**DPPL - 01**

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

Xpoint

untuk:

Customer yang mencari tempat makan atau tempat *hangout*

Dipersiapkan oleh:

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

Program Studi Informatika

Fakultas Informatika

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Logo, company name  Description automatically generated | **Prodi S1- Informatika**  **Universitas Telkom** | Nomor Dokumen | | Halaman |
| *DPPL-01 <Xpoint>* | | *1/24* |
| Revisi | *<nomor revisi>* | *Tgl. 18-4-2021* |

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](#_Toc74807444)

[Daftar Gambar 7](#_Toc74807445)

[1 Pendahuluan 8](#_Toc74807446)

[1.1 Tujuan Penulisan Dokumen 8](#_Toc74807447)

[1.2 Lingkup Masalah 8](#_Toc74807448)

[1.3 Definisi dan Istilah 8](#_Toc74807449)

[1.4 Referensi 8](#_Toc74807450)

[1.5 Sistematika Pembahasan 8](#_Toc74807451)

[2 Deskripsi Perancangan Global 9](#_Toc74807452)

[2.1 Deskripsi Arsitektural 9](#_Toc74807453)

[2.2 Deskripsi Komponen 9](#_Toc74807454)

[3 Perancangan Rinci 10](#_Toc74807455)

[*3.1* Realisasi Use Case 10](#_Toc74807456)

[3.1.1 Use Case #1 <Registrasi> 10](#_Toc74807457)

[1. Perancangan Antarmuka Usecase #1 11](#_Toc74807458)

[3.1.1.2 Identifikasi Object Baru 13](#_Toc74807459)

[3.1.1.3 Diagram Kelas 13](#_Toc74807460)

[3.1.1.4 Sequence Diagram 13](#_Toc74807461)

[3.1.2 Use Case #2 <Login> 13](#_Toc74807462)

[3.1.2.1 Perancangan Antarmuka Usecase #2 14](#_Toc74807463)

[3.1.2.2 Identifikasi Object Baru 15](#_Toc74807464)

[3.1.2.3 Diagram Kelas 15](#_Toc74807465)

[3.1.2.4 Sequence Diagram 15](#_Toc74807466)

[3.1.3 Use Case #3 <Booking> 16](#_Toc74807467)

[3.1.3.1 Perancangan Antarmuka Usecase #3 16](#_Toc74807468)

[3.1.3.2 Identifikasi Object Baru 17](#_Toc74807469)

[3.1.3.3 Diagram Kelas 17](#_Toc74807470)

[3.1.3.4 Sequence Diagram 17](#_Toc74807471)

[3.1.4 Use Case #4 <Payment> 18](#_Toc74807472)

[3.1.4.1 Perancangan Antarmuka Usecase #3 18](#_Toc74807473)

[3.1.4.2 Identifikasi Object Baru 19](#_Toc74807474)

[3.1.4.3 Diagram Kelas 20](#_Toc74807475)

[3.1.4.4 Sequence Diagram 20](#_Toc74807476)

[4. Perancangan Detail 20](#_Toc74807477)

[4.1. Perancangan Detail Kelas 20](#_Toc74807478)

[4.2. Perancangan Kelas Persistensi (\*\*\*Basis Data Skema Tabel) 22](#_Toc74807479)

[4.3. Perancangan Algoritma 22](#_Toc74807480)

[4.3.1 Algoritma #1 (Restoran) 22](#_Toc74807481)

[4.3.2 Algoritma #2 (Menu) 23](#_Toc74807482)

[4.3.3 Algoritma #3 (Order) 23](#_Toc74807483)

[4.4. Perancangan Query 23](#_Toc74807484)

[5. Matriks Kerunutan (Requirement Traceability Matrix) 24](#_Toc74807485)

# Daftar Tabel

**No table of figures entries found.**

# Daftar Gambar

**No table of figures entries found.**

# Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) merupakan sebuah 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 sistem 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 detail 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 terakhir 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 sistem perangkat lunak yang akan dibuat.

## Deskripsi Arsitektural

1. Komponen Diagram

Diagram

Description automatically generated

Gambar 1 Komponen Diagram

1. Deployment Diagram

Diagram

Description automatically generated

Gambar 2 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. |
|  | 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 |

Table 1 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, username, password. |
| #2 | Login | User (Admin dan *Customer*) melakukan *login* untuk dapat mengakses aplikasi Xpoint ini |
| #3 | Booking | Customer melakukan *booking*/pemesanan tempat duduk beserta menu tambahan |
| #4 | Payment | Customer melakukan pembayaran terhadap sejumlah uang berdasarkan pesan sebelumnya |

Table 2 Usecase

### Use Case #1 <Registrasi>

*Skenario Use Case #1 :*

Nama *Usecase* : Buat Akun

Aktor : User (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 |  |
| 2 User memilih menu registrasi |  |
| 3 User meng-inputkan data |  |
|  | 4 Sistem menerima data dari pelanggan. |
|  | 5 Jika data sesuai kriteria maka sistem membuat akun baru dan mengupdate database lalu menampilkan laman login. Jika tidak kembali ke langkah 3. |

Table 3 Usecase Scenario Registrasi

#### 3.1.1.1 Perancangan Antarmuka Usecase #1

*Antarmuka Registrasi*

Graphical user interface, application

Description automatically generated

Gambar 3 Antarmuka Registrasi

*Page Registrasi*

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| *Reg\_Button1* | *Button* | *REGISTER* | *Jika diklik,sistem akan memverifikasi data yang diinput lalu membuat akun baru di database* |
| *Reg\_text1* | *Text* | *Email* | *Menerima Inputan User* |
| *Reg\_text2* | *Text* | *Username* | *Menerima Inputan User* |
| *Reg\_text3* | *Text* | *Password* | *Menerima Inputan User* |

Table 4 Objek Registrasi

#### 3.1.1.2 Identifikasi Object Baru\

*TABEL OBJECT PERANCANGAN*

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

Table 5 Identifikasi Objek Baru

#### 3.1.1.3 Diagram Kelas

Diagram

Description automatically generated with low confidence

Gambar 4 Diagram Kelas Registrasi

#### 3.1.1.4 Sequence Diagram

Diagram

Description automatically generated

Gambar 5 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. |

Table 6 Usecase Scenario Login

#### 3.1.2.1 Perancangan Antarmuka Usecase #2

*Antarmuka Laman Login*

*![Graphical user interface, application

Description automatically generated]()*

Gambar 6 Antarmuka Login

*Page Login*

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| *Login\_text1* | *Text* | *USERNAME* | *Menerima inputan User* |
| *Login\_text2* | *Text* | *PASSWORD* | *Menerima inputan User* |
| *Login\_button* | *Button* | *SIGN IN* | *Jika diklik, maka user akan dialihkan ke page dashboard* |
| *Register\_bbutton* | *Button* | *Register* | *Jika diklik, maka user akan dialihkan ke page register* |

Table 7 Objek Login

#### 3.1.2.2 Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| 1 |  |  |

Table 8 Objek Baru Login

#### 3.1.2.3 Diagram Kelas

Diagram

Description automatically generated with low confidence

Gambar 7 Diagram Kelas Login

#### 3.1.2.4 Sequence Diagram

Diagram, timeline

Description automatically generated

Gambar 8 Sequence Diagram Login

### Use Case #3 <Booking>

*Skenario Use Case #3*

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 “Finish Order” |  |
|  | 4. Menampilkan daftar pesanan. |
| 5. User memilih metode pembayaran. |  |

Table 9 Usecase Scenario Booking

#### 3.1.3.1 Perancangan Antarmuka Usecase #3

*Antarmuka Laman Booking*

*Graphical user interface, text, application, email

Description automatically generated*

Gambar 9 Antarmuka Booking

*Page Detail Booking*

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| *noMeja\_text1* | *Text* | *\*NO TABLE\** | *Menerima inputan user berupa nomor meja* |
| *booking\_text1* | *Text* | *Please Select One* | *Menerima inputan user berupa menu makanan, minuman dan snacks* |
| *booking\_button1* | *Button* | *Add* | *Menambah text area baru untuk menerima inputan menu makananan, minuman atau snacks dari user* |
| *booking\_button2* | *Button* | *Remove* | *Menghapus text area* |
| *booking\_button3* | *Button* | *Finish Order* | *Jika diklik,user diharapkan segera membayar dengan metode pembayaran yang tersedia.* |

Table 10 Objek Booking

#### 3.1.3.2 Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| 1 | Order Details | Entity |

Table 11 Identifikasi Objek Baru

#### 3.1.3.3 Diagram Kelas

*Table

Description automatically generated*

Gambar 10 Diagram Kelas Booking

#### 3.1.3.4 Sequence Diagram

Graphical user interface, application

Description automatically generated with medium confidence

Gambar 11 Sequence Diagram Booking

### Use Case #4 <Payment>

*Skenario Use Case #4*

Nama *Usecase* : Payment

Aktor : User

*Pre-Condition* : User sudah memilih pesanan

Post-Condition : User membayar pesanan yang di pesan

Deskripsi : Untuk membayar pesanan

|  |  |
| --- | --- |
| Aktor | Sistem |
| 1. Memilih metode pembayaran |  |
|  | 2. Menampilkan metode pembayaran yang dipilih |
| 3. Melakukan pembayaran |  |

Table 12 Usecase Scenario Booking

#### 3.1.4.1 Perancangan Antarmuka Usecase #3

*Antarmuka Laman Payment*

*A screenshot of a computer

Description automatically generated with medium confidence*

Gambar 12 Antarmuka Payment

*Page Detail Payment*

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| *booking\_Button1* | *Button* | *E-Money* | *Merupakan radio button untuk memilih metode pembayaran E-Money* |
| *booking\_button2* | *Button* | *Transfer Bank* | *Merupakan radio button untuk memilih metode pembayaran Transfer Bank* |
| *booking\_text1* | *Text* | *\*Nomor Virtual Account\** | *Memberikan nomor virtual account Xpoint* |
| *booking\_button3* | *Button* | *Done Payment* | *Jika diklik,user diharapkan segera membayar dengan metode pembayaran yang sudah dipilih* |
| *booking\_button4* | *Button* | *Close* | *Jika diklik, user akan dialihkan keluar dari menu payment* |

*Tabel 13 Objek Booking*

#### 3.1.4.2 Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| 1 | Order List | Entity |

Table 13 Identifikasi Objek Baru

#### 3.1.4.3 Diagram Kelas

*Table

Description automatically generated*

Gambar 13 Diagram Kelas Payment

#### 3.1.4.4 Sequence Diagram

Timeline

Description automatically generated with low confidence

Gambar 14 Sequence Diagram Payment

# Perancangan Detail

## Perancangan Detail 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* | *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* |
| *2* | *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* |
| *3* | *Menu* | 1. *id\_menu(private)* 2. *nama\_menu* 3. *harga(private)* | 1. *getId\_menu* 2. *getNama\_menu* 3. *getHarga* 4. *setNama\_menu* 5. *setHarga* |
| *4* | *Order* | 1. *id\_order(private)* 2. *tanggal\_order* | 1. *getId\_order* 2. *getTanggal\_order* 3. *setTanggal\_order* |

Table 14 Perancangan Detail Kelas

## Perancangan Kelas Persistensi (\*\*\*Basis Data Skema Tabel)Graphical user interface, application Description automatically generated

*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 sub bab 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 (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  }  }  } |

### 4.3.3 Algoritma #3 (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  } |

## 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* |

Table 15 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 | Booking | Booking |

Table 16 Matriks Kerunutan