**DPPL-01**

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

KOSU-TAN

untuk:

Pencari Kos dan Pemilik Kos

Dipersiapkan oleh:

Dimitri Irfan D 1301180256

Fazainsyah Azka W 1301184335

Muhammad Aziz Pratama 1301180018

Rifqi Luthfi 1301184138

Program Studi Informatika

Fakultas Informatika – Telkom University

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Prodi S1-**  **Informatika**  **Universitas Telkom** | Nomor Dokumen | | Halaman |
| DPPL-01 | | 35 |
| Revisi | *<nomor revisi>* | Tgl: 26/04/2020 |

DAFTAR PERUBAHAN

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX  TGL | - | A | B | C | D | E | F | G |
| Ditulis oleh | Dimitri Irfan D, Muhammad Aziz P, Fazainsyah A.W., Rifqi Luthfi |  |  |  |  |  |  |  |
| Diperiksa oleh |  |  |  |  |  |  |  |  |
| Disetujui oleh |  |  |  |  |  |  |  |  |

Daftar Halaman Perubahan

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

Daftar Isi

[Daftar Tabel 6](#_Toc38828306)

[Daftar Gambar 7](#_Toc38828307)

[1. Pendahuluan 8](#_Toc38828308)

[1.1 Tujuan Penulisan Dokumen 8](#_Toc38828309)

[1.2 Lingkup Masalah 8](#_Toc38828310)

[1.3 Definisi dan Istilah 8](#_Toc38828311)

[1.4 Referensi 8](#_Toc38828312)

[1.5 Sistematika Pembahasan 8](#_Toc38828313)

[2 Deskripsi Perancangan Global 9](#_Toc38828314)

[2.1 Deskripsi Arsitektural 9](#_Toc38828315)

[2.2 Deskripsi Komponen 9](#_Toc38828316)

[3 Perancangan Rinci 11](#_Toc38828317)

[3.1 Realisasi Use Case 11](#_Toc38828318)

[3.1.1 Use Case #1 Sewa kost-an 11](#_Toc38828319)

[3.1.1.1 Perancangan Antarmuka Usecase Sewa kost-an 12](#_Toc38828320)

[3.1.1.2 Tabel Identifikasi Antarmuka / Layar / Page 12](#_Toc38828321)

[3.1.1.3 Identifikasi Object Baru 12](#_Toc38828322)

[3.1.1.4 Robustness Diagram 13](#_Toc38828323)

[3.1.1.5 Diagram Kelas 13](#_Toc38828324)

[3.1.1.6 Sequence Diagram 14](#_Toc38828325)

[3.1.2 Use Case #2 Cari Kost-an 14](#_Toc38828326)

[3.1.2.1 Perancangan Antarmuka Usecase #2 15](#_Toc38828327)

[3.1.2.1.1 Tabel Identifikasi Antarmuka / Layar / Page 15](#_Toc38828328)

[3.1.2.2 Identifikasi Object Baru 15](#_Toc38828329)

[3.1.2.3 Robustness Diagram 16](#_Toc38828330)

[3.1.2.4 Diagram Kelas 16](#_Toc38828331)

[3.1.2.5 Sequence Diagram 16](#_Toc38828332)

[3.1.3 Use Case #3 Cek Riwayat Transaksi 17](#_Toc38828333)

[3.1.3.1 Perancangan Antarmuka Usecase #3 17](#_Toc38828334)

[3.1.3.1.1 Tabel Identifikasi Antarmuka / Layar / Page 18](#_Toc38828335)

[3.1.3.2 Identifikasi Object Baru 19](#_Toc38828336)

[3.1.3.3 Robustness Diagram 19](#_Toc38828337)

[3.1.3.4 Diagram Kelas 19](#_Toc38828338)

[3.1.3.5 Sequence Diagram 19](#_Toc38828339)

[3.1.4 Use Case #4 Post Kost-an 20](#_Toc38828340)

[3.1.4.1 Perancangan Antarmuka Usecase #4 21](#_Toc38828341)

[3.1.4.1.1 Tabel Identifikasi Antarmuka / Layar / Page 21](#_Toc38828342)

[3.1.4.2 Identifikasi Object Baru 21](#_Toc38828343)

[3.1.4.3 Robustness Diagram 22](#_Toc38828344)

[3.1.4.4 Diagram Kelas 22](#_Toc38828345)

[3.1.4.5 Sequence Diagram 22](#_Toc38828346)

[3.1.5 Use Case #5 Update Kost-an 23](#_Toc38828347)

[3.1.5.1 Perancangan Antarmuka Usecase #5 23](#_Toc38828348)

[3.1.5.1.1 Tabel Identifikasi Antarmuka / Layar / Page 23](#_Toc38828349)

[3.1.5.2 Identifikasi Object Baru 24](#_Toc38828350)

[3.1.5.3 Robustness Diagram 24](#_Toc38828351)

[3.1.5.4 Diagram Kelas 24](#_Toc38828352)

[3.1.5.5 Sequence Diagram 25](#_Toc38828353)

[3.1.6 Use Case #6 Delete Kost-an 25](#_Toc38828354)

[3.1.6.1 Perancangan Antarmuka Usecase #6 26](#_Toc38828355)

[3.1.6.1.1 Tabel Identifikasi Antarmuka / Layar / Page 26](#_Toc38828356)

[3.1.6.2 Identifikasi Object Baru 26](#_Toc38828357)

[3.1.6.3 Robustness Diagram 26](#_Toc38828358)

[3.1.6.4 Diagram Kelas 28](#_Toc38828359)

[3.1.6.5 Sequence Diagram 28](#_Toc38828360)

[4 Perancangan Detil 29](#_Toc38828361)

[4.1 Perancangan Detil Kelas 29](#_Toc38828362)

[4.2 Perancangan Kelas Persistensi (\*\*\*Basis Data Skema Tabel) 30](#_Toc38828363)

[4.3 Perancangan Algoritma 31](#_Toc38828364)

[4.3.1 Algoritma #1 31](#_Toc38828365)

[4.3.2 Algoritma #2 31](#_Toc38828366)

[4.3.3 Algoritma #3 31](#_Toc38828367)

[4.3.4 Algoritma #4 32](#_Toc38828368)

[4.3.5 Algoritma #5 32](#_Toc38828369)

[4.3.6 Algoritma #6 32](#_Toc38828370)

[4.3.7 Algoritma #7 32](#_Toc38828371)

[4.3.8 Algoritma #8 32](#_Toc38828372)

[4.4 Perancangan Query 33](#_Toc38828373)

[5 Matriks Kerunutan (Requirement Traceability Matrix) 33](#_Toc38828374)

# Daftar Tabel

[Table 1. 1 Deskripsi Komponen 10](#_Toc38831723)

[Table 1. 2 Realisasi Use Case 11](#_Toc38831724)

[Table 1. 3 Tabel Indentifikasi UI Sewa Kost-an 12](#_Toc38831725)

[Table 1. 4 Tabel Indentifikasi Objek UI Sewa Kost-an 12](#_Toc38831726)

[Table 1. 5 Identifikasi Object Baru Use Case Sewa kost-an 13](#_Toc38831727)

[Table 1. 6 Tabel Identifikasi UI Cari Kost-an 15](#_Toc38831728)

[Table 1. 7 Tabel Identifikasi Object UI Cari Kost-an 16](#_Toc38831729)

[Table 1. 8 Identifiklasi Object Baru Cari Kost-an 16](#_Toc38831730)

[Table 1. 9 Tabel Identifikasi UI Cek Riwayat Transaksi 19](#_Toc38831731)

[Table 1. 10 Tabel Identifikasi Page UI Cek Riwayat Transaksi Pencari 19](#_Toc38831732)

[Table 1. 11 Tabel Identifiasi Page UI Cek Riwayat Transaksi Pemilik 19](#_Toc38831733)

[Table 1. 12 Identifikasi Object Baru Cek Riwayat Transaksi 19](#_Toc38831734)

[Table 1. 13 Tabel Identifikasi UI Post Kost-an 22](#_Toc38831735)

[Table 1. 14 Tabel Identifikasi page UI Post Kost-an 23](#_Toc38831736)

[Table 1. 15 Identifikasi Object Baru Post Kost-an 23](#_Toc38831737)

[Table 1. 16 Tabel Identifikasi UI Update Kost-an 25](#_Toc38831738)

[Table 1. 17 Tabel Identifikasi page UI Update Kost-an 25](#_Toc38831739)

[Table 1. 18 Identifikasi Object Baru Update Kost-an 25](#_Toc38831740)

[Table 1. 19 Tabel Identifikasi Object Delete Kost-an 28](#_Toc38831741)

[Table 1. 20 Tabel Identifikasi page UI Delete Kost-an 28](#_Toc38831742)

[Table 1. 21 Identifikasi Object Baru Delete Kost-an 28](#_Toc38831743)

[Table 1. 22 Perancangan Detil Kelas TABEL KELAS 31](#_Toc38831744)

[Table 1. 23 KEL1 / Controller\_Sewa Operasi 31](#_Toc38831745)

[Table 1. 24 KEL1 / Controller\_Sewa Atribut 31](#_Toc38831746)

[Table 1. 25 KEL2 / Controller\_User Operasi 31](#_Toc38831747)

[Table 1. 26 KEL2 / Controller\_User Atribut 31](#_Toc38831748)

[Table 1. 27KEL3 / Controller\_Kost Operasi 32](#_Toc38831749)

[Table 1. 28 KEL3 / Controller\_Kost Atribut 32](#_Toc38831750)

[Table 1. 29 Skema Tabel 32](#_Toc38831751)

[Table 1. 30 Perancangan Query 35](#_Toc38831752)

[Table 1. 31 Matriks Kerunutan 35](#_Toc38831753)

# Daftar Gambar

[Image 1. 1 Deskripsi Arsitektural 9](#_Toc38831697)

[Image 1. 2 UI Sewa Kost-an 12](#_Toc38831698)

[Image 1. 3 Robustness Diagram Sewa Kost-an 13](#_Toc38831699)

[Image 1. 4 Diagram Kelas Sewa Kost-an 14](#_Toc38831700)

[Image 1. 5 Sequance Diagram Sewa Kost-an 14](#_Toc38831701)

[Image 1. 6 UI Cari Kost-an 15](#_Toc38831702)

[Image 1. 7 Robustness Diagram Cari Kost-an 16](#_Toc38831703)

[Image 1. 8 Diagram kelas Cari Kost-an 16](#_Toc38831704)

[Image 1. 9 Sequence Diagram Cari Kost-an 17](#_Toc38831705)

[Image 1. 10 UI Cek Riwayat Transaksi Penyewa 18](#_Toc38831706)

[Image 1. 11 UI Cek Riwayat Transaksi Pemilik 18](#_Toc38831707)

[Image 1. 12 Robustness Diagram Cek Riwayat Transaksi 20](#_Toc38831708)

[Image 1. 13 Diagram Kelas Cek Riwayat Transaksi 20](#_Toc38831709)

[Image 1. 14 Sequence Diagram Cek Riwayat Transaksi 21](#_Toc38831710)

[Image 1. 15 UI Post Kost-an 22](#_Toc38831711)

[Image 1. 16 Robustness Diagram Post Kost-an 23](#_Toc38831712)

[Image 1. 17 Diagram Kelas Post Kost-an 24](#_Toc38831713)

[Image 1. 18 Sequance Diagram Post Kost-an 24](#_Toc38831714)

[Image 1. 19 UI Update Kost-an 25](#_Toc38831715)

[Image 1. 20 Robustness Diagram Update Kost-an 26](#_Toc38831716)

[Image 1. 21 Diagram Kelas Update Kost-an 26](#_Toc38831717)

[Image 1. 22 Sequence Diagram Update Kost-an 27](#_Toc38831718)

[Image 1. 23 UI Delete Kost-an 28](#_Toc38831719)

[Image 1. 24 Robustness Diagram Delete Kost-an 29](#_Toc38831720)

[Image 1. 25 Diagram Kelas Delete Kost-an 29](#_Toc38831721)

[Image 1. 26 Sequence Diagram Delete Kost-an 30](#_Toc38831722)

# 1. Pendahuluan

## Tujuan Penulisan Dokumen

Tujuan pembuatan dokumen DPPL (Deskripsi Perancangan Perangkat Lunak ) ini adalah untuk mendokumentasikan dan menjelaskan aplikasi atau perangkat yang akan dikembangkan yaitu Kosutan. Dokumen ini akan digunakan dengan harapan dapat menjadi titik temu antara kami sebagai sistem analis dengan pelanggan kami yang akan menggunakan sistem yang kita buat. Dokumen ini juga akan memudahkan kami sebagai pengembang aplikasi untuk mendokumentasikan perubahan fitur dari aplikasi kami. Isi dari dokumen ini meliputi deskripsi dari aplikasi yang akan kami buat.

## Lingkup Masalah

Aplikasi ini bernama Kosutan,aplikasi ini berbasis sebuah web yang diperuntukan untuk penyewaan kost. Target pasar utamanya adalah seseorang yang ingin menyewakan kost nya (Pemilik kost) dan juga orang yang ingin menyewa kost (Penyewa kost). Website Kosutan ini masih dalam tahap kerangka yang dimana website ini masih dapat dikembangkan lagi.

## Definisi dan Istilah

* ERD

Entity Relationship Diagram adalah satu model yang digunakan untuk mendesign database dengan tujuan menggambarkan data yang berelasi pada sebuah database.

* SKPL

Spesifikasi Kebutuhan Perangkat Lunak merupakan spesifikasi dari perangkat lunak yang akan dikembangkan.

* DPPL

Dokumen Deskripsi Perancangan Perangkat Lunak merupakan dokumen deskripsi dari perancangan lunak yang akan dikembangkan dan bertujuan untuk memberikan landasan yang diperlukan dalam proses pengkodean aplikasi.

## Referensi

* Spesifikasi Kebutuhan Perangkat Lunak Kosutan
* Template Dokumen Deskripsi Perancangan Perangkat Lunak Telkom University

## Sistematika Pembahasan

Dokumen ini akan membahas mengenai deskripsi perancangan perangkat lunak yang dikembangkan berdasarkan dokumen SKPL. Dalam dokumen ini terdiri dari tiga bahasan utama yang terbagi menjadi tiga bab, yakni :

* **Bab 1** adalah pendahuluan. Dimana pada bab ini menjelaskan mengenai tujuan pembuatan dokumen DPPL, serta ruang lingkup masalah dari dokumen ini,definisi dan istilah-istilah yang digunakan pada dokumen, serta referensi apa saja yang menjadi acuan dalam pembuatan dokumen serta dalam pengembangan aplikasi dan terakhir adalah deskripsi umum dokumen.
* **Bab 2** adalah deskripsi perancangan. Dimana pada bab ini menjelaskan mengenai fungsional modul, perancangan data.
* **Bab 3** adalah perancangan rinci yang terdiri dari beberapa sub bab yakni, gambaran umum antar muka pengguna, identifikasi kelas domain, sequence diagram, diagram kelas dan robustness diagram.
* **Bab 4** adalah perancangan detail dari kebutuhan perangkat lunak yang meliputi algorithma serta query penting, dan representasi presistensi kelas.
* **Bab 5** adalah matriks kerunutan yaitu Mapping requirement dengan Use Case yang direalisasikan.

# Deskripsi Perancangan Global

## Deskripsi Arsitektural

Berikut penjelasan Arsitektur yang akan dibangun :

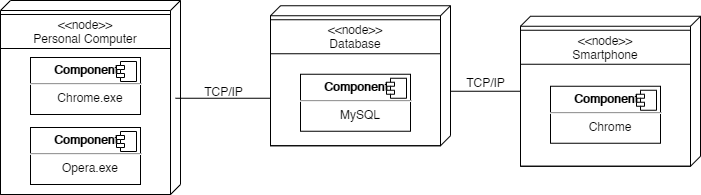
**

Image 1. 1 Deskripsi Arsitektural

## Deskripsi Komponen

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Komponen** | **Keterangan** |
| 1 | Login (Pemilik) | Login akan meminta data email dan password pemilik yang sudah terdaftar di database yang akan dilangsungkan ke tampilan homepage pemilik |
| 2 | Login (Penyewa) | Login akan meminta data email dan password penyewa yang sudah terdaftar dalam database yang akan dilangsungkan ke tampilan homepage penyewa |
| 3 | Reset Password (Pemilik) | Pemilik akan diminta data email yang bersangkutan dan nantinya akan diminta data new password yang dimana akan dijadikan password yang baru |
| 4 | Reset Password (Penyewa) | Penyewa akan diminta data email yang bersangkutan dan nantinya akan diminta data new password yang dimana akan dijadikan password yang baru |
| 5 | Lihat Profil (Pemilik) | Halaman dimana user (pemilik) dapat melihat profile mereka berdasarkan form yang sudah diisi saat register |
| 6 | Lihat Profil (Penyewa) | Halaman dimana user (penyewa) dapat melihat profile mereka berdasarkan form yang sudah diisi saat register |
| 7 | Edit Profile (Pemilik) | Halaman dimana Pemilik dapat mengubah data profil yang dimiliki user |
| 8 | Edit Profile (Penyewa) | Halaman dimana Penyewa dapat mengubah data profil yang dimiliki user |
| 9 | Register (Pemilik) | Halaman dimana user dapat mendaftarkan diri sebagai pemilik yang dimana akan diminta Nama, Username,Email, Password, Confirm Password, Birth date, dan alamat |
| 10 | Register (Penyewa) | Halaman dimana user dapat mendaftarkan diri sebagai pemilik yang dimana akan diminta Nama, Username, Email, Password, Confirm Password, Birth date, dan alamat |
| 11 | Pemilik | User yang dapat mengupdate kost, hapus kost, tambah kost, kelola user(Pemilik), riwayat kost |
| 12 | Update Kost | Halaman dimana pemilik dapat mengupdate data kost yang telah di post/ditambah |
| 13 | Hapus Kost | Halaman dimana pemilik kost dapat menghapus kost yang telah di post/ditambah dan selanjutnya data kost-an tersebut akan dihapus dalam database |
| 14 | Tambah Kost | Halaman dimana pemilik kost dapat menambahkan kost yang dipunyainya ke dalam database kosutan untuk ditampilkan dalam website |
| 15 | Kelola User (Pemilik) | Halaman dimana pemilik dapat mengubah data-data yang dimiliki user |
| 16 | Riwayat Kost | Halaman dimana pemilik kost dapat melihat siapa saja yang pernah sedang tinggal. |
| 17 | Penyewa | User yang dapat menyewa kost, pesan kost, kelola user(Pemilik) |
| 18 | Sewa Kost | Halaman dimana penyewa dapat menyewa kost yang dipilih oleh user |
| 20 | Kelola User (Penyewa) | Halaman dimana pemilik dapat mengubah data-data yang dimiliki user |

Table 1. 1 Deskripsi Komponen

# Perancangan Rinci

## Realisasi Use Case

Berisi TABEL USE CASE sebagai berikut :

|  |  |  |
| --- | --- | --- |
| No | Nama Use Case | Deskripsi Use Case |
| #1 | Sewa kost-an (penyewa) | Penyewa kost-an yang telah berhasil memilih kost-san dapat melakukan sewa sesuai detail pembayaran yang telah diberikan. |
| #2 | Cari kost-an (penyewa, pemilik) | Penyewa atau Pemilik dapat mencari kost-an melalui search bar yang telah disediakan |
| #3 | Cek riwayat kost-an (penyewa, pemilik) | Penyewa atau Pemilik dapat cek riwayat kos setelah berhasil log-in dan memilih icon profil lalu pilih opsi riwayat kos. |
| #4 | Post kost-an (pemilik) | Pemilik Kost-an yang telah berhasil log-in dapat melakukan penambahan data kost-an sesuai inputannya |
| #5 | Update kost-an (pemilik) | Pemilik kost-an dapat mengupdate data kost-an setelah berhasil Post kost-an |
| #6 | Delete kost-an (pemilik) | Pemilik kost-an dapat mendelete data kost-an setelah berhasil Post kost-an |

Table 1. 2 Realisasi Use Case

### Use Case #1 Sewa kost-an

Skenario : Use Case Sewa kost-an

Precondition : Kost-an tersedia dan tersedia ruangan/kamar

Primary Flow :

1. Penyewa Kost-an memilih kost-an yang telah tersedia

2. System menampilkan detail data kost-an yang dipilih Penyewa Kost-an.  
 3. Penyewa Kost-an dapat melihat detail kost-an

4. Penyewa Kost-an mengklik tombol sewa  
 5. system menampilkan detail pembayaran  
 6. Penyewa Kost-an mengklik tombol konfirmasi

7. System mengeluarkan invoice penyewaan

8. System mengupdate data kost-an sementara

Pasca kondisi : Kost-an tersewa, penyewa kost-an mendapatkan ruangan/kamar

Alternate Flow :

1. Penyewa Kost-an belum melunasi sewa

2. Penyewa Kost-an salah melakukan pengambilan sewa kost-an

#### Perancangan Antarmuka Usecase Sewa kost-an

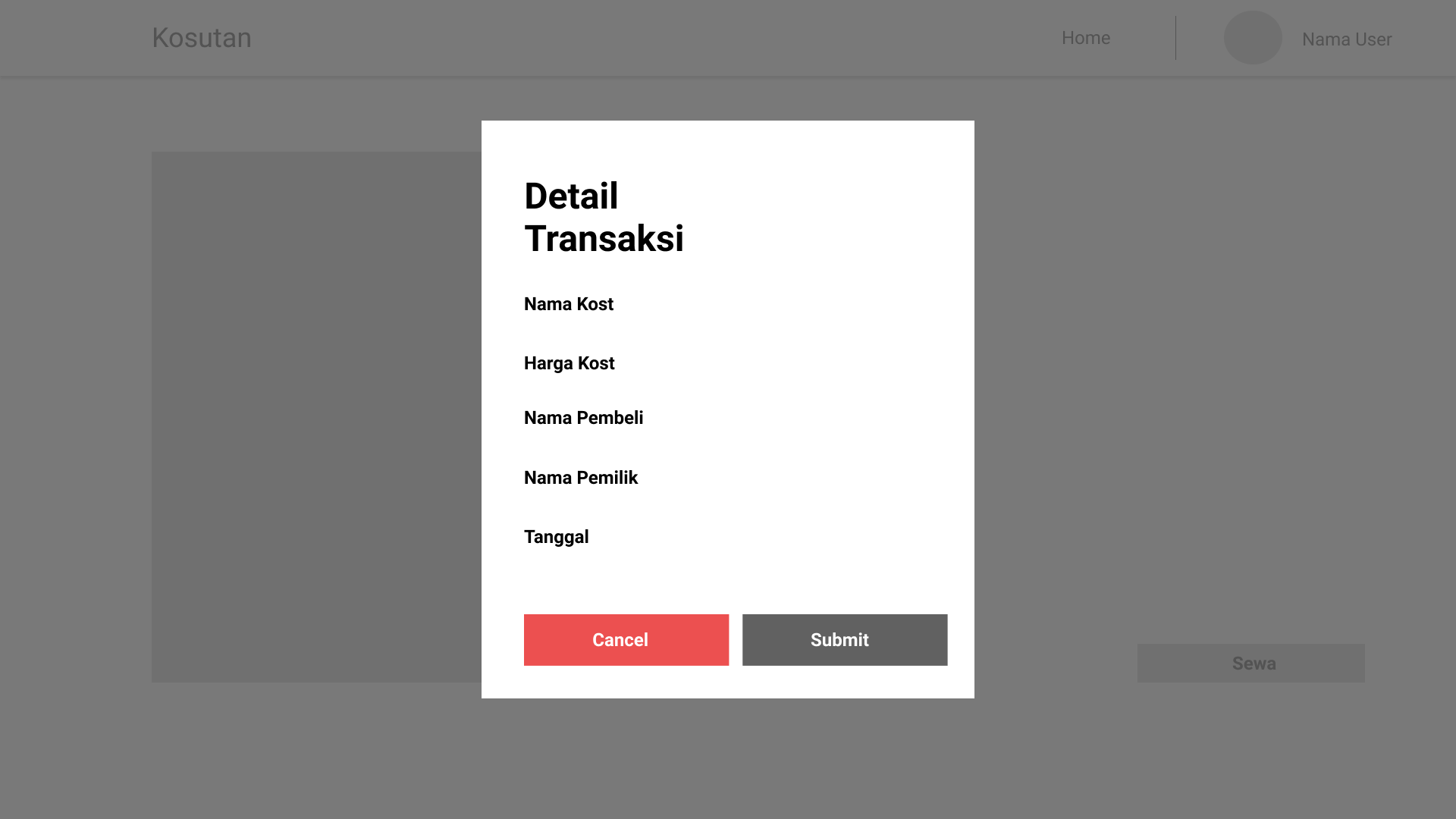


Image 1. 2 UI Sewa Kost-an

#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| UI1 | Page Sewa kost-an | Page ini digunakan untuk penyewa melakukan penyewaan yang dimana nantinya akan menampilkan data kost yang akan disewa dan penyewa dapat melakukan submit jikalau ingin melakukan penyewaan |

Table 1. 3 Tabel Indentifikasi UI Sewa Kost-an

*Page Sewa kost-an*

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| UI1.1 | Button | Btn-sewa | Jika diklik, akan mengaktifkan Function….. |
| UI1.2 | Button | Btn-submit | Jika diklik akan mengaktifkan …… |
| UI1.3 | Button | Btn-cancel | Isi Teks yang disimpan pada File xxx |

Table 1. 4 Tabel Indentifikasi Objek UI Sewa Kost-an

#### Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| 1. | View\_sewa | View |
| 2. | Controller\_sewa | Controller |
| 3. | Controller\_kost | Controller |
| 4. | Sewa | Model |
| 5. | Kost | Model |

Table 1. 5 Identifikasi Object Baru Use Case Sewa kost-an

#### Robustness Diagram

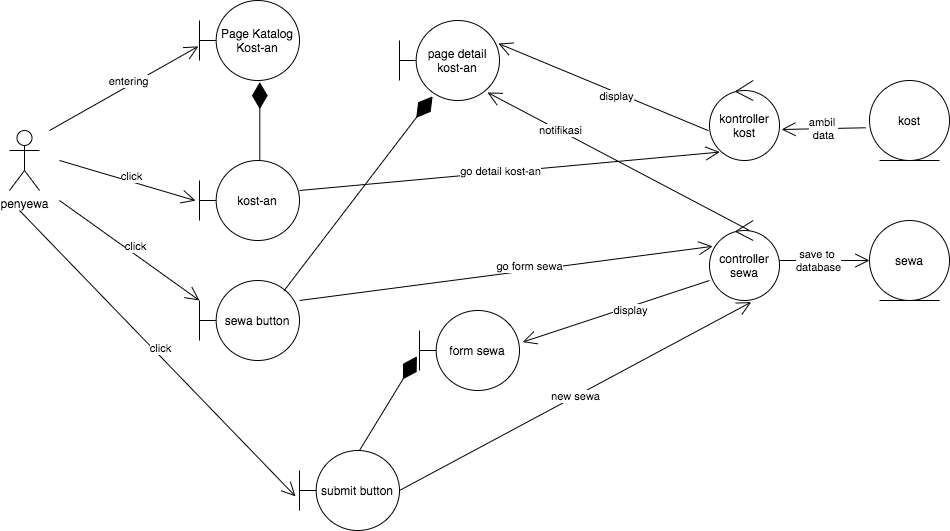


Image 1. 3 Robustness Diagram Sewa Kost-an

#### Diagram Kelas

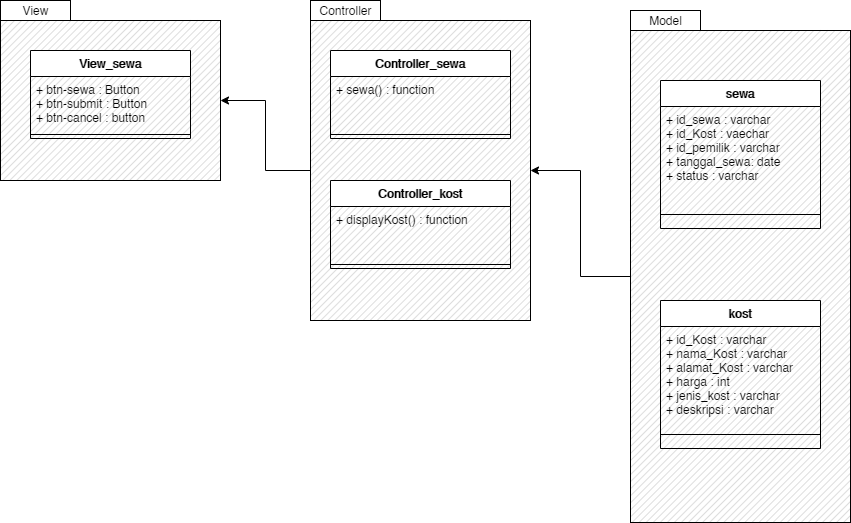


Image 1. 4 Diagram Kelas Sewa Kost-an

#### Sequence Diagram

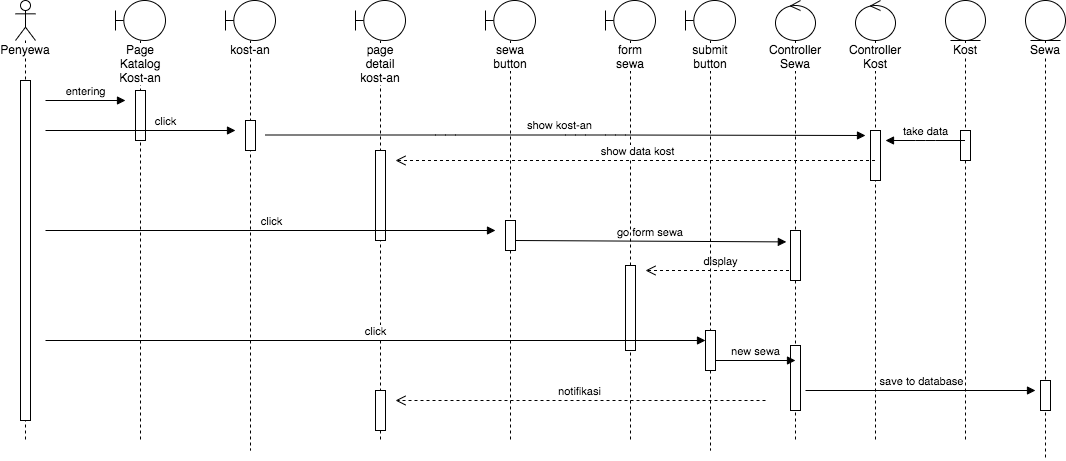


Image 1. 5 Sequance Diagram Sewa Kost-an

### Use Case #2 Cari Kost-an

Skenario : Use Case Cari Kost-an

Precondition : Data-data kosan tersedia

Primary Flow :

1. Penyewa Kost-an mengklik bar search

2. System menampilkan tampilan search  
 3. Penyewa Kost-an menginputkan data Kost-an yang ingin dicari

4. System menampilkan list kosan yang sesuai dengan kriteria inputan dari Penyewa Kost-an

Pasca kondisi : Dapat melihat list data-data kosan yang tersedia

Alternate Flow:

1. Penyewa Kost-an salah melakukan pencarian kost-an

#### Perancangan Antarmuka Usecase #2

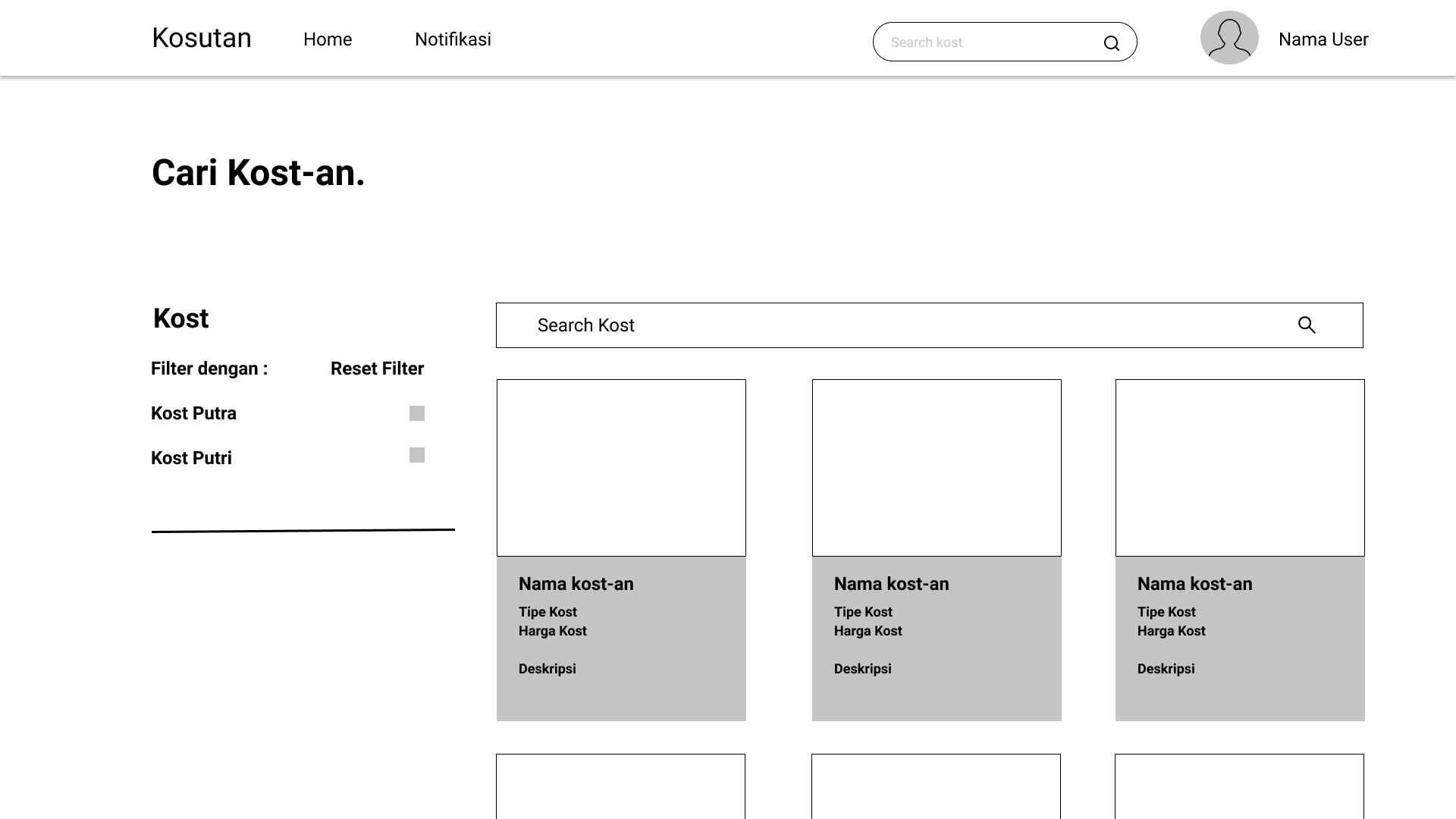


Image 1. 6 UI Cari Kost-an

#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| UI2 | Page Search | Page ini digunakan untuk melakukan searching kost-an yang tersedia oleh user dan juga dapat melakukan filtering berupa kost putra atau kost putri |

Table 1. 6 Tabel Identifikasi UI Cari Kost-an

*Page Search*

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| UI2.1 | Input | Input-search | Keyword yang user masukkan untuk mencari kost |
| UI2.2 | Button | Btn-search | Jika diklik akan mengaktifkan Function Search |
| UI2.3 | Checkbox | Check-putra | Filter jika ingin hanya menampilkan kost putra |
| UI2.4 | Checkbox | Check-putri | Filter jika ingin hanya menampilkan kost putri |

Table 1. 7 Tabel Identifikasi Object UI Cari Kost-an

#### Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
|  | View\_search | View |
|  | Controller\_kost | Controller |
|  | kost | Model |

Table 1. 8 Identifiklasi Object Baru Cari Kost-an

#### Robustness Diagram

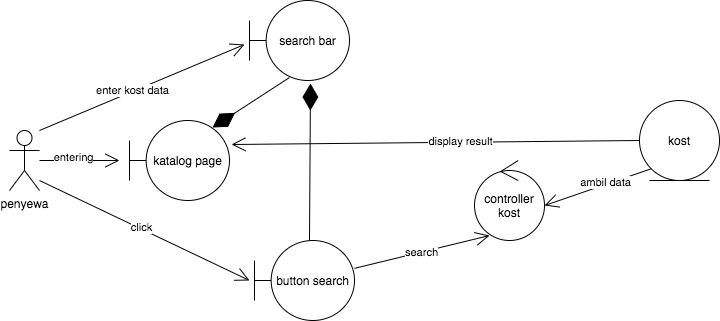


Image 1. 7 Robustness Diagram Cari Kost-an

#### Diagram Kelas

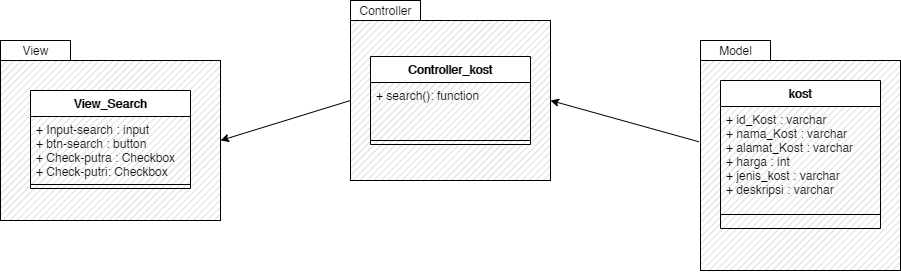


Image 1. 8 Diagram kelas Cari Kost-an

#### Sequence Diagram

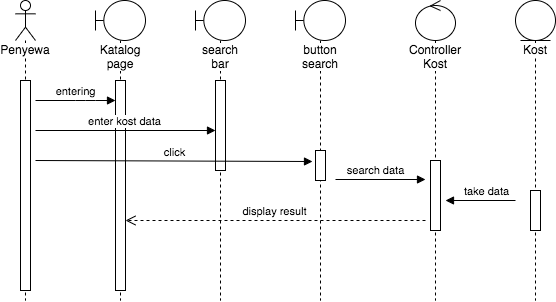


Image 1. 9 Sequence Diagram Cari Kost-an

### Use Case #3 Cek Riwayat Transaksi

Skenario : Use Case Cek Riwayat Transaksi

Precondition : Pencari/penyewa Kost-an pernah melakukan transaksi sewa kosan

Primary Flow :

1. Pencari/penyewa Kost-an memilih menu profile

2. System menampilkan profil dari Pencari/penyewa Kost-an

3. Pencari/penyewa Kost-an memilih menu riwayat

4. System menampilkan riwayat transaksi dari Pencari/penyewa Kost-an

Pasca kondisi : Pencari/penyewa kosan dapat melihat transaksi yang pernah dilakukannya

Alternate Flow: -

#### Perancangan Antarmuka Usecase #3

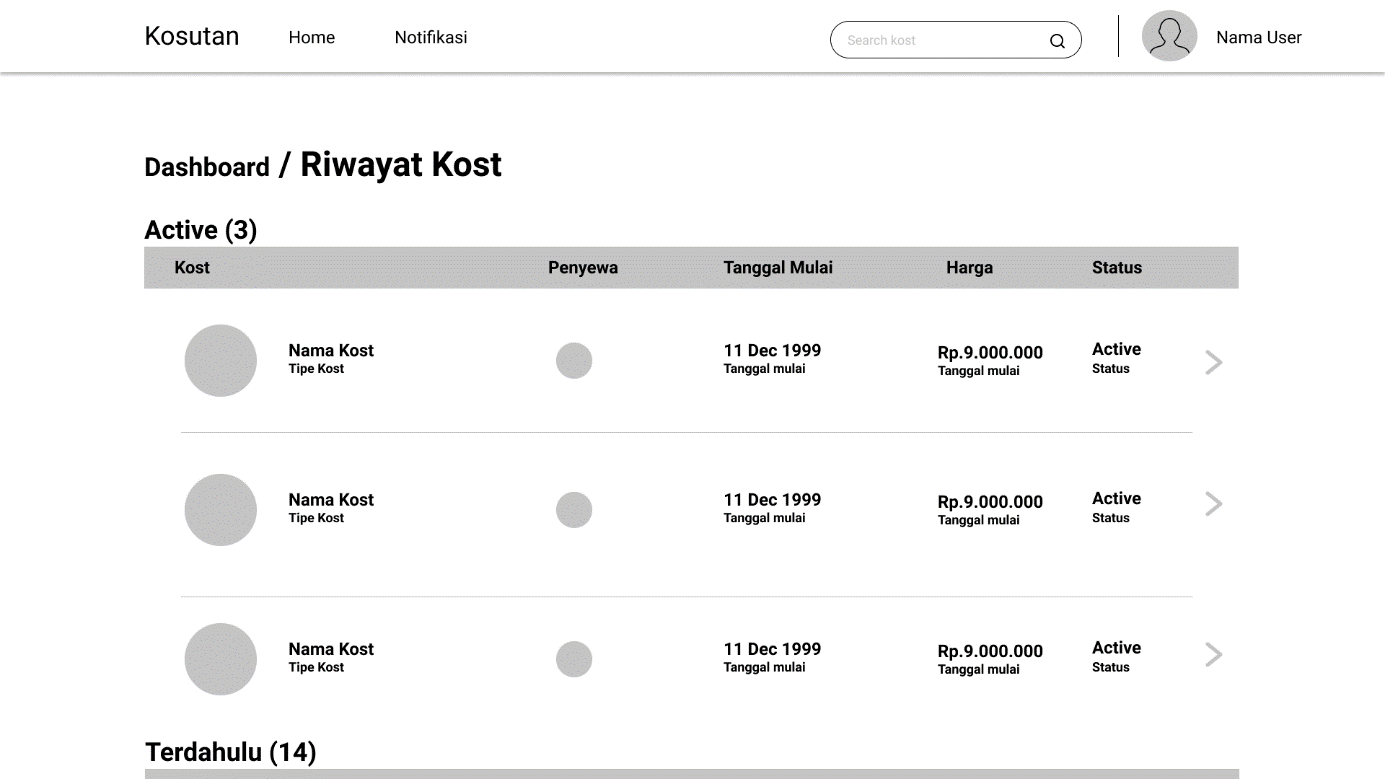


Image 1. 10 UI Cek Riwayat Transaksi Penyewa

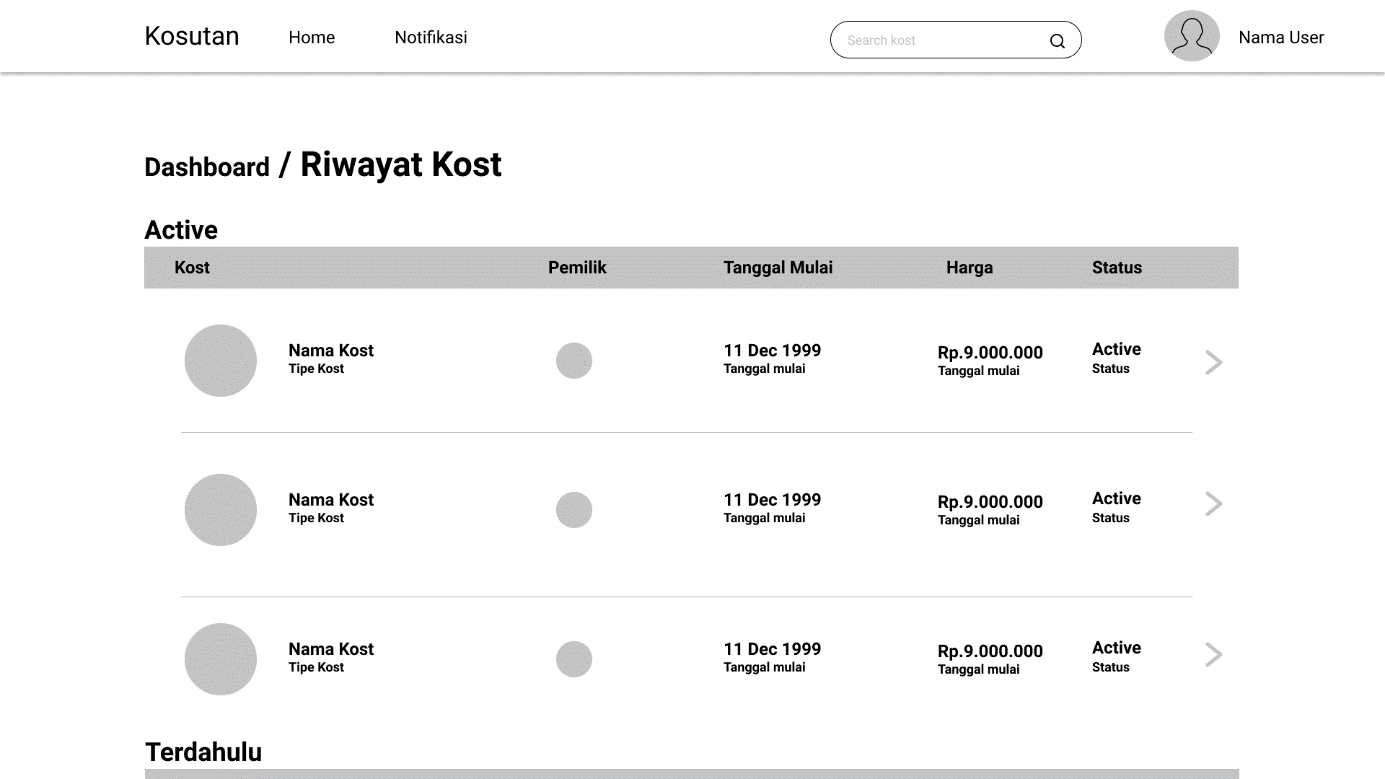


Image 1. 11 UI Cek Riwayat Transaksi Pemilik

#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| UI3 | Page Riwayat Kost (pencari) | Menampilkan seluruh informasi riwayat kost yang sedang disewa oleh pemilik atau kost yang sudah berlalu |
| UI4 | Page Riwayat Kost (pemilik) | Menampilkan seluruh informasi riwayat kost yang dimiliki oleh pemilik |

Table 1. 9 Tabel Identifikasi UI Cek Riwayat Transaksi

*Page Riwayat Kost (pencari)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| UI3.1 | Table | Table-active | Menampilkan kost yang sedang disewa |
| UI3.2 | Table | Table-terdahulu | Menampilkan kost yang pernah disewa |
| UI3.3 | Button | Btn-detail | Menampilkan detail dari kost pada table |
| UI3.4 | Image | Img-pemilik | Menampilkan informasi dari pemilik kost pada table |
| U3.5 | Button | Btn-dashboard | Jika diklik akan kembali ke menu dashboard |

Table 1. 10 Tabel Identifikasi Page UI Cek Riwayat Transaksi Pencari

*Page Riwayat Kost (pemilik)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| UI4.1 | Table | Table-active | Menampilkan kost yang sedang disewa pencari kost |
| UI4.2 | Table | Table-terdahulu | Menampilkan riwayat kost yang pernah disewa oleh pencari kost |
| UI4.3 | Button | Btn-detail | Menampilkan detail dari kost pada table |
| UI4.4 | Image | Img-pencari | Menampilkan informasi dari pemcari kost |
| UI4.5 | button | Btn-dashboard | Jika diklik akan kembali ke menu dashboard |

Table 1. 11 Tabel Identifiasi Page UI Cek Riwayat Transaksi Pemilik

#### Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| 1. | View\_riwayat\_pencari | View |
| 2. | View\_riwayat\_pemilik | View |
| 3. | Controller\_user | Controller |
| 4. | Sewa | Model |

Table 1. 12 Identifikasi Object Baru Cek Riwayat Transaksi

#### Robustness Diagram

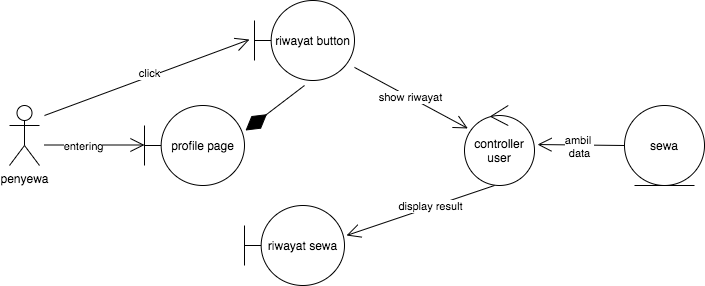


Image 1. 12 Robustness Diagram Cek Riwayat Transaksi

#### Diagram Kelas

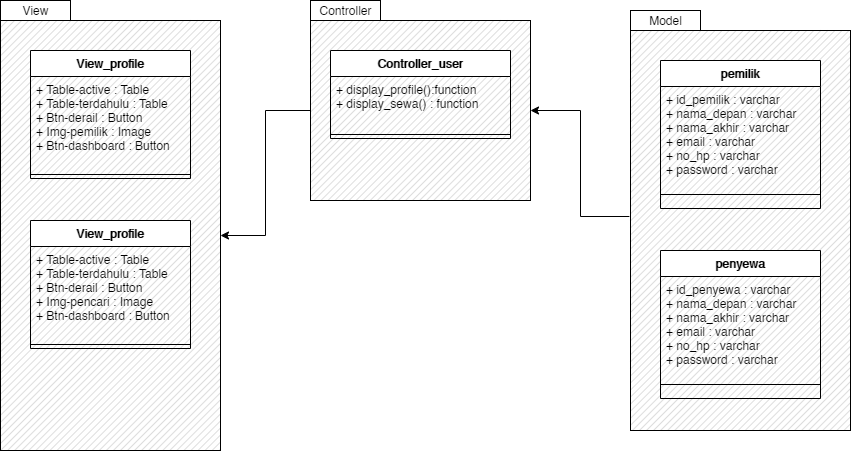


Image 1. 13 Diagram Kelas Cek Riwayat Transaksi

#### Sequence Diagram

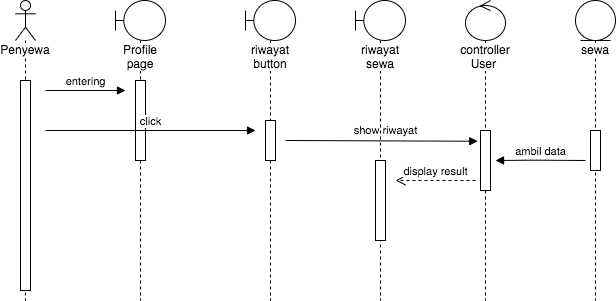


Image 1. 14 Sequence Diagram Cek Riwayat Transaksi

### Use Case #4 Post Kost-an

Skenario : Use Case Post Kost-an

Precondition : Kosan belum pernah didaftarkan

Primary Flow :

1. Pemilik Kost-an memilih post kosan

2. System menampilkan form post kosan

3. Pemilik Kost-an menginputkan data kosan

4. System menambahkan data yang diinputkan Pemilik Kost-an ke database

5. System Menampilkan data yang telah berhasil diinputkan

6. Pemilik Kost-an mengecek kembali data yang telah diinput

Pasca kondisi : Pemilik kosan telah menginputkan data-data kosan dan data-data kosan telah masuk ke database

Alternate Flow :

1. Pemilik Kost-an belum menginputkan data kosan

#### Perancangan Antarmuka Usecase #4

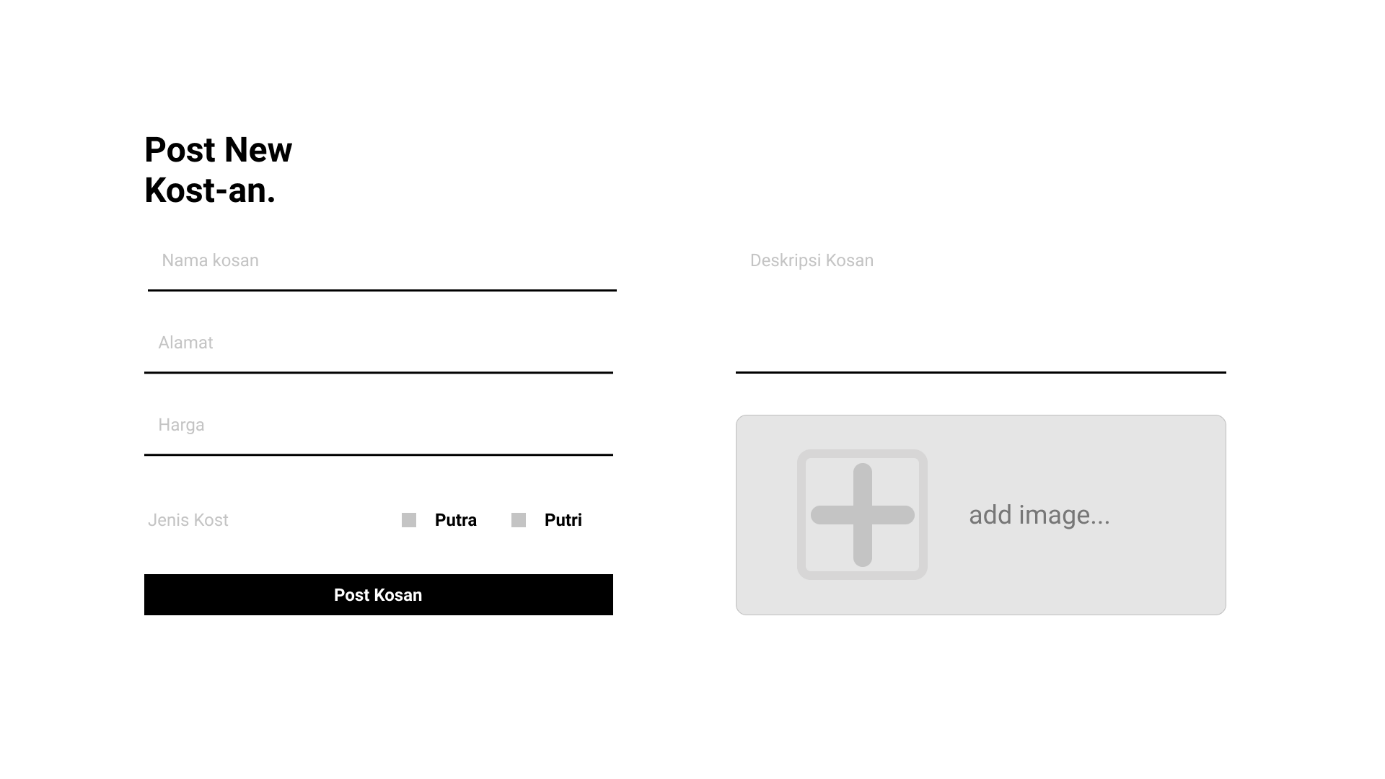


Image 1. 15 UI Post Kost-an

#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| U15 | Form Post(pemilik) | Form untuk menginputkan data kost-an |

Table 1. 13 Tabel Identifikasi UI Post Kost-an

*Page Form Post(pemilik)*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Id\_Objek*** | ***JENIS*** | ***LABEL\**** | ***Keterangan\*\**** |
| UI5.1 | Input | Input-namaKost | Text pengimputan untuk penamaan kost yang akan di post |
| UI5.2 | Input | Input-alamat | Text pengimputan untuk alamat kost yang akan di post |
| UI5.3 | Input | Input-harga | Text pengimputan untuk harga kost yang akan di post |
| UI5.4 | CheckBox | Check-jenisKost | Checkbox untuk jenis kost yang akan di post |
| UI5.5 | Input | Input-deskripsi | Text pengimputan untuk deskripsi kost yang akan di post |
| UI5.6 | Input | Input-imgKost | Memasukkan inputan gambar kost yang akan di post |

Table 1. 14 Tabel Identifikasi page UI Post Kost-an

#### Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| 1. | View\_addKost | View |
| 2. | Controller\_kost | Controller |
| 3. | Kost | Model |

Table 1. 15 Identifikasi Object Baru Post Kost-an

#### Robustness Diagram

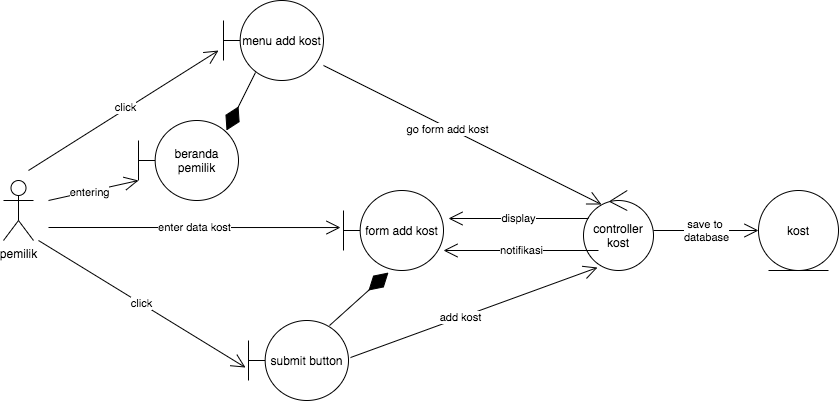


Image 1. 16 Robustness Diagram Post Kost-an

#### Diagram Kelas

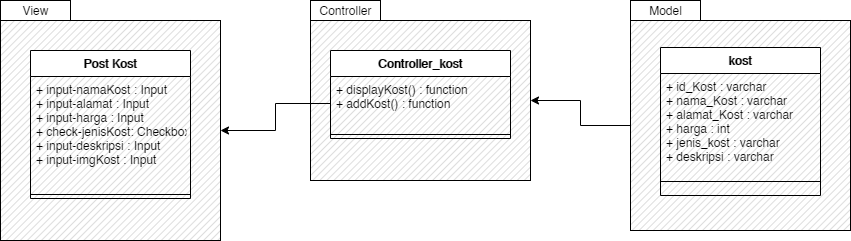


Image 1. 17 Diagram Kelas Post Kost-an

#### Sequence Diagram

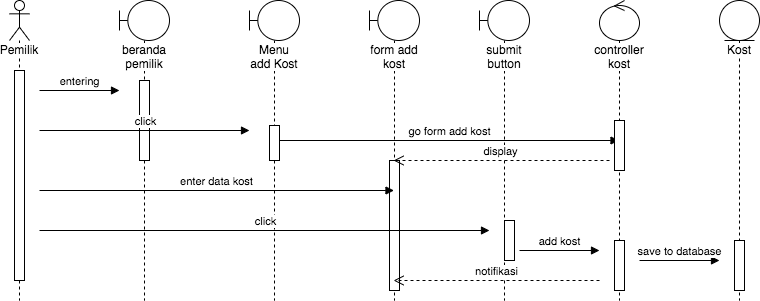


Image 1. 18 Sequance Diagram Post Kost-an

### Use Case #5 Update Kost-an

Skenario : Use Case Update Kost-an

Precondition : Data kosan sudah ada

Primary Flow :

1. Pemilik Kosan memilih kosan yang ingin di update

2. System menampilkan form update kosan

3. Pemilik Kosan menginputkan data yang ingin di update

4. System mengupdate data yang sesuai dengan inputan Pemilik Kosan

5. System menampilkan data yang telah berhasil diupdate

6. Pemilik Kosan melihat data yang telah berhasil diinputkan dan diupdate

Pasca kondisi : Data kosan berhasil diperbaharui

Alternate Flow :

1. Pemilik Kost-an salah menginputkan data kosan

#### Perancangan Antarmuka Usecase #5

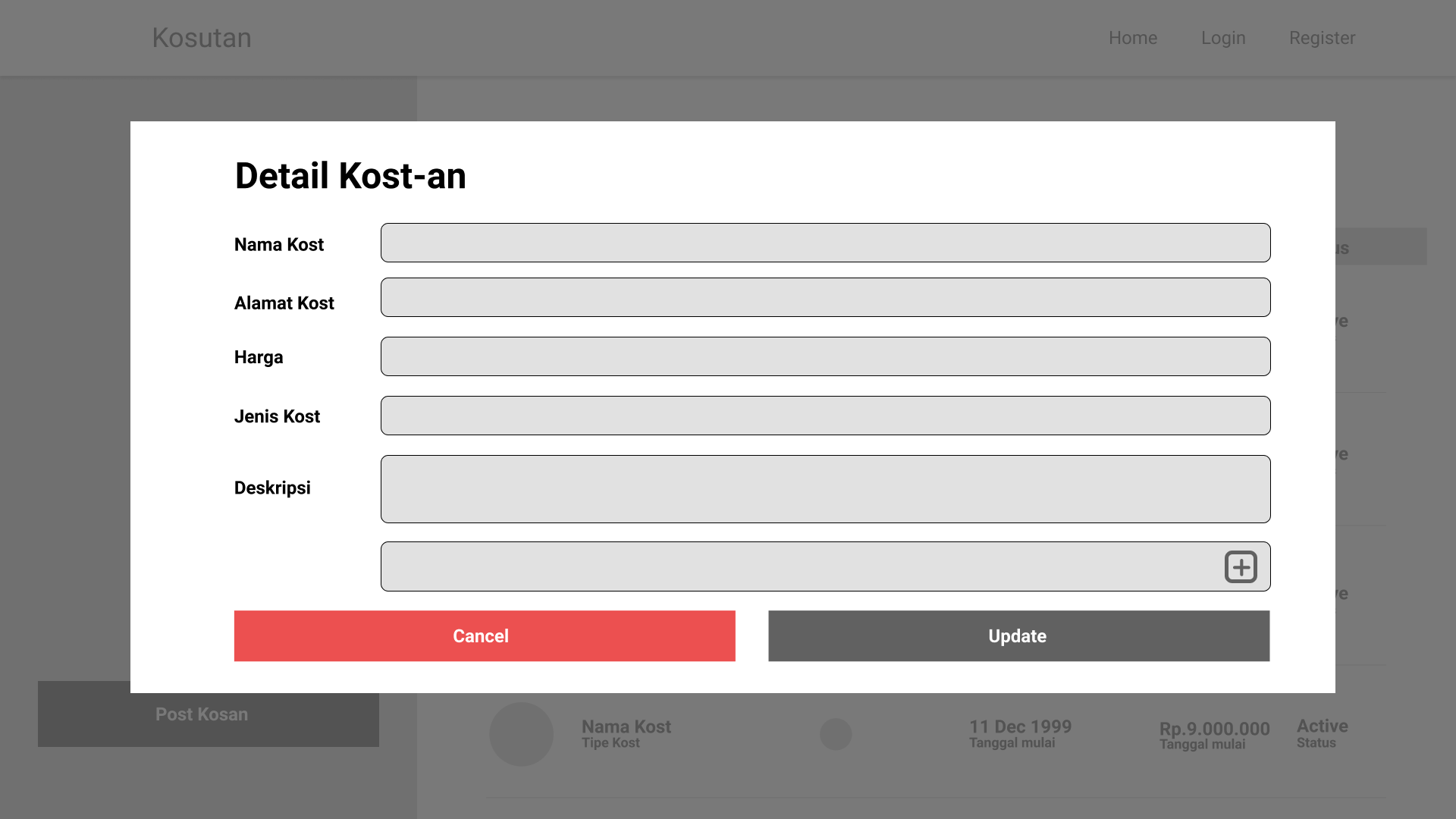


Image 1. 19 UI Update Kost-an

#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| UI6 | Form update kost | Form untuk mengupdate data kost dari database data kost-an |

Table 1. 16 Tabel Identifikasi UI Update Kost-an

*Page Update Kost*

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| UI6.1 | Input | Input-nama | Memasukan nilai input nama |
| UI6.2 | Input | Input-alamat | Memasukan nilai input alamat |
| UI6.3 | Input | Input-harga | Memasukan nilai input harga |
| UI6.4 | input | Input-jenis | Memasukan nilai input jenis |
| UI6.5 | input | Input-deskripsi | Memasukan nilai input deksripsi |
| UI6.6 | Button | Btn-cancel | Cancel update |
| UI6.7 | button | Btn-update | Update perubahan pada input |

Table 1. 17 Tabel Identifikasi page UI Update Kost-an

#### Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| 1. | View\_update\_kost | View |
| 2. | Controller\_kost | Controller |
| 3. | Model\_kost | Model |

Table 1. 18 Identifikasi Object Baru Update Kost-an

#### Robustness Diagram

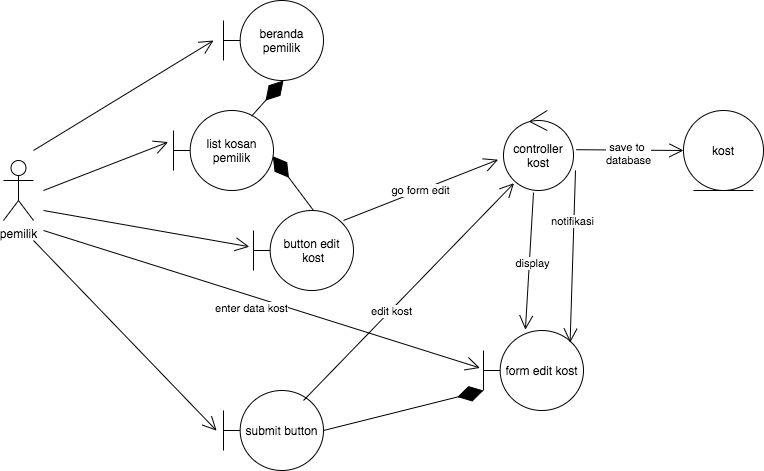


Image 1. 20 Robustness Diagram Update Kost-an

#### Diagram Kelas

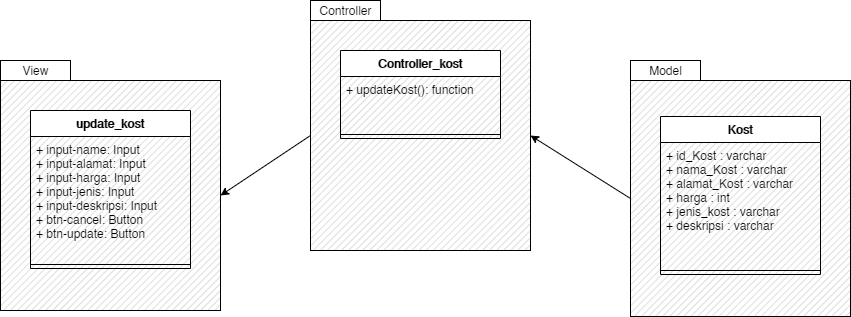


Image 1. 21 Diagram Kelas Update Kost-an

#### Sequence Diagram

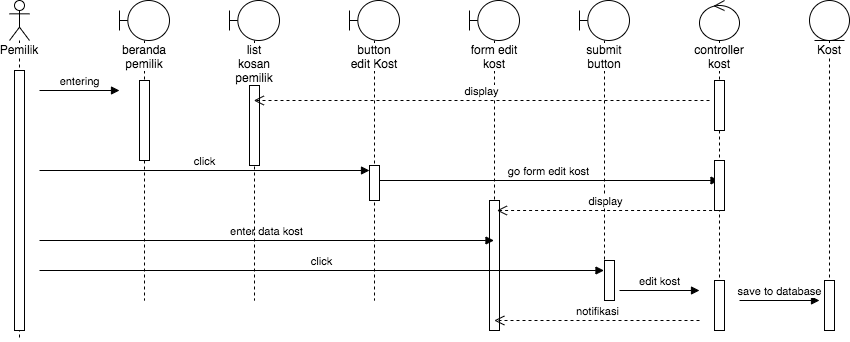


Image 1. 22 Sequence Diagram Update Kost-an

### Use Case #6 Delete Kost-an

Skenario : Use Case Delete Kost-an

Precondition : Data kosan sudah ada

Primary Flow :

1. Pemilik Kosan memilih kost-an yang ingin dihapus

2. System menampilkan notifikasi kepastian apakah benar ingin menghapus kost-an

3. Pemilik Kosan memilih hapus kost-an

4. System menghapus data kost-an yang dipilih actor

5. System berhasil menghapus data kost-an

Pasca kondisi : Data kosan berhasil diperbaharui

Alternate Flow:

1. Pemilik Kost-an salah menginputkan data kosan

#### Perancangan Antarmuka Usecase #6

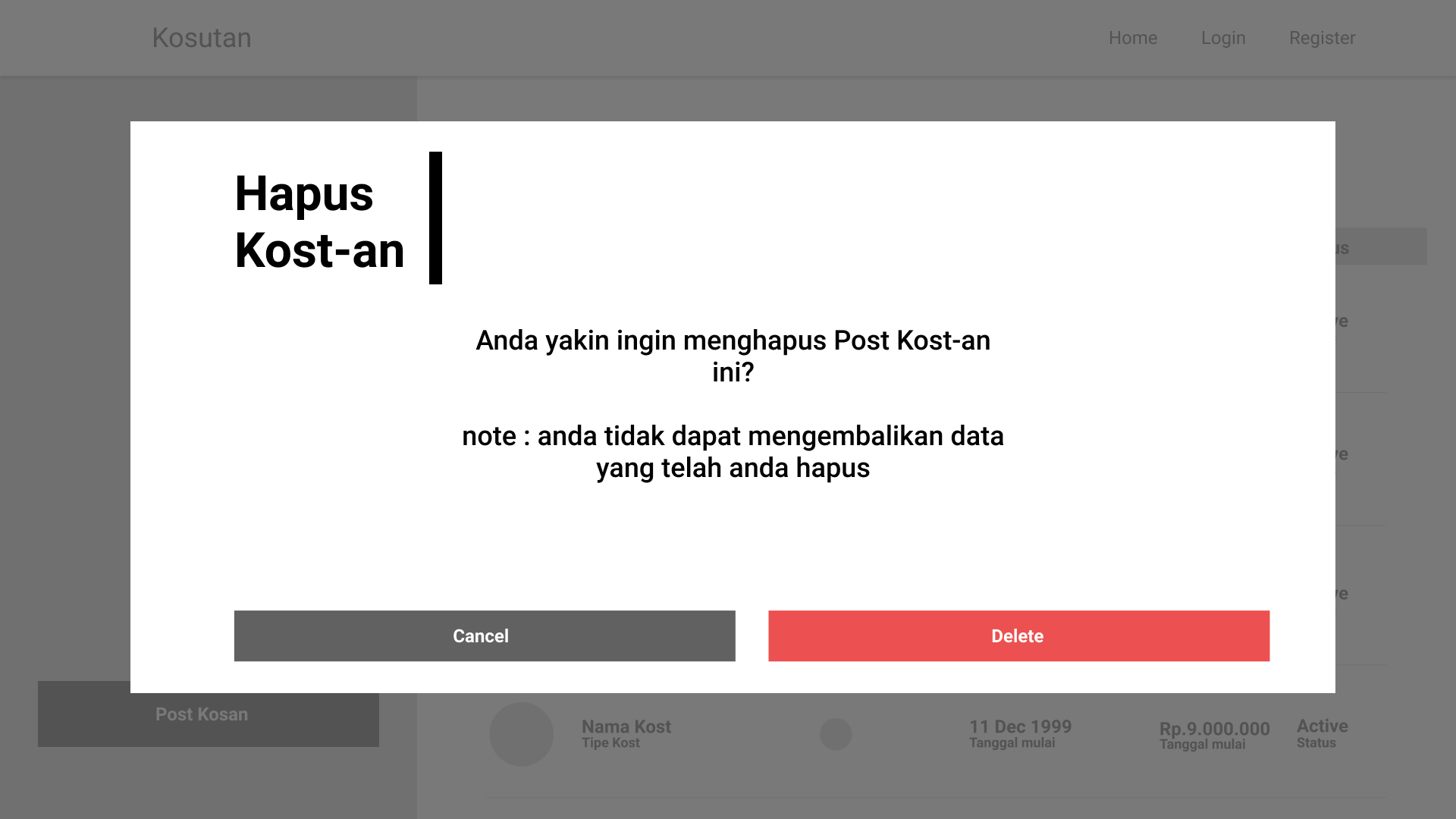


Image 1. 23 UI Delete Kost-an

#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| U16 | Page delete kost-an | Page ini dapat dilakukan oleh pemilik kost-an untuk menghapus data dari database data kosan |

Table 1. 19 Tabel Identifikasi Object Delete Kost-an

*Page HOMEPAGE*

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| UI6.1 | Button | Btn-cancel | Button ini berfungsi untuk cancel function delete post |
| UI6.2 | Button | Btn-delete | Button ini berfungsi untuk mengkonfirmasi pada saat melakukan delete post |

Table 1. 20 Tabel Identifikasi page UI Delete Kost-an

#### Identifikasi Object Baru

TABEL OBJECT PERANCANGAN

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| 1. | View\_delete\_kost | View |
| 2. | Controller\_kost | Controller |
| 3. | Kost | Model |

Table 1. 21 Identifikasi Object Baru Delete Kost-an

#### Robustness Diagram

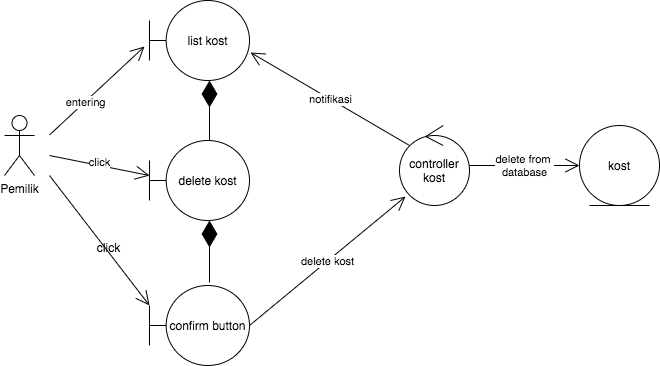


Image 1. 24 Robustness Diagram Delete Kost-an

#### Diagram Kelas

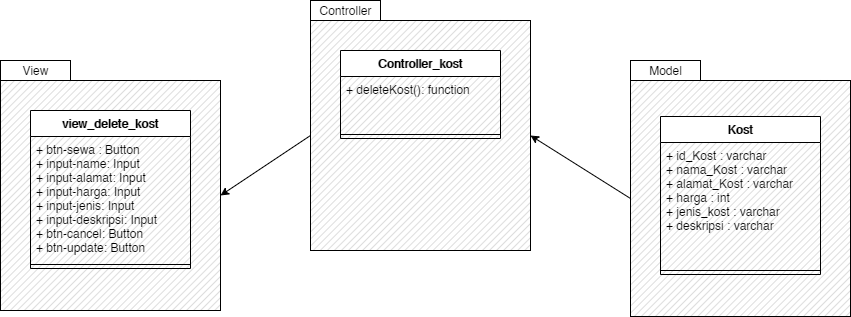


Image 1. 25 Diagram Kelas Delete Kost-an

#### Sequence Diagram

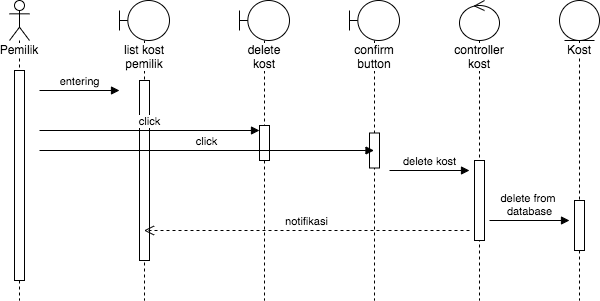


Image 1. 26 Sequence Diagram Delete Kost-an

# Perancangan Detil

#### Perancangan Detil Kelas

**TABEL KELAS :**

|  |  |  |
| --- | --- | --- |
| **ID Kelas** | **Nama Kelas Perancangan** | **Nama Kelas Perancangan Terkait (Model Domain)** |
| KEL1 | Controller\_sewa | Sewa |
| KEL2 | Controller\_user | User |
| KEL3 | Controller\_kost | Kost |

Table 1. 22 Perancangan Detil Kelas TABEL KELAS

* + 1. **Kelas KEL1 / Controller\_Sewa**

|  |  |  |
| --- | --- | --- |
| **Nama Operasi** | **Visibility**  **(private, public)** | **Keterangan** |
| sewa() | public | Fungsi sewa digunakan untuk menjalankan proses sewa |

Table 1. 23 KEL1 / Controller\_Sewa Operasi

|  |  |  |
| --- | --- | --- |
| **Nama Atribut** | **Visibility**  **(private, public)** | **Tipe** |
| id\_sewa | public | Varchar(10) |
| id\_kost | public | Varchar(10) |
| id\_pemilik | public | Varchar(10) |
| tanggal\_sewa | public | Date |
| status | public | Varchar(10) |

Table 1. 24 KEL1 / Controller\_Sewa Atribut

* + 1. **Kelas KEL2 / Controller\_User**

|  |  |  |
| --- | --- | --- |
| **Nama Operasi** | **Visibility**  **(private, public)** | **Keterangan** |
| display\_profile() | public | Fungsi display\_profile digunakan untuk menampilkan data user |
| display\_sewa() | public | Fungsi display\_ sewa digun  digunakan |

Table 1. 25 KEL2 / Controller\_User Operasi

|  |  |  |
| --- | --- | --- |
| **Nama Atribut** | **Visibility**  **(private, public)** | **Tipe** |
| id\_pemilik | public | Varchar(10) |
| Id\_penyewa | public | Varchar(10) |
| nama\_depan | public | Varchar(30) |
| Nama\_akhir | public | Varchar(30) |
| Email | public | Varchar(40) |
| No\_hp | public | Varchar(15) |
| Password | public | Varchar(40) |

Table 1. 26 KEL2 / Controller\_User Atribut

* + 1. **Kelas KEL3 / Controller\_Kost**

|  |  |  |
| --- | --- | --- |
| **Nama Operasi** | **Visibility**  **(private, public)** | **Keterangan** |
| displayKost() | public | Fungsi yang menampilkan data dari kost |
| deleteKost() | public | Fungsi yang menghapus data kost dari database |
| search() | public | Fungsi yang mencari data kost sesuai dengan keyword yang diberikan |
| updateKost() | public | Fungsi yang memperbarui data kost |
| addKost() | public | Fungsi yang dapat menambahkan data kost yang baru |

Table 1. 27KEL3 / Controller\_Kost Operasi

|  |  |  |
| --- | --- | --- |
| **Nama Atribut** | **Visibility**  **(private, public)** | **Tipe** |
| id\_kost | public | Varchar(10) |
| nama\_kost | public | Varchar(50) |
| alamat\_kost | public | Varchar(200) |
| harga | public | Int(100) |
| jenis\_kost | public | Varchar(20) |
| deskripsi | public | Varchar(300) |

Table 1. 28 KEL3 / Controller\_Kost Atribut

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



Table 1. 29 Skema Tabel

#### Perancangan Algoritma

#### Algoritma #1

Nama Kelas : Controller\_sewa

Nama Operasi : sewa()

Algoritma : ALGO001

|  |
| --- |
| Function sewa() {  Input(data)  Sql <- “INSERT into Sewa VALUES (data)”  } |

#### Algoritma #2

Nama Kelas : Controller\_sewa

Nama Operasi : display\_sewa()

Algoritma : ALGO002

|  |
| --- |
| Function display\_sewa() {  Sql <- “SELECT \* FROM sewa WHERE id\_penyewa= ‘id’‘ # untuk pnyewa kost  Sql <- “SELECT \* FROM sewa JOIN pemilik USING id\_kost WHERE id\_pemilik = ‘id’” #untuk pemilik kost    Output(sql)  } |

#### Algoritma #3

Nama Kelas: Controller\_user

Nama Operasi: display\_profile()

Algoritma : ALGO003

|  |
| --- |
| function display\_profile() {  Sql <- “SELECT \* FROM penyewa WHERE id\_penyewa = ‘id’” # untuk penyewa kost  Sql <- “SELECT \* FROM pemilik WHERE id\_pemilik = ‘id’” #untuk pemilik kost  Output(sql)  } |

#### Algoritma #4

Nama Kelas : Controller\_kost

Nama Operasi : display\_kost()

Algoritma : ALGO004

|  |
| --- |
| Function display\_kost() {  Sql <- “SELECT \* FROM kost”  Output(sql)  } |

#### Algoritma #5

Nama Kelas: Controller\_kost

Nama Operasi: deleteKost()

Algoritma: ALGO005

Nama Kelas : Controller\_kost

Nama Operasi : deleteKost()

|  |
| --- |
| Function deleteKost() {  Sql <- “DELETE FROM kost WHERE id\_kost = ‘id’”  } |

#### Algoritma #6

Nama Kelas : Controller\_kost

Nama Operasi : search()

Algoritma : ALGO006

|  |
| --- |
| Function search() {  Sql <- “SELECT \* FROM kost WHERE nama\_kost LIKE ‘%keyword%’ OR alamat LIKE ‘%keyword%’”  } |

#### Algoritma #7

Nama Kelas : Controller\_kost

Nama Operasi : updateKost()

Algoritma : ALGO007

|  |
| --- |
| Function updateKost() {  Sql <- “UPDATE kost set (‘...’) where id\_kost = ‘...’”  Output(sql)  } |

#### Algoritma #8

Nama Kelas : Controller\_kost

Nama Operasi : addKost()

Algoritma : ALGO008

|  |
| --- |
| Function addKost() {  Sql <- “INSERT INTO KOST VALUES (data)”  } |

#### Perancangan Query

Query :

|  |  |  |
| --- | --- | --- |
| **No Query** | **Query** | **Keterangan** |
| Q-001 | *“SELECT \* FROM kost”* | Menghasilkan hasil query semua kost |
| Q-002 | *“SELECT \* FROM kost WHERE id\_kosan = ‘(“...”)’”* | Menghasilkan hasil query kost berdasarkan id |
| Q-003 | *“SELECT \* FROM kost WHERE nama\_kosan LIKE ‘%(“...”)%’ or alamat LIKE ‘%keyword%’”* | Menghasilkan hasil query kost  Berdasarkan keyword tertentu |
| Q-004 | *“SELECT \* FROM sewa WHERE id\_penyewa = ‘(“...”)’”* | Menghasilkan daftar sewa yang telah dilakukan |
| Q-005 | *“SELECT \* FROM sewa WHERE id\_penyewa = ‘(“...”)’”* | Menghasilkan daftar sewa yang telah dilakukan |
| Q-005 | *“INSERT into kost VALUES(“...”)”* | Memasukkan data kost-an yang dilakukan |
| Q-006 | *“UPDATE kost SET (“...” ) WHERE id\_kost = (“...”) ”* | Mengupdate/edit data kost-an yang sudah ada |
| Q-007 | *Delete kost by id “DELETE FROM* kost *WHERE* id\_kost=’(“...”)’;*”* | Menghapus kost-an tertentu dari database |
| Q-008 | *“SELECT \* FROM kost WHERE category = ‘(“...”)’ ”* | Menghasilkan daftar kost-an berdasarkan kategori tertentu |

Table 1. 30 Perancangan Query

# Matriks Kerunutan (Requirement Traceability Matrix)

|  |  |  |
| --- | --- | --- |
| **Kode FR** | **Nama Functional Requirement** | **Nama Use Case** |
| FR 1 | Menampilkan output berupa detail transaksi | Sewa kost-an |
| FR 2 | Menampilkan output berupa kost-an yang diinput | Cari kost-an |
| FR 3 | Menampilkan output berupa riwayat kost-an | Cek riwayat kost-an |
| FR 4 | Pemilik dapat menambahkan data kost-an | Post kost-an |
| FR 5 | Pemilik dapat mengupdate data kost-an | Update kost-an |
| FR 6 | Pemilik dapat menghapus data kost-an | Delete kost-an |

Table 1. 31 Matriks Kerunutan