

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

<Medgency>

untuk:

Orang yang membutuhkan konsultasi kesehatan

Dipersiapkan oleh:

Muhammad Farrel(1301184453)

Daffashiddiq Nur Awan(1301180311)

Huda Rizky Prasetyo(1301180362)

Haris Subekti(1301183323)

Program Studi Informatika

Fakultas Informatika

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Prodi S1- Informatika**  **Universitas Telkom** | Nomor Dokumen | | Halaman |
| *DPPL-01* | | *59* |
| Revisi | *1* | *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 |  |  |  |  |  |  |  |  |
| Diperiksa oleh |  |  |  |  |  |  |  |  |
| Disetujui oleh |  |  |  |  |  |  |  |  |

Daftar Halaman Perubahan

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

Daftar Isi

[1. Pendahuluan](#_heading=h.30j0zll) **7**

[Tujuan Penulisan Dokumen](#_heading=h.3znysh7) 7

[Lingkup Masalah](#_heading=h.2et92p0) 7

[Definisi dan Istilah](#_heading=h.tyjcwt) 7

[Referensi](#_heading=h.3dy6vkm) 7

[Sistematika Pembahasan](#_heading=h.1t3h5sf) 8

[Deskripsi Perancangan Global](#_heading=h.4d34og8) **9**

[Deskripsi Arsitektural](#_heading=h.2s8eyo1) 9

[Deskripsi Komponen](#_heading=h.17dp8vu) 10

[Perancangan Rinci](#_heading=h.26in1rg) **11**

[Realisasi Use Case](#_heading=h.lnxbz9) 11

[Use Case #1 <cari mitra>](#_heading=h.35nkun2) 11

[Perancangan Antarmuka Usecase #1](#_heading=h.1ksv4uv) 11

[Tabel Identifikasi Antarmuka / Layar / Page](#_heading=h.44sinio) 12

[Identifikasi Object Baru](#_heading=h.2jxsxqh) 12

[Robustness Diagram](#_heading=h.z337ya) 13

[Diagram Kelas](#_heading=h.3j2qqm3) 13

[Sequence Diagram](#_heading=h.2xcytpi) 14

[Use Case #2 <booking>](#_heading=h.1ci93xb) 14

[Perancangan Antarmuka Usecase #2](#_heading=h.3whwml4) 14

[Tabel Identifikasi Antarmuka / Layar / Page](#_heading=h.2bn6wsx) 15

[Identifikasi Object Baru](#_heading=h.qsh70q) 16

[Robustness Diagram](#_heading=h.3as4poj) 16

[Diagram Kelas](#_heading=h.1pxezwc) 17

[Sequence Diagram](#_heading=h.49x2ik5) 18

[3.1.3 Use Case #3 <olah profile>](#_heading=h.4molmooe5ve8) 19

[3.1.3.1 Perancangan Antarmuka Usecase #3](#_heading=h.lox48jixbrnm) 20

[3.1.3.1.1 Tabel Identifikasi Antarmuka / Layar / Page](#_heading=h.ebd1e7gzoiu8) 20

[ID. LAYAR](#_heading=h.22eq3m5kpp7q) 20

[NAMA LAYAR](#_heading=h.22eq3m5kpp7q) 20

[DESKRIPSI](#_heading=h.22eq3m5kpp7q) 20

[page isi/edit profile](#_heading=h.22eq3m5kpp7q) 20

[menampilkan profile saat ini](#_heading=h.22eq3m5kpp7q) 20

[Page isi/edit profile](#_heading=h.22eq3m5kpp7q) 20

[Id\_Objek](#_heading=h.22eq3m5kpp7q) 20

[JENIS](#_heading=h.22eq3m5kpp7q) 20

[LABEL\*](#_heading=h.22eq3m5kpp7q) 20

[Keterangan\*\*](#_heading=h.22eq3m5kpp7q) 20

[edit/isi prifle](#_heading=h.22eq3m5kpp7q) 20

[Button](#_heading=h.22eq3m5kpp7q) 20

[edit profile](#_heading=h.22eq3m5kpp7q) 20

[jika di klik akan menredirect ke form edit profile](#_heading=h.22eq3m5kpp7q) 20

[3.1.3.2 Identifikasi Object Baru](#_heading=h.flo5os2tn4t) 21

[TABEL OBJECT PERANCANGANedi](#_heading=h.22eq3m5kpp7q) 21

[No](#_heading=h.22eq3m5kpp7q) 21

[Nama Object Baru](#_heading=h.22eq3m5kpp7q) 21

[Jenis / Tipe Kelas](#_heading=h.22eq3m5kpp7q) 21

[1](#_heading=h.22eq3m5kpp7q) 21

[homepage user](#_heading=h.22eq3m5kpp7q) 21

[view](#_heading=h.22eq3m5kpp7q) 21

[2](#_heading=h.22eq3m5kpp7q) 21

[form isi profile](#_heading=h.22eq3m5kpp7q) 21

[view](#_heading=h.22eq3m5kpp7q) 21

[3](#_heading=h.22eq3m5kpp7q) 21

[form edit profile](#_heading=h.22eq3m5kpp7q) 21

[view](#_heading=h.22eq3m5kpp7q) 21

[4](#_heading=h.22eq3m5kpp7q) 21

[controller user](#_heading=h.22eq3m5kpp7q) 21

[controller](#_heading=h.22eq3m5kpp7q) 21

[5](#_heading=h.22eq3m5kpp7q) 21

[user](#_heading=h.22eq3m5kpp7q) 21

[model](#_heading=h.22eq3m5kpp7q) 21

[3.1.3.3 Robustness Diagram](#_heading=h.6ng4ceotp1ht) 22

[3.1.3.4 Diagram Kelas](#_heading=h.3v7yg7krjgj6) 23

[3.1.3.5 Sequence Diagram](#_heading=h.pj4uxja8hzc9) 24

[3.1.4 Use Case #4 <input dokter 4>](#_heading=h.nj570ygk8br5) 24

[3.1.4.1 Perancangan Antarmuka Usecase #4](#_heading=h.v3ig0u2t6kyd) 25

[3.1.4.1.1 Tabel Identifikasi Antarmuka / Layar / Page](#_heading=h.k75610k6d2fc) 25

[3.1.4.2 Identifikasi Object Baru](#_heading=h.aumrcf85aq9j) 26

[3.1.4.3 Robustness Diagram](#_heading=h.mvbv5dk5a6l) 27

[3.1.4.4 Diagram Kelas](#_heading=h.yvht1bimub2o) 28

[3.1.4.5 Sequence Diagram](#_heading=h.z7ls0xkginp8) 28

[3.1.5 Use Case #5 <hapus dokter 5>](#_heading=h.7fhmtsmjdl96) 29

[3.1.5.1 Perancangan Antarmuka Usecase #5](#_heading=h.yf1uby8p35ej) 29

[3.1.5.1.1 Tabel Identifikasi Antarmuka / Layar / Page](#_heading=h.9w0heop8om2b) 30

[3.1.5.2 Identifikasi Object Baru](#_heading=h.zi1hnwrnds6s) 31

[3.1.5.3 Robustness Diagram](#_heading=h.pwhwrntv7vmd) 31

[3.1.5.4 Diagram Kelas](#_heading=h.nvr59c19h91p) 32

[3.1.5.5 Sequence Diagram](#_heading=h.34yoeq34vxq0) 33

[3.1.6 Use Case #6 <lihat riwayat 6>](#_heading=h.7vq4d9su35oh) 34

[3.1.6.1 Perancangan Antarmuka Usecase #6](#_heading=h.grk17dht70wl) 34

[3.1.6.1.1 Tabel Identifikasi Antarmuka / Layar / Page](#_heading=h.5qzubrx0ry92) 35

[3.1.6.2 Identifikasi Object Baru](#_heading=h.7ki82vqsi0e4) 36

[3.1.6.3 Robustness Diagram](#_heading=h.4pfi9se2v6js) 36

[3.1.6.4 Diagram Kelas](#_heading=h.3b0gcldgc9yp) 37

[3.1.6.5 Sequence Diagram](#_heading=h.wyb0ukcjh0ae) 38

[3.1.7 Use Case #7 <hapus riwayat 7>](#_heading=h.hjx7xfbot1vg) 38

[3.1.7.1 Perancangan Antarmuka Usecase #7](#_heading=h.kd0rhqk1yo5x) 39

[3.1.7.1.1 Tabel Identifikasi Antarmuka / Layar / Page](#_heading=h.ak6riwllpoee) 39

[3.1.7.2 Identifikasi Object Baru](#_heading=h.ndcespcz9t65) 40

[3.1.7.3 Robustness Diagram](#_heading=h.l6h8u3qf6i2i) 40

[3.1.7.4 Diagram Kelas](#_heading=h.g8lp6trflg0) 41

[3.1.7.5 Sequence Diagram](#_heading=h.5btcxsorzh8x) 41

[Perancangan Detil](#_heading=h.2p2csry) **42**

[Perancangan Detil Kelas](#_heading=h.147n2zr) 42

[Perancangan Kelas Persistensi (\*\*\*Basis Data Skema Tabel)](#_heading=h.3o7alnk) 43

[Perancangan Algoritma](#_heading=h.ihv636) 44

[Algoritma cariMitra](#_heading=h.32hioqz) 44

[Algoritma Booking](#_heading=h.2obhcpcxzk20) 45

[Algoritma editProfile](#_heading=h.4awqsqldnky5) 45

[Algoritma isiProfile](#_heading=h.u0r9hrsqjfu7) 45

[Algoritma inputDokter](#_heading=h.1drv5aeohvwe) 45

[Algoritma hapusDokter](#_heading=h.n9zoinxt2atv) 45

[Algoritma lihatBooking](#_heading=h.hne9uzt8kwr0) 46

[Algoritma hapusBooking](#_heading=h.o3lx94ybbm6n) 46

[Algoritma showProfile](#_heading=h.og4h2j1e07a7) 46

[Perancangan Query](#_heading=h.1hmsyys) 46

[Matriks Kerunutan (Requirement Traceability Matrix)](#_heading=h.41mghml) **47**

# 

# 1. Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen DPPL-APG ini berisi Deskripsi Perancangan Perangkat Lunak (DPPL) yang

memiliki tujuan untuk memberikan penjelasan mengenai perangkat lunak yang akan dibangun baik berupa gambaran umum maupun penjelasan detail dan menyeluruh.

Tujuan dari kami membuat dokumen ini adalah untuk menjelaskan sebuah sistem layanan kesehatan berbasis web yang bernama Medgency. Dokumen ini akan digunakan sebagai bahan referensi dalam proses pengembangan dan sebagai bahan evaluasi pada setiap tahap pengembangan. Dengan adanya dokumen DPPL ini kami berharap agar pengembangan perangkat lunak akan lebih terarah dan tidak menimbulkan pertanyaan bagi pengembang sistem.

## Lingkup Masalah

Medgency merupakan perangkat lunak berbasis web yang memudahkan user mendapatkan layanan kesehatan karena terdapat beberapa tingkat penyakit yang membutuhkan penanganan yang cepat maka dari itu kami membuat terobosan untuk memberikan pelayanan kesehatan yang cepat , target user dari Medgency adalah masyarakat yang membutuhkan layanan kesehatan, kami membuat medgency dikarenakan sistem penanganan pasien yang ada di rumah sakit atau klinik tidak efisien. User dari perangkat lunak kami adalah mereka yang membutuhkan pelayanan kesehatan secara cepat, terdapat juga mitra dimana mitra tersebut adalah pelayanan kesehatan yang berkerja sama dengan kami dan terdapat juga admin yang berfungsi sebagai mengatur akun mitra dan akun customer yang ada pada perangkat lunak kami.

## Definisi dan Istilah

• DPPL adalah Deskripsi Perancangan Perangkat Lunak Merupakan dokumen yang

berisi deskripsi perancangan perangkat lunak.

• SKPL adalah Spesifikasi Kebutuhan Perangkat Lunak, atau dalam bahasa Inggrisnya

sering juga disebut sebagai Software Requirements Spesification (SRS), dan merupakan

spesifikasi dari perangkat lunak yang akan dikembangkan.

• ERD adalah Entity Relationship Diagram, diagram dan notasi yang digunakan untuk

merepresentasikan struktur data statis pada perangkat lunak.

• UML adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap

sebuah sistem atau perangkat lunak berbasis objek.

• IEEE adalah Institute of Electrical and Electronics Engineering Standar internasional

untuk pengembangan dan perancangan produk.

• OOP (Object Oriented Programming) adalah jenis pemrograman dengan metode

enkapsulasi dengan membuat kelas-kelas dalam pendefinisian suatu kasus.

## Referensi

1) Panduan Penggunaan dan Pengisian Spesifikasi Perangkat Lunak (SKPL), Jurusan Teknik Informatika, 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 global. Dimana pada bab ini menjelaskan mengenai

deskripsi tentang arsitektural dan komponen sistem.

▪ Bab 3 adalah penjelasan rinci dari use case diagram yang telah dibuat pada SKKPL berisi perancangan antara muka manusia yang terdiri dari beberapa sub bab

yakni, gambaran umum antar muka pengguna, identifikasi objek baru, identifikasi kelas diagram, robustness diagram, sequence diagram.

▪ Bab 4 adalah perancangan detil yang terdiri dari perancangan detil dari kelas tersebut, perancangan kelas persistensi, perancangan algoritma, perancangan query

▪ Bab 5 adalah matriks kerunutan adalah mapping requirement dengan usecase yang direalisasikan

# Deskripsi Perancangan Global

## Deskripsi Arsitektural

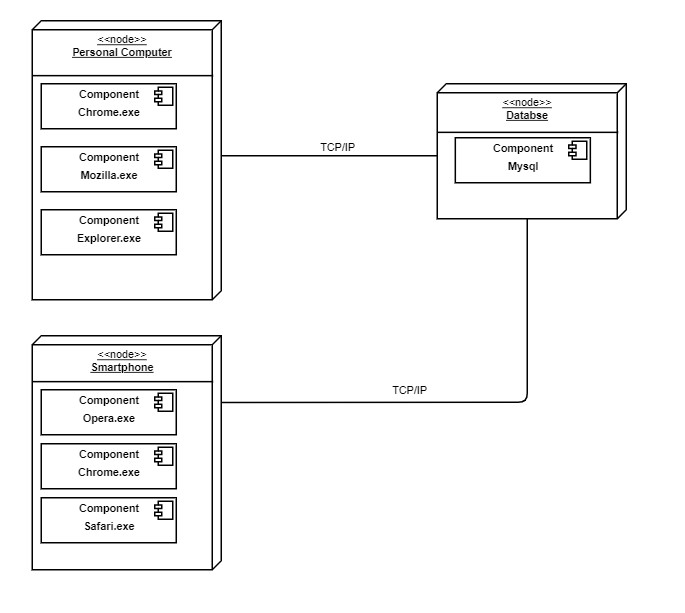
• Sistem operasi : Windows 10, macOS Catalina

• Development Tools : VSCode, XAMPP, PhpMyAdmin(database)

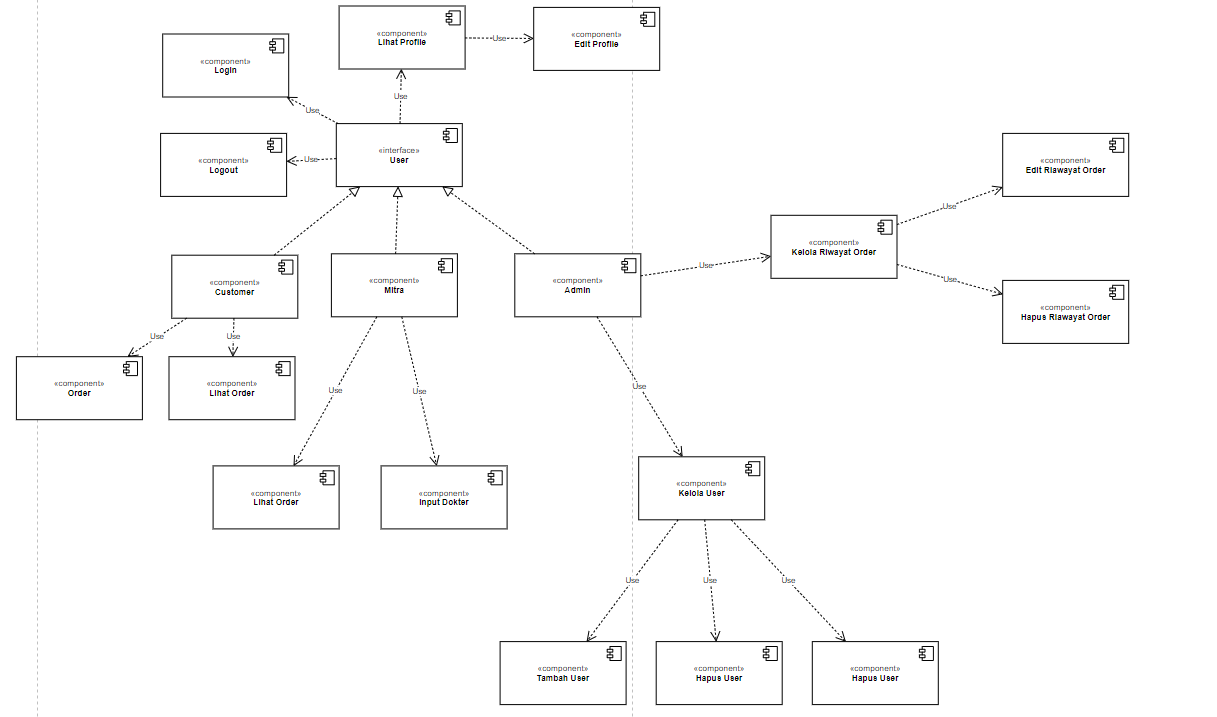
• Bahasa Pemrograman : php, javascript

• Database management : MySQL

• Deployment Diagram



## Deskripsi Komponen

******

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Komponen** | **Keterangan** |
| 1. | Customer | Customer dapat melihat profil, edit profil, login, order, lihat order |
| 2. | Mitra | Mitra dapat melihat profil, edit profil, login,lihat order, input dokter |
| 3. | Admin | Admin dapat melakukan login, lihat profile, edit profile, kelola riwayat order, kelola user |

# Perancangan Rinci

## Realisasi Use Case

|  |  |  |
| --- | --- | --- |
| No | Nama Use Case | Deskripsi Use Case |
| #1 | Cari mitra | customer melakukan pencarian terhadap mitra |
| #2 | Booking | customer melakukan proses booking jadwal dokter |
| #3 | Olah profil | melakukan yang edit/input terhadap profile user |
| #4 | input dokter | mitra melakukan input dokter |
| #5 | hapus dokter | mitra melakukan penghapusan data dokter |
| #6 | lihat pembookingan | mitra melihat pembookingan yang sedang terjadi atau yang telah selesai yang terjadi pada mitra tersebut |
| #7 | hapus pembookingan | menghapus pembookingan yang statusnya telah selesai atau dibatalkan |

### Use Case #1 <cari mitra>

*Skenario Use Case #1 :*

*Precondition: customer melakukan login*

*Primary Flow :*

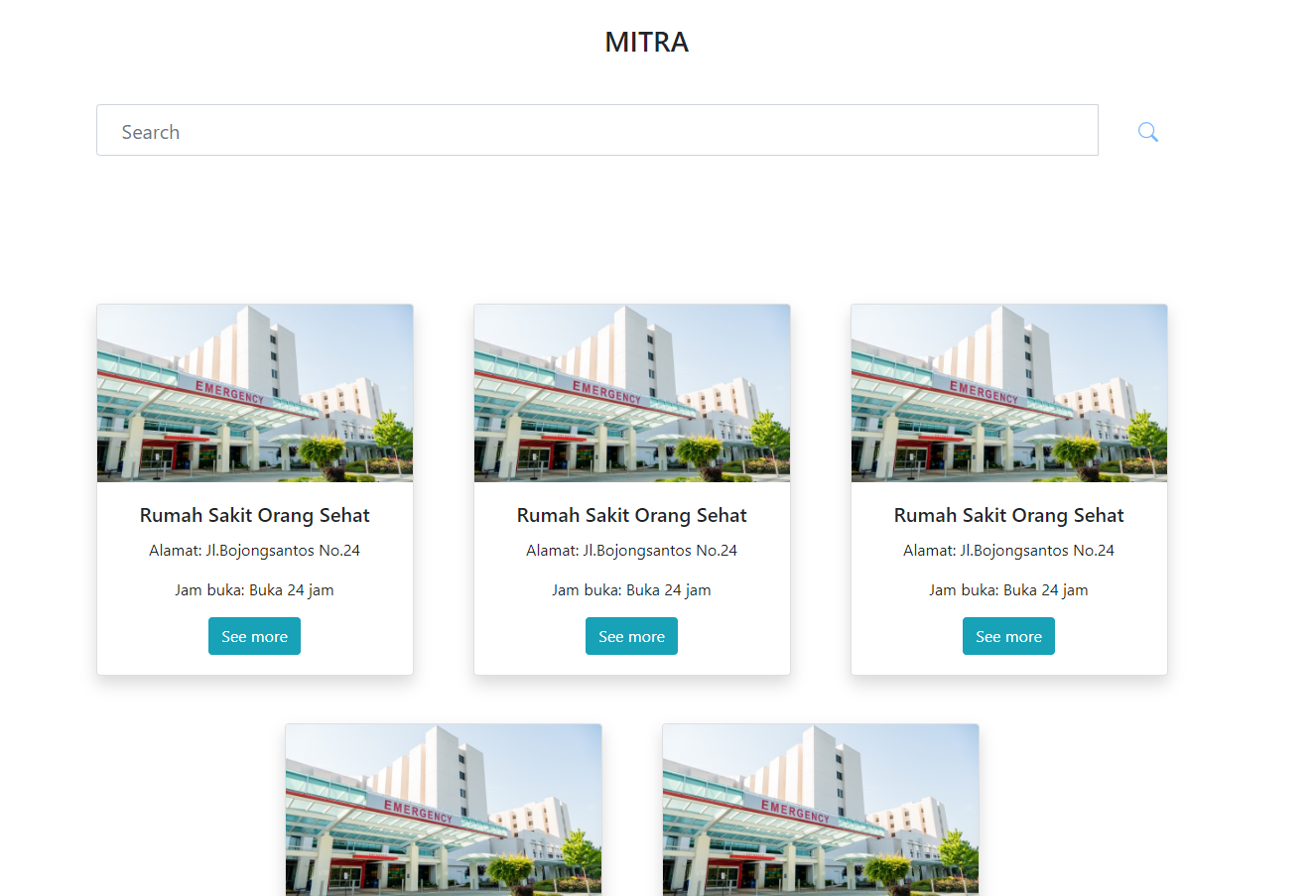
* *sistem menampilkan home customer*
* *customer melakukan search mitra*
* *system menampilkan mitra yang dicari*

*Alternate Flow :*

* *sistem menampilkan home customer*
* *customer melakukan search mitra*
* *system tidak dapat menemukan mitra yang dicari*

*Postcondition: system menampilkan mitra yang dicari oleh customer*

#### Perancangan Antarmuka Usecase #1

**

#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
| *1* | *homepage user* | *Merupakan tampilan utama user dimana mitra ditampilakn* |

*Page search mitra*

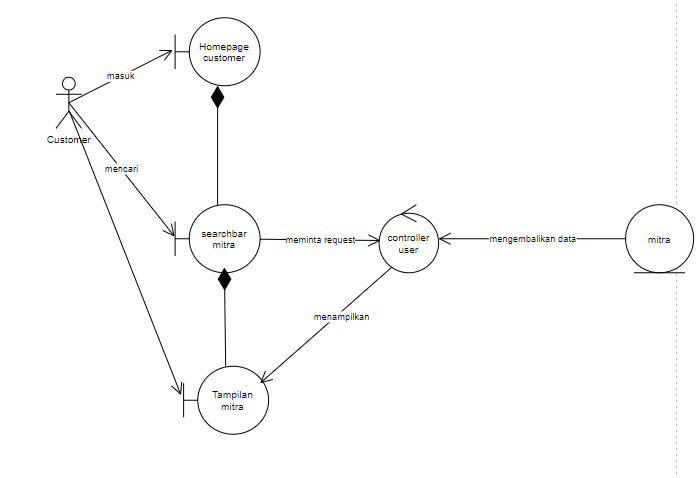
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| *Button search* | *Button* | *icon search* | *Jika diklik, akan mengaktifkan function search mitra* |
| *Button see more* | *Button* | *see more* | *Jika diklik akan mendirect user menuju page mitra* |
| *searchbar* | *input* | *search* | *menerima inputan dari user* |
|  |  |  |  |

#### Identifikasi Object Baru

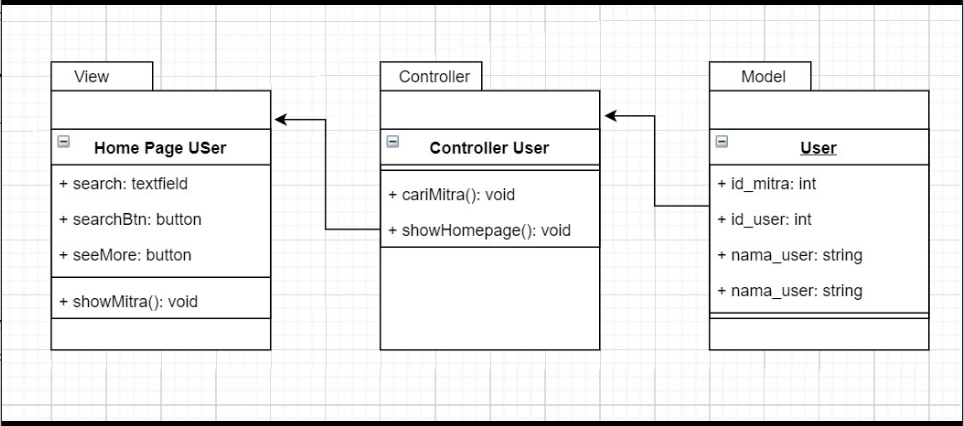
*TABEL OBJECT PERANCANGAN*

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1* | *home page user* | *view* |
| *2* | *controller user* | *controller* |
| *3* | *user* | *model* |

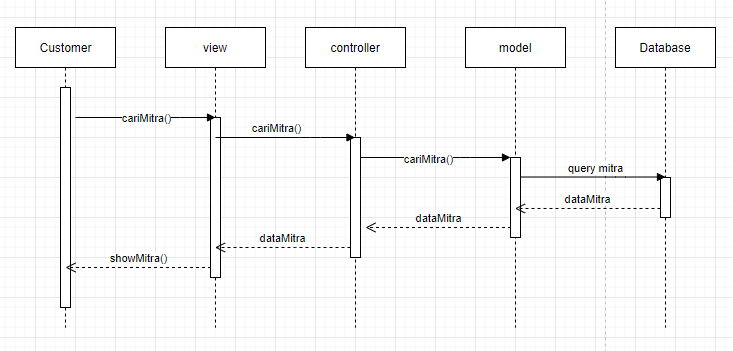
#### Robustness Diagram

**

#### Diagram Kelas



#### Sequence Diagram

**

### Use Case #2 <booking>

*Skenario Use Case #2*

*Precondition: customer melakukan login*

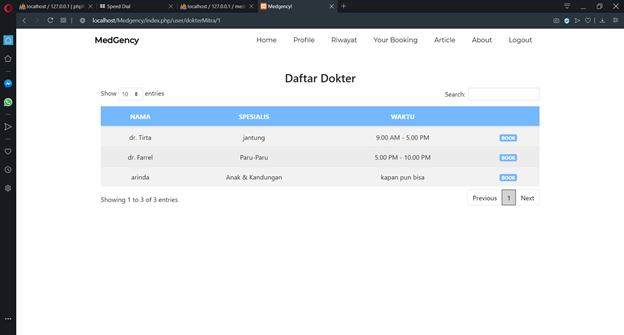
*Primary Flow:*

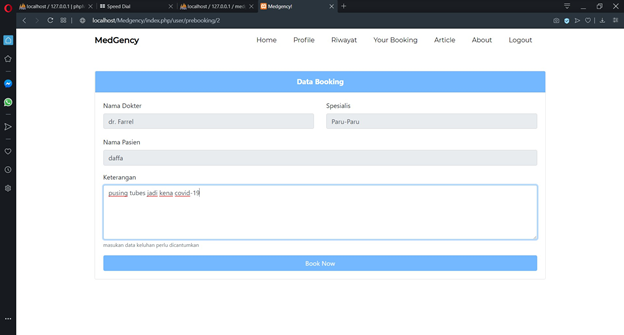
* *system menampilkan home user*
* *customer menentukan mitra mana yang akan dipilih*
* *system menampilkan data dokter dalam bentuk tabel*
* *customer memilih salah satu dokter yang terdapat pada tabel*
* *customer menekan tombol book*
* *system menampilkan form pembookingan*
* *customer mengisi form dan submit*
* *system akan mengirimkan kode booking dan bukti pembayaran janji temu*

*Alternate Flow*

*Postcodition: customer berhasil melakukan proses booking dan data booking akan dimasukan ke dalam database*

#### Perancangan Antarmuka Usecase #2

**

**

#### Tabel Identifikasi Antarmuka / Layar / Page

|  |  |  |
| --- | --- | --- |
| **ID. LAYAR** | **NAMA LAYAR** | **DESKRIPSI** |
|  | *table dokter* | *menampilkan data dokter dalam bentuk table* |
|  | *form book* | *mengisikan data booking* |

*Page table dokter*

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| *Button book* | *Button* | *book* | *jika di klik akan membuka form book* |
| *searchbar* | *input* | *search* | *menerima input dari user untuk ditampilkan* |
| *pagination* | *pagination* | *pagination* | *menunjukan terdapat berapa halaman data* |

*Page form book*

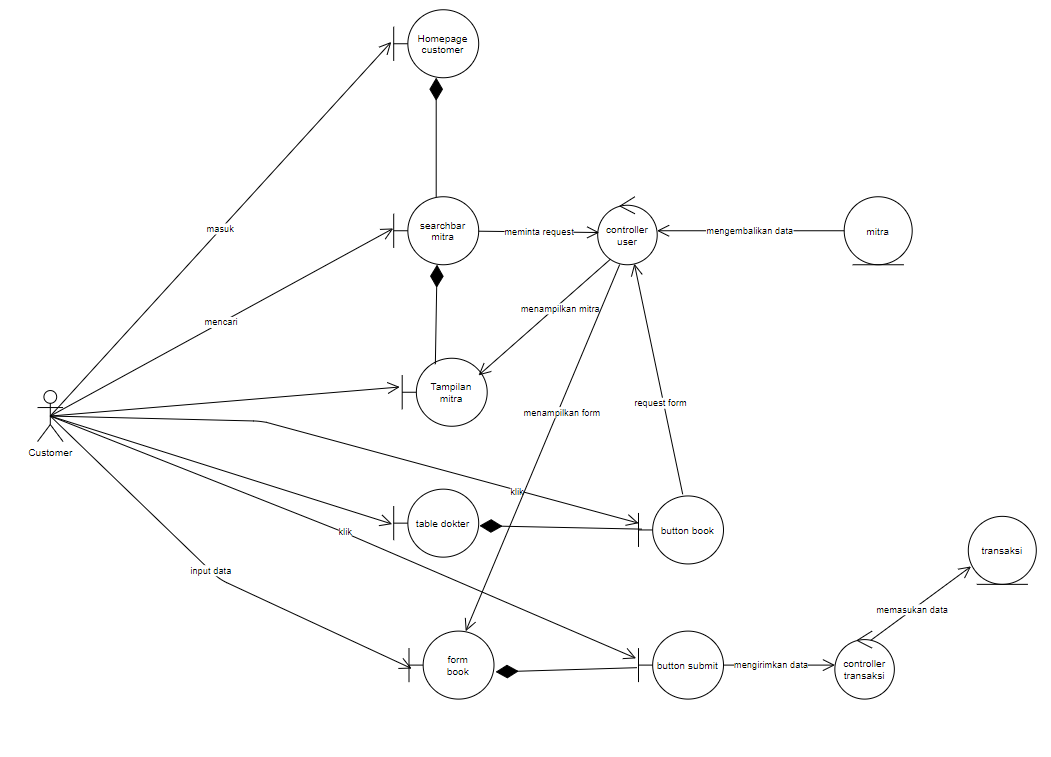
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **JENIS** | **LABEL\*** | **Keterangan\*\*** |
| *input nama dokter* | *input* | *nama dokter* | *menampilkan nama dokter yang di book* |
| *input spesialis dokter* | *input* | *spesialis dokter* | *menampilkan spesialis dokter* |
| *input nama pasien* | *input* | *nama pasien* | *input nama pasien* |
| *text area keterangan* | *text area* | *text area* | *input keterangan yang dibutuhkan* |

#### Identifikasi Object Baru

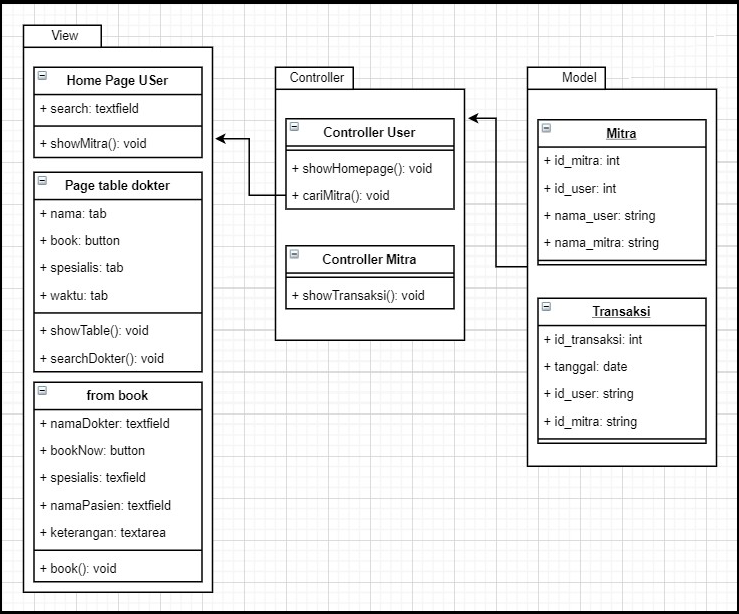
TABEL OBJECT PERANCANGAN Mitra

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Object Baru** | **Jenis / Tipe Kelas** |
| *1* | *home page user* | *view* |
| *2* | *page table dokter* | *view* |
| *3* | *form book* | *view* |
| *4* | *controller user* | *controller* |
| *5* | *controller transaksi* | *controller* |
| *6* | *mitra* | *model* |
| *7* | *transaksi* | *model* |

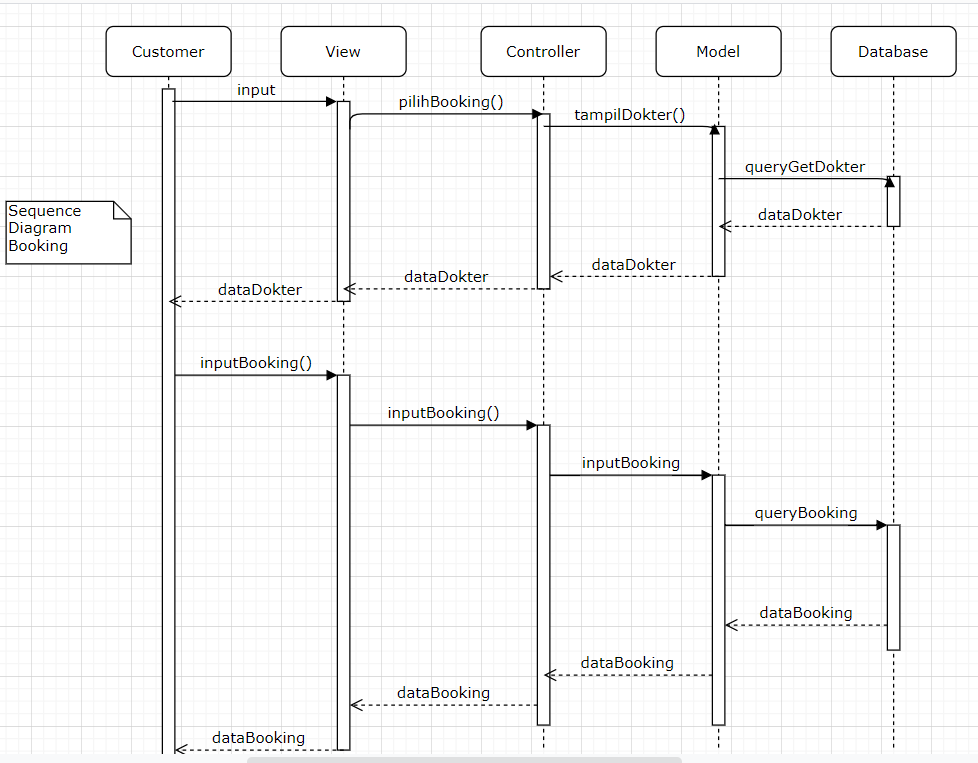
#### Robustness Diagram

**

#### Diagram Kelas

**

#### Sequence Diagram

**

### *3.1.3 Use Case #3 <olah profile>*

*Skenario Use Case #3*

*Precondition: customer melakukan login*

*Primary Flow:*

* *system menampilkan home user*
* *customer memilih ikon user*
* *system menampilkan edit profile dan isi profile*

*Alternate Flow*

*Postconditon: customer mengupdate profule*

#### *3.1.3.1 Perancangan Antarmuka Usecase #3*

#### 

#### 

#### *3.1.3.1.1 Tabel Identifikasi Antarmuka / Layar / Page*

|  |  |  |
| --- | --- | --- |
| *ID. LAYAR* | *NAMA LAYAR* | *DESKRIPSI* |
|  | *page isi/edit profile* | *menampilkan profile saat ini* |

#### 

#### 

#### 

#### *Page isi/edit profile*

|  |  |  |  |
| --- | --- | --- | --- |
| *Id\_Objek* | *JENIS* | *LABEL\** | *Keterangan\*\** |
| *edit/isi prifle* | *Button* | *edit profile* | *jika di klik akan menredirect ke form edit profile* |

#### 

#### *3.1.3.2 Identifikasi Object Baru*

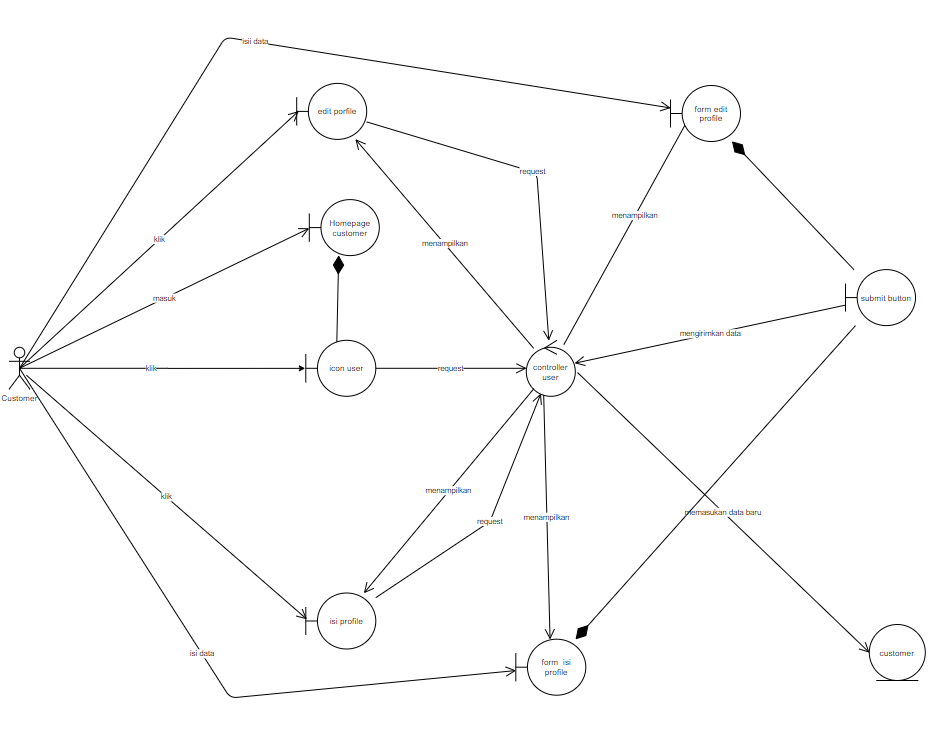
#### *TABEL OBJECT PERANCANGANedi*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Object Baru* | *Jenis / Tipe Kelas* |
| *1* | *homepage user* | *view* |
| *2* | *form isi profile* | *view* |
| *3* | *form edit profile* | *view* |
| *4* | *controller user* | *controller* |
| *5* | *user* | *model* |

#### 

#### *3.1.3.3 Robustness Diagram*

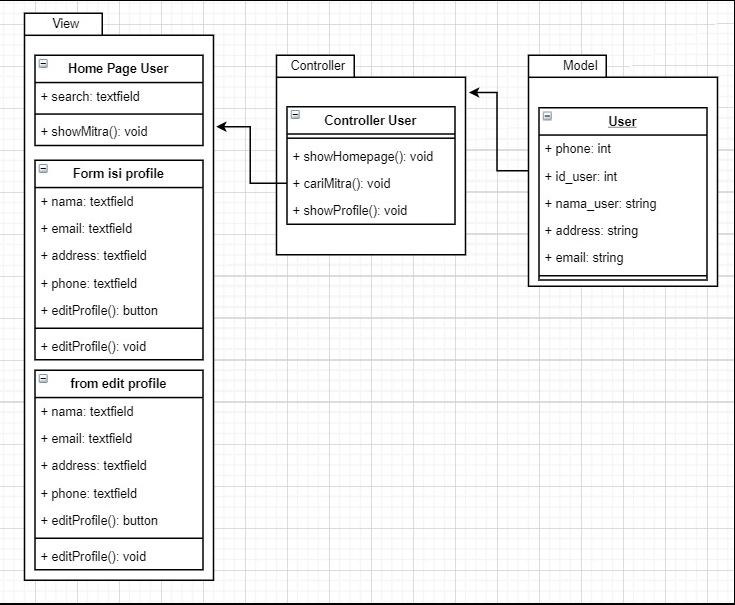
#### 



#### 

#### *3.1.3.4 Diagram Kelas*

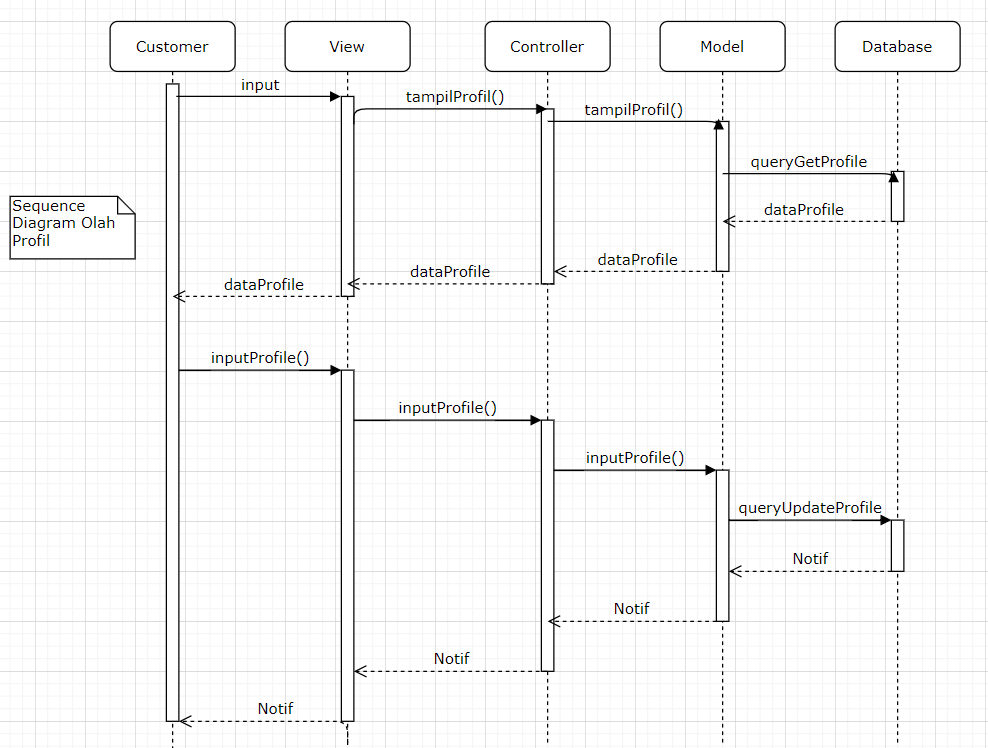
#### 



#### 

#### *3.1.3.5 Sequence Diagram*

#### 



### *3.1.4 Use Case #4 <input dokter 4>*

*Skenario Use Case #4*

*Precondition: mitra melakukan login*

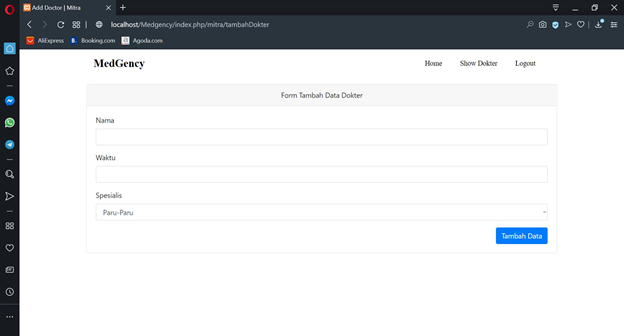
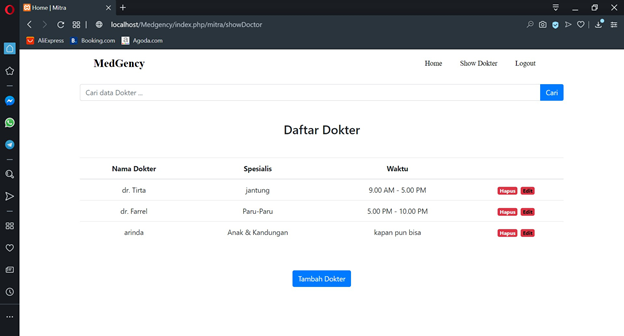
*Primary Flow:*

* *system menampilkan home mitra*
* *mitra memilih menu input dokter*
* *system menampilkan form isi data dokter baru*
* *mitra menginput data dokter yang baru*
* *mitra menekan tonbol submit*
* *data dokter yang baru akan dimasukan ke dalam database*

*Alternate Flow*

*postcondition: mitra memasukan data dokter kedalam database*

#### *3.1.4.1 Perancangan Antarmuka Usecase #4*

**

#### *3.1.4.1.1 Tabel Identifikasi Antarmuka / Layar / Page*

|  |  |  |
| --- | --- | --- |
| ***ID. LAYAR*** | ***NAMA LAYAR*** | ***DESKRIPSI*** |
|  | *table dokter* | *menampilkan data dokter* |
|  | *form tambah dokter* | *megnisikan data dokter yang baru* |

*Page table dokter*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Id\_Objek*** | ***JENIS*** | ***LABEL\**** | ***Keterangan\*\**** |
| *searchbar dokter* | *input* | *search* | *mencari data dokter yang diinginkan* |
| *button search* | *Button* | *search button* | *jika di klik akan mencari data dokter sesuai dengan yang ada pada searchbar* |
| *badge hapus* | *badge* | *badge hapus* | *menghapus data dokter* |
| *badge edit* | *badge* | *badge edit* | *edit data dokter* |
| *tambah dokter* | *button* | *tambah dokter* | *jika diklik akan mendirect ke form dokter* |

*Page form tambah dokter*

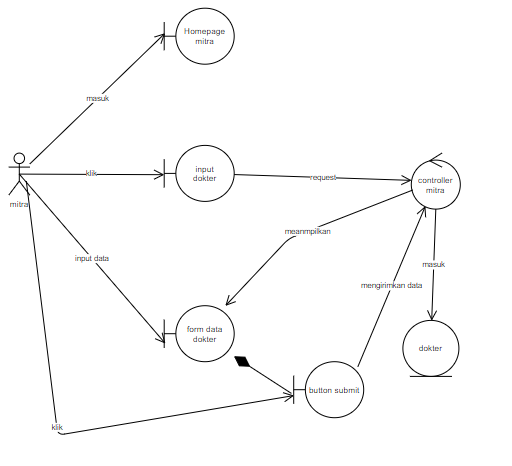
|  |  |  |  |
| --- | --- | --- | --- |
| ***Id\_Objek*** | ***JENIS*** | ***LABEL\**** | ***Keterangan\*\**** |
| *input nama* | *input* | *nama* | *input nama dokter* |
| *input waktu* | *input* | *waktu* | *input jadwal dokter* |
| *input spesialis* | *input* | *spesialis* | *input spesialis* |
| *button tambah data* | *button* | *tambah data* | *mensubmit data yang sudah di isi ke dalam database dokter* |

#### *3.1.4.2 Identifikasi Object Baru*

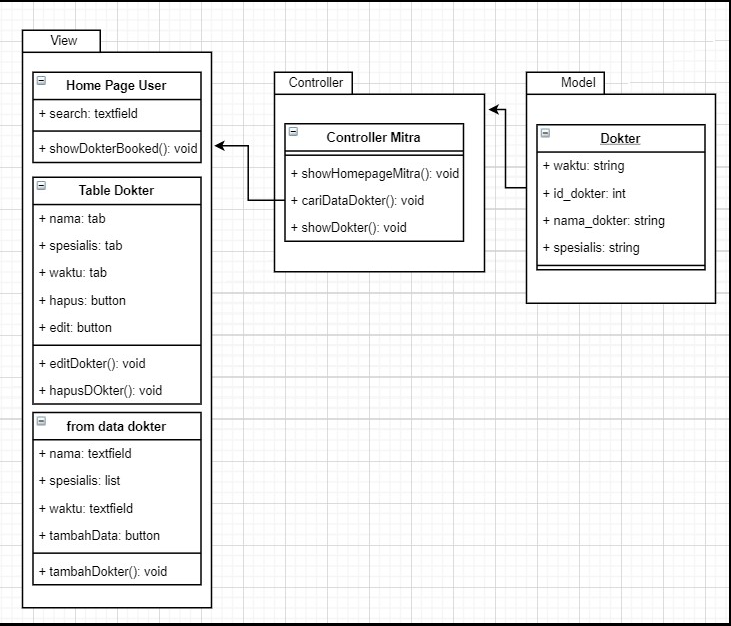
*TABEL OBJECT PERANCANGAN*

|  |  |  |
| --- | --- | --- |
| ***No*** | ***Nama Object Baru*** | ***Jenis / Tipe Kelas*** |
| *1* | *homepage mitra* | *view* |
| *2* | *table dokter* | *view* |
| *3* | *form data dokter* | *view* |
| *4* | *controller mitra* | *controller* |
| *5* | *dokter* | *model* |

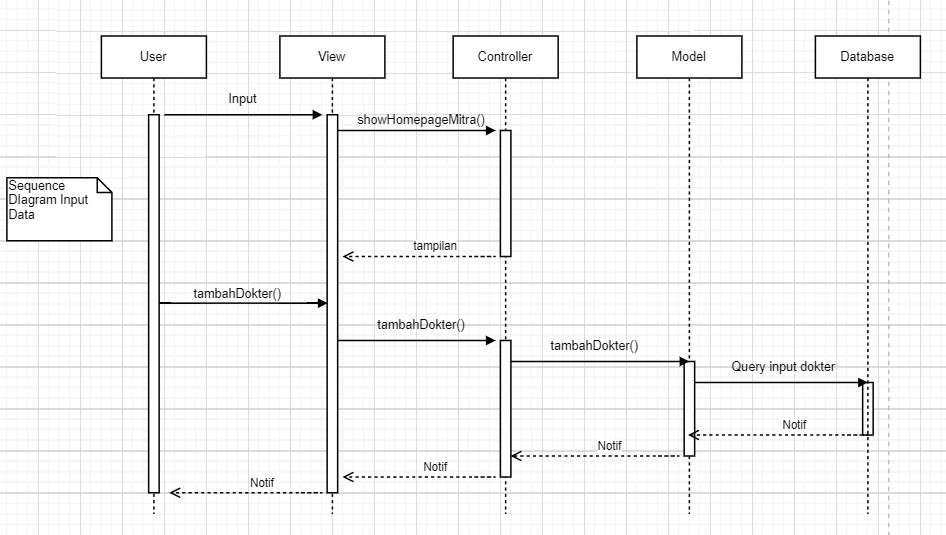
#### *3.1.4.3 Robustness Diagram*

**

#### *3.1.4.4 Diagram Kelas*

**

#### *3.1.4.5 Sequence Diagram*

**

### *3.1.5 Use Case #5 <hapus dokter 5>*

*Skenario Use Case #5*

*Precondition: mitra melakukan login*

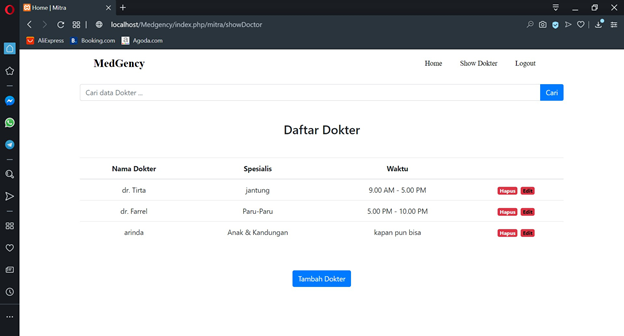
*Primary Flow:*

* *system menampilkan home mitra*
* *mitra memilih menu hapus dokter*
* *sustem menampilkan table dokter*
* *mitra memilih dokter mana yang akan di hapus dan menekan tombol hapus*
* *system menghapus data dokter dari database*

*Alternate Flow*

*postcondition: mitra memenghapus data dokter*

#### *3.1.5.1 Perancangan Antarmuka Usecase #5*

**

#### *3.1.5.1.1 Tabel Identifikasi Antarmuka / Layar / Page*

|  |  |  |
| --- | --- | --- |
| ***ID. LAYAR*** | ***NAMA LAYAR*** | ***DESKRIPSI*** |
|  | *table dokter* | *menampilkan data dokter* |

*Page table dokter*

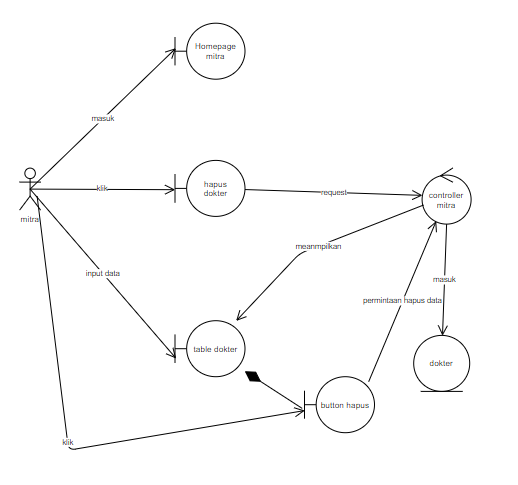
|  |  |  |  |
| --- | --- | --- | --- |
| ***Id\_Objek*** | ***JENIS*** | ***LABEL\**** | ***Keterangan\*\**** |
| *searchbar dokter* | *input* | *search* | *mencari data dokter yang diinginkan* |
| *button search* | *Button* | *search button* | *jika di klik akan mencari data dokter sesuai dengan yang ada pada searchbar* |
| *badge hapus* | *badge* | *badge hapus* | *menghapus data dokter* |
| *badge edit* | *badge* | *badge edit* | *edit data dokter* |
| *tambah dokter* | *button* | *tambah dokter* | *jika diklik akan mendirect ke form dokter* |

#### *3.1.5.2 Identifikasi Object Baru*

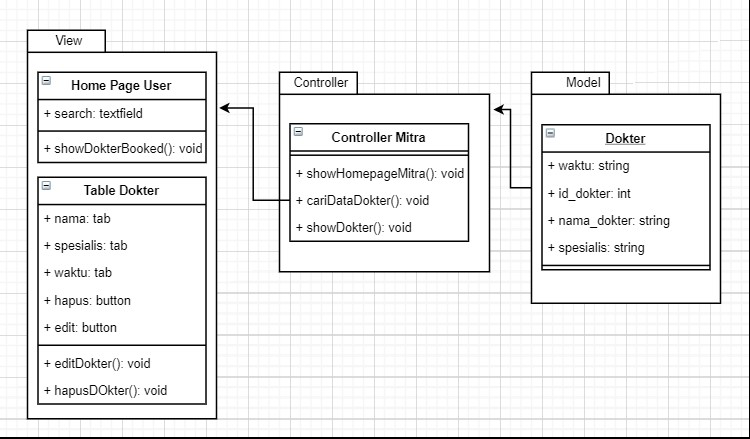
*TABEL OBJECT PERANCANGAN*

|  |  |  |
| --- | --- | --- |
| ***No*** | ***Nama Object Baru*** | ***Jenis / Tipe Kelas*** |
| *1* | *homepage mitra* | *view* |
| *2* | *table dokter* | *view* |
| *3* | *controller mitra* | *controller* |
| *4* | *dokter* | *model* |

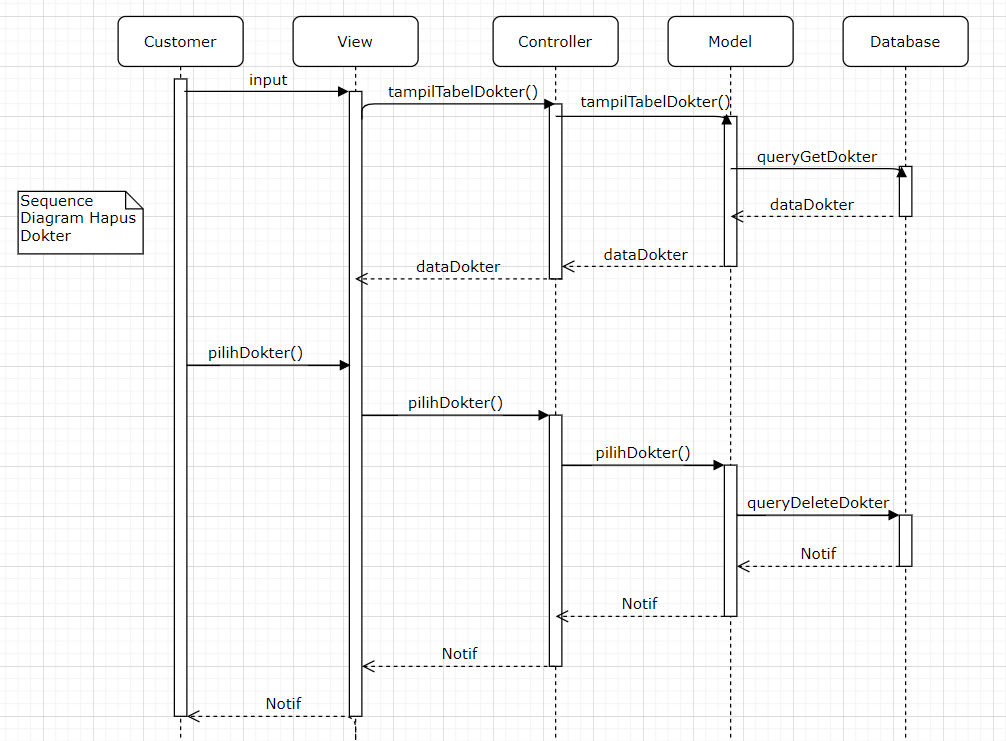
#### *3.1.5.3 Robustness Diagram*

**

#### *3.1.5.4 Diagram Kelas*

**

#### *3.1.5.5 Sequence Diagram*

**

### *3.1.6 Use Case #6 <lihat riwayat 6>*

*Skenario Use Case #6*

*Precondition: mitra melakukan login*

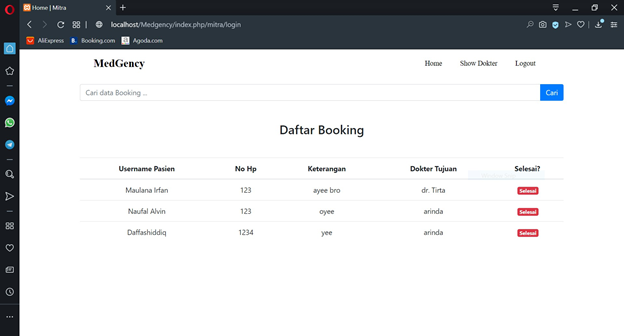
*Primary Flow:*

* *system menampilkan home mitra*
* *mitra memilih menu lihat riwayat*
* *system mengambil data riwayat dari database*
* *system menampilkan riwayat dalam bentuk table*

*Alternate Flow*

*postcondition: mitra dapat melihat riwayat*

#### *3.1.6.1 Perancangan Antarmuka Usecase #6*

**

#### *3.1.6.1.1 Tabel Identifikasi Antarmuka / Layar / Page*

|  |  |  |
| --- | --- | --- |
| ***ID. LAYAR*** | ***NAMA LAYAR*** | ***DESKRIPSI*** |
|  | *table booking* | *menampilkan data booking pada suatu mitra* |

*Page table booking*

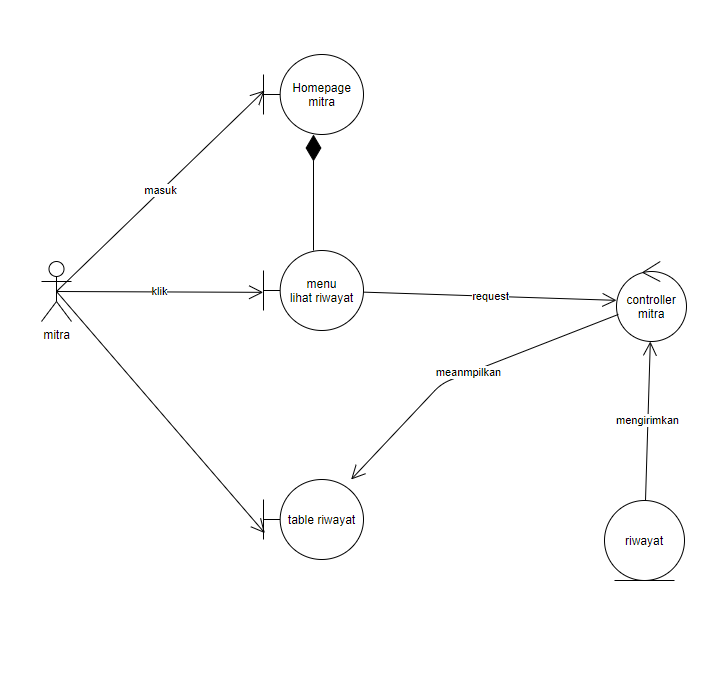
|  |  |  |  |
| --- | --- | --- | --- |
| ***Id\_Objek*** | ***JENIS*** | ***LABEL\**** | ***Keterangan\*\**** |
| *search bar riwayat* | *searchbar* | *search bar riwayat* | *menerima input yang dicari user* |
| *button search* | *Button* | *search button* | *jika di klik akan mengaktifkan fungsi search* |
| *badge selesai* | *badge* | *selesai* | *menuntaskan book atau menghapus* |

#### *3.1.6.2 Identifikasi Object Baru*

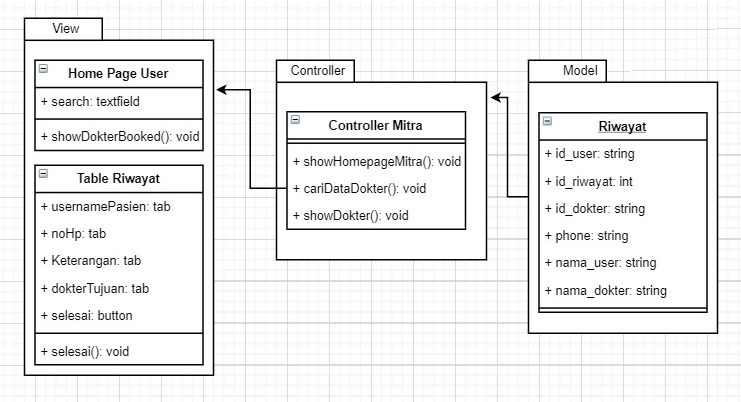
*TABEL OBJECT PERANCANGAN*

|  |  |  |
| --- | --- | --- |
| ***No*** | ***Nama Object Baru*** | ***Jenis / Tipe Kelas*** |
| *1* | *homepage mitra* | *view* |
| *2* | *table riwayat* | *view* |
| *3* | *controller mitra* | *controller* |
| *4* | *riwayat* | *model* |

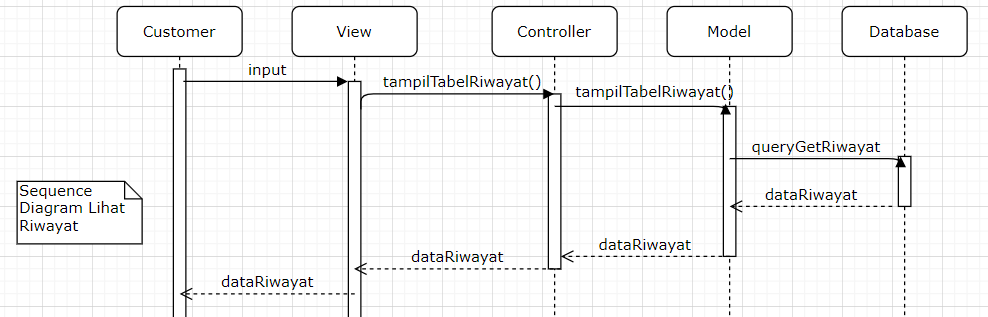
#### *3.1.6.3 Robustness Diagram*

**

#### *3.1.6.4 Diagram Kelas*

**

#### *3.1.6.5 Sequence Diagram*

**

### *3.1.7 Use Case #7 <hapus riwayat 7>*

*Skenario Use Case #7*

*Precondition: mitra melakukan login*

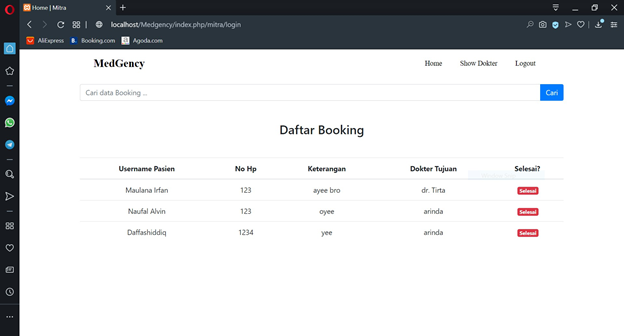
*Primary Flow:*

* *system menampilkan home mitra*
* *mitra memilih menu lihat riwayat*
* *system mengambil data riwayat dari database*
* *system menampilkan riwayat dalam bentuk table*
* *mitra menekan tombol hapus pada salah satu riwayat*

*Alternate Flow*

*postcondition: mitra memenghapus data riwayat*

#### *3.1.7.1 Perancangan Antarmuka Usecase #7*

**

#### *3.1.7.1.1 Tabel Identifikasi Antarmuka / Layar / Page*

|  |  |  |
| --- | --- | --- |
| ***ID. LAYAR*** | ***NAMA LAYAR*** | ***DESKRIPSI*** |
|  | *table riwayat* | *menampilkan data riwayat* |

*Page table riwayat*

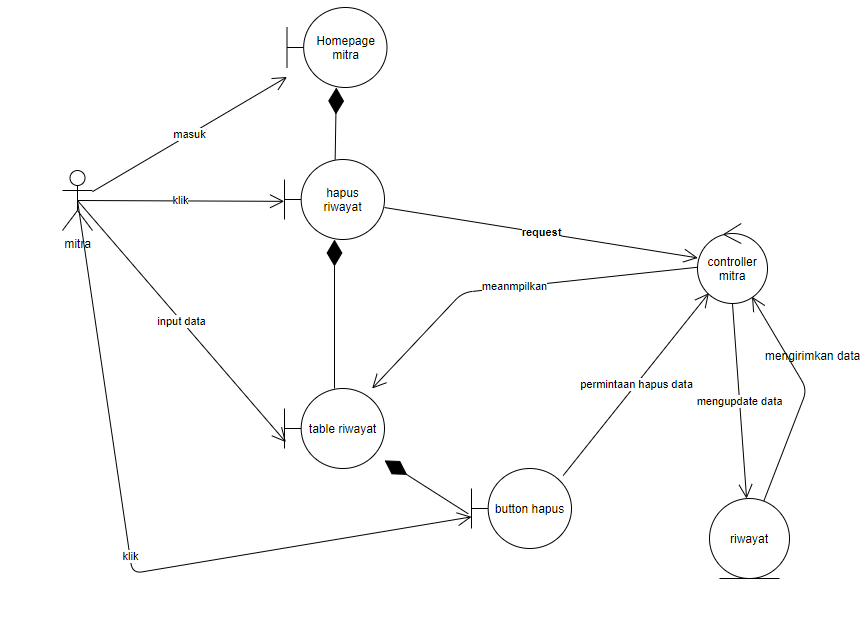
|  |  |  |  |
| --- | --- | --- | --- |
| ***Id\_Objek*** | ***JENIS*** | ***LABEL\**** | ***Keterangan\*\**** |
| *search bar riwayat* | *searchbar* | *search bar riwayat* | *menerima input yang dicari user* |
| *button search* | *Button* | *search button* | *jika di klik akan mengaktifkan fungsi search* |
| *badge selesai* | *badge* | *selesai* | *menuntaskan book atau menghapus* |

#### *3.1.7.2 Identifikasi Object Baru*

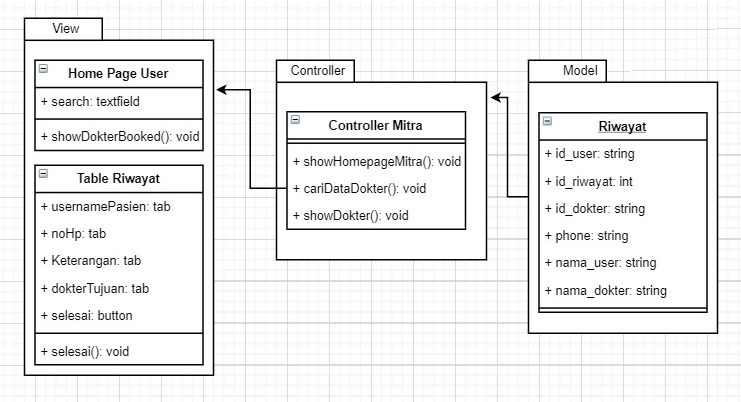
*TABEL OBJECT PERANCANGAN*

|  |  |  |
| --- | --- | --- |
| ***No*** | ***Nama Object Baru*** | ***Jenis / Tipe Kelas*** |
| *1* | *homepage mitra* | *view* |
| *2* | *table riwayat* | *view* |
| *3* | *controller mitra* | *controller* |
| *4* | *riwayat* | *model* |

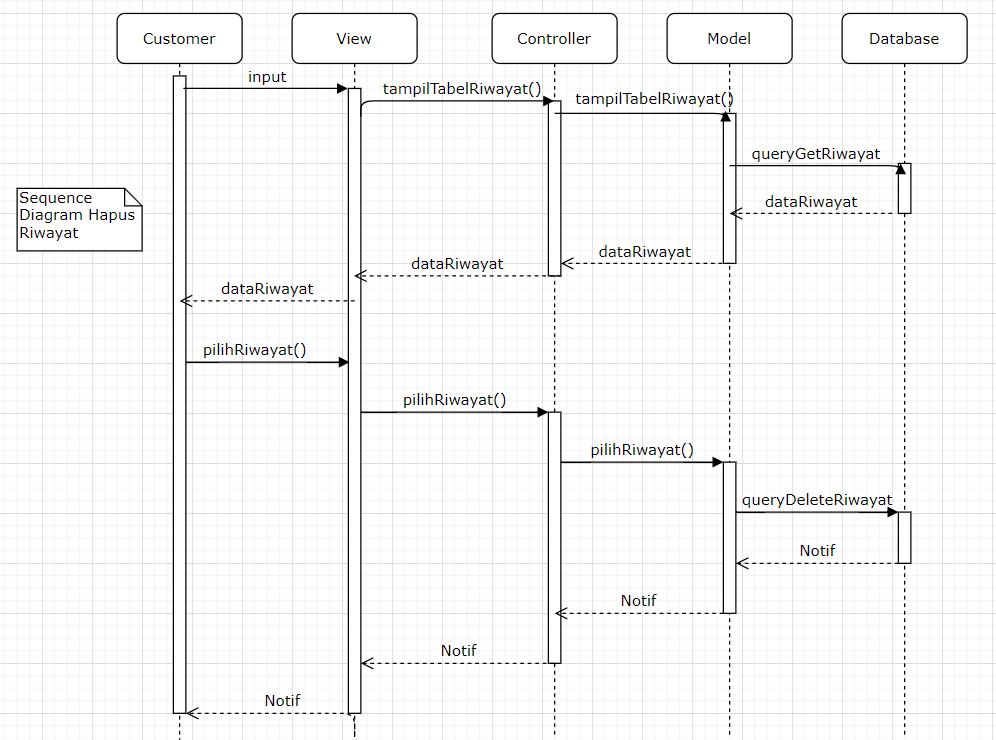
#### *3.1.7.3 Robustness Diagram*

**

#### *3.1.7.4 Diagram Kelas*

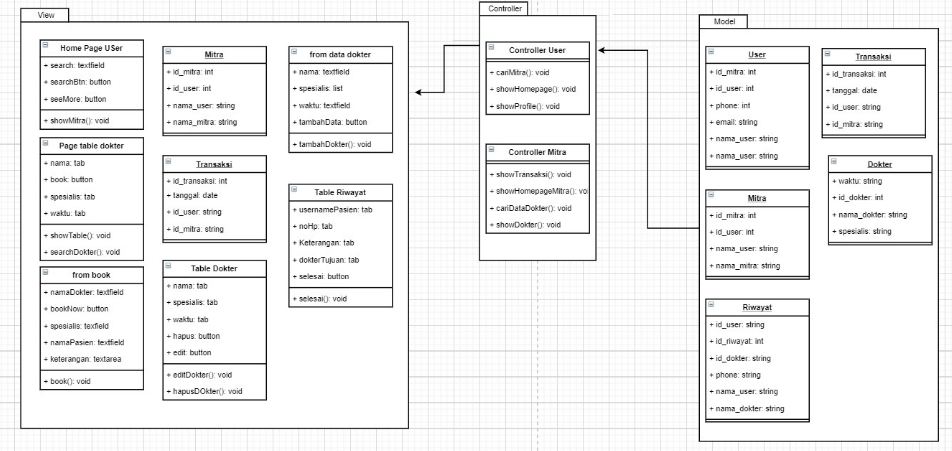
**

#### *3.1.7.5 Sequence Diagram*



# Perancangan Detil

#### Perancangan Detil Kelas

**

**TABEL KELAS :**

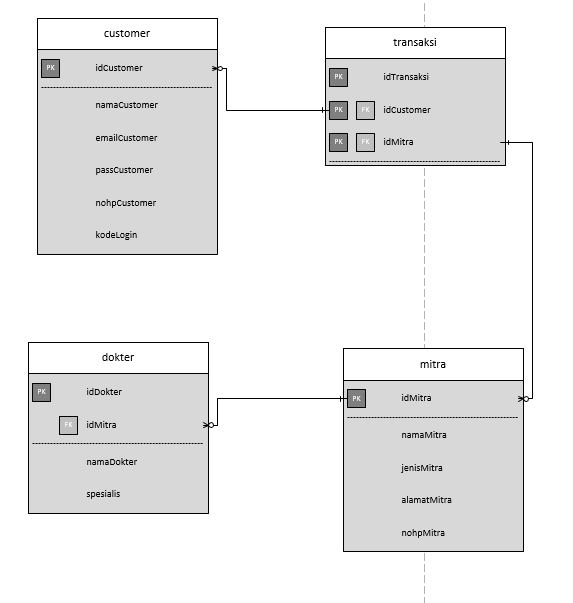
|  |  |  |  |
| --- | --- | --- | --- |
| **ID Kelas** | **Nama Kelas Perancangan** | **Atribute (visibility)** | **Method / Operation** |
| 1 | user | public | controller |
| 2 | m\_user | public | model |
| 3 | mitra | public | controller |
| 4 | m\_mitra | public | model |
| 5 | home\_page\_user | public | view |
| 6 | page\_table\_dokter | public | view |
| 7 | form\_book | public | view |
| 8 | form\_isi\_profile | public | view |
| 9 | form\_edit\_profile | public | view |
| 10 | home\_page\_mitra | public | view |
| 11 | table\_dokter | public | view |
| 12 | form\_data\_dokter | public | view |
| 13 | table\_riwayat | public | view |
| 14 | transaksi | public | model |
| 15 | dokter | public | model |
| 16 | riwayat | public | model |

*Untuk setiap kelas:*

* *user*

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

Skema Relasi database yang digunakan



#### Perancangan Algoritma

#### Algoritma cariMitra

*Nama Kelas : user*

*Nama Operasi : carimitra*

*Algoritma : ALGO001*

|  |
| --- |
| sql : String ; data : array of string  sql ← “*SELECT \* FROM `mitra` WHERE `nama` LIKE '%$keyword%' OR `address` LIKE '%$keyword%' OR `phonenumber` LIKE '%$keyword%'*”  data = db→run(sql)  if (sql <> NULL) then  output(data)  else  output(NULL) |

#### Algoritma Booking

*Nama Kelas : user*

*Nama Operasi : booking*

*Algoritma : ALGO002*

|  |
| --- |
| id = int ;  if (nama <> NULL and spesialis <> NULL and nama\_user <> NULL and keterangan <> NULL and id\_user <> NULL and id\_mitra <> NULL and id\_dokter <> NULL) then  query = “*INSERT INTO `transaksi` (`id`, `id\_user`, `id\_mitra`, `id\_dokter`, `keterangan`) VALUES (NULL, '$i', '$n', '$p', 'sakit')*”  db → run(query)  else  output(“data tidak valid”) |

#### Algoritma editProfile

*Nama Kelas : user*

*Nama Operasi : editProfile*

*Algoritma : ALGO003*

|  |
| --- |
| id = int ;  if (email <> NULL and hp <> NULL and address <> NULL) then  query = “*UPDATE `user` SET `email` = 'email', `phonenumber` = 'hp', `address` = 'address' WHERE `user`.`id` = 'id'*”  db → run(query)  else  output(“data tidak valid”) |

#### Algoritma isiProfile

*Nama Kelas : user*

*Nama Operasi : isiProfile*

*Algoritma : ALGO004*

|  |
| --- |
| id = int ;  if (nama <> NULL and username <> NULL and email <> NULL and password <> NULL and phonenumber <> NULL and address <> NULL) then  query = “*INSERT INTO `user` (`id`, `username`, `nama`, `email`, `password`, `phonenumber`, `address`) VALUES (NULL, 'username', 'nama', 'email', 'password', 'phonenumber', 'address')*”  db → run(query)  else  output(“data tidak valid”) |

#### Algoritma inputDokter

*Nama Kelas : mitra*

*Nama Operasi : inputDokter*

*Algoritma : ALGO005*

|  |
| --- |
| id = int ; $keyword : array of string  if ($keyword <> NULL) then  query = “*INSERT INTO `dokter` (`id`, `id\_mitra`, `nama`, `spesialis`, `waktu`) VALUES (NULL, '$keyword[‘id\_mitra’]', '$keyword[‘nama’]', '$keyword[‘spesialis’]', '$keyword[‘waktu’]')*”  db → run(query)  else  output(“data tidak valid”) |

#### Algoritma hapusDokter

*Nama Kelas : mitra*

*Nama Operasi : hapusDokter*

*Algoritma : ALGO006*

|  |
| --- |
| id = int ;  query = “*DELETE FROM `dokter` WHERE `dokter`.`id` = ‘id’*”  db → run(query) |

#### Algoritma lihatBooking

*Nama Kelas : mitra*

*Nama Operasi : lihatBooking*

*Algoritma : ALGO007*

|  |
| --- |
| id = int ;  query = “*SELECT `dokter`.`nama` AS `nama\_dokter`, `mitra`.`nama` AS `nama\_mitra` FROM `transaksi` INNER JOIN `mitra` ON `mitra`.`id`=`transaksi`.`id\_mitra` INNER JOIN `dokter` ON `dokter`.`id`=`transaksi`.`id\_dokter` WHERE `id\_user`= id*”  db → run(query) |

#### Algoritma hapusBooking

*Nama Kelas : user*

*Nama Operasi : hapusBooking*

*Algoritma : ALGO008*

|  |
| --- |
| id = int ;  query = “*DELETE FROM `transaksi` WHERE `transaksi`.`id` = id*”  db → run(query) |

#### Algoritma showProfile

*Nama Kelas : user*

*Nama Operasi : showProfile*

*Algoritma : ALGO009*

|  |
| --- |
| id = int ;  query = “*SELECT \* FROM `user` WHERE `user`.`id`=’id’*”  db → run(query) |

#### Perancangan Query

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-001* | *SELECT \* FROM `mitra` WHERE `nama` LIKE '%$keyword%' OR `address` LIKE '%$keyword%' OR `phonenumber` LIKE '%$keyword%'* | *Query untuk cari mitra* |
| *Q-002* | *INSERT INTO `transaksi` (`id`, `id\_user`, `id\_mitra`, `id\_dokter`, `keterangan`) VALUES (NULL, '$i', '$n', '$p', 'sakit')* | *Query Booking* |
| *Q-003* | *UPDATE `user` SET `email` = '$email', `phonenumber` = '$phone', `address` = '$address' WHERE `user`.`id` = '$id'* | *Query edit profile* |
| *Q-004* | *INSERT INTO `user` (`id`, `username`, `nama`, `email`, `password`, `phonenumber`, `address`) VALUES (NULL, '$keyword', '$keyword', '$keyword', '$keyword', '$keyword', '$keyword')* | *Query isi profile* |
| *Q-005* | *INSERT INTO `dokter` (`id`, `id\_mitra`, `nama`, `spesialis`, `waktu`) VALUES (NULL, '$keyword[‘id\_mitra’]', '$keyword[‘nama’]', '$keyword[‘spesialis’]', '$keyword[‘waktu’]')* | *Query input dokter* |
| *Q-006* | *DELETE FROM `dokter` WHERE `dokter`.`id` = ‘$keyword’* | *Query hapus dokter* |
| *Q-007* | *SELECT `dokter`.`nama` AS `nama\_dokter`, `mitra`.`nama` AS `nama\_mitra` FROM `transaksi` INNER JOIN `mitra` ON `mitra`.`id`=`transaksi`.`id\_mitra` INNER JOIN `dokter` ON `dokter`.`id`=`transaksi`.`id\_dokter` WHERE `id\_user`='$id'* | *Query lihat booking* |
| *Q-008* | *DELETE FROM `transaksi` WHERE `transaksi`.`id` = id* | *Query hapus booking* |
| *Q-009* | *SELECT \* FROM `user` WHERE `user`.`id`=’$keyword’* | *Query nampilin profile user* |

# Matriks Kerunutan (Requirement Traceability Matrix)

|  |  |  |
| --- | --- | --- |
| **Kode FR** | **Nama Functional Requirement** | **Nama Use Case** |
| MK-1 | Melakukan booking | booking |
| MK-2 | Mencari Mitra | cari mitra |
| MK-3 | Melihat pembookingan | lihat pembookingan |
| MK-4 | Melakukan edit profile | Olah profile |
| MK-5 | Menambah dokter | Input dokter |
| MK-6 | Menghapus dokter | Hapus dokter |
| MK-7 | Menghapus pembokingan | hapus pembookingan |

# 