

## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1 Analisis Sistem

Tahap analisis sistem digunakan untuk melakukan kajian terhadap permasalahan yang terjadi pada sistem Paris (*parking information system*) dan kebutuhan-kebutuhan yang diharapkan sehingga dapat dibuat usulan perbaikan atau pengembangan.

##### 3.1.1 Identifikasi Masalah

Tahapan analisis dimulai dengan identifikasi masalah dengan cara melakukan observasi untuk meninjau langsung sistem yang berada di Stikom Surabaya. Langkah ini dilakukan untuk memahami sistem terlebih dahulu sebelum kemudian melakukan wawancara. Dimulai dari mendatangi pengembang sistem yaitu SSI (Solusi Sistem Informasi) kemudian melihat aplikasi baik aplikasi *desktop* ataupun *website* dan juga cara kerja perangkat keras yang digunakan. Terakhir yang perlu diketahui adalah SOP (Standar Operasional Prosedur) parkir yang diterapkan di Stikom agar lebih memahami jalan sistem.

Langkah selanjutnya adalah wawancara untuk mendapatkan informasi tentang kekurangan ataupun kendala yang saat ini belum dapat diselesaikan. Wawancara dilakukan secara informal berdialog dengan staf SSI di ruang kerja. Dari wawancara ini SSI mengungkapkan rencana pengembangan Paris sebagai produk massal dan mengutarakan kekurangan yang belum terdapat pada sistem.

Kekurangan Paris antara lain adalah jika Paris memiliki banyak klien dan juga setiap klien memiliki banyak lokasi parkir yang berjauhan sehingga tidak mungkin dihubungkan dengan menggunakan kabel maka aplikasi Paris tidak bisa disatukan sehingga perlu dibuat satu aplikasi Paris pada setiap lokasi parkirnya. Kemudian dari kekurangan tersebut muncul permasalahan baru yaitu sistem Paris yang terpisah akan menyulitkan klien ataupun SSI dalam melakukan pengaturan, menambah data, ataupun membuat laporan parkir.

Sistem Paris yang diterapkan di Stikom Surabaya memiliki sistem *membership* atau keanggotaan atau bisa disebut parkir berlangganan. Bagi mahasiswa ataupun karyawan Stikom yang parkir di lokasi Stikom tidak dikenakan biaya, namun untuk pihak luar akan dikenakan biaya parkir. Keanggotaan parkir ini sudah dapat ditangani oleh sistem Paris hanya saja, jika kemudian Paris dikembangkan menjadi multi lokasi untuk setiap klien maka data *member* seharusnya juga ikut tersebar ke semua lokasi parkir Paris.

Bagi pengguna parkir memiliki keanggotaan parkir memiliki keuntungan dibandingkan bukan anggota, yaitu biaya parkir yang lebih murah bahkan gratis. Dan jika saja pengguna parkir bisa mengetahui ketersediaan *slot* parkir di lokasi yang akan dikunjunginya maka itu akan sangat membantu pengguna dalam memutuskan pergi ke lokasi atau menunda hingga terlihat ketersediaan *slot* parkir lebih banyak.

### **3.1.2 Kebutuhan Sistem**

Untuk mempermudah melakukan analisis kebutuhan sistem, diperlukan pengelompokkan kebutuhan yaitu menjadi kebutuhan fungsional dan *non*

fungsional. Kebutuhan fungsional adalah kebutuhan proses-proses untuk menyelesaikan masalah yang telah diidentifikasi sebelumnya. Sedangkan kebutuhan *non* fungsional adalah kebutuhan-kebutuhan tambahan untuk mendukung terwujudnya sistem.

#### A. Kebutuhan Fungsional

Kebutuhan-kebutuhan untuk membuat Paris Manajemen Klien dan kebutuhan fungsional Paris Manajemen Klien dijelaskan pada tabel 3.1 Kebutuhan Pengguna, dan tabel 3.2 Kebutuhan Fungsional.

Tabel 3. 1 Kebutuhan Pengguna

No.	Kebutuhan Pengguna
KP.1	Membuat aplikasi yang dapat mengelola Paris milik klien
KP.2	Membuat aplikasi yang dapat mengelola lokasi parkir milik Klien
KP.3	Membuat aplikasi android yang dapat menginformasikan ketersediaan <i>slot</i> parkir di lokasi Paris

Tabel 3. 2 Kebutuhan Fungsional

No.	Kebutuhan Fungsional	No. Kebutuhan Pengguna
KF.1	Membuat master klien	KP.1
KF.2	Membuat master lokasi parkir	KP.2
KF.3	Menjalankan Paris <i>online</i>	KP.2
KF.4	Membuat master <i>slot</i> parkir	KP.3
KF.7	Membuat master <i>user</i> Paris Manajemen Klien	KP.1

KF.8	Sinkronisasi <i>database</i>	KP.2
KF.9	Sinkronisasi data <i>slot</i> parkir	KP.3
KF.10	Mendapatkan Informasi ketersediaan <i>slot</i> parkir di suatu lokasi	KP.3

## B. Kebutuhan Non Fungsional

Kebutuhan *Non Fungsional* atau kebutuhan lain yang tidak termasuk fungsi atau proses adalah sebagai berikut:

### 1. Karakteristik Sistem

- a. Membatasi hak akses aplikasi dengan fungsi *login*
- b. Sinkronisasi *database* bisa berjalan sendiri tanpa ada pengguna yang menjalankan
- c. Laporan disajikan dalam bentuk grafik

### 2. Perangkat lunak

- a. Menggunakan minimal Windows 7
- b. XAMPP 5.6.3
  - i. PHP 5.6
  - ii. MySQL 5.0
  - iii. Filezilla *server* 0.9.41
- c. *Browser* Google Chrome atau Mozilla Firefox

### 3. Kebutuhan Perangkat Keras

- a. Kebutuhan RAM komputer *server* minimal 1GB
- b. Kebutuhan *Storage* 1GB
- c. Jaringan Internet

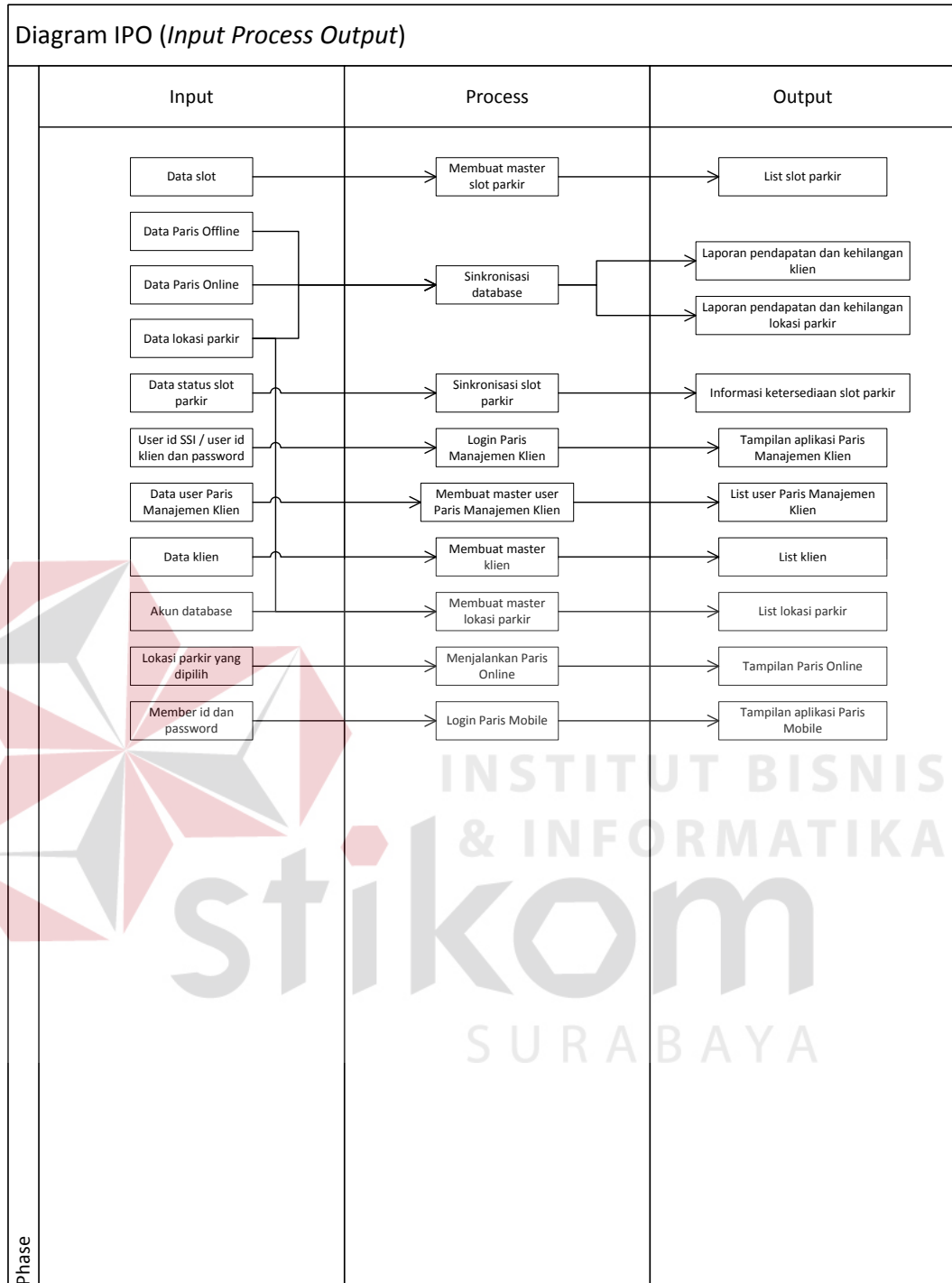
### 3.2 Perancangan Sistem

Berdasarkan hasil analisis, selanjutnya dibuat perancangan sistem dengan tujuan membuat gambaran bagaimana solusi sistem akan dibangun dan berjalan. Perancangan sistem diharapkan mampu menyelesaikan permasalahan yang terjadi yaitu membuat aplikasi yang dapat mengelola klien dan banyak lokasi parkir, membuat aplikasi android untuk memberikan informasi ketersediaan *slot* parkir kepada *member* parkir.

Aplikasi yang akan dibangun ini selanjutnya akan disebut PMK (Paris Manajemen Klien). PMK akan dibuat berbasis *web* dengan menggunakan bahasa pemrograman HTML, PHP, Javascript, dan MySQL juga menggunakan *framework* CodeIgniter. Adapun beberapa tahapan yang perlu dilakukan yaitu pembuatan diagram IPO (*Input Process Output*), *System Flow*, diagram berjenjang, DFD (*Data Flow Diagram*), ERD (*Entity Relationship Diagram*), struktur *database*, dan desain *input output*.

#### 3.2.1 Diagram IPO (*Input Process Output*)

Diagram IPO dibuat untuk membuat gambaran data apa saja yang dimasukkan sistem, tahapan-tahapan yang dilakukan sistem serta apa saja hasil dari aplikasi PMK.



Gambar 3. 1 Diagram IPO (Input Process Output)

Berikut adalah penjelasan dari diagram IPO berdasarkan gambar 3.1

Diagram IPO:

**a. Input**

Data-data yang diperlukan sebagai masukan aplikasi PMK adalah:

1. *Data Slot*

Merupakan pengelompokan suatu area parkir di lokasi parkir Paris yang paling kecil (misalnya area parkir *basement* baris A1, *basement* baris A2, dll.).

2. *Data Paris Offline*

Merupakan data yang berasal dari *database* di lokasi parkir untuk *update database online*. Data yang diupdate dari *database offline* adalah data transaksi parkir, catatan kehilangan, data *gate* parkir, *counter* tiket, histori *reset* tiket, dan data *log* aplikasi Paris.

3. *Data Paris Online*

Merupakan data yang berasal dari *database mirror* milik lokasi parkir, *database* ini yang akan digunakan untuk aplikasi Paris *Parking Slot Mobile* melihat ketersediaan *slot* parkir di semua lokasi parkir. Data ini juga digunakan untuk melakukan *update* data *master* di *database* lokasi parkir (contoh: data *member*, biaya parkir, *user*/petugas parkir, dll.).

4. *Data Lokasi Parkir*

Merupakan data pembeda untuk masuk ke *database Paris online*. Di dalam data lokasi parkir terdapat akun *database* dari setiap Paris *online* (nama *database*, *user database*, dan *password database*).

5. Data status *slot* parkir

Adalah data status apakah masing-masing *slot* parkir sedang dalam keadaan terisi oleh kendaraan atau tidak.

6. Data *user id* SSI / klien dan *password*

Data *user id* dan *password* adalah identitas yang akan digunakan baik oleh SSI atau klien untuk masuk ke aplikasi Paris Manajemen Klien.

7. Data *user* Paris Manajemen Klien

Adalah data pengguna Paris Manajemen Klien. Pengguna di sini adalah SSI yang akan mengelola Paris Manajemen Klien.

8. Data Klien

Merupakan data identitas unik setiap klien Paris, Id klien nantinya akan menjadi kunci mencari lokasi parkir mana saja yang termasuk dalam satu *group* parkir.

9. Akun *database*

Merupakan data masing-masing *database Paris online*. Data ini akan digunakan untuk bisa masuk ke masing-masing *database* dari aplikasi Paris Manajemen Klien.

10. Lokasi Parkir yang dipilih

Merupakan id lokasi parkir .yang dipilih *user* untuk menjalankan aplikasi Paris Manajemen Klien.



### 11. *Member id dan password*

Adalah identitas *member* untuk menggunakan aplikasi Paris *Mobile*.



### ***b. Process***

#### **1. Membuat master *slot* parkir**

Adalah proses menambah, merubah, dan menghapus data *slot* parkir di lokasi parkir Paris.

#### **2. Sinkronisasi *database***

Adalah proses mengirim data transaksi dan kehilangan dari *database* Paris *offline* untuk di *update* di *database* Paris *online* dan selanjutnya mengambil data master dan pengaturan dari *database* Paris *online* kemudian di *update* di Paris *offline*. Proses sinkronisasi *database* akan dilakukan secara otomatis dengan melihat pengaturan jam sinkronisasi.

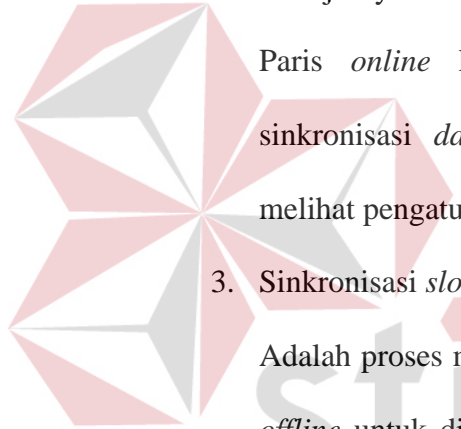
#### **3. Sinkronisasi *slot* parkir**

Adalah proses mengirim data status *slot* parkir dari *database* Paris *offline* untuk di *update* di *database* Paris *online*. Proses ini akan dilakukan secara otomatis dengan melihat pengaturan interval waktu sinkronisasi.

#### **4. Login Paris Manajemen Klien**

Adalah proses untuk masuk dan menggunakan aplikasi Paris Manajemen Klien. Proses ini dilakukan oleh *user* SSI dan klien dengan cara memasukkan *user* id dan *password* dan akan divalidasi oleh sistem apakah *user* tersebut berhak atau tidak.

#### **5. Membuat master *user* Paris Manajemen Klien**



Adalah proses untuk menambah, merubah, dan menghapus data *user* SSI yang akan bisa menggunakan aplikasi Paris Manajemen Klien.

6. Membuat master klien

Adalah proses untuk menambah, merubah, dan menghapus data klien Paris.

7. Membuat master lokasi parkir

Adalah proses untuk menambah, merubah, dan menghapus data lokasi parkir milik klien Paris.

8. Menjalankan aplikasi Paris *online*

Adalah proses untuk masuk dan menjalankan aplikasi Paris *online* milik klien. Setelah masuk Paris *online*, SSI ataupun klien dapat melakukan perubahan data master atau pengaturan.

9. *Login Paris Mobile*

Adalah proses untuk menggunakan aplikasi Paris *mobile* yang dapat memberikan informasi ketersediaan *slot* parkir di semua lokasi parkir milik klien tempat terdaftarnya *member* tersebut.

**c. Output**

1. *List slot* parkir

Adalah *list* atau daftar *slot* parkir dan id perangkat yang terpasang di parkir milik klien yang sebelumnya sudah disimpan.

2. Laporan pendapatan dan kehilangan klien

Adalah laporan pendapatan dan kehilangan masing-masing klien yang didapat dari sinkronisasi *database*.

3. Laporan pendapatan dan kehilangan lokasi parkir

Adalah laporan pendapatan dan kehilangan masing-masing lokasi parkir milik klien yang didapat dari sinkronisasi *database*.

4. Informasi ketersediaan slot parkir

Adalah informasi jumlah *slot* parkir yang tersedia atau belum terisi kendaraan di semua lokasi parkir milik klien.

5. Tampilan aplikasi Paris Manajemen Klien

Adalah halaman utama dari Paris Manajemen Klien. Dari halaman ini, *user* dapat berpindah ke menu-menu yang dimiliki Paris Manajemen Klien.

6. *List user* Paris Manajemen Klien

Adalah *list* atau daftar *user* SSI yang sebelumnya sudah pernah tersimpan di sistem.

7. *List* klien

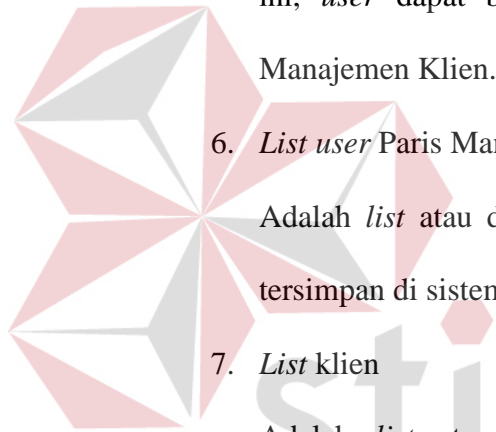
Adalah *list* atau daftar klien Paris yang sebelumnya sudah tersimpan di sistem.

8. *List* lokasi parkir

Adalah *list* atau daftar lokasi parkir dan akun *database* Paris *online* masing-masing lokasi yang sebelumnya sudah pernah tersimpan di sistem.

9. Tampilan Paris *online*

Adalah halaman utama aplikasi Paris *online*. Melalui halaman ini, *user* dapat masuk ke menu-menu lainnya untuk melakukan tambah, ubah, ataupun hapus data master dan pengaturan.



INSTITUT BISNIS  
& INFORMATIKA  
stikom  
SURABAYA

#### 10. Tampilan aplikasi Paris *Mobile*

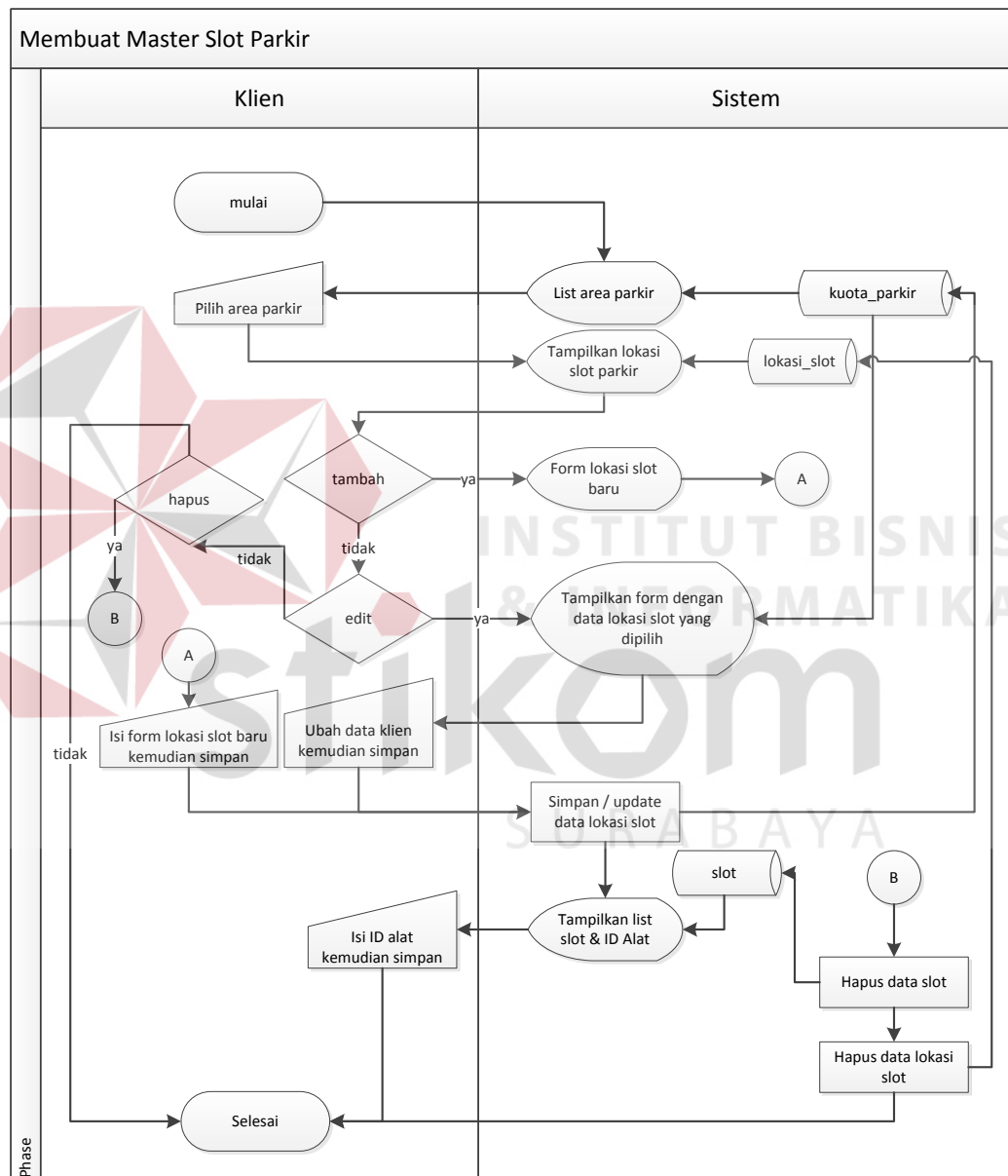
Adalah halaman utama aplikasi Paris *mobile* yang berisi informasi ketersediaan *slot* parkir di semua lokasi parkir milik klien.



### 3.2.2 System Flow

#### A. Paris Offline

##### 1. Membuat Master Slot parkir



Gambar 3. 2 System Flow Membuat Master Slot Parkir

Tabel 3. 3 Penjelasan *System Flow* Membuat Master *Slot* Parkir

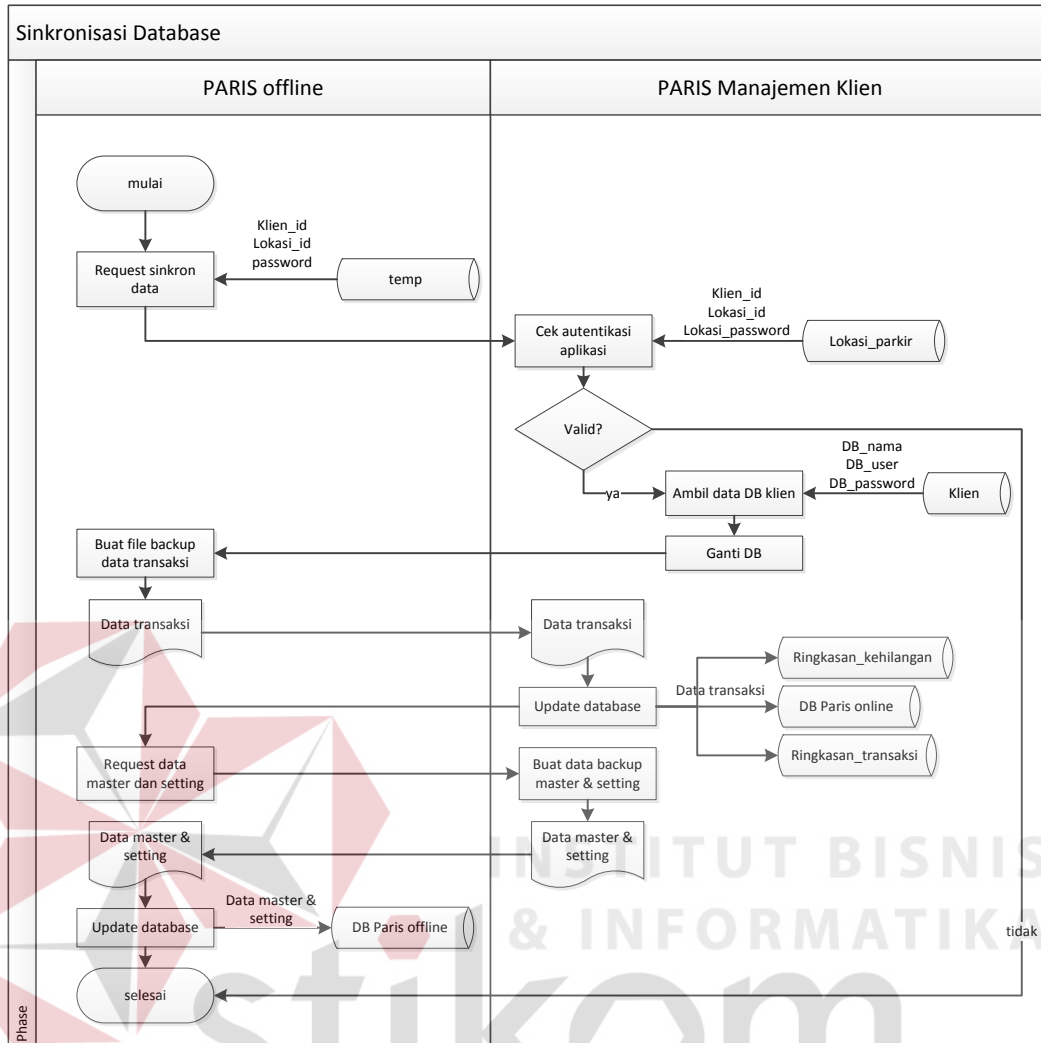
Membuat Master <i>Slot</i> Parkir	
Tujuan	Untuk menambah, merubah, dan menghapus <i>slot</i> parkir di suatu lokasi parkir klien
Masukan	<ul style="list-style-type: none"> <li>- Nama lokasi <i>slot</i></li> <li>- Jumlah <i>slot</i></li> <li>- Id perangkat</li> </ul>
Proses	<p>Tambah</p> <ol style="list-style-type: none"> <li>1. Pilih area parkir yang ingin ditambah lokasi <i>slot</i> nya</li> <li>2. Akan tampil <i>list</i> lokasi <i>slot</i> yang sudah ada</li> <li>3. Isi <i>form</i> lokasi <i>slot</i> baru</li> <li>4. Klik simpan</li> <li>5. Akan tampil tabel <i>slot</i> di sebelah kanan</li> <li>6. Isi id perangkat dari setiap <i>slot</i></li> <li>7. Klik simpan</li> </ol> <p>Ubah</p> <ol style="list-style-type: none"> <li>1. Pilih area parkir yang ingin diubah lokasi <i>slot</i> nya</li> <li>2. Akan tampil <i>list</i> lokasi <i>slot</i> yang sudah ada</li> <li>3. Klik tombol <i>edit</i> pada lokasi <i>slot</i> yang ingin diubah</li> <li>4. Nama lokasi <i>slot</i>, jumlah kapasitas <i>slot</i> dan tabel id perangkat akan tampil</li> <li>5. Ubah data sesuai yang ingin diubah</li> <li>6. Klik tombol simpan</li> </ol>

	<p>Hapus</p> <ol style="list-style-type: none"><li>1. Pilih area parkir yang ingin diubah lokasi <i>slot</i> nya</li><li>2. Akan tampil <i>list</i> lokasi <i>slot</i> yang sudah ada</li><li>3. Klik tombol hapus pada lokasi <i>slot</i> yang ingin dihapus</li><li>4. Akan tampil konfirmasi hapus, tekan tombol ya</li></ol>
Keluaran	Data Lokasi <i>slot</i> , <i>slot</i> dan id perangkat
Pemeran	SSI/klien

## 2. Sinkronisasi Database







Gambar 3. 3 System Flow Sinkronisasi Database

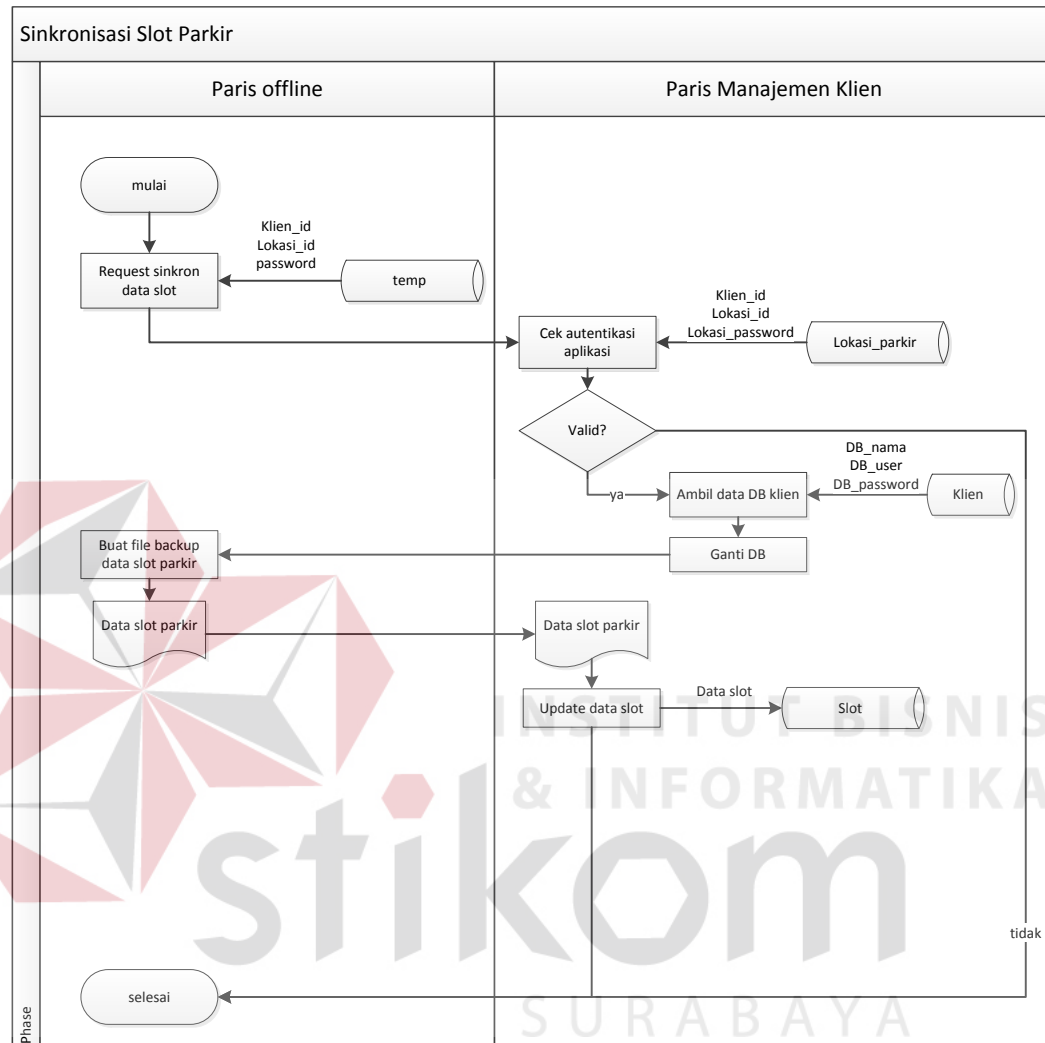
Tabel 3. 4 Penjelasan System Flow Sinkronisasi Database

Sinkronisasi Database	
Tujuan	Untuk menyamakan data di <i>database Paris online</i> dan <i>Paris offline</i>
Masukan	-
Proses	1. Pada <i>server Paris offline</i> , masuk ke aplikasi <i>Paris offline</i>

	<ol style="list-style-type: none"><li>2. Pilih menu <i>auto sync</i> DB</li><li>3. Selesai, sinkronisasi <i>database</i> akan otomatis berjalan sesuai jam sinkronisasi yang telah ditentukan SSI dan disimpan di <i>database</i> tabel temp</li></ol>
Keluaran	Database Paris <i>online</i> dan <i>offline</i> sama
Pemeran	Klien



3. Sinkronisasi slot parkir



Gambar 3. 4 System Flow Sinkronisasi Slot Parkir

Tabel 3. 5 Penjelasan System Flow Sinkronisasi Slot Parkir

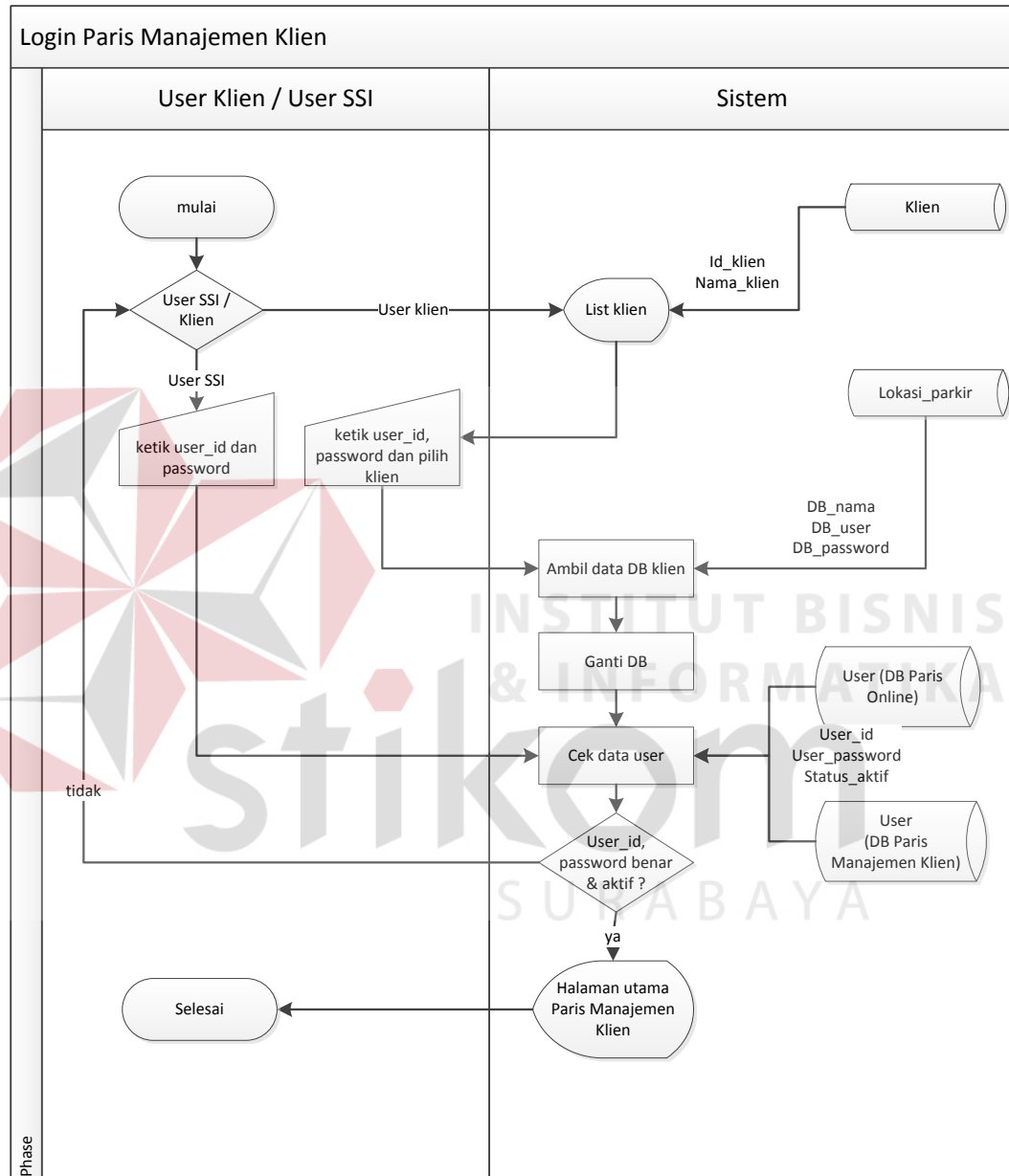
Sinkronisasi Slot Parkir	
Tujuan	Untuk menyamakan status slot parkir di database Paris online dan Paris offline
Masukan	-
Proses	1. Pada server Paris offline, masuk ke aplikasi Paris

	<p><i>offline</i></p> <ol style="list-style-type: none"><li>2. Pilih menu <i>auto sync</i> DB</li><li>3. Selesai, sinkronisasi <i>slot</i> parkir akan otomatis berjalan sesuai interval waktu sinkronisasi yang telah ditentukan SSI dan disimpan di <i>database</i> tabel temp</li></ol>
Keluaran	Data status <i>slot</i> Paris <i>online</i> dan <i>offline</i> sama
Pemeran	Klien



## B. Paris Manajemen Klien

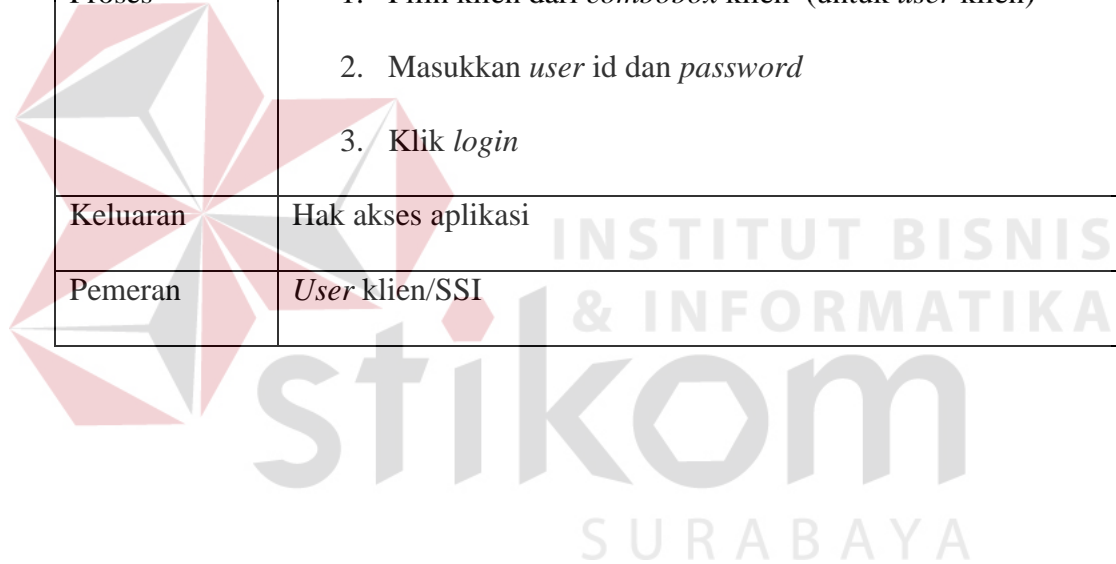
### 1. Login Paris Manajemen Klien



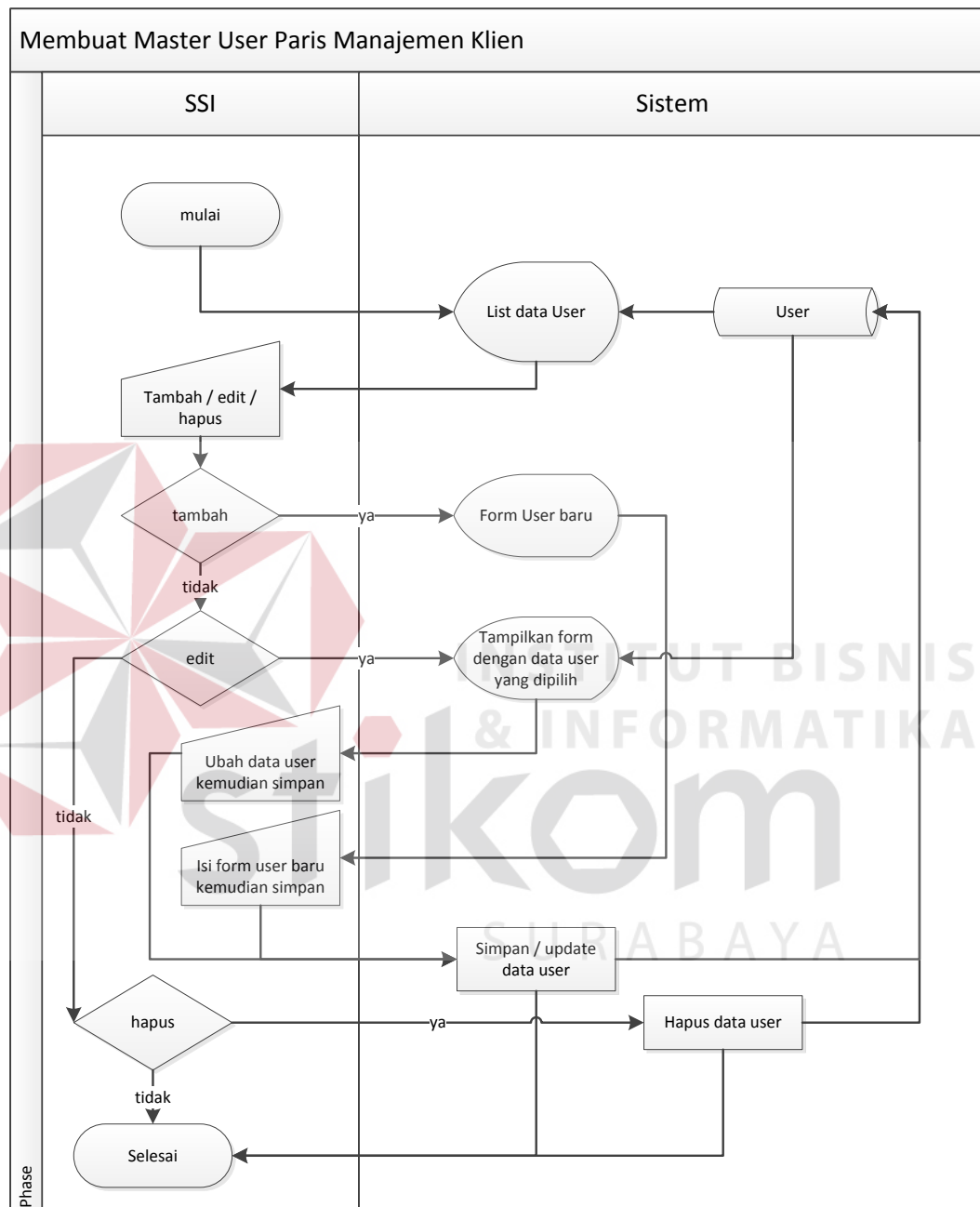
Gambar 3. 5 System Flow Login Paris Manajemen Klien

Tabel 3. 6 Penjelasan *System Flow Login* Paris Manajemen Klien

<i>Login</i> Paris Manajemen Klien	
Tujuan	Mendapatkan hak akses untuk masuk ke aplikasi Paris Manajemen Klien
Masukan	- klien id (untuk <i>user</i> klien) - <i>user</i> id - <i>password</i>
Proses	1. Pilih klien dari <i>combobox</i> klien (untuk <i>user</i> klien) 2. Masukkan <i>user</i> id dan <i>password</i> 3. Klik <i>login</i>
Keluaran	Hak akses aplikasi
Pemeran	<i>User</i> klien/SSI



## 2. Membuat Master *User* Paris Manajemen Klien



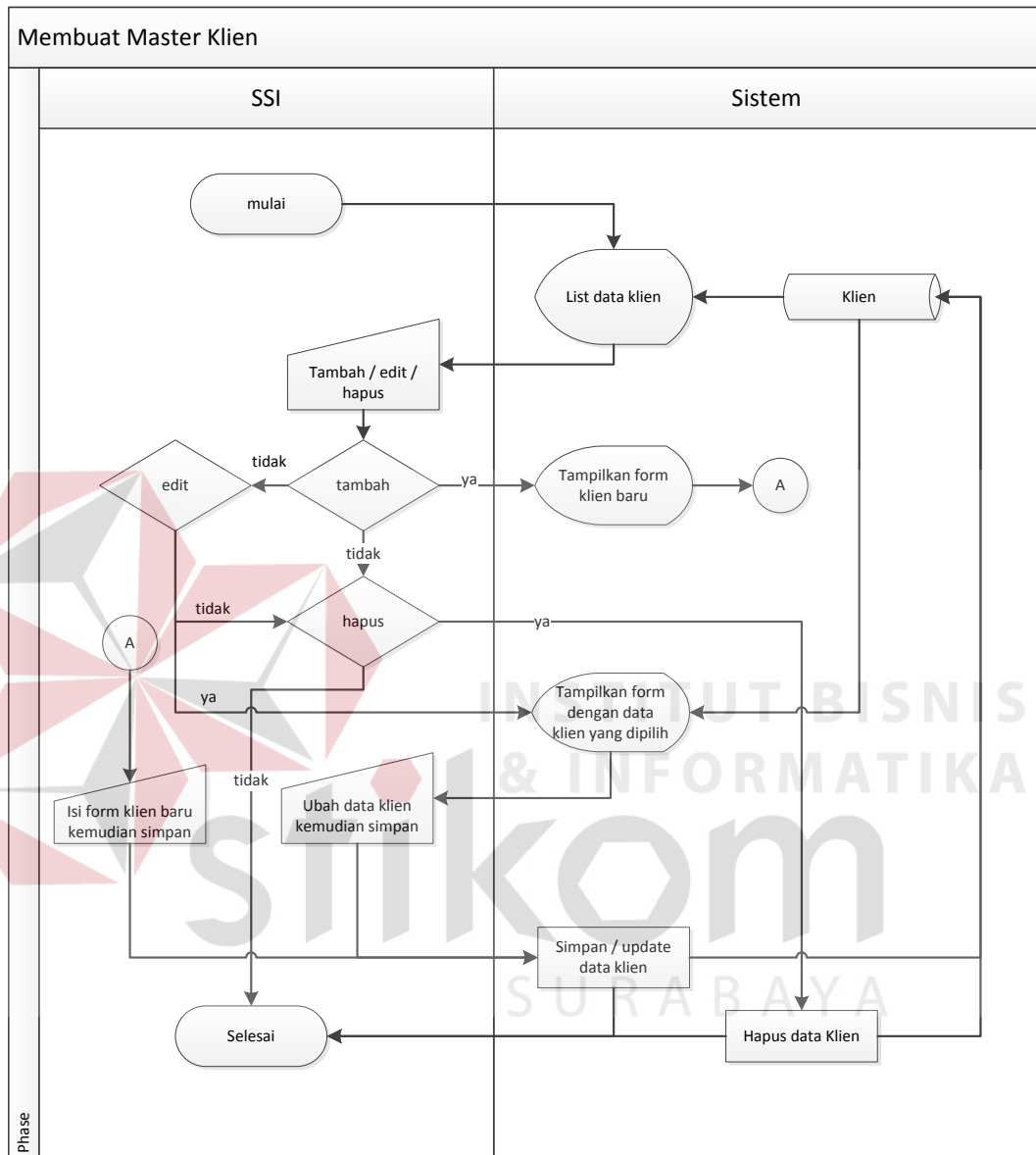
Gambar 3. 6 *System Flow* Membuat Master *User* Paris Manajemen Klien

Tabel 3. 7 Penjelasan *System Flow* Membuat Master *User*

Membuat Master <i>User</i> Paris Manajemen Klien	
Tujuan	Menambah, merubah, atau hapus data <i>user</i> Paris Manajemen Klien ( <i>user</i> untuk SSI)
Masukan	- Nama <i>user</i> - <i>password</i> - status aktif
Proses	<p>Tambah</p> <ol style="list-style-type: none"> <li>1. Klik tombol tambah <i>user</i> baru</li> <li>2. Isi semua kolom pada <i>form</i></li> <li>3. Klik simpan</li> </ol> <p>Ubah</p> <ol style="list-style-type: none"> <li>1. Klik tombol ubah pada data <i>user</i> yang ingin diubah</li> <li>2. Ubah data sesuai yang diinginkan</li> <li>3. Klik simpan</li> </ol> <p>Hapus</p> <ol style="list-style-type: none"> <li>1. Klik tombol hapus pada data <i>user</i> yang ingin dihapus</li> <li>2. Akan ada konfirmasi, tekan tombol ya</li> </ol>
Keluaran	Data <i>user</i>
Pemeran	SSI



3. Membuat Master Klien



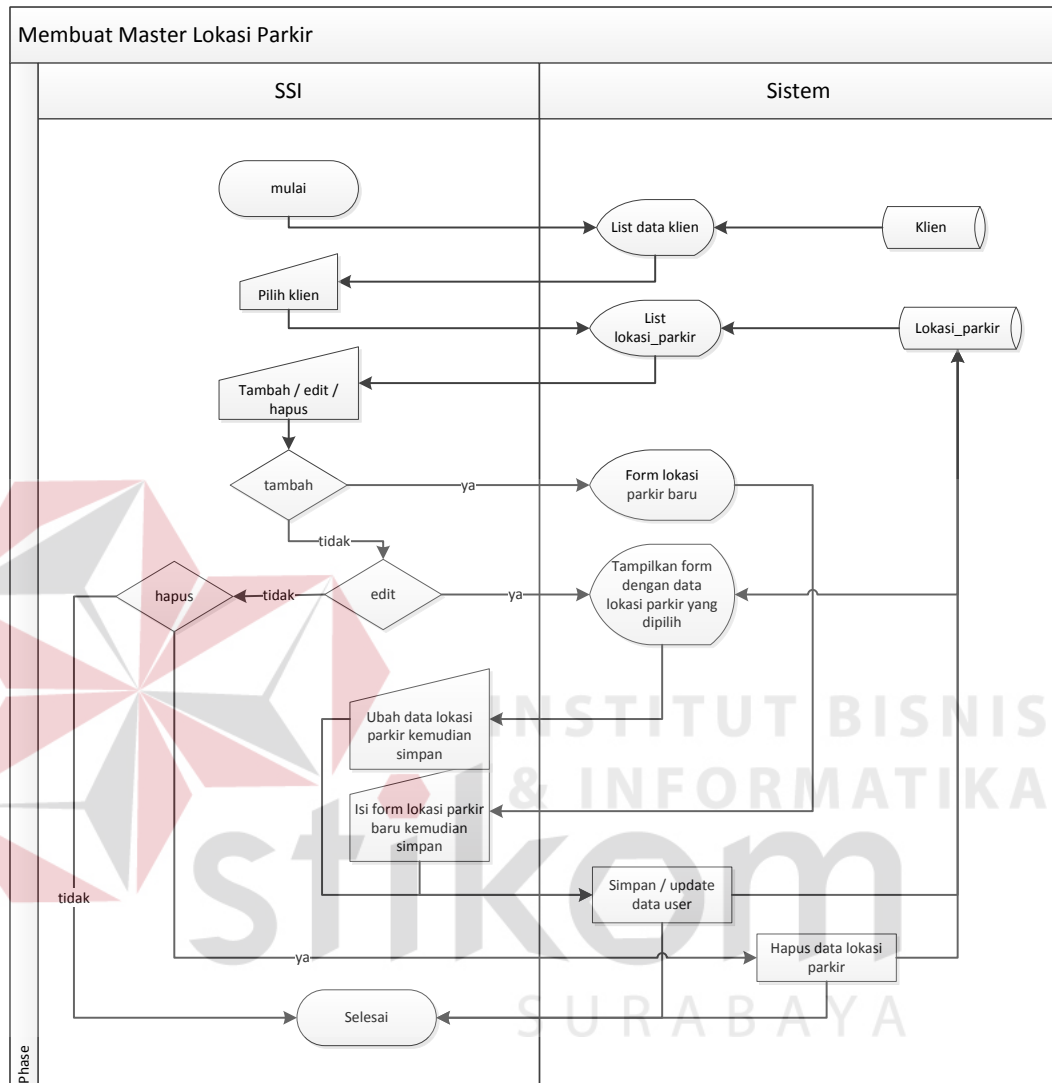
Gambar 3. 7 System Flow Membuat Master Klien

Tabel 3. 8 Penjelasan System Flow Membuat Master Klien

Membuat Master Klien	
Tujuan	Menambah, merubah, atau hapus data klien Paris
Masukan	- Nama Klien - tanggal daftar

	- status aktif
Proses	<p>Tambah</p> <ol style="list-style-type: none"> <li>1. Klik tombol tambah klien baru</li> <li>2. Isi semua kolom pada <i>form</i></li> <li>3. Klik simpan</li> </ol> <p>Ubah</p> <ol style="list-style-type: none"> <li>1. Klik tombol ubah pada data klien yang ingin diubah</li> <li>2. Ubah data sesuai yang diinginkan</li> <li>3. Klik simpan</li> </ol> <p>Hapus</p> <ol style="list-style-type: none"> <li>1. Klik tombol hapus pada data klien yang ingin dihapus</li> <li>2. Akan ada konfirmasi, tekan tombol ya.</li> </ol>
Keluaran	Data Klien
Pemeran	SSI

4. Membuat Master Lokasi Parkir



Gambar 3. 8 System Flow Membuat Master Lokasi Parkir

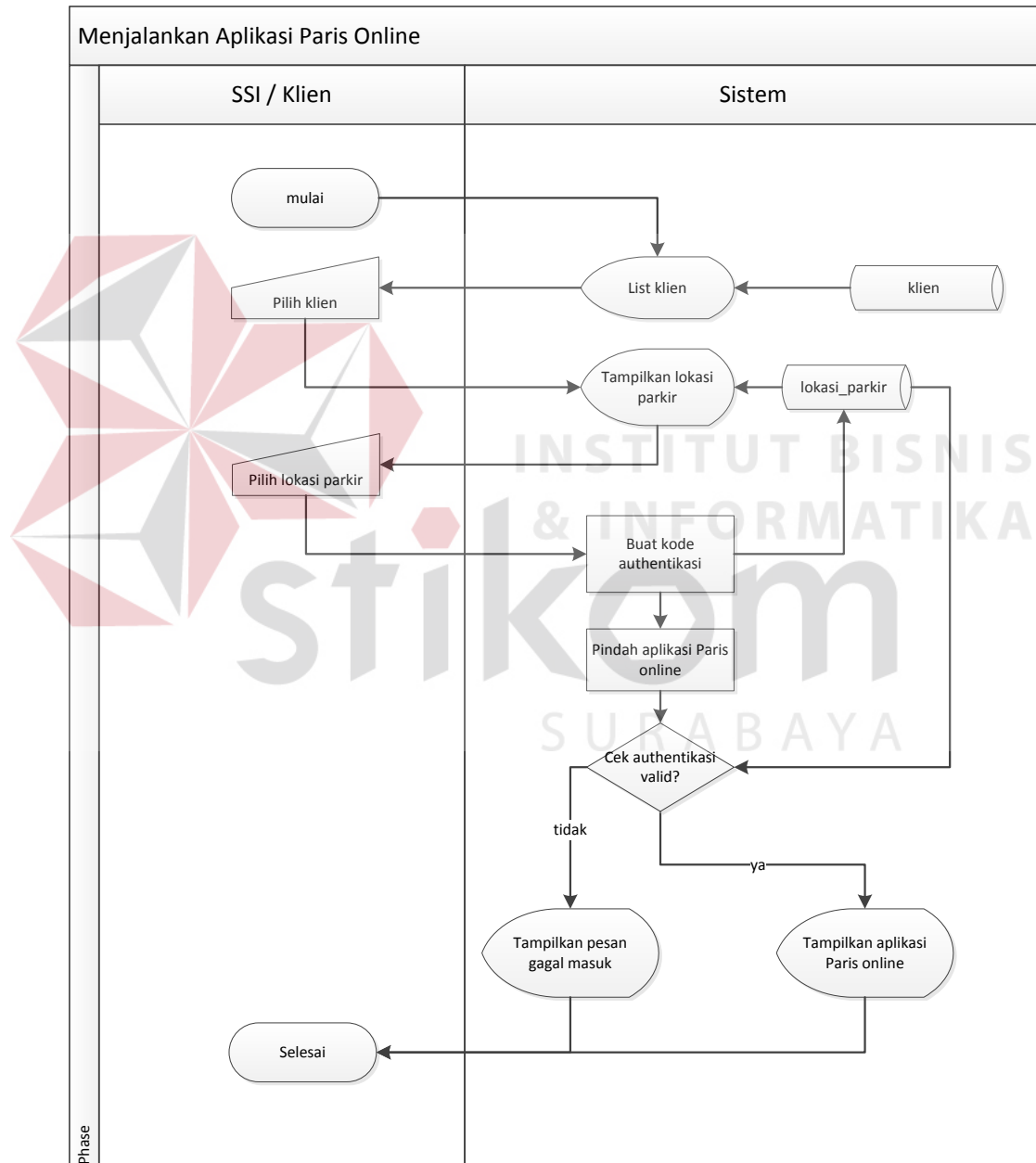
Tabel 3. 9 Penjelasan System Flow Membuat Master Lokasi Parkir

Membuat Master Lokasi Parkir	
Tujuan	<ul style="list-style-type: none"> <li>- Menambah, merubah, atau hapus data lokasi parkir Paris</li> <li>- memasukkan akun <i>database Paris online</i></li> <li>- memasukkan <i>password</i> untuk sinrkonasi dengan <i>Paris offline</i></li> </ul>
Masukan	- Nama lokasi

	<ul style="list-style-type: none"> <li>- alamat lokasi</li> <li>- status aktif</li> <li>- akun <i>database</i></li> <li>- <i>password</i> sinkronasi</li> </ul>
Proses	<p>Tambah</p> <ol style="list-style-type: none"> <li>1. Buat akun <i>database</i> baru (di MySQL)</li> <li>2. Buat <i>database</i> untuk Paris <i>online</i></li> <li>3. <i>Import database</i> Paris <i>offline</i> ke <i>database online</i></li> <li>4. Beri hak akses untuk akun yang tadi dibuat untuk dapat mengakses <i>database</i>.</li> <li>5. Klik tombol tambah lokasi parkir baru</li> <li>6. Isi semua kolom pada <i>form</i> termasuk data akun <i>database</i> dan <i>password</i> sinkronasi</li> <li>7. Klik simpan</li> </ol> <p>Ubah</p> <ol style="list-style-type: none"> <li>1. Klik tombol ubah pada data lokasi parkir yang ingin diubah</li> <li>2. Ubah data sesuai yang diinginkan</li> <li>3. Klik simpan</li> </ol> <p>Hapus</p> <ol style="list-style-type: none"> <li>4. Klik tombol hapus pada data lokasi parkir yang ingin dihapus</li> <li>5. Akan ada konfirmasi, tekan tombol ya</li> <li>6. <i>Delete database</i></li> </ol>

	7. Delete akun database
Keluaran	Data lokasi parkir
Pemeran	SSI

### 5. Menjalankan Aplikasi Paris Online



Gambar 3. 9 System Flow Menjalankan Aplikasi Paris Online

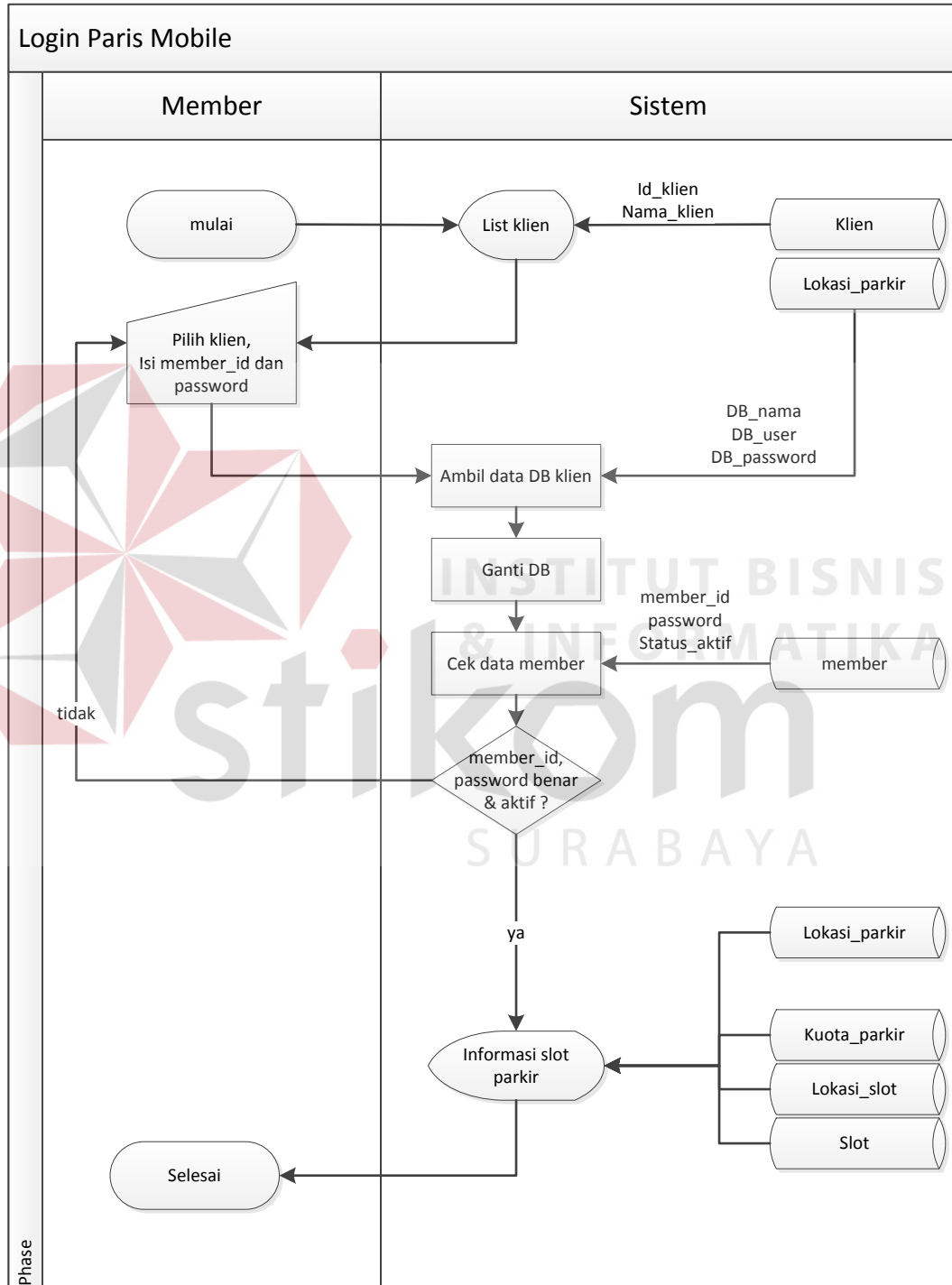
Tabel 3. 10 Penjelasan *System Flow* Menjalankan Aplikasi Paris *Online*

Menjalankan Aplikasi Paris <i>Online</i>	
Tujuan	Untuk mengakses paris <i>online</i>
Masukan	-
Proses	1. Klik lokasi parkir yang ingin diakses
Keluaran	Halaman Paris <i>online</i>
Pemeran	SSI / Klien



### C. Paris

#### 1. Login Paris Mobile



Gambar 3. 10 System Flow Login Paris Mobile

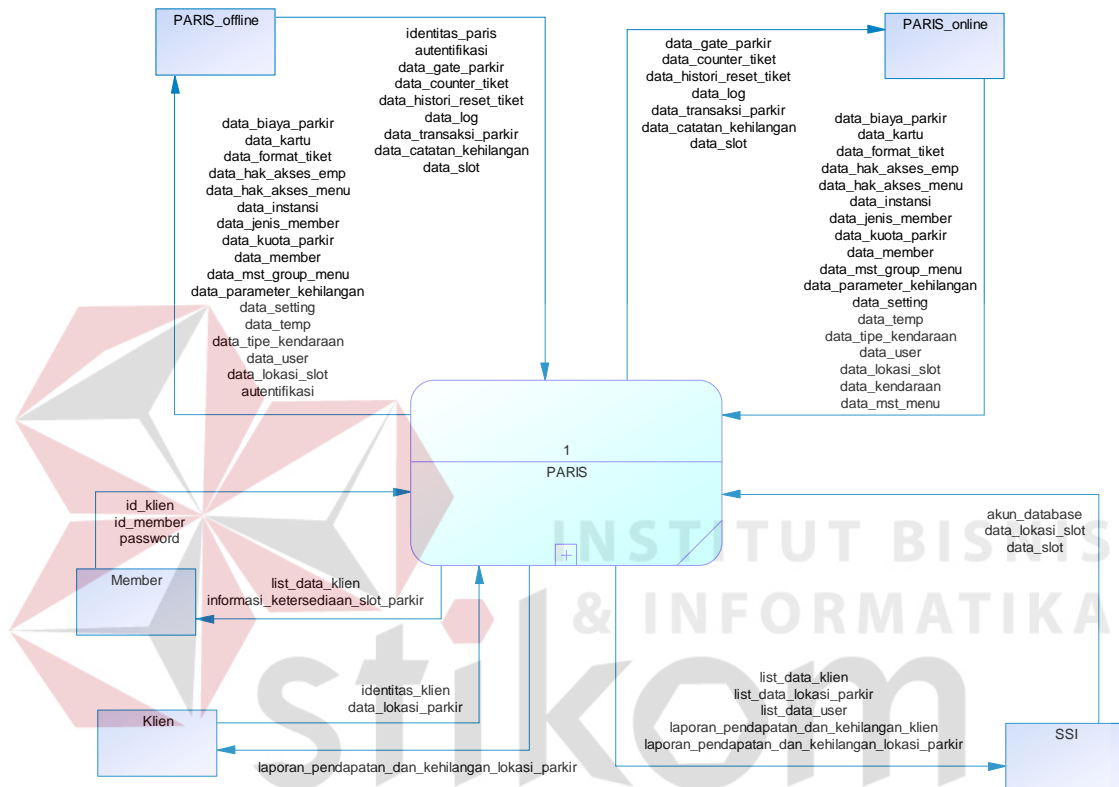
Tabel 3. 11 Penjelasan *System Flow Login Paris Mobile*

<i>Login Paris Mobile</i>	
Tujuan	Mendapatkan informasi ketersediaan <i>slot</i> parkir
Masukan	<ul style="list-style-type: none"> <li>- <i>Member</i> id</li> <li>- <i>Password</i></li> <li>- Klien id</li> </ul>
Proses	<ol style="list-style-type: none"> <li>1. Aplikasi akan menampilkan <i>list</i> klien Paris</li> <li>2. Pilih klien tempat <i>member</i> terdaftar</li> <li>3. Masukkan <i>member</i> id dan <i>password</i></li> <li>4. Klik tombol <i>login</i></li> <li>5. Aplikasi akan menampilkan semua lokasi parkir klien tersebut dan informasi ketersediaan <i>slot</i> parkir di lokasi-lokasi</li> </ol>
Keluaran	Informasi ketersediaan <i>slot</i> parkir
Pemeran	<i>Member</i>



### 3.2.3 Konteks Diagram

Konteks Diagram adalah diagram paling dasar pada DFD. Pada konteks diagram digambarkan semua entitas luar dan satu proses secara keseluruhan tanpa memuat simbol data *store*.



Gambar 3. 11 Konteks Diagram

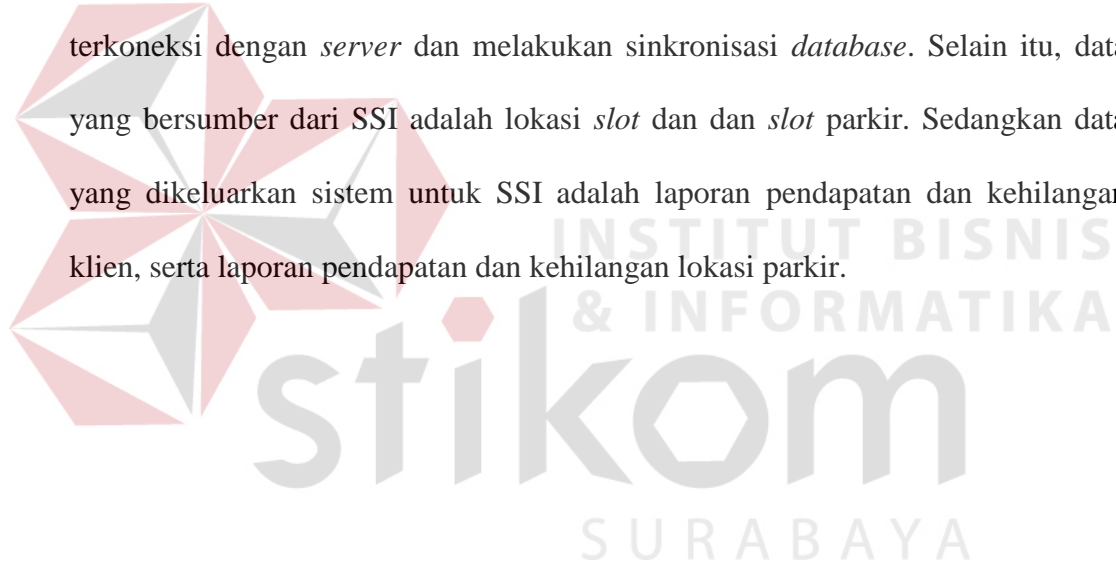
Pada Gambar 3.13 konteks diagram terdapat satu proses utama yaitu Paris yang merupakan nama sistem dan lima entitas luar yaitu Paris *online* dan Paris *offline* yang merupakan aplikasi berbeda, kemudian *member* parkir, klien Paris, dan terakhir adalah SSI.

Paris *offline* akan mengirim data transaksi parkir, catatan kehilangan dan beberapa data lain kemudian Paris *online* akan menerimanya dan berganti mengirimkan data pengaturan dan data master untuk di *update* di Paris *offline*.

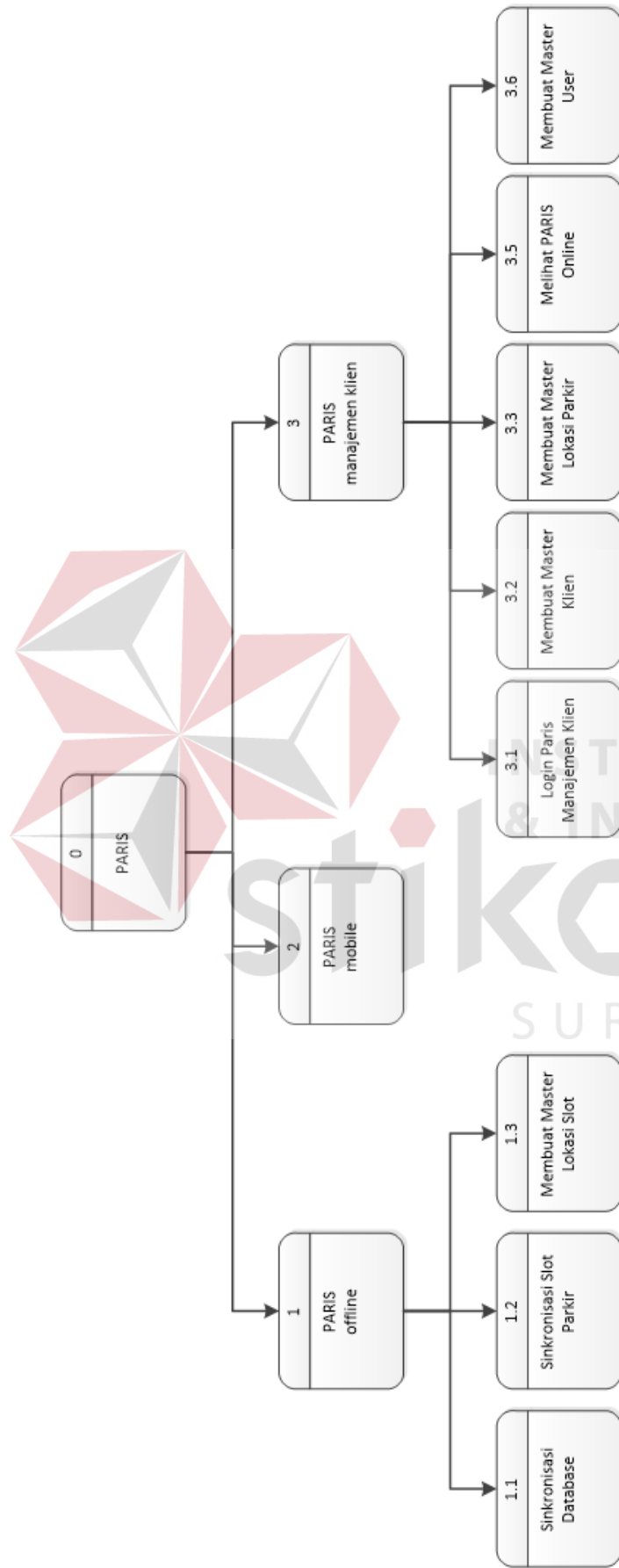
*Member* akan memberikan masukan berupa data untuk melakukan validasi identitas dan sistem akan memberikan informasi tentang ketersediaan *slot* parkir di lokasi-lokasi Paris klien tempat terdaftarnya *member* tersebut.

Klien Paris dalam konteks diagram memberikan data identitasnya seperti nama dan alamat, juga lokasi-lokasi parkirnya. Data yang akan diterima kembali oleh klien adalah laporan pendapatan dan kehilangan lokasi parkir.

SSI dalam sistem ini memberikan masukan berupa akun *database* untuk aplikasi Paris *online* dan *password* aplikasi untuk Paris *offline* untuk dapat terkoneksi dengan *server* dan melakukan sinkronisasi *database*. Selain itu, data yang bersumber dari SSI adalah lokasi *slot* dan dan *slot* parkir. Sedangkan data yang dikeluarkan sistem untuk SSI adalah laporan pendapatan dan kehilangan klien, serta laporan pendapatan dan kehilangan lokasi parkir.







Gambar 3. 12 Diagram Jenjang

Diagram berjenjang diatas menjelaskan fungsional yang dibuat untuk sistem Paris. Terdapat tiga aplikasi yang dibuat atau ditambah fungsionalitasnya dan akan dijelaskan sebagai berikut:

1. Aplikasi Paris *Offline*

Aplikasi Paris *offline* adalah aplikasi yang sudah ada sebelumnya dan dipakai di lokasi parkir untuk mengatur atau membuat laporan parkir. Aplikasi ini akan ditambahkan tiga fungsionalitas baru yaitu sinkronisasi *database*, sinkronisasi *slot* parkir, dan membuat master lokasi *slot*.

2. Aplikasi Paris *Mobile*

Aplikasi ini adalah aplikasi android yang akan digunakan oleh *member* Paris untuk mendapatkan informasi tentang ketersediaan *slot* parkir di lokasi-lokasi parkir Paris.

3. Aplikasi Paris Manajemen Klien

Aplikasi Paris Manajemen Klien adalah aplikasi yang dibuat untuk menjembatani semua aplikasi Paris klien. Aplikasi ini juga akan digunakan untuk membuat laporan gabungan dari semua klien dan semua lokasi parkir klien.

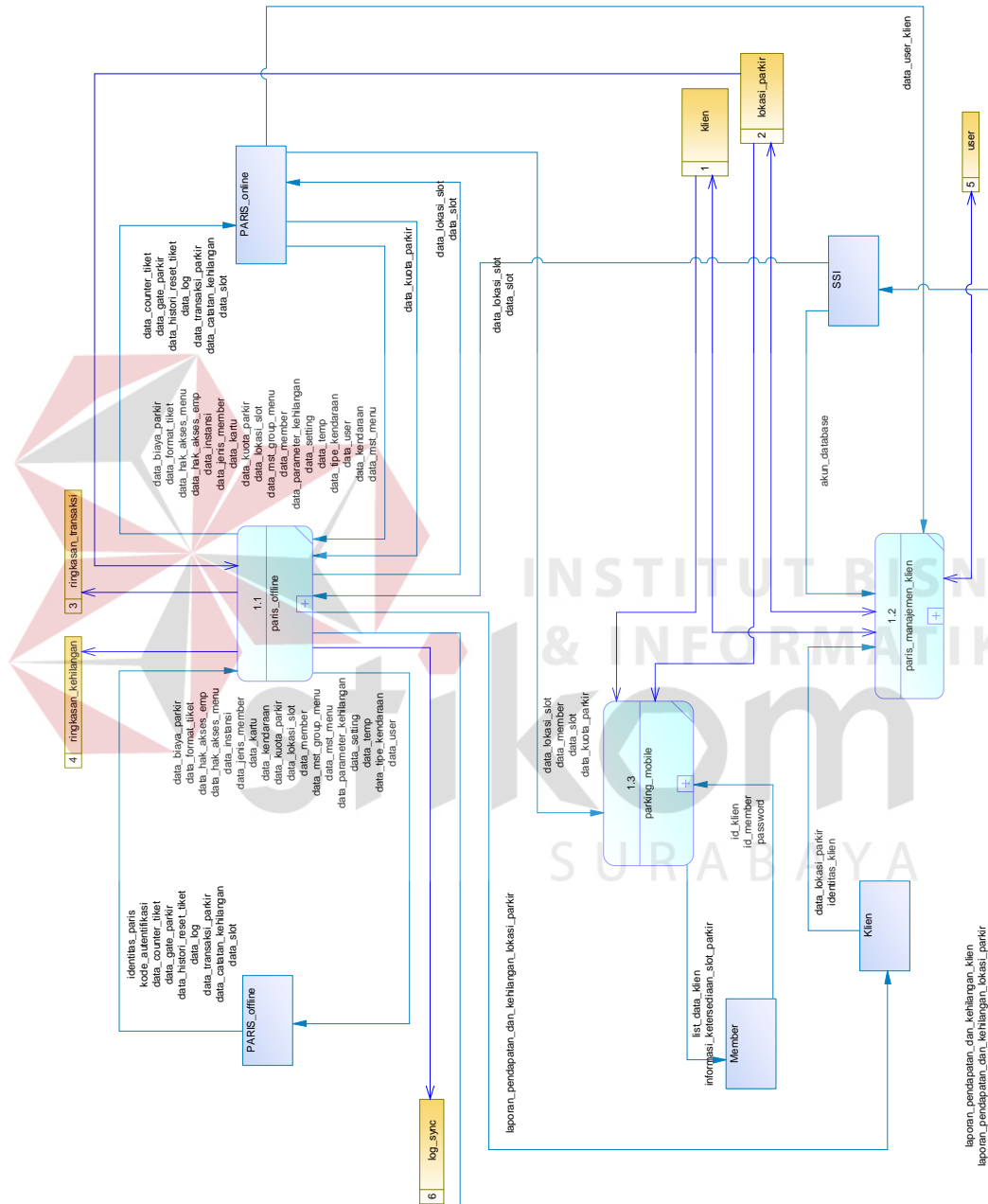
### 3.2.4 DFD (*Data Flow Diagram*)

1. DFD Level 0

DFD level 0 adalah *decompose* dari konteks diagram yang menjelaskan lebih rinci tiap aliran data dan proses aplikasi sistem Paris. Proses utama dari Paris terbagi menjadi tiga proses utama yang mewakili nama aplikasi baru dan aplikasi yang ditambah fungsionalitasnya yaitu Paris *offline*, Paris *mobile*, dan aplikasi

utama di tugas akhir ini yaitu Paris Manajemen Klien. Proses dari ketiga aplikasi ini akan dijelaskan pada level selanjutnya.



Gambar 3. 13 DFD  
Level 0

## 2. DFD Level 1

### A. Sub Proses Paris *Offline*

Proses yang pertama adalah master *slot* parkir. Fungsi proses ini adalah menyimpan detail area-area parkir di lokasi parkir dan id perangkat yang terhubung dengan *server offline*. Id perangkat yang dimaksud adalah sebuah alat infra merah atau gelombang *ultrasonic* yang akan mendeteksi sebuah *slot* parkir sedang dalam keadaan kosong atau dipakai. Perangkat tersebut akan mengirim status kepada *server* jika terdapat objek yang berada di *slot* parkir.

Proses yang kedua adalah sinkronisasi *database*. Proses ini adalah proses menyamakan data antara Paris *online* dan Paris *offline*. Penyamaan data ini diperlukan untuk klien yang mengakses Paris dari luar lokasi parkir untuk mendapatkan data yang terbaru. Penyamaan data juga diperlukan oleh Paris *offline* karena klien yang ingin mengubah pengaturan atau menambah data tidak perlu lagi datang ke setiap lokasi, hanya perlu mengakses aplikasi Paris *online* dan otomatis data akan diupdate di Paris *offline*.

Tahapan proses yang dilakukan pada sinkronisasi *database* adalah *login* aplikasi, *export* data Paris *offline*, *import* data Paris *offline*, *export* data Paris *online*, *import* data Paris *online*, dan membuat ringkasan transaksi dan ringkasan catatan kehilangan.

Tahapan pertama adalah *login* Paris Manajemen Klien aplikasi. Setiap paris *offline* akan diberi id klien, id lokasi parkir, dan *password* aplikasi yang sebelumnya sudah diatur pada master lokasi parkir. Data-data itu akan digunakan untuk *login* dan *server* Paris akan melakukan cek apakah semua data valid atau tidak. Proses ini diperlukan untuk mencegah tidak ada pihak yang tidak berhak

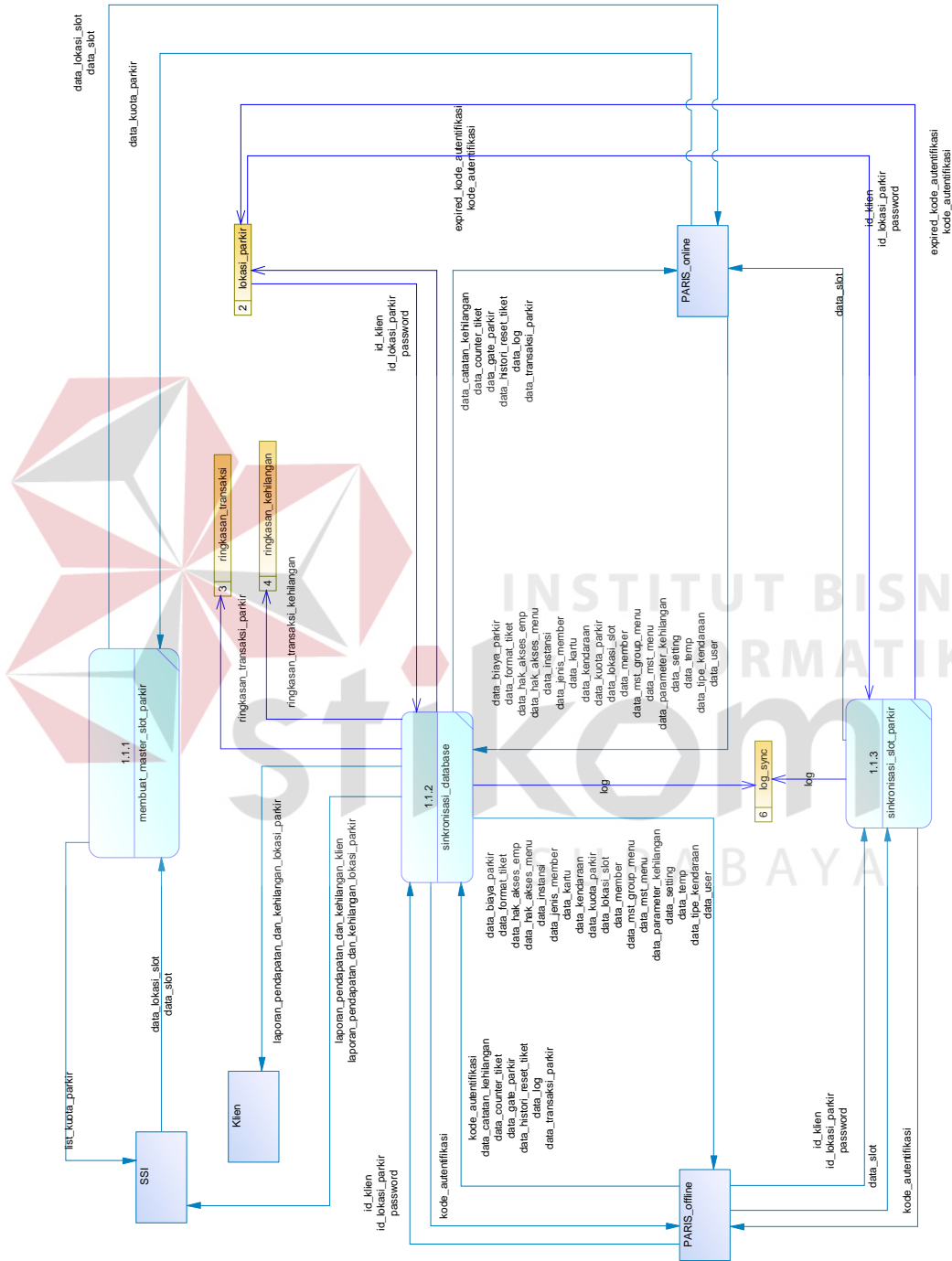


masuk dan mengirim atau mengambil data di *server* Paris kecuali yang sudah terdaftar.

Tahapan yang kedua adalah *export* data. *Export* data akan dilakukan oleh Paris *offline* dengan membuat *file* berformat JSON (Javascript *Object Orientation*). *File* JSON berisi data transaksi, catatan kehilangan dan beberapa data lain yang akan diupload ke *server* dan selanjutnya akan diupdate di *database* Paris *offline*.

Tahapan keempat adalah *export* data Paris *online* yang berupa data master seperti biaya, *user* aplikasi Paris, kemudian data pengaturan seperti jam sinkronisasi *database* dan interval sinkronisasi data *slot*. Data *export* tersebut akan dimasukkan kedalam *file* JSON juga dan kemudian didownload oleh Paris *offline* dan akan diupdate di *database* Paris *online*.

Proses ketiga adalah sinkronisasi *slot* parkir. Proses ini melakukan penyamaan data *slot* parkir dari *database* Paris *offline* ke *database* Paris *online*. Data yang diupdate hanya statusnya saja, apakah *slot* parkir sedang terisi atau tidak.



Gambar 3.14 DF D Level 1 Sub Proses Paris Off

## B. Sub Proses Paris Manajemen Klien

Proses Paris Manajemen Klien memiliki banyak sub-sub proses yaitu *login* Paris Manajemen Klien, membuat master *user*, membuat master klien, membuat master lokasi parkir, dan menjalankan aplikasi Paris *online*.

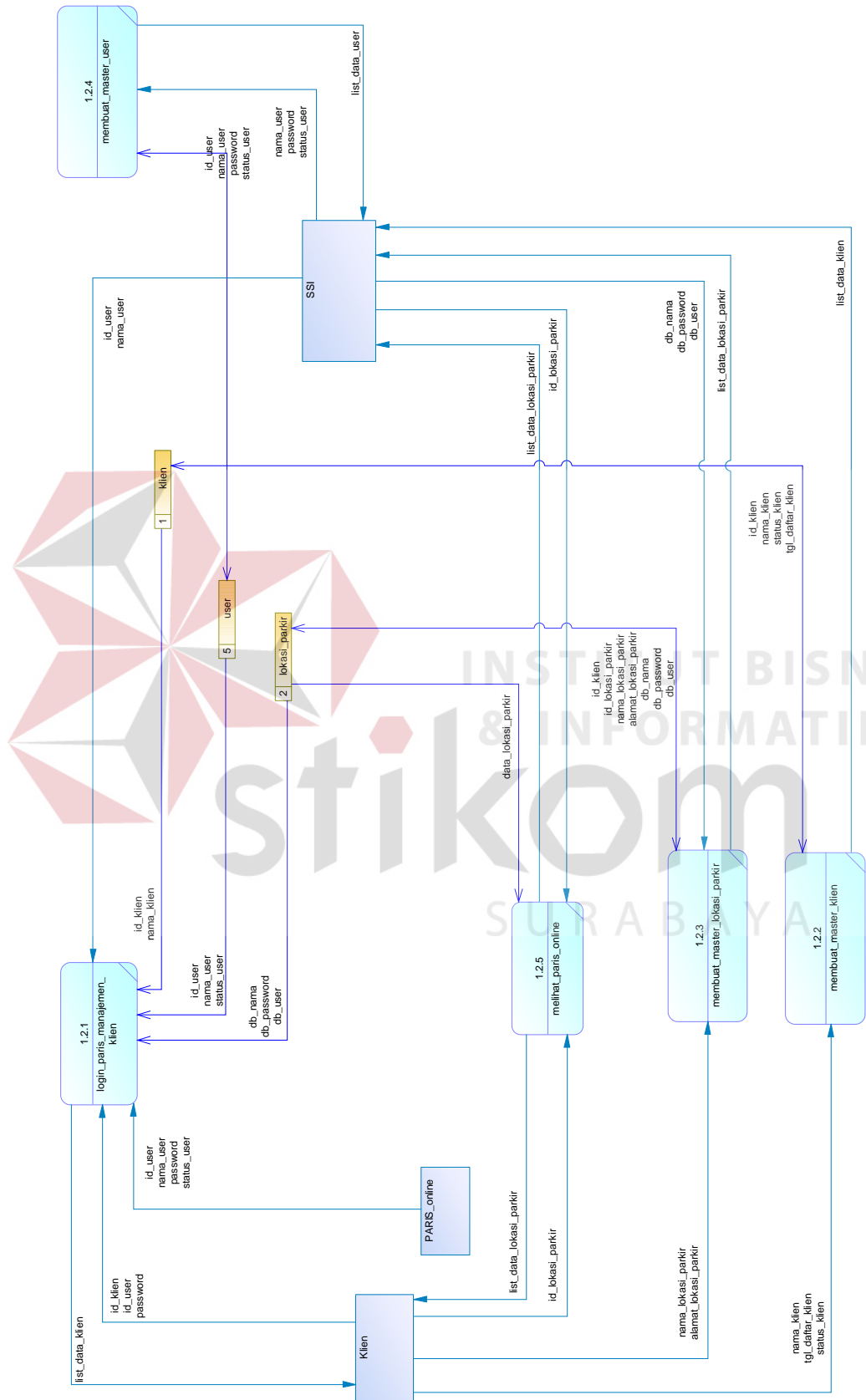
Proses pertama adalah *login* Paris Manajemen Klien. Proses ini diperlukan untuk dapat mengakses proses-proses lainnya. Proses dimulai dengan memasukkan id *user* dan *password* dan id klien untuk *user* dari klien. Sistem akan mengecek apakah data yang dimasukkan valid atau tidak.

Proses kedua adalah membuat master klien. Data yang diperlukan adalah nama, tanggal daftar, dan status aktif klien. Data keluaran dari proses ini adalah *list* data klien yang akan dialirkan ke SSI.

Proses ketiga adalah membuat master lokasi parkir. Data bersumber dari dua entitas yaitu klien dan SSI, data dari klien adalah nama lokasi parkir dan alamat. Sedangkan dari SSI, data yang diberikan adalah *password* aplikasi dan akun *database* (nama, *user*, dan *password*). Keluaran dari sistem ini berupa *list* data lokasi parkir yang sebelumnya sudah pernah disimpan.

Proses keempat adalah membuat master *user*. Data bersumber dari SSI yang berupa nama, *password*, dan status yang akan diberikan ke sistem untuk disimpan. Sedangkan keluaran yang dihasilkan sistem adalah *list user* dari data yang sudah pernah disimpan.

Proses kelima adalah menjalankan Paris *online*. Data yang dikirim oleh klien atau SSI adalah id lokasi parkir yang ingin dilihat. Kemudian sistem akan melakukan *redirect* halaman ke Paris *online* sesuai yang dipilih.



Gambar 3.15 DFD Level 1 Sub Proses Paris Manaj

### C. Sub Proses Paris *Mobile*

Pada sub proses Paris *mobile* hanya memiliki satu proses yaitu *login* Paris *mobile*. Proses ini menggambarkan aliran data untuk *member* parkir Paris. *Member* yang ingin mengetahui ketersediaan parkir di suatu lokasi parkir Paris mengakses aplikasi Paris *mobile* kemudian memasukkan id *member*, *password* dan memilih klien tempat terdaftarnya *member* tersebut maka sistem akan menampilkan halaman utama Paris *mobile* yang berisi informasi ketersediaan *slot* parkir di lokasi-lokasi parkir klien. Terdapat enam tahapan proses Paris *mobile* yaitu:

1. Tampilkan *list* klien paris

Sebelum melakukan *login*, *member* akan diberikan daftar klien Paris.

Dan *member* akan memilih klien mana yang merupakan tempat *member* tersebut terdaftar.

2. *Login member*

Pada sub proses ini *member* memberikan *member* id, *password* dan id klien, kemudian berlanjut ke tahapan selanjutnya.

3. Ambil akun *database* klien

Karena data *member* ada di *database* Paris *online*, dan bisa terdapat lebih dari satu Paris *online* milik klien tempat *member* terdaftar, maka sistem akan mencari akun *database* salah satu Paris *online* milik klien yang diinputkan id kliennya kemudian berlanjut ke tahapan selanjutnya.

4. Ganti *database* klien

Setelah mendapat akun *database*, selanjutnya sistem akan berganti untuk kemudian mengambil data *member* di *database* tersebut.

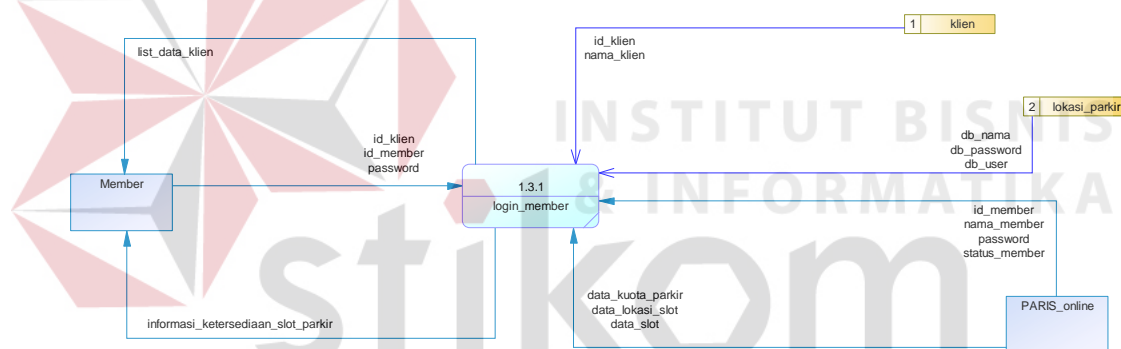


### 5. Ambil data *member*

Untuk mengecek apakah *id member* dan *password* benar, sistem akan mengambil data *member* dengan *id* yang telah dimasukkan sebelumnya. Kemudian sistem akan memberikan hak akses jika data *member* valid.

### 6. Buat informasi ketersediaan *slot* parkir

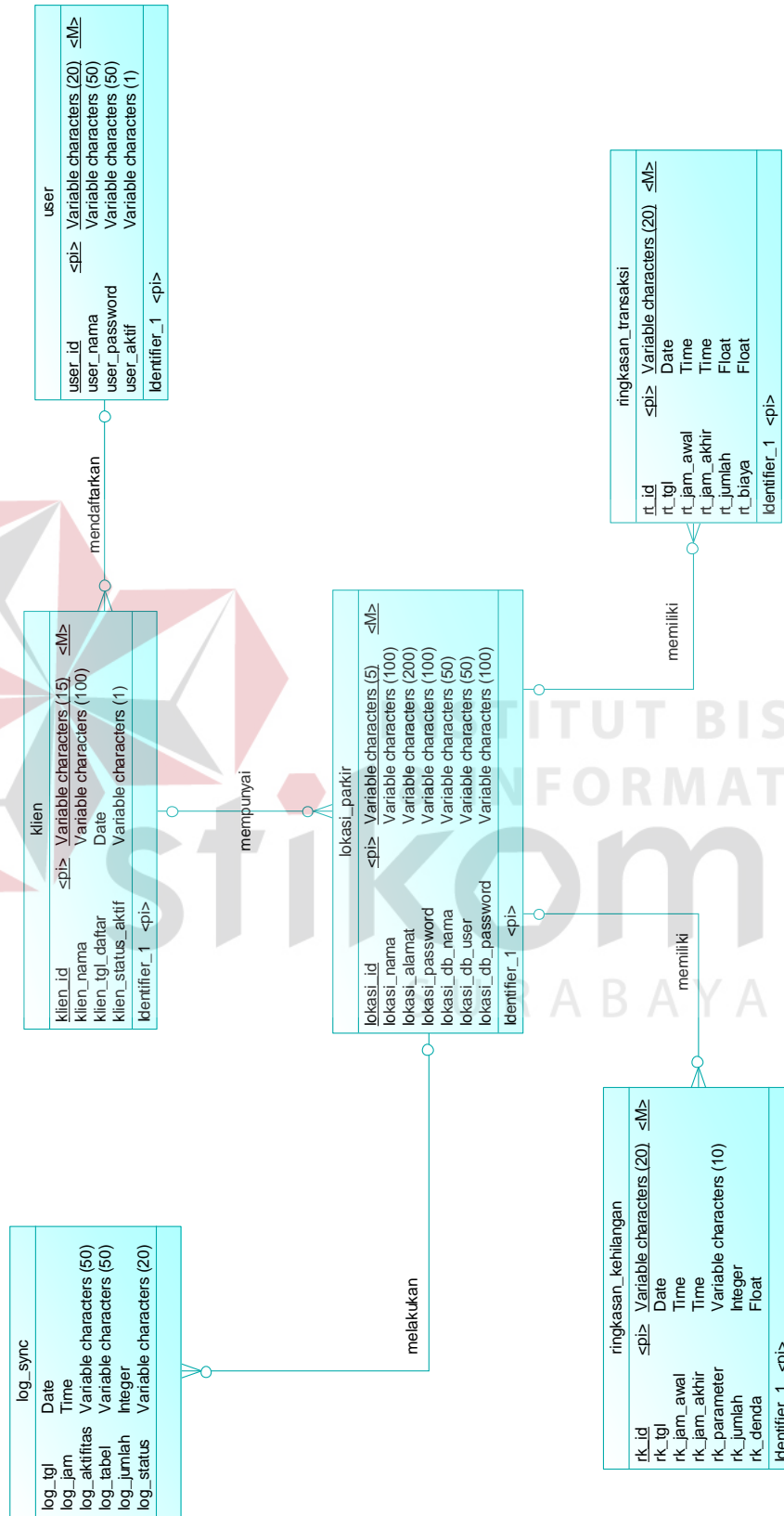
Jika data *login* valid, sistem juga akan otomatis membuat informasi ketersediaan *slot* parkir dari *database-database* Paris *online* milik klien.



Gambar 3. 12 DFD Level 1 Sub Proses Paris *Mobile*

### 3.2.6 ERD (*Entity Relationship Diagram*)

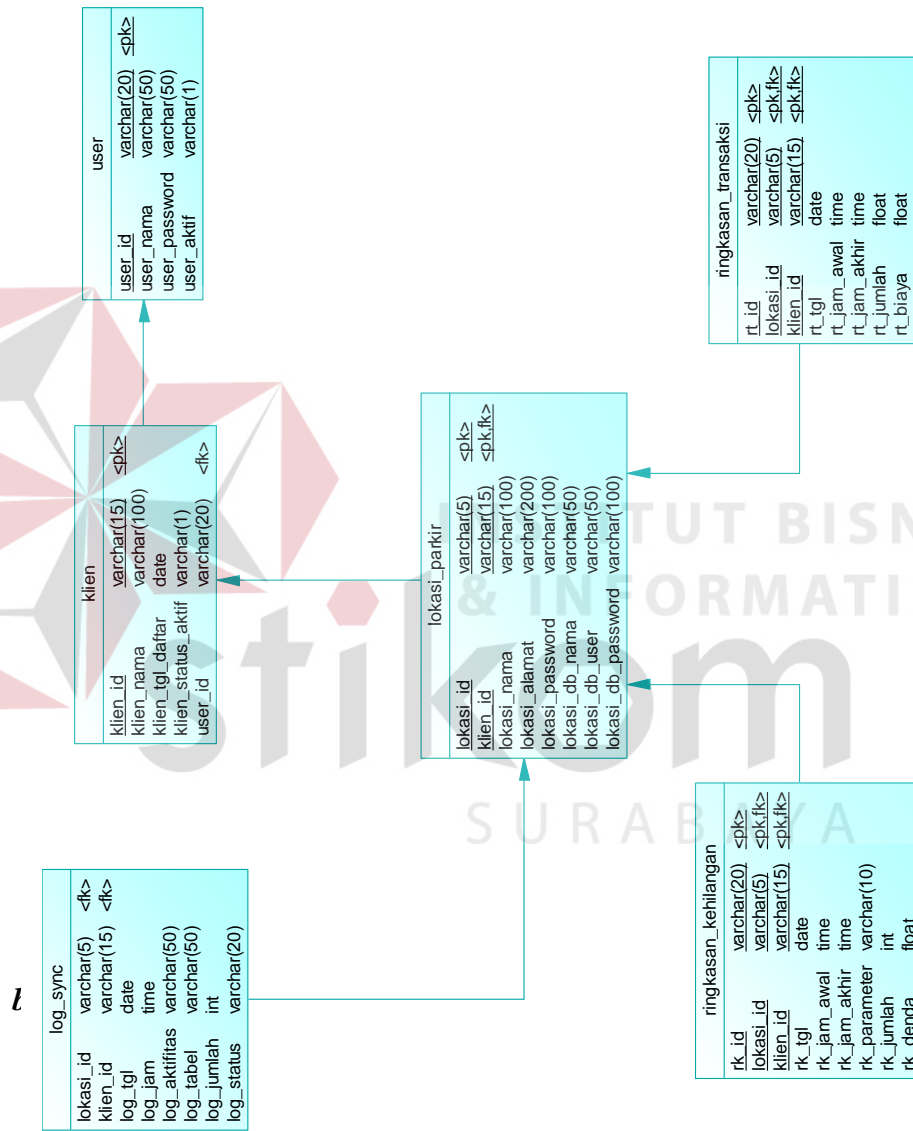
*Entity Relationship* Diagram (ERD) adalah suatu desain sistem yang digunakan untuk merepresentasikan, menentukan dan mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan *database*. ERD juga menyediakan bentuk untuk menunjukkan struktur keseluruhan dari data pemakai. Dalam perencanaan sistem ini telah terbentuk ERD yang merupakan lanjutan dari pembuatan desain dengan menggunakan DFD. ERD terbagi menjadi menjadi CDM dan PDM. Berikut adalah penjelasannya:



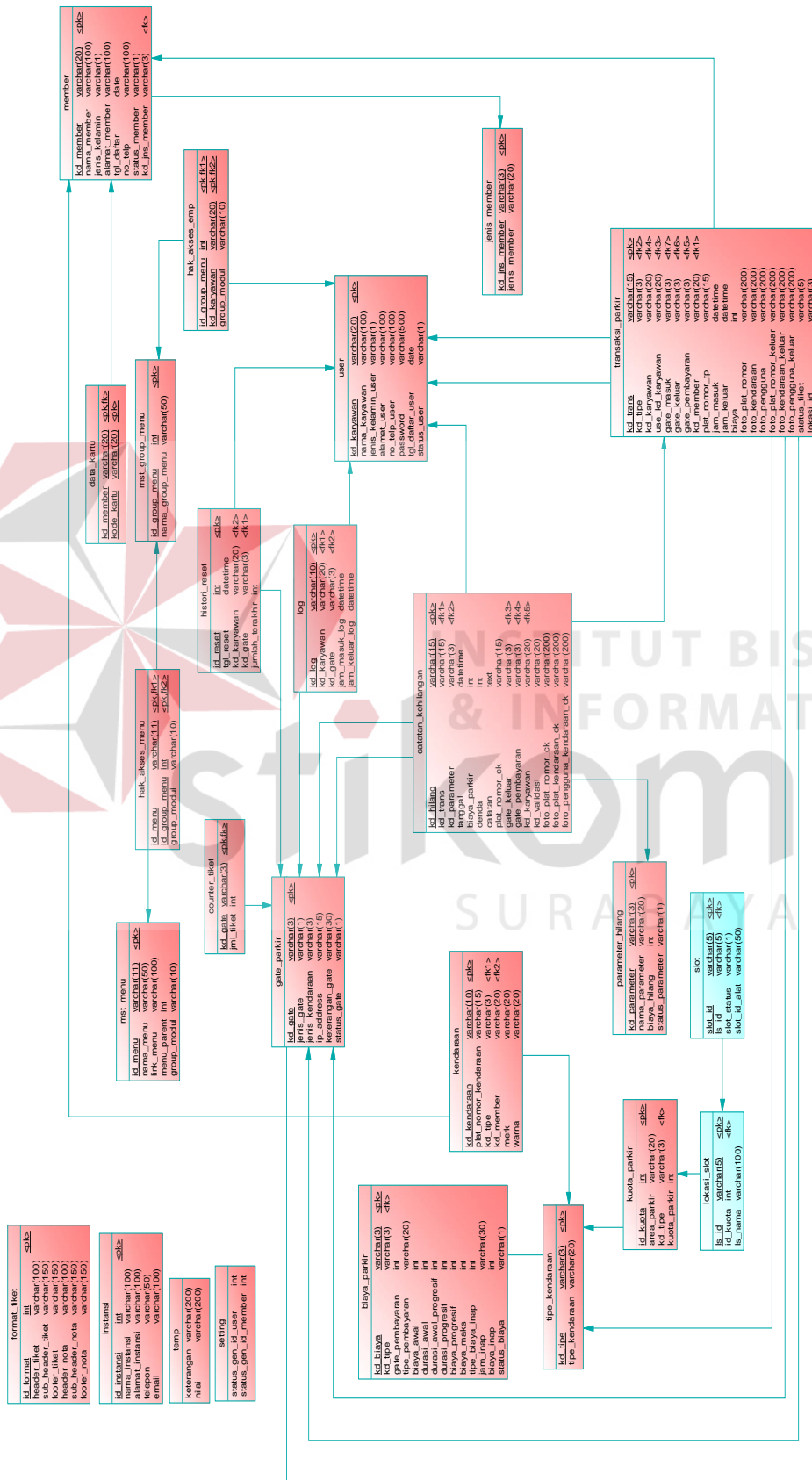
Gambar 3.17 CDM Paris Manajemen Klien







Gambar 3.19 PD M Par is Ma naj em en Kli en



Gambar 3.20 Diagram Database

### 3.2.7 Struktur Database

Struktur *database* adalah uraian struktur fisik dari table-table yang terdapat pada *database*. Fungsi dari struktur *database* adalah menyimpan data-data yang digunakan sistem untuk mengolah dan menghasilkan suatu hasil. Berikut adalah struktur penjelasan secara detail *database* yang digunakan:

#### A. Tambahan Tabel Paris

##### 1. Tabel lokasi\_slot

Nama tabel : lokasi\_slot

Primary Key : ls\_id

Foreign key : id\_kuota

Fungsi : menyimpan data lokasi *slot*

Tabel 3. 12 Tabel Lokasi Slot

No	Nama Field	Tipe Data (Lebar)	Constraint	Foreign Key	
				Tabel	Kolom
1	ls_id	varchar(5)	PK	-	-
2	id_kuota	Int	FK	kuota_parkir	id_kuota
3	ls_nama	varchar(100)	Not Null	-	-

##### 2. Tabel slot

Nama tabel : slot

Primary Key : slot\_id

Foreign key : ls\_id

Fungsi : menyimpan data *slot*

Tabel 3. 13 Tabel *Slot*

No	Nama Field	Tipe Data (Lebar)	Constraint	Foreign Key	
				Tabel	Kolom
1	slot_id	varchar(5)	PK	-	-
2	ls_id	varchar(5)	FK	lokasi_slot	ls_id
3	slot_status	varchar(1)	Not Null	-	-
4	slot_id_alat	varchar(50)			

## B. DATABASE PARIS MK

### 1. Tabel *log\_sync*

Nama tabel : *log\_sync*

Primary Key : slot\_id

Foreign key : -

Fungsi : menyimpan data *log\_sync*

Tabel 3. 14 Tabel *log Sync*

No	Nama Field	Tipe Data (Lebar)	Constraint	Foreign Key	
				Tabel	Kolom
1	lokasi_id	varchar(5)	PK	-	-
2	klien_id	varchar(15)	FK	Klien	klien_id
3	log_tgl	date	Not Null	-	-
4	log_jam	time	Not Null	-	-
5	log_aktifitas	varchar(50)	Not Null	-	-
6	log_tabel	varchar(50)	Not Null	-	-

7	log_jumlah	int	Not Null	-	-
8	log_status	varchar(20)	Not Null	-	-

## 2. Tabel klien

Nama tabel : klien

Primary Key: klien\_id

Foreign key: -

Fungsi: menyimpan data klien

Tabel 3. 15 Tabel Klien

No	Nama Field	Tipe Data (Lebar)	Constraint	Foreign Key	
				Tabel	Kolom
1	klien_id	varchar(15)	PK	-	-
2	klien_nama	varchar(100)	Not Null	-	-
3	klien_tgl_daftar	Date	Not Null	-	-
4	klien_status_aktif	varchar(1)	Not Null	-	-
5	user_id	varchar(20)	FK	user	user_id

## 3. Tabel user

Nama tabel : user

Primary Key : user\_id

Foreign key : -

Fungsi : menyimpan data user

Tabel 3. 16 Tabe *User*

No	Nama Field	Tipe Data (Lebar)	Constraint	Foreign Key	
				Tabel	Kolom
1	user_id	varchar(20)	PK	-	-
2	user_nama	varchar(50)	Not Null	-	-
3	user_password	varchar(50)	Not Null	-	-
4	user_aktif	varchar(1)	Not Null	-	-

## 4. Tabel ringkasan\_kehilangan

Nama tabel : ringkasan\_kehilangan

Primary Key : rk\_id, lokasi\_id, klien\_id

Foreign key : -

Fungsi : menyimpan data ringkasan kehilangan

Tabel 3. 17 Tabel Ringkasan Kehilangan

No	Nama Field	Tipe Data (Lebar)	Constraint	Foreign Key	
				Tabel	Kolom
1	rk_id	varchar(20)	PK	-	-
2	lokasi_id	varchar(5)	FK	lokasi_parkir	lokasi_id
3	klien_id	varchar(15)	FK	klien	klien_id
4	rk_tgl	date	Not Null	-	-
5	rk_jam_awal	time	Not Null	-	-
6	rk_jam_akhir	time	Not Null	-	-
7	rk_parameter	varchar(10)	Not Null	-	-

8	rk_jumlah	int	Not Null	-	-
9	rk_denda	float	Not Null	-	-

5. Tabel ringkasan\_transaksi

Nama tabel : ringkasan\_transaksi

Primary Key : rt\_id, lokasi\_id, klien\_id

Foreign key : -

Fungsi : menyimpan data ringkasan transaksi

Tabel 3. 18 Tabel Ringkasan Transaksi

No	Nama Field	Tipe Data (Lebar)	Constraint	Foreign Key	
				Tabel	Kolom
1	rt_id	varchar(20)	PK	-	-
2	lokasi_id	varchar(5)	FK	lokasi_parkir	lokasi_id
3	klien_id	varchar(15)	FK	klien	klien_id
4	rt_tgl	date	Not Null	-	-
5	rt_jam_awal	time	Not Null	-	-
6	rt_jam_akhir	time	Not Null	-	-
7	rt_jumlah	float	Not Null	-	-
8	rt_biaya	float	Not Null	-	-

6. Tabel lokasi\_parkir

Nama tabel : lokasi\_parkir

Primary Key : rt\_id, lokasi\_id, klien\_id

Foreign key : -



Fungsi : menyimpan data lokasi parkir

Tabel 3. 19 Tabel Lokasi Parkir

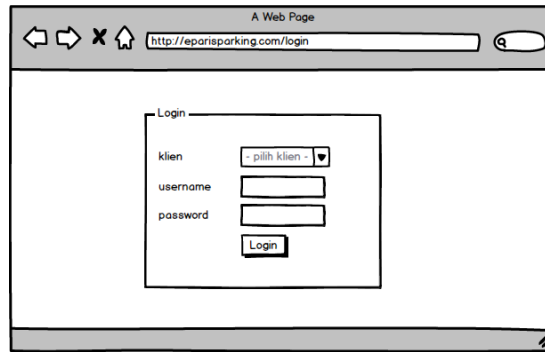
No	Nama Field	Tipe Data (Lebar)	Constraint	Foreign Key	
				Tabel	Kolom
1	lokasi_id	varchar(5)	PK	-	-
2	klien_id	varchar(15)	FK	klien	klien_id
3	lokasi_nama	varchar(100)	Not Null	-	-
4	lokasi_alamat	varchar(200)	Not Null	-	-
5	lokasi_password	varchar(100)	Not Null	-	-
6	lokasi_db_nama	varchar(50)	Not Null	-	-
7	lokasi_db_user	varchar(50)	Not Null	-	-
8	lokasi_db_password	varchar(100)	Not Null	-	-

### 3.2.8 Desain Input Output

Desain *input output* dibuat untuk memberikan gambaran kepada *programmer* dan pengguna sistem bagaimana rupa aplikasi PMK dan cara interaksi sistem dengan penggunanya. Berikut adalah desain *input output* aplikasi PMK:

#### 1. Login Paris Manajemen Klien

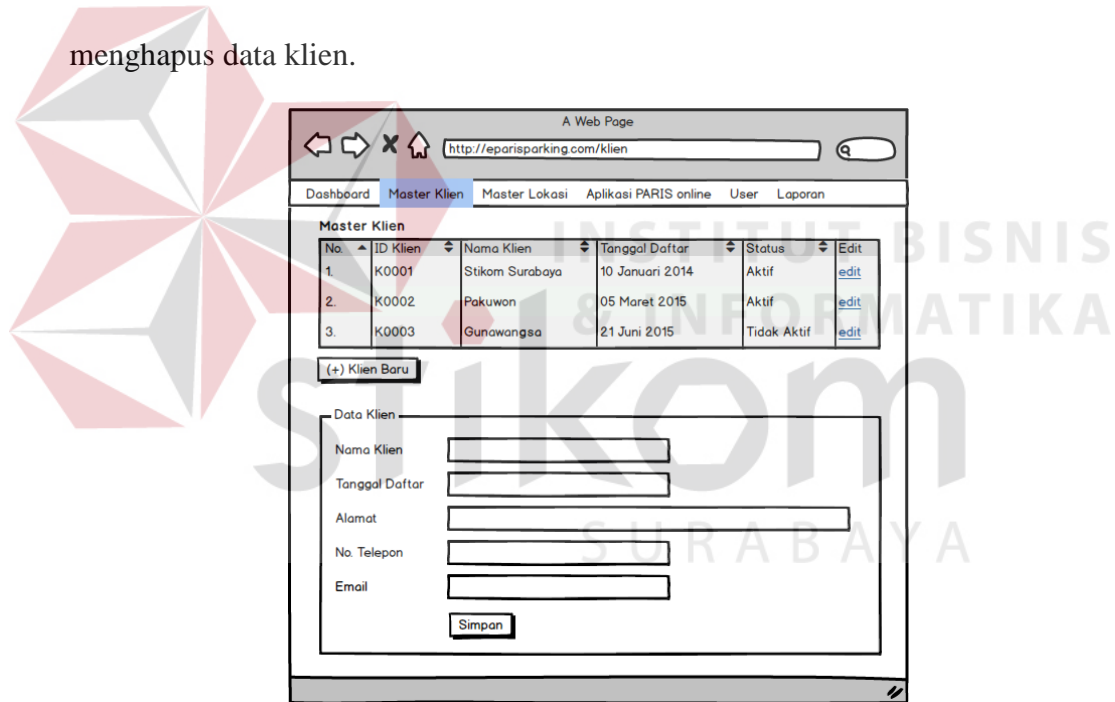
Halaman *login* digunakan untuk keamanan sistem. *User* klien atau *user* SSI akan melakukan *login* dengan memilih nama klien yang terdaftar di sistem kemudian memasukkan id *user* dan *password*. Jika data yang dimasukkan valid maka halaman akan berpindah ke halaman *dashboard* Paris Manajemen Klien. Jika *login* gagal, sistem akan menampilkan jika data yang dimasukkan salah.



Gambar 3. 13 Desain *Login* Paris Manajemen Klien

## 2. Membuat Master Klien

Halaman ini digunakan oleh SSI untuk menambah, merubah, atau menghapus data klien.

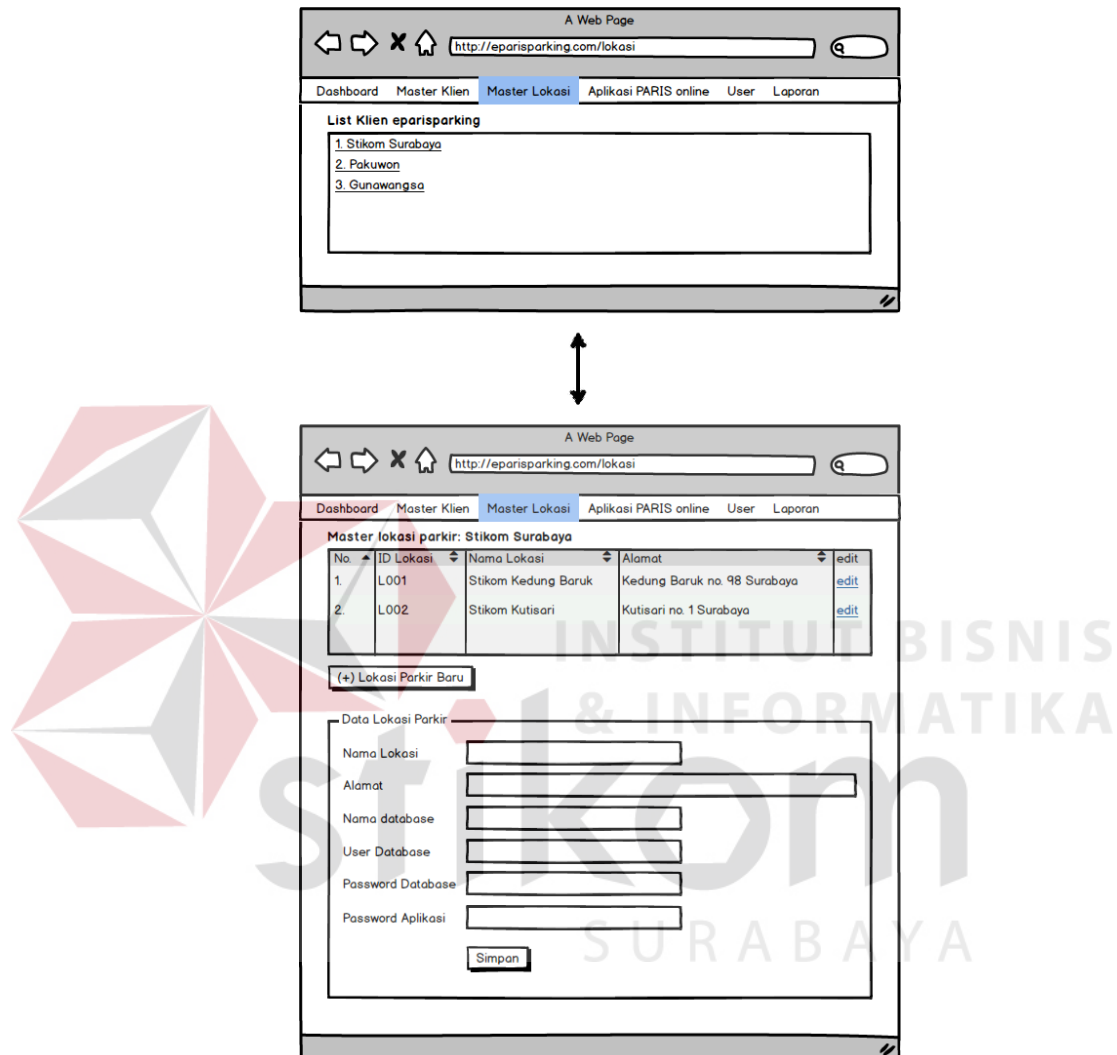


Gambar 3. 14 Desain Membuat Master Klien

## 3. Membuat Master Lokasi Parkir

Halaman ini digunakan untuk menambah, merubah, atau menghapus data lokasi parkir klien. Dimulai dengan sistem yang menampilkan daftar klien yang terdaftar di sistem, kemudian SSI memilih klien yang akan ditambah atau diubah

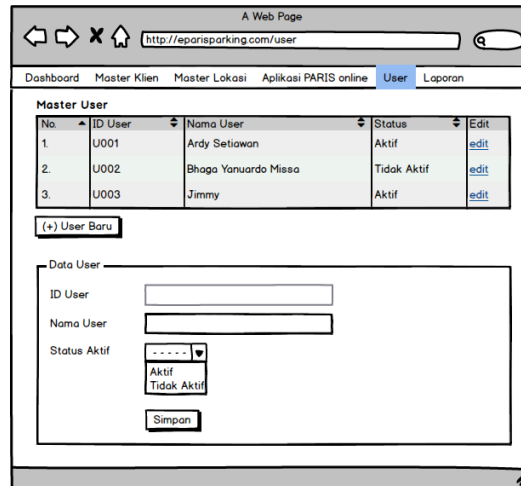
datanya maka sistem akan menampilkan daftar lokasi parkir dari klien yang telah dipilih. Terdapat tombol tambah, *edit* atau hapus untuk melakukan proses-proses tersebut.



Gambar 3. 15 Desain Membuat Master Lokasi Parkir

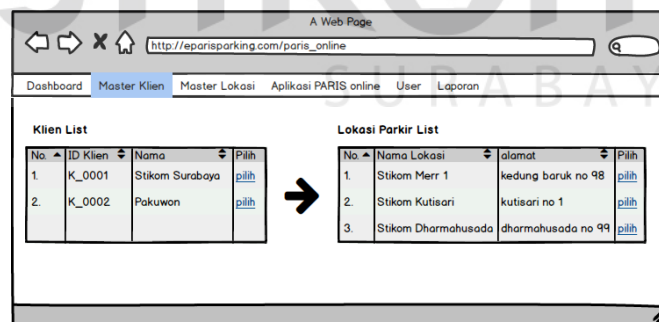
#### 4. Membuat Master *User*

Halaman ini digunakan oleh SSI untuk menambah, merubah, atau menghapus data *user* SSI yang dapat masuk ke aplikasi Paris Manajemen Klien.

Gambar 3. 16 Desain Membuat Master *User*

## 5. Menjalankan Paris *Online*

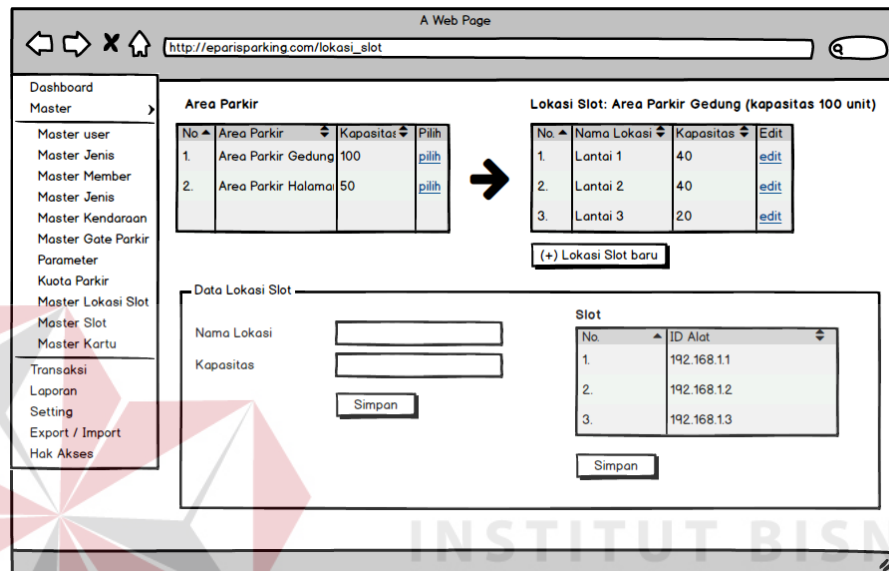
Menjalankan paris *online* digunakan klien atau SSI untuk masuk ke Paris-*paris online*. Pertama yang harus dilakukan adalah memilih klien dari daftar klien yang ditampilkan sistem, setelah dipilih akan muncul daftar lokasi parkir dari klien tersebut. Pilih lokasi parkir yang diinginkan dan halaman *website* akan berpindah ke Paris *online* lokasi yang telah dipilih.

Gambar 3. 17 Desain Menjalankan Paris *Online*

## 6. Membuat Master *Slot* Parkir

Halaman ini digunakan untuk menambah, merubah, atau menghapus data slot parkir. Dimulai dengan sistem yang menampilkan daftar area parkir yang

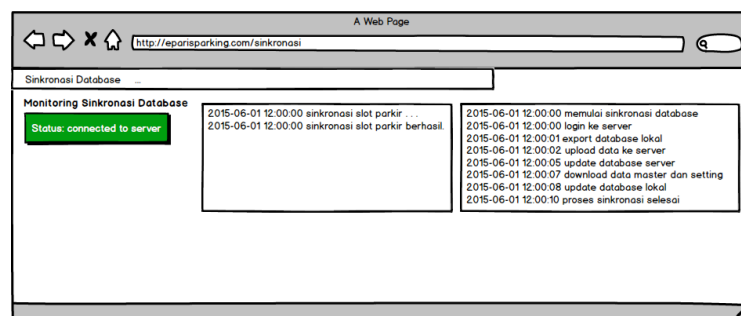
terdaftar di sistem, kemudian SSI memilih area parkir yang akan ditambah atau diubah datanya maka sistem akan menampilkan daftar lokasi slot dari area yang telah dipilih. Terdapat tombol tambah, *edit* atau hapus untuk melakukan proses-proses tersebut.



Gambar 3. 18 Desain Membuat Master *Slot* Parkir

## 7. Sinkronisasi *Database* dan Sinkronisasi *Slot* Parkir

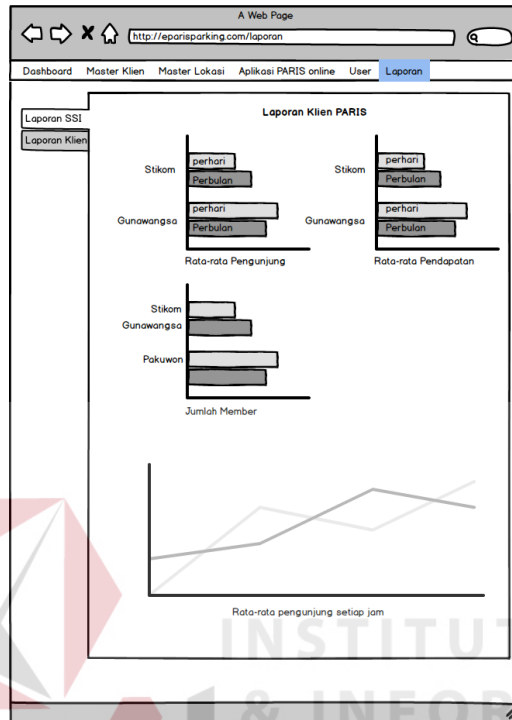
Sinkronisasi *database* dan sinkronisasi *slot* parkir adalah menu tambahan di Paris *offline*, tampilannya akan seperti gambar dibawah ini. Terdapat 2 teks area yang menampilkan *history* sinkronisasi *database* dan sinkronisasi slot parkir.



Gambar 3. 19 Desain sinkronisasi *database* dan sinkronisasi *slot* parkir

## 8. Laporan Pendapatan dan Kehilangan Klien

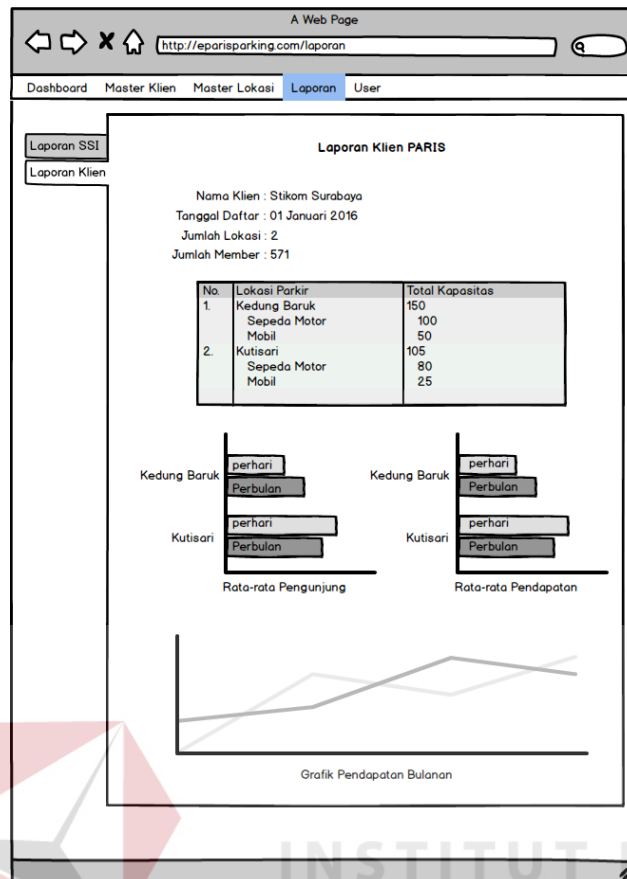
Laporan pendapatan dan kehilangan klien akan terdapat 3 informasi yaitu transaksi, pendapatan, dan kehilangan. Desainnya akan tampak seperti berikut:



Gambar 3. 20 Desain Laporan Pendapatan Klien

## 9. Laporan Pendapatan dan Kehilangan Lokasi Parkir

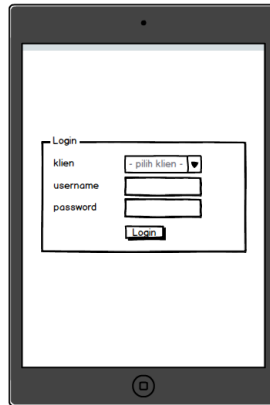
Laporan pendapatan dan kehilangan lokasi parkir akan terdapat 4 informasi yaitu transaksi, pendapatan, kehilangan, dan *member*. Desainnya akan tampak seperti berikut:



Gambar 3. 21 Desain Laporan Pendapatan Lokasi Parkir

#### 10. *Login Paris Mobile*

*Login Paris mobile* ini digunakan untuk memberikan hak akses untuk *member* parkir yang terdaftar di Paris untuk mendapatkan informasi ketersediaan *slot* parkir di lokasi-lokasi parkir Paris.



Gambar 3. 22 Desain *Login Paris Mobile*



Gambar 3. 23 Desain informasi Ketersediaan *Slot* Parkir

### 3.2.9 Rancangan Pengujian dan evaluasi

Rancangan uji coba bertujuan untuk memastikan bahwa aplikasi telah dibuat dengan benar sesuai kebutuhan atau tujuan yang diharapkan. Kekurangan atau kelemahan sistem pada tahap ini akan dievaluasi sebelum diimplementasikan. Proses pengujian menggunakan *Black Box Testing* yaitu aplikasi akan diuji dengan melakukan berbagai percobaan untuk membuktikan bahwa aplikasi yang



telah dibuat sudah sesuai dengan tujuan. Berikut ini adalah tabel pengujian yang akan dilakukan untuk menguji aplikasi:

Tabel 3. 20 Tabel Pengujian

Requirement yang diuji	Fungsi yang diuji
Master lokasi <i>slot</i>	<ol style="list-style-type: none"> <li>1. Tambah lokasi <i>slot</i></li> <li>2. <i>Edit</i> lokasi <i>slot</i></li> <li>3. Hapus lokasi <i>slot</i></li> <li>4. Simpan id perangkat</li> </ol>
Sinkronisasi <i>database</i>	<ol style="list-style-type: none"> <li>1. Sinkronisasi <i>database</i></li> </ol>
Sinkronisasi <i>slot</i> parkir	<ol style="list-style-type: none"> <li>1. Sinkronisasi <i>slot</i> parkir</li> </ol>
Paris <i>mobile</i>	<ol style="list-style-type: none"> <li>1. <i>Login</i> Paris <i>Mobile</i></li> </ol>
<i>Login</i> Paris Manajemen Klien	<ol style="list-style-type: none"> <li>1. <i>login user</i> klien</li> <li>2. <i>login user</i> SSI</li> </ol>
Membuat Master klien	<ol style="list-style-type: none"> <li>1. Tambah klien baru</li> <li>2. <i>Edit</i> klien</li> </ol>
Membuat Master Lokasi Parkir	<ol style="list-style-type: none"> <li>1. Tambah lokasi parkir</li> <li>2. <i>Edit</i> lokasi parkir</li> </ol>
Membuat Master <i>User</i>	<ol style="list-style-type: none"> <li>1. Tambah <i>user</i> baru</li> <li>2. <i>Edit user</i></li> </ol>
Menjalankan Paris <i>online</i>	<ol style="list-style-type: none"> <li>1. Menjalankan Paris <i>online</i></li> </ol>
Buat Laporan	<ol style="list-style-type: none"> <li>1. Laporan pendapatan dan kehilangan klien</li> <li>2. Laporan pendapatan dan</li> </ol>

	kehilangan lokasi parkir
--	--------------------------

### A. Desain Uji Coba Membuat Master Lokasi *Slot*

Desain uji coba ini bertujuan untuk menguji apakah fungsi membuat master lokasi *slot* dapat berjalan sesuai yang diharapkan. Desain uji coba master lokasi *slot* dilihat pada Tabel 3.21.

Tabel 3. 21 Tabel Uji Coba Master Lokasi *Slot*

<i>Test Case ID</i>	Tujuan	<i>Input</i>	Output yang diharapkan	Status
1.	Tambah <i>slot</i> parkir	Nama lokasi <i>slot</i> , jumlah <i>slot</i>	Data berhasil disimpan	
2.	Mencoba menambah <i>slot</i> tanpa isi nama dan jumlah <i>slot</i>	-	Harap lengkapi form	
3.	<i>Edit</i> lokasi <i>slot</i>	Nama lokasi <i>slot</i> , jumlah <i>slot</i>	Data berhasil disimpan	
4.	Hapus lokasi <i>slot</i>	Klik tombol hapus pada lokasi yang ingin dihapus	Data lokasi <i>slot</i> berhasil dihapus	
5.	Simpan id perangkat	Id perangkat masing-masing <i>slot</i>	Data berhasil disimpan	

### B. Sinkronisasi *Database*

Desain uji coba ini bertujuan untuk menguji apakah fungsi sinkronisasi *database* dapat berjalan sesuai yang diharapkan. Desain uji coba sinkronisasi *database* dilihat pada Tabel 3.22.

Tabel 3. 22 Tabel Uji Coba Sinkronisasi *Database*

<i>Test Case ID</i>	Tujuan	<i>Input</i>	Output yang diharapkan	Status
---------------------	--------	--------------	------------------------	--------

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1.	Sinkronisasi <i>database</i>	-	Sinkronisasi <i>database</i> berhasil	

### C. Sinkronisasi Slot Parkir

Desain uji coba ini bertujuan untuk menguji apakah fungsi sinkronisasi *slot* parkir dapat berjalan sesuai yang diharapkan. Desain uji coba sinkronisasi *slot* parkir dilihat pada Tabel 3.23.

Tabel 3. 23 Tabel Uji Coba Sinkronisasi *Database*

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1.	Sinkronisasi <i>slot</i> parkir	-	Sinkronisasi data <i>slot</i> berhasil	

### D. Paris Mobile

Desain uji coba ini bertujuan untuk menguji apakah fungsi Paris *mobile* dapat berjalan sesuai yang diharapkan. Desain uji coba Paris *mobile* dilihat pada Tabel 3.24.

Tabel 3. 24 Tabel Uji Coba Paris *Mobile*

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1.	<i>Login Paris Mobile</i>	Id klien, id member, dan password	Masuk ke aplikasi Paris <i>Mobile</i> dan menampilkan Informasi <i>slot</i> kosong	
2.	<i>Login Paris Mobile</i> dengan password yang salah	Id klien, id member, password salah	Data tidak ditemukan atau password salah	

### E. *Login Paris Manajemen Klien*

Desain uji coba ini bertujuan untuk menguji apakah fungsi *login* Paris Manajemen Klien dapat berjalan sesuai yang diharapkan. Desain uji coba *login* Paris Manajemen Klien dapat dilihat pada Tabel 3.25.

Tabel 3. 25 Tabel Uji Coba *Login* Paris Manajemen Klien

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1.	<i>Login user klien</i>	Klien id, <i>user id</i> , dan <i>password</i>	Masuk ke aplikasi Paris Manajemen Klien	
2.	<i>Login user klien dengan password salah</i>	Klien id, <i>user id</i> , <i>password</i>	<i>Username</i> atau <i>password</i> tidak valid, silahkan masukkan kembali	
3.	<i>Login user SSI</i>	<i>user id</i> , dan <i>password</i>	Masuk ke aplikasi Paris Manajemen Klien	
4.	<i>Login user SSI dengan password salah</i>	<i>user id</i> , <i>password</i>	<i>Username</i> atau <i>password</i> tidak valid, silahkan masukkan kembali	

### F. *Membuat Master Klien*

Desain uji coba ini bertujuan untuk menguji apakah fungsi membuat membuat master klien dapat berjalan sesuai yang diharapkan. Desain uji coba membuat master klien dapat dilihat pada Tabel 3.26.

Tabel 3. 26 Tabel Uji Coba Membuat Master Klien

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1.	Tambah klien baru	Nama klien, tanggal daftar, dan status aktif	Data berhasil disimpan	
2.	Tambah klien baru tanpa memasukkan nama	Tanggal daftar dan status aktif	Data belum lengkap, mohon dicek dan dilengkapi terlebih dahulu	
3.	<i>Edit data klien</i>	Nama klien, tanggal daftar, dan status aktif	Data berhasil disimpan	
4.	<i>Edit klien tanpa memasukkan nama</i>	Tanggal daftar dan status aktif	Data belum lengkap, mohon dicek dan dilengkapi terlebih dahulu	

#### G. Membuat Master Lokasi Parkir

Desain uji coba ini bertujuan untuk menguji apakah fungsi membuat master lokasi parkir dapat berjalan sesuai yang diharapkan. Desain uji coba membuat master lokasi parkir dapat dilihat pada Tabel 3.27.

Tabel 3. 27 Tabel Uji Coba Membuat Master Lokasi Parkir

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1.	Tambah lokasi parkir baru	Nama lokasi parkir, alamat, <i>password</i> sinkronisasi <i>database</i> , nama <i>database</i> , <i>user database</i> , <i>password</i>	Data berhasil disimpan	

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
		<i>database</i>		
2.	Tambah lokasi parkir baru tanpa memasukkan nama	Alamat, <i>password database</i> , nama <i>database</i> , <i>user database</i> , <i>password database</i>	Data belum lengkap, mohon dicek dan dilengkapi terlebih dahulu	
3.	<i>Edit</i> lokasi parkir	Nama lokasi parkir, alamat, <i>password database</i> , nama <i>database</i> , <i>user database</i> , <i>password database</i>	Data berhasil disimpan	
4.	<i>Edit</i> lokasi parkir tanpa memasukkan nama lokasi parkir	Alamat, <i>password database</i> , nama <i>database</i> , <i>user database</i> , <i>password database</i>	Data belum lengkap, mohon dicek dan dilengkapi terlebih dahulu	

#### H. Membuat Master *User*

Desain uji coba ini bertujuan untuk menguji apakah fungsi membuat master *user* dapat berjalan sesuai yang diharapkan. Desain uji coba membuat master *user* dapat dilihat pada Tabel 3.28.

Tabel 3. 28 Tabel Uji Coba Membuat Master *User*

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1.	Tambah <i>user</i> baru	Nama <i>user</i> , <i>password</i> , dan status aktif	Data berhasil disimpan	
2.	Tambah <i>user</i> baru tanpa memasukkan	<i>Password</i> dan	Data belum lengkap, mohon	

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
	nama	status aktif	dicek dan dilengkapi terlebih dahulu	
3.	<i>Edit data user</i>	Nama <i>user</i> , <i>password</i> , dan status aktif	Data berhasil disimpan	
4.	<i>Edit user</i> tanpa memasukkan nama	<i>Password</i> dan status aktif	Data belum lengkap, mohon dicek dan dilengkapi terlebih dahulu	

### I. Menjalankan Paris Online

Desain uji coba ini bertujuan untuk menguji apakah fungsi menjalankan Paris *online* dapat berjalan sesuai yang diharapkan. Desain uji coba menjalankan Paris *online* dapat dilihat pada Tabel 3.29.

Tabel 3. 29 Tabel Uji Coba Menjalankan Paris *Online*

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1.	Menjalankan Paris <i>online</i>	Klik lokasi parkir yang ingin dikunjungi Paris <i>onlinenya</i>	Masuk ke aplikasi Paris <i>online</i> yang dituju	

### J. Evaluasi Kesesuaian Hasil Akhir Sistem

Evaluasi kesesuaian hasil akhir sistem adalah sebuah pengujian untuk mengetahui apakah sistem yang dibangun telah memenuhi kebutuhan-kebutuhan yang terdapat pada latar belakang permasalahan. Sistem yang dibangun juga harus mengatasi permasalahan-pemersalahan yang telah diungkapkan pada latar belakang. Desain uji coba kesesuaian hasil akhir sistem dapat dilihat pada Tabel

3.31.

Tabel 3. 30 Tabel Evaluasi Sistem

<i>No.</i>	<b>Kebutuhan</b>	<i>Hasil</i>	<b>Bukti</b>
1.	Sistem dapat mengelola klien-klien Paris		
2.	Sistem dapat mengelola lokasi-lokasi parkir Paris		
3.	Sistem dapat menginformasikan ketersediaan <i>slot</i> parkir di suatu lokasi Paris		

