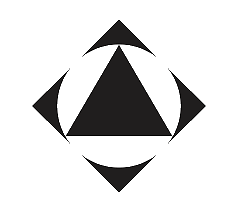
Aplikasi Pemerograman Berorientasi Objek

“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 pemerograman berorientasi objek, laporan ini berisi perancangan aplikasi serta fungsinya secara umum, serta berisikan kode pemerograman 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 pemerograman 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](#_Toc46419463)

[Daftar Isi 5](#_Toc46419464)

[BAB I](#_Toc46419465) [PENDAHULUAN 6](#_Toc46419466)

[1.1. Latar Belakang Masalah 6](#_Toc46419467)

[1.2. Tujuan Penelitian 8](#_Toc46419468)

[1.3. Rumusan Masalah 8](#_Toc46419469)

[1.4. Batasan Masalah 9](#_Toc46419470)

[1.5. Sistematika Penulisan 10](#_Toc46419471)

[BAB II](#_Toc46419472) [LANDASAN TEORI 11](#_Toc46419473)

[2.1 Konsep Dasar Sistem 11](#_Toc46419474)

[2.2 Pengertian Sistem Aplikasi Resepsionis Rumah Sakit 12](#_Toc46419475)

[2.3 Aplikasi 12](#_Toc46419476)

[2.4 Java 12](#_Toc46419477)

[2.5 NetBeans 13](#_Toc46419478)

[BAB III](#_Toc46419479) [PERANCANGAN 14](#_Toc46419480)

[3.1 Studi Kasus 14](#_Toc46419481)

[3.2 Fitur 15](#_Toc46419482)

[3.3 Diagram Alir 16](#_Toc46419483)

[3.4 Penjelasan Alur Proses 17](#_Toc46419484)

[3.5 Diagram Kelas 18](#_Toc46419485)

[3.6 Basis Data 19](#_Toc46419486)

[3.7 Sitemaps Aplikasi 21](#_Toc46419487)

[3.8 Rancangan Mock-Up UI 22](#_Toc46419488)

[BAB IV](#_Toc46419489) [IMPLEMENTASI 26](#_Toc46419490)

[4.1 Hasil Output 26](#_Toc46419491)

[4.2 Program & Penjelasan Program 31](#_Toc46419492)

[4.2.1 Kelas Config 31](#_Toc46419493)

[4.2.2 Kelas Create 32](#_Toc46419494)

[4.2.3 Kelas Read 33](#_Toc46419495)

[4.2.4 Contoh Instansiasi Kelas Create 34](#_Toc46419496)

[4.2.5 Contoh Instansiasi Kelas Read 35](#_Toc46419497)

[4.2.6 Kelas Model 36](#_Toc46419498)

[4.2.7 Kelas Update 38](#_Toc46419499)

[4.2.8 Kelas Edit 39](#_Toc46419500)

[4.2.9 Contoh Instansiasi Kelas Delete 40](#_Toc46419501)

[BAB V](#_Toc46419502) [PENUTUP 41](#_Toc46419503)

[5.1 Kesimpulan 41](#_Toc46419504)

[5.2 Saran 42](#_Toc46419505)

[Daftar Pustaka 43](#_Toc46419506)

# **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’.

## 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 pemikirian 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.

## Tujuan Penelitian

Tujuan yang ingin dicapai, sebagai berikut :

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

## Rumusan Masalah

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

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

## 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 pemerograman Java dan IDE NetBeans
5. Tampilan pengguna berupa gui
6. Menggunakan system CRUD

## Sistematika Penulisan

1. Halaman Judul

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

1. Halaman Daftar Isi

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

1. 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.

1. Halaman Bab 1 Pendahuluan

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

1. Halaman Bab 2 Landasan Teori

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

1. Halaman Bab 3 Perancangan

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

1. Halaman Bab 4 Implementasi

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

1. Halaman Bab 5 Penutup

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

1. 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 pemerograman berorientasi objek dengan judul ‘resepsionis rumah sakit’.

## Konsep Dasar Sistem

Suatu sistem pada dasarnya marupakan 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 memudah kan pelaksanaan kegiatan utama dari suatu organisasi.

## 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 mengguanakan system aplikasi.

## Aplikasi

Aplikasi merupakan terapan bahasa pemerograman 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 smartphone yang di rancang sedemikian rupa agar bisa mengolah suatu masukan (input) menjadi keluaran (output).

## 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 pemerograman 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.

## 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’.

## 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 khusunya pengolahan data.

## 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

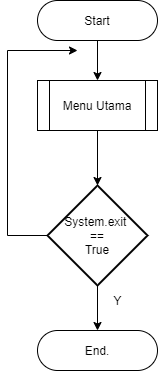
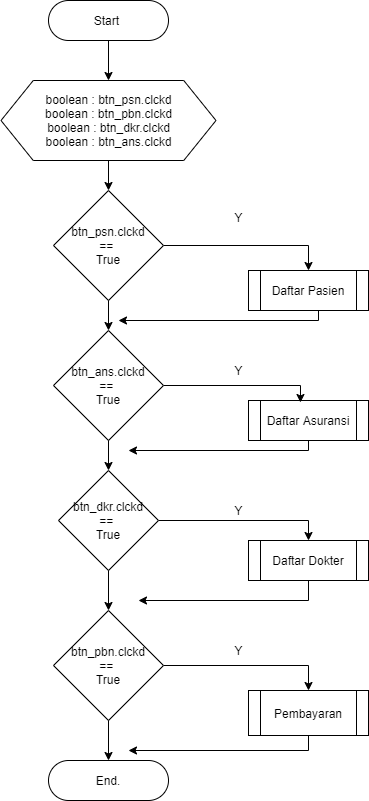
Fitur ini merupakan kemampuan system aplikasi agar pengguna dapat menambahkan data, mengubah data, menghapus data dan menampilkan data pada data pasien, dokter maupun asuransi.

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.

## Diagram Alir

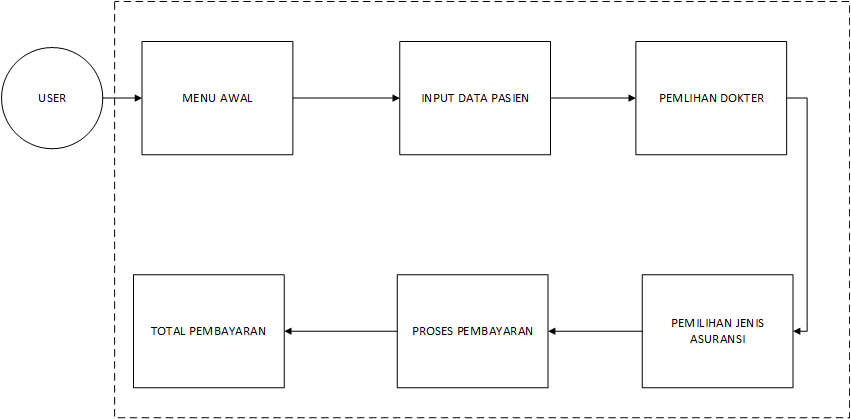
Aplikasi resepsionis rumah sakit ini memiliki rancangan aplikasi, sebagai berikut:

Gambar 3.1 Sistem Kerja Aplikasi

## Penjelasan Alur Proses

Pengguna aplikasi ini berupa resepsionis pada rumah sakit, resepsionis bertugas untuk mengatur masuknya data pasien, dokter dan data asuransi, setelah memasukan 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

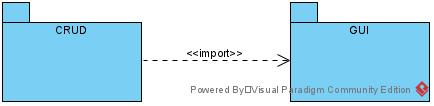
2. Menentukan jadwal dokter beserta antrian pasien

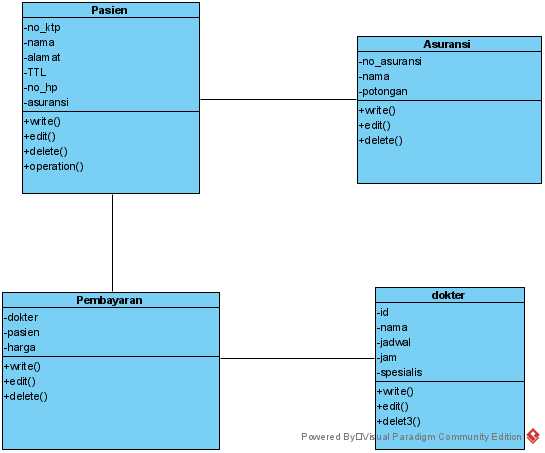
3. Pasien melakukan cek oleh dokter

4. Melakukan proses administrasi

## Diagram Kelas

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



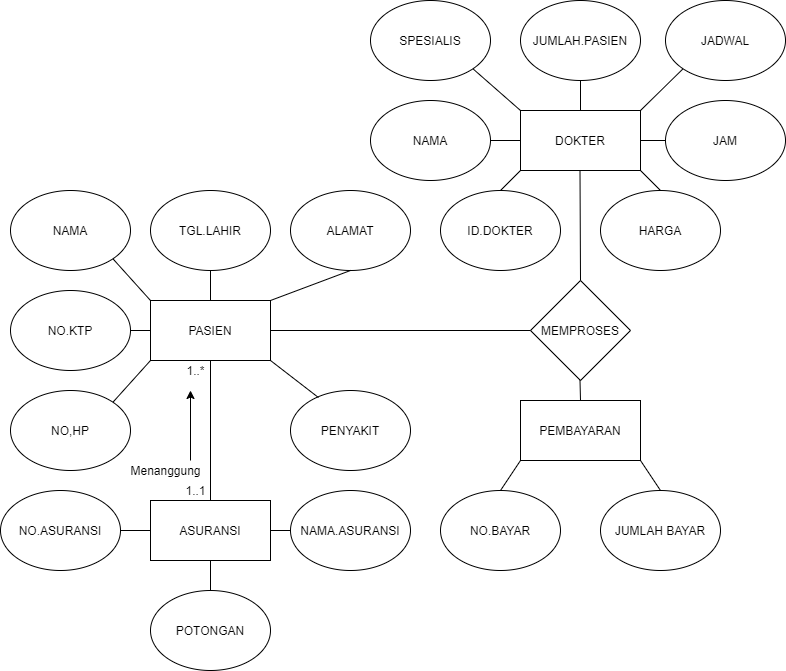


Gambar 3.3 UML Aplikasi

## Basis Data

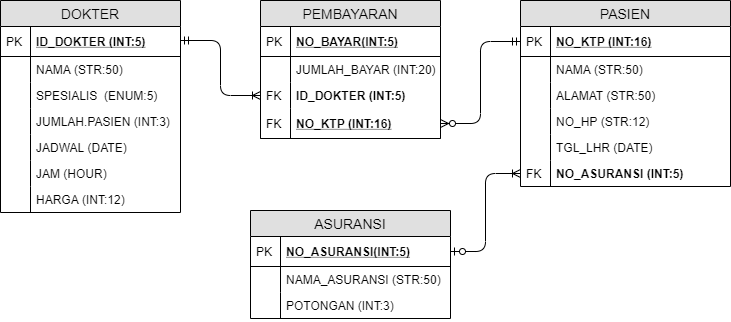
Aplikasi resepsionis rumah sakit ini memiliki rancangan basis data, sebagai berikut:

3.6.1 ERD



Gambar 3.4 ERD Aplikasi

3.6.2 TRD



Gambar 3.5 TRD Aplikasi

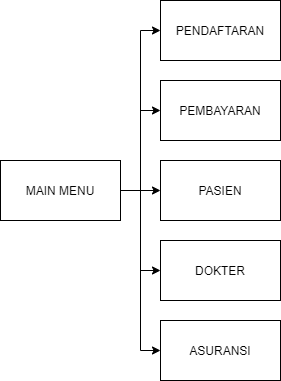
3.6.3 DDS



Gambar 3.6 DDS Aplikasi

## Sitemaps Aplikasi

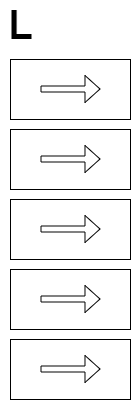
Aplikasi resepsionis rumah sakit ini memiliki rancangan penggunaan, sebagai berikut:



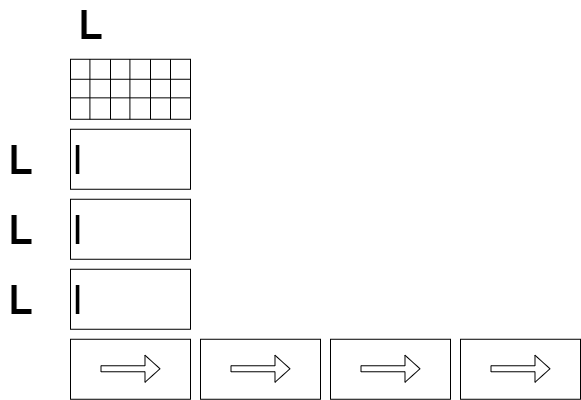
Gambar 3.7 Sitemaps Aplikasi

## Rancangan Mock-Up UI

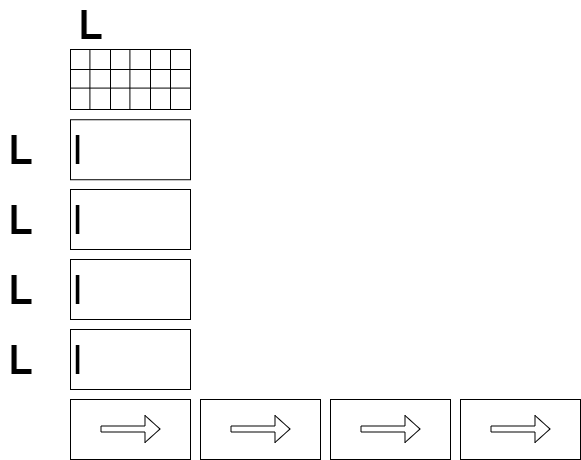
Aplikasi resepsionis rumah sakit ini memiliki rancangan tampilan pengguna, sebagai berikut:



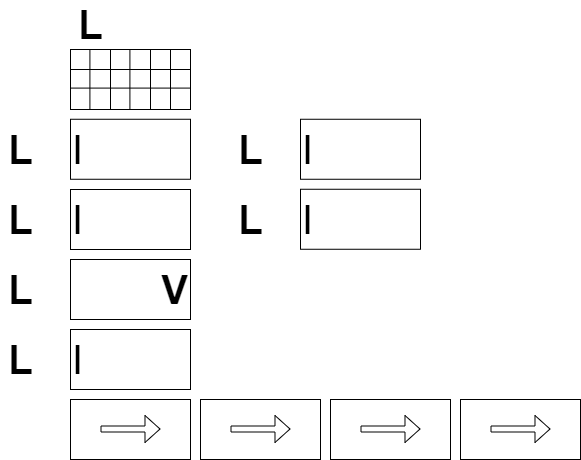
Gambar 3.8 Mockup Main Menu



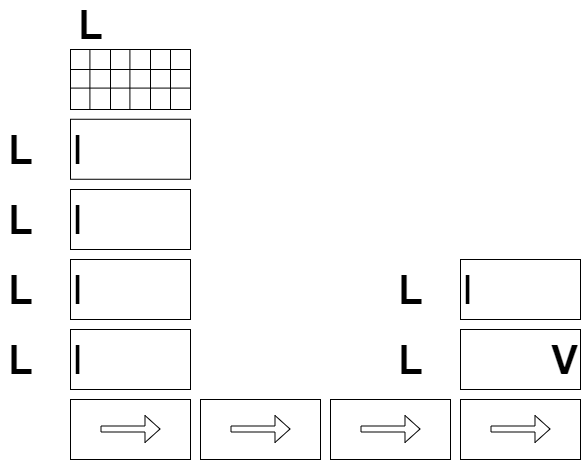
Gambar 3.9 Mockup Data Asuransi



Gambar 3.10 Mockup Data Bayar



Gambar 3.11 Mockup Data Dokter



Gambar 3.12 Mockup Data Pasien

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 Asuansi’. 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 disi antara lain no. KTP, nama, ttl, no.HP alamat dan jenis asuansi. Untuk menyimpan data dapat dilakukan dengan menekan button save. Selain dilakukan pengisian data pasien dapat dilihat juga data pien 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 dapa rumah sakit tersebut (Gambar3.8) dan ditampilakn 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.

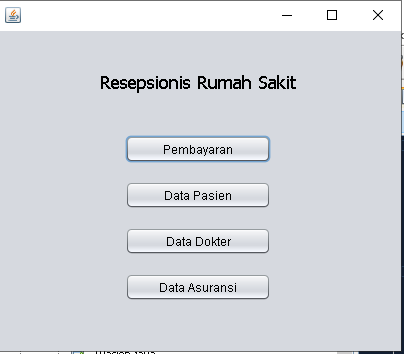
# **BAB IV**

# **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’.

## Hasil Output

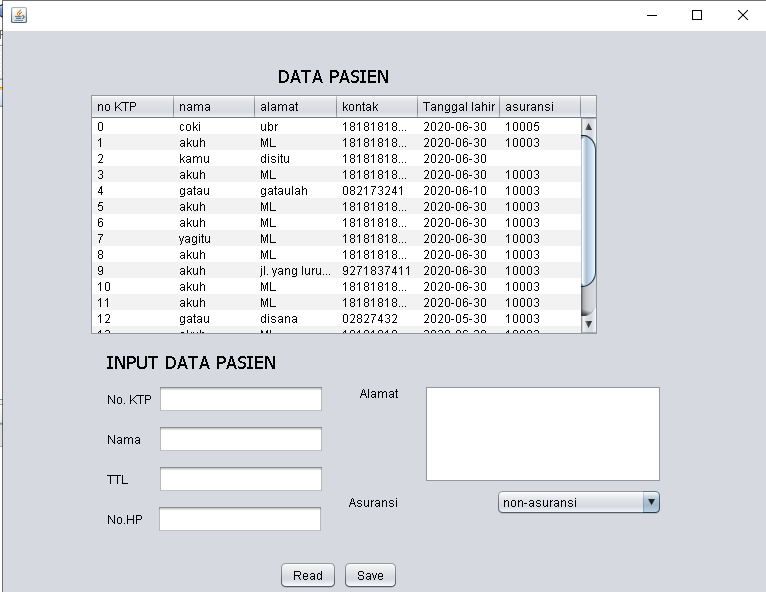
Setelah melakukan pembuatan dan pengembangan aplikasi resepsionis rumah sakit ini memiliki hasil tampilan, sebagai berikut:



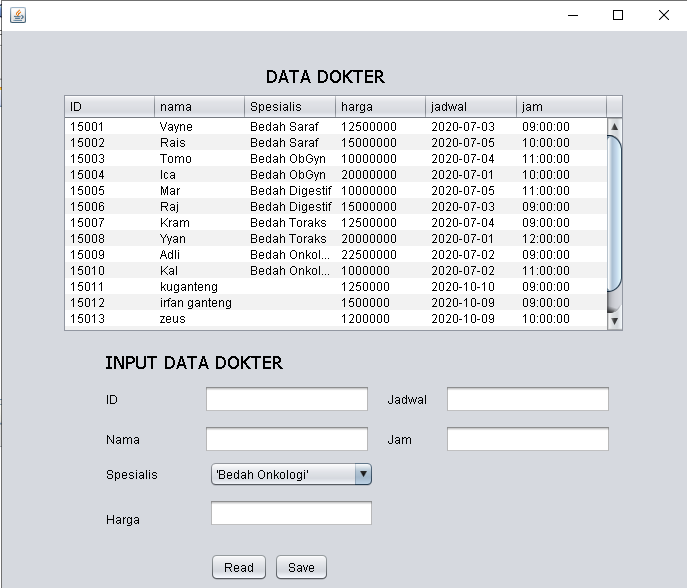
Gambar 4.1 Tampilan Main Menu



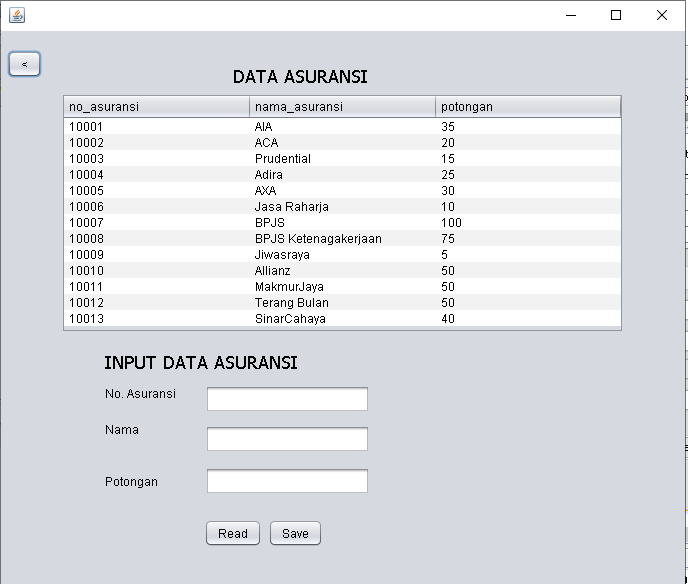
Gambar 4.2 Tampilan Bagian Pembayaran



Gambar 4.3 Tampilan Bagian Input Data Pasien



Gambar 4.4 Tampilan Bagian Input Data Dokter



Gambar 4.5 Tampilan Bagian Input Data Asuransi

## 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.getString("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 analisi hasil pengerjaan kegiatan aplikasi pemerograman berorientasi objek dengan judul ‘resepsionis rumah sakit’.

## Kesimpulan

Setelah melakukan pembuatan dan pengembangan aplikasi pemerograman berorientasi objek dengan judul ‘respsionis 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 pengolaan 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

## 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 mengubungkan 1 table dengan 1 tampilan

# Daftar Pustaka

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

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