

Aplikasi Pemrograman Berorientasi Objek

## “Resepsionis Rumah Sakit”



Disusun Oleh :

Kelompok C7

Adwityo S.P 15-2018-080

Annisa O.Z 15-2018-084

Irfan H.P 15-2018-089

Rijal R.F.F 15-2018-104

**PROGRAM STUDI INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL  
BANDUNG  
2020**

## Kata Pengantar

Segala puji bagi Allah Subhanahu wa Ta'ala, yang telah melimpahkan rahmat dan karunia kepada hamba-Nya sehingga laporan aplikasi berjudul “Resepsionis Rumah Sakit” dapat diselesaikan dengan penuh kemudahan.

Laporan ini bertujuan untuk memenuhi nilai tugas akhir mata kuliah pemrograman berorientasi objek, laporan ini berisi perancangan aplikasi serta fungsinya secara umum, serta berisikan kode pemrograman serta basis data dengan penjelasannya.

Penulis menyadari bahwa laporan ini masih belum sempurna, oleh karenanya penulis menerima saran dan masukan guna menyempurnakannya, penulis berharap laporan ini dapat bermanfaat bagi penulis dan pembaca khususnya dalam mempelajari pemrograman berorientasi objek.

Bandung,... Juli 2020

Penulis

## Daftar Gambar

Gambar 3.1 Sistim Kerja Aplikasi .....	
Gambar 3.2 Alur Sistem Aplikasi .....	
Gambar 3.3 UML Aplikasi .....	
Gambar 3.4 ERD Aplikasi .....	
Gambar 3.5 TRD Aplikasi .....	
Gambar 3.6 DDS Aplikasi .....	
Gambar 3.7 Sitemaps Aplikasi .....	
Gambar 3.8 Mockup Main Menu.....	
Gambar 3.9 Mockup Data Asuransi.....	
Gambar 3.10 Mockup Data Bayar .....	
Gambar 3.11 Mockup Data Dokter.....	
Gambar 3.12 Mockup Data Pasien .....	
Gambar 4.1 Tampilan Main Menu.....	
Gambar 4.2 Tampilan Bagian Pembayaran .....	
Gambar 4.3 Tampilan Bagian Input Data Pasien.....	
Gambar 4.4 Tampilan Bagian Input Data Dokter .....	
Gambar 4.5 Tampilan Bagian Input Data Asuransi.....	
Gambar 4.6 SourceCode kelas config.....	
Gambar 4.7 SourceCode kelas Create .....	
Gambar 4.8 SourceCode kelas Read.....	
Gambar 4.9 Contoh SourceCode kelas Create.....	
Gambar 4.10 Contoh SourceCode kelas Read .....	
Gambar 4.11 SourceCode kelas Model .....	
Gambar 4.12 SourceCode kelas Update .....	
Gambar 4.13 Contoh SourceCode kelas Edit .....	
Gambar 4.14 Contoh SourceCode kelas Delete.....	

## Daftar Isi

Kata Pengantar .....	2
Daftar Isi .....	4
BAB I PENDAHULUAN .....	6
1.1. Latar Belakang Masalah .....	6
1.2. Tujuan Penelitian .....	7
1.3. Rumusan Masalah .....	7
1.4. Batasan Masalah .....	8
1.5. Sistematika Penulisan .....	9
BAB II LANDASAN TEORI .....	10
2.1 Konsep Dasar Sistem .....	10
2.2 Pengertian Sistem Aplikasi Resepsionis Rumah Sakit .....	11
2.3 Aplikasi .....	11
2.4 Java .....	11
2.5 NetBeans .....	12
BAB III PERANCANGAN .....	13
3.1 Studi Kasus .....	13
3.2 Fitur .....	14
3.3 Diagram Alir .....	15
3.4 Penjelasan Alur Proses .....	16
3.5 Diagram Kelas .....	17
3.6 Basis Data .....	18
3.7 Sitemaps Aplikasi .....	20
3.8 Rancangan Mock-Up UI .....	21
BAB IV IMPLEMENTASI .....	25
4.1 Hasil Output .....	25
4.2 Program & Penjelasan Program .....	30
4.2.1 Kelas Config .....	30
4.2.2 Kelas Create .....	31
4.2.3 Kelas Read .....	32
4.2.4 Contoh Instansiasi Kelas Create .....	33
4.2.5 Contoh Instansiasi Kelas Read .....	34
4.2.6 Kelas Model .....	35
4.2.7 Kelas Update .....	37
4.2.8 Kelas Edit .....	38
4.2.9 Contoh Instansiasi Kelas Delete .....	39

BAB V <u>P</u> ENUTUP .....	40
5.1    Kesimpulan.....	40
5.2    Saran.....	41
Daftar Pustaka .....	42

# **BAB I**

## **PENDAHULUAN**

Pada bab I akan dibahas mengenai latar belakang masalah, rumusan masalah, tujuan penelitian, Batasan masalah dan tinjauan pustaka mengenai aplikasi pemerograman berorientasi objek dengan judul ‘resepsionis rumah sakit’.

### **1.1. Latar Belakang Masalah**

Pada banyak bidang profesional khususnya dunia medis terdapat banyak masalah yang dihadapi terutama mengenai data. mulai dari pengambilan data, pengklasifikasian data, pengenalan data, analisis data hingga data dapat digunakan dalam berbagai pengambilan keputusan sebagai informasi, salah satu komponen yang sangat berharga adalah data diri pasien terutama idenitas dan riwayat penyakit sendiri.

Dengan perkembangan teknologi computer terutama dalam bidang basis data dan aplikasi khusunya pada pengolahan data agar mudah untuk dimasukan, diubah dan di operasikan dalam system, data dapat diatur dengan mudahnya dan disimpan dengan aman.

Muncul sebuah pemikiran mengenai pembuatan aplikasi yang dapat mengolah data baik masukan baru maupun perubahan data yang dapat diandalkan berbasi pemerograman berorientasi objek dengan bahasa pemerograman java beserta tampilan user interface dari java dan system basis data yang terhubung.

## **1.2. Tujuan Penelitian**

Tujuan yang ingin dicapai, sebagai berikut :

1. Menghasilkan aplikasi yang baik & bermanfaat dalam pengolahan data dengan system CRUD
2. Mengetahui manfaat dari aplikasi pemrograman berorientasi objek dalam pengolahan data
3. Mengetahui proses perancangan system aplikasi pemrograman berorientasi objek
4. Mengetahui penerapan pemrograman berorientasi objek khususnya Bahasa pemrograman java

## **1.3. Rumusan Masalah**

Berdasarkan latar belakang maka masalah yang akan diuraikan, sebagai berikut :

1. Bagaimana membangun aplikasi pemrograman berorientasi objek
2. Bagaimana manfaat aplikasi pemrograman berorientasi objek
3. Bagaimana perancangan system aplikasi pemrograman berorientasi objek
4. Bagaimana penerapan pemrograman berorientasi objek khususnya Bahasa pemrograman java

## **1.4. Batasan Masalah**

Adapun Batasan masalah dalam aplikasi ini adalah sebagai berikut :

1. Penggunaan aplikasi terdiri dari login dan main program
2. Terdapat 1 pengguna yaitu bagian resepsionis
3. Aplikasi menggunakan prinsip dasar oop dengan integrasi basis data
4. Menggunakan bahasa pemrograman Java dan IDE NetBeans
5. Tampilan pengguna berupa gui
6. Menggunakan system CRUD



## **1.5. Sistematika Penulisan**

### **a. Halaman Judul**

Pada bagian ini berisikan tampilan cover berisikan judul kegiatan, identitas penulis dan instansi penulis.

### **b. Halaman Daftar Isi**

Pada bagian ini berisikan daftar bagian-bagian dalam laporan ini, dengan format judul bagian disertai halaman.

### **c. Halaman Kata Pengantar**

Pada bagian ini berisikan kata-kata dari penulis mengenai laporan kegiatan yaitu pemaparan singkat serta harapan penulis pada laporan kegiatan yang telah dibuat.

### **d. Halaman Bab 1 Pendahuluan**

Pada bab ini berisikan latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan sistematika penulisan pada laporan

### **e. Halaman Bab 2 Landasan Teori**

Pada bab ini berisikan keterangan dari system yang akan digunakan baik dari IDE dan system itu sendiri

### **f. Halaman Bab 3 Perancangan**

Pada bab ini berisikan studi kasus, algoritma program, flowchart dan design proyek dari kegiatan yang telah dilakukan.

### **g. Halaman Bab 4 Implementasi**

Pada bab ini berisikan penggunaan kode program yang digunakan dan hasil dari eksekusi program.

### **h. Halaman Bab 5 Penutup**

Pada bab ini berisikan kesimpulan dari kegiatan serta saran penulis terhadap pembaca

### **i. Halaman Daftar Pustaka**

Pada bagian ini berisikan acuan yang digunakan oleh penulis.

## **BAB II**

### **LANDASAN TEORI**

Teori merupakan dasar yang digunakan sebagai sarana untuk mendukung sekaligus memperkuat dalam penyusunan suatu sistem informasi yang akan dibangun. Pada bab ini akan dijelaskan mengenai landasan teori yang berkaitan erat mengenai aplikasi pemrograman berorientasi objek dengan judul ‘resepsionis rumah sakit’.

#### **2.1 Konsep Dasar Sistem**

Suatu sistem pada dasarnya merupakan suatu susunan yang teratur dari kegiatan yang berhubungan satu sama lain dan prosedur-prosedur yang berkaitan yang melaksanakan dan mempermudah pelaksanaan kegiatan utama dari suatu organisasi.

Menurut (Jogiyanto, 2001) Sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu ”.

Komponen-komponen atau subsistem dalam suatu sistem tidak dapat berdiri lepas sendiri-sendiri. Komponen-komponen atau subsistem saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai. Suatu sistem mempunyai maksud tertentu. Ada yang menyebutkan maksud dari suatu sistem adalah untuk mencapai suatu tujuan (goal) dan ada yang menyebutkan untuk mencapai sasaran (objective). Suatu sistem pada dasarnya merupakan suatu susunan yang teratur. dari kegiatan yang berhubungan satu sama lainnya dan prosedur-prosedur yang berkaitan yang melaksanakan dan memudahkan pelaksanaan kegiatan utama dari suatu organisasi.

## **2.2 Pengertian Sistem Aplikasi Resepsionis Rumah Sakit**

System yang mengelola sebuah transaksi data pada rumah sakit khususnya bagian penerimaan pasien dan penjadwalan dokter serta administrasi, yang dimana tujuannya untuk dapat mempermudah memasukan data pasien maupun dokter dan juga mempermudah pembayaran dengan baik dan lebih mudah karena sudah bisa menggunakan system aplikasi.

## **2.3 Aplikasi**

Aplikasi merupakan terapan bahasa pemrograman tertentu yang diciptakan untuk mempermudah mengolah data dengan ketentuan tertentu dan dapat dengan mudah digunakan serta dapat dikembangkan berdasarkan kebutuhan pengguna.

Menurut (Jogiyanto, 2001) Aplikasi merupakan suatu intruksi / pernyataan yang ada di suatu perangkat keras (Hardware) baik itu komputer atau smartphome yang di rancang sedemikian rupa agar bisa mengolah suatu masukan (input) menjadi keluaran (output).

## **2.4 Java**

Merupakan bahasa pemrograman berbasis windows yang menyediakan kemampuan untuk membuat algoritma atau system data yang memiliki sifat pada tiap bentukannya, pada pembuatan aplikasi java dapat menggunakan user interface dan system pemrograman berorientasi pada tiap objek yang di buat. java memberikan kemudahan dalam menggunakan kode program karena tidak bersifat tahap atau prosedural yang jika terjadi error maka harus ditemukan terlebih dahulu dan harus diubah, namun pada java jika terdapat error biasanya hanya terdapat pada satu kelas atau satu objek dan dapat segera dikathui dan di perbaiki.

## **2.5 NetBeans**

Netbeans merupakan sebuah aplikasi Integrated Development Environment (IDE) berbasis Java dengan swing. Swing disini adalah sebuah teknologi Java untuk pengembangan aplikasi dekstop yang dapat berjalan di berbagai macam platform seperti windows, linux, Mac OS. Integrated Development Environment adalah lingkup pemrograman yang di integrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan Graphic User Interface (GUI), yaitu suatu kode editor atau text, suatu compiler dan suatu debugger.

Netbeans bisa digunakan progammer untuk menulis, meng-compile, mencari kesalahan dan menyebarkan program netbeans yang ditulis dalam bahasa pemrograman java namun selain itu netbeans bisa mendukung bahasa pemrograman lainnya dan program ini juga bebas untuk digunakan dan untuk membuat professional dekstop, web, enterprise, and mobile applications, Netbeans memiliki fitur-fitur yang dimanfaatkan dalam pembangunan aplikasi.

## **BAB III**

### **PERANCANGAN**

Pada bagian ini akan dibahas mengenai bentuk model rancangan berisikan berisikan studi kasus, mockup, fitur, basis data, site maps, diagram alir dan penjelasan diagram alurnya yang berkaitan erat mengenai aplikasi pemerograman berorientasi objek dengan judul ‘resepsionis rumah sakit’.

#### **3.1 Studi Kasus**

Dalam rancangan tugas akhir ini yang di ajukan oleh tim penulis, data pasien yang dahulunya dilakukan secara manual atau di data secara bertumpuk, pasien diharuskan datang ke resepsionis maupun rumah sakit lalu mengisikan data dengan menulis data diri pasien. permasalahan yang ditemui adalah setiap pasien datang maka harus melakukan pendaftaran kembali dan tiap pasien jika beralih rumah sakit maka harus melakukan pendaftaran dari awal, serta mengenai jadwal dokter yang harus di buat secara manual dan system antrian yang diurutkan menggunakan nomor. Dengan memanfaatkan aplikasi diharapkan system dapat menangani masalah tersebut seperti pendataan pasien yang lebih mudah serta membuat jadwal dokter dan nomor urut secara otomatis lalu melakukan jumlah pembayaran secara tepat agar mempermudah kerja rumah sakit.

Aplikasi resepsionis rumah sakit ini adalah sebuah system yang dimana dapat mengelola data pasien serta jadwal pemeriksaan dokter dan menentukan jadwal pemeriksaan lalu menentukan pembayaran. Aplikasi ini diharapkan dapat mempermudah kegiatan dalam rumah sakit khususnya pengolahan data.

## **3.2 Fitur**

Aplikasi resepsionis rumah sakit ini memiliki beberapa fitur utama, sebagai berikut:

1. Fitur CRUD pada data pasien serta dokter dan dapat di lakukannya pada asuransi

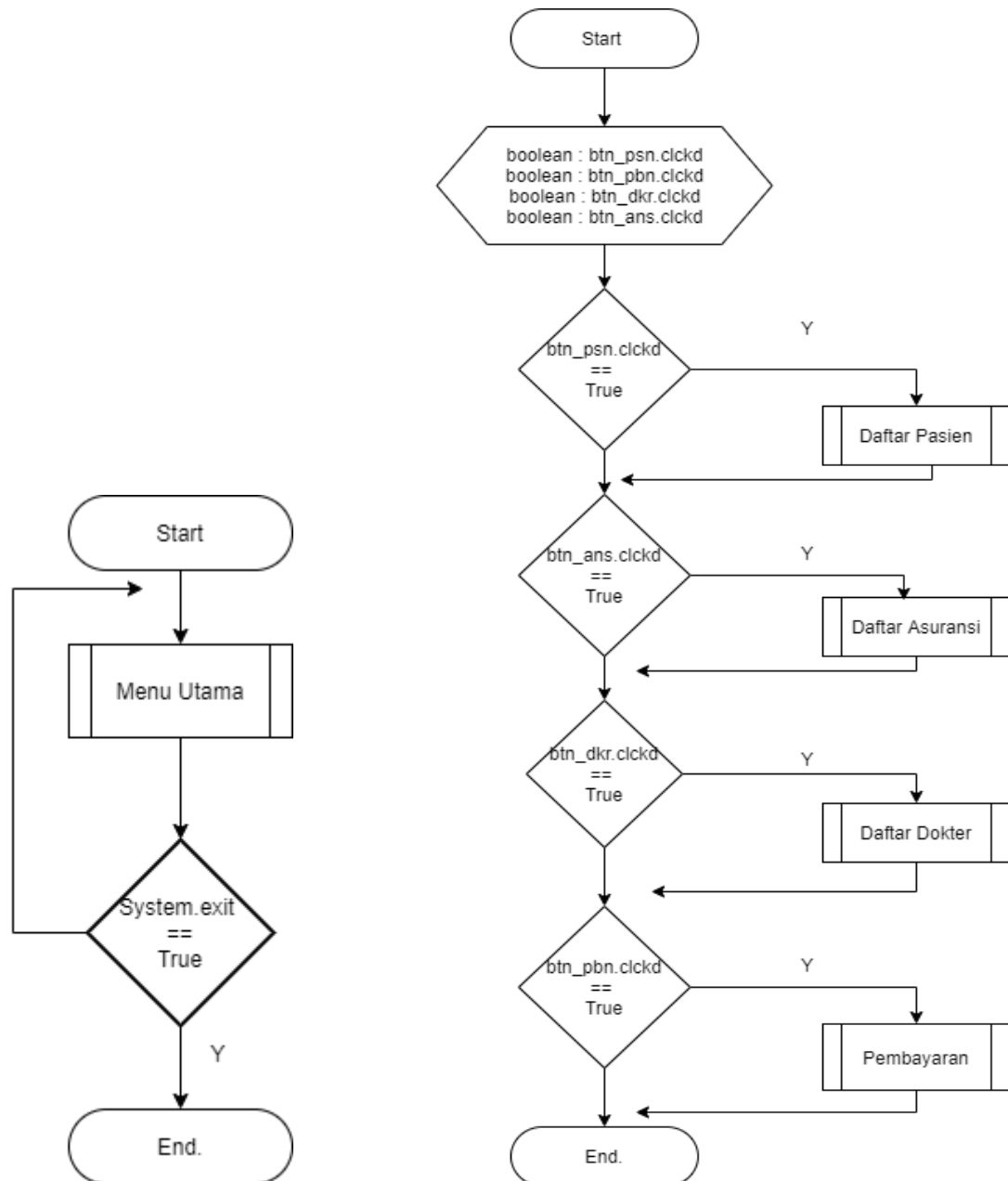
Fitur ini merupakan kemampuan system aplikasi agar pengguna dapat menambahkan data, mengubah data, menghapus data dan menampilkan data pada data pasien, dokter maupun asuransi.

2. Fitur Administrasi & Asuransi

Fitur ini merupakan kemampuan system aplikasi agar pasien dapat menambahkan klaim asuransi yang digunakan pasien lalu secara otomatis pasien dapat mendapatkan klaim asuransi dan dapat melakukan pembayaran setelah melakukan pengobatan.

### 3.3 Diagram Alir

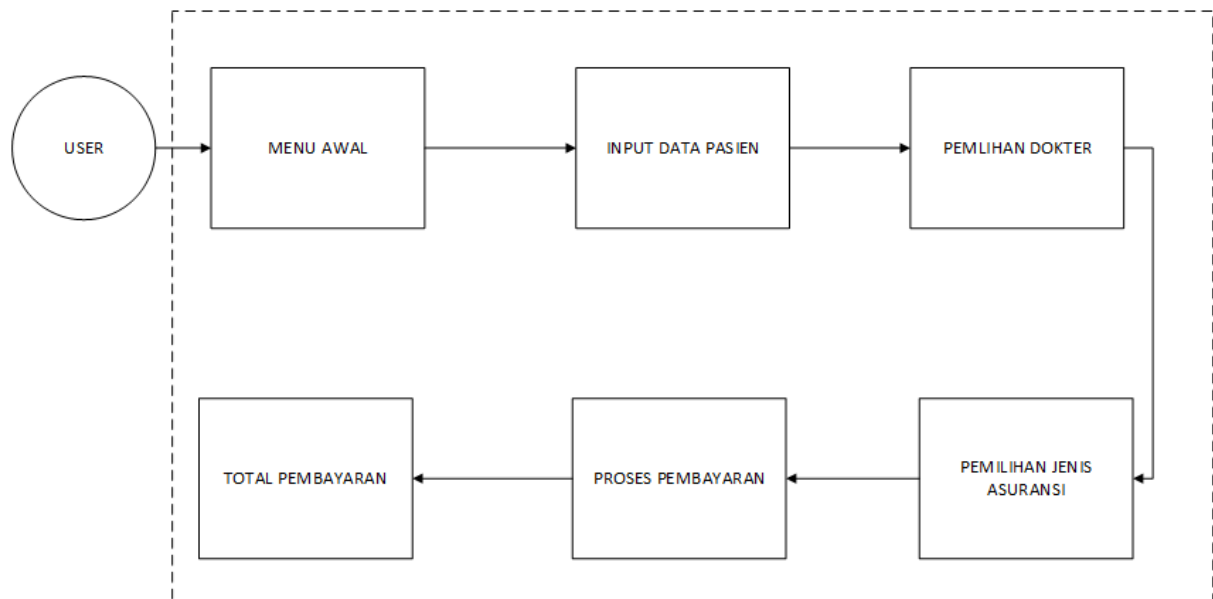
Aplikasi resepsionis rumah sakit ini memiliki rancangan aplikasi, sebagai berikut:



Gambar 3.1 Sistem Kerja Aplikasi

### 3.4 Penjelasan Alur Proses

Pengguna aplikasi ini berupa resepsionis pada rumah sakit, resepsionis bertugas untuk mengatur masuknya data pasien, dokter dan data asuransi, setelah memasukkan data tersebut resepsionis dapat menentukan jadwal praktik dokter serta nomor antrian pasien, saat pasien telah melakukan cek maka akan dihasilkan bagian pembayaran dengan potongan asuransi.



Gambar 3.2 Alur Proses Aplikasi

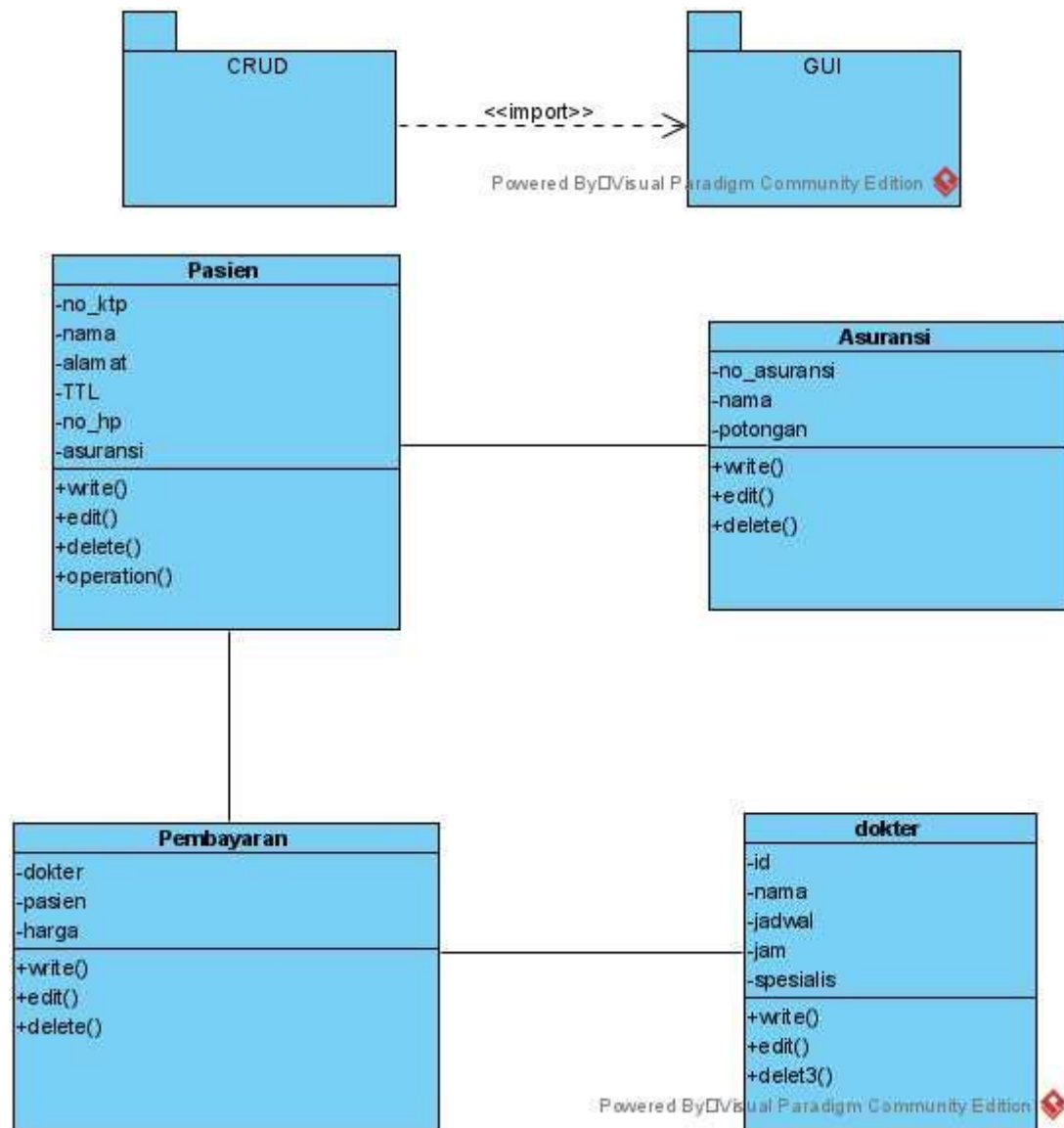
Secara urutan dapat dilihat, sebagai berikut :

1. Memasukan data pasien, dokter & asuransi
2. Menentukan jadwal dokter beserta antrian pasien
3. Pasien melakukan cek oleh dokter
4. Melakukan proses administrasi



### 3.5 Diagram Kelas

Aplikasi resepsionis rumah sakit ini memiliki rancangan diagram aplikasi, sebagai berikut:

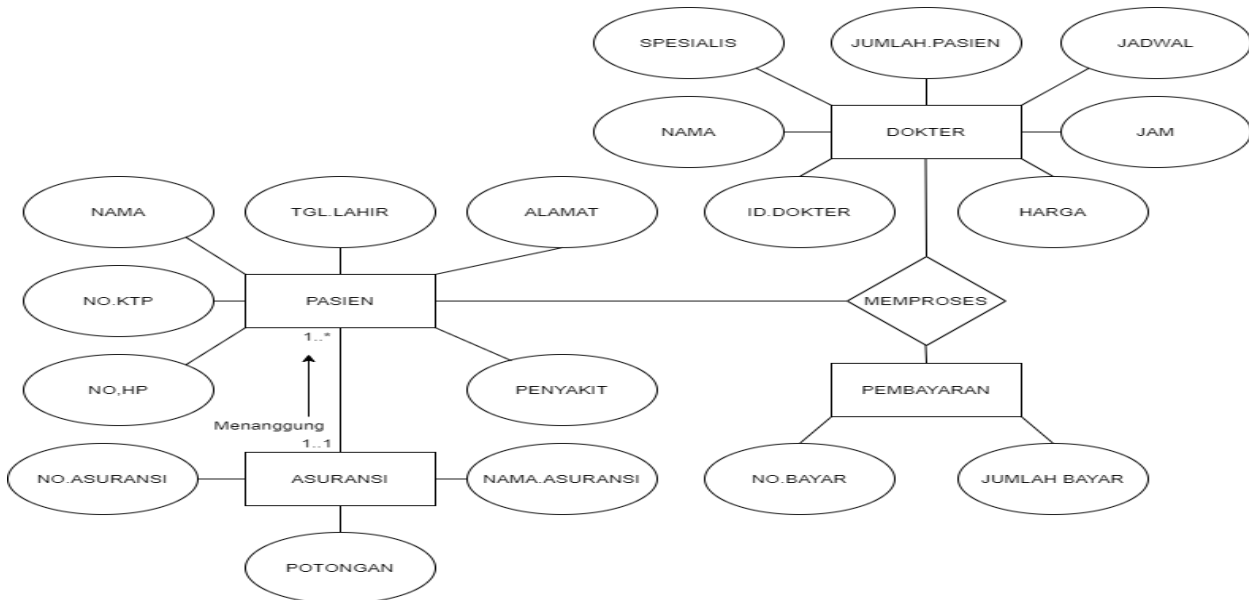


Gambar 3.3 UML Aplikasi

### 3.6 Basis Data

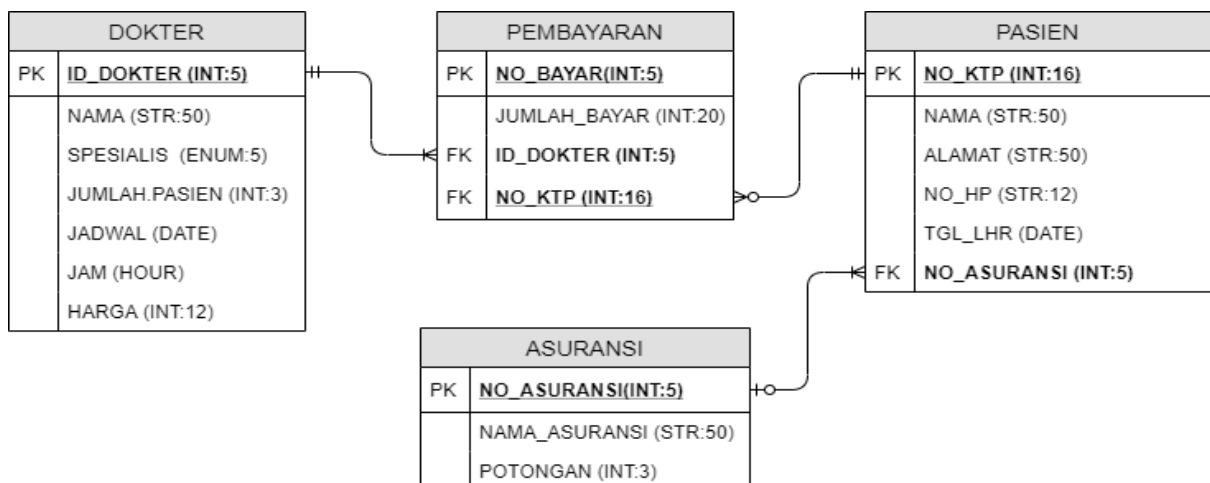
Aplikasi resepsionis rumah sakit ini memiliki rancangan basis data, sebagai berikut:

#### 3.6.1 ERD



Gambar 3.4 ERD Aplikasi

#### 3.6.2 TRD



Gambar 3.5 TRD Aplikasi

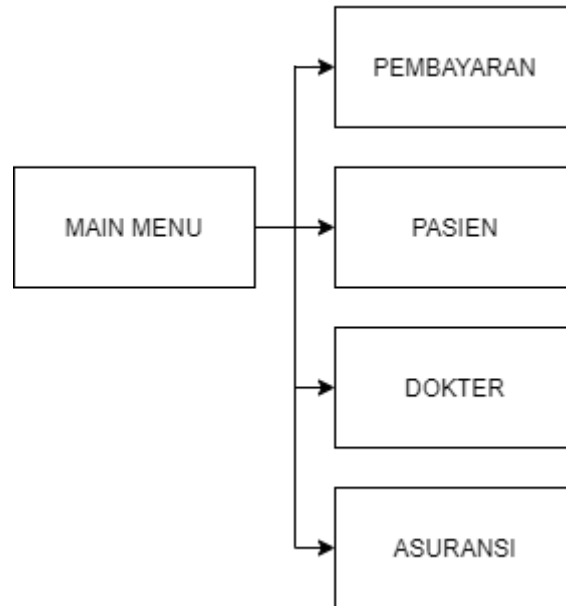
### 3.6.3 DDS

NO	DATA	TYPE	LENGTH	PK/FK	N/NN	FORM	ALIAS	DESCRIPTION
1	ALAMAT	VARCHAR	50	-	N	XXXX	LOKASI	ALAMAT TINGGAL
2	HARGA	INT	12	-	N	XXXX,RP	BIAYA	HARGA PRAKTIK
3	ID_DOKTER	VARCHAR	20	PK/FK	NN	XXXX	NOMOR DOKTER	NOMOR IDENTITAS DOKTER
4	JADWAL	DATE	8	-	N	XXYYZZZZ	JAM KERJA	JADWAL PRAKTIK DOKTER
5	JAM	TIME	6	-	N	XXYYZZ	JAM MULAI	JAM MULAI PRAKTIK DOKTER
6	JUMLAH_BAYAR	INT	20	-	N	XXXX,RP	BAYAR JADWAL	JUMLAH BAYAR CEK
7	JUMLAH_PASIEN	INT	3	-	N	XXXX	PRAKTIK PASIEN	JUMLAH PASIEN PRAKTIK
8	NAMA	VARCHAR	50	-	N	XXXX	NAMA ORANG	NAMA PASIEN/DOKTER
9	NAMA_ASURANSI	VARCHAR	51	-	N	XXXX	PENUNJANG BIAYA	NAMA ASURANSI
10	NO_ASURANSI	INT	5	PK	NN	XXXX	NOMOR ASURANSI	NOMOR JENIS ASURANSI
11	NO_BAYAR	INT	5	PK	NN	XXXX	NOMOR ANTRI	NOMOR BAYAR TIAP PASIEN
12	NO_HP	VARCHAR	13	-	N	XXXX	NO PONSEL	NOMOR PONSEL PENGGUNA
13	NO_KTP	VARCHAR	16	PK	NN	XXXX	NO PENDUDUK	ALAMAT EMAIL PENGGUNA
14	POTONGAN	INT	3	-	N	X%	POTONGAN BIAYA	POTONGAN BIAYA OLEH ASURANSI
15	PENYAKIT	ENUM	10	-	N	XXXX	GEJALA\SAKIT	PENYAKIT DIDERITA PASIEN
16	SPEKIALIS	VARCHAR	20	-	N	XXXX	KEAHLIAN DOKTER	BIDANG KEAHLIAN DOKTER
17	TGL_LAHIR	INT	20	-	N	XXXX,RP	TEBUSAN	JUMLAH BAYAR - ASURANSI

Gambar 3.6 DDS Aplikasi

### 3.7 Sitemaps Aplikasi

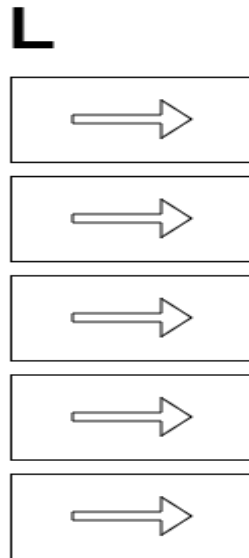
Aplikasi resepsionis rumah sakit ini memiliki rancangan penggunaan, sebagai berikut:



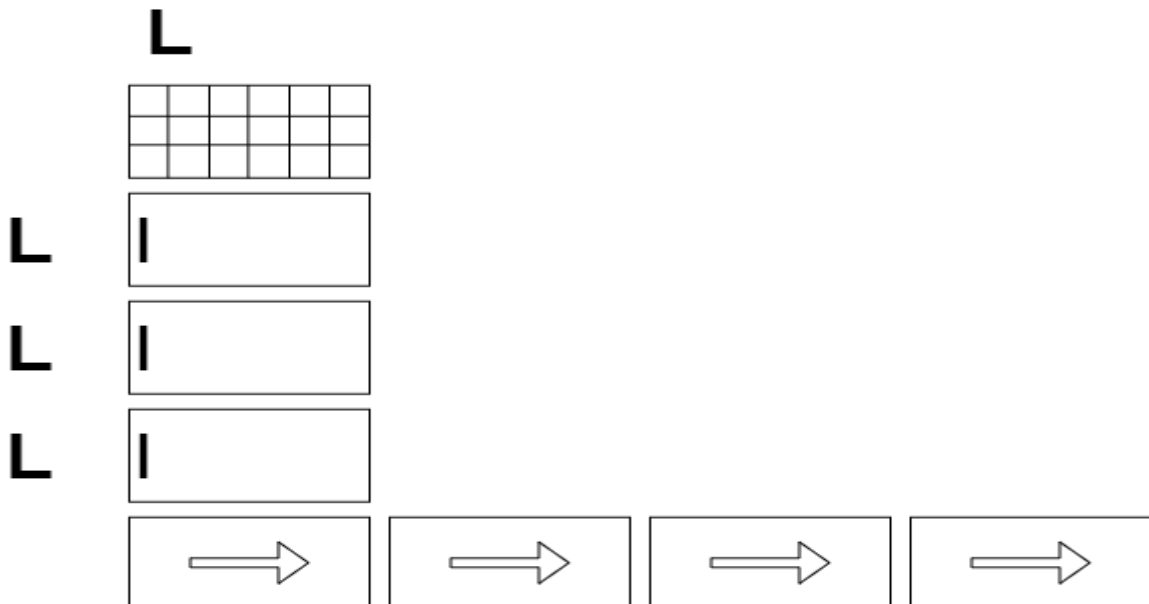
Gambar 3.7 Sitemaps Aplikasi

### 3.8 Rancangan Mock-Up UI

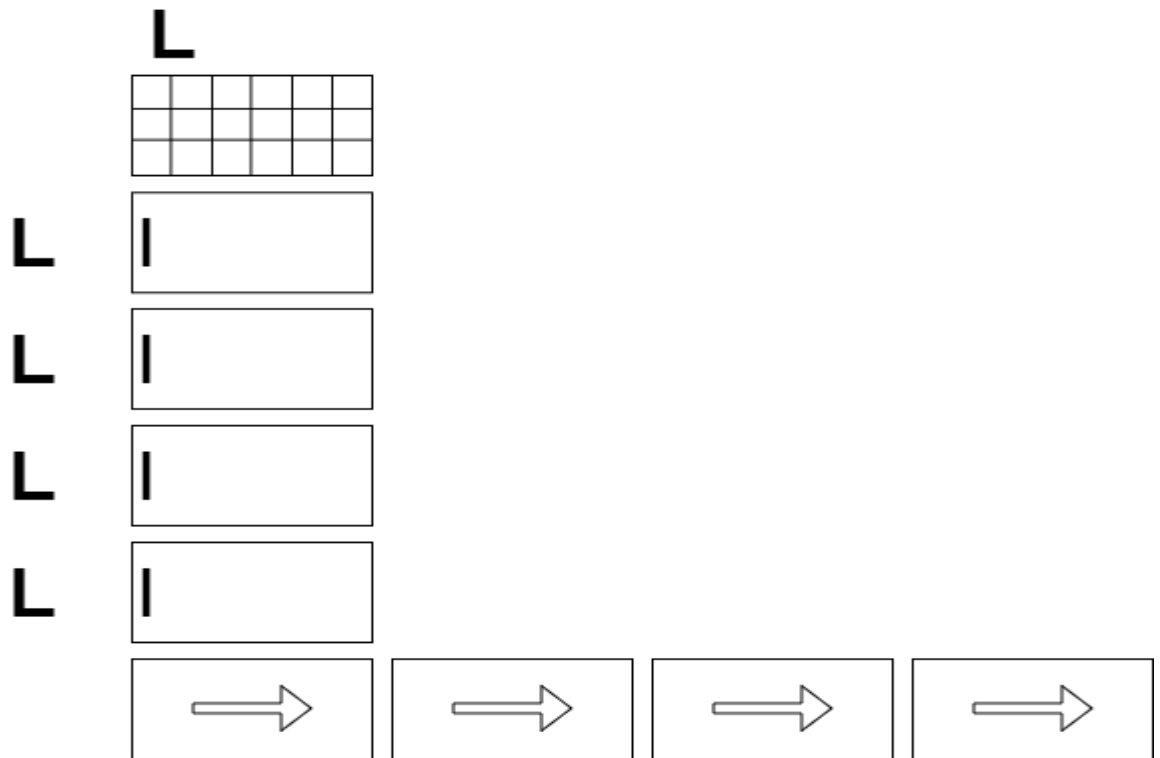
Aplikasi resepsionis rumah sakit ini memiliki rancangan tampilan pengguna, sebagai berikut:



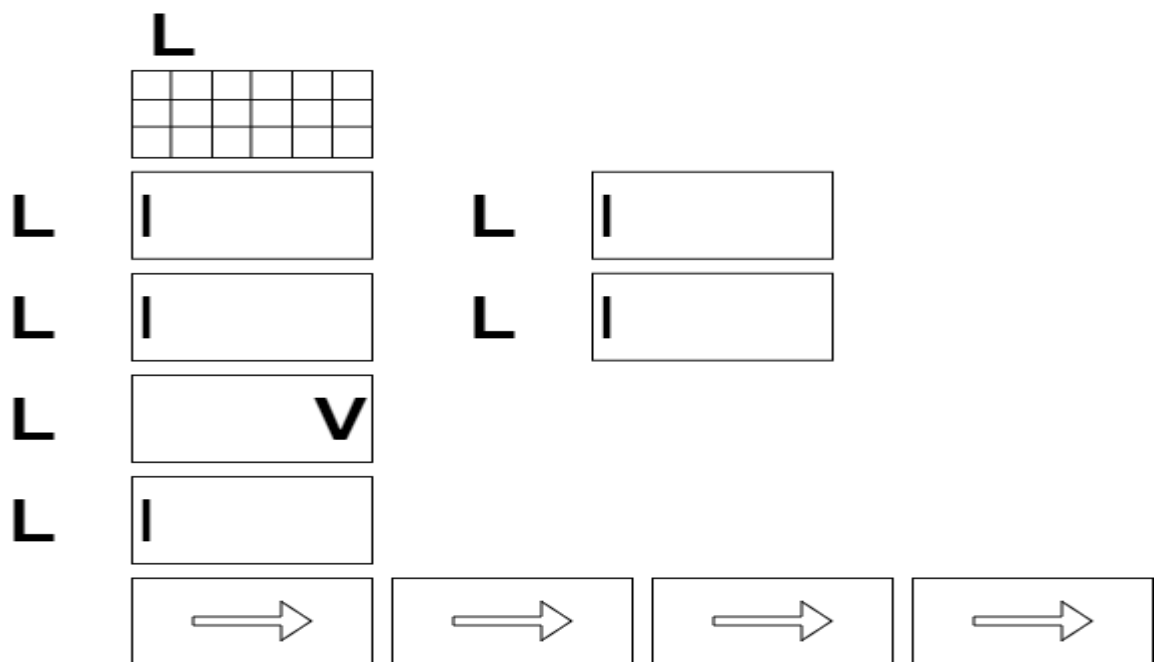
Gambar 3.8 Mockup Main Menu



Gambar 3.9 Mockup Data Asuransi



Gambar 3.10 Mockup Data Bayar



Gambar 3.11 Mockup Data Dokter



Pada saat pertama kali membuka atau menjalankan aplikasi, user akan melihat form menu utama (Gambar 3.7) yang terdiri dari 3 button yaitu 'Data Pasien', 'Data Dokter' dan 'Data Asuransi'. Untuk memasukkan data ke table data pembayaran, dapat dilakukan dengan mengisi data field yang dibutuhkan seperti No.KTP dan ID Dokter dan apabila user menekan button 'Total Bayar' akan memunculkan jumlah total pembayaran user setelah memilih dokter dan memasukkan data asuransi apa yang digunakan.

Apabila user memilih atau menekan button 'Data Pasien' user akan diarahkan pada form yang berisi data pasien (Gambar 3.11), apabila pasien akan melakukan pembayaran maka harus dilakukan pengisian data diri pada form ini. Data diri yang perlu diisi antara lain no. KTP, nama, ttl, no.HP alamat dan jenis asuransi. Untuk menyimpan data dapat dilakukan dengan menekan button save. Selain dilakukan pengisian data pasien dapat dilihat juga data pasien lain dari sebuah table yang terdapat pada form ini.

Apabila user memilih atau menekan button 'Data Dokter' user akan diarahkan pada form yang berisi data dokter yang terdapat pada sebuah table (Gambar 3.10), dan apabila user ingin menambahkan Data Dokter maka dapat dilakukan dengan pengisian data seperti ID, nama, spesialis, harga dokter jadwal dan jam dokter tersebut. Untuk menyimpan data dapat dilakukan dengan menekan button save.

Apabila user memilih atau menekan button 'Data Asuransi' user akan diarahkan pada form yang berisi data asuransi yang terdaftar pada rumah sakit tersebut (Gambar 3.8) dan ditampilkan melalui sebuah tabel, dan apabila user ingin menambahkan Data Asuransi maka dapat dilakukan dengan pengisian data seperti nomor asuransi, nama dan potongan harga dari asuransi tersebut. Untuk menyimpan data dapat dilakukan dengan menekan button save.



## **BAB IV**

### **IMPLEMENTASI**

Pada bagian ini akan dibahas mengenai hasil pengerjaan dari rancangan yang telah dibuat berisikan tampilan dan source code dari program aplikasi pemerograman berorientasi objek dengan judul ‘resepsionis rumah sakit’.

#### **4.1 Hasil Output**

Setelah melakukan pembuatan dan pengembangan aplikasi resepsionis rumah sakit ini memiliki hasil tampilan, sebagai berikut:



Gambar 4.1 Tampilan Main Menu

### Data dokter

ID	nama	Spesialis	harga	jadwal	jam
15001	Vayne	Bedah Sa...	12500000	2020-07-03	09:00:00
15002	Rais	Bedah Sa...	15000000	2020-07-05	10:00:00
15003	Tomo	Bedah Ob...	10000000	2020-07-04	11:00:00
15004	Ica	Bedah Ob...	20000000	2020-07-01	10:00:00
15005	Mar	Bedah Di...	10000000	2020-07-05	11:00:00
15006	Raj	Bedah Di...	15000000	2020-07-03	09:00:00
15007	Kram	Bedah Tor...	12500000	2020-07-04	09:00:00
15008	Yyan	Bedah Tor...	20000000	2020-07-01	12:00:00
15009	Adli	Bedah On...	22500000	2020-07-02	09:00:00
15010	Kal	Bedah On...	1000000	2020-07-02	11:00:00
15011	kuganteng		1250000	2020-10-10	09:00:00
15012	irfan gant...		1500000	2020-10-09	09:00:00
15013	zeus		1200000	2020-10-09	10:00:00

### DATA PEMBAYARAN

nama Pasien	nama Dokter	total bayar	jadwal
akuh	Vayne	20000000	2020-07-03
kamu	Kram	15000000	2020-07-04
akuh	Mar	22500000	2020-07-05
gatau	Ica	10000000	2020-07-01
akuh	Rais	17500000	2020-07-05
akuh	Rais		2020-07-05
gatau	Vayne		2020-07-03
coki	Vayne	8125000	2020-07-03
gatau	Vayne	11875000	2020-07-03
akuh	Vayne	9375000	2020-07-03
kamu	Rais	12000000	2020-07-05
akuh	hercules	750000	2020-10-07
akuh	Tomo	10000000	2020-07-04

### INPUT DATA PEMBAYARAN

No. KTP

ID Dokter

Total Bayar

Gambar 4.2 Tampilan Bagian Pembayaran

### DATA PASIEN

no KTP	nama	alamat	kontak	Tanggal lahir	asuransi
0	coki	ubr	18181818...	2020-06-30	10005
1	akuh	ML	18181818...	2020-06-30	10003
2	kamu	disitu	18181818...	2020-06-30	
3	akuh	ML	18181818...	2020-06-30	10003
4	gatau	gataulah	082173241	2020-06-10	10003
5	akuh	ML	18181818...	2020-06-30	10003
6	akuh	ML	18181818...	2020-06-30	10003
7	yagitu	ML	18181818...	2020-06-30	10003
8	akuh	ML	18181818...	2020-06-30	10003
9	akuh	jl. yang lurus...	9271837411	2020-06-30	10003
10	akuh	ML	18181818...	2020-06-30	10003
11	akuh	ML	18181818...	2020-06-30	10003
12	gatau	disana	02827432	2020-05-30	10003

### INPUT DATA PASIEN

No. KTP

Nama

TTL

No.HP

Alamat

Asuransi 

non-asuransi ▼

Gambar 4.3 Tampilan Bagian Input Data Pasien

**DATA DOKTER**

ID	nama	Spesialis	harga	jadwal	jam
15001	Vayne	Bedah Saraf	12500000	2020-07-03	09:00:00
15002	Rais	Bedah Saraf	15000000	2020-07-05	10:00:00
15003	Tomo	Bedah ObGyn	10000000	2020-07-04	11:00:00
15004	Ica	Bedah ObGyn	20000000	2020-07-01	10:00:00
15005	Mar	Bedah Digestif	10000000	2020-07-05	11:00:00
15006	Raj	Bedah Digestif	15000000	2020-07-03	09:00:00
15007	Kram	Bedah Toraks	12500000	2020-07-04	09:00:00
15008	Yyan	Bedah Toraks	20000000	2020-07-01	12:00:00
15009	Adli	Bedah Onkol...	22500000	2020-07-02	09:00:00
15010	Kal	Bedah Onkol...	10000000	2020-07-02	11:00:00
15011	kuganteng		1250000	2020-10-10	09:00:00
15012	irfan ganteng		1500000	2020-10-09	09:00:00
15013	zeus		1200000	2020-10-09	10:00:00

**INPUT DATA DOKTER**

ID

Jadwal

Nama

Jam

Spesialis 

Bedah Onkologi ▼

Harga

Gambar 4.4 Tampilan Bagian Input Data Dokter

**DATA ASURANSI**

no_asuransi	nama_asuransi	potongan
10001	AIA	35
10002	ACA	20
10003	Prudential	15
10004	Adira	25
10005	AXA	30
10006	Jasa Raharja	10
10007	BPJS	100
10008	BPJS Ketenagakerjaan	75
10009	Jiwasraya	5
10010	Allianz	50
10011	MakmurJaya	50
10012	Terang Bulan	50
10013	SinarCahaya	40

**INPUT DATA ASURANSI**

No. Asuransi

Nama

Potongan

Gambar 4.5 Tampilan Bagian Input Data Asuransi

## 4.2 Program & Penjelasan Program

### 4.2.1 Kelas Config

```
package CRUD;
import java.sql.*;
public class Config {
    private static Connection mysqlconfig;
    public Connection configDB()throws SQLException{
        try {
            String url="jdbc:mysql://localhost:3306/ta_oop"; //url database
            String user="root"; //user database
            String pass="";
            String driver = "com.mysql.jdbc.Driver";
            Class.forName(driver);
            mysqlconfig = DriverManager.getConnection(url, user, pass);
            System.out.println("Sukses");
        } catch (Exception e) {
            System.err.println(e); //perintah menampilkan error pada koneksi
        }
        return mysqlconfig;
    }
}
```

Gambar 4.6 SourceCode kelas config

Kelas config berfungsi untuk mengatur konfigurasi agar bisa terkoneksi dengan database mysql. untuk terkoneksi dengan database, butuh driver dari java yaitu jdbc. parameter masukan akan digunakan untuk mengeksekusi method execute statement dari library sql java

#### 4.2.2 Kelas Create

```
package CRUD;
import java.sql.*;
public class Create {

    public String create(String qry) throws SQLException{

        Config config = new Config();
        Connection kon = config.configDB();
        java.sql.PreparedStatement pst = kon.prepareStatement(qry);

        try {
            pst.executeUpdate();
            return "Berhasil Buat!";

        } catch (Exception e) {
            return e.getMessage();
        }

    }

}
```

Gambar 4.7 SourceCode kelas Create

Create adalah kelas untuk memasukan data ke dalam database. parameter masukan fungsinya adalah sintaks sql yang ditaruh dalam variabel bertipe String

#### 4.2.3 Kelas Read

```
package crud;
import java.sql.*;
public class Read {
    public ResultSet exec(String qry) throws SQLException{

        Config config = new Config();
        Connection kon = config.configDB();
        java.sql.Statement query = kon.createStatement();
        java.sql.ResultSet res;
        res = query.executeQuery(qry);
        return res;

    }
}
```

Gambar 4.8 SourceCode kelas Read

Read adalah kelas untuk mengambil data yang ada di sebuah data base. parameter masukan fungsinya adalah sintaks sql yang ditaruh dalam variabel bertipe String. parameter masukan akan digunakan untuk mengeksekusi method execute statement dari library sql java



#### 4.2.4 Contoh Instansiasi Kelas Create

```
String sql = "INSERT INTO asuransi (no_asuransi,nama_asuransi,potongan)
\n";
    String koma = ",";
    String petik = "\"";
    String tutup = "\"";
    sql = sql.concat("VALUES( ");
    sql = sql.concat(txtNoAsuransi.getText()+koma);
    sql = sql.concat(petik +txtNama.getText() +petik + koma);
    sql = sql.concat(txtPotongan.getText()+ tutup);

    try {
        System.out.print(new Create().create(sql));
    } catch (SQLException ex) {
        Logger.getLogger(Pendaftaran.class.getName()).log(Level.SEVERE,
null, ex);
    }
```

Gambar 4.9 Contoh SourceCode kelas Create

Kelas config berfungsi untuk mengatur konfigurasi agar bisa terkoneksi dengan database mysql. untuk terkoneksi dengan database, butuh driver dari java yaitu jdbc. parameter masukan akan digunakan untuk mengeksekusi method execute statement dari library sql java. Keluarannya berupa respon apakah data berhasil disimpan atau tidak.

#### 4.2.5 Contoh Instansiasi Kelas Read

```
DefaultTableModel model = new DefaultTableModel();

    model.addColumn("no_asuransi");
    model.addColumn("nama_asuransi");
    model.addColumn("potongan");
    String sql="select * from asuransi";
    try {
        ResultSet res = new Read().exec(sql);
        while(res.next()){
            model.addRow(new
Object[]{res.getString("no_asuransi"),res.getString("nama_asuransi"),res.getS
tring("potongan")});
        }
        this.tbl_asuransi.setModel(model);
    } catch (SQLException ex) {
        Logger.getLogger(TAsuransi.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
```

Gambar 4.10 Contoh SourceCode kelas Read

Ini adalah instansiasi dari kelas Read. Exec dari kelas read adalah method untuk menampilkan data dari database. PString sql berisi sintaks query dari bahasa sql untuk mengambil data dari tabel.keluaran method dari method exec di kelas read, adalah data bertipe result set. result set ini akan ditampilkan di tabel jframe library untuk gui di java.

#### 4.2.6 Kelas Model

```
package Model;
import CRUD.Create;
import CRUD.Update;
import GUI.Pendaftaran;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
public class Asuransi {
    String id;
    String nama;
    String ptng;
    public Asuransi(String id, String nama, String ptng) {
        this.id = id;
        this.nama = nama;
        this.ptng = ptng;
    }
    public void write(){
String sql = "INSERT INTO asuransi (no_asuransi,nama_asuransi,potongan) \n";
        String koma = ",";
        String petik = "\"";
        String tutup = "\"";
        sql = sql.concat("VALUES( ");
        sql = sql.concat(id+koma);
        sql = sql.concat(petik +nama +petik + koma);
        sql = sql.concat(ptng+ tutup);
        try {
            System.out.print(new Create().create(sql));
        } catch (SQLException ex) {
            Logger.getLogger(Pendaftaran.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    public void delete(){
String sql = "delete from asuransi where no_asuransi=";
        sql = sql.concat(id);

        try {
            System.out.print(new Update().update(sql));
        } catch (SQLException ex) {
            Logger.getLogger(Pasien.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    public void edit(){
String sql = "update asuransi \n";
        String koma = ",";
        String petik = "\"";
        sql = sql.concat("set nama_asuransi="+ petik +nama +petik + koma);
        sql = sql.concat("potongan="+ ptng +"\n");
        sql = sql.concat("where no_asuransi =");
        sql = sql.concat(id);
        System.out.print(sql);
        try {
            System.out.print(new Update().update(sql));
        } catch (SQLException ex) { Logger.getLogger(Pasien.class.getName()).log(Level.SEVERE, null,
ex);
        }
    };
}
```

Gambar 4.11 SourceCode kelas Model

Package model berisi class yang akan menampung atribut-atribut dari entitas yang sudah ditentukan (pasien,dokter,dll). Semua class tersebut memiliki tiga method, yaitu write, edit, delete. ketiga method tersebut berisi method untuk membuat, menyunting,dan menghapus record, sesuai entitasnya masing-masing. Diatas adalah contoh class dari package model

#### 4.2.7 Kelas Update

```
package CRUD;
import java.sql.Connection;
import java.sql.SQLException;
public class Update {
    public String update(String sql) throws SQLException {
        Config config = new Config();
        Connection kon = config.configDB();
        java.sql.Statement st = kon.createStatement();
        st.executeUpdate(sql);
        return "berhasil dibuat!";
    }
}
```

Gambar 4.12 SourceCode kelas Update

Update adalah kelas untuk memasukan data baru ke sebuah data base. parameter masukan fungsinya adalah sintaks sql yang ditaruh dalam variabel bertipe String. parameter masukan akan digunakan untuk mengeksekusi method execute statement dari library sql java

#### 4.2.8 Kelas Edit

```
public void edit(){
    String sql = "update asuransi \n";
    String koma = ",";
    String petik = "\"";
    sql = sql.concat("set nama_asuransi="+ petik +nama +petik + koma);
    sql = sql.concat("potongan="+ ptng +"\n");
    sql = sql.concat("where no_asuransi =");
    sql = sql.concat(id);
    System.out.print(sql);
    try {
        System.out.print(new Update().update(sql));
    } catch (SQLException ex) { Logger.getLogger(Pasien.class.getName()).log(Level.SEVERE, null,
ex);
    }
};
}
```

Gambar 4.13 Contoh SourceCode kelas Edit

Edit adalah kelas untuk memperbaharui data yang ada di sebuah data base. parameter masukan fungsinya adalah sintaks sql yang ditaruh dalam variabel bertipe String. parameter masukan akan digunakan untuk mengeksekusi method execute statement dari library sql java

#### 4.2.9 Contoh Instansiasi Kelas Delete

```
public void delete(){  
    String sql = "delete from asuransi where no_asuransi=";  
    sql = sql.concat(id);  
  
    try {  
        System.out.print(new Update().update(sql));  
    } catch (SQLException ex) {  
        Logger.getLogger(Pasien.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

Gambar 4.14 Contoh SourceCode kelas Delete

Delete adalah kelas untuk menghapus data yang ada di sebuah data base. parameter masukan fungsinya adalah sintaks sql yang ditaruh dalam variabel bertipe String. parameter masukan akan digunakan untuk mengeksekusi method execute statement dari library sql java

## **BAB V**

### **PENUTUP**

Pada bagian ini akan dibahas mengenai kesimpulan dan saran dari analisis hasil pengerjaan kegiatan aplikasi pemerograman berorientasi objek dengan judul ‘resepsionis rumah sakit’.

#### **5.1 Kesimpulan**

Setelah melakukan pembuatan dan pengembangan aplikasi pemerograman berorientasi objek dengan judul ‘respisionis rumah sakit’, dapat disimpulkan bahwa :

1. Dapat menghubungkan aplikasi dengan basis data
2. Dapat menggunakan tampilan berupa GUI
3. Dapat menghasilkan aplikasi yang baik & bermanfaat dalam pengolahan data dengan system CRUD
4. Dapat mengetahui manfaat dari aplikasi pemerograman berorientasi objek dalam pengolahan data
5. Dapat mengetahui proses perancangan system aplikasi pemerograman berorientasi objek
6. Dapat mengetahui penerapan pemerograman berorientasi objek khususnya Bahasa pemerograman java



## 5.2 Saran

Setelah melakukan pembuatan dan pengembangan aplikasi pemerograman berorientasi objek dengan judul ‘respsionis rumah sakit’, saran yang dapat dipertimbangkan kedepannya :

1. Dapat menggunakan java IDE yang lebih baik seperti Apache NeatBeans maupun yang lainnya
2. Membuat system CRUD dengan menghubungkan 1 table dengan 1 tampilan

## Daftar Pustaka

Adi Nugroho (2008) Pemrograman Java menggunakan Eclipse IDE. Yogyakarta,Indonesia : Penerbit Andi.

Bambang Haryanto (2007) Esensi-Esensi Bahasa Pemrograman Java .Bandung,Indonesia:Informatika.