# BAB IV PERENCANAAN DAN ANALISA PERANCANGAN SISTEM

Perencanaan dan analisa perancangan sistem merupakan salah satu tahapan dalam pembuatan sistem. Tahap ini bertujuan mengidentifikasi masalah-masalah yang ditemukan dalam pembuatan sistem, serta mencari solusi untuk memecahkan masalah dalam pembuatan sistem. Sedangkan perancangan sistem merupakan rancangan dari berbagai solusi pemecahan masalah yang ditemukan dalam perencanaan sistem. Dalam perancangan sistem yang dibuat ini dipaparkan rancangan tampilan beserta alur diagram dari program atau aplikasi.

## Analisa Sistem

Tahap analisis digunakan untuk mendapatkan data-data yang diperlukan untuk membangun sistem. Dengan dibangunnya sistem informasi, Manajemen diharapkan bisa menghasilkan sebuah proses pengelolaan data yang jelas dan mudah dipahami.

## Analisa prosedur lama

Pada Rumah Kost Dpavillion pengolahan data pada persewaan kamar kost sebagian masih dilakukan dengan sistem manual. Sistem manual ini mempunyai beberapa kendala, diantaranya pengecekan kamar kosong, pencatatan data penyewa baru, pencatatan dan pelaporan keuangan relatif lebih lama, lebih sulit, dan kurang akurat. Karena itu proses pendatan terkadang terhambat karena faktor manual tersebut dan juga untuk pembayaran terkadang masih banyak kendala. Untuk itu Rumah Kost Dpavillion membutuhkan sebuah Sistem Management untuk mempermudah admin dalam mengelola data laporan agar lebih mengefiseinsi waktu juga agar dapat meningkatkan efektivitas, efisiensi, dan produktivitas kerja.

## Identifikasi masalah

Berdasarkan hasil analisa sistem yang sudah berjalan di Rumah Kost Dpavillion penulis menguraikan beberapa masalah. Diantaranya:

* + - 1. Bagaimana mengembangkan sebuah sistem berbasis komputer yang dapat mendukung proses manajemen kost dengan menggunakan *framework* laravel?
      2. Bagaimana sistem dengan menggunakan metode RAD (*Rapid Application Development*) dapat digunakan untuk mendukung proses manajemen kost?

## Pemecahan Masalah

Untuk memecahkan masalah tersebut, maka perlu dibangun sebuah sistem manajemen kost yang bertujuan untuk dapat membantu dan mempermudah pengelolaan rumah kost berdasarkan kriteria - kriteria yang telah ditentukan oleh pihak Rumah Kost Dpavillion.

## Analisa Kebutuhan Sistem

### Analisa Kebutuhan *Software*

Dalam pembuatan dan pengembangan perancangan sistem ini penulis menggunakan perangkat lunak/*software* sebagai berikut :

* + - 1. *Draw.io*

*Draw.io* digunakan untuk memodelkan sistem sebelum dituliskan ke dalam program dalam suatu bahasa.

* + - 1. *Balsamiq Mockup*

*Balsamiq Mockup* digunakan dalam pembuatan *user interface* dalam sebuah aplikasi.

* + - 1. *Notepad ++*

*Notepad ++* digunakan untuk meng-*coding* program berbasis web.

* + - 1. *Xampp*

*Xampp* digunakan untuk *database* dan *web services*.

### Analisa Kebutuhan *Hardware*

Agar Sistem Management di Rumah Kost Dpavillion dapat dioperasikan secara maksimal, maka diperlukan sebuah perangkat keras (*hardware)* sebagai sarana dalam menjalankan program aplikasi sebagai berikut:

* + - 1. *Processor Intel Dual Core* atau lebih tinggi
      2. Memori RAM 2GB.
      3. *Harddisk* 500 GB.
      4. *Monitor.*
      5. *Keyboard*
      6. *Mouse.*
      7. *Printer*

## Analisa Kebutuhan Data

Analisa Kebutuhan Data adalah sebuah proses untuk mendapatkan informasi, model, spesifikasi tentang perangkat lunak yang diinginkan klien/pengguna. Kedua belah pihak antara klien dan pembuat perangkat lunak terlibat aktif dalam tahap ini. Data yang di butuhkan adalah:

1. Mengelola Data Ruangan
2. Mengelola Data Pemesanan Ruangan
3. Mengelola Data Pembayaran
4. Mengelola Data Kebutuhan Barang Ruangan
5. Mengelola Data Fasilitas Ruangan
6. Mengelola Data Pelanggan
7. Mengelola Data Laporan

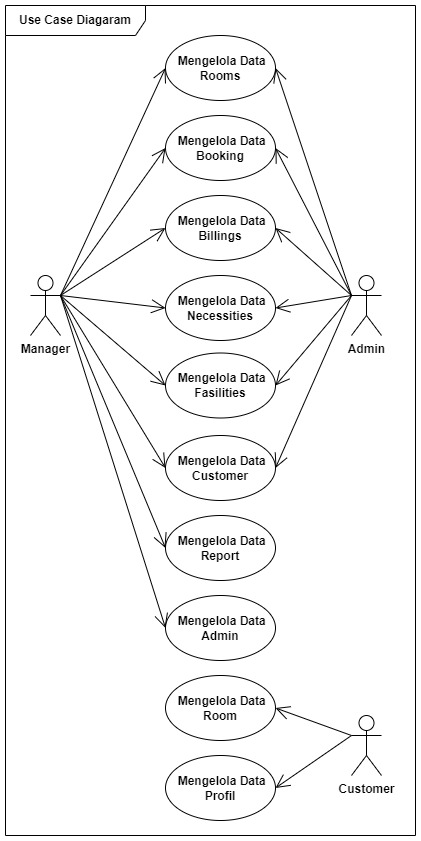
## Perancangan Sistem

Perancangan sistem yang diusulkan akan berorientasi objek dengan menggunakan notasi UML (*Unified Modeling Language*) yang berfungsi untuk memberikan gambaran umum tentang sistem.

Sesuai dengan permasalahan yang berada didalam Rumah Kost Dpavillion, maka penulis melakukan pendekatan sistem berorientasi objek yakni dengan membuat *use case* diagram, *sequence* diagram, *activity* diagram dan *class* diagram. Penjelasan tersebut akan dibahas berikut ini :

### Use Case Diagram

*Use Case diagram* menggambarkan apa saja aktivitas yang dilakukan oleh suatu sistem dari pengamatan sudut pandang luar. Diagram *use case* adalah sebuah diagram yang menjelaskan apa yang harus dilakukan oleh sistem pada level konseptual sehingga bisa memahami apakah keputusan yang diambil oleh sistem apakah benar atau tidak.



Gambar 4. 1. Use Case Diagram

* 1. **Skenario *Use Case***

Detail skenario dari masing-masing *use case* setelah pre-kondisi terpenuhi dapat dilihat pada uraian berikut :

1. **Skenario *Use Case* *Dashboard***

Nama *Use case* : *Dashboard*

Aktor : *Manager* dan Admin

Deskripsi : Proses ini adalah sebuah kegiatan untuk

dapat melihat data

*Precondition* : *Manager* dan Admin sudah *login* ke dalam

sistem.

*Postcondicition* : *Manager* dan Admin dapat melihat data

Tabel 4. 1. Alur Use Case Dashboard

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Manager* dan Admin, *Log in* ke sistem. | **Langkah 2 :**  Menampilkan data *Dashboard*. |

1. **Skenario *Use Case* *Rooms***

**a) Skenario Use Case *Available Rooms***

Nama *Use case* : *Available Rooms*

Aktor : *Manager* dan Admin

Deskripsi : Proses ini adalah sebuah kegiatan

untuk dapat melihat, menambah,

mengadit, menghapus dan mencari

data

*Precondition* : *Manager* dan Admin sudah *login* ke

dalam sistem.

*Postcondicition* : *Manager* dan Admin dapat melihat,

menambah,mengadit, menghapus

dan mencari data

Tabel 4. 2. Alur Use Case Available Rooms

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Manager* dan Adminpilih *Available Rooms*. | **Langkah 2 :**  Menampilkan *form Available Rooms*. |
| **Alternatif *Manager* (Tambah Data *Available Rooms*)** | |
| **Langkah 1 :**  *Manager* klik tombol *Add Room*. | **Langkah 2 :**  Validasi data yang akan di input. |
| **Langkah 3 :**  *Manager* menginput data *Available Rooms* dan klik tombol *save*. | **Langkah 4 :**  Data tersimpan ke *database* dan menampilkan data *Available Rooms*. |
| **Alternatif *Manager* dan Admin (Melihat Data *Available Rooms*)** | |
| **Langkah 1 :**  *Manager* dan Adminklik tombol *icon* mata. | **Langkah 2 :**  Menampilkan data *Available Rooms* |
| **Alternatif *Manager* (Edit Data *Available Rooms*)** | |
| **Langkah 1 :**  *Manager* klik tombol *icon* edit | **Langkah 2 :**  Validasi data yang akan di edit. |
| **Langkah 3 :**  *Manager* mengedit data dan klik tombol *save*. | **Langkah 4 :**  Data *Available Rooms* terupdate ke *database* dan menampilkan data *Available Rooms*. |
| **Alternatif *Manager* (Hapus Data *Available Rooms*)** | |
| **Langkah 1 :**  *Manager* klik *icon* sampah dan menghapus data *Available Rooms*. | **Langkah 2 :**  Data *Available Rooms* terhapus dari *database* dan menampilkan data *Available Rooms*. |
| **Alternatif *Manager* dan Admin (Cari Data *Available Rooms*)** | |
| **Langkah 1 :**  *Manager* dan Adminmemilih *Search* dan memasukan data yang dicari. | **Langkah 2 :**  Mengambil data dari *database* dan menampilkan data yang dicari. |
| **Alternatif *Manager* (*Add Booking* Data *Available Rooms*)** | |
| **Langkah 1 :**  *Manager klik tombol icon add booking* | **Langkah 2 :**  Mengambil data dari *database* dan memindahkan data ke *Unvailable Rooms*. |

**b) Skenario Use Case *Unavailable Rooms***

Nama *Use case* : *Unavailable Rooms*

Aktor : *Manager* dan Admin

Deskripsi : Proses ini adalah sebuah kegiatan

untuk dapat melihat, menambah,

mengadit, menghapus dan mencari

data

*Precondition* : *Manager* dan Admin sudah *login* ke

dalam sistem.

*Postcondicition* : *Manager* dan Admin dapat melihat,

menambah,mengadit, menghapus

dan mencari data

Tabel 4. 3. Alur Use Case Unavailable Rooms

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Manager* dan Adminpilih *Unavailable Rooms*. | **Langkah 2 :**  Menampilkan *form Unavailable Rooms*. |
| **Alternatif *Manager* (Tambah Data *Unavailable Rooms*)** | |
| **Langkah 1 :**  *Manager* klik tombol *Add Room*. | **Langkah 2 :**  Validasi data yang akan di input. |
| **Langkah 3 :**  *Manager* menginput data *Unavailable Rooms* dan klik tombol *save*. | **Langkah 4 :**  Data tersimpan ke *database* dan menampilkan data *Unavailable Rooms*. |
| **Alternatif *Manager* dan Admin (Melihat Data *Unavailable Rooms*)** | |
| **Langkah 1 :**  *Manager* dan Adminklik tombol *icon* mata. | **Langkah 2 :**  Menampilkan data *Unavailable Rooms* |
| **Alternatif *Manager* (Edit Data *Unavailable Rooms*)** | |
| **Langkah 1 :**  *Manager* klik tombol *icon* edit | **Langkah 2 :**  Validasi data yang akan di edit. |
| **Langkah 3 :**  *Manager* mengedit data dan klik tombol *save*. | **Langkah 4 :**  Data *Unavailable Rooms* terupdate ke *database* dan menampilkan data *Unavailable Rooms*. |

Tabel 4. 3. Alur Use Case Unavailable Rooms (lanjutan)

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Alternatif *Manager* (Hapus Data *Unavailable Rooms*)** | |
| **Langkah 1 :**  *Manager* klik *icon* sampah dan menghapus data *Unavailable Rooms*. | **Langkah 2 :**  Data *Unavailable Rooms* terhapus dari *database* dan menampilkan data *Unavailable Rooms*. |
| **Alternatif *Manager* dan Admin (Cari Data *Unavailable Rooms*)** | |
| **Langkah 1 :**  *Manager* dan Adminmemilih *Search* dan memasukan data yang dicari. | **Langkah 2 :**  Mengambil data dari *database* dan menampilkan data yang dicari. |

**c) Skenario Use Case *All Rooms***

Nama *Use case* : *All Rooms*

Aktor : *Manager* dan Admin

Deskripsi : Proses ini adalah sebuahkegiatan

untuk dapat melihat, menambah,

mengadit, menghapus dan mencari

data

*Precondition* : *Manager* dan Admin sudah *login* ke

dalamsistem.

*Postcondicition* : *Manager* dan Admin dapat melihat,

menambah,mengadit, menghapus

dan mencari data

Tabel 4. 4. Alur Use Case All Rooms

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Manager* dan Adminpilih *All Rooms*. | **Langkah 2 :**  Menampilkan *form All Rooms*. |
| **Alternatif *Manager* (Tambah Data *All Rooms*)** | |
| **Langkah 1 :**  *Manager* klik tombol *Add Room*. | **Langkah 2 :**  Validasi data yang akan di input. |
| **Langkah 3 :**  *Manager* menginput data *All Rooms* dan klik tombol *save*. | **Langkah 4 :**  Data tersimpan ke *database* dan menampilkan data *All Rooms*. |

Tabel 4. 4. Alur Use Case All Rooms (lanjutan)

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Alternatif *Manager* dan Admin (Melihat Data *All Rooms*)** | |
| **Langkah 1 :**  *Manager* dan Adminklik tombol *icon* mata. | **Langkah 2 :**  Menampilkan data *Unavailable Rooms* |
| **Alternatif *Manager* (Edit Data *All Rooms*)** | |
| **Langkah 1 :**  *Manager* klik tombol *icon* edit | **Langkah 2 :**  Validasi data yang akan di edit. |
| **Langkah 3 :**  *Manager* mengedit data dan klik tombol *save*. | **Langkah 4 :**  Data *All Rooms* terupdate ke *database* dan menampilkan data *All Rooms*. |
| **Alternatif *Manager* (Hapus Data *All Rooms*)** | |
| **Langkah 1 :**  *Manager* klik *icon* sampah dan menghapus data *All Rooms*. | **Langkah 2 :**  Data *All Rooms* terhapus dari *database* dan menampilkan data *All Rooms*. |
| **Alternatif *Manager* dan Admin (Cari Data *All Rooms*)** | |
| **Langkah 1 :**  *Manager* dan Adminmemilih *Search* dan memasukan data yang dicari. | **Langkah 2 :**  Mengambil data dari *database* dan menampilkan data yang dicari. |

1. **Skenario *Use Case* *Booking***
2. **Skenario *Use Case* *New***

Nama *Use case* : *New*

Aktor : *Manager* dan Admin

Deskripsi : Proses ini adalah sebuah kegiatan untuk

dapat melihat, mencari, dan menghapus

data.

*Precondition* : *Manager* dan Admin sudah *login* ke dalam

sistem.

*Postcondicition* : *Manager* dan Admin dapat melihat,

mencari,danmenghapus data.

Tabel 4. 5. Alur Use Case New

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Manager* dan Adminpilih *New*. | **Langkah 2 :**  Menampilkan *form New*. |
| **Alternatif *Manager* dan Admin (Melihat Data *New*)** | |
| **Langkah 1 :**  *Manager* dan Adminklik tombol *icon* mata. | **Langkah 2 :**  Menampilkan data *New* |
| **Alternatif *Manager dan Admin* (Hapus Data *New*)** | |
| **Langkah 1 :**  *Manager* dan Adminklik *icon* sampah dan menghapus data *New*. | **Langkah 2 :**  Data *New* terhapus dari *database* dan menampilkan data *New*. |
| **Alternatif *Manager* dan Admin (Cari Data *New*)** | |
| **Langkah 1 :**  *Manager* dan Adminmemilih *Search* dan memasukan data yang dicari. | **Langkah 2 :**  Mengambil data dari *database* dan menampilkan data yang dicari. |

1. **Skenario *Use Case* *Grace***

Nama *Use case* : *Grace*

Aktor : *Manager* dan Admin

Deskripsi : Proses ini adalah sebuah kegiatan untuk

dapat melihat, mencari, dan menghapus

data.

*Precondition* : *Manager* dan Admin sudah *login* ke dalam

sistem.

*Postcondicition* : *Manager* dan Admin dapat melihat,

mencari, dan*,* menghapus data.

Tabel 4. 6. Alur Use Case Grace

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Manager* dan Adminpilih *Grace*. | **Langkah 2 :**  Menampilkan *form Grace*. |
| **Alternatif *Manager* dan Admin (Melihat Data *Grace*)** | |
| **Langkah 1 :**  *Manager* dan Adminklik tombol *icon* mata. | **Langkah 2 :**  Menampilkan data *Grace* |

Tabel 4. 6. Alur Use Case Grace (lanjutan)

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Alternatif *Manager dan Admin* (Hapus Data *Grace*)** | |
| **Langkah 1 :**  *Manager* dan Adminklik *icon* sampah dan menghapus data *Grace*. | **Langkah 2 :**  Data *Grace* terhapus dari *database* dan menampilkan data *Grace*. |
| **Alternatif *Manager* dan Admin (Cari Data *Grace*)** | |
| **Langkah 1 :**  *Manager* dan Adminmemilih *Search* dan memasukan data yang dicari. | **Langkah 2 :**  Mengambil data dari *database* dan menampilkan data yang dicari. |

1. **Skenario *Use Case* *Late***

Nama *Use case* : *Late*

Aktor : *Manager* dan Admin

Deskripsi : Proses ini adalah sebuah kegiatan untuk

dapat melihat, mencari, dan menghapus

data.

*Precondition* : *Manager* dan Admin sudah *login* ke dalam

sistem.

*Postcondicition* : *Manager* dan Admin dapat melihat,

mencari, danmenghapus data.

Tabel 4. 7. Alur Use Case Late

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Manager* dan Adminpilih *Late*. | **Langkah 2 :**  Menampilkan *form Late*. |
| **Alternatif *Manager* dan Admin (Melihat Data *Late*)** | |
| **Langkah 1 :**  *Manager* dan Adminklik tombol *icon* mata. | **Langkah 2 :**  Menampilkan data *Late* |
| **Alternatif *Manager dan Admin* (Hapus Data *Late*)** | |
| **Langkah 1 :**  *Manager* dan Adminklik *icon* sampah dan menghapus data *Late*. | **Langkah 2 :**  Data *Late* terhapus dari *database* dan menampilkan data *Late*. |

Tabel 4. 7. Alur Use Case Late (lanjutan)

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Alternatif *Manager* dan Admin (Cari Data *Late*)** | |
| **Langkah 1 :**  *Manager* dan Adminmemilih *Search* dan memasukan data yang dicari. | **Langkah 2 :**  Mengambil data dari *database* dan menampilkan data yang dicari. |

1. **Skenario *Use Case* Normal**

Nama *Use case* : Normal

Aktor : *Manager* dan Admin

Deskripsi : Proses ini adalah sebuah kegiatan untuk

dapat melihat, mencari, dan menghapus

data.

*Precondition* : *Manager* dan Admin sudah *login* ke dalam

sistem.

*Postcondicition* : *Manager* dan Admin dapat melihat,

mencari,danmenghapus data.

Tabel 4. 8. Alur Use Case Normal

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Manager* dan Adminpilih Normal. | **Langkah 2 :**  Menampilkan *form* Normal. |
| **Alternatif *Manager* dan Admin (Melihat Data Normal)** | |
| **Langkah 1 :**  *Manager* dan Adminklik tombol *icon* mata. | **Langkah 2 :**  Menampilkan data Normal |
| **Alternatif *Manager dan Admin* (Hapus Data Normal)** | |
| **Langkah 1 :**  *Manager* dan Adminklik *icon* sampah dan menghapus data Normal. | **Langkah 2 :**  Data Normalterhapus dari *database* dan menampilkan data Normal. |
| **Alternatif *Manager* dan Admin (Cari Data Normal)** | |
| **Langkah 1 :**  *Manager* dan Adminmemilih *Search* dan memasukan data yang dicari. | **Langkah 2 :**  Mengambil data dari *database* dan menampilkan data yang dicari. |

1. **Skenario *Use Case* *All Booking***

Nama *Use case* : *All Booking*

Aktor : *Manager* dan Admin

Deskripsi : Proses ini adalah sebuah kegiatan untuk

dapat melihat, mencari, dan menghapus

data.

*Precondition* : *Manager* dan Admin sudah *login* ke dalam

sistem.

*Postcondicition* : *Manager* dan Admin dapat melihat,

mencari,danmenghapus data.

Tabel 4. 9. Alur Use Case All Booking

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Manager* dan Adminpilih *All Booking*. | **Langkah 2 :**  Menampilkan *form All Booking*. |
| **Alternatif *Manager* dan Admin (Melihat Data *All Booking*)** | |
| **Langkah 1 :**  *Manager* dan Adminklik tombol *icon* mata. | **Langkah 2 :**  Menampilkan data *All Booking* |
| **Alternatif *Manager dan Admin* (Hapus Data *All Booking*)** | |
| **Langkah 1 :**  *Manager* dan Adminklik *icon* sampah dan menghapus data *All Booking*. | **Langkah 2 :**  Data *All Booking* terhapus dari *database* dan menampilkan data *All Booking*. |
| **Alternatif *Manager* dan Admin (Cari Data *All Booking*)** | |
| **Langkah 1 :**  *Manager* dan Adminmemilih *Search* dan memasukan data yang dicari. | **Langkah 2 :**  Mengambil data dari *database* dan menampilkan data yang dicari. |

1. **Skenario *Use Case* *Billing***

Nama *Use case* : *Billing*

Aktor : *Manager* dan Admin

Deskripsi : Proses ini adalah sebuah kegiatan untuk

dapat melihat, menghapus, dan mencari

data

*Precondition* : *Manager* dan Admin sudah *login* ke dalam

sistem.

*Postcondicition* : *Manager* dan Admin dapat melihat,

menghapus, dan mencari data

Tabel 4. 10. Alur Use Case Billing

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Manager* dan Adminpilih *Billing*. | **Langkah 2 :**  Menampilkan *form Billing*. |
| **Alternatif *Manager* dan Admin (Melihat Data *Billing*)** | |
| **Langkah 1 :**  *Manager* dan Adminklik tombol *icon* mata. | **Langkah 2 :**  Menampilkan data *Billing* |
| **Alternatif *Manager* (Hapus Data *Billing*)** | |
| **Langkah 1 :**  *Manager* klik *icon* sampah dan menghapus data *Billing*. | **Langkah 2 :**  Data *Billing* terhapus dari *database* dan menampilkan data *Billing*. |
| **Alternatif *Manager* dan Admin (Cari Data *Billing*)** | |
| **Langkah 1 :**  *Manager* dan Adminmemilih *Search* dan memasukan data yang dicari. | **Langkah 2 :**  Mengambil data dari *database* dan menampilkan data yang dicari. |

**5. Skenario *Use Case* *Necessities***

Nama *Use Case* : *Necessities*

Aktor : *Manager* dan Admin

Deskripsi : Proses ini adalah sebuah kegiatan untuk

dapat menambah, mengedit dan menghapus

data.

*Precondition* : *Manager* dan Admin sudah *login*  kedalam

sistem.

*Postconditon* : *Manager* dan Admin menambah, mengedit

dan menghapus data.

Tabel 4. 11. Alur Use Case Necessities

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Manager* dan Adminpilih *Necessities*. | **Langkah 2 :**  Menampilkan *form Necessities*. |
| **Alternatif *Manager dan Admin* (Tambah Data *Necessities*)** | |
| **Langkah 1 :**  *Manager dan Admin* klik tombol *Add Necessities*. | **Langkah 2 :**  Validasi data yang akan di input. |
| **Langkah 3 :**  *Manager dan Admin* menginput data *Necessities* dan klik tombol *save*. | **Langkah 4 :**  Data tersimpan ke *database* dan menampilkan data *Necessities*. |
| **Alternatif *Manager dan Admin* (Edit Data *Necessities*)** | |
| **Langkah 1 :**  *Manager dan Admin* klik tombol *icon* edit | **Langkah 2 :**  Validasi data yang akan di edit. |
| **Langkah 3 :**  *Manager dan Admin* mengedit data dan klik tombol *save*. | **Langkah 4 :**  Data *Necessities* terupdate ke *database* dan menampilkan data *Necessities*. |
| **Alternatif *Manager dan Admin* (Hapus Data *Necessities*)** | |
| **Langkah 1 :**  *Manager dan Admin* klik *icon* sampah dan menghapus data *Necessities*. | **Langkah 2 :**  Data *Necessities* terhapus dari *database* dan menampilkan data *Necessities*. |

**6. Skenario *Use Case* *Fasilities***

Nama *Use case* : *Fasilities*

Aktor : *Manager* dan Admin

Deskripsi : Proses ini adalah sebuah kegiatan untuk

menambah dan menghapus data.

*Precondition* : *Manager* dan Admin sudah *login* ke dalam

sistem.

*Postcondicition* : *Manager* dan Admin menambah dan

menghapus data.

Tabel 4. 12. Alur Use Case Fasilities

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Manager* dan Adminpilih *Fasilities*. | **Langkah 2 :**  Menampilkan *form Fasilities*. |
| **Alternatif *Manager dan Admin* (Tambah Data *Fasilities*)** | |
| **Langkah 1 :**  *Manager dan Admin* klik *Insert New Fasilities*. | **Langkah 2 :**  Validasi data yang akan di input. |
| **Langkah 3 :**  *Manager dan Admin* menginput data *Fasilities* dan *enter*. | **Langkah 4 :**  Data tersimpan ke *database* dan menampilkan data *Fasilities*. |
| **Alternatif *Manager dan Admin* (Hapus Data *Fasilities*)** | |
| **Langkah 1 :**  *Manager dan Admin* klik *icon* sampah dan menghapus data *Fasilities*. | **Langkah 2 :**  Data *Fasilities* terhapus dari *database* dan menampilkan data *Fasilities*. |

**7. Skenario *Use Case* *Customers***

Nama *Use case* : *Customers*

Aktor : *Manager* dan Admin

Deskripsi : Proses ini adalah sebuah kegiatan untuk

melihat, menambah, mengedit dan

menghapus data.

*Precondition* : *Manager* dan Admin sudah *login* ke dalam

sistem.

*Postcondicition* : *Manager* dan Admin melihat, menambah,

mengedit dan menghapus data.

Tabel 4. 13.Alur Use Case Customers

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Manager* dan Adminpilih *Customers*. | **Langkah 2 :**  Menampilkan *form Customers*. |

Tabel 4. 13.Alur Use Case Customers (lanjutan)

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Alternatif *Manager dan Admin* (Tambah Data *Customers*)** | |
| **Langkah 1 :**  *Manager* dan Adminklik tombol *Add Customers*. | **Langkah 2 :**  Validasi data yang akan di input. |
| **Langkah 3 :**  *Manager* dan Adminmenginput data *Customers* dan klik tombol *save*. | **Langkah 4 :**  Data tersimpan ke *database* dan menampilkan data *Customers*. |
| **Alternatif *Manager* dan Admin (Melihat Data *Customers*)** | |
| **Langkah 1 :**  *Manager* dan Adminklik tombol *icon* mata. | **Langkah 2 :**  Menampilkan data *Customers* |
| **Alternatif *Manager* dan Admin(Edit Data *Customers*)** | |
| **Langkah 1 :**  *Manager* dan Adminklik tombol *icon* edit | **Langkah 2 :**  Validasi data yang akan di edit. |
| **Langkah 3 :**  *Manager* dan Adminmengedit data dan klik tombol *save*. | **Langkah 4 :**  Data *Customers* terupdate ke *database* dan menampilkan data *Customers*. |
| **Alternatif *Manager* (Hapus Data *Customers*)** | |
| **Langkah 1 :**  *Manager* dan Admin klik *icon* sampah dan menghapus data *Customers*. | **Langkah 2 :**  Data *Customers* terhapus dari *database* dan menampilkan data *Customers*. |

**8. Skenario *Use Case* *Report***

Nama *Use case* : *Report*

Aktor : *Manager*

Deskripsi : Proses ini adalah sebuah kegiatan untuk

mengolah data *Report*.

*Precondition* : *Manager* sudah *login* ke dalam sistem.

*Postcondicition* : *Manager* mencetak data *Report*..

Tabel 4. 14. Alur Use Case Laporan

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Alternatif (Print Data *Monthly Report*)** | |
| **Langkah 1 :**  *Manager* memilih *Monthly Report*. | **Langkah 2 :**  Mengambil data dari *database* dan mencetak data *Monthly Report* |
| **Alternatif (Print Data *Years Report*)** | |
| **Langkah 1 :**  *Manager* memilih *Years Report*. | **Langkah 2 :**  Mengambil data dari *database* dan mencetak data *Years Report* |

**9. Skenario *Use Case* Admin**

Nama *Use case* : Admin

Aktor : *Manager*

Deskripsi : Proses ini adalah sebuah kegiatan untuk

melihat, menambah, mengedit dan

menghapus data.

*Precondition* : *Manager* sudah *login* ke dalam sistem.

*Postcondicition* :*Manager* melihat, menambah, mengedit dan

menghapus data.

Tabel 4. 15. Alur Use Case Admin

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Manager* pilih Admin. | **Langkah 2 :**  Menampilkan *form* Admin. |
| **Alternatif *Manager* (Tambah Data Admin)** | |
| **Langkah 1 :**  *Manager* klik tombol *Add* Admin. | **Langkah 2 :**  Validasi data yang akan di input. |
| **Langkah 3 :**  *Manager* menginput data Admin dan klik tombol *save*. | **Langkah 4 :**  Data tersimpan ke *database* dan menampilkan data Admin. |
| **Alternatif *Manager* (Melihat Data Admin)** | |
| **Langkah 1 :**  *Manager* klik tombol *icon* mata. | **Langkah 2 :**  Menampilkan data Admin |

Tabel 4. 16. Alur Use Case Admin

|  |  |
| --- | --- |
| **Alternatif *Manager* (Edit Data Admin)** | |
| **Langkah 1 :**  *Manager* klik tombol *icon* edit | **Langkah 2 :**  Validasi data yang akan di edit. |
| **Langkah 3 :**  *Manager* mengedit data dan klik tombol *save*. | **Langkah 4 :**  Data Admin terupdate ke *database* dan menampilkan data Admin. |
| **Alternatif *Manager* (Hapus Data Admin)** | |
| **Langkah 1 :**  *Manager* klik *icon* sampah dan menghapus data Admin. | **Langkah 2 :**  Data Admin terhapus dari *database* dan menampilkan data Admin. |

**10. Skenario *Use Case* Data *Room***

Nama *Use case* : Data *Room*

Aktor : *Customer*

Deskripsi : Proses ini adalah sebuah kegiatan untuk melihat,

dan menambah data.

*Precondition* : *Customer* sudah *login* ke dalam sistem.

*Postcondicition* : *Customer* melihat dan menambah data

Tabel 4. 17. Alur Use Case Data Room

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Manager* pilih Data *Room*. | **Langkah 2 :**  Menampilkan Data *Room*. |
| **Alternatif *Customer* (Tambah Data *Room*)** | |
| **Langkah 1 :**  *Customer* klik Data *Room*. | **Langkah 2 :**  Validasi data yang akan di tampilkan. |
| **Langkah 3 :**  *Customer* menginput data *Booking* dan klik tombol *Book Now*. | **Langkah 4 :**  Data tersimpan ke *database* dan menampilkan data halaman berikutnya. |
| **Langkah 5 :**  *Customer* mengunggah bukti pembayaran *Booking Room*. | **Langkah 4 :**  Data tersimpan ke *database* dan menampilkan halaman utama *Customer*. |
| **Alternatif *Customer* (Melihat Data *Room*)** | |
| **Langkah 1 :**  *Customer* klik gambar ruangan kos. | **Langkah 2 :**  Menampilkan data ruangan kos. |

**11. Skenario *Use Case* Data Profil**

Nama *Use case* : Data Profil

Aktor : *Customer*

Deskripsi : Proses ini adalah sebuah kegiatan untuk

melihat, dan mengedit data.

*Precondition* : *Customer* sudah *login* ke dalam sistem.

*Postcondicition* : *Customer* melihat dan mengedit data.

Tabel 4. 18. Alur Use Case Data Profil

|  |  |
| --- | --- |
| **Actor** | **Sistem** |
| **Langkah 1 :**  *Customer* pilih *My Profile*. | **Langkah 2 :**  Menampilkan Data Profil. |
| **Alternatif *Customer* (Melihat Data Profil)** | |
| **Langkah 1 :**  *Customer* klik gambar ruangan kos. | **Langkah 2 :**  Menampilkan data ruangan kos. |
| **Alternatif *Customer* (Edit Data Profil)** | |
| **Langkah 1 :**  *Customer* klik tombol ubah | **Langkah 2 :**  Validasi data yang akan di edit. |
| **Langkah 3 :**  *Customer* mengedit data dan klik tombol *save*. | **Langkah 4 :**  Data Profil terupdate ke *database* dan menampilkan data Profil. |

### Sequence Diagram

*Sequence Diagram* adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah *object.* Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object.*

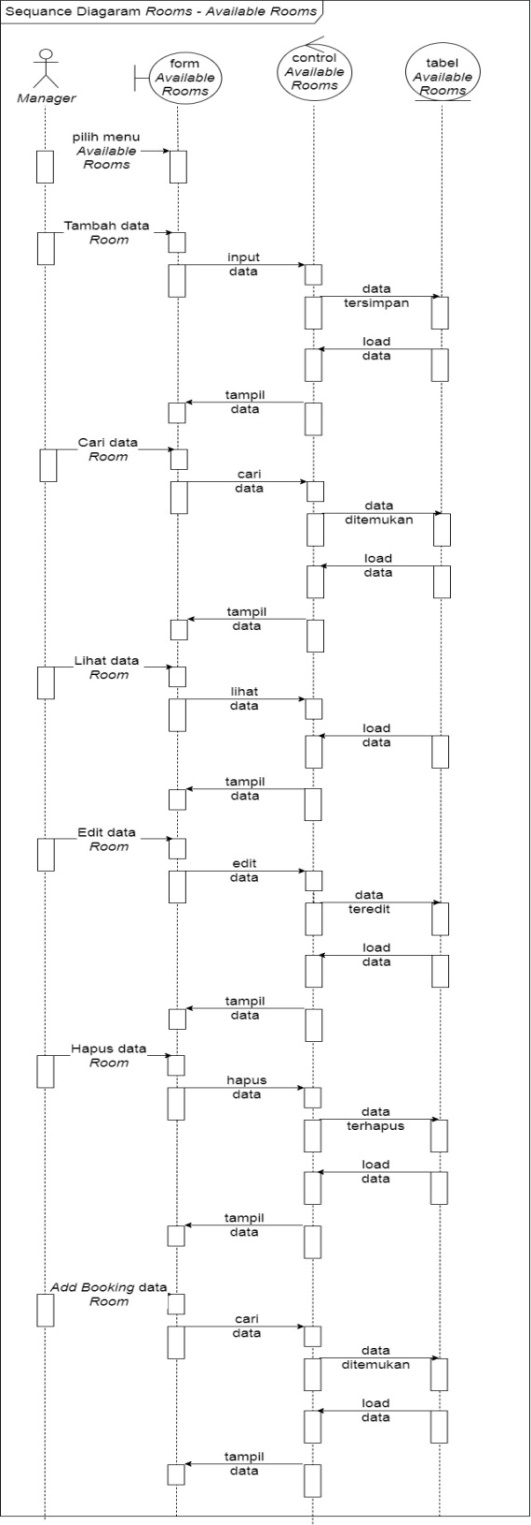
1. ***Sequence diagram Dashboard***

*Sequence Diagram Dashboard* yang menyajikan informasi mengenai indikator utama dari aktifitas sistem secara sekilas dalam layar tunggal.



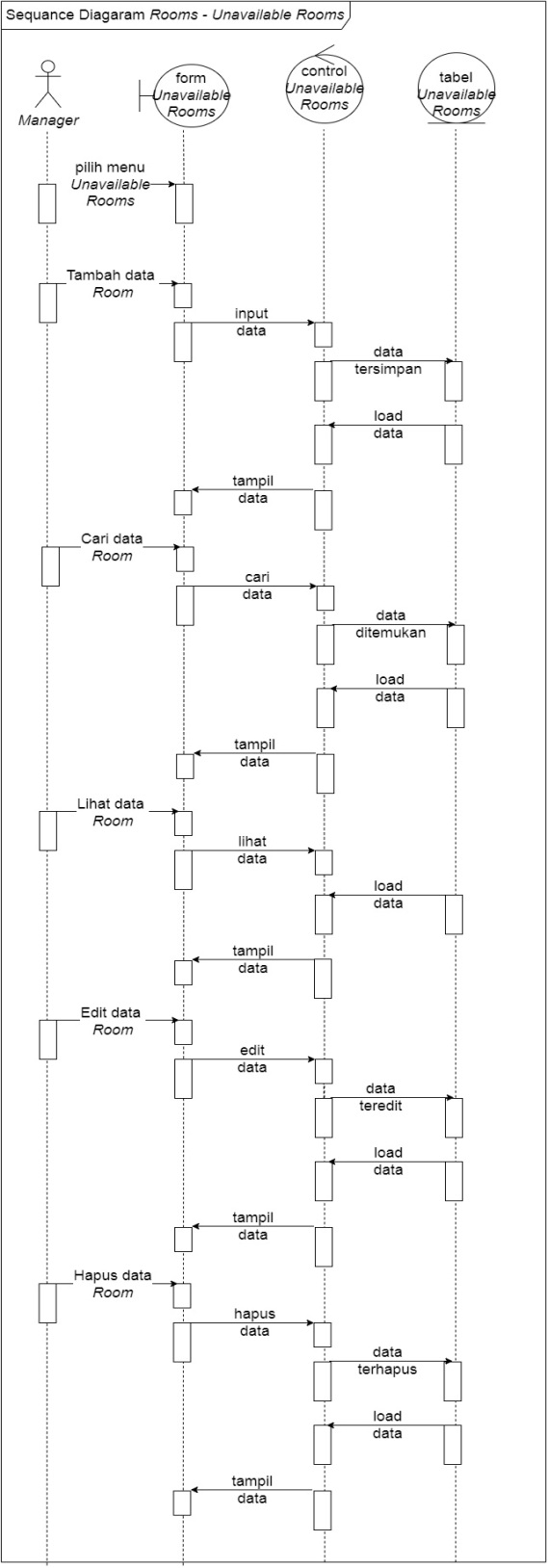
Gambar 4. 2. Sequence diagram Dashboard

1. ***Sequence diagram Rooms***
2. *Sequence Diagram Room* - *Manager*
3. *Sequence Diagram Available Rooms* menggambarkan alur sistem pada data yang digunakan oleh *Manager* untuk melihat, menambah, mencari, mengedit dan menghapus data.



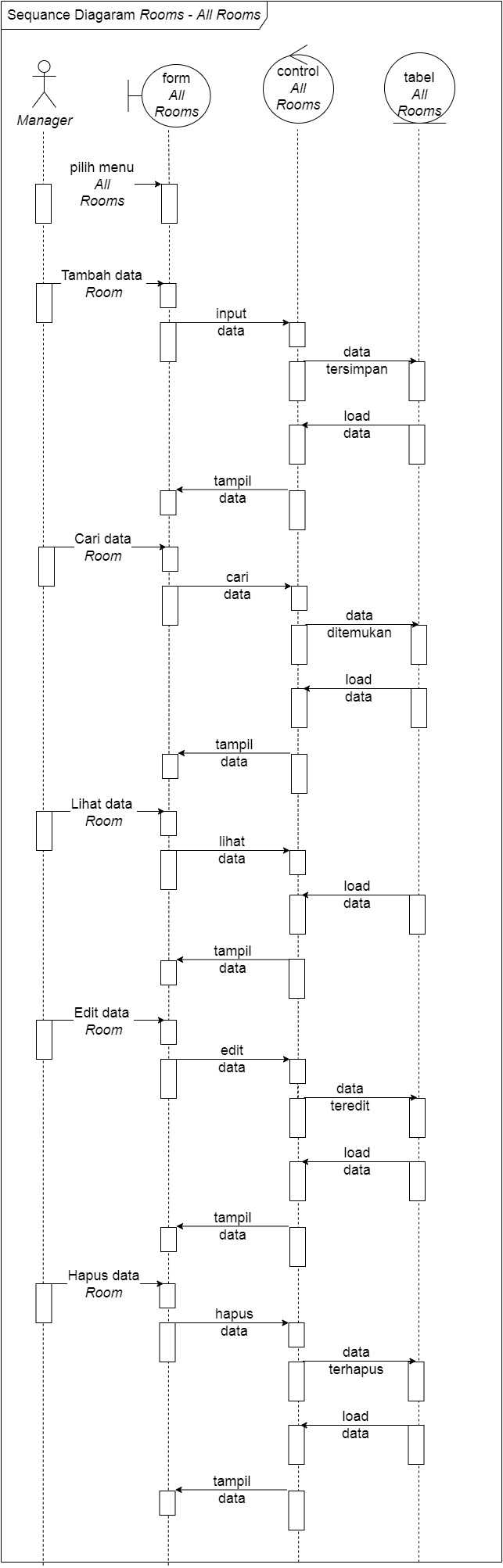
Gambar 4. 3. Sequence diagram Available Rooms Manager

1. *Sequence Diagram Unavailable Rooms* menggambarkan alur sistem pada data yang digunakan oleh *Manager* untuk melihat, menambah, mencari, mengedit dan menghapus data.



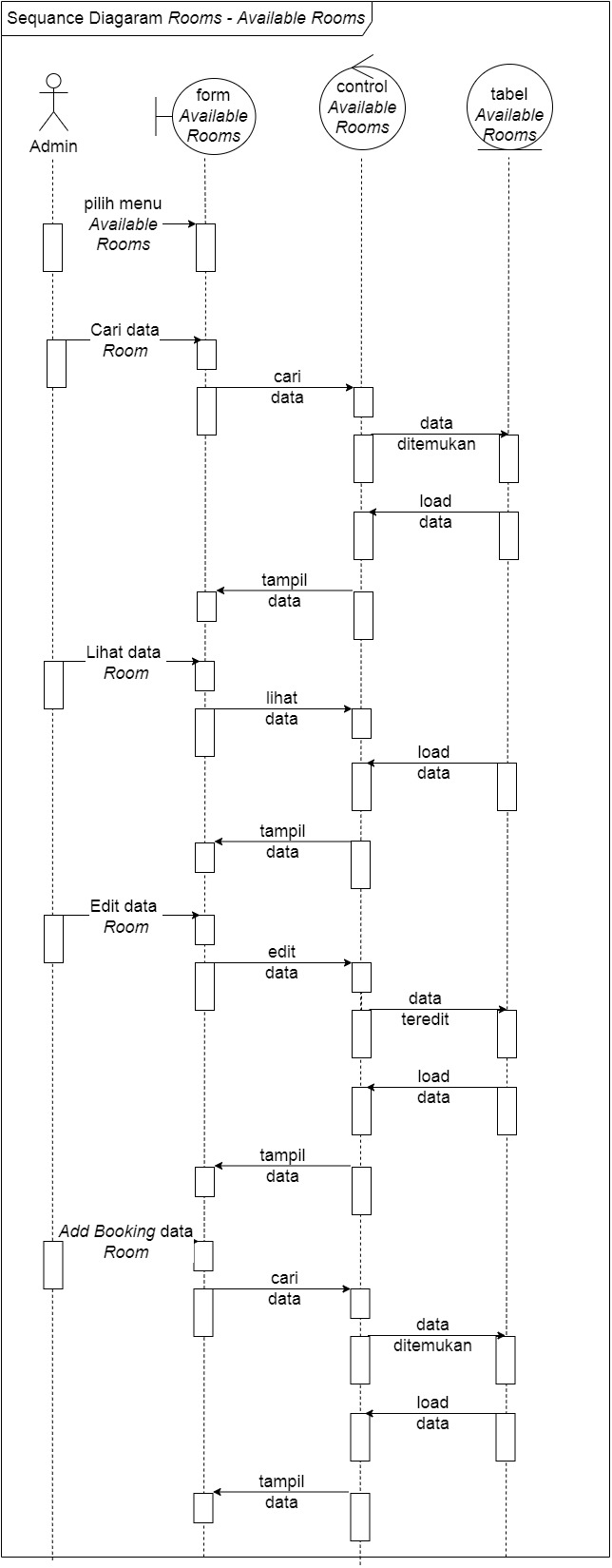
Gambar 4. 4. Sequence diagram Unavailable Rooms Manager

1. *Sequence Diagram All Rooms* menggambarkan alur sistem pada data yang digunakan oleh *Manager* untuk melihat, menambah, mencari, mengedit dan menghapus data.



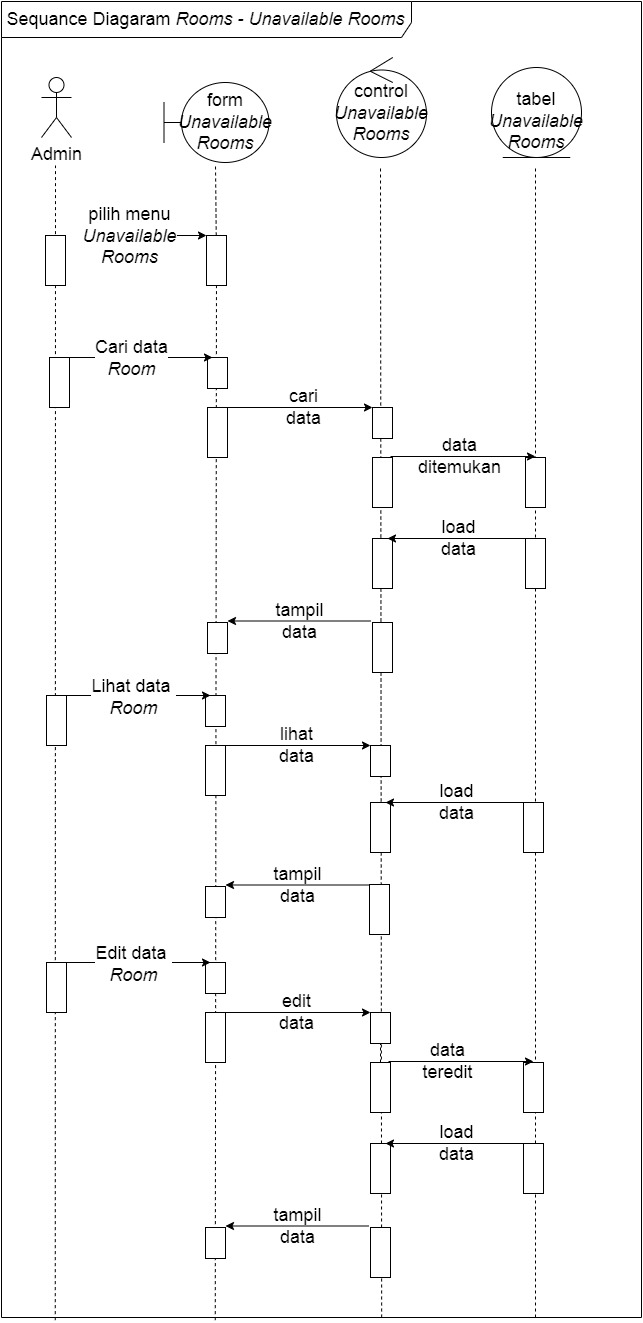
Gambar 4. 5.Sequence diagram All Rooms Manager

1. *Sequence Diagram Room* - Admin
2. *Sequence Diagram Available Rooms* menggambarkan alur sistem pada data yang digunakan oleh Adminuntuk melihat, mencari dan mengedit data.



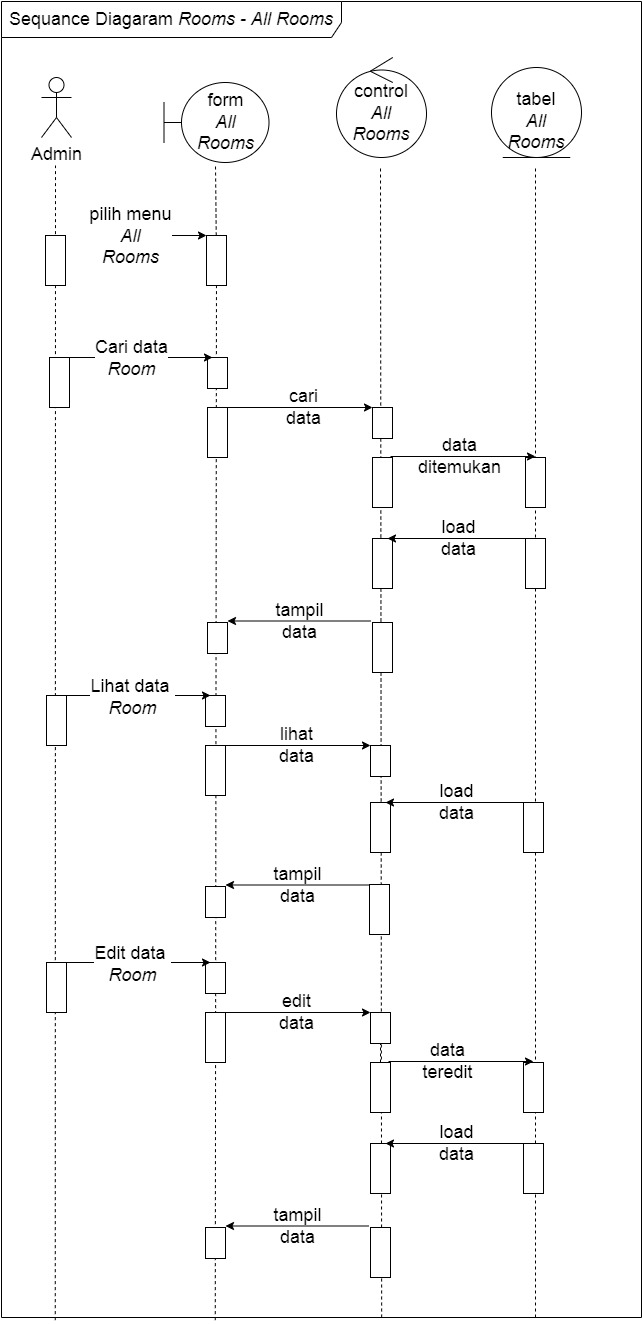
Gambar 4. 6. Sequence diagram Available Rooms Admin

1. *Sequence Diagram Unavailable Rooms* menggambarkan alur sistem pada data yang digunakan oleh Adminuntuk melihat, mencari dan mengedit data.



Gambar 4. 7. Sequence diagram Unavailable Rooms Admin

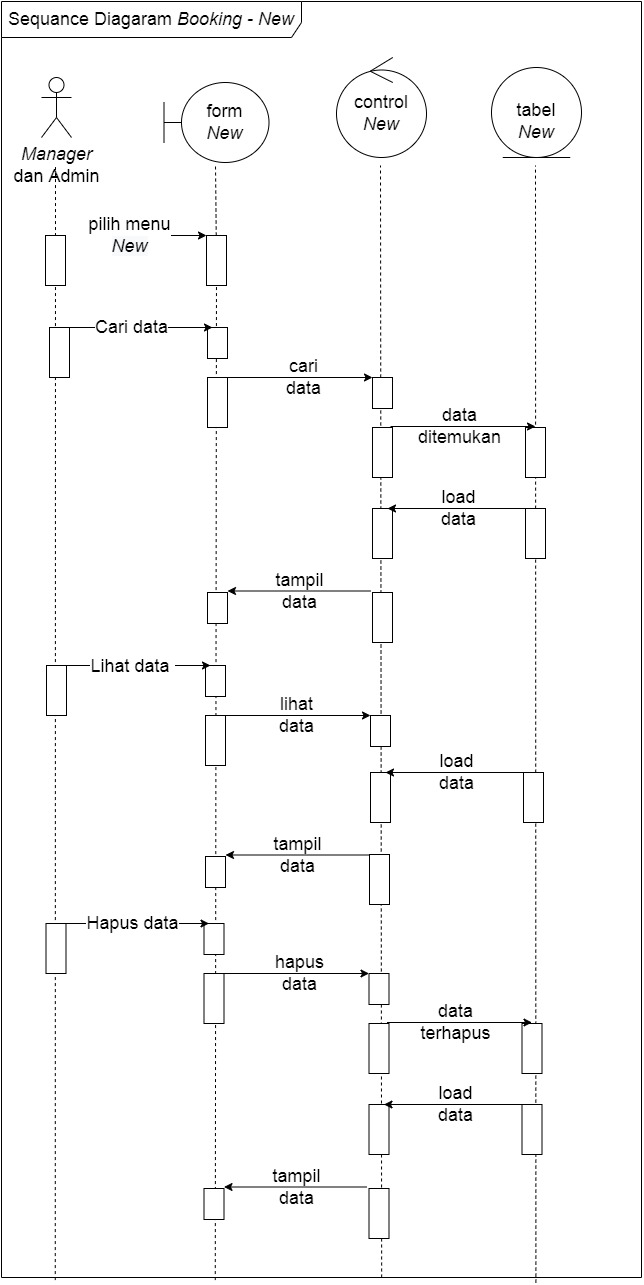
1. *Sequence Diagram All Rooms* menggambarkan alur sistem pada data yang digunakan oleh Adminuntuk melihat, mencari dan mengedit data.



Gambar 4. 8. Sequence diagram All Rooms Admin

1. ***Sequence diagram* Booking**
2. *Sequence diagram New* – *Manager* dan Admin

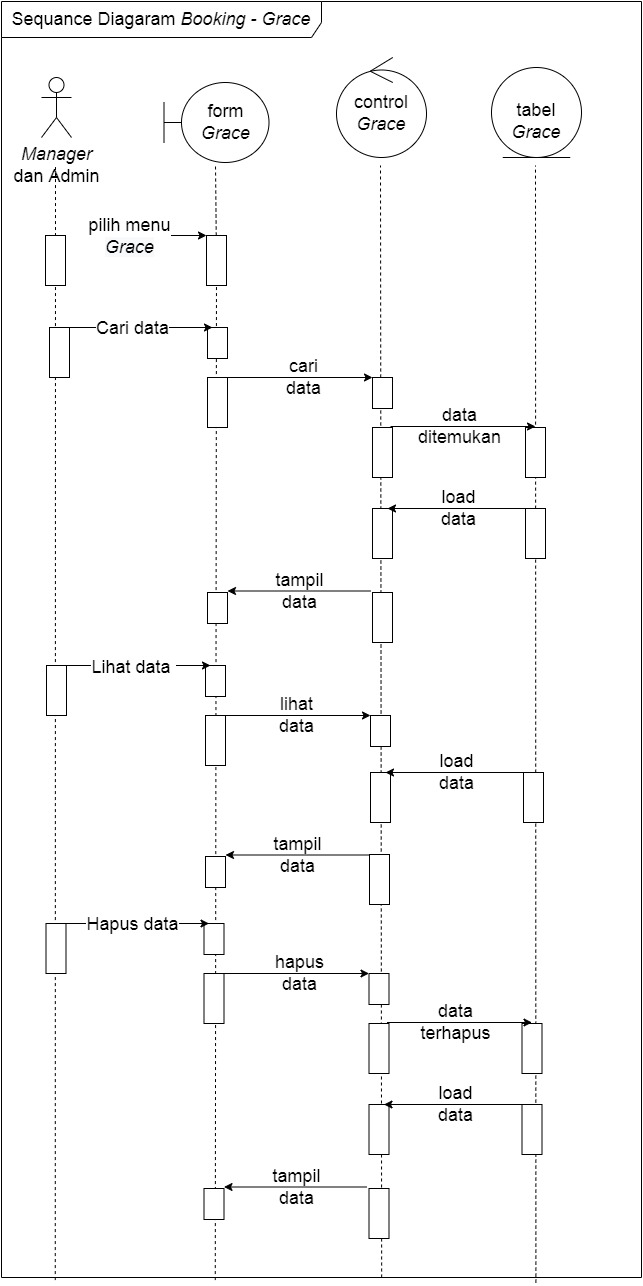
*Sequence Diagram New* menggambarkan alur sistem pada data yang digunakan oleh *Manager* dan Admin untuk mencari, melihat dan menghapus data.



Gambar 4. 9. Sequence diagram New Manager dan Admin

1. *Sequence diagram Grace* – *Manager* dan Admin

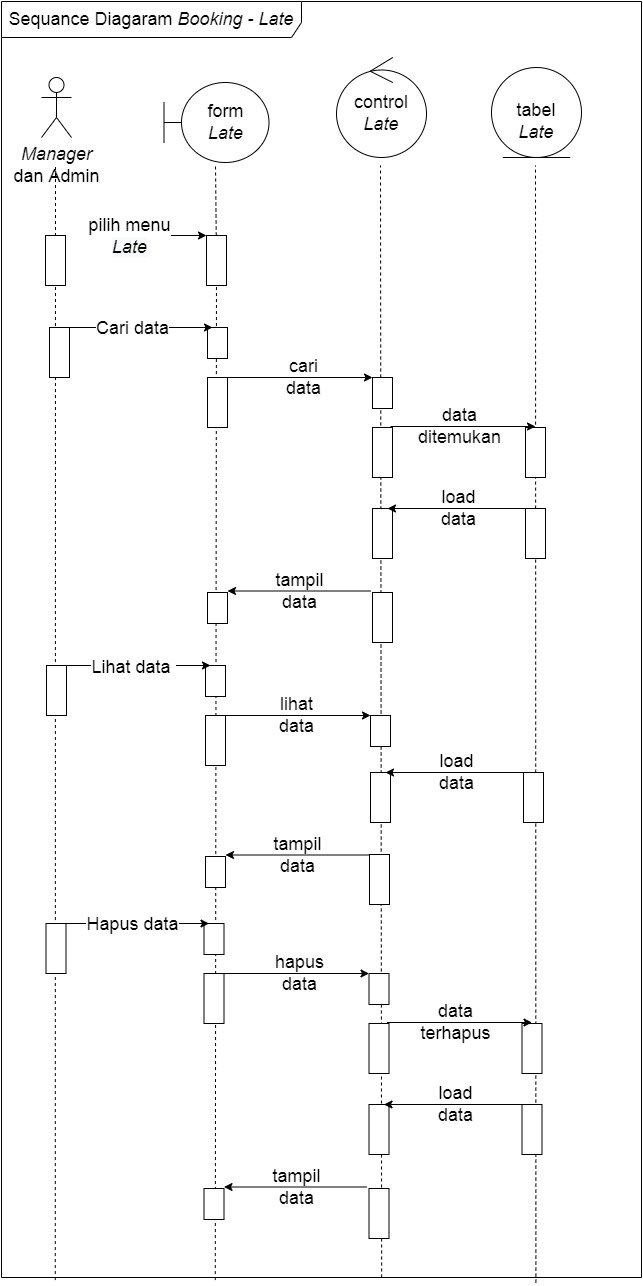
*Sequence Diagram Grace* menggambarkan alur sistem pada data yang digunakan oleh *Manager* dan Admin untuk mencari, melihat dan menghapus data.



Gambar 4. 10. Sequence diagram Grace Manager dan Admin

1. *Sequence diagram Late* – *Manager* dan Admin

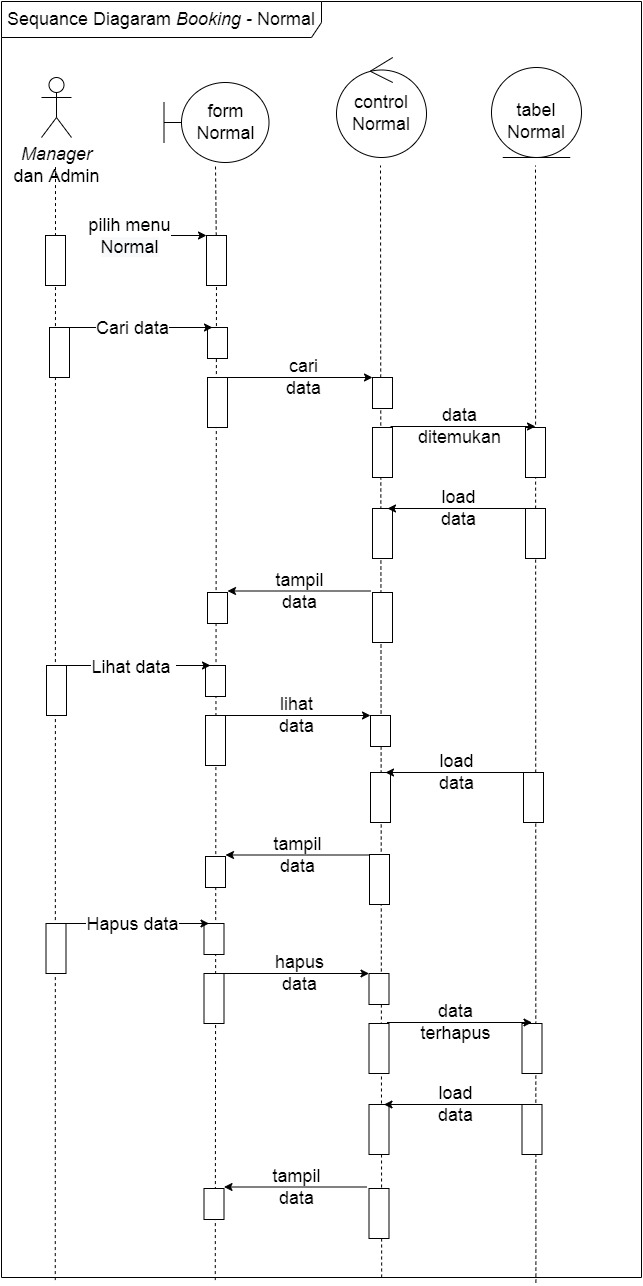
*Sequence Diagram Late* menggambarkan alur sistem pada data yang digunakan oleh *Manager* dan Admin untuk mencari, melihat dan menghapus data.



Gambar 4. 11. Sequence diagram Late Manager dan Admin

1. *Sequence diagram* Normal– *Manager* dan Admin

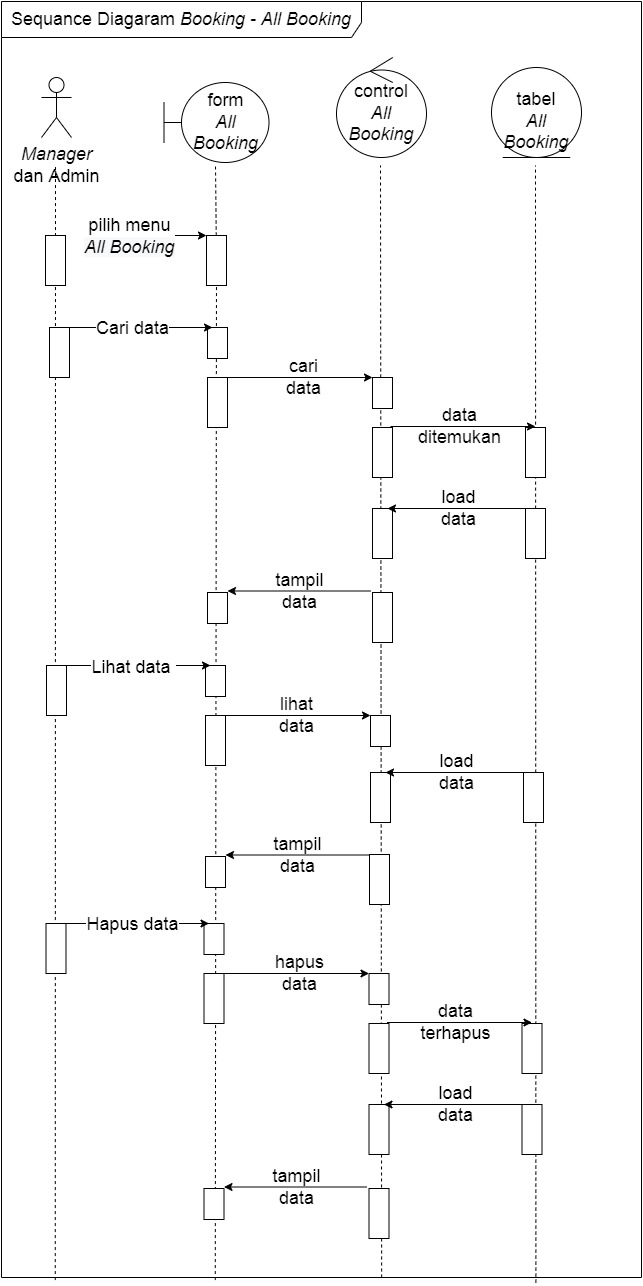
*Sequence Diagram* Normal menggambarkan alur sistem pada data yang digunakan oleh *Manager* dan Admin untuk mencari, melihat dan menghapus data.



Gambar 4. 12. Sequence diagram Normal Manager dan Admin

1. *Sequence diagram All Booking* – *Manager* dan Admin

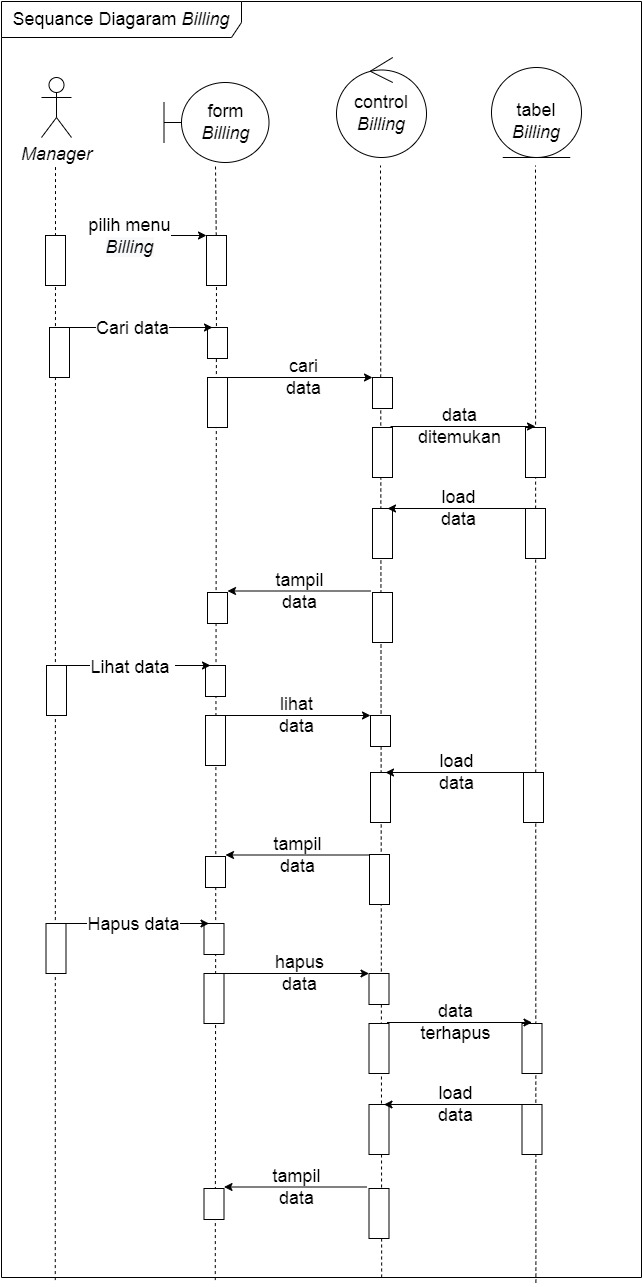
*Sequence Diagram All Booking* menggambarkan alur sistem pada data yang digunakan oleh *Manager* dan Admin untuk mencari, melihat dan menghapus data.



Gambar 4. 13. Sequence diagram All Booking Manager dan Admin

1. ***Sequence diagram Billing***
   * 1. *Sequence diagram Billing - Manager*

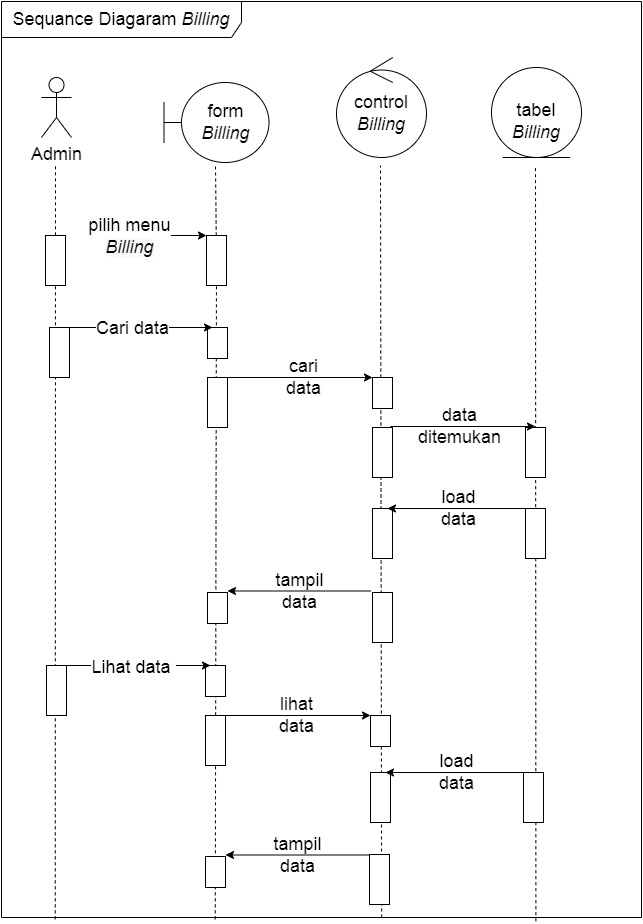
*Sequence Diagram Billing* menggambarkan alur sistem pada data yang digunakan oleh *Manager* untuk melihat, mencari dan menghapus data.



Gambar 4. 14. Sequence diagram Billing Manager

* + 1. *Sequence diagram Billing -* Admin

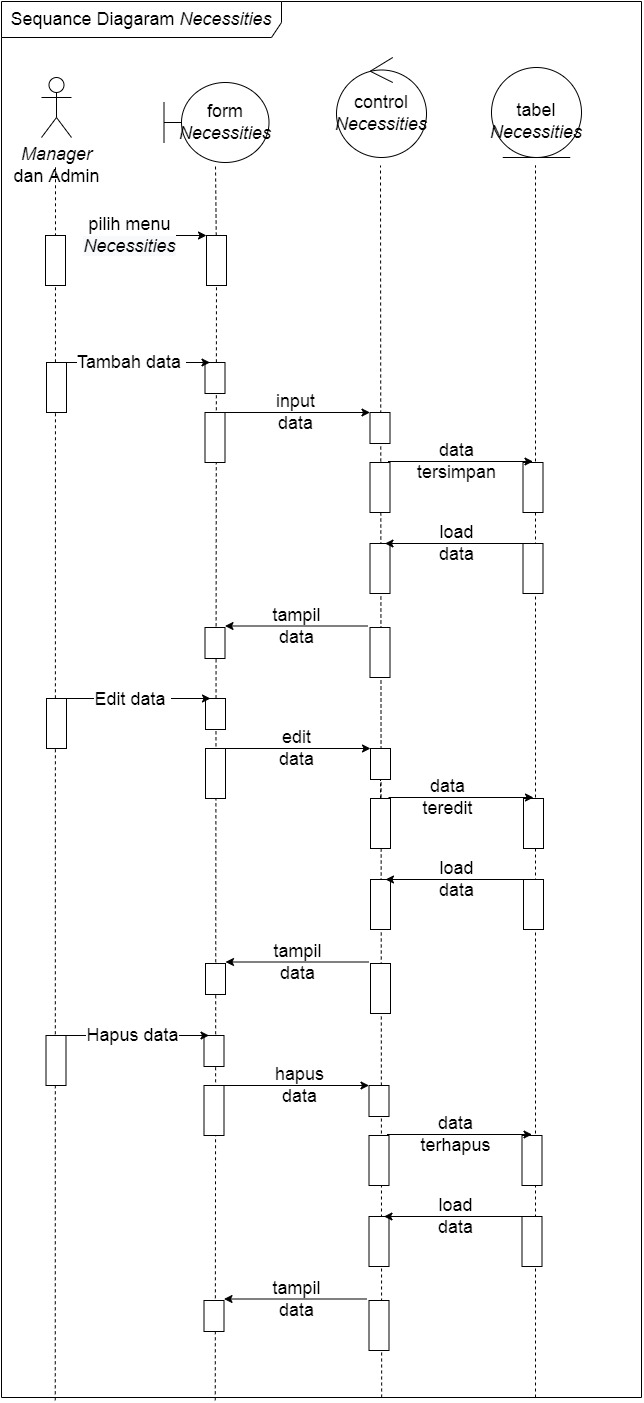
*Sequence Diagram Billing* menggambarkan alur sistem pada data yang digunakan olehAdmin untuk melihat dan mencari data.



Gambar 4. 15. Sequence diagram Billing Admin

1. ***Sequence diagram Necessities***

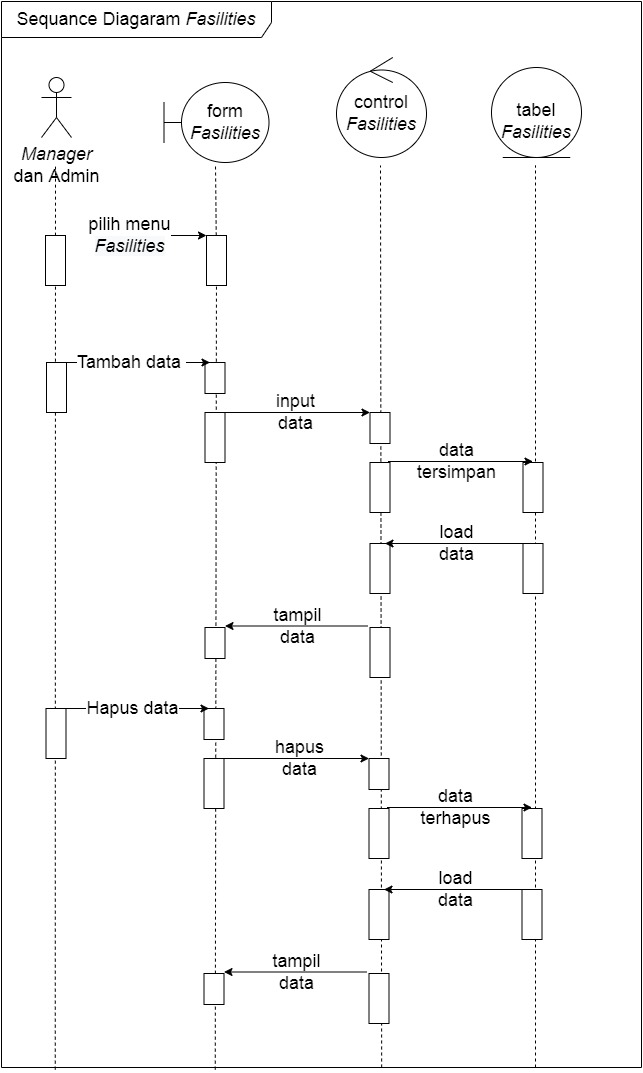
*Sequence Diagram Necessities* menggambarkan alur sistem pada data yang digunakan oleh *Manager* dan admin untuk menambah, mengedit dan menghapus data.



Gambar 4. 16. Sequence diagram Necessities Manager dan admin

1. ***Sequence diagram Fasilities***

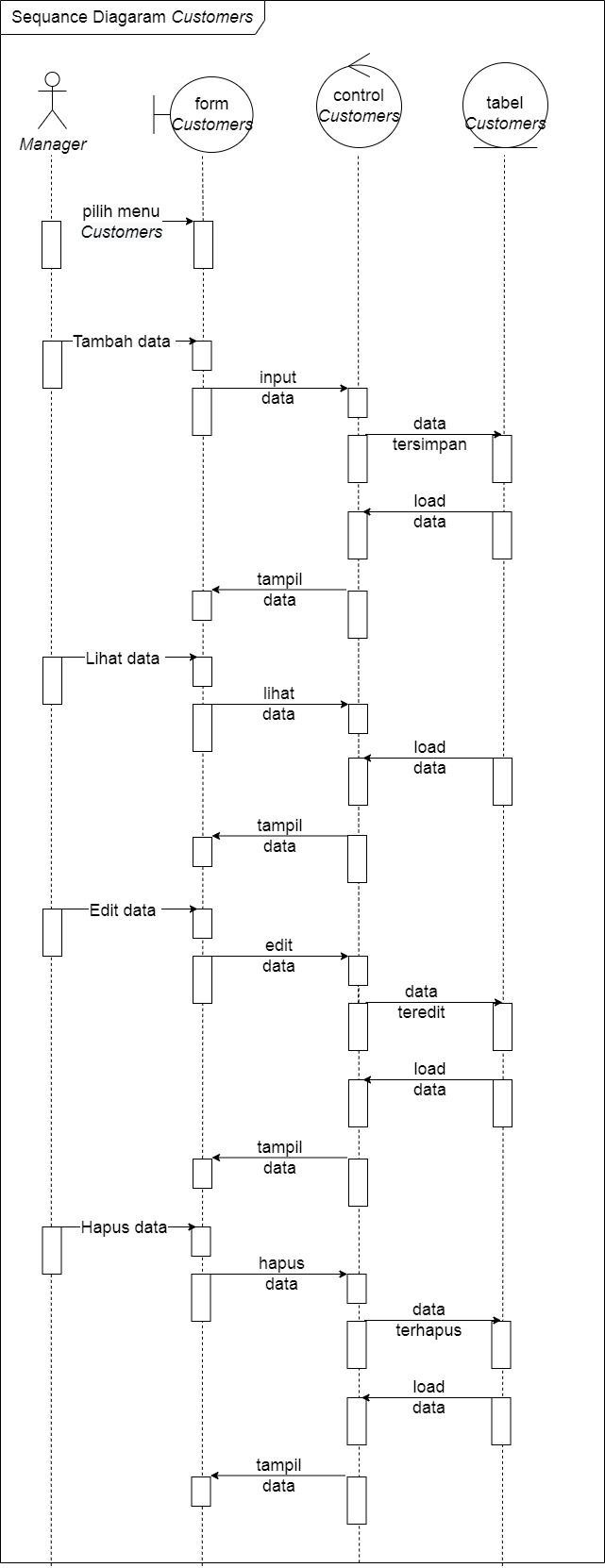
*Sequence Diagram Fasilities* menggambarkan alur sistem pada data yang digunakan oleh *Manager* dan admin untuk menambah dan menghapus data.



Gambar 4. 17. Sequence diagram Fasilities Manager dan admin

1. ***Sequence diagram Customer***
   * 1. *Sequence diagram Customer - Manager*

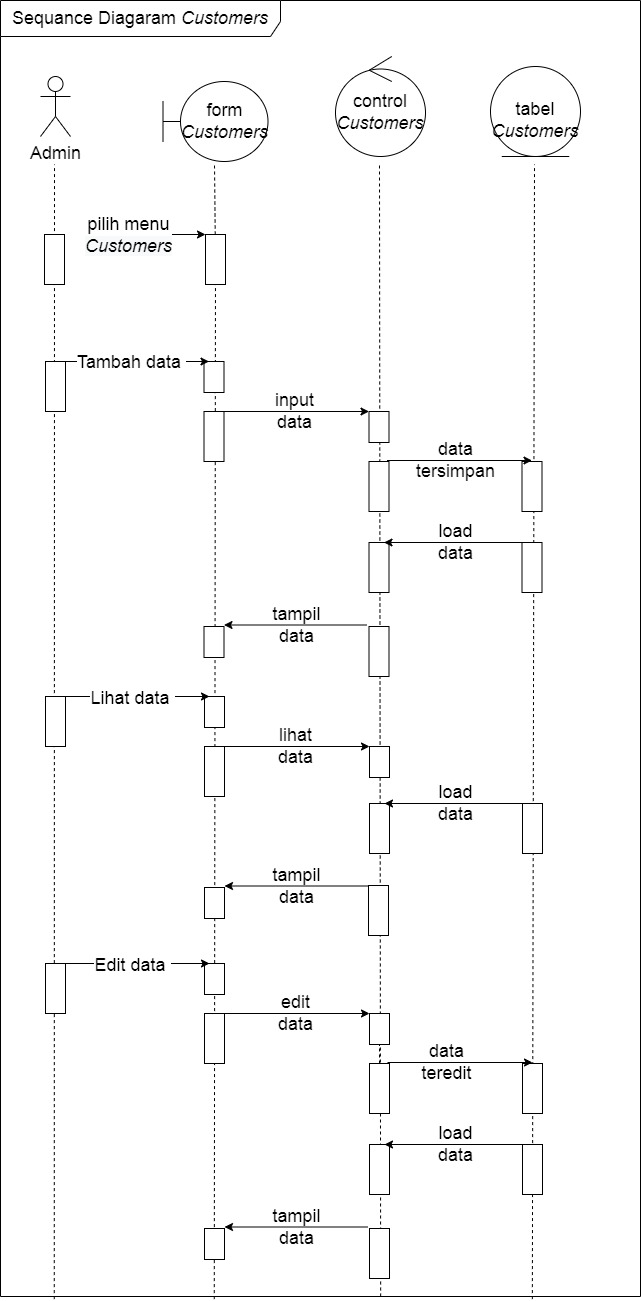
*Sequence Diagram Customer* menggambarkan alur sistem pada data yang digunakan oleh *Manager* untuk menambah, melihat, mengedit dan menghapus data.



Gambar 4. 18. Sequence diagram Customer Manager

* + 1. *Sequence diagram Customer -* Admin

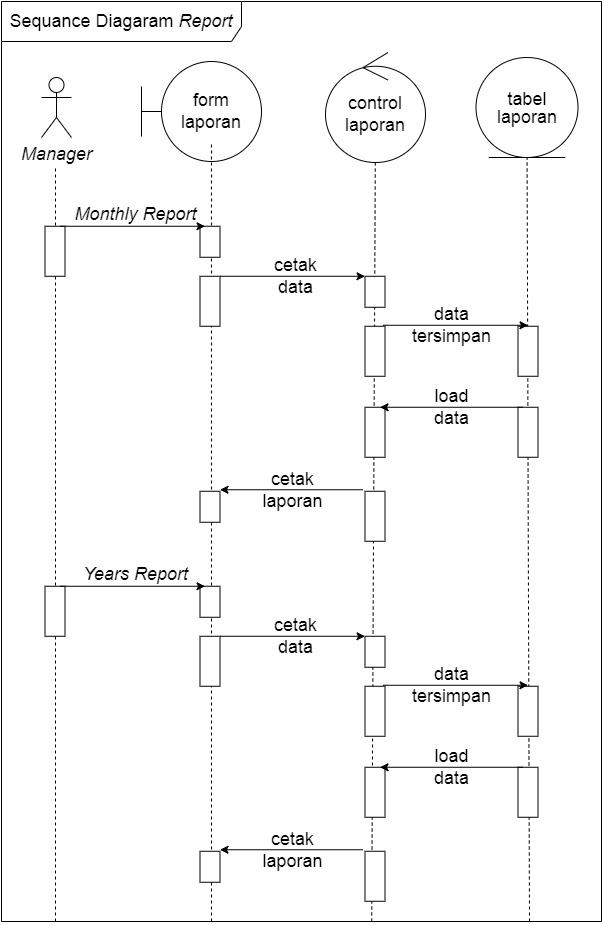
*Sequence Diagram Customer* menggambarkan alur sistem pada data yang digunakan oleh Adminuntuk menambah, melihat dan mengedit data.



Gambar 4. 19. Sequence diagram Customer Admin

1. ***Sequence diagram Report***

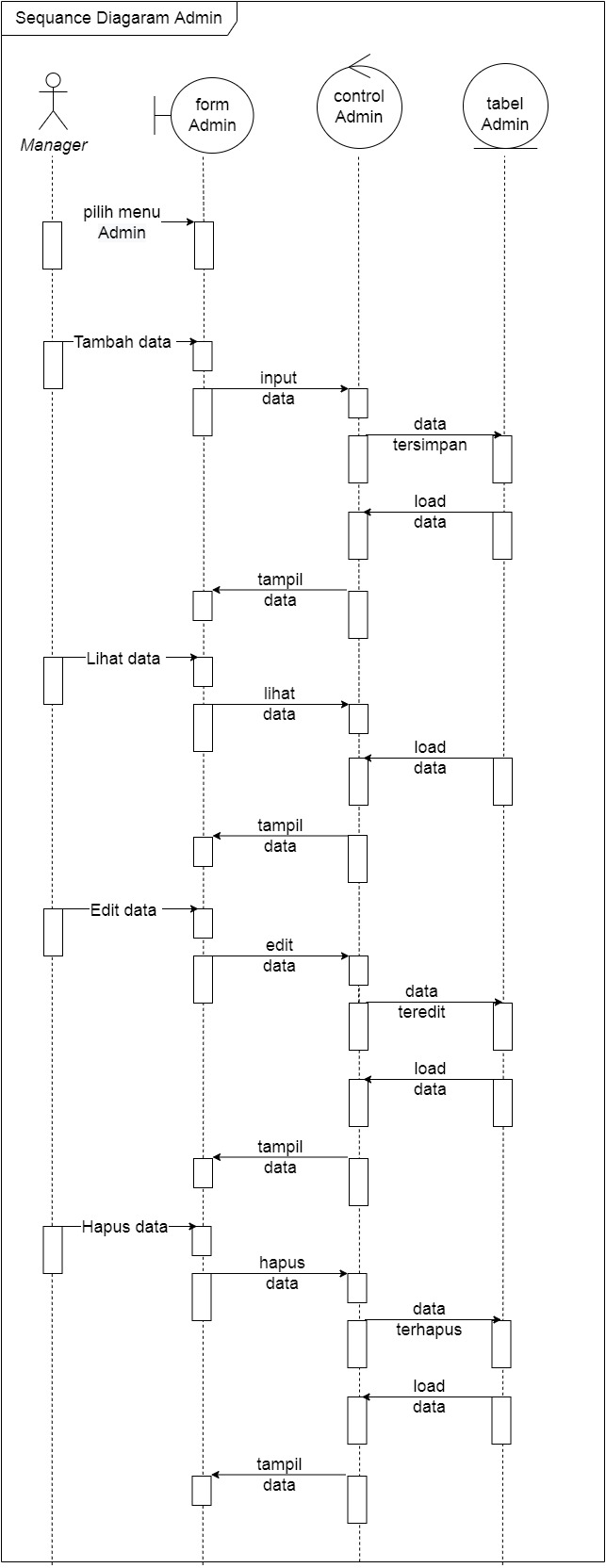
*Sequence Diagram Report* menggambarkan alur sistem pada data yang digunakan oleh *Manager* untuk mencetak data laporan.



Gambar 4. 20. Sequence diagram Report Manager

1. ***Sequence diagram* Admin**

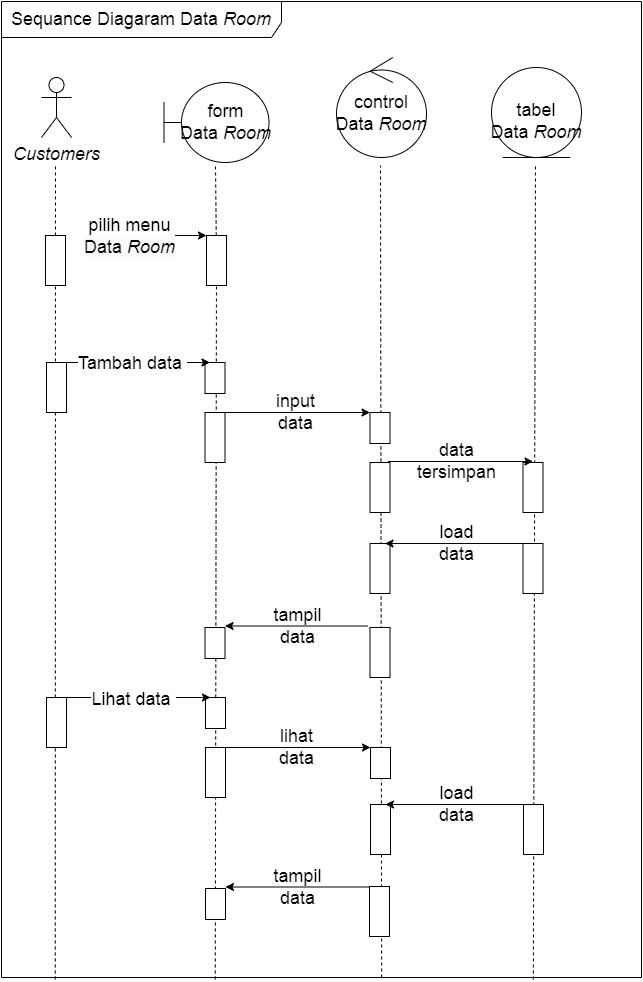
*Sequence Diagram* Admin menggambarkan alur sistem pada data yang digunakan oleh *Manager* untuk menambah, melihat, mengedit dan menghapus data.



Gambar 4. 21. Sequence diagram Admin Manager

1. ***Sequence diagram* Data *Room***

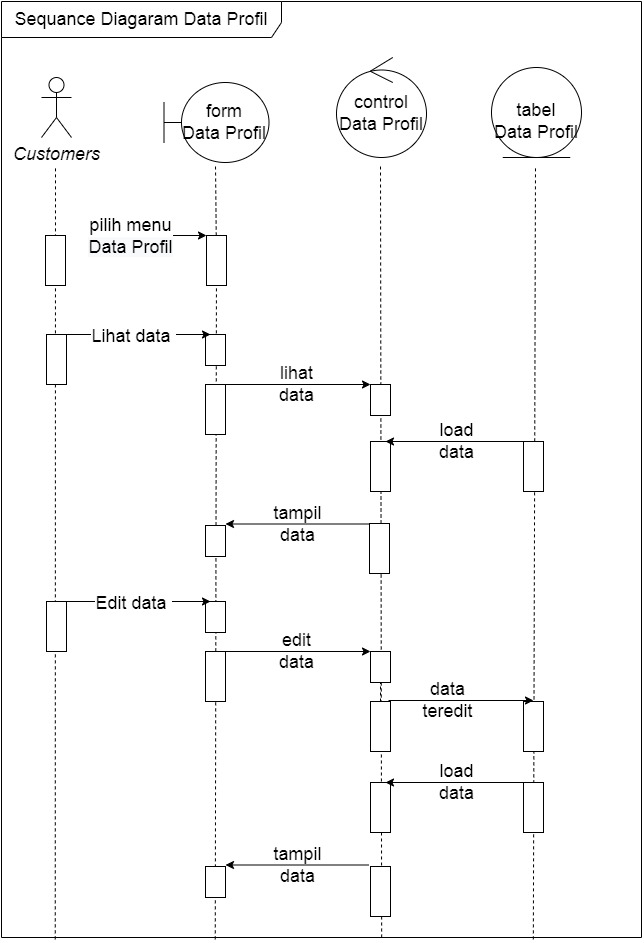
*Sequence Diagram* Data *Room* menggambarkan alur sistem pada data yang digunakan oleh *Customer* untuk menambah dan melihat data.



Gambar 4. 22. Sequence diagram Data Room Customer

1. ***Sequence diagram* Data Profil**

*Sequence Diagram* Data Profil menggambarkan alur sistem pada data yang digunakan oleh *Customer* untuk mengedit dan melihat data.



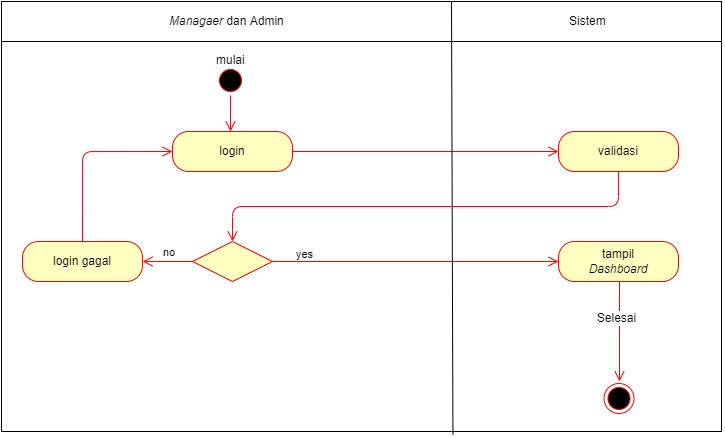
Gambar 4. 23. Sequence diagram Data Profil Customer

### Activity Diagram

*Activity diagram* menggambarkan aktivitas yang dilakukan oleh *actor*. Aktivitas-aktivitas tersebut adalah mengelola data monitor, data perhitungan SPK, data permohonan, master data, data jam keluar dan data jam masuk.

* + - 1. ***Activity diagram* *Dashboard***

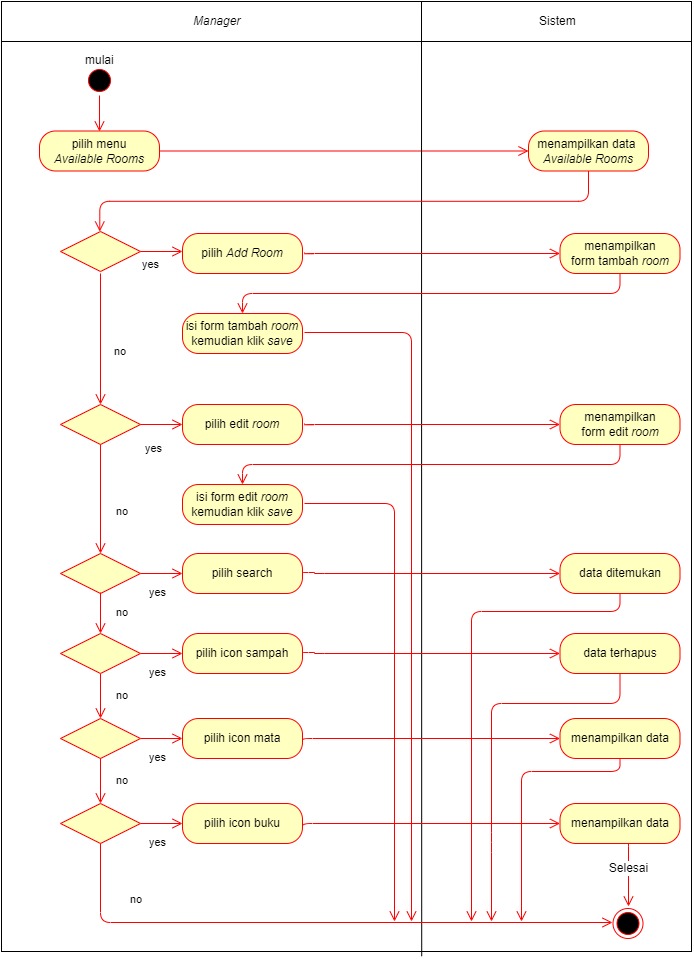
*Activity Diagram Dashboard* yang menyajikan informasi mengenai indikator utama dari aktifitas sistem secara sekilas dalam layar tunggal.



Gambar 4. 24. Activity diagram Dashboard

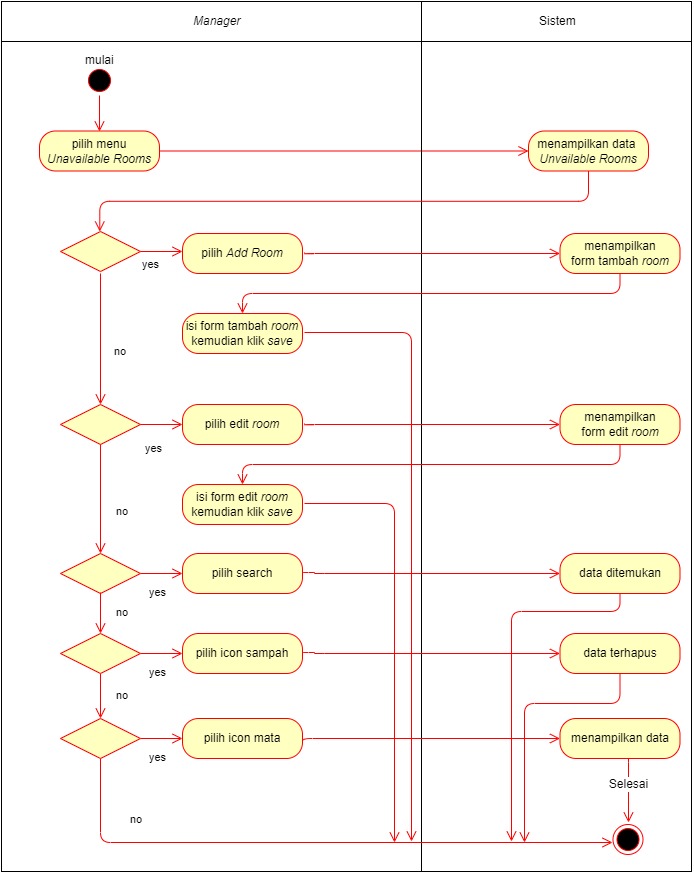
* + - 1. ***Activity diagram Rooms***

1. *Activity diagram**Room* - *Manager*
2. *Activity diagram**Available Rooms* menggambarkan alur sistem pada data yang digunakan oleh *Manager* untuk melihat, menambah, mencari, mengedit dan menghapus data.



Gambar 4. 25. Activity diagram Available Rooms Manager

1. *Activity diagram**Unavailable Rooms* menggambarkan alur sistem pada data yang digunakan oleh *Manager* untuk melihat, menambah, mencari, mengedit dan menghapus data.



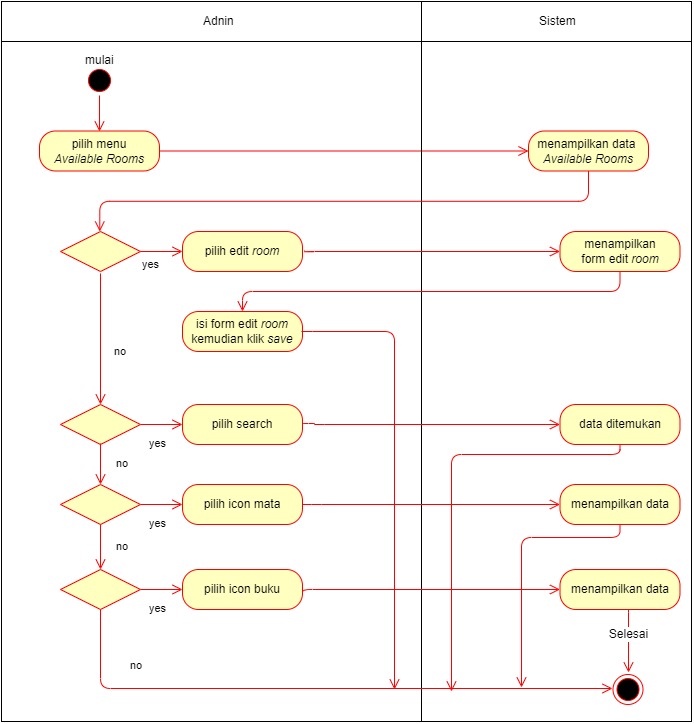
Gambar 4. 26. Activity diagram Unavailable Rooms Manager

1. *Activity diagram**All Rooms* menggambarkan alur sistem pada data yang digunakan oleh *Manager* untuk melihat, menambah, mencari, mengedit dan menghapus data.



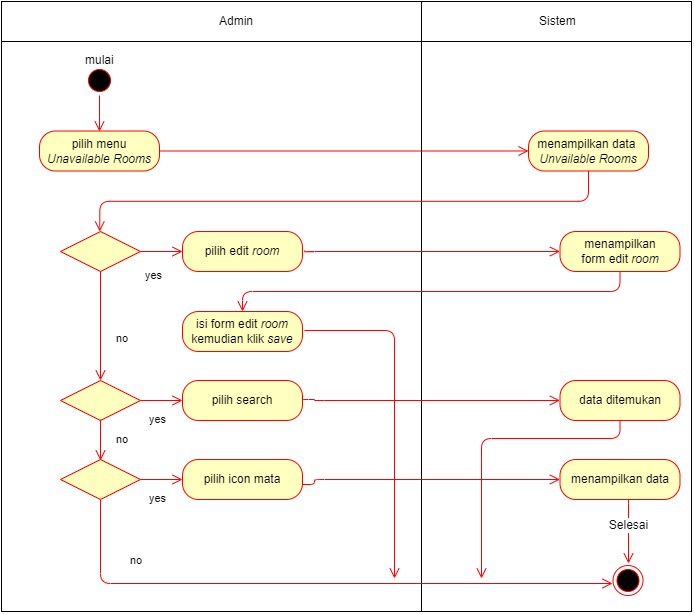
Gambar 4. 27. Activity diagram All Rooms Manager

1. *Activity diagram**Room* - Admin
2. *Activity diagram**Available Rooms* menggambarkan alur sistem pada data yang digunakan oleh Adminuntuk melihat, mencari dan mengedit data.



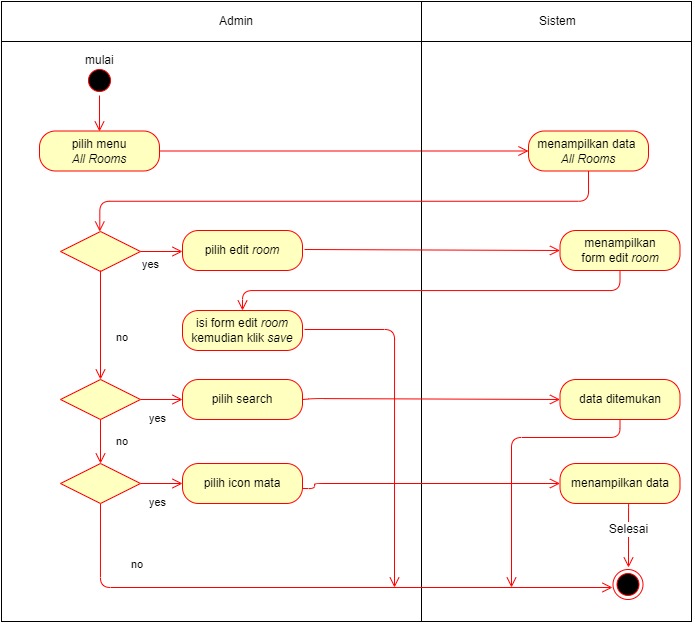
Gambar 4. 28. Activity diagram Available Rooms Admin

1. *Activity diagram**Unavailable Rooms* menggambarkan alur sistem pada data yang digunakan oleh Adminuntuk melihat, mencari dan mengedit data.



Gambar 4. 29. Activity diagram Unavailable Rooms Admin

1. *Activity diagram**All Rooms* menggambarkan alur sistem pada data yang digunakan oleh Adminuntuk melihat, mencari dan mengedit data.

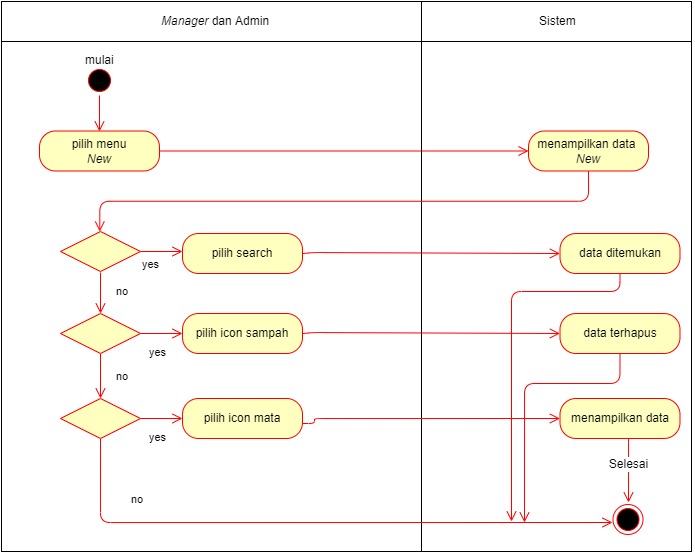


Gambar 4. 30. Activity diagram All Rooms Admin

* + - 1. ***Activity diagram* Booking**

1. *Activity diagram**New* – *Manager* dan Admin

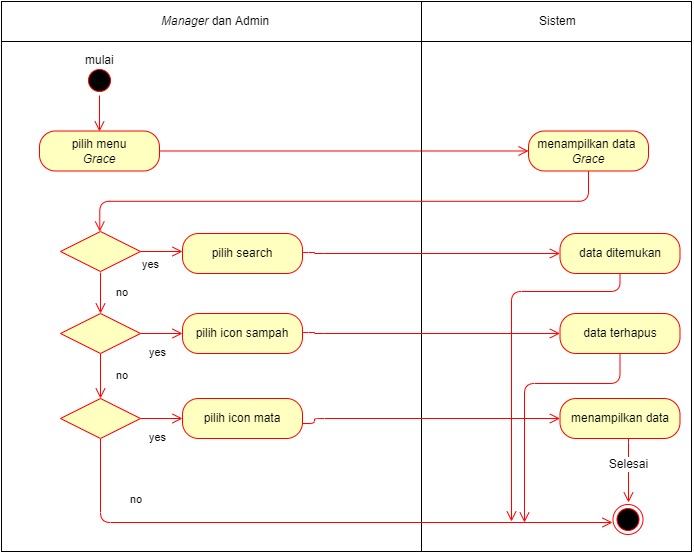
*Activity diagram**New* menggambarkan alur sistem pada data yang digunakan oleh *Manager* dan Admin untuk mencari, melihat dan menghapus data.



Gambar 4. 31. Activity diagram New Manager dan Admin

1. *Activity diagram**Grace* – *Manager* dan Admin

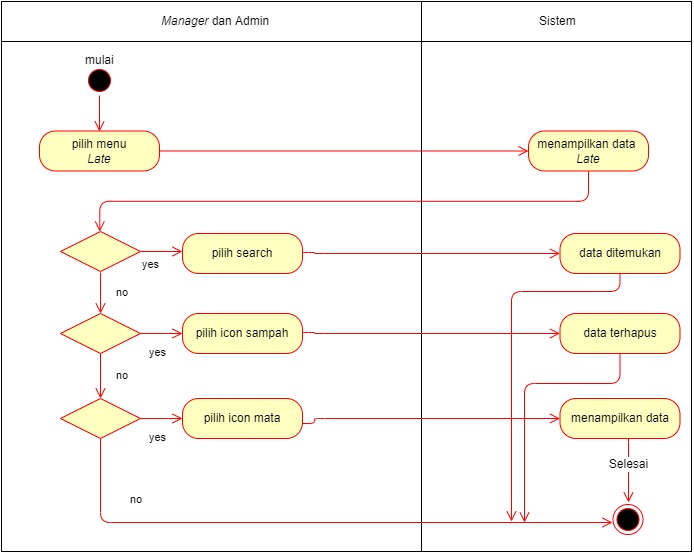
*Activity diagram**Grace* menggambarkan alur sistem pada data yang digunakan oleh *Manager* dan Admin untuk mencari, melihat dan menghapus data.



Gambar 4. 32. Activity diagram Grace Manager dan Admin

1. *Activity diagram**Late* – *Manager* dan Admin

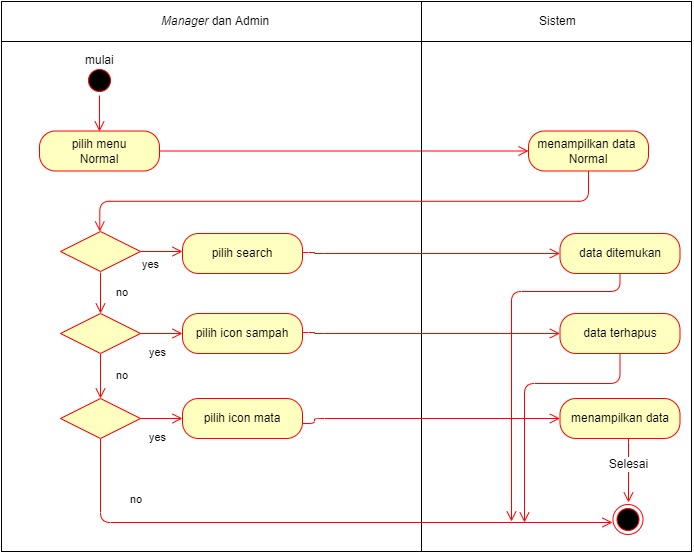
*Activity diagram**Late* menggambarkan alur sistem pada data yang digunakan oleh *Manager* dan Admin untuk mencari, melihat dan menghapus data.



Gambar 4. 33. Activity diagram Late Manager dan Admin

1. *Activity diagram*Normal– *Manager* dan Admin

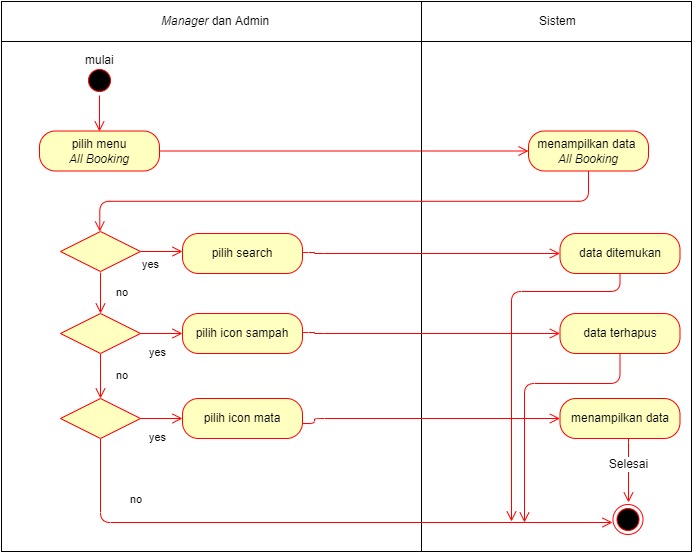
*Activity diagram*Normal menggambarkan alur sistem pada data yang digunakan oleh *Manager* dan Admin untuk mencari, melihat dan menghapus data.



Gambar 4. 34. Activity diagram Normal Manager dan Admin

1. *Activity diagram**All Booking* – *Manager* dan Admin

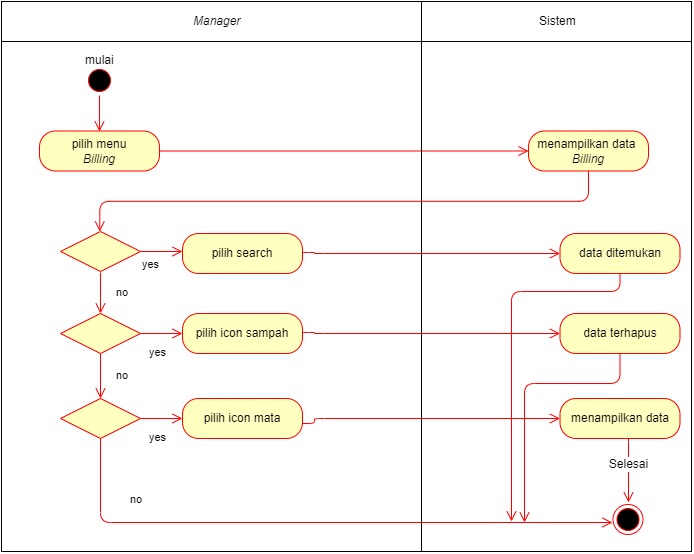
*Activity diagram**All Booking* menggambarkan alur sistem pada data yang digunakan oleh *Manager* dan Admin untuk mencari, melihat dan menghapus data.



Gambar 4. 35. Activity diagram All Booking Manager dan Admin

* + - 1. ***Activity diagram Billing***
    1. *Activity diagram**Billing - Manager*

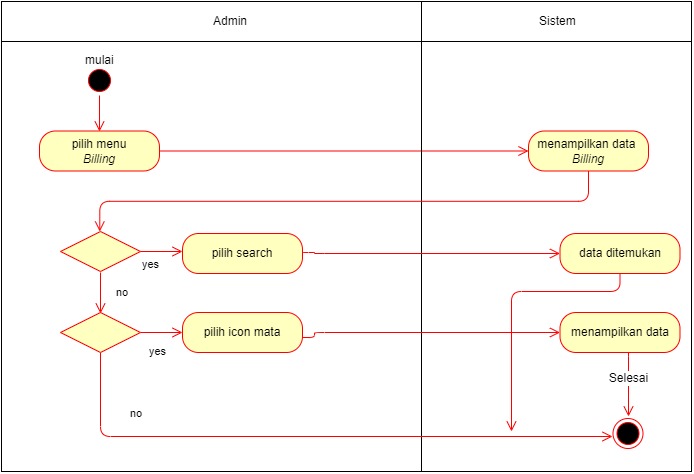
*Activity diagram**Billing* menggambarkan alur sistem pada data yang digunakan oleh *Manager* untuk melihat, mencari dan menghapus data.



Gambar 4. 36. Activity diagram Billing Manager

* + 1. *Activity diagram**Billing -* Admin

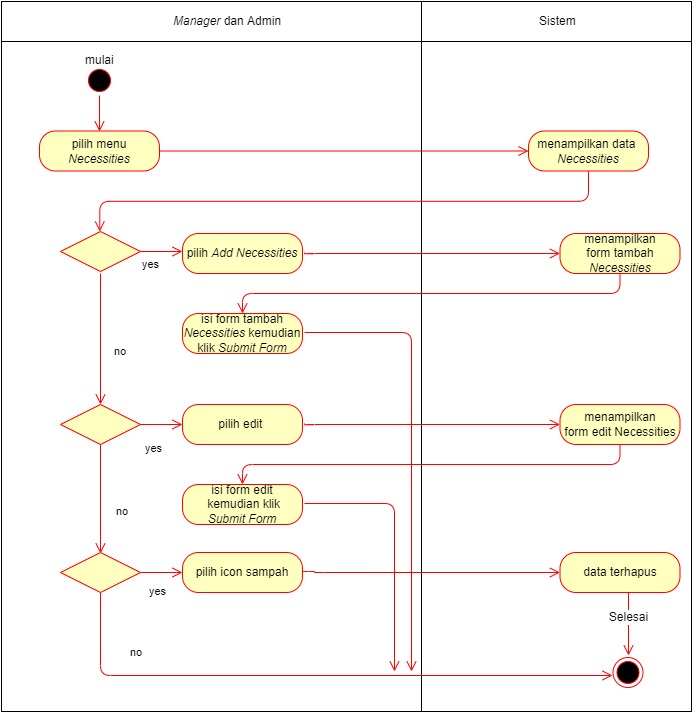
*Activity diagram**Billing* menggambarkan alur sistem pada data yang digunakan olehAdmin untuk melihat dan mencari data.



Gambar 4. 37. Activity diagram Billing Admin

* + - 1. ***Activity diagram Necessities***

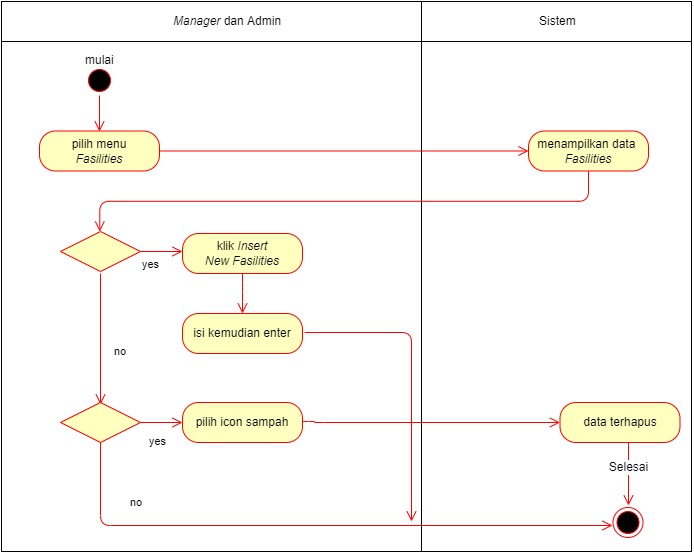
*Activity diagram**Necessities* menggambarkan alur sistem pada data yang digunakan oleh *Manager* dan admin untuk menambah, mengedit dan menghapus data.



Gambar 4. 38. Activity diagram Necessities Manager dan admin

* + - 1. ***Activity diagram Fasilities***

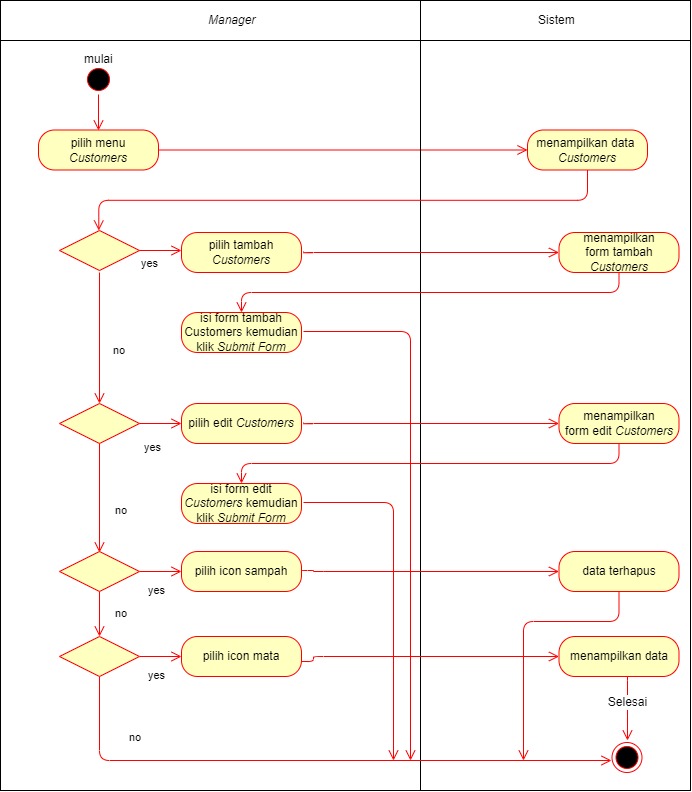
*Activity diagram**Fasilities* menggambarkan alur sistem pada data yang digunakan oleh *Manager* dan admin untuk menambah dan menghapus data.



Gambar 4. 39 Activity diagram Fasilities Manager dan admin

* + - 1. ***Activity diagram Customer***
    1. *Activity diagram**Customer - Manager*

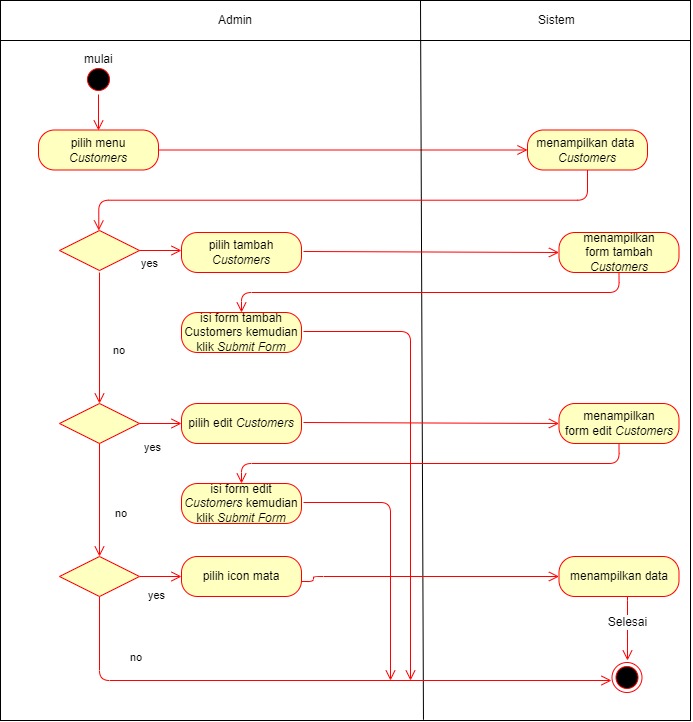
*Activity diagram**Customer* menggambarkan alur sistem pada data yang digunakan oleh *Manager* untuk menambah, melihat, mengedit dan menghapus data.



Gambar 4. 40 Activity diagram Customer Manager

* + 1. *Activity diagram**Customer -* Admin

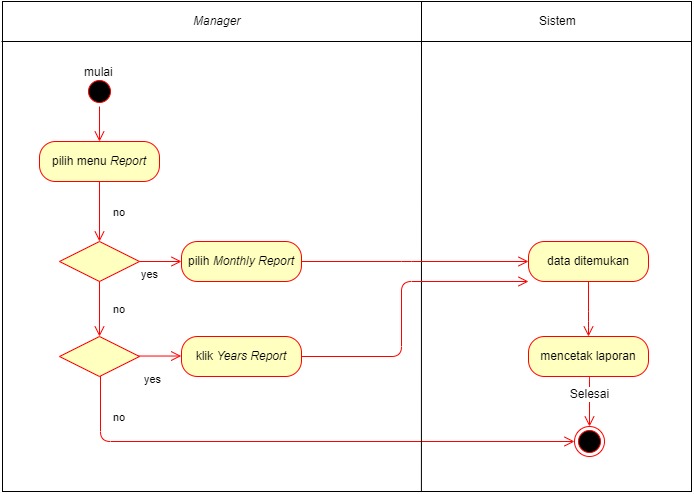
*Activity diagram**Customer* menggambarkan alur sistem pada data yang digunakan oleh Adminuntuk menambah, melihat dan mengedit data.



Gambar 4. 41Activity diagram Customer Admin

1. ***Activity diagram Report***

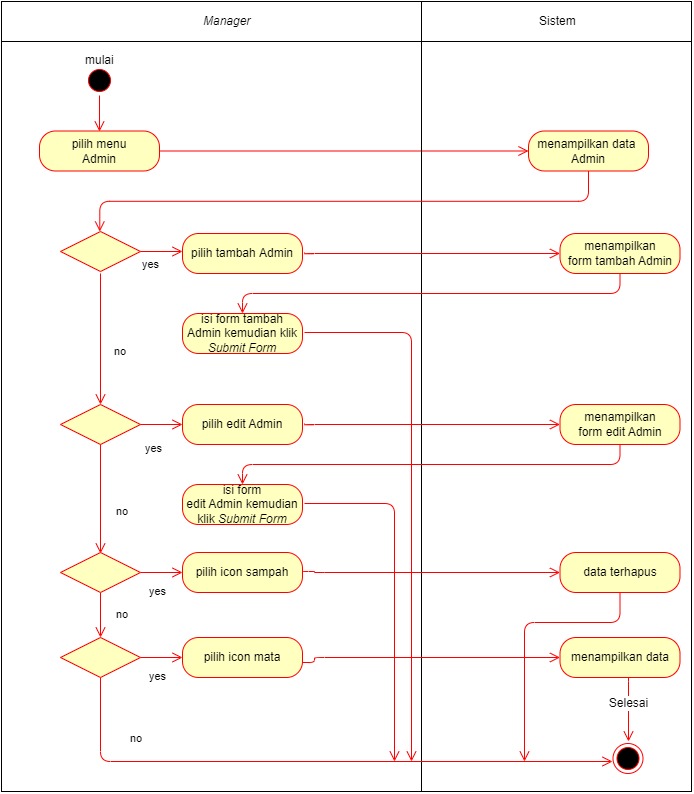
*Activity diagram**Report* menggambarkan alur sistem pada data yang digunakan oleh *Manager* untuk mencetak data laporan.



Gambar 4. 42 Activity diagram Report Manager

1. ***Activity diagram* Admin**

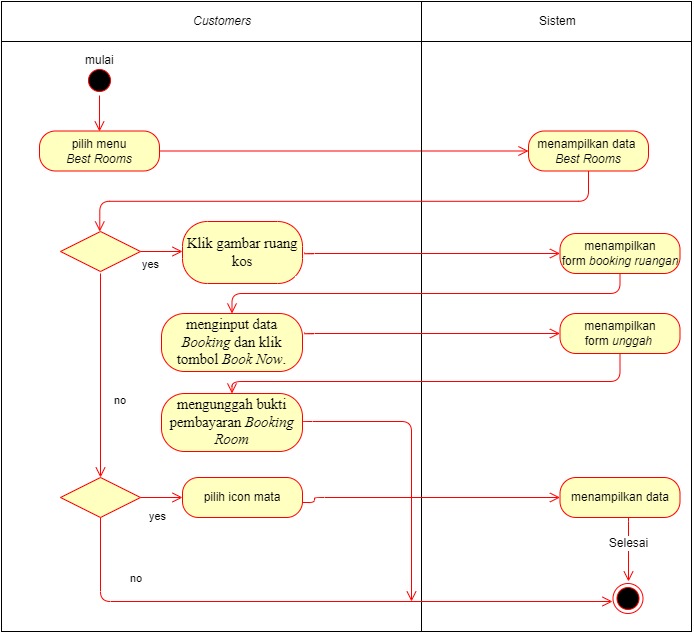
*Activity diagram*Admin menggambarkan alur sistem pada data yang digunakan oleh *Manager* untuk menambah, melihat, mengedit dan menghapus data.



Gambar 4. 43 Activity diagram Admin Manager

1. ***Activity diagram* Data *Room***

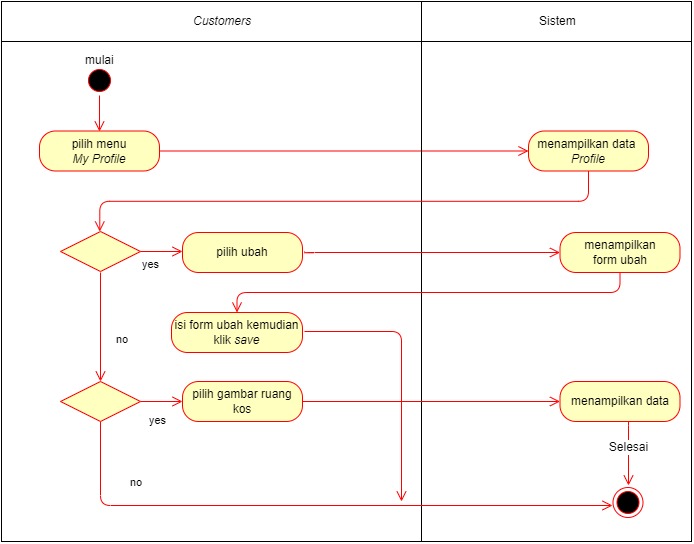
*Activity diagram*Data *Room* menggambarkan alur sistem pada data yang digunakan oleh *Customer* untuk menambah dan melihat data.



Gambar 4. 44 Activity diagram Data Room Customer

1. ***Activity diagram* Data Profil**

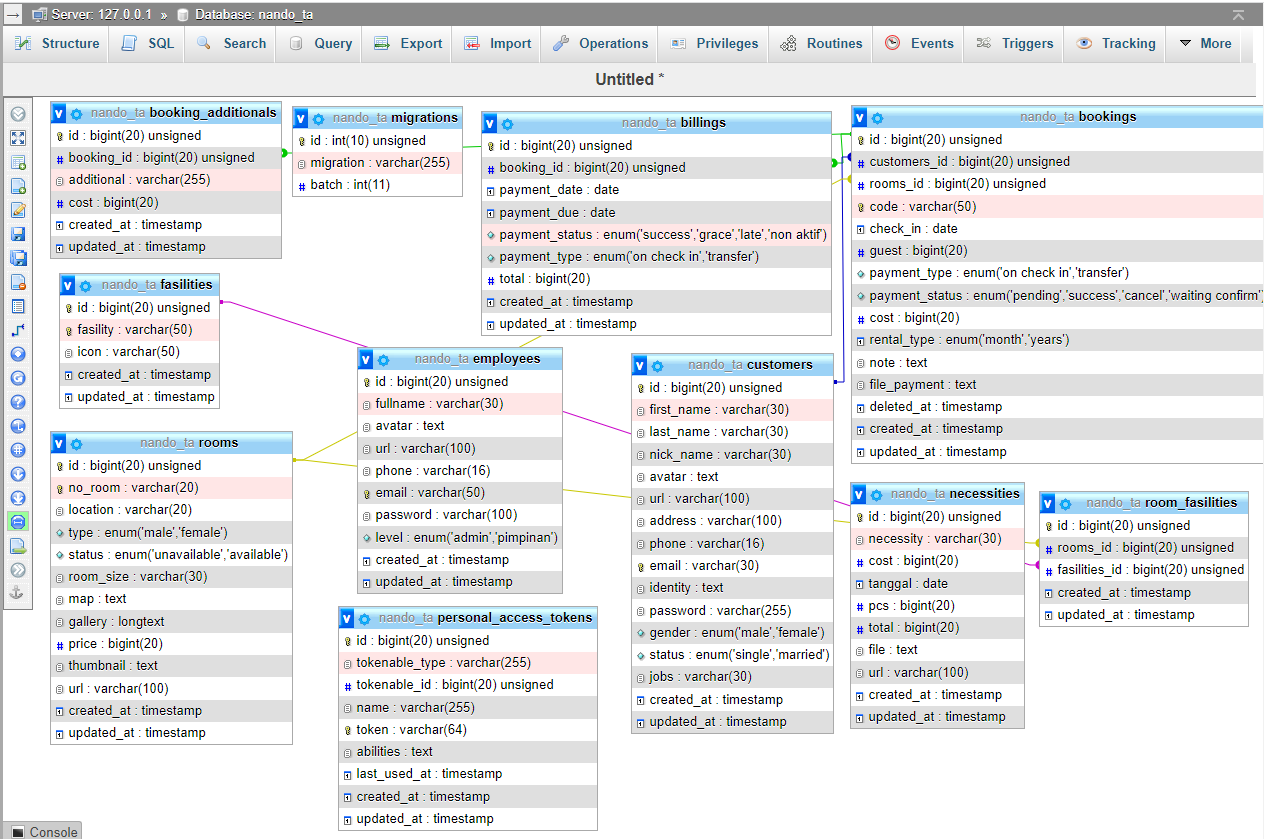
*Activity diagram*Data Profil menggambarkan alur sistem pada data yang digunakan oleh *Customer* untuk mengedit dan melihat data.



Gambar 4. 45 Activity diagram Data Profil Customer

### Class Diagram

*Class Diagram* adalah diagram yang menggambarkan struktur sistem dari kelas-kelas yang dibuat untuk membangun sistem.



Gambar 4. 46 Class Diagram Sistem Management Kost Dpavillion Menggunakan Laravel

*Class Diagram* diatas, berisi mengenai gambaran database yang berada di sistem di bagi menjadi beberapa tabel yaitu : billings, bookings, booking\_additionals, customers, employees, fasilities, migrations, necessities, personal\_access\_tokens, rooms dan room\_fasilities.

### Perancangan *Database*

*Database* merupakan suatu media atau bagian dari sistem untuk menyimpan data, dapat berjalan secara tunggal dan berdiri sendiri. Data yang tersimpan di dalam *database* hanya dapat ditampilkan, diubah, dan dihapus dari komputer yang telah terpasang aplikasi tersebut oleh *user* yang memiliki hak akses sebagai *administrator*.

Rancangan basis data yang penulis gunakan adalah sebagai berikut :

**Data Billings**

Nama Tabel : billings

Primary Key : id

Tabel 4. 19. tabel billings

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| id | bigint (20) | Primary Key (PK) |
| booking\_id | bigint (20) | Foreign Key (FK) |
| payment\_date | date |  |
| payment\_due | date |  |
| payment\_status | enum |  |
| payment\_type | enum |  |
| total | bigint (20) |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

**Data Migrations**

Nama Tabel : migrations

Primary Key : id

Tabel 4. 20 tabel migrations

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| id | Int (10) | Primary Key (PK) |
| migrations | varchar (255) |  |
| batch | Int (11) |  |

**Data Booking Additionals**

Nama Tabel : booking\_additionals

Primary Key : id

Tabel 4. 21tabel booking\_additionals

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| id | bigint (20) | Primary Key (PK) |
| booking\_id | bigint (20) | Foreign Key (FK) |
| additional | varchar (255) |  |
| cost | bigint (20) |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

**Data Bookings**

Nama Tabel : bookings

Primary Key : id

Tabel 4. 22 tabel bookings

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| id | bigint (20) | Primary Key (PK) |
| customers\_id | bigint (20) | Foreign Key (FK) |
| rooms\_id | bigint (20) | Foreign Key (FK) |
| code | varchar (50) | Foreign Key (FK) |
| check\_in | date |  |
| guest | bigint (20) |  |
| payment\_type | enum |  |
| payment\_status | enum |  |
| cost | bigint (20) |  |
| rental\_type | enum |  |
| note | text |  |
| file\_payment | text |  |
| deleted\_at | timestamp |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

**Data Employees**

Nama Tabel : enployees

Primary Key : id

Tabel 4. 23 tbl\_user

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| id | bigint (20) | Primary Key (PK) |
| fullname | varchar (30) |  |
| avatar | text |  |
| url | varchar (100) |  |
| phone | varchar (16) |  |
| email | varchar (50) | Foreign Key (FK) |
| password | varchar (100) |  |
| level | enum |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

**Data Rooms**

Nama Tabel : rooms

Primary Key : id

Tabel 4. 24 tabel rooms

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| id | bigint (20) | Primary Key (PK) |
| no\_room | varchar (20) | Foreign Key (FK) |
| location | varchar (20) |  |
| type | enum |  |
| status | enum |  |
| room\_size | varchar (20) |  |
| map | text |  |
| gallery | longtext |  |
| price | bigint (20) |  |
| thumbnail | text |  |
| url | varchar (100) |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

**Data Necessities**

Nama Tabel : necessities

Primary Key : id

Tabel 4. 25 tabel necessities

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| id | bigint (20) | Primary Key (PK) |
| necessity | varchar (30) |  |
| cost | bigint (20) |  |
| tanggal | date |  |
| pcs | bigint (20) |  |
| total | bigint (20) |  |
| file | text |  |
| url | varchar (100) |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

**Data Personal Access**

Nama Tabel : personal\_access\_tokens

Primary Key : id

Tabel 4. 26 tabel personal\_access\_tokens

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| id | bigint (20) | Primary Key (PK) |
| tokenable\_type | varchar (255) | Foreign Key (FK) |
| tokenable\_id | bigint (20) | Foreign Key (FK) |
| name | varchar (255) |  |
| token | varchar (64) | Foreign Key (FK) |
| abilities | text |  |
| last\_used\_at | timestamp |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

**Data Fasilities**

Nama Tabel : fasilities

Primary Key : id

Tabel 4. 27 tabel fasilitas

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| id | bigint (20) | Primary Key (PK) |
| fasility | varchar (50) | Foreign Key (FK) |
| icon | varchar (50) |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

**Data Room Fasilities**

Nama Tabel : room\_fasilities

Primary Key : id

Tabel 4. 28 tabel room\_fasilities

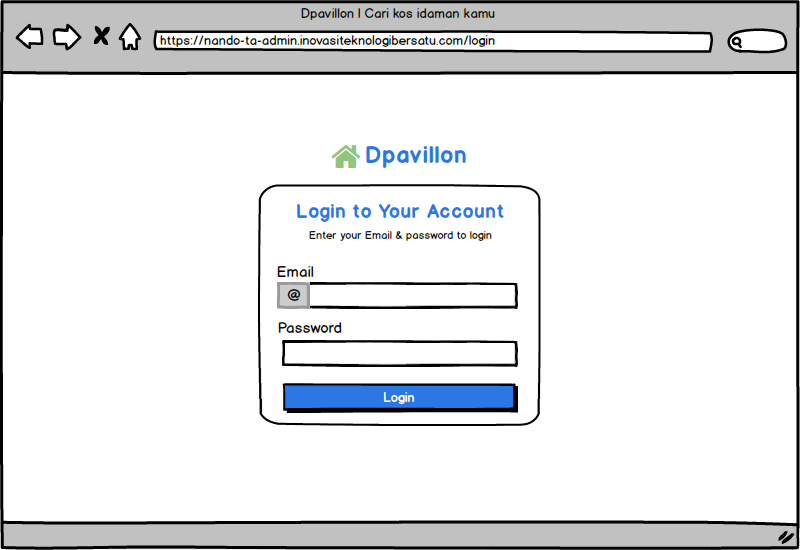
|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| id | bigint (20) | Primary Key (PK) |
| Rooms\_id | bigint (20) | Foreign Key (FK) |
| Fasilities\_id | bigint (20) | Foreign Key (FK) |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

### Perancangan Antarmuka

Perancangan antarmuka program yang diusulkan dibuat sedemikian rupa untuk memudahkan admin dalam mengelola sistem. Rancangan antarmuka ini dibuat menggunakan web sehingga dapat dibuka oleh beberapa *platforms.* Berikut gambar dibawah ini :

1. **Rancangan Halaman *Login***

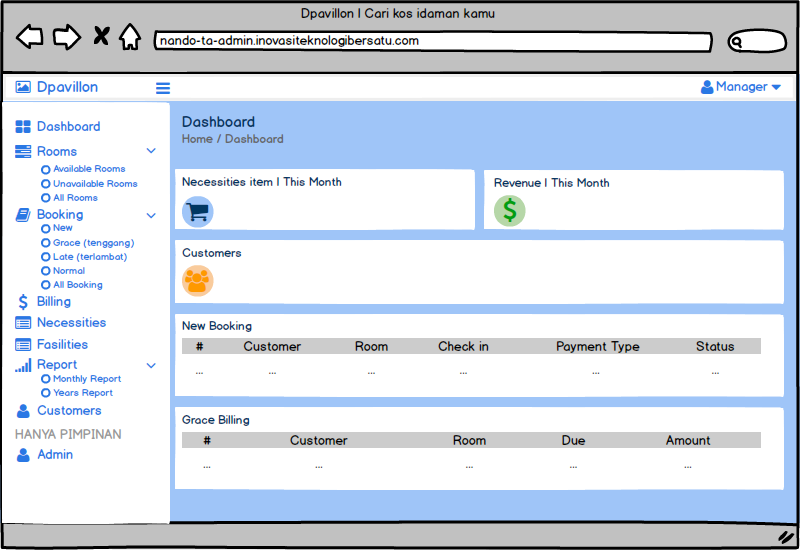
Rancangan halaman *login* ini berisi tampilan *form login* yang digunakan sebelum *user* masuk kedalam menu utama, supaya dapat masuk kedalam menu utama *user* diharuskan memasukan *username* dan *password.*



Gambar 4. 47 Halaman Log In

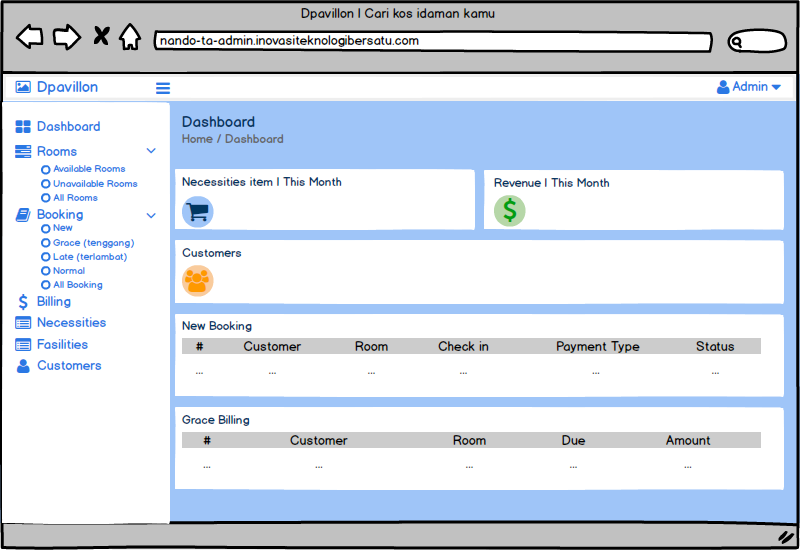
1. **Rancangan Halaman *Dashboard***
2. *Dashboard* – *Manager*

Menyajikan informasi mengenai indikator utama dari aktifitas sistem secara sekilas dalam layar tunggal.



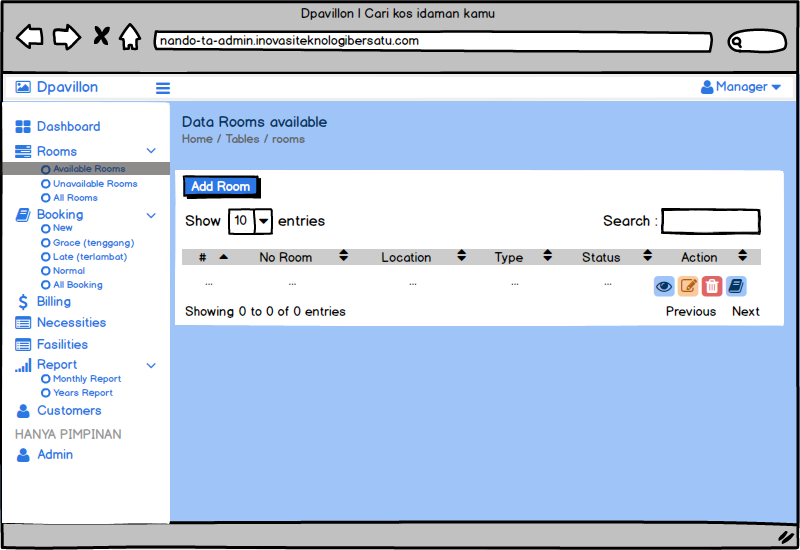
Gambar 4. 48 Halaman Dashboard Manager

1. *Dashboard* - Admin

Menyajikan informasi mengenai indikator utama dari aktifitas sistem secara sekilas dalam layar tunggal.

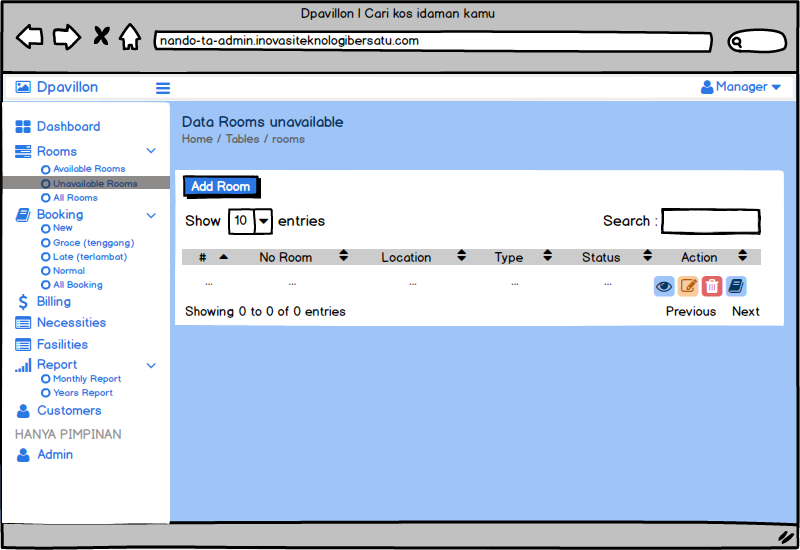
Gambar 4. 49 Halaman Dashboard Admin

1. **Rancangan Halaman *Manager***
2. **Rancangan Halaman *Rooms***
3. Rancangan Halaman *Available Rooms*



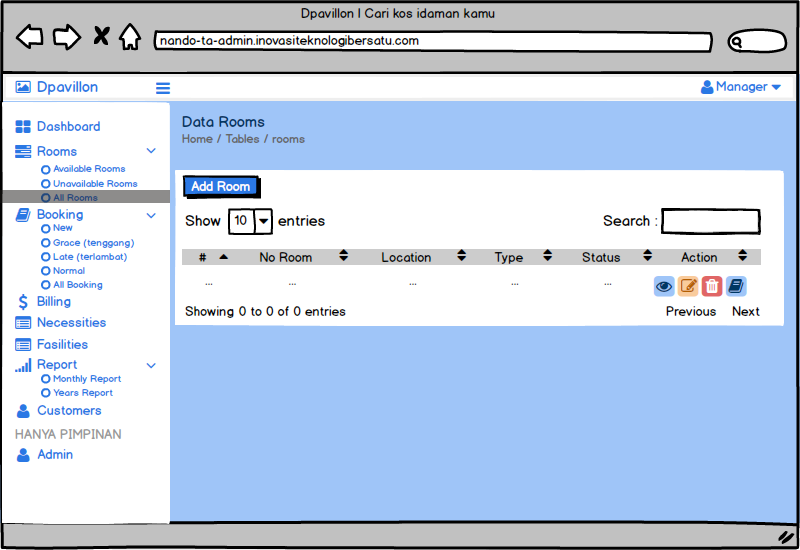
Gambar 4. 50 Available Rooms Manager

1. Rancangan Halaman *Unavailable Rooms*



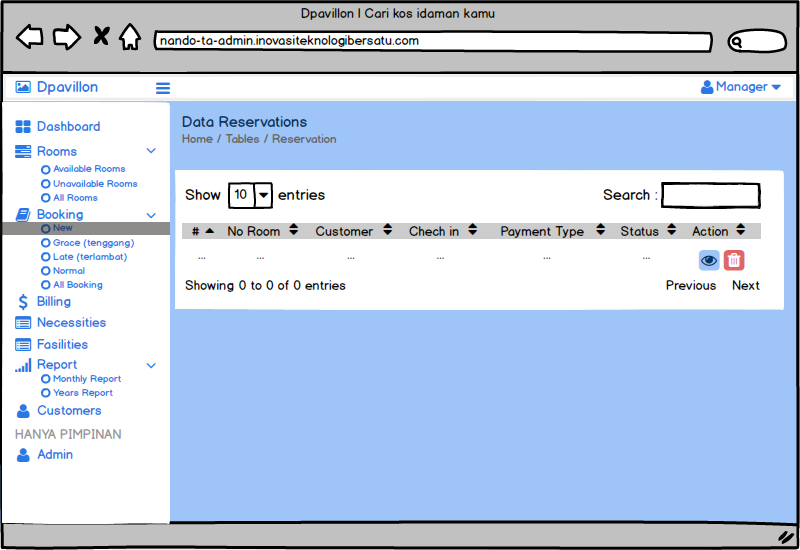
Gambar 4. 51Unavailable Rooms Manager

1. Rancangan Halaman *All Rooms*



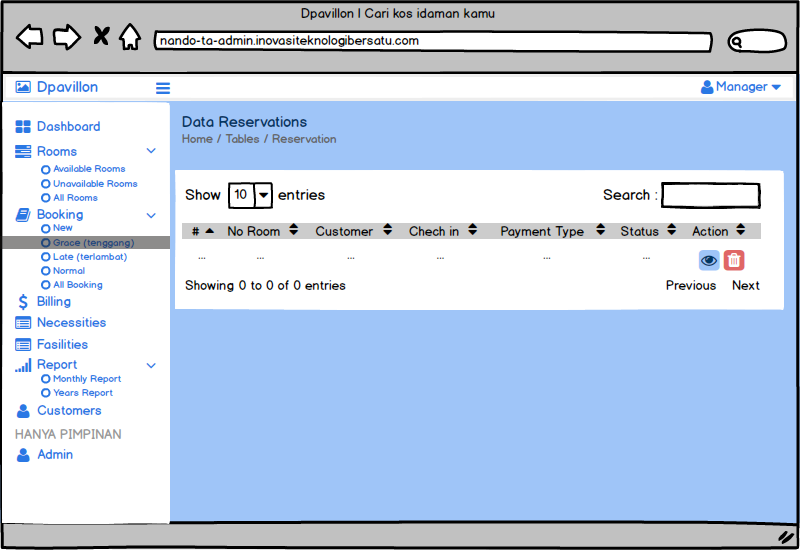
Gambar 4. 52 All Rooms Manager

1. **Rancangan Halaman *Booking***
2. Rancangan Halaman *New*



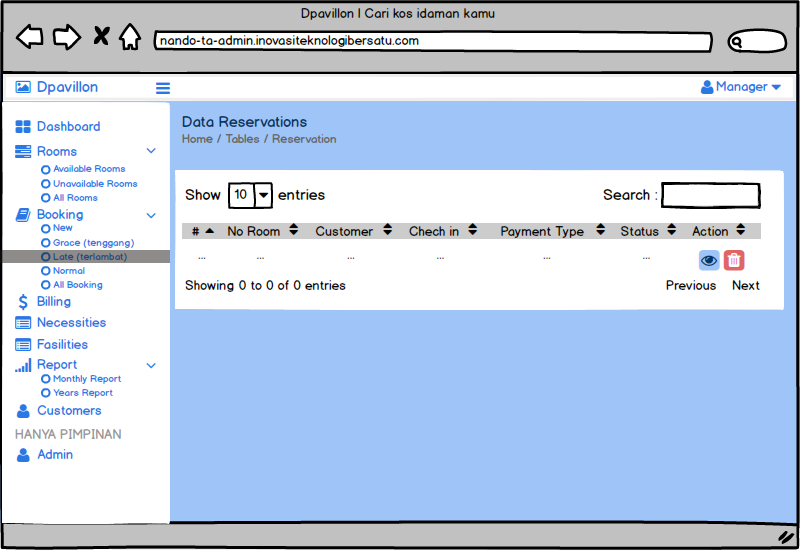
Gambar 4. 53 New Manager

1. Rancangan Halaman *Grace*



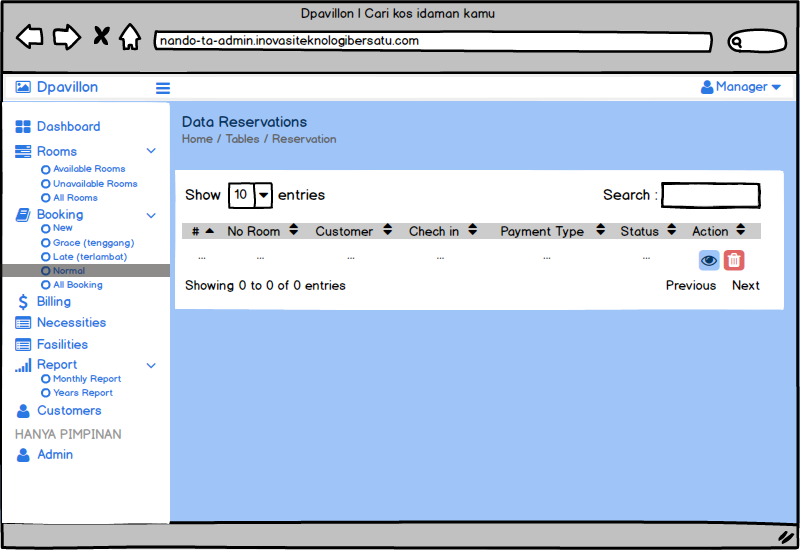
Gambar 4. 54 Grace Manager

1. Rancangan Halaman *Late*



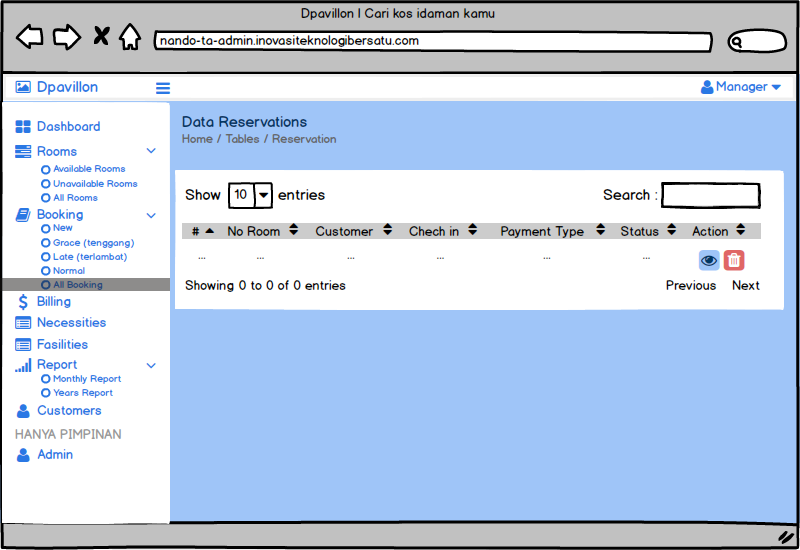
Gambar 4. 55 Late Manager

1. Rancangan Halaman Normal



Gambar 4. 56 Normal Manager

1. Rancangan Halaman *All Booking*



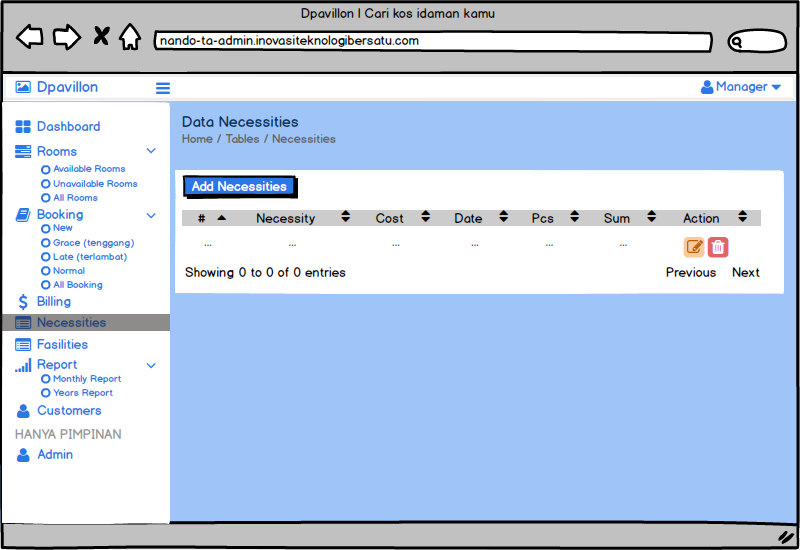
Gambar 4. 57 All Booking Manager

1. **Rancangan Halaman *Billing***



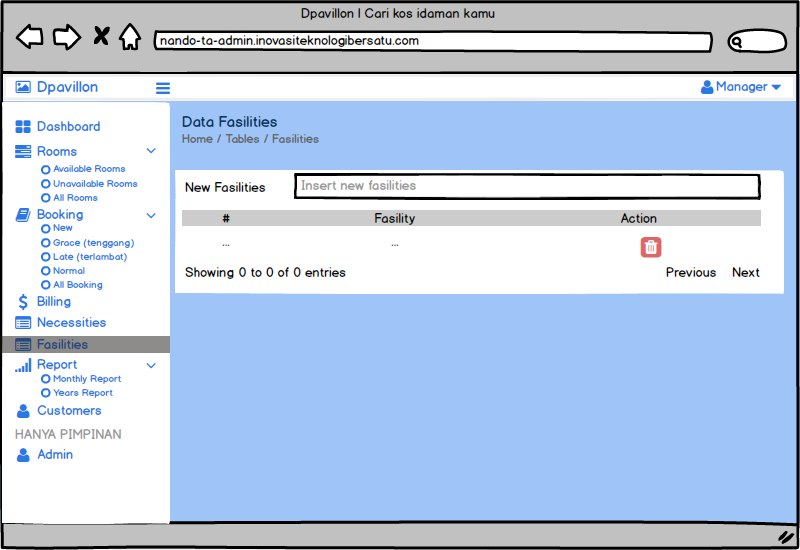
Gambar 4. 58 Billing Manager

1. **Rancangan Halaman *Necessities***



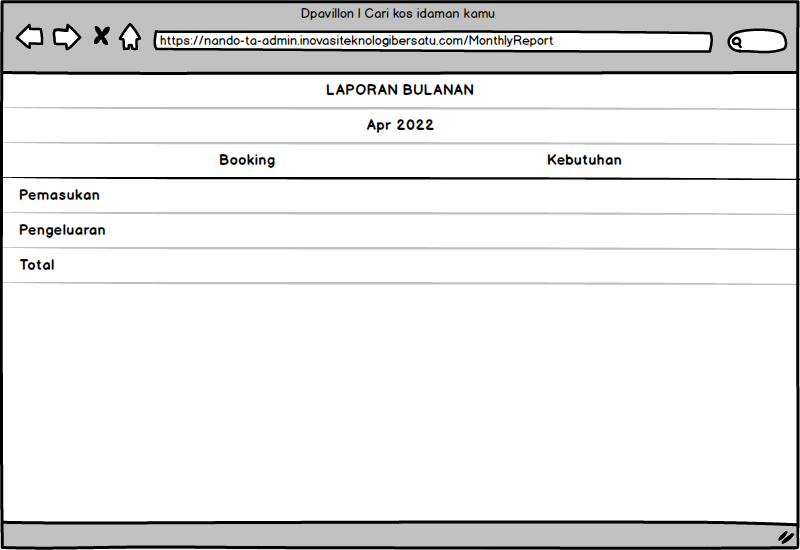
Gambar 4. 59 Necessities Manager

1. **Rancangan Halaman *Fasilities***



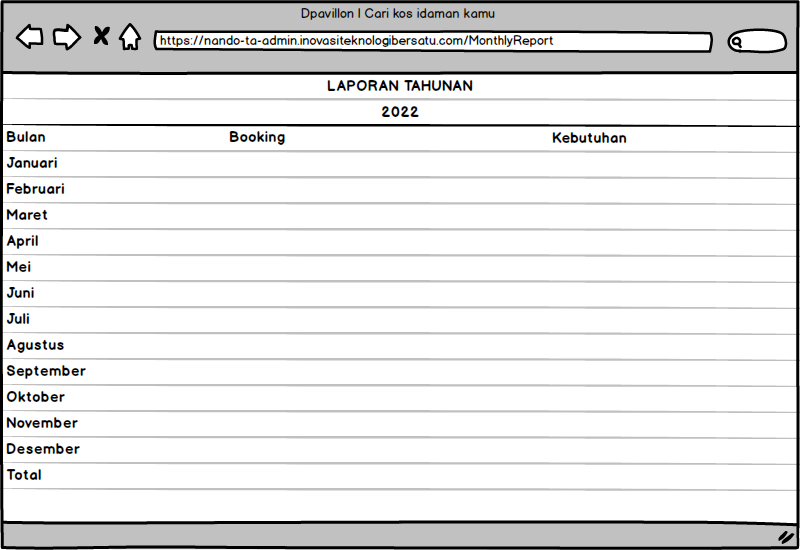
Gambar 4. 60 Fasilities Manager

1. **Rancangan Halaman *Report***
2. Rancangan Halaman *Monthly Report*



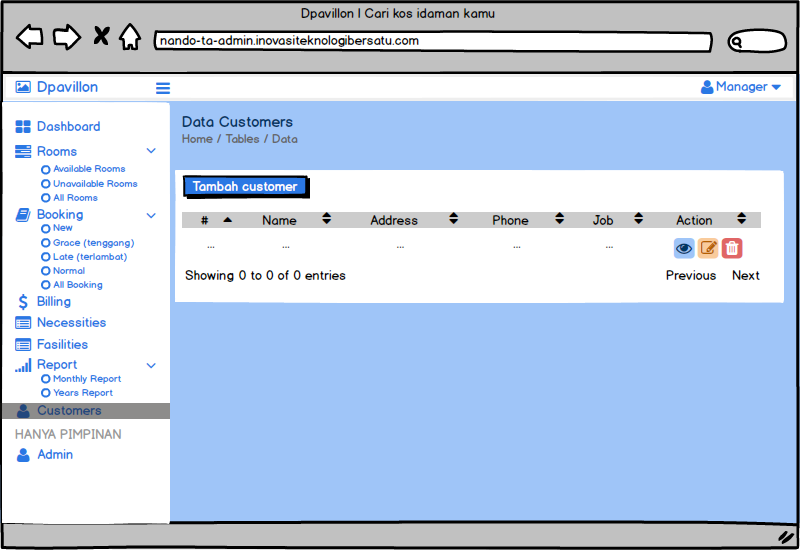
Gambar 4. 61 Monthly Report Manager

1. Rancangan Halaman *Years Report*



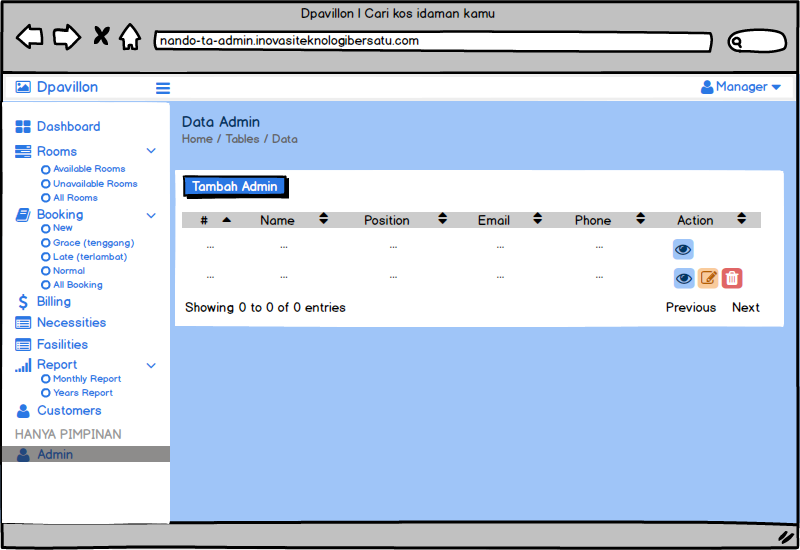
Gambar 4. 62 Years Report Manager

1. **Rancangan Halaman *Customers***



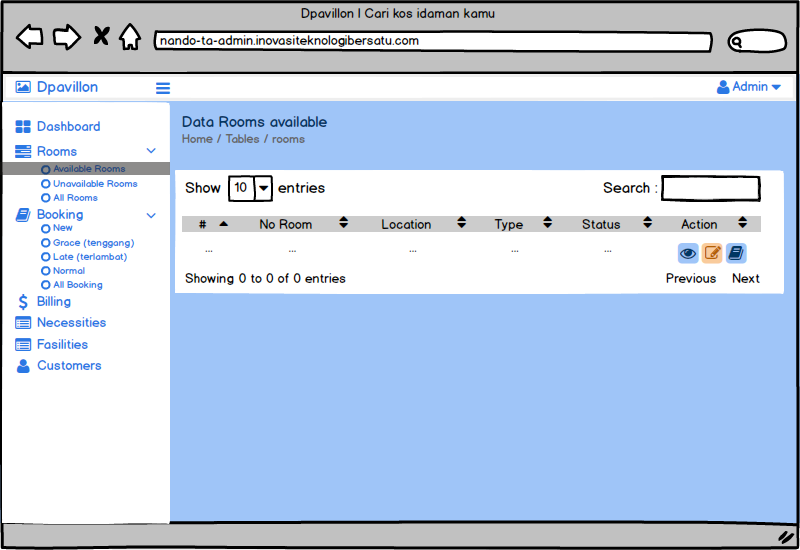
Gambar 4. 63 Customers Manager

1. **Rancangan Halaman Admin**



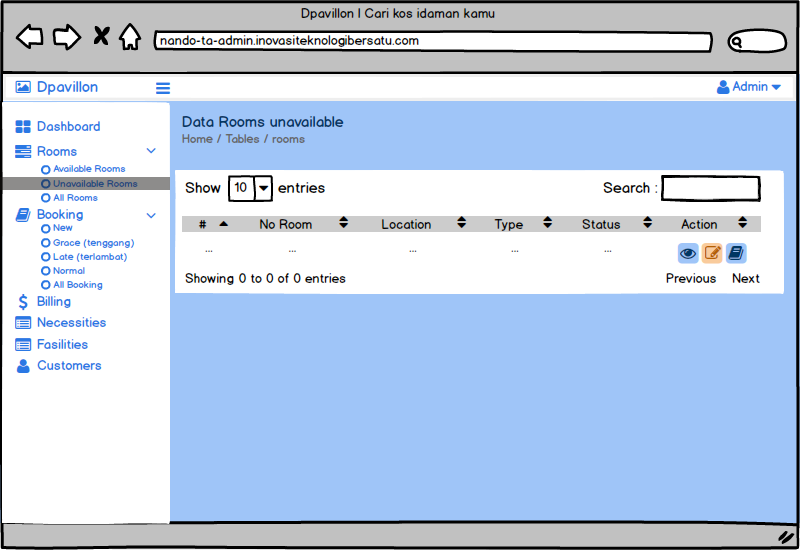
Gambar 4. 64 Admin Manager

1. **Rancangan Halaman Admin**
2. **Rancangan Halaman *Rooms***
3. Rancangan Halaman *Available Rooms*



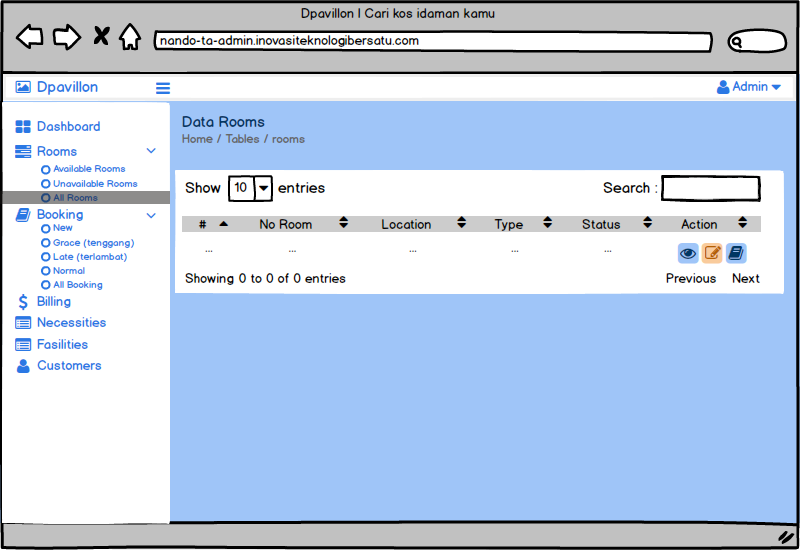
Gambar 4. 65 Available Rooms Admin

1. Rancangan Halaman *Unavailable Rooms*



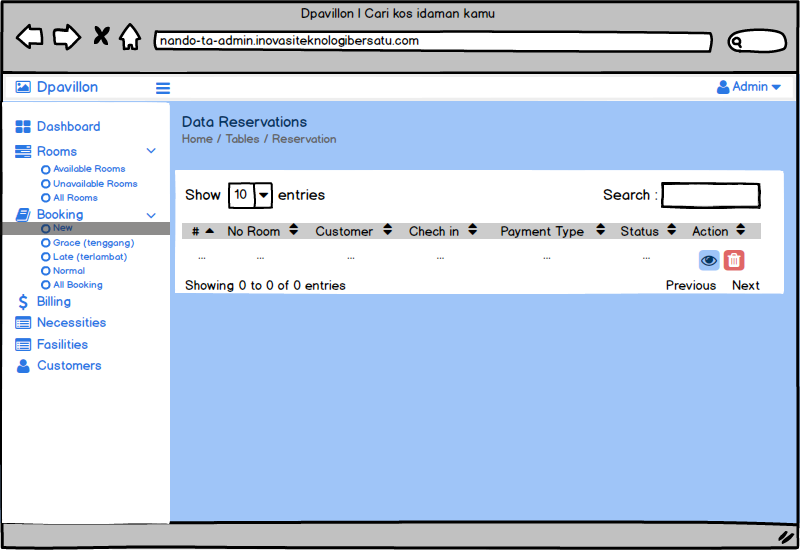
Gambar 4. 66 Unavailable Rooms Admin

1. Rancangan Halaman *All Rooms*



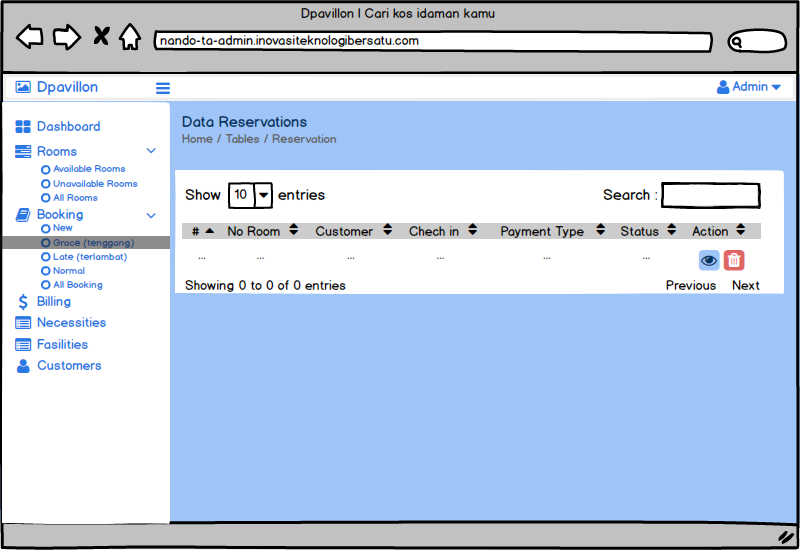
Gambar 4. 67 All Rooms Admin

1. **Rancangan Halaman *Booking***
2. Rancangan Halaman *New*



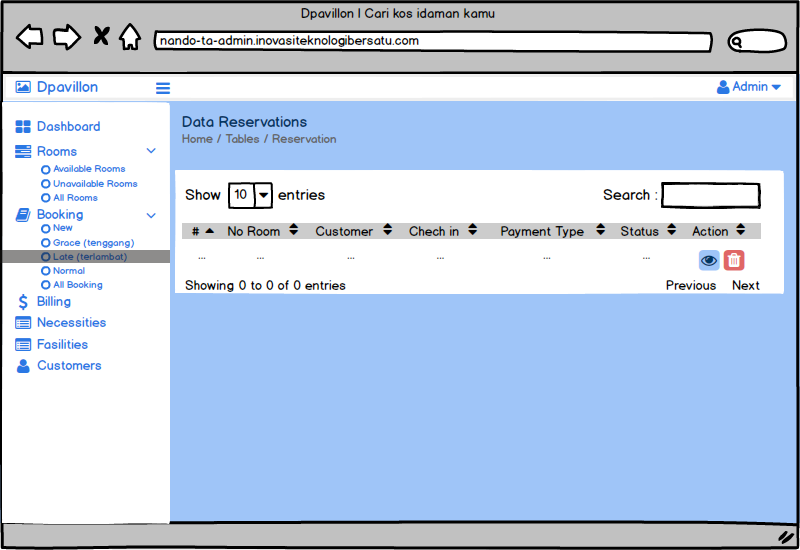
Gambar 4. 68 New Admin

1. Rancangan Halaman *Grace*



Gambar 4. 69 Grace Admin

1. Rancangan Halaman *Late*



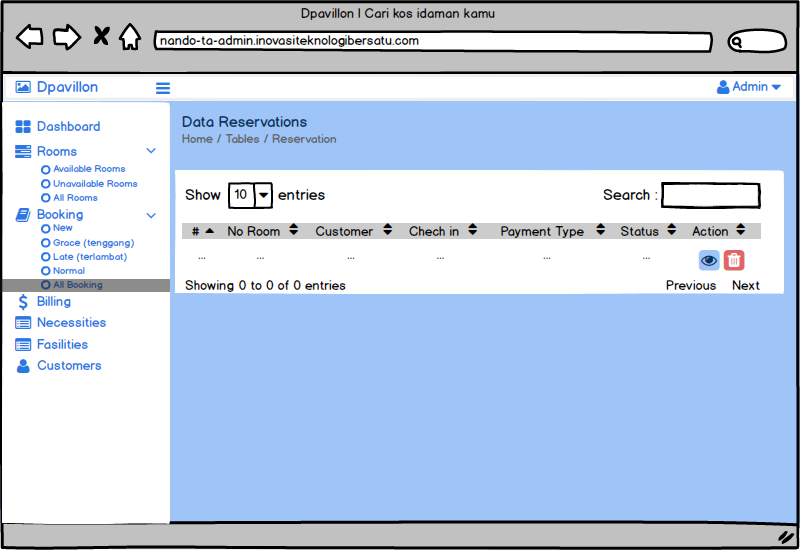
Gambar 4. 70 Late Admin

1. Rancangan Halaman Normal



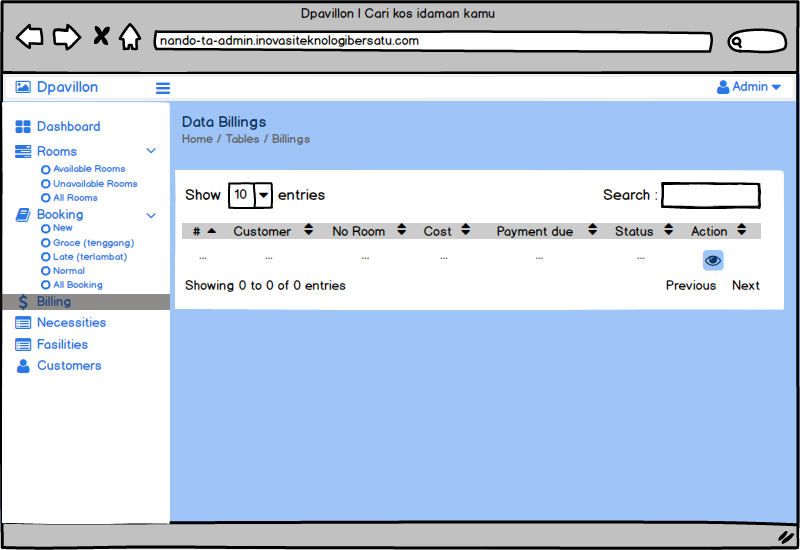
Gambar 4. 71 Normal Admin

1. Rancangan Halaman *All Booking*



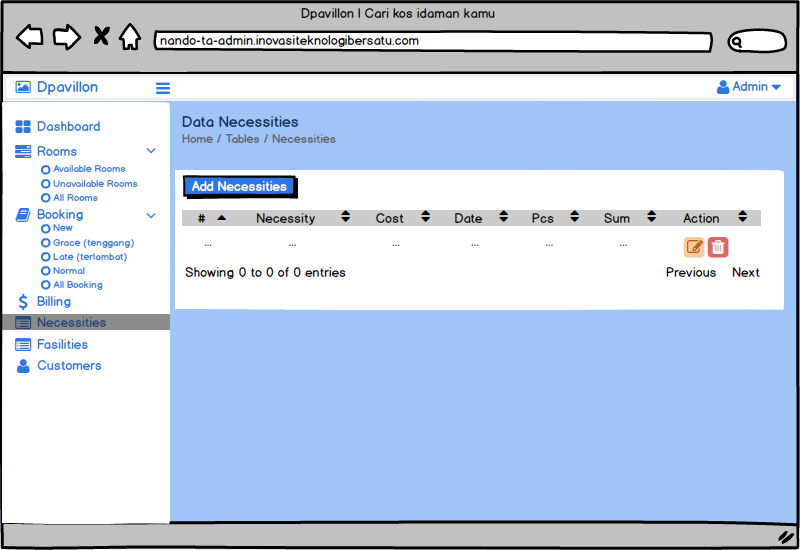
Gambar 4. 72 All Booking Admin

1. **Rancangan Halaman *Billing***



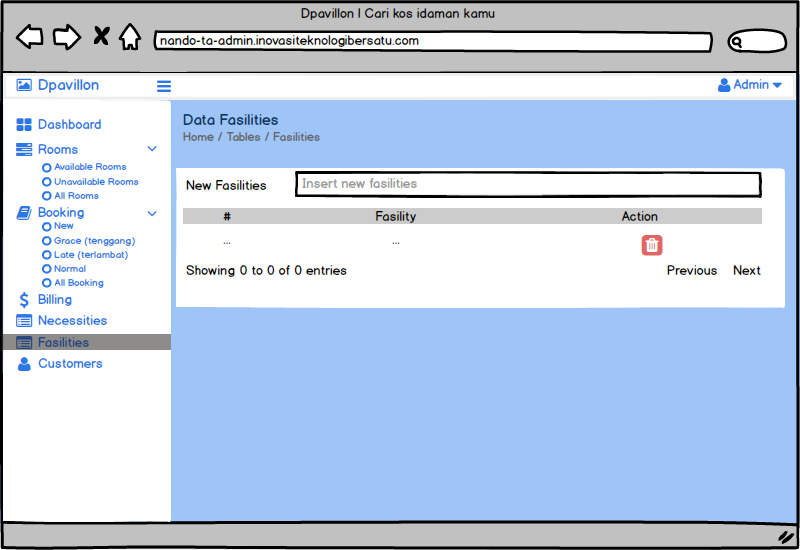
Gambar 4. 73 Billing Admin

1. **Rancangan Halaman *Necessities***



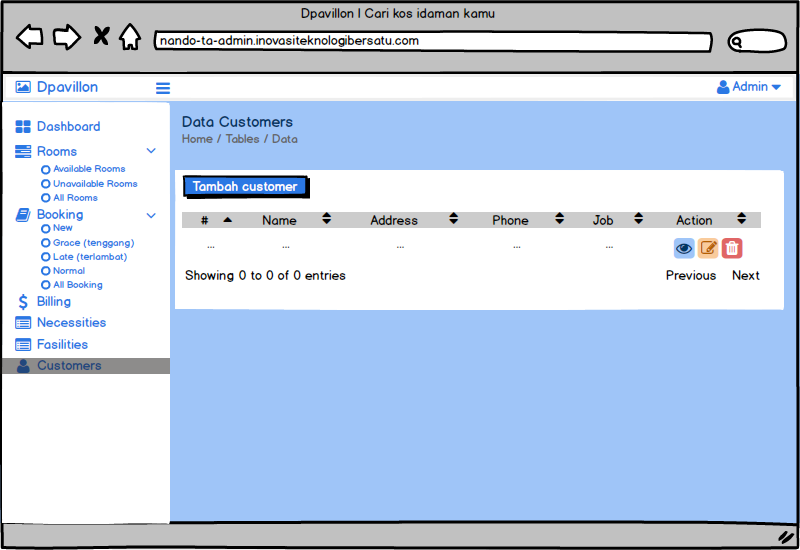
Gambar 4. 74 Necessities Admin

1. **Rancangan Halaman *Fasilities***



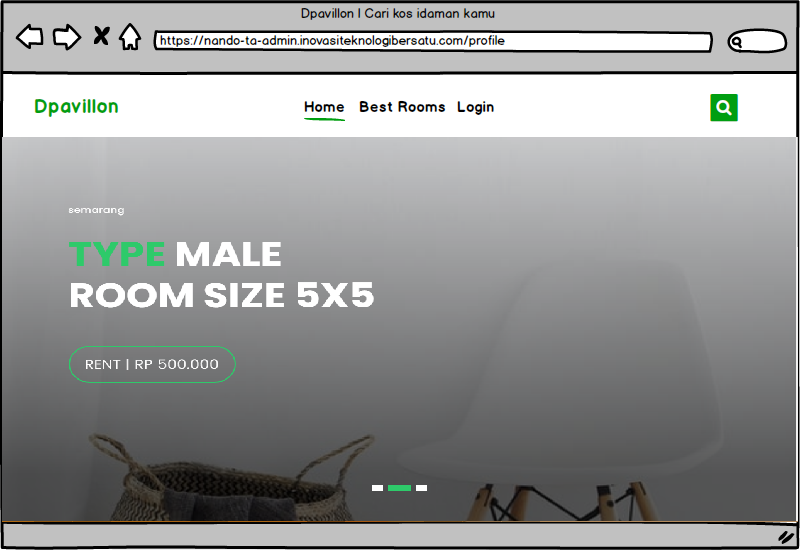
Gambar 4. 75 Fasilities Admin

1. **Rancangan Halaman *Customers***



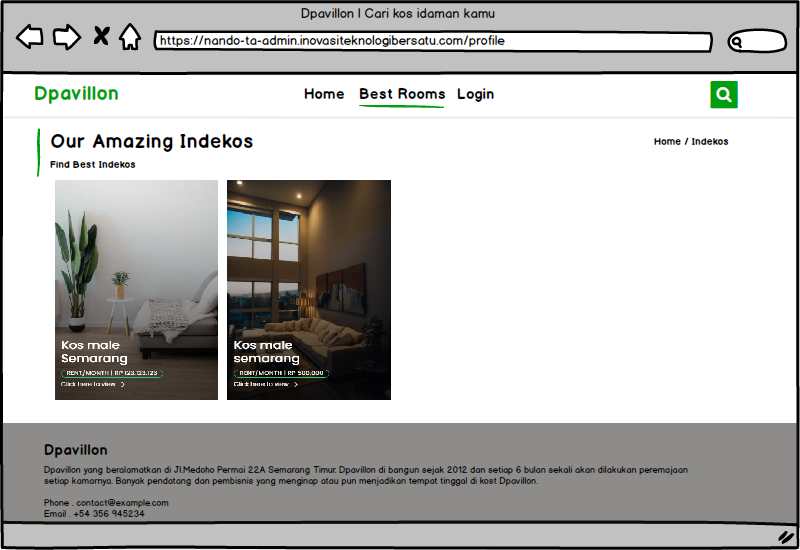
Gambar 4. 76 Customers Admin

1. **Rancangan Halaman *Customer***
2. **Rancangan Halaman *Sebelum Log in***
3. Rancangan Halaman *Home*



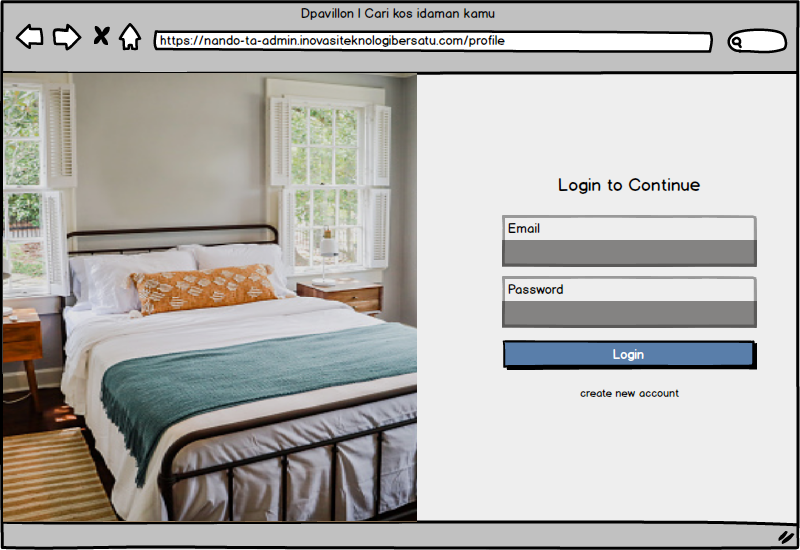
Gambar 4. 77 Home Customer

1. Rancangan Halaman *Best Rooms*



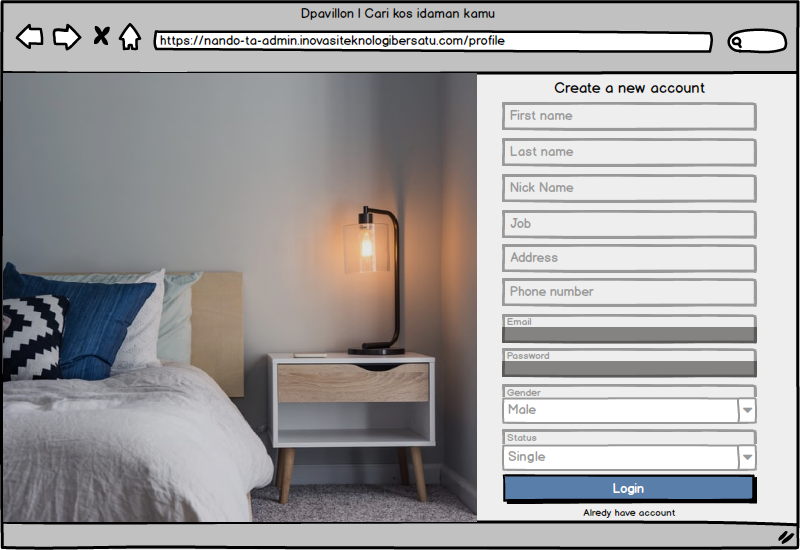
Gambar 4. 78 Best Rooms Customer

1. **Rancangan Halaman *Log in***



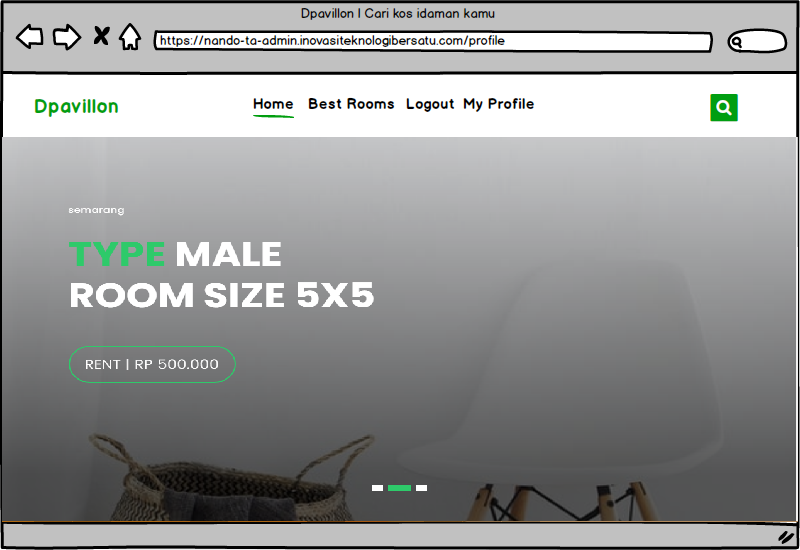
Gambar 4. 79 Log In Customer

1. **Rancangan Halaman *Create Account***



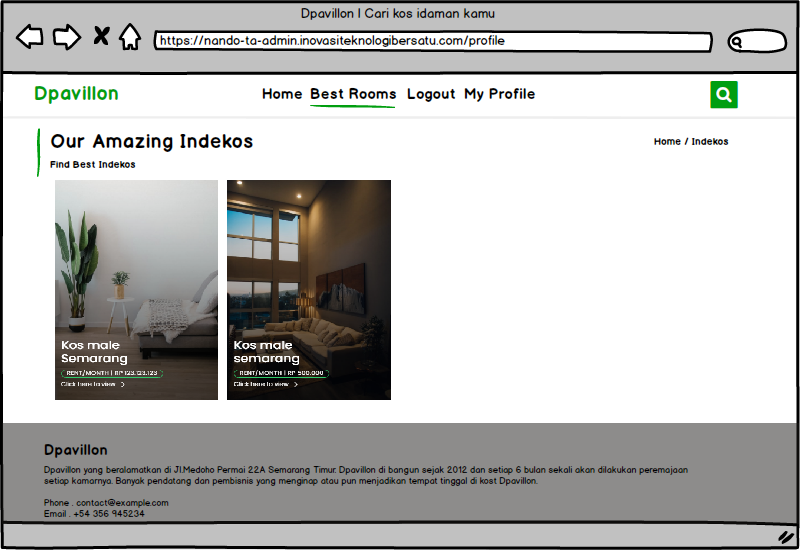
Gambar 4. 80 Create Account Customer

1. **Rancangan Halaman *Sesudah Log in***
2. Rancangan Halaman *Home*



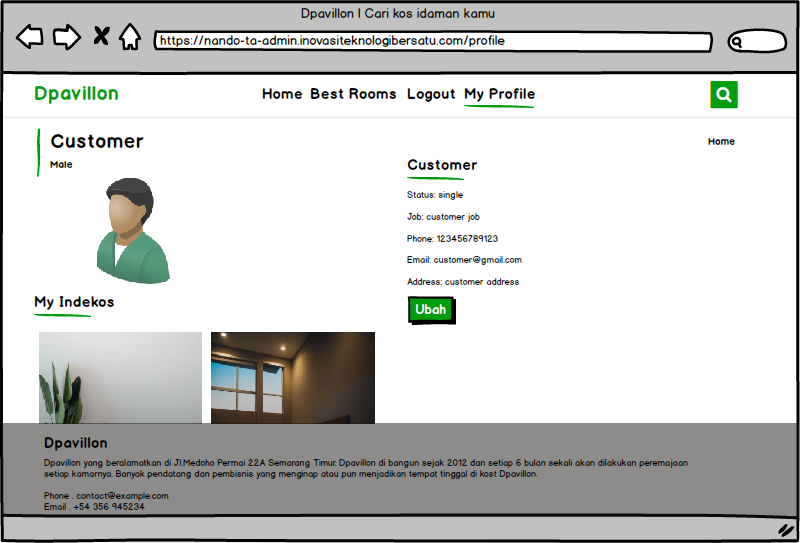
Gambar 4. 81 Home Customer

1. Rancangan Halaman *Best Rooms*



Gambar 4. 82 Best Rooms Customer

1. Rancangan Halaman *My Profile*



Gambar 4. 83 My Profile Customer