­­­­­

**DPPL - 01**

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

Xpoint

untuk:

Customer yang mencari tempat makan atau tempat *hangout*

Dipersiapkan oleh:

Anak Agung Sada Abhyakta (1301183440)  
Muhammad Fadhil Mubaraq (1301184031)  
Philip Wanderrienov (1301180170)  
Irfanul Arifa (1301183418)

Program Studi Informatika

Fakultas Informatika

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Prodi S1- Informatika**  **Universitas Telkom** | Nomor Dokumen | | Halaman |
| *DPPL-xx <xx:no grp>* | | *<#>/<jml #* |
| Revisi | *<nomor revisi>* | *Tgl: <isi tanggal>* |

DAFTAR PERUBAHAN

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A |  |
| B |  |
| C |  |
| D |  |
| E |  |
| F |  |
| G |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX  TGL | - | A | B | C | D | E | F | G |
| Ditulis oleh |  |  |  |  |  |  |  |  |
| Diperiksa oleh |  |  |  |  |  |  |  |  |
| Disetujui oleh |  |  |  |  |  |  |  |  |

Daftar Halaman Perubahan

|  |  |  |  |
| --- | --- | --- | --- |
| Halaman | Revisi | Halaman | Revisi |
|  |  |  |  |

Daftar Isi

[Daftar Tabel 6](#_Toc39835784)

[Daftar Gambar 7](#_Toc39835785)

[1. Pendahuluan 8](#_Toc39835786)

[1.1 Tujuan Penulisan Dokumen 8](#_Toc39835787)

[1.2 Lingkup Masalah 8](#_Toc39835788)

[1.3 Definisi dan Istilah 8](#_Toc39835789)

[1.4 Referensi 8](#_Toc39835790)

[1.5 Sistematika Pembahasan 8](#_Toc39835791)

[2 Deskripsi Perancangan Global 9](#_Toc39835792)

[2.1 Deskripsi Arsitektural 9](#_Toc39835793)

[2.2 Deskripsi Komponen 10](#_Toc39835794)

[3 Perancangan Rinci 11](#_Toc39835795)

[3.1 Realisasi Use Case 11](#_Toc39835796)

[3.1.1 Use Case #1 <Registrasi> 11](#_Toc39835797)

[3.1.1.1 Perancangan Antarmuka Usecase #1 12](#_Toc39835798)

[3.1.1.2 Identifikasi Object Baru 12](#_Toc39835799)

[3.1.1.3 Diagram Kelas 13](#_Toc39835800)

[3.1.1.4 Sequence Diagram 13](#_Toc39835801)

[3.1.2 Use Case #2 <Login> 13](#_Toc39835802)

[3.1.2.1 Perancangan Antarmuka Usecase #2 14](#_Toc39835803)

[3.1.2.2 Identifikasi Object Baru 14](#_Toc39835804)

[3.1.2.3 Diagram Kelas 15](#_Toc39835805)

[3.1.2.4 Sequence Diagram 15](#_Toc39835806)

[3.1.3 Use Case #3 <Searching> 15](#_Toc39835807)

[3.1.3.1 Perancangan Antarmuka Usecase #3 16](#_Toc39835808)

[3.1.3.2 Tabel Identifikasi Antarmuka / Layar / Page 16](#_Toc39835809)

[3.1.3.3 Identifikasi Object Baru 17](#_Toc39835810)

[3.1.3.4 Diagram Kelas 17](#_Toc39835811)

[3.1.3.5 Sequence Diagram 18](#_Toc39835812)

[3.1.4 Use Case #4 <Booking> 18](#_Toc39835813)

[3.1.4.1 Perancangan Antarmuka Usecase #4 19](#_Toc39835814)

[3.1.4.2 Identifikasi Object Baru 19](#_Toc39835815)

[3.1.4.3 Diagram Kelas 20](#_Toc39835816)

[3.1.4.4 Sequence Diagram 20](#_Toc39835817)

[3.1.5 Use Case #5 <Rating> 20](#_Toc39835818)

[3.1.5.1 Perancangan Antarmuka Usecase #5 21](#_Toc39835819)

[3.1.5.2 Identifikasi Object Baru 21](#_Toc39835820)

[3.1.5.3 Diagram Kelas 22](#_Toc39835821)

[3.1.5.4 Sequence Diagram 22](#_Toc39835822)

[3.1.6 Use Case #6 <Hitz> 22](#_Toc39835823)

[3.1.6.1 Perancangan Antarmuka Usecase #6 23](#_Toc39835824)

[3.1.6.2 Identifikasi Object Baru 24](#_Toc39835825)

[3.1.6.3 Diagram Kelas 24](#_Toc39835826)

[3.1.6.4 Sequence Diagram 24](#_Toc39835827)

[4 Perancangan Detil 25](#_Toc39835828)

[4.1 Perancangan Detil Kelas 25](#_Toc39835829)

[4.2 Perancangan Kelas Persistensi (\*\*\*Basis Data Skema Tabel) 26](#_Toc39835830)

[4.3 Perancangan Algoritma 26](#_Toc39835831)

[4.3.1 Algoritma #1 (Restoran) 26](#_Toc39835832)

[4.3.2 Algoritma #2 (Rating) 27](#_Toc39835833)

[4.3.3 Algoritma #3 (Menu) 27](#_Toc39835834)

[4.3.4 Algoritma #4 (Order) 27](#_Toc39835835)

[4.3.5 Algoritma #5 (Payement) 28](#_Toc39835836)

[4.3.6 Algoritma #6 (Hitz) 28](#_Toc39835837)

[4.3.7 Algoritma #7 (Menu) 29](#_Toc39835838)

[4.4 Perancangan Query 29](#_Toc39835839)

[5 Matriks Kerunutan (Requirement Traceability Matrix) 30](#_Toc39835840)

# Daftar Tabel

[Tabel 1Deskripsi Komponen Diagram 11](#_Toc39859302)

[Tabel 2 Usecase 12](#_Toc39859303)

[Tabel 3 Usecase Scenario Registrasi 12](#_Toc39859304)

[Tabel 4 Objek Registrasi 13](#_Toc39859305)

[Tabel 5 Identifikasi Objek Baru 13](#_Toc39859306)

[Tabel 6 Usecase Scenario Login 15](#_Toc39859307)

[Tabel 7 Objek Login 15](#_Toc39859308)

[Tabel 8 Objek Baru Login 16](#_Toc39859309)

[Tabel 9 Identifikasi Antarmuka Searching 17](#_Toc39859310)

[Tabel 10 Objek Searching 18](#_Toc39859311)

[Tabel 11 Objek Baru Searching 18](#_Toc39859312)

[Tabel 12 Usecase Scenario Booking 19](#_Toc39859313)

[Tabel 13 Objek Booking 21](#_Toc39859314)

[Tabel 14 Identifikasi Objek Baru 21](#_Toc39859315)

[Tabel 15 Diagram Kelas Booking 21](#_Toc39859316)

[Tabel 16 Sequence Diagram Booking 22](#_Toc39859317)

[Tabel 17 Usecase Scenario Rating 22](#_Toc39859318)

[Tabel 18 Objek Rating 24](#_Toc39859319)

[Tabel 19 Identifikasi Objek Baru Rating 24](#_Toc39859320)

[Tabel 20 Objek Hitz 26](#_Toc39859321)

[Tabel 21 Identifikasi Objek Baru Hitz 26](#_Toc39859322)

[Tabel 22 Perancangan Detil Kelas 28](#_Toc39859323)

[Tabel 23 Perancangan Query 32](#_Toc39859324)

[Tabel 24 Matriks Kerunutan 32](#_Toc39859325)

# Daftar Gambar

[Gambar 1 Komponen Diagram 10](#_Toc39859339)

[Gambar 2 Deployment Diagram 10](#_Toc39859340)

[Gambar 3 Antarmuka Registrasi 13](#_Toc39859341)

[Gambar 4 Diagram Kelas Registrasi 14](#_Toc39859342)

[Gambar 5 Sequence Diagram Registrasi 14](#_Toc39859343)

[Gambar 6 Antarmuka Login 15](#_Toc39859344)

[Gambar 7 Diagram Kelas Login 16](#_Toc39859345)

[Gambar 8 Sequence Diagram Login 16](#_Toc39859346)

[Gambar 9 Antarmuka Searching 17](#_Toc39859347)

[Gambar 10 Diagram Kelas Searching 18](#_Toc39859348)

[Gambar 11 Sequence Diaram Searching 19](#_Toc39859349)

[Gambar 12 Antarmuka Booking 20](#_Toc39859350)

[Gambar 13 Antarmuka Booking(Invoice) 20](#_Toc39859351)

[Gambar 14 Beri Rating 23](#_Toc39859352)

[Gambar 15 Masukkan Rating 23](#_Toc39859353)

[Gambar 16 Diagram Kelas Rating 24](#_Toc39859354)

[Gambar 17 Sequence Diagram Rating 24](#_Toc39859355)

[Gambar 18 Antarmuka Hitz 25](#_Toc39859356)

[Gambar 19 Diagram Kelas Hitxz 26](#_Toc39859357)

[Gambar 20 Sequence Diagram Hitz 27](#_Toc39859358)

[Gambar 21 Perancangan Kelas Persistensi 28](file:///C:\Users\User\Downloads\DDPL%20-%20Xpoint(2).docx#_Toc39859359)

# Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) merupakan sebuh dokumen yang mendeskripsikan perancangan perangkat lunak yang akan dikembangkan dan bertujuan untuk menjadi sebuah pedoman dalam proses pengkodean aplikasi booking restoran “Xpoint”. Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan teknis pengembangan perangkat lunak pada tahap selanjutnya. Dengan disusunnya dokumen DPPL ini, diharapkan pembangunan perangkat lunak akan lebih terkonsep dan tidak menimbulkan ambiguitas pada saat pembangunannya.

## Lingkup Masalah

Xpoint merupakan sebuah aplikasi yang berfungsi untuk memberikan informasi tentang tempat makan atau tempat *hangout* kepada *customer*. Aplikasi ini dapat digunakan oleh seluruh kalangan masyarakat terlebih pada kalangan muda. Selain berfungsi untuk memberikan informasi tentang tempat makan atau tempat *hangout*, Xpoint juga dapat digunakan untuk *booking* tempat makan atau tempat *hangout* serta dapat membeli produk pada tempat tersebut. Aplikasi ini diharapkan dapat mempermudah masyarakat dalam mencari atau ­*booking­* tempat makan atau tempat *hangout*.

## Definisi dan Istilah

DPPL : Dokumen Perancangan Perangkat Lunak

Xpoint : Nama aplikasi yang akan dibuat

Hangout : Berkumpul

Customer : Pelanggan

## Referensi

1. Template Dokumen Deskripsi Perancangan Perangkat Lunak
2. Spesifikasi Kebutuhan Perangkat Lunak Xpoint

## Sistematika Pembahasan

Dalam dokumen DPPL ini menggunakan jenis font *Cambria* dengan ukuran 12 kecuali untuk penulisan judul menggunakan ukuran 16 dan sub judul menggunakan ukuran 14, dan *line spacing 1,5*. Penggunaan Bahasa asing menggunakan tulisan cetak miring / *italic* dan penggunaan huruf tebal untuk judul dan sub judul di daftar isi.

Dalam bab pertama dijelaskan tentang permasalahan yang terjadi dengan orang-orang yang terkadang bingung untuk mencari tempat makan atau tempat *hangout*. Dan dijelaskan juga sedikit tentang fungsi Xpoint bagi para pengguna.

Dalam bab dua dijelaskan tentang deskripsi perancangan global dari system perangkat lunak yang akan kami buat yaitu Xpoint. Dalam bab ini pun dijelaskan keterangan untuk setiap komponen yang nantinya akan ada dalam Xpoint.

Dalam bab ketiga dijelaskan tentang perancangan rinci perangkat lunak Xpoint ini. Dijelaskan juga tentang use case yang aka nada dalam aplikasi Xpoint ini. Dalam bab ini pun akan terdapat perancangan antarmuka dari tiap-tiap use case nya.

Dalam bab keempat dijelaskan tentang perancangan detil dari aplikasi Xpoint ini. Dalam bab ini akan diberikan perancangan detil dari tiap-tiap kelas yang akan digunakan untuk membuat aplikasi ini. Lalu akan dijelaskan juga tentang perancangan algoritma yang akan digunakan untuk membuat aplikasi ini. Selain algoritma, dalam bab ini pun akan dijelaskan tentang penjelasan perancangan query.

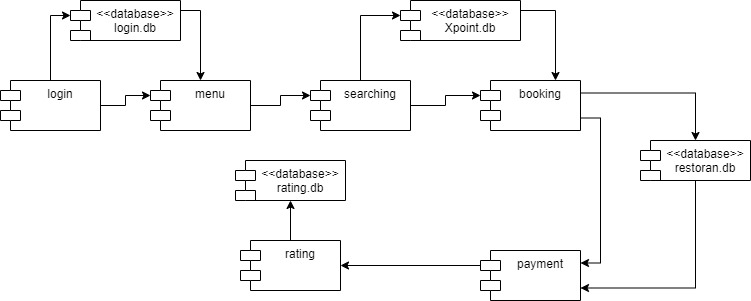
Dalam bab kelima dijelaskan tentang matriks kerunutan pada aplikasi ini dengan menjelaskan proses penggunaan aplikasi ini.

# Deskripsi Perancangan Global

Pada bagian ini akan dijelaskan deskripsi perancangan secara global dari system perangkat lunak yang akan dibuat.

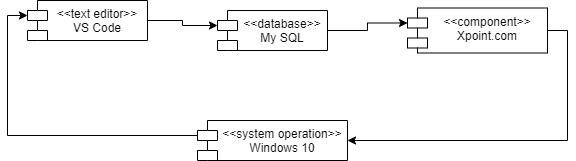
## Deskripsi Arsitektural

1. Komponen Diagram



Gambar Komponen Diagram

1. Deployment Diagram



Gambar Deployment Diagram

## Deskripsi Komponen

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Komponen** | **Keterangan** |
|  | Login | Modul ini digunakan oleh admin dan *customer* untuk dapat mengakses aplikasi. |
|  | Menu | Modul ini adalah sebuah *interface* yang menampilkan beberapa list dan foto-foto tempat makan berdasarkan relevansi. |
|  | Pencarian (*searching*) | Modul ini berfungsi untuk mencari tempat makan nya. Di modul ini pun terdapat filter yang berfungsi untuk mencari tempat makan sesuai dengan apa yang diinginkan oleh *customer*. |
|  | Pemesanan (*booking*) | Modul ini digunakan oleh *customer* untuk melakukan pemesanan tempat makan sebelum *customer* tersebut sampai ke tempat makan atau restoran tersebut. |
|  | Pembayaran (*payment*) | Modul ini digunakan oleh *customer* untuk melakukan transaksi |
|  | Kelola Akun | Modul ini dapat digunakan oleh *customer* untuk mengelola akunnya, seperti mengubah email, alamat, nomor telpon, *password* dan foto profil. |
|  | Memberi Ulasan (*rating*) | Modul ini dapat digunakan oleh pengguna untuk memberi ulasan terhadap tempat makan atau tempat *hangout* yang telah dikunjungi untuk menjadi referensi kepada *customer* lainnya. |
|  | Kelola menu tempat makan | Modul ini dapat digunakan oleh admin untuk mengelola tempat makan, seperti mengganti nama, meng-*update* menu produk, menghapus menu produk dan lain sebagainya. |

Tabel Deskripsi Komponen Diagram

# Perancangan Rinci

## Realisasi Use Case

Berisi TABEL USE CASE sebagai berikut :

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Use Case** | **Deskripsi Use Case** |
| #1 | Registrasi | User (Admin dan *Customer*) melakukan registrasi dengan meng-*input* email, password, alamat, nomor telpon, foto dan lain sebagainya |
| #2 | Login | User (Admin dan *Customer*) melakukan *login* untuk dapat mengakses aplikasi Xpoint ini |
| #3 | Searching | Customer dapat mencari tempat makan atau tempat *hangout* |
| #4 | Booking | Customer melakukan *booking*/pemesanan tempat duduk |
| #5 | Rating | Customer dapat memberikan *rating*/ulasan terhadap tempat makan atau tempat *hangout* yang telah mereka kunjungi |
| #6 | Hitz | Menampilkan tempat makan atau tempat *hangout* yang sedang hitz |

Tabel Usecase

### Use Case #1 <Registrasi>

*Skenario Use Case #1 :*

Nama *Usecase* : Buat Akun

Aktor : User (Admin dan Customer)

*Pre-Condition* : User belum mempunyai akun

Post-Condition : User sudah mempunyai akun

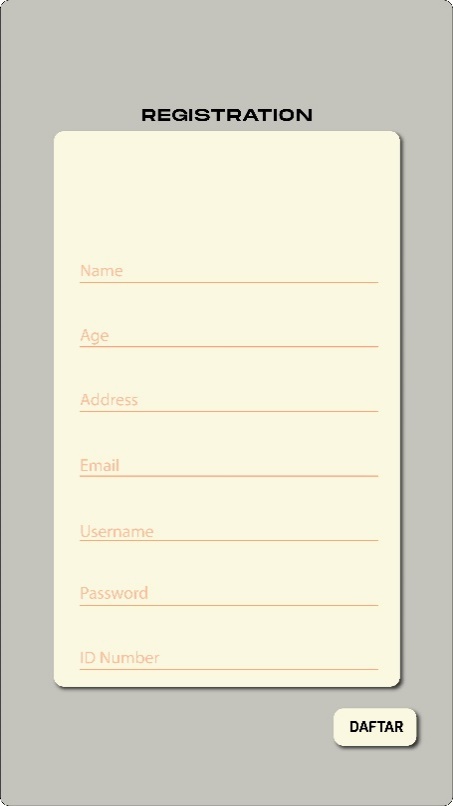
Deskripsi : Menjelaskan proses user yang awalnya belum mempunyai akun sampai telah mempunyai akun

|  |  |
| --- | --- |
| Aktor | Sistem |
| 1. User membuka app Xpoint |  |
| 1. User memilih menu registrasi |  |
| 1. User meng-inputkan data |  |
|  | 1. Sistem menerima data dari pelanggan. |
|  | 1. Jika data sesuai kriteria maka sistem membuat akun baru dan mengupdate database lalu menampilkan laman login. Jika tidak kembali ke langkah 3. |

Tabel Usecase Scenario Registrasi

* + - 1. Perancangan Antarmuka Usecase #1

*Antarmuka Registrasi*



Gambar Antarmuka Registrasi

*Page Registrasi*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *Reg\_Button1* | Button | DAFTAR | Jika diklik,sistem akan memverifikasi data yang diinput lalu membuat akun baru didatabase |
| *Reg\_text1* | *Text* | *Name* | Menerima Inputan User |
| *Reg\_text2* | *Text* | *Age* | Menerima Inputan User |
| *Reg\_text3* | *Text* | *Email* | Menerima Inputan User |
| *Reg\_text4* | *Text* | *Address* | Menerima Inputan User |
| *Reg\_text5* | *Text* | *Username* | Menerima Inputan User |
| *Reg\_text6* | *Text* | *Password* | Menerima Inputan User |
| *Reg\_text7* | *Text* | *ID Number* | Menerima Inputan User |

Tabel Objek Registrasi

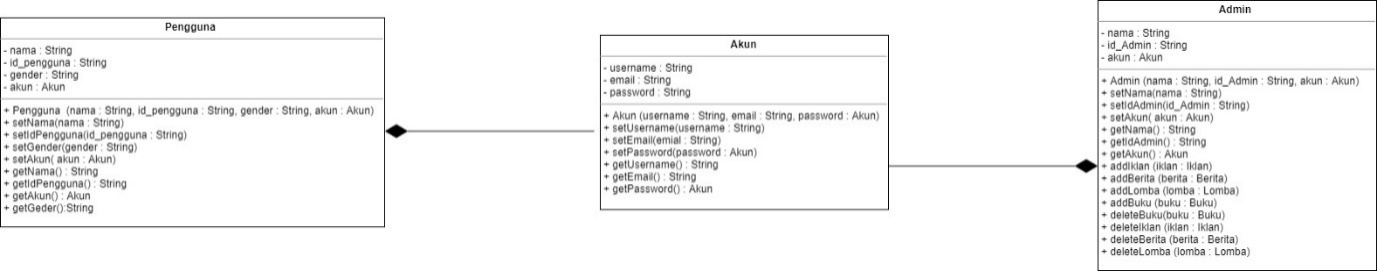
3.1.1.2 Identifikasi Object Baru

*TABEL OBJECT PERANCANGAN*

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| 1 | Customer | Entity |
| 2 | Admin | Entity |
| 3 | Akun | Entity |

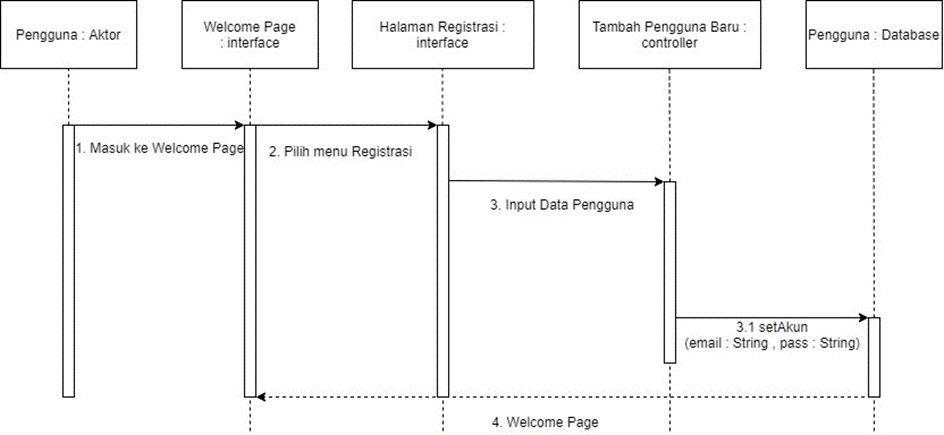
Tabel Identifikasi Objek Baru

3.1.1.3 Diagram Kelas



Gambar Diagram Kelas Registrasi

3.1.1.4 Sequence Diagram



Gambar Sequence Diagram Registrasi

### Use Case #2 <Login>

*Skenario Use Case #2*

Nama *Usecase* : Login

Aktor : User (Admin dan Customer)

*Pre-Condition* : input data user

Post-Condition : Konfirmasi berhasil atau tidak

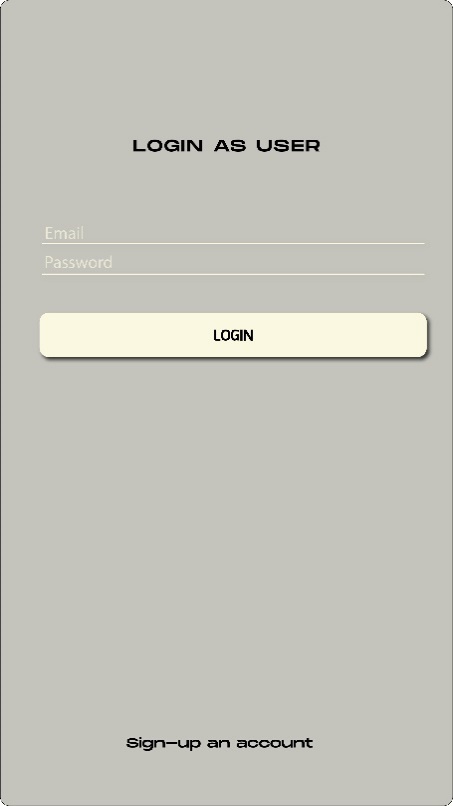
Deskripsi : Untuk masuk ke halaman utama.

|  |  |
| --- | --- |
| Aktor | Sistem |
| 1. Membuka laman login. |  |
|  | 1. Menampilkan form login. |
| 3. Menginputkan data. |  |
| 4. Menekan tombol “login” |  |
|  | 1. Memverifikasi data inputan pengguna. |
|  | 1. Jika data sesuai menampilkan halaman utama, jika data tidak sesuai maka akan dikembalikan ke langkah ke 3. |

Tabel Usecase Scenario Login

* + - 1. Perancangan Antarmuka Usecase #2

*Antarmuka Laman Login*



Gambar Antarmuka Login

*Page Login*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *Login\_Button1* | Button | LOGIN | Jika diklik, akan mengaktifkan Function login. |
| *Login\_text1* | *Text* | *EMAIL* | Menerima inputan User |
| *Login\_text2* | *Text* | *PASSWORD* | Menerima inputan User |
| *Signup\_button* | *Button* | *SIGN-UP AN ACCOUNT* | Jika diklik,akan mengarahkan ke halaman registrasi |

Tabel Objek Login

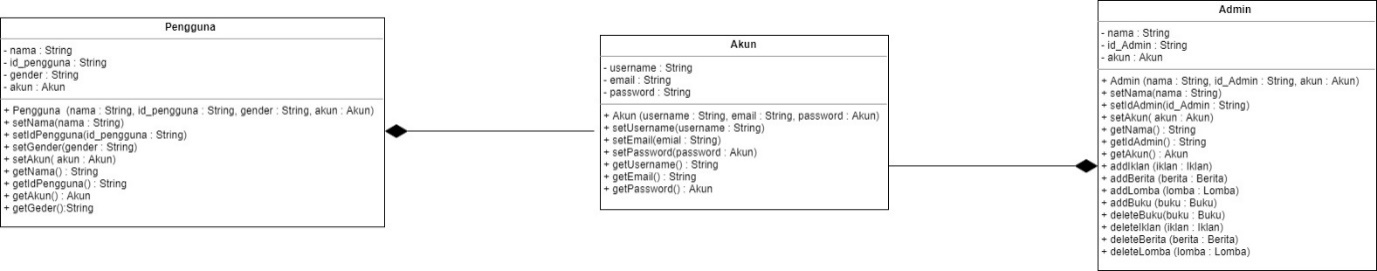
3.1.2.2 Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| 1 | Customer | Entity |
| 2 | Admin | Entity |
| 3 | Akun | Entity |

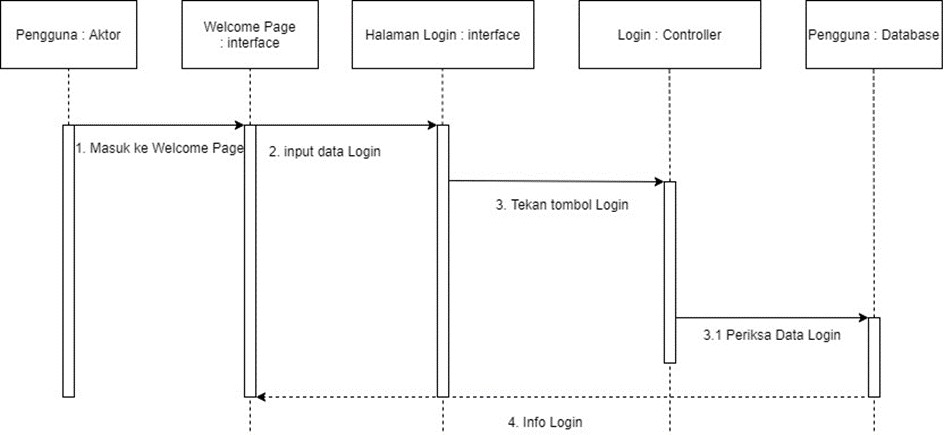
Tabel Objek Baru Login

3.1.2.3 Diagram Kelas



Gambar Diagram Kelas Login

3.1.2.4 Sequence Diagram



Gambar Sequence Diagram Login

### 3.1.3 Use Case #3 <Searching>

*Skenario Use Case #3 :*

Nama *Usecase* : Searching

Aktor : Customer

­*Pre-Condition* : Customer belum memilihtempat makan yang diinginkan

*Post-Condition* : Customer telah memilih tempat makan yang diinginkan

Deskripsi : Menjelaskan proses customer dalam memilih tempat makan atau tempat *hangout*

3.1.3.1 Perancangan Antarmuka Usecase #3

**

Gambar Antarmuka Searching

3.1.3.2 Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| 1 | *Searching Page* | Halaman untuk melakukan pencarian tempat makan atau tempat *hangout* |

Tabel Identifikasi Antarmuka Searching

*Page SEARCHING*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *Field\_cari* | textField | Cari | Field untuk mencari nama tempat makan atau tempat hangout |
| *Button\_place* | *button* | *MANA COFFEE BANDUNG* | Apabila user menekan tombol ini, maka akan diarahkan ke page tempat makan tersebut |
| *Button\_place* | *Button* | *BIIRU: NOT A SUSHI BAR BANDUNG* | Apabila user menekan tombol ini, maka akan diarahkan ke page tempat makan tersebut |
| *Button\_place* | *Button* | *KUKUMAMA CAFÉ BANDUNG* | Apabila user menekan tombol ini, maka akan diarahkan ke page tempat makan tersebut |
| *Button\_place* | *Button* | *LO KA SI CAFÉ BANDUNG* | Apabila user menekan tombol ini, maka akan diarahkan ke page tempat makan tersebut |
| *Button\_cari* | *Button* | *Password* | Apabila user menekan tombol ini, maka sistem akan mencari tempat makan sesuai dengan inputan user di textField cari |
| *Button\_hits* | *button* | *Id Number* | Apabila user menekan tombol ini, maka akan diarahkan ke hitz page |

Tabel Objek Searching

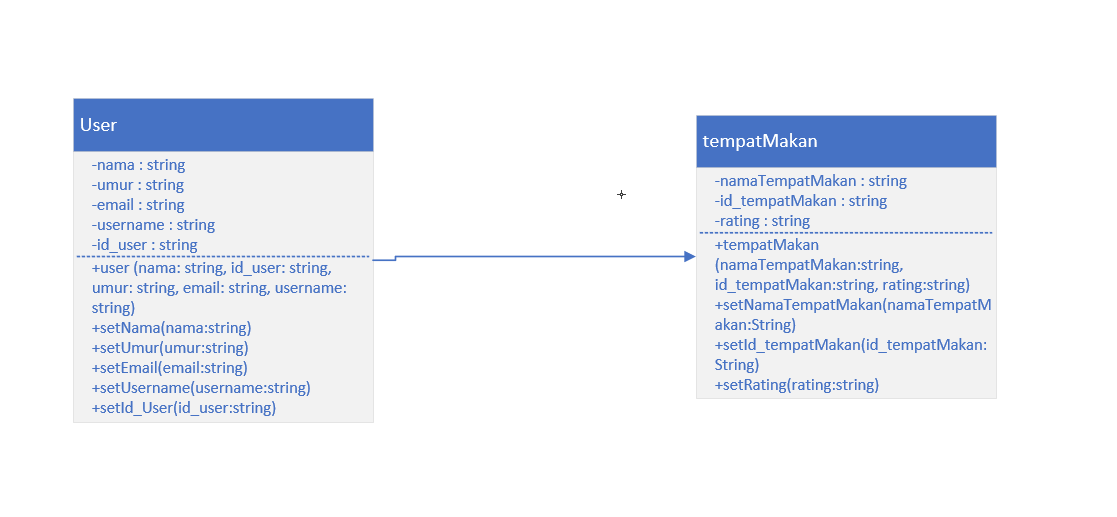
3.1.3.3 Identifikasi Object Baru

*TABEL OBJECT PERANCANGAN*

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| 1 | Sistem | Entity |
| 2 | Customer | Entity |

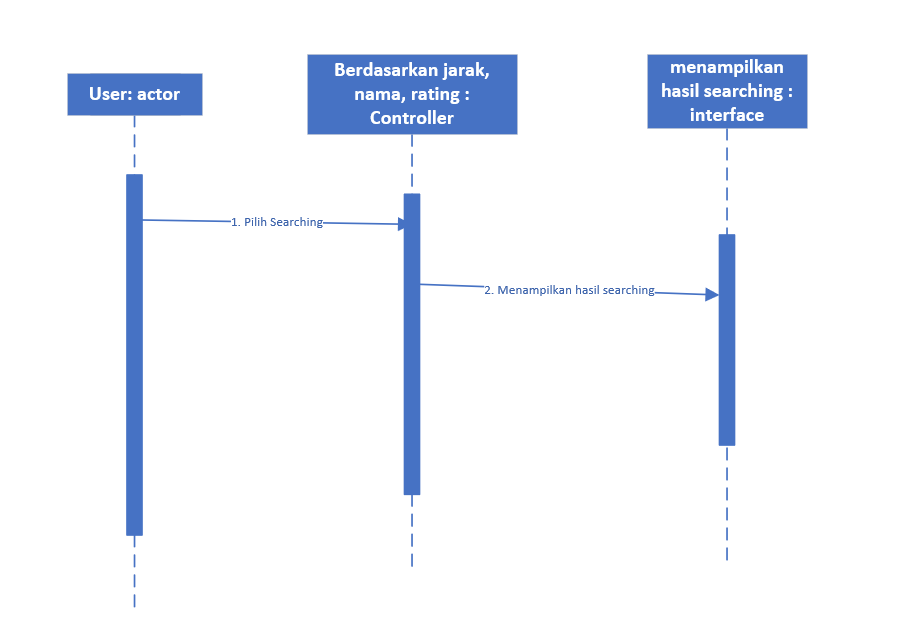
Tabel Objek Baru Searching

* + - 1. Diagram Kelas

**

Gambar Diagram Kelas Searching

* + - 1. Sequence Diagram

**

Gambar Sequence Diaram Searching

### 3.1.4 Use Case #4 <Booking>

*Skenario Use Case #4*

Nama *Usecase* : Booking

Aktor : User

*Pre-Condition* : User sudah mendapatkan restoran yang sesuai

Post-Condition : User memesan reservasi pada restoran

Deskripsi : Untuk memesan reservasi restoran

|  |  |
| --- | --- |
| Aktor | Sistem |
| 1. Memilih restoran yang akan di booking. |  |
|  | 1. Menampilkan laman detail restoran. |
| 3. Menekan tombol “Pesan Reservasi” |  |
|  | 4. Menampilkan invoice pemesanan. |
| 5. User memilih metode pembayaran. |  |

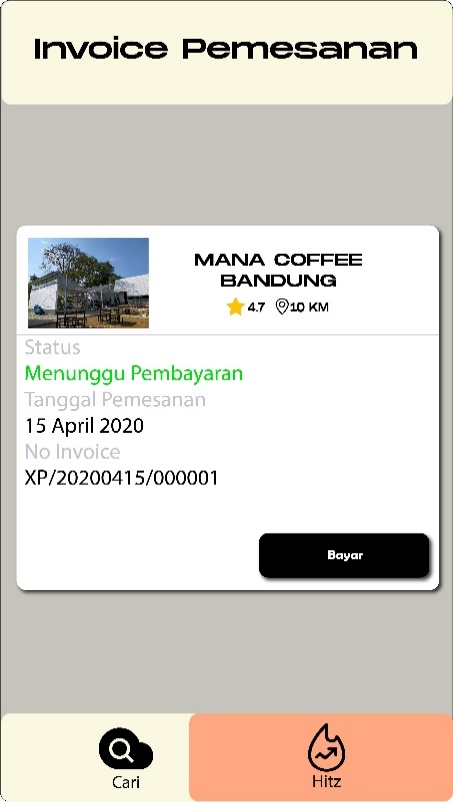
Tabel Usecase Scenario Booking

3.1.4.1 Perancangan Antarmuka Usecase #4

*Antarmuka Laman Booking*

**

Gambar Antarmuka Booking

**

Gambar Antarmuka Booking(Invoice)

*Page Detail Pesanan dan Rating*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *booking\_Button1* | Button | \*namarestoran\* | Jika diklik, akan mengarahkan user ke laman detail restoran. |
| *booking\_text1* | *Text* | *Invoice Pemesanan* | Menampilkan invoice pemesanan. |
| *booking\_button2* | *Button* | *Pesan Reservasi* | Mengarahkan ke laman Invoice Pemesanan. |
| *booking\_button3* | *Button* | *Bayar* | Jika diklik,user diharapkan segera membayar dengan metode pembayaran yang sudah dipilih. |

Tabel Objek Booking

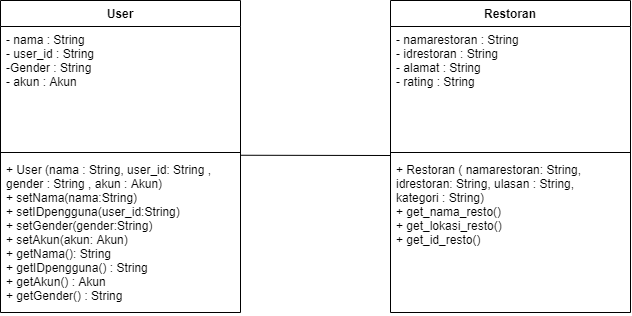
3.1.4.2 Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| 1 | User | Entity |
| 2 | Restoran | Entity |

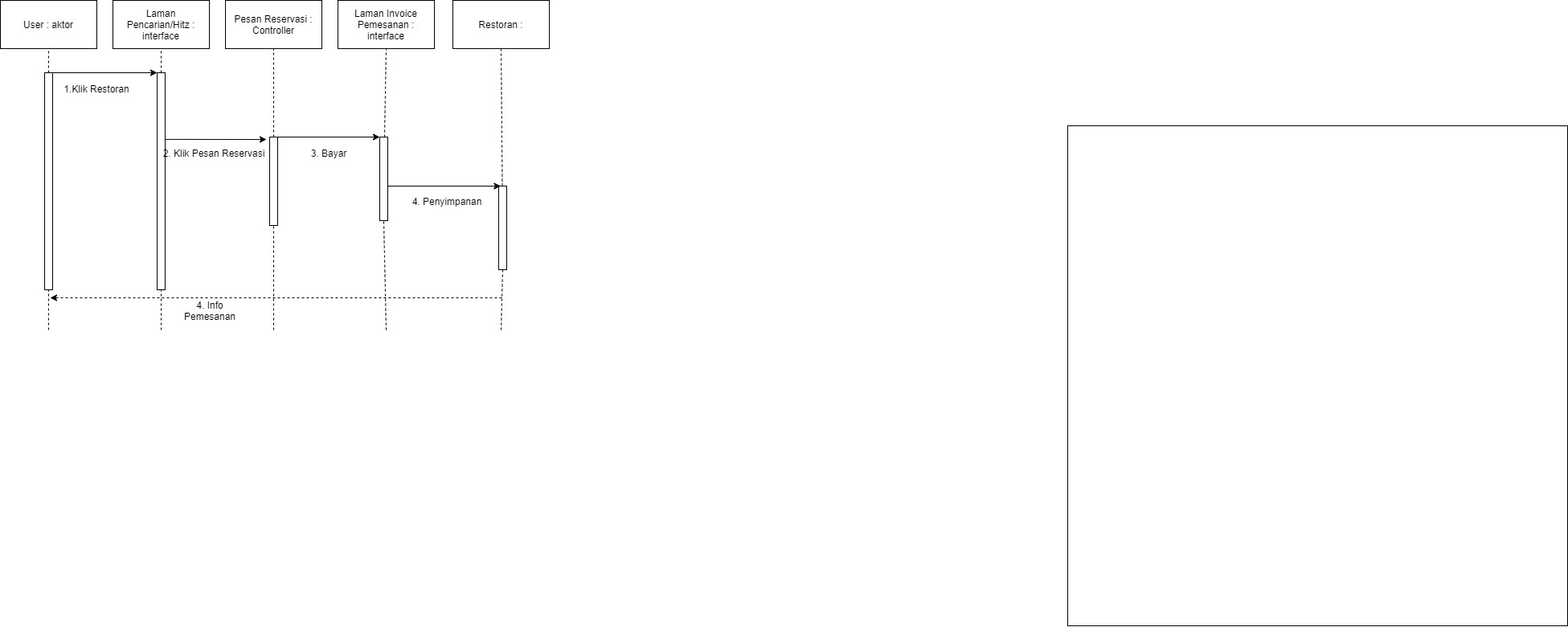
Tabel Identifikasi Objek Baru

3.1.4.3 Diagram Kelas

**

Tabel Diagram Kelas Booking

3.1.4.4 Sequence Diagram



Tabel Sequence Diagram Booking

### 3.1.5 Use Case #5 <Rating>

*Skenario Use Case #5*

Nama *Usecase* : Rating

Aktor : User

*Pre-Condition* : user sudah menyelesaikan pemesanan

Post-Condition : User sudah memberi ulasan

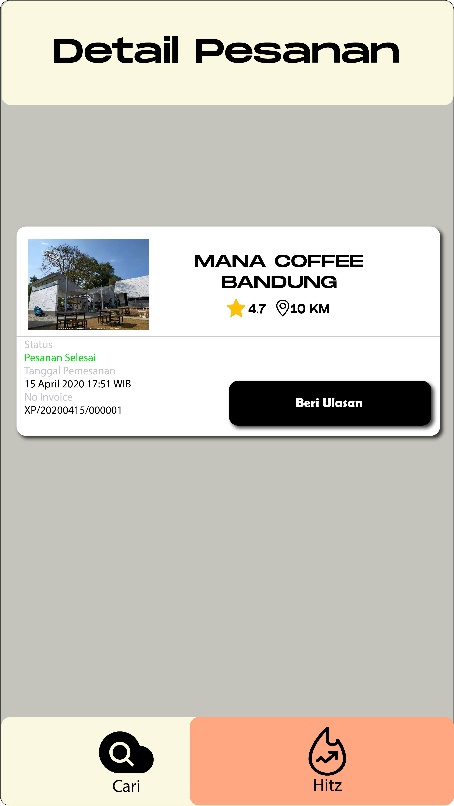
Deskripsi : Untuk memberi ulasan

|  |  |
| --- | --- |
| Aktor | Sistem |
| 1. Menyelesaikan pesanan. |  |
|  | 1. Menampilkan laman detail pesanan |
| 3. Menekan tombol “beri ulasan” |  |
|  | 4. Menampilkan laman beri ulasan |
| 5. User member rating(1-5) dan menulis ulasan dan apabila perlu meng-upload foto. |  |
|  | 1. Ulasan user dikirim ke database dan meng-update rating dari restoran. |

Tabel Usecase Scenario Rating

3.1.5.1 Perancangan Antarmuka Usecase #5

*Antarmuka Laman Rating*



Gambar Beri Rating

**

Gambar Masukkan Rating

*Page Detail Pesanan dan Rating*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *rating\_Button1* | Button | Beri Ulasan | Jika diklik, akan mengarahkan user ke laman beri ulasan. |
| *rating\_text1* | *Text* | *Ulasan Saya* | Menerima inputan User |
| *rating\_button2* | *Button* | *Ikon Bintang* | Menerima inputan User sesuai berapa bintang yang diberikan. |
| *rating\_button3* | *Button* | *Tambah Foto* | Jika diklik,user diminta meng-upload foto. |

Tabel Objek Rating

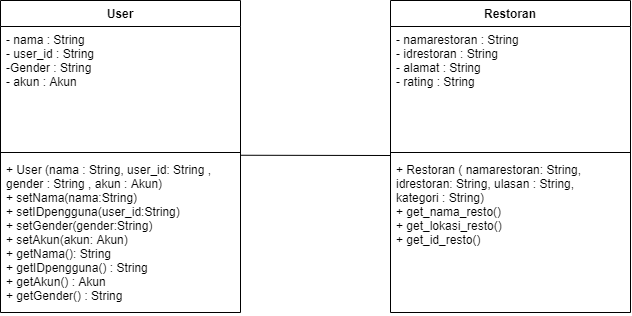
3.1.5.2 Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| 1 | User | Entity |
| 2 | Restoran | Entity |

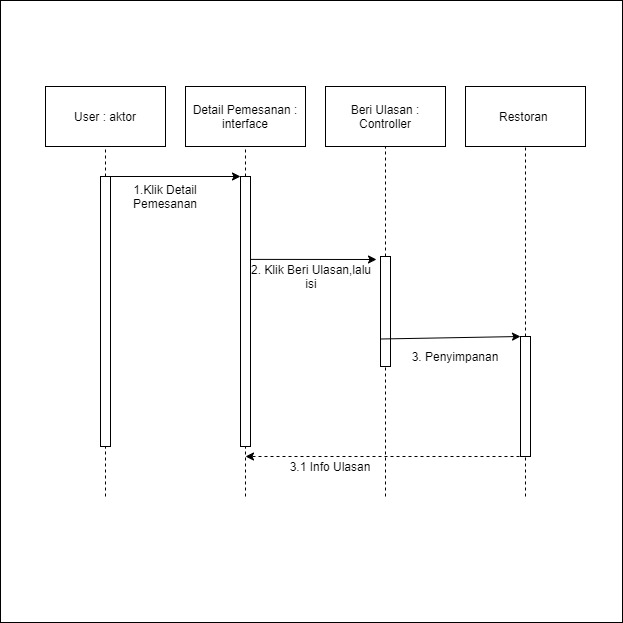
Tabel Identifikasi Objek Baru Rating

3.1.5.3 Diagram Kelas

**

Gambar Diagram Kelas Rating

3.1.5.4 Sequence Diagram



Gambar Sequence Diagram Rating

### 3.1.6 Use Case #6 <Hitz>

*Skenario Use Case #6*

Nama *Usecase* : Hitz

Aktor : Member

*Pre-Condition* : User belum mengetahui tempat makan atau tempta *hangout* yang hitz

Post-Condition : Tampil tempat makan atau tempat *hangout* yang hitz

Deskripsi : Menampilkan nama-nama tempat makan atau tempat *hangout* yang hitz

3.1.6.1 Perancangan Antarmuka Usecase #6

*Antarmuka Laman Hitz*



Gambar Antarmuka Hitz

*Page HITZ*

| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| --- | --- | --- | --- |
| *buttonName1* | Button | MANA COFFE BANDUNG | Apabila user menekan tombol ini, maka akan diarahkan ke page tempat makan tersebut |
| *buttonName2* | *Button* | *BIIRU: NOT A SUSHI BAR BANDUNG* | Apabila user menekan tombol ini, maka akan diarahkan ke page tempat makan tersebut |
| *buttonName3* | *Button* | *KUKUMAMA CAFÉ BANDUNG* | Apabila user menekan tombol ini, maka akan diarahkan ke page tempat makan tersebut |
| *buttonName4* | *Button* | *LO KA SI CAFÉ BANDUNG* | Apabila user menekan tombol ini, maka akan diarahkan ke page tempat makan tersebut |
| *Button\_cari* | *Button* | *Password* | Apabila user menekan tombol ini, maka sistem akan mencari tempat makan sesuai dengan inputan user di textField cari |

Tabel Objek Hitz

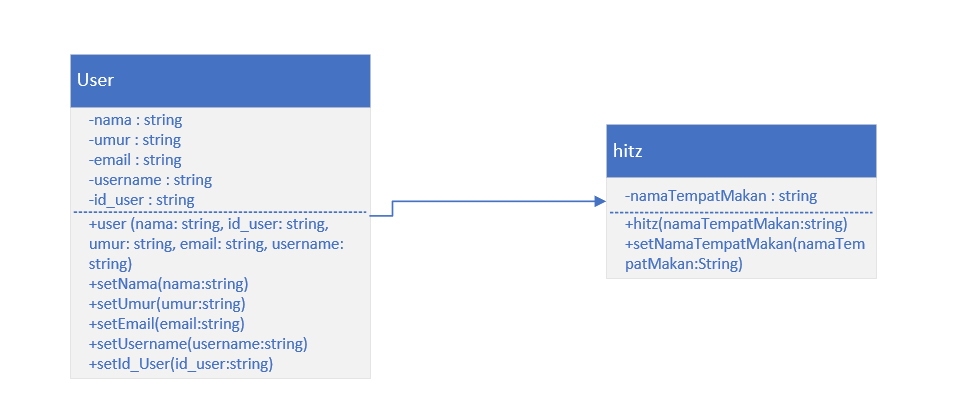
3.1.6.2 Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| 1 | Customer | Entity |
| 2 | Sistem | Entity |

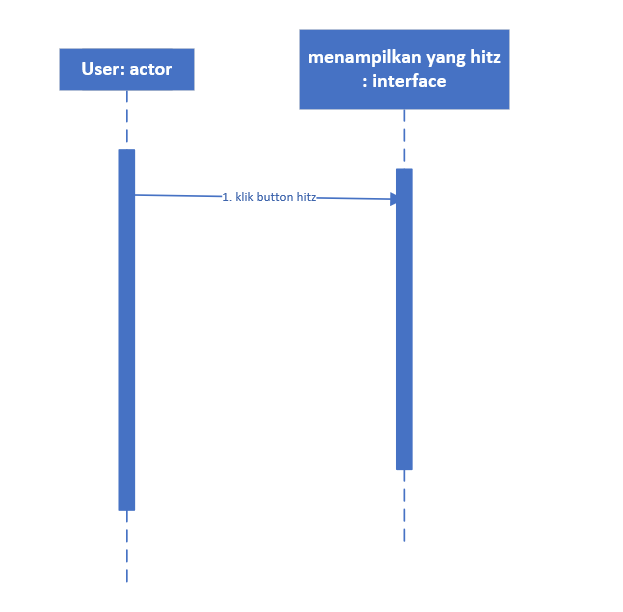
Tabel Identifikasi Objek Baru Hitz

3.1.6.3 Diagram Kelas



Gambar Diagram Kelas Hitxz

3.1.6.4 Sequence Diagram



Gambar Sequence Diagram Hitz

# Perancangan Detil

## Perancangan Detil Kelas

Bagian ini diisi dengan daftar keseluruhan kelas yang akan digunakan dalam PL. menggunakan model MVC

TABEL KELAS :

|  |  |  |  |
| --- | --- | --- | --- |
| **ID Kelas** | **Nama Kelas Perancangan** | **Atribute (visibility)** | **Method / Operation** |
| *1* | *Admin* | 1. *id\_admin(private)*   *nama\_admin(private)* | 1. *getNama\_admin* 2. *getId\_admin* 3. *setNama\_admin,* 4. *addRestoran* 5. *deleteRestoran* 6. *addMenu* 7. *deleteMenu* |
| *2* | *Customer* | 1. *id\_customer(private)* 2. *nama\_customer(private)* 3. *username(private)* 4. *email(private)* 5. *password(private)* 6. *no\_telp(private)* 7. *alamat(private)* | 1. *getId\_customer* 2. *getNama\_customer* 3. *getUsername* 4. *getEmail* 5. *getPassword* 6. *getNo\_telp* 7. *getAlamat* 8. *setNama\_customer* 9. *setUsername* 10. *setEmail* 11. *setPassword* 12. *setNo\_telp* 13. *setAlamat* |
| *3* | *Restoran* | 1. *id\_restoran(private)* 2. *nama\_restoran(private)* 3. *alamat(private)* | 1. *getId\_restoran* 2. *getNama\_restoran* 3. *getAlamat\_restoran* 4. *setNama\_restoran* 5. *setAlamat\_restoran* |
| *4* | *Menu* | 1. *id\_menu(private)* 2. *nama\_menu* 3. *harga(private)* | 1. *getId\_menu* 2. *getNama\_menu* 3. *getHarga* 4. *setNama\_menu* 5. *setHarga* |
| *5* | *Order* | 1. *id\_order(private)* 2. *tanggal\_order* | 1. *getId\_order* 2. *getTanggal\_order* 3. *setTanggal\_order* |

Tabel Perancangan Detil Kelas

## Perancangan Kelas Persistensi (\*\*\*Basis Data Skema Tabel)

Gambar Perancangan Kelas Persistensi

## Perancangan Algoritma

Bagian ini hanya diisi untuk kerangka algoritma untuk proses-proses yang dianggap cukup penting. Implementasi skeleton code juga sudah dapat dilakukan untuk kelas-kelas yang terdefinisi pada bahasa pemrograman tertentu. Boleh dibuat subbab per kelas.

### 4.3.1 Algoritma #1 (Restoran)

*Nama Kelas : Restoran*

*Nama Operasi : Searching Restoran*

*Algoritma :*

|  |
| --- |
| procedure search\_resto(array filter,array resto)  {  for i=1 to length(resto)  {  for y=1 to length(filter)  {  if( filter[y] == resto[i].filter[y])  {  check = true  }else  check = false  }  if (check){  output (resto[i])  }  }  } |

### 4.3.2 Algoritma #2 (Rating)

*Nama Kelas : Rating*

*Nama Operasi : Input Rating*

*Algoritma :*

|  |
| --- |
| procedure input\_rating(string id\_resto\_tmp,string id\_user\_tmp,int rating\_user,string tgl\_now)  {  INSERT INTO rating ( id\_resto,id\_customer,rating,tgl ) VALUES (id\_resto\_tmp,id\_user\_tmp,rating,tgl\_now );  } |

### 4.3.3 Algoritma #3 (Menu)

*Nama Kelas : Menu*

*Nama Operasi : Pilih Menu*

*Algoritma :*

|  |
| --- |
| function pilih\_menu(string id\_customer,string id\_resto,array menu)  {  array keranjang of order  check = false  i = 0;  while (check) {  keranjang[i] = order\_input(menu,id\_resto,selected\_id\_menu,id\_customer,qty,date)  i++  if (done\_selecting\_item)  {  check = true  }  }  } |

### 4.3.4 Algoritma #4 (Order)

*Nama Kelas : Order*

*Nama Operasi : Input Order*

*Algoritma :*

|  |
| --- |
| function order\_input(array menu,string id\_resto,string id\_menu,string id\_customer, int qty, string date)  {  order x  x.id\_resto = id\_resto  x.id\_menu = id\_menu  x.id\_customer = id\_customer  x.quantity = qty  x.tanggal = date  x.harga = get\_menu(menu,id\_menu).harga \* qty  return x  } |

### 4.3.5 Algoritma #5 (Payment)

*Nama Kelas : Payment*

*Nama Operasi : Transaksi Pembayaran*

*Algoritma :*

|  |
| --- |
| function payment(id\_customer,array keranjang,array transaksi)  {  move kerajang to transaksi  if (payment\_onsite)  {  output(total\_price\_on\_cashier)  }  if (payment\_online)  {  output(total\_price)  }  } |

### 4.3.6 Algoritma #6 (Hitz)

*Nama Kelas : Hitz*

*Nama Operasi : Tempat Hitz*

*Algoritma :*

|  |
| --- |
| procedure Hitz(array transaksi)  {  array hitz\_resto of hitz  date\_checking = date\_now - 2\_months  for (date\_checking to date\_now) do  {  while (date\_checking == transaksi.tgl)  {  if (transaksi[date\_checking].id\_resto already in hitz\_resto)  {  while (hitz\_resto[x] != transaksi[date\_checking].id\_resto) dp  {  x++  }  hitz\_resto[x].total\_transaction + 1  }  if (transaksi[date\_checking].id\_resto not in hitz\_resto && near\_from\_location\_customer)  {  hitz\_resto[last] = transaksi[date\_checking].id\_resto  hitz\_resto[last].total\_transaction + 1  }  next transaksi item;  }  date\_checking = next\_date\_from\_date\_checking  }  } |

### 4.3.7 Algoritma #7 (Menu)

*Nama Kelas : Menu*

*Nama Operasi : Mendapatkan Pilihan Menu dari Customer*

*Algoritma :*

|  |
| --- |
| function get\_menu(array menu,string id\_menu)  {  while (menu[i].id\_menu != id\_menu)  {  i++  }  return menu[i]  } |

## Perancangan Query

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| Insert Restoran | *INSERT INTO restoran ( field1, field2,...fieldN ) VALUES ( value1, value2,...valueN );* | *Berfungsi untuk memasukkan data restoran kedalam database* |
| *Insert Menu* | *INSERT INTO menu ( field1, field2,...fieldN ) VALUES ( value1, value2,...valueN );* | *Berfungsi untuk memasukkan menu-menu dari restoran tersebut kedalam database* |
| *Delete Restoran* | *DELETE FROM restoran WHERE id\_resto=x;* | *Berfungsi untuk menghapus seluruh data restoran tersebut dari dalam database* |
| *Delete Menu Restoran* | *DELETE FROM menu WHERE id\_resto=x & id\_menu=y;* | *Berfungsi untuk menghapus menu yang ada di dalam restoran tersebut* |
| *Insert Rating* | *INSERT INTO rating ( id\_resto,Rating ) VALUES ( x,y );* | *Berfungsi untuk menerima hasil rating dari user dan dimasukkan ke dalam database* |

Tabel Perancangan Query

# Matriks Kerunutan (Requirement Traceability Matrix)

|  |  |  |
| --- | --- | --- |
| **Kode FR** | **Nama Functional Requirement** | **Nama Use Case** |
| FR-01 | Login | Login |
| FR-02 | Registrasi | Registrasi |
| FR-03 | Searching | Search |
| FR-04 | Ter-Hitz | Hitz |
| FR-05 | Rating | Rating |
| FR-06 | Booking | Booking |

Tabel Matriks Kerunutan