

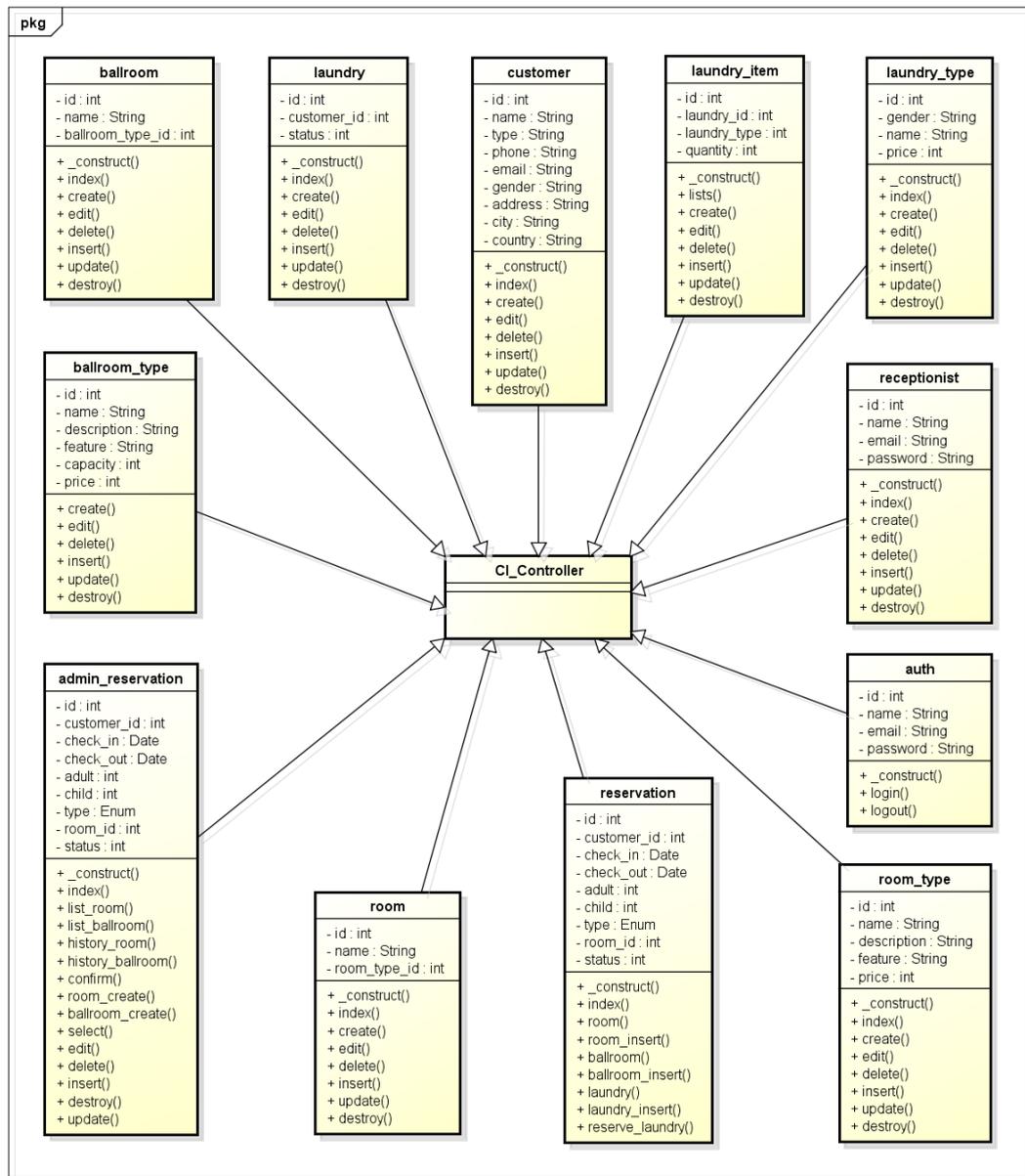
## BAB 5 PERANCANGAN SISTEM

Pada bab ini berisi perancangan dari sistem yang akan dibangun, perancangan sistem ini terdiri atas perancangan *design-level class diagram*, *sequence diagram*, perancangan *class diagram*, *relational data model*, perancangan komponen dan perancangan antarmuka.

### 5.1 Design-Level Class Diagram

Pada *level* perancangan, *class diagram* merepresentasikan *class*, atribut, fungsi dan hubungan antar *class* pada suatu perangkat lunak. Setelah dilakukan *analysis-level class diagram*, *class diagram* dilengkapi dengan fungsi di tiap *class* pada *class diagram*. Pada tahap perancangan, konsep *model view controller* (MVC) digunakan sesuai dengan konfigurasi *class controller* dan *model* di *framework* PHP CodeIgniter. Berdasarkan *class* yang sudah didefinisikan pada tahap *analysis-level class diagram*, dibentuk *class-class* baru yang terhubung dengan *controller* dan *model* milik CodeIgniter.

Gambar 5.1 merupakan *design-level class diagram* untuk *controller* yang ada di dalam sistem. Masing-masing *class* berisi fungsi yang siap dipanggil untuk memanggil *view* dan *model* yang ada di dalam sistem. Semua *class* memiliki hubungan *generalization* dengan *class* *CI\_Controller* agar dapat mengakses konfigurasi *controller* yang telah disediakan di *class* *CI\_Controller*.



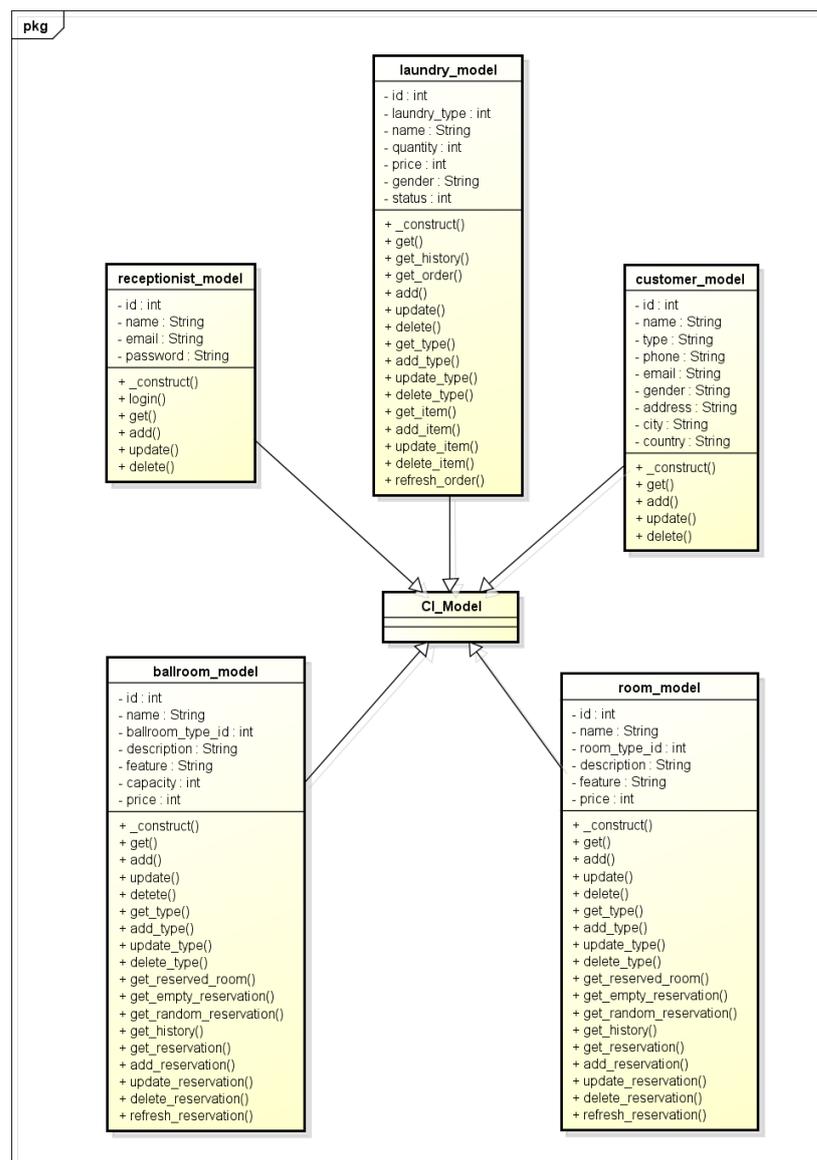
powered by Astah

**Gambar 5.1 Design-Level Class Diagram Controller**

Pada Gambar 5.1 menjelaskan bahwa *Design-Level Class Diagram Controller* untuk implementasi sistem *Customer Relationship Management (CRM) Self-service Reservation* memiliki 12 *class controller* yang terdiri dari *class laundry\_type, laundry\_item* dan *laundry* berfungsi untuk pemesanan *laundry*. Selanjutnya pada *class customer* digunakan untuk mendaftarkan setiap pelanggan yang telah melakukan pemesanan. Kemudian pada *class reservation* digunakan untuk melakukan setiap pemesanan *ballroom* dan *kamar*, sedangkan untuk *class admin\_reservation* digunakan untuk fungsi konfirmasi pemesanan yang dilakukan oleh *receptionist*. Selanjutnya pada *class controller room* dan *room\_type* berfungsi untuk mengatur pemesanan kamar yang tersedia pada jadwal tertentu. Kemudian pada *class auth* dan *receptionist* digunakan untuk hak akses yang dilakukan

*receptionist* untuk masuk ke dalam sistem. Masing-masing *controller* terhubung dengan *CI\_Controller*.

Kemudian Gambar 5.2 merupakan *design-level class diagram* untuk *model* yang ada di dalam sistem. Masing-masing *class* berisi fungsi yang siap dipanggil untuk menjalankan *query* yang berfungsi untuk mengakses data yang ada di dalam *database*. Data yang didapatkan melalui *query* dikirim kembali ke *controller* dan dapat ditampilkan dalam *view*. Semua *class* memiliki hubungan *generalization* dengan *class CI\_Model* agar dapat mengakses konfigurasi *model* yang telah disediakan di *class CI\_Model*

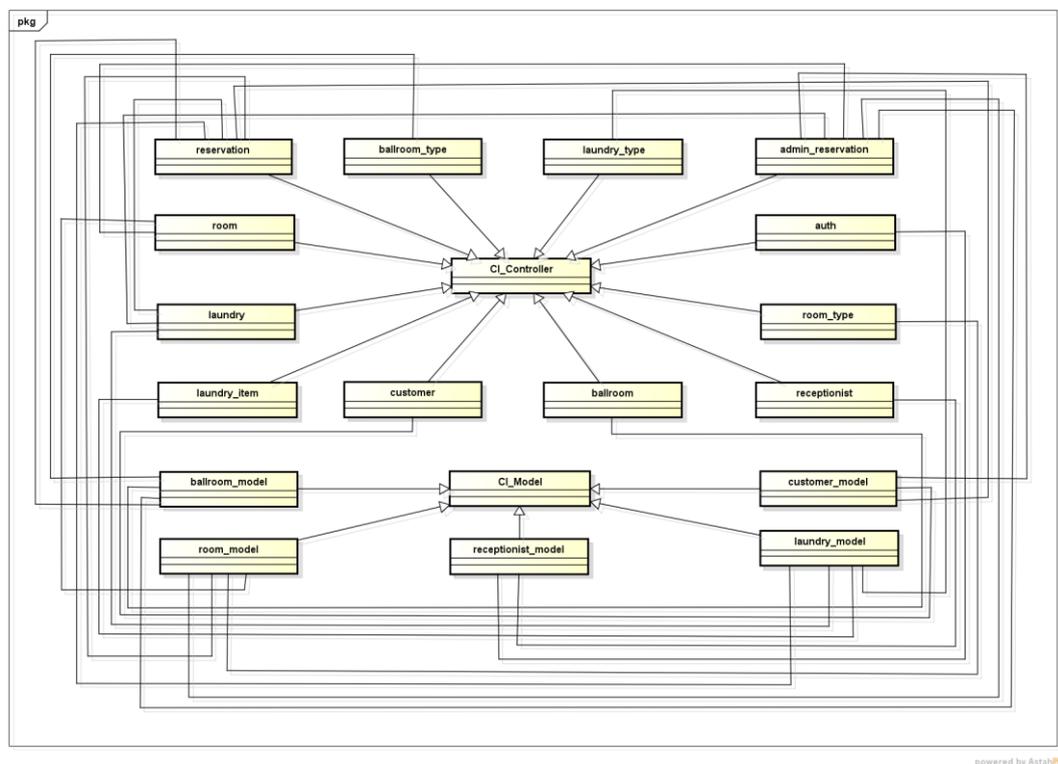


**Gambar 5.2 Design Level Class Diagram Model**

Pada Gambar 5.2 menjelaskan bahwa *Design-Level Class Diagram Model* untuk implementasi sistem *Customer Relationship Management (CRM) Self-service Reservation* memiliki 5 *class model* yang terdiri dari *class laundry\_model* berfungsi

untuk pemesanan *laundry*. Selanjutnya pada *class* *customer\_model* digunakan untuk mendata setiap pelanggan yang telah melakukan pemesanan. Kemudian pada *class* *room\_model* berfungsi untuk mengatur pemesanan kamar yang tersedia pada jadwal tertentu. Selanjutnya pada *class* *ballroom\_model* berfungsi untuk mengatur jadwal *ballroom* yang tersedia. Kemudian pada *class* *receptionist\_model* digunakan untuk hak akses yang dilakukan *receptionist* untuk masuk ke dalam sistem. Masing-masing *controller* terhubung dengan *CI\_Model*.

Gambar 5.3 merupakan *class diagram* keseluruhan yang menggambarkan hubungan antara *controller* dan *model* yang ada di dalam sistem. Satu *controller* dapat mengakses beberapa *model* yang ada di dalam sistem. Semua *class controller* memiliki hubungan *generalization* dengan *CI\_Controller*. Semua *class model* memiliki hubungan *generalization* dengan *CI\_Model*.



**Gambar 5.3 Domain Class Diagram Sistem Self-service Reservation**

Gambar 5.3 merupakan *domain class diagram* keseluruhan yang menggambarkan hubungan antara *controller* dan *model* yang ada di dalam sistem untuk pengembangan *Sistem Customer Relationship Management (CRM) Self-service Reservation* pada Everyday Smart Hotel Malang. Setiap *controller* yang dapat mengakses beberapa *model* yang ada di dalam sistem. Semua *class controller* memiliki hubungan *generalization* dengan *CI\_Controller*. Semua *class model* memiliki hubungan *generalization* dengan *CI\_Model*.

## 5.2 Perancangan Perangkat Lunak

Pada tahap perancangan dilakukan beberapa hal yang akan digunakan sebagai acuan dalam proses implementasi, yakni pembuatan diagram-diagram

perancangan perangkat lunak yang meliputi *sequence diagram*, *diagram class*, *relational data model* dan perancangan antarmuka pengguna.

### 5.2.1 Perancangan Arsitektur

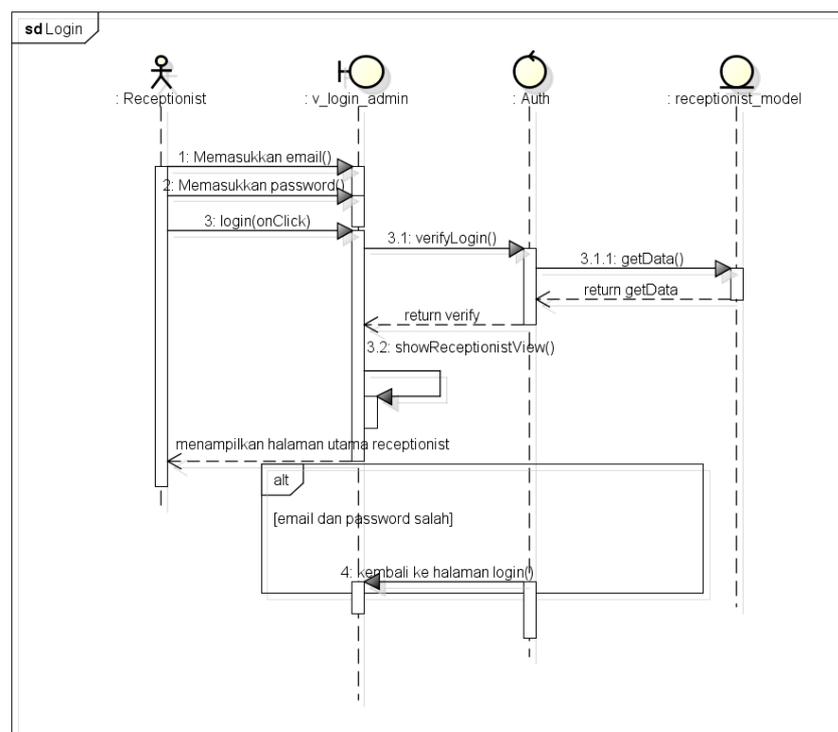
Perancangan arsitektur adalah tahap pertama dalam proses perancangan perangkat lunak. Perancangan arsitektur berkaitan dengan bagaimana sistem harus terorganisir dengan cara merancang struktur keseluruhan sistem. *Output* dari perancangan arsitektur adalah *class diagram* berdasarkan dari identifikasi objek menggunakan *sequence diagram*

#### 5.2.1.1 Perancangan Sequence Diagram

Perancangan *sequence diagram* dibuat berdasarkan pada masing-masing skenario *use-case* yang telah didefinisikan pada saat proses analisis kebutuhan fungsional perangkat lunak sistem *Customer Relationship Management (CRM) Self-service reservation* pada Everyday Smart Hotel Malang

##### 1. Sequence Diagram Login

*Sequence diagram* ini digunakan *receptionist* untuk memastikan atau validasi masuk ke dalam hak akses sistem *receptionist*



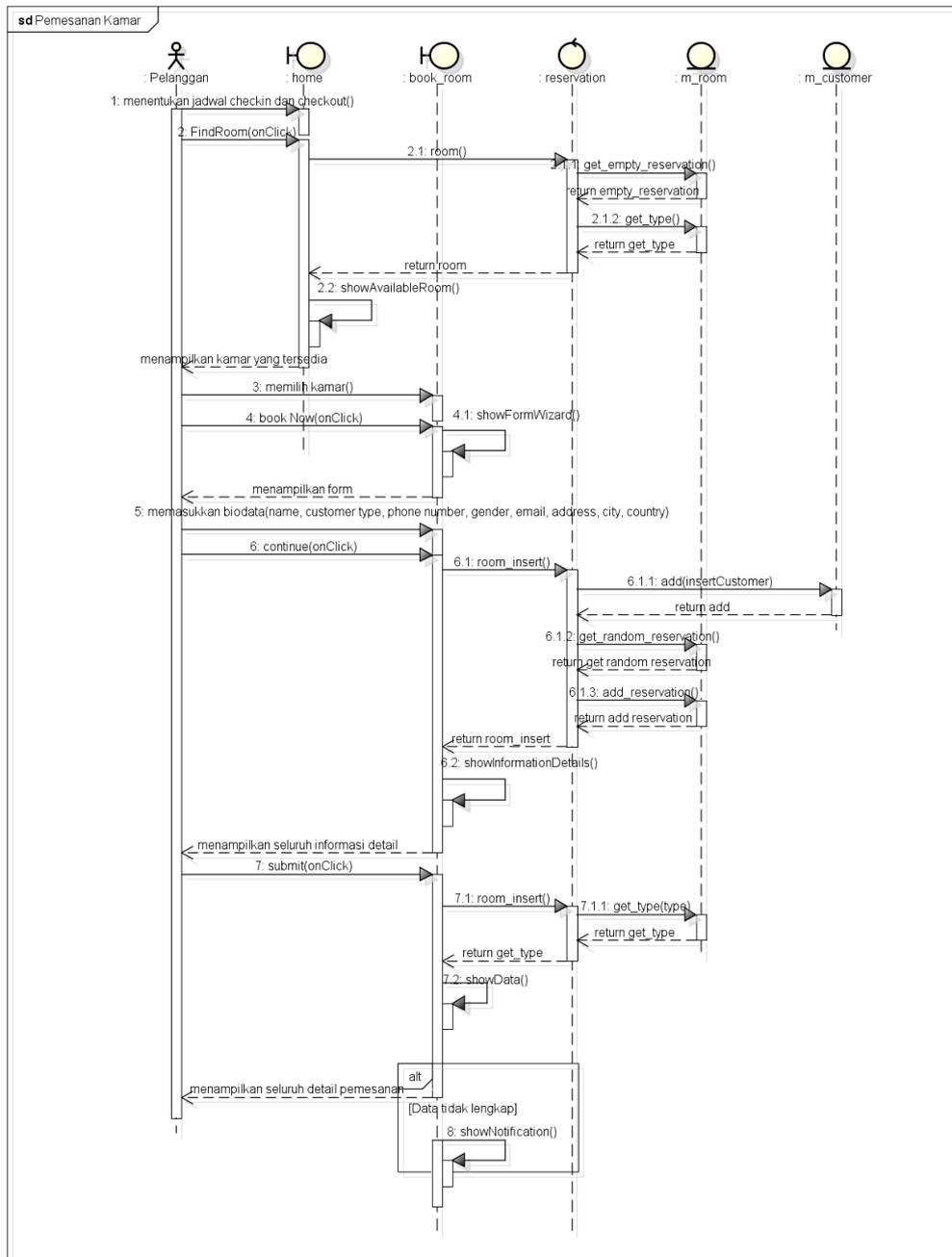
powered by Astah

**Gambar 5.4 Sequence Diagram Login**

Gambar 5.4 menjelaskan bahwa proses login yang dilakukan oleh *receptionist* untuk masuk ke dalam sistem, yang terdapat *controller* yang diinisialisasi dengan *Auth* dan menggunakan model *receptionist\_model*.

## 2. Sequence Diagram Pemesanan Kamar

Sequence diagram ini digunakan oleh pelanggan untuk melakukan pemesanan kamar beserta jadwal yang diinginkan dengan menggunakan sistem *Customer Relationship Management (CRM) Self-service reservation*



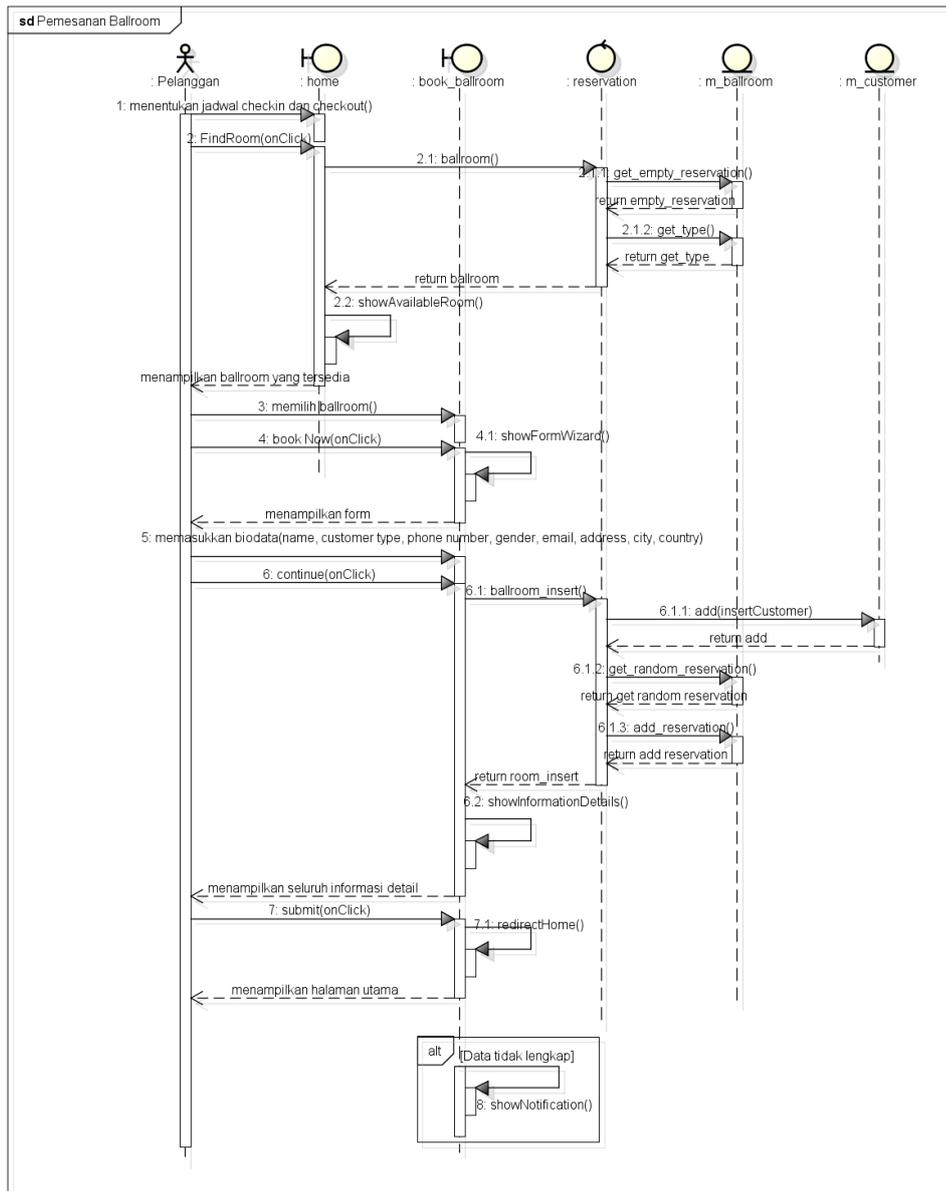
Gambar 5.5 Sequence Diagram Pemesanan Kamar

Gambar 5.5 menjelaskan bahwa proses pemesanan kamar yang dilakukan oleh pelanggan menggunakan fungsi `room` yang terdapat pada `controller reservation` dari view `home` setelah pelanggan menekan tombol `Find Room`, kemudian pada `controller reservation` melakukan pengecekan kamar yang tersedia pada model

m\_room dan pelanggan menambahkan biodata menggunakan fungsi room\_insert dan melakukan validasi pemesanan dengan menampilkan informasi detail pemesanan.

### 3. Sequence Diagram Pemesanan Ballroom

Sequence diagram ini digunakan oleh pelanggan untuk melakukan pemesanan ballroom beserta jadwal yang diinginkan dengan menggunakan sistem Customer Relationship Management (CRM) Self-service reservation.



powered by Astah

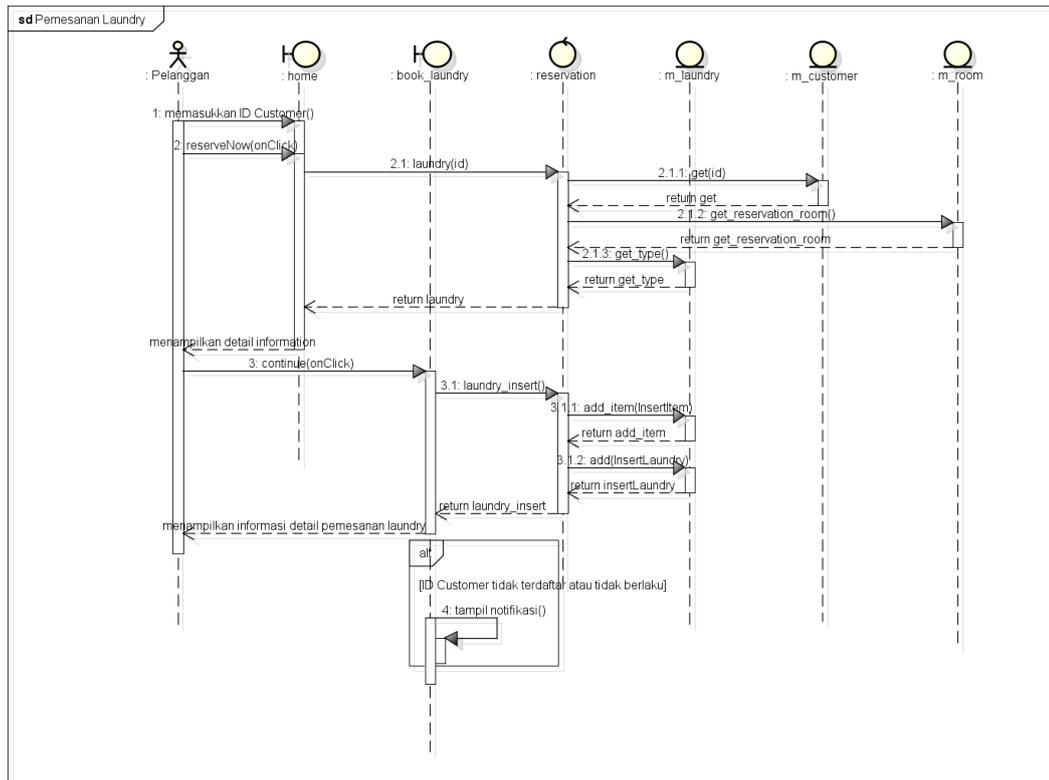
**Gambar 5.6 Sequence Diagram Pemesanan Ballroom**

Gambar 5.6 menjelaskan bahwa proses pemesanan ballroom yang dilakukan oleh pelanggan menggunakan fungsi ballroom yang terdapat pada controller reservation dari view home setelah pelanggan menekan tombol Find Room, kemudian pada controller reservation melakukan pengecekan ballroom yang

tersedia pada model *m\_ballroom* dan pelanggan menambahkan biodata menggunakan fungsi *ballroom\_insert*.

#### 4. Sequence Diagram Pemesanan Laundry

*Sequence diagram* ini digunakan oleh pelanggan untuk melakukan pemesanan *laundry* beserta jadwal yang diinginkan dengan menggunakan sistem *Customer Relationship Management (CRM) self-service reservation*



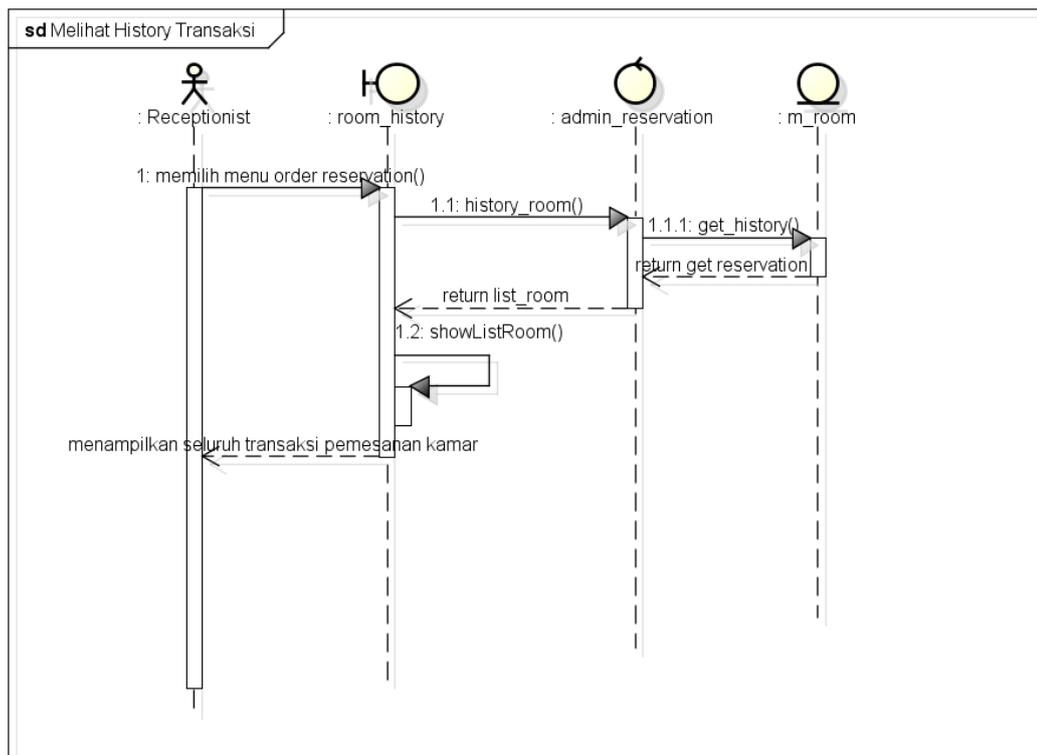
powered by Astah

**Gambar 5.7 Sequence Diagram Pemesanan Laundry**

Gambar 5.7 menjelaskan bahwa proses pemesanan *laundry* yang dilakukan oleh *receptionist* menggunakan fungsi *laundry* yang dapat memvalidasi ID *Customer* yang terdaftar pada sistem, kemudian melakukan validasi ke model *m\_customer*, model *m\_room* dan model *m\_laundry* yang terdapat pada *controller reservation*. Kemudian pelanggan melakukan pemilihan *item* yang dipilih sesuai kebutuhan pelanggan, sistem menampilkan biaya atau harga setiap jenis *laundry* yang tersedia. Selanjutnya sistem dapat melakukan perhitungan secara otomatis sesuai jumlah *laundry* yang dipilih oleh pelanggan dan melakukan konfirmasi bahwa pemesanan *laundry* telah dilakukan.

#### 5. Sequence Diagram Melihat History Transaksi Pemesanan

*Sequence diagram* ini digunakan oleh *receptionist* untuk melihat *history* transaksi pemesanan beserta data pelanggan dan jumlah kamar yang telah dikonfirmasi pada sistem



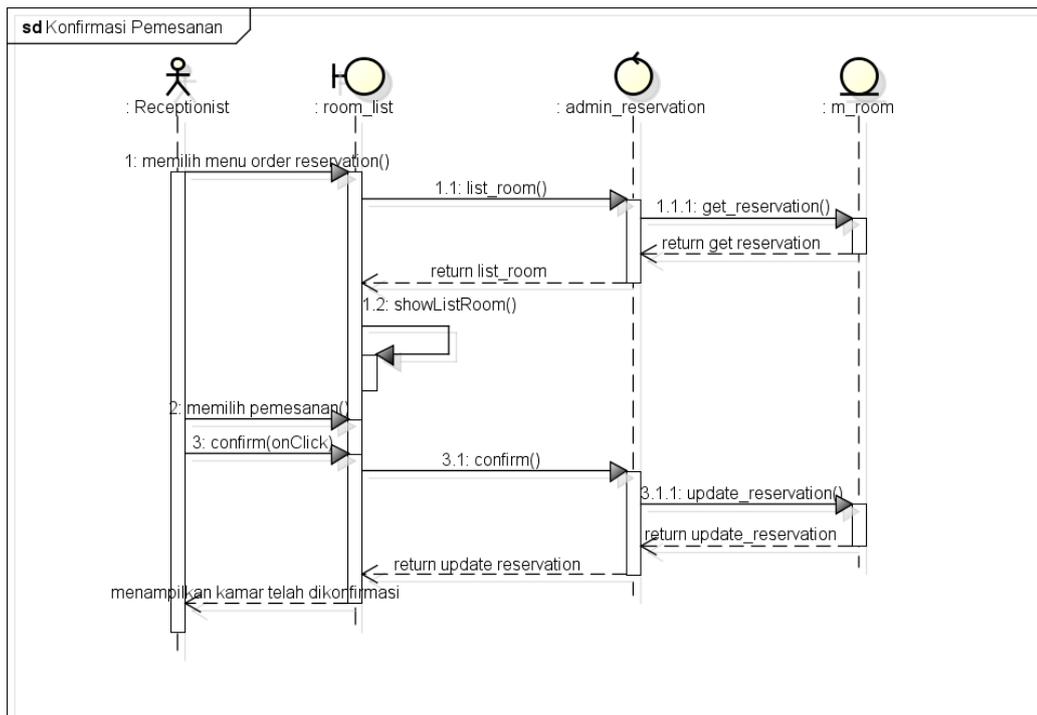
powered by Astah

**Gambar 5.8 Sequence Diagram Melihat History Transaksi Pemesanan**

Gambar 5.8 menjelaskan bahwa proses melihat *history* transaksi pemesanan yang dimulai dari *receptionist* memilih menu *reservation* dengan menggunakan fungsi *history\_room* yang terdapat pada *controller admin\_reservation* dan mengambil data *history* transaksi pada model *m\_room* untuk melihat kamar yang telah dipesan pada jadwal tertentu beserta data pelanggannya.

#### 6. Sequence Diagram Konfirmasi Pemesanan

*Sequence diagram* ini digunakan oleh *receptionist* untuk melakukan konfirmasi pemesanan beserta data pelanggan dan jumlah kamar yang akan dikonfirmasi pada sistem



powered by Astah

**Gambar 5.9 Sequence Diagram Konfirmasi Pemesanan**

Gambar 5.9 menjelaskan bahwa proses konfirmasi pemesanan yang dilakukan oleh *receptionist* dimulai dari *receptionist* memilih menu *reservation* dengan menggunakan fungsi *list\_room* yang terdapat pada *controller admin\_reservation* dan mengambil data *order* transaksi pada model *m\_room* untuk melihat kamar yang telah dipesan oleh pelanggan. Kemudian *receptionist* melakukan konfirmasi pemesanan menggunakan fungsi *confirm* yang terdapat pada *controller reservation* dan *update\_reservation* pada model *m\_room* untuk melakukan penambahan pemesanan baru.

## 5.2.2 Perancangan Komponen

Perancangan komponen mendefinisikan struktur data atau algoritma yang digunakan dalam suatu komponen perangkat lunak. Di dalam perancangan komponen ini hanya mencantumkan algoritma proses pemesanan kamar sehingga tidak semua algoritma dicantumkan. Algoritma pemesanan ini akan diletakkan pada *controller reservation*, *model room\_model*, *ballroom\_model*, *laundry\_model* dan *receptionist\_model* pada sistem *Customer Relationship Management (CRM) Self-service reservation* pada Everyday Smart Hotel Malang

### 5.2.2.1 Algoritma Method room()

Perancangan komponen fungsi *room()* merupakan algoritma dari fungsi sistem untuk melakukan pemesanan kamar. Fungsi *room()* merupakan bagian dari *controller reservation*. Tabel 5.1 merupakan algoritma dari fungsi *room()*.

**Tabel 5.1 Algoritma Method room()**

```
BEGIN room()
DB get checkin, checkout
data as new array old, roomlist, typelist, count
IF checkout<checkin || checkout==checkin
Set flash data error
FOREACH data typelist
data count type id = 0
FOREACH data roomlist
IF type id == room_type_id
data count id++
END IF
END FOR
END FOR
date_create as new checkin checkout
END IF
END room()
```

**5.2.2.2 Algoritma Method ballroom()**

Perancangan komponen fungsi *ballroom()* merupakan algoritma dari fungsi sistem untuk melakukan pemesanan *ballroom*. Fungsi *ballroom()* merupakan bagian dari *controller* reservation. Tabel 5.2 merupakan algoritma dari fungsi *ballroom()*.

**Tabel 5.2 Algoritma Method ballroom()**

```
BEGIN ballroom()
DB get checkin, checkout
data as new array old, roomlist, typelist, count
FOREACH data typelist
data count type id = 0
FOREACH data roomlist
IF type id == ballroom_type_id
data count id++
END IF
END FOR
END FOR
END ballroom_insert()
```

**5.2.2.3 Algoritma Method laundry()**

Perancangan komponen fungsi *laundry()* merupakan algoritma dari fungsi sistem untuk melakukan pemesanan *laundry*. Fungsi *laundry()* merupakan bagian dari *controller* reservation. Tabel 5.3 merupakan algoritma dari fungsi *laundry()*.

**Tabel 5.3 Algoritma Method Laundry()**

```
BEGIN laundry(id=null)
IF (id=null)
data error laundry
ELSE IF
Get reservation user
IF today room checkin || room checkout
redirect reservation laundry
END IF
END IF
END IF
END laundry()
```

### 5.2.2.4 Algoritma Method confirm()

Perancangan komponen fungsi confirm() merupakan algoritma dari fungsi sistem untuk melakukan konfirmasi pemesanan. Fungsi confirm() merupakan bagian dari *controller* reservation yang dimiliki *receptionist*. Tabel 5.4 merupakan algoritma dari fungsi confirm().

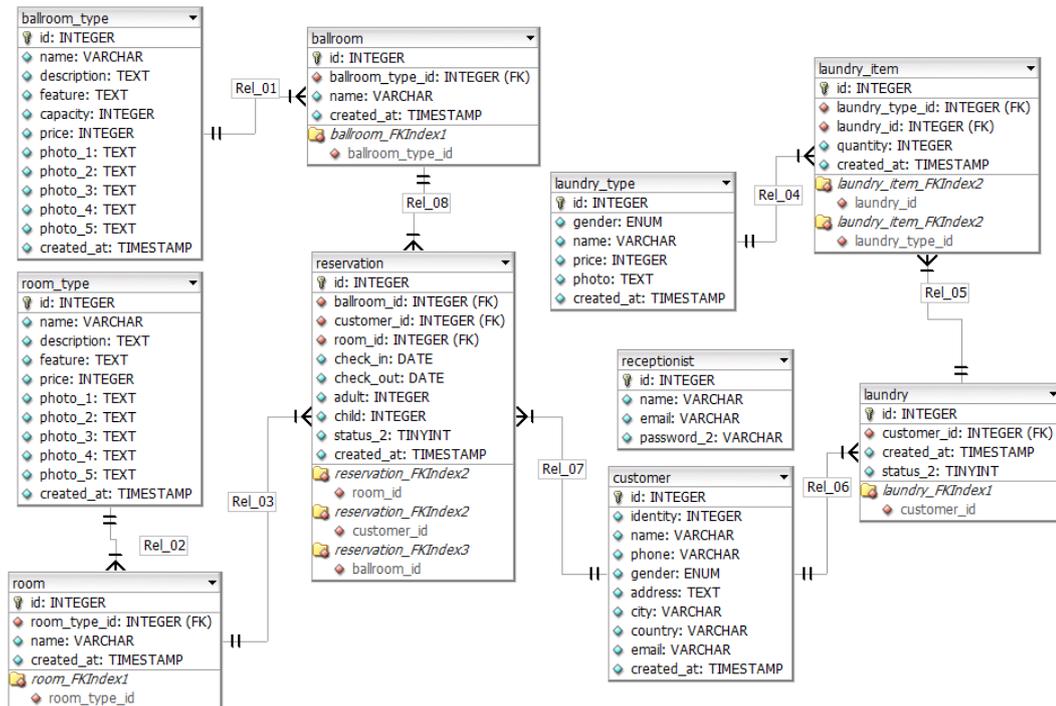
**Tabel 5.4 Algoritma Method confirm()**

```

BEGIN confirm(type, id)
  Data = array status => 1
  IF type ballroom
    Update reservation ballroom
  Else
    Update reservation room
  END IF
END confirm()
  
```

### 5.2.3 Perancangan Basis Data

Perancangan digunakan untuk membuat rancangan bagaimana data direkan dalam sistem untuk mendukung proses pemesanan. Berdasarkan diagram *class level analisis* yang telah dimodelkan sebelumnya, maka diperoleh rancangan basis data yang direpresentasikan dalam bentuk *relational data model* seperti ditunjukkan pada Gambar 5.10



**Gambar 5.10 Perancangan Basis Data**

Gambar 5.10 menjelaskan bahwa perancangan basis data digunakan untuk membuat rancangan yang akan diimplementasikan pada sistem *Customer Relationship Management (CRM) Self-service reservation* yang sesuai kebutuhan sistem. Melihat fasilitas dan melakukan pemesanan kamar memiliki tabel *room*

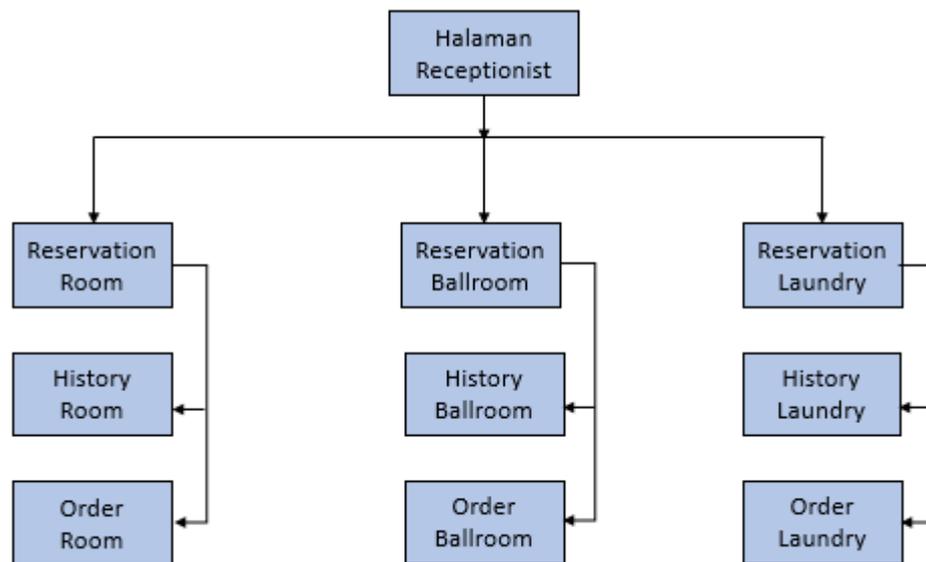
dan *room type*, sedangkan untuk melihat fasilitas dan melakukan pemesanan *ballroom* memiliki tabel *ballroom\_type* dan *ballroom*. Kemudian untuk melakukan pemesanan *laundry* beserta harganya pada tabel *laundry\_item* dan *laundry*. Selanjutnya pada tabel *customer* digunakan untuk pendataan pelanggan yang telah melakukan pemesanan dan terdaftar pada Everyday Smart Hotel Malang yang terhubung dengan tabel *reservation*. Kemudian pada tabel *receptionist* digunakan untuk hak akses masuk kedalam sistem yang dilakukan oleh *receptionist*.

### 5.3 Perancangan Sitemap

Perancangan *site map* adalah perancangan tampilan dari sebuah sistem yang terdiri dari struktur menu dari masing masing aktor. Untuk memudahkan membuat perancangan antarmuka, terlebih dahulu membuat struktur menu dari sistem tersebut. Berikut adalah struktur menu dari sistem *Customer Relationship Management (CRM) Self-service reservation*.

#### 1. Struktur Menu Receptionist

Struktur menu *receptionist* merupakan kumpulan menu-menu atau fungsi yang terdapat pada halaman *receptionist*, terdapat 3 fungsi utama yaitu *reservation room*, *reservation ballroom*, dan *reservation laundry*. Struktur menu *receptionist* ditunjukkan pada Gambar 5.11

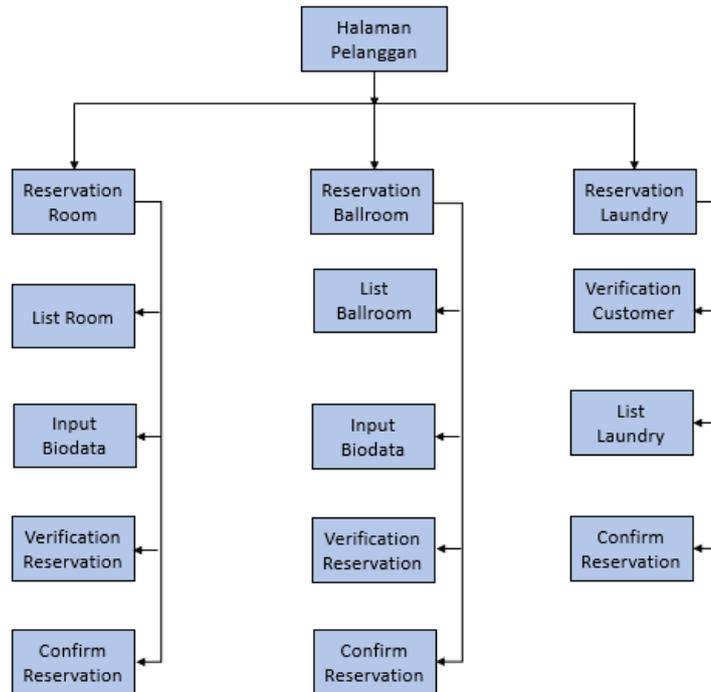


**Gambar 5.11 Struktur Menu Halaman Receptionist**

Struktur menu halaman *receptionist* yang ditunjukkan pada Gambar 5.11 menjelaskan bahwa 3 fungsi dari *reservation room*, *ballroom* dan *laundry* didapatkan dari usecase melihat *history* transaksi dan konfirmasi pemesanan oleh aktor *receptionist*

## 2. Struktur Menu Pelanggan

Struktur menu pelanggan merupakan kumpulan menu-menu atau fungsi yang terdapat pada halaman pelanggan, terdapat 3 fungsi utama yaitu *reservation room*, *reservation ballroom*, dan *reservation laundry*. Struktur menu receptionist ditunjukkan pada Gambar 5.12



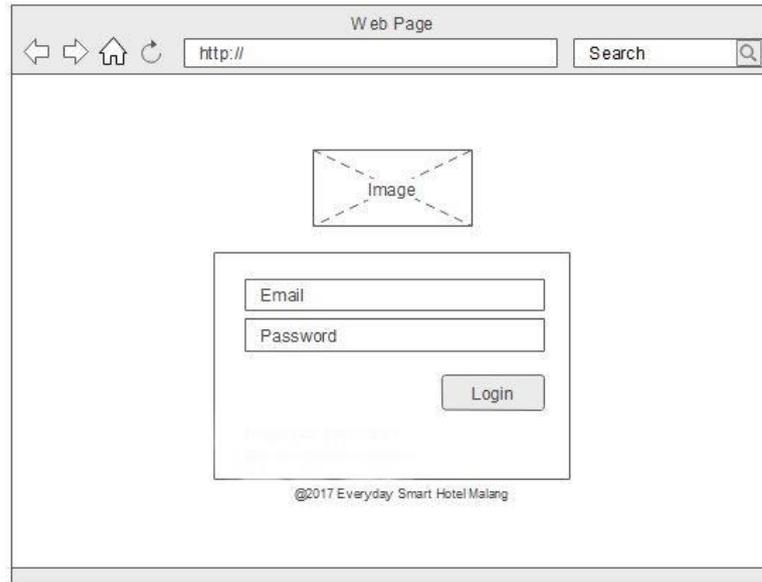
**Gambar 5.12 Struktur Menu Pelanggan**

Struktur menu halaman pelanggan yang ditunjukkan pada Gambar 5.12 menjelaskan bahwa 3 fungsi dari *reservation room*, *ballroom* dan *laundry* didapatkan dari usecase memesan kamar, memesan ballroom dan memesan laundry oleh aktor pelanggan.

## 5.4 Perancangan Antarmuka

Pada tahap perancangan antarmuka pengguna merupakan proses pembuatan rancangan tampilan sistem yang akan dibangun agar pengguna dapat berinteraksi dengan sistem pemesanan. Berikut adalah rancangan antarmuka untuk setiap halaman yang ada pada sistem.

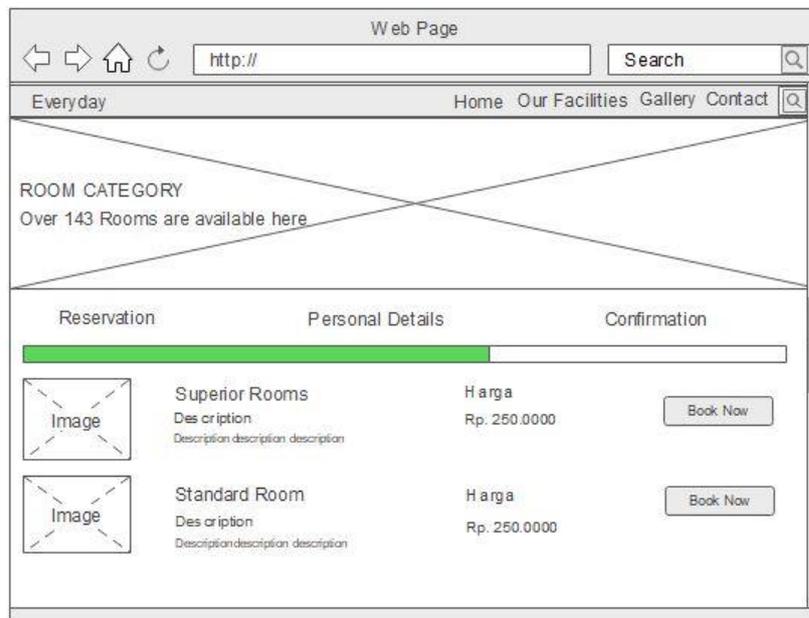
## 1. Perancangan Antarmuka Login



**Gambar 5.13 Antarmuka Login**

Halaman antamuka login menampilkan logo Everyday Smart Hotel dan berupa text area yaitu *email* dan *password* yang akan dimasukkan oleh *receptionist* untuk masuk ke dalam sistem beserta tombol *login*.

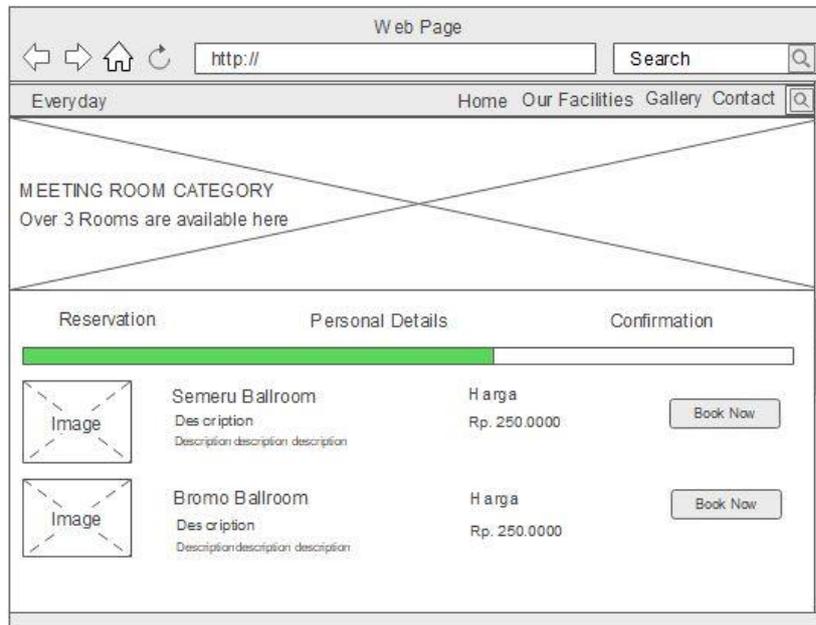
## 2. Perancangan Antarmuka Pemesanan Kamar



**Gambar 5.14 Antarmuka Pemesanan Kamar**

Halaman antamuka pemesanan kamar menampilkan informasi daftar kamar beserta gambar, deskripsi dan harganya. Selanjutnya terdapat *progress bar* yang bertujuan untuk informasi proses pemesanan dan terdapat tombol *book now* untuk memilih kamar yang dipilih.

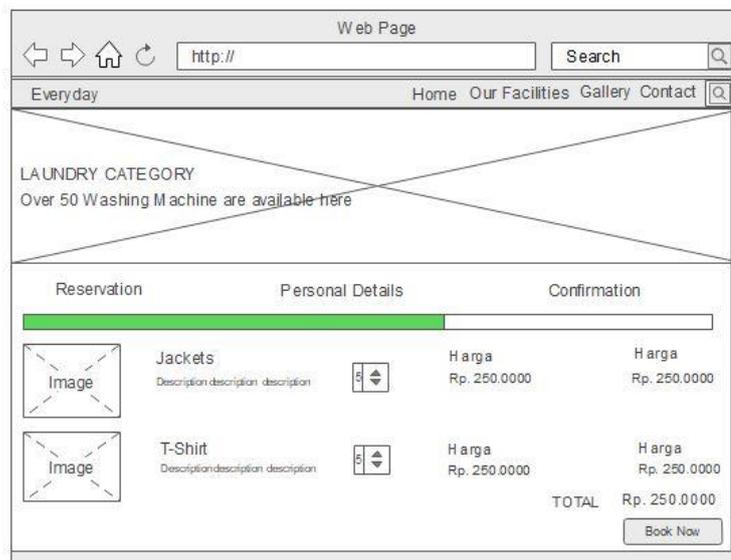
### 3. Perancangan Antarmuka Pemesanan *Ballroom*



**Gambar 5.15** Antarmuka Pemesanan *Ballroom*

Halaman antamuka pemesanan *ballroom* menampilkan informasi daftar *ballroom* beserta gambar, deskripsi dan harganya. Selanjutnya terdapat *progress bar* yang bertujuan untuk informasi proses pemesanan dan terdapat tombol *book now* untuk memilih *ballroom* yang dipilih.

### 4. Perancangan Antarmuka Pemesanan *Laundry*

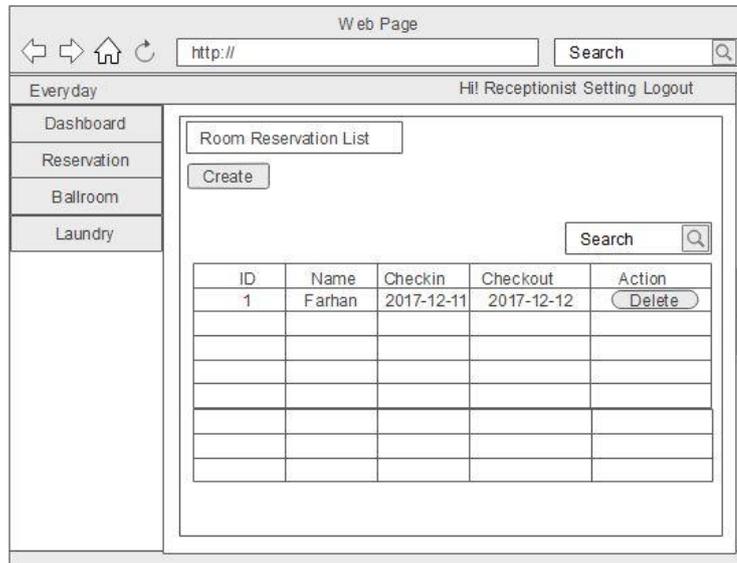


**Gambar 5.16** Antarmuka Pemesanan *Laundry*

Halaman antamuka pemesanan *laundry* menampilkan informasi daftar *laundry* yang tersedia di Everyday Smart Hotel Malang beserta gambar, deskripsi, jumlah pakaian dan harganya. Selanjutnya terdapat *progress bar* yang bertujuan

untuk informasi proses pemesanan dan terdapat tombol *submit* untuk memilih *laundry* yang dipilih.

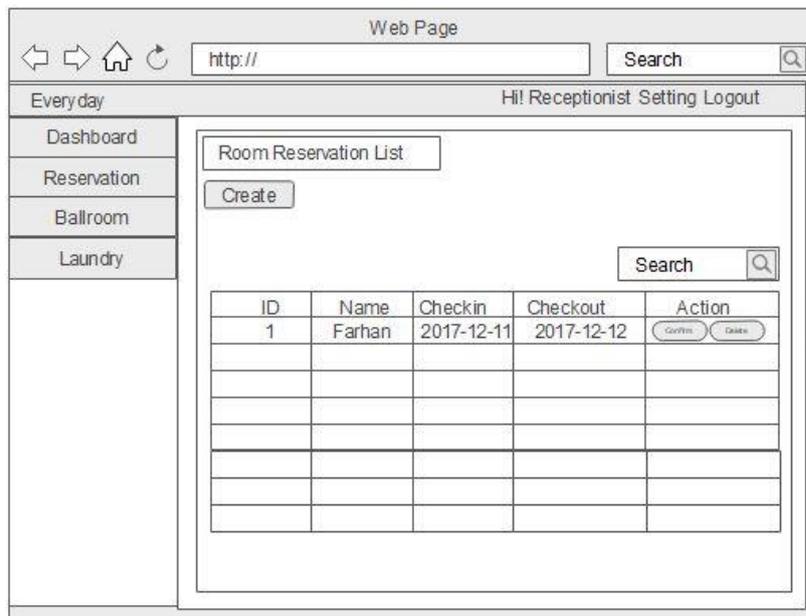
### 5. Perancangan Antarmuka Melihat History Transaksi



**Gambar 5.17 Antarmuka Melihat Informasi Transaksi**

Halaman antarmuka melihat informasi transaksi menampilkan seluruh daftar transaksi yang telah dikonfirmasi dan terdapat *menu bar* untuk memilih menu yang dibutuhkan oleh *receptionist*. Pada tabel terdapat action “*delete*” untuk menghapus data transaksi.

### 6. Perancangan Antarmuka Konfirmasi Pemesanan



**Gambar 5.18 Antarmuka Konfirmasi Pemesanan**

Halaman antarmuka melihat informasi transaksi menampilkan seluruh daftar transaksi yang telah dipesan oleh pelanggan namun belum dikonfirmasi oleh

*receptionist* dan terdapat *menu bar* untuk memilih *menu* yang dibutuhkan oleh *receptionist*. Pada tabel terdapat tombol *action "confirm"* yang bertujuan untuk mengkonfirmasi pemesanan sedangkan tombol *action "delete"* untuk membatalkan pemesanan.