.59 menunjukkan tampilan dari laporan rumah sakit berdasarkan jumlah *complain* pertahun.



Gambar 3.59 Tampilan Laporan RS Berdasarkan Jumlah *Complain*

y. Laporan Modul Sering Terdapat *Complain*

Tampilan modul sering terdapat *complain* adalah sebuah tampilan yang berfungsi untuk menampilkan sebuah laporan yang berisi data modul yang sering terjadi *error* dan sering di *complain* kan oleh beberapa rumah sakit. Gambar 3.60 menunjukkan tampilan dari laporan modul sering terdapat *complain*.



Gambar 3.60 Tampilan Laporan Modul Sering Terdapat *Complain*

3.5.6 Perancangan Pengujian

Perancangan pengujian bertujuan untuk memastikan bahwa aplikasi telah dibuat dengan benar sesuai dengan kebutuhan spesifikasi dan tujuan yang diharapkan. Pengujian aplikasi yang dibuat ini menggunakan metode *black box*. Berikut ini adalah beberapa perancangan uji coba pada aplikasi penjadwalan penanganan *complain* yang akan digunakan untuk mengetahui hasil yang dari aplikasi yang diharapkan.

a. Perancangan Uji Coba Mengecek Menu *Login*

Pada tabel 3.34 adalah tabel tuser yang digunakan untuk proses uji coba mengecek menu login. Pengujian yang dilakukan saat menekan tombol *login*.



Tabel 3.34 Data Desain Uji Coba Mengecek Menu *Login*

| NIK | id_kota | id_jabatan | password | nama_user | alamat | tlp_user | email |
|---------|---------|------------|----------------------------------|------------------|-----------------|---------------|--|
| 987 | 3578 | 3 | 9e1e06ec8e02f0a0074f2fcc6b26303b | Aan | Madiun | 5654645647 | aan@gmail.com |
| 101010 | 3510 | 5 | 6d071901727aec1ba6d8e2497ef5b709 | Rizal Lare Osing | Jalan Penataran | (0333) 420861 | inbox@rizallaros.com |
| 12345 | 3502 | 2 | 827ccb0eea8a706c4c34a16891f84e7b | Angga | Ponorogo | 857123123 | anggoro_nci@gmail.com |
| 5020150 | 3578 | 6 | f35718098cc7f9f51813030136e36238 | Hardianto | Surabaya | 0818806606 | hardianto_nci@gmail.com |
| 909090 | 3578 | 5 | df780a97b7d6a8f779f14728bccd3c4c | Haidar | Surabaya | 031848234 | baadilla2000@gmail.com |

Tabel 3.35 Desain Uji Coba Mengecek Menu *Login*

| Desain Uji Coba Mengecek Menu Login | | |
|-------------------------------------|---|--|
| Skenario | Percobaan | Hasil yang diharapkan |
| 1 | Proses login dengan Kolom NIK dan Password kosong atau salah satu tidak di isi. | Menampilkan pesan bahwa kolom NIK dan Password harus di isi. |
| 2 | Proses login dengan Kolom NIK atau Password salah. | Menampilkan pesan bahwa NIK atau Password salah. |

b. Perancangan Uji Coba Menu Master Rumah Sakit

Setelah melakukan uji coba terhadap menu login selanjutnya akan dilakukan perancangan uji coba untuk menu master rumah sakit.

Tabel 3.36 Data Desain Uji Coba Menu Master Rumah Sakit

| ID RS | Rumah Sakit | Alamat | Kota | Telepon | Status RS | Tanggal Kontrak | Tanggal Habis Kontrak | Nilai Kontrak |
|-------|-----------------------|-----------------|----------|------------|--------------------|-----------------|-----------------------|---------------|
| 1 | RSUD Soedono Madiun | Madiun | Madiun | 928734982 | Kontrak Berjalan | 10/5/2010 | 10/5/2017 | 1500000000 |
| 2 | RSUD Harjono Ponorogo | Ponrogo | Ponorogo | 9238498234 | Kontrak Berjalan | 10/10/2011 | 10/10/2018 | 1000000000 |
| 3 | RSUD Bangil | Bangil-Pasuruan | Pasuruan | 283849812 | Kontrak Baru | 6/5/2016 | 6/5/2020 | 750000000 |
| 4 | RSUD Sidoarjo | Sidoarjo | Sidoarjo | 92384913 | Kontrak Akan Habis | 12/10/2010 | 12/10/2016 | 1000000000 |

Tabel 3.37 Desain Uji Coba Mengecek Menu Master Rumah Sakit

| Desain Uji Coba Menu Master Rumah Sakit | | |
|---|--|---|
| Skenario | Percobaan | Hasil yang diharapkan |
| 1 | <i>id rs generate</i> secara otomatis. Data kolom ID Rumah Sakit. | <i>Textbox ID Rumah Sakit</i> secara otomatis terisi dengan cara <i>autogenerate</i> . |
| 2 | Proses simpan dengan semua data pada kolom <i>form</i> master rumah sakit terisi | Data rumah sakit dapat tersimpan ke dalam <i>database</i> dan dapat ditampilkan pada aplikasi. |
| 3 | Proses simpan dengan kolom rumah sakit kosong | Aplikasi menampilkan pesan kolom rumah sakit harus diisi atau <i>alrert</i> pada kolom rumah sakit. |
| 4 | Proses simpan dengan kolom alamat kosong | Aplikasi menampilkan pesan kolom alamat harus diisi atau <i>alrert</i> pada kolom alamat. |
| 5 | Proses simpan dengan kolom kota kosong | Aplikasi menampilkan pesan kolom kota harus dipilih atau <i>alrert</i> pada |

| Desain Uji Coba Menu Master Rumah Sakit | | |
|---|---|---|
| Skenario | Percobaan | Hasil yang diharapkan |
| | | kolom kota. |
| 6 | Proses simpan dengan kolom telepon kosong | Aplikasi menampilkan pesan kolom telepon harus diisi atau <i>alrert</i> pada kolom telepon. |
| 7 | Proses simpan dengan kolom status rs kosong | Aplikasi menampilkan pesan kolom status rs harus dipilih atau <i>alrert</i> pada kolom status rs. |
| 8 | Proses simpan dengan kolom status rs kosong tanggal kontrak | Aplikasi menampilkan pesan kolom tanggal kontrak harus diisi atau <i>alrert</i> pada kolom tanggal kontrak. |
| 9 | Proses simpan dengan kolom status rs kosong tanggal habis kontrak | Aplikasi menampilkan pesan kolom tanggal habis kontrak harus diisi atau <i>alrert</i> pada kolom tanggal habis kontrak. |
| 10 | Proses simpan dengan kolom status rs kosong nilai kontrak | Aplikasi menampilkan pesan kolom nilai kontrak harus diisi atau <i>alrert</i> pada kolom nilai kontrak. |
| 11 | Proses delete pada salah satu data rumah sakit | Data rumah sakit berhasil dihapus dari database dan tidak akan tampil di dalam aplikasi. |

c. Perancangan Uji Coba Menampilkan Data Rumah Sakit

Setelah data rumah sakit berhasil di simpan, maka data rumah sakit akan tampil ke dalam aplikasi sebagai list data di bawah form input data rumah sakit.

Tabel 3.38 Data Desain Uji Coba Menampilkan Data Rumah Sakit

| ID RS | Rumah Sakit | Alamat | Kota | Telepon | Status RS | Tanggal Kontrak | Tanggal Habis Kontrak | Nilai Kontrak |
|-----------------------------------|-----------------------|-----------------|----------|------------|--------------------|-----------------|-----------------------|---------------|
| 1 | RSUD Soedono Madiun | Madiun | Madiun | 928734982 | Kontrak Berjalan | 10/5/2010 | 10/5/2017 | 1500000000 |
| 2 | RSUD Harjono Ponorogo | Ponrogo | Ponorogo | 9238498234 | Kontrak Berjalan | 10/10/2011 | 10/10/2018 | 1000000000 |
| 3 | RSUD Bangil | Bangil-Pasuruan | Pasuruan | 283849812 | Kontrak Baru | 6/5/2016 | 6/5/2020 | 750000000 |
| 4 | RSUD Sidoarjo | Sidoarjo | Sidoarjo | 92384913 | Kontrak Akan Habis | 12/10/2010 | 12/10/2016 | 1000000000 |
| Data yang di cari tidak ditemukan | | | | | | | | |

Tabel 3.39 Desain Uji Coba Menampilkan Data Rumah Sakit

| Desain Uji Coba Menampilkan Data Rumah Sakit | | |
|--|---|--|
| Skenario | Percobaan | Hasil yang diharapkan |
| 1 | Proses mencari data rumah sakit dengan mengetik “RSUD Soedono Madiun ” | Aplikasi menampilkan data rumah sakit dengan nama rumah sakit “RSUD Soedono Madiun” |
| 2 | Proses mencari data rumah sakit dengan mengetik “Ponorogo ” | Aplikasi akan menampilkan data rumah sakit dengan alamat “Ponorogo” |
| 3 | Proses mencari data rumah sakit dengan mengetik “Pasuruan” | Aplikasi akan menampilkan data rumah sakit dengan kota “pasuruhan” |
| 4 | Proses mencari data rumah sakit dengan mengetik “283849812” | Aplikasi akan menampilkan data rumah sakit dengan telepon “283849812” |
| 5 | Proses mencari data rumah sakit dengan mengetik “Kontrak Baru” | Aplikasi akan menampilkan data rumah sakit dengan status rumah sakit “Kontrak Baru” |
| 6 | Proses mencari data rumah sakit dengan mengetik “Kontrak Berjalan” | Aplikasi akan menampilkan data rumah sakit dengan status rumah sakit “Kontrak Berjalan” |
| 7 | Proses mencari data rumah sakit dengan mengetik “Kontrak Akan Habis” | Aplikasi akan menampilkan data rumah sakit dengan status rumah sakit “Akan Habis” |
| 8 | Proses mencari data rumah sakit dengan mengetik “12/10/2010” | Aplikasi akan menampilkan data rumah sakit dengan tanggal kontrak “12/10/2010” |
| 9 | Proses mencari data rumah sakit dengan mengetik “12/10/2016” | Aplikasi akan menampilkan data rumah sakit dengan tanggal habis kontrak kontrak “12/10/2016” |
| 10 | Proses mencari data rumah sakit dengan mengetik “750000000” | Aplikasi akan menampilkan data rumah sakit dengan nilai kontrak “750000000” |
| 11 | Proses mencari data rumah sakit dengan data yang salah atau belum tersimpan | Aplikasi menampilkan pesan bahwa data yang di cari tidak ditemukan |

d. Perancangan Uji Coba Menu Master Modul Rumah Sakit

Setelah melakukan uji coba terhadap menu master rumah sakit selanjutnya akan dilakukan perancangan uji coba untuk menu master modul rumah sakit.

Tabel 3.40 Data Desain Uji Coba Menu Master Modul Rumah Sakit

| ID Modul RS | Rumah Sakit | Modul | Tanggal Instal |
|-------------|-----------------------|--------------------|----------------|
| 1 | RSUD Soedono Madiun | Kasir Rawat Jalan | 2010-10-05 |
| 2 | RSUD Soedono Madiun | Kasir Rawat Inap | 2010-10-05 |
| 3 | RSUD Soedono Madiun | Kasir IGD | 2010-10-05 |
| 4 | RSUD Harjono Ponorogo | Kasir Rawat Jalan | 2011-10-10 |
| 5 | RSUD Harjono Ponorogo | Kasir Rawat Inap | 2011-10-10 |
| 6 | RSUD Harjono Ponorogo | Kasir IGD | 2011-10-10 |
| 7 | RSUD Harjono Ponorogo | OK (Kamar Operasi) | 2011-10-10 |

Tabel 3.41 Desain Uji Coba Menu Master Modul Rumah Sakit

| Desain Uji Coba Menu Master Modul Rumah Sakit | | |
|---|--|--|
| Skenario | Percobaan | Hasil yang diharapkan |
| 1 | <i>id modul rs generate</i> secara otomatis. | Saat menyimpan data modul rs id modul rs otomatis akan <i>autogenerate</i> |
| 2 | Proses <i>search</i> modul rs berdasarkan nama rumah sakit | Aplikasi akan menampilkan modul yang sudah di isi berdasarkan nama rumah sakit yang dipilih |
| 3 | Proses simpan dengan kolom tanggal instal kosong | Aplikasi menampilkan pesan kolom tanggal instal harus diisi atau <i>alert</i> pada kolom tanggal instal. |

| Desain Uji Coba Menu Master Modul Rumah Sakit | | |
|---|--|---|
| Skenario | Percobaan | Hasil yang diharapkan |
| 4 | Proses simpan dengan semua data pada kolom <i>form</i> master modul rumah sakit terisi | Data modul rumah sakit dapat tersimpan ke dalam <i>database</i> dan dapat ditampilkan pada aplikasi. |
| 5 | Proses delete pada salah satu data modul rumah sakit | Data modul rumah sakit berhasil dihapus dari <i>database</i> dan tidak akan tampil di dalam aplikasi. |

d. Perancangan Uji Coba Menu Menampilkan Data Modul Rumah Sakit

Setelah data modul rumah sakit berhasil di simpan, maka data modul rumah sakit akan tampil ke dalam aplikasi sebagai list data di bawa form input data modul rumah sakit.

Tabel 3.42 Data Uji Coba Menampilkan Data Modul Rumah Sakit

| ID Modul RS | Rumah Sakit | Modul | Tanggal Instal |
|-----------------------------------|-----------------------|--------------------|----------------|
| 1 | RSUD Soedono Madiun | Kasir Rawat Jalan | 2010-10-05 |
| 2 | RSUD Soedono Madiun | Kasir Rawat Inap | 2010-10-05 |
| 3 | RSUD Soedono Madiun | Kasir IGD | 2010-10-05 |
| 4 | RSUD Harjono Ponorogo | Kasir Rawat Jalan | 2011-10-10 |
| 5 | RSUD Harjono Ponorogo | Kasir Rawat Inap | 2011-10-10 |
| 6 | RSUD Harjono Ponorogo | Kasir IGD | 2011-10-10 |
| 7 | RSUD Harjono Ponorogo | OK (Kamar Operasi) | 2011-10-10 |
| Data yang di cari tidak ditemukan | | | |

Tabel 3.43 Desain Uji Coba Menampilkan Data Modul Rumah Sakit

| Desain Uji Coba Menampilkan Data Modul Rumah Sakit | | |
|--|--|---|
| Skenario | Percobaan | Hasil yang diharapkan |
| 1 | Proses mencari data modul rumah sakit dengan mengetik “RSUD Soedono Madiun ” | Aplikasi menampilkan data modul rumah sakit dengan nama rumah sakit “RSUD Soedono Madiun” |
| 2 | Proses mencari data rumah sakit dengan mengetik “Kasir Rawat Inap ” | Aplikasi menampilkan data modul rumah sakit dengan nama modul “Kasir Rawat Inap” |
| 3 | Proses mencari data rumah sakit dengan mengetik “2011-10-10” | Aplikasi menampilkan data modul rumah sakit dengan tanggal instal “2011-10-10” |
| 4 | Proses mencari data rumah sakit dengan data yang salah atau belum tersimpan | Aplikasi menampilkan pesan bahwa data yang di cari tidak ditemukan |

e. Perancangan Uji Coba Input Data Complain per *onsite*

Perancangan uji coba selanjutnya adalah perancangan pada *form* input data complain. Data uji coba untuk input data complain dapat di lihat pada tabel 3.34

Tabel 3.44 Data Uji Coba Input Data Complain Per Onsite

| ID Complain | Rumah Sakit | Modul | Keterangan | Tanggal Complain |
|----------------|-----------------------------|----------------------|-------------------------------------|-----------------------------|
| E9QypDIU Tw | RSUD Harjono Ponorogo | Kasir Rawat Jalan | Complain Modul kasir Rawat Jalan | 28 June 2016 09:00:49 AM |
| NPURtEE3 Ah | RSUD Harjono Ponorogo | Kasir Rawat Inap | Complain Modul Kasir Rawat Inap | 28 June 2016 03:54:25 PM |
| G9PypFIU Ti | RSUD Harjono Ponorogo | Kasir IGD | Complain Modul Kasir IGD | 28 June 2016 04:10:25 PM |

Tabel 3.45 Desain Uji Coba Input Data Complain Per Onsite

| Desain Uji Coba Input Data Complain Per Onsite | | |
|--|--|---|
| Skenario | Percobaan | Hasil yang diharapkan |
| 1 | <i>id complain rs autogenerate random character</i> secara otomatis. | Pada <i>textbox</i> ID complain akan <i>autogenerate random character</i> secara otomatis. Dan setiap form akan berbeda ID meskipun waktu penginputan complain bersamaan. |
| 2 | Kolom rumah sakit terisi otomatis dengan data rumah sakit tempat onsite ditempatkan. | Pada <i>textbox</i> rumah sakit akan terisi dengan nama rumah sakit sesuai dengan data onsite yang sudah di isi pada database. |
| 3 | Kolom tanggal complain terisi dengan tanggal komputer | Pada <i>textbox</i> tanggal complain akan otomatis terisi dengan tanggal pada komputer. |
| 4 | Simpan data dengan kolom keterangan kosong | Aplikasi menampilkan pesan kolom keterangan harus diisi atau <i>alrert</i> pada kolom keterangan. |

f. Perancangan Uji Coba Pemilihan Complain

Perancangan uji coba selanjutnya setelah perancangan uji coba input data complain adalah perancangan uji coba pada *form* pemilihan *complain*. Data uji coba untuk pemilihan complain dapat di lihat pada tabel 3.46

Tabel 3.46 Data Uji Coba Pemilihan Complain

| ID Complain | Rumah Sakit | Modul | Keterangan | Tanggal Complain |
|----------------|--------------------------|----------------------|--|-----------------------------|
| E9QypDI UTw | RSUD Harjono Ponorogo | Kasir Rawat Jalan | Complain Modul kasir Rawat Jalan | 28 June 2016 09:00:49 AM |
| NPURtEE 3Ah | RSUD Harjono Ponorogo | Kasir Rawat Inap | Complain Modul Kasir Rawat Inap | 28 June 2016 03:54:25 PM |
| G9PypFI UTi | RSUD Harjono Ponorogo | Kasir IGD | Complain Modul Kasir IGD | 28 June 2016 04:10:25 PM |

| ID Complain | Rumah Sakit | Modul | Keterangan | Tanggal Complain |
|----------------|------------------------|---------------------|---|-----------------------------|
| O6PtpFIU Ri | RSUD Soedono Madiun | Bank Darah | Complain Modul Bank Darah | 28 June 2016 09:10:49 AM |
| B0PypFI WTi | RSUD Soedono Madiun | Kasir Rawat Inap | Complain Modul Kasir Rawat Inap | 28 June 2016 09:07:39 AM |
| S1PypSIU T8 | RSUD Sidoarjo | Medical Record | Complain Modul Medical Record | 28 June 2016 09:27:19 AM |
| J9FyhFIU Tr | RSUD Bangil | General Cashier | Complain Modul General Cashier | 28 June 2016 09:22:32 AM |

Tabel 3.47 Desain Uji Coba Pemilihan Complain

| Desain Uji Coba Pemilihan Complain | | |
|------------------------------------|--|---|
| Skenario | Percobaan | Hasil yang diharapkan |
| 1 | Data <i>complain onsite</i> tampil | Data yang tampil pada menu pemilihan complain adalah data complain yang di inputkan seluruh onsite |
| 2 | Data tampil disertai dengan tombol <i>approve</i> dan <i>reject</i> | Ketika data tampil akan disertai dengan tampilnya dua tombol dengan fungsi <i>approve complain</i> dan <i>reject complain</i> . |
| 3 | Complain hilang dari list complain setelah tombol <i>approve</i> atau <i>reject</i> di tekan | Setelah tombol <i>approve</i> dan <i>reject</i> di tekan maka complain yang di pilih akan hilang dari list complain. |

g. Perancangan Uji Coba Penambahan Kategori

Perancangan uji coba selanjutnya setelah perancangan uji coba pemilihan *complain* adalah perancangan uji coba pada *form* penambahan kategori. Data uji coba untuk penambahan kategori dapat di lihat pada tabel 3.48

Tabel 3.48 Data Uji Coba Penambahan Kategori

| ID Complain | Rumah Sakit | Modul | Keterangan | Tanggal Complain |
|----------------|--------------------------|----------------------|-------------------------------------|-----------------------------|
| E9QypDI UTw | RSUD Harjono Ponorogo | Kasir Rawat Jalan | Complain Modul kasir Rawat Jalan | 28 June 2016 09:00:49 AM |
| NPURtEE 3Ah | RSUD Harjono Ponorogo | Kasir Rawat Inap | Complain Modul Kasir Rawat Inap | 28 June 2016 03:54:25 PM |
| G9PypFI UTi | RSUD Harjono Ponorogo | Kasir IGD | Complain Modul Kasir IGD | 28 June 2016 04:10:25 PM |
| O6PtpFIU Ri | RSUD Soedono Madiun | Bank Darah | Complain Modul Bank Darah | 28 June 2016 09:10:49 AM |
| B0PypFI WTi | RSUD Soedono Madiun | Kasir Rawat Inap | Complain Modul Kasir Rawar Inap | 28 June 2016 09:07:39 AM |
| S1PypSIU T8 | RSUD Sidoarjo | Medical Record | Complain Modul Medical Record | 28 June 2016 09:27:19 AM |
| J9FyhFIU Tr | RSUD Bangil | General Cashier | Complain Modul General Cashier | 28 June 2016 09:22:32 AM |

Tabel 3.49 Desain Uji Coba Penambahan Kategori

| Desain Uji Coba Penambahan Kategori | | |
|-------------------------------------|---|---|
| Skenario | Percobaan | Hasil yang diharapkan |
| 1 | Data complain tampil | Data complain yang tampil pada form ini adalah data complain yang disetujui oleh koordinator onsite melalui pemilihan <i>complain</i> . |
| 2 | Data tampil disertai dengan tombol tambah kategori | Data yang tampil masing-masing disertai dengan tombol tambah kategori. |
| 3 | Setelah menekan tombol tambah kategori akan tampil list kategori complain yang akan di tambahkan. | Aplikasi akan menampilkan tampilan untuk memilih kategori complain yang sesuai dengan keterangan complain. |
| 4 | Complain akan hilang dari list sesudah kategori ditambahkan | Complain akan hilang dari tampilan aplikasi setelah kategori di tambahkan. |

h. Perancangan Uji Coba Update Status Penjadwalan (*Sofdev*)

Perancangan uji coba selanjutnya setelah perancangan uji coba penambahan kategori adalah perancangan uji coba pada *form* update status penjadwalan (*Sofdev*). Data uji coba untuk update status penjadwalan (*Sofdev*) dapat di lihat pada tabel 3.50

Tabel 3.50 Data Uji Coba Update Status Penjadwalan (*Sofdev*)

| Tanggal | RS | Nilai Kontrak | Modul | Kategori | Status RS | Bobot |
|-----------------------------|-----------------------|---------------|--------------------|--------------|--------------------|-------|
| 28 June 2016 03:54:25 AM | RSUD Harjono Ponorogo | 1000000000 | Kasir Rawat Inap | Front Office | Kontrak Berjalan | 8 |
| 28 June 2016 09:00:49 AM | RSUD Harjono Ponorogo | 1000000000 | Kasir IGD | Front Office | Kontrak Berjalan | 8 |
| 28 June 2016 09:01:17 AM | RSUD Sidoarjo | 1000000000 | Medical Record | Back Office | Kontrak Akan Habis | 8 |
| 28 June 2016 09:01:43 AM | RSUD Bangil | 750000000 | OK (Kamar Operasi) | Penunjang | Kontrak Baru | 8 |
| 28 June 2016 09:02:15 AM | RSUD Soedono Madiun | 1500000000 | Bank Darah | Penunjang | Kontrak Berjalan | 8 |

Tabel 3.51 Desain Uji Coba Update Status Penjadwalan (*Sofdev*)

| Desain Uji Coba Update Status Penjadwalan (<i>Sofdev</i>) | | |
|---|----------------------|---|
| Skenario | Percobaan | Hasil yang diharapkan |
| 1 | Data complain tampil | Data complain yang tampil pada form ini adalah data complain sudah terjadwal, data yang dipilih oleh <i>sofdev</i> harus data yang paling atas terlebih |

| Desain Uji Coba Update Status Penjadwalan (<i>Sofdev</i>) | | |
|---|--|---|
| Skenario | Percobaan | Hasil yang diharapkan |
| | | dahulu. |
| 2 | Data tampil disertai dengan tombol pilih complain | Data yang tampil masing-masing disertai dengan tombol pilih <i>complain</i> |
| 3 | Jadwal akan tampil setelah tombol pilih complain di tekan. | Aplikasi akan menampilkan jadwal penanganan complain setelah <i>sofdev</i> memilih complain yang akan dikerjakan. |
| 4 | Complain akan hilang dari list sesudah menekan tombol pilih complain | Complain akan hilang dari tampilan aplikasi setelah <i>sofdev</i> memilih complain yang akan dikerjakan. |
| 5 | Tampil notifikas bila complain yang dikerjakan melebihi deadline | Aplikasi akan menampilkan notifikasi pada jadwal apabila complain yang dikerjakan melebihi deadline. |

i. Perancangan Uji Coba Update Status Penjadwalan (Koordinator *Sofdev*)

Perancangan uji coba selanjutnya setelah update status penjadwalan (*Sofdev*) adalah perancangan uji coba pada *form* update status penjadwalan (Koordinator *Sofdev*). Data uji coba untuk update status penjadwalan (Koordinator *Sofdev*) dapat di lihat pada tabel 3.52

Tabel 3.52 Data Uji Coba Update Status Penjadwalan (Koordinator *Sofdev*)

| Tanggal | RS | Nilai Kontrak | Modul | Kategori | Status RS |
|-----------------------------|-----------------------|---------------|------------------|--------------|--------------------|
| 28 June 2016 03:54:25 AM | RSUD Harjono Ponorogo | 1000000000 | Kasir Rawat Inap | Front Office | Kontrak Berjalan |
| 28 June 2016 09:00:49 AM | RSUD Harjono Ponorogo | 1000000000 | Kasir IGD | Front Office | Kontrak Berjalan |
| 28 June 2016 09:01:17 AM | RSUD Sidoarjo | 1000000000 | Medical Record | Back Office | Kontrak Akan Habis |

| Tanggal | RS | Nilai Kontrak | Modul | Kategori | Status RS |
|-----------------------------|---------------------|---------------|--------------------|-----------|------------------|
| 28 June 2016 09:01:43 AM | RSUD Bangil | 750000000 | OK (Kamar Operasi) | Penunjang | Kontrak Baru |
| 28 June 2016 09:02:15 AM | RSUD Soedono Madiun | 1500000000 | Bank Darah | Penunjang | Kontrak Berjalan |

Tabel 3.53 Desain Uji Coba Update Status Penjadwalan (Koordinator *Sofdev*)

| Desain Uji Coba Update Status Penjadwalan (Koordinator <i>Sofdev</i>) | | |
|--|--|--|
| Skenario | Percobaan | Hasil yang diharapkan |
| 1 | Data complain tampil | Data complain yang tampil pada form ini adalah data complain sudah diselesaikan oleh <i>sofdev</i> |
| 2 | Data tampil disertai dengan tombol selesai | Karena yang menentukan complain selesai adalah coordinator <i>sofdev</i> maka koordinator <i>sofdev</i> akan mengecek <i>complain</i> sekali lagi jika <i>complain</i> sudah benar-benar selesai maka koordinator <i>sofdev</i> akan menekan tombol selesai. |
| 3 | Complain akan hilang dari list sesudah menekan tombol selesai. | Complain akan hilang dari tampilan aplikasi setelah coordinator <i>sofdev</i> selesai. |