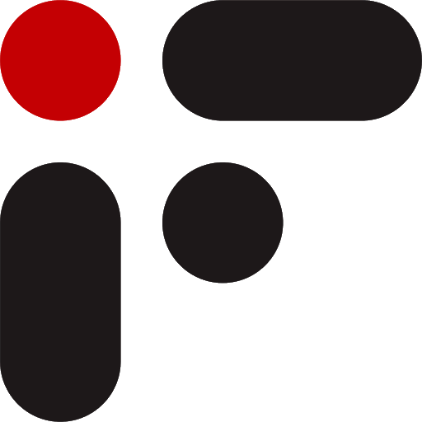
**TUGAS AKHIR**

“POLIKLINIK”

*Diajukan untuk memenuhi tugas akhir mata kuliah Pemrograman Berorientasi Objek*

****

DISUSUN OLEH :

FARIS SUMAWIJAYA 152018076

RAYYAN 152018079

M.RAZHALDY IHZA.D 152018086

M.AMMAR NUGRAHA 152018091

KELOMPOK : C5

KELAS : C

TGL PENUGASAN : 26 JUNI 2020

**LABORATORIUM BASIS DATA**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

**JURUSAN TEKNIK INFORMATIKA**

**INSTITUT TEKNOLOGI NASIONAL**

**BANDUNG**

**2020**

**Kata Pengantar**

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa yang atas rahmat dan karunia – Nya penulis dapat menyelesaikan project akhir yang berjudul “**Poliklinik”**.

Terimakasih kepada asisten laboratorium dan rekan – rekan yang telah membantu penulis dalam pengerjaan project akhir ini. Kami sangat berharap laporan ini dapat berguna dalam rangka menambah wawasan bagi pembaca dan penulis.

Semoga laporan project akhir ini dapat dipahami oleh pembaca. Penulis menyadari bahwa pada laporan ini terdapat kekurangan. Oleh karena itu, penulis berharap adanya kritik dan saran yang membangun dari pembaca mengigat tidak ada sesuatu yang sempurna tanpa adanya kritik dan saran yang membangun.

Bandung, Juli 2020

Penulis

**DAFTAR ISI**

**DAFTAR GAMBAR**

**BAB I**

**PENDAHULUAN**

Pada bab ini membahas mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, sistematika penulisan mengenai project akhir pemrograman berorientasi objek. **1.1. Latar Belakang**

Poliklinik merupakan tempat pelayanan yang bertugas melakukan pemeriksaan pasien secara umum dengan melihat indikasi atau gejala – gejala yang di derita oleh pasien. Poliklik Merupakan tempat pelayanan yang bertugas melakukan penanganan dan perawatan medis serta seleksi terhadap pasien. Pengunjung dilayani oleh Dokter Umum dan Perawat yang bertugas melakukan anamnesa dan diagnosa awal terhadap pasien. Di sistem Poliklinik saat ini terdapat 2 Pelayanan yaitu Poli Umum dan Poli Spesialis untuk menangani masalah penyakit yang tidak ada di dokter umum. Sistem poliklinik ini bertanggung jawab atas setiap pasien yang diperiksa agar dapat memberikan pelayanan yang sesuai setandar dan dapat di pertanggung jawabkan dalam mendiagnosa setiap keluahan setiap pasien yang berkunjung dipoliklinik tersebut. Data pemeriksaan diinput berdasarkan data pasien yang bersangkutan jadi data keluaran obat dan pemeriksaan berdasarkan data pasien yang terdata dalam data di poliklinik tersebut. Poliklinik juga dapat memberikan rujukan rawat jalan dan rawat inap dengan memberikan surat rujukan untuk mengajukan rujukan ke rumah sakit, setelah ada data diagnosa dari poliklinik. Setelah Dokter dari klinik mendiagnosa penyakit pasien, pasien dirujuk ke rumah sakit yang telah ditentukan oleh dokter atau perawat dan telah di rekomendasikan oleh kepala poliklinik.

**1.2. Rumusan Masalah**

Berikut rumusan masalah atau ruang lingkup aplikasi yaitu diantaranya:

1. Apa manfaat dan fungsi pembuatan aplikasi tersebut?
2. Bagaimana cara kerja sistem tersebut?

**1.3. Tujuan**

Tujuan yang akan dicapai dalam penulisan laporan ini adalah :

1. Memudahkan pasien poliklinik dalam konsultasi.
2. Memudahkan pasien yang ingin melakukan pemeriksaan kesehatan kepada dokter umum maupun spesialis.

**1.4. Batasan Masalah**

Adapun batasan masalah nya, yaitu :

1. Pengerjaan aplikasi dikerjakan secara online, karena kasus covid 19 sehinngaa sulit utk mendiskusikannya.
2. Pengidentifikasian suatu penyakit hanya menggunakan pilihan item dari combobox.
3. Aplikasi hanya bisa dioperasikan oleh pegawai poliklinik.
4. Keterbatasan kuota, karena kita melakukan nya secara online jadi sering dari kita itu terkendala dengan keterbatasan kuota.

**1.5. Sistematika Penulisan**

1. Pada bab I, menjelaskan mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, dan sistematika penulisan.
2. Pada bab II, membahas mengenai teori – teori dari segenap proses pemrograman berorientasi objek
3. Pada bab III, menjelaskan mengenai studi kasus, fitur, role user, Alur proses, class diagram, database, dan Mockup UI
4. Pada bab IV, menjelaskan mengenai UI, dan Source Code
5. Pada bab V, menjelaskan mengenai Kesimpulan, dan saran

**BAB II**

**LANDASAN TEORI**

* 1. **Java**

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995.

* 1. **Netbeans**

NetBeans adalah suatu serambi pengembangan perangkat lunak yang ditulis dalam bahasa pemrograman Java. Serambi Pada NetBeans, pengembangan suatu aplikasi dapat dilakukan dimulai dari setelan perangkat lunak modular bernama modules. Semula, aplikasi NetBeans IDE ini diperuntukkan bagi pengembangan dalam Java.

* 1. **SQL**

SQL adalah sebuah bahasa yang digunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara de facto merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya.

* 1. **Komponen GUI Netbeans**

GUI (Graphical User Interface) adalah tampilan tatap muka dalam sebuah aplikasi, tentu saja GUI ini memiliki komponen-komponen tersendiri, berikut adalah komponen – komponen yang kelompok kita pakai di GUI netbeans, antara lain

* **Combo box**

adalah sebuah Compunent yang ada di Java Netbeans, dimana fungsinya untuk sebuah Pemilihan, seperti pemilihan Jenis Kelamin, atau Negara, atau Bahasa

* **Jlabel**

JLabel, berfungsi sebagai penunjuk tulisan/angka pada GUI.

* **Jform**

Jfrom adalah Sebuah frrom atau jendela utama yang di gunakan untuk kita menampilkan komponen yang di inginkan pada layar, JForm juga digunakan untuk kita mendesain tampilan model yang didalamnya bisa kita menetapkan komponen lainnya.

* **Jbutton**

JButton, berfungsi sebagai tombol pada GUI, komponen ini paling sering digunakan karena dibutuhkan user untuk menklik tombol tersebut

* **JTabel**

JTable adalah kelas turunan dari kelas JComponent. Kelas JTable digunakan untuk menampilkan atau merubah data yang disajikan dalam format baris dan kolom (two-dimensional grid). JTable dapat menampilkan data dalam jumlah banyak.

**BAB III**

**Perancangan**

**3.1 Studi Kasus**

Poliklinik adalah salah satu unit pelayanan masyarakat yang bergerak pada bidang Kesehatan. Sebuah poliklinik yang menawarkan fasilitas perawatan kesehatan yang di khususkan untuk perawatan pasien rawat jalan. Pada pengerjaan aplikasi ini, aplikasi dibuat bertujuan agar user mudah untuk mencari dan memilih dokter spesialis dengan media aplikasi.

**3.2 Fitur**

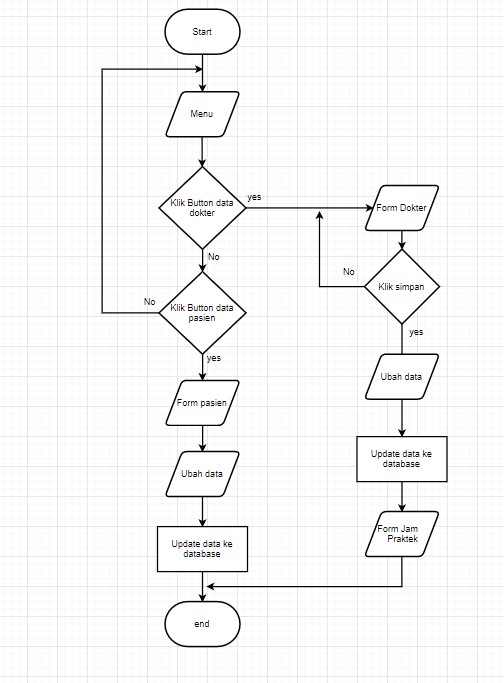
Aplikasi dapat menginput, menghapus dan mengedit sebuah data dan kemudian data tersebut masuk kedalam database melalui button yang telah disediakan, yaitu Button :

* Tambah data : Untuk menambahkan sebuah data.
* Simpan : Untuk menyimpan sebuah data yang telah diinput kedalam database.
* Edit : Untuk mengubah sebuah data melalui aplikasi.
* Hapus : Untuk menghapus sebuah data.
* Keluar : Untuk Keluar dari aplikasi.

**3.3 Role User**

Role user pada aplikasi ini adalah petugas manajemen data pada poliklinik yang bertujuan untuk memberi informasi dan mengubah sebuah informasi dari data dokter, pasien dan jam praktek.

**3.4 Alur Proses**

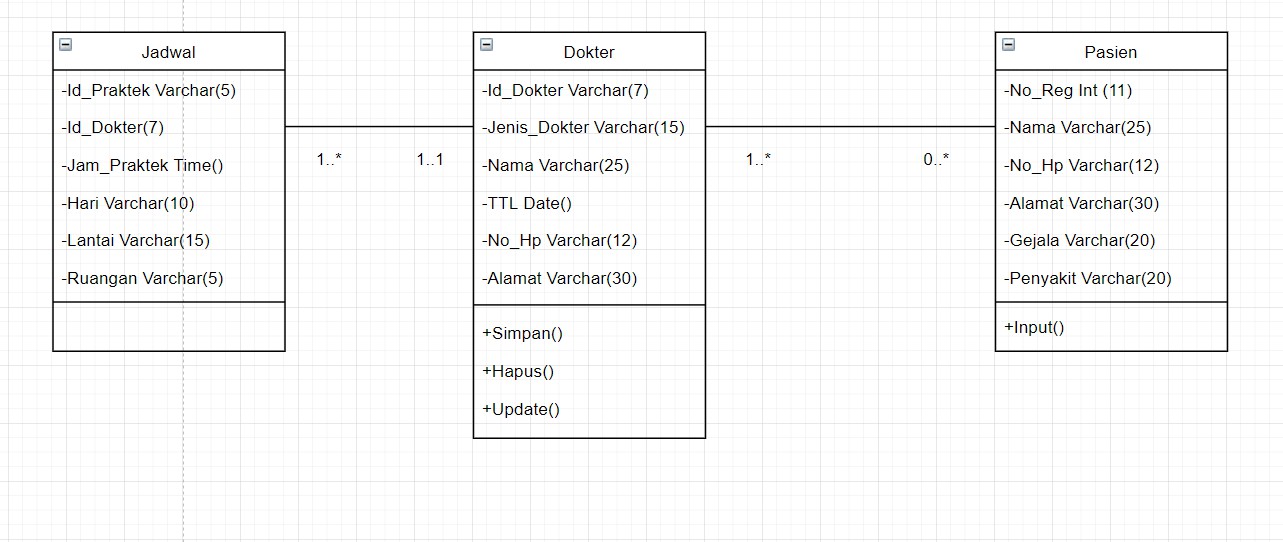
****

Gambar 3.3.1 GUI

Mula-mula, user diwajibkan untuk menekan button *Tambah data* lalu mengisi sebuah data yang telah disediakan pada edit box dan combo box, lalu user diperlukan menyimpan data agar data yang telah diisi tersimpan. User dapat menghapus data dengan cara menekan salah satu data pada sebuah tabel dan kemudian tekan button *hapus.* User dapat mengedit sebuah data dengan cara menekan salah satu data pada tabel lalu menekan button *Edit* untuk mengganti/memperbaharui sebuah data. Kemudian terdapat tombol *Keluar* untuk keluar dari sebuah aplkasi.

**3.5 Class Diagram**

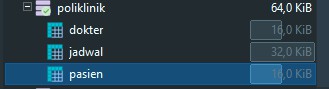
Diagram kelas adalah diagram UML yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain, serta dimasukkan pula atribut dan operasi. Berikut adalah sebuah UML pada aplikasi poliklinik.



Gambar 3.4.1 UML

**3.6 Database**

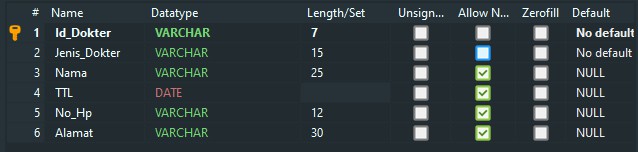
Database adalah kumpulan data yang terorganisir, yang umumnya disimpan dan diakses secara elektronik dari suatu sistem komputer. Pada saat pangkalan data menjadi semakin kompleks, maka pangkalan data dikembangkan menggunakan teknik perancangan dan pemodelan secara formal. Pada aplikasi ini kami mengisi data base sebagai berikut :

****

Gambar 3.5.1 Database

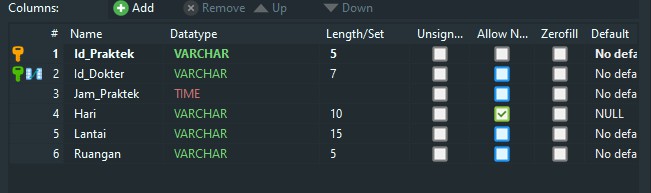
Database diberi nama poliklinik yang memiliki 3 tabel, yaitu dokter, jadwal, dan pasien,

a. Tabel Dokter



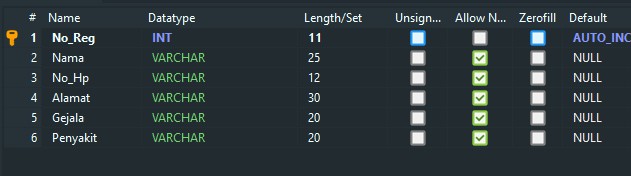
Gambar 3.5.2

b. Tabel Jadwal



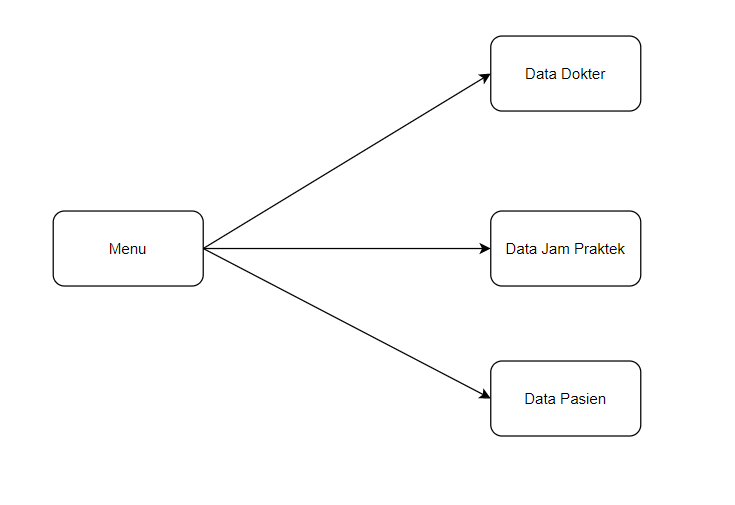
Gambar 3.5.3

c. Tabel Pasien



Gambar 3.5.3

**3.6 Sitemaps**

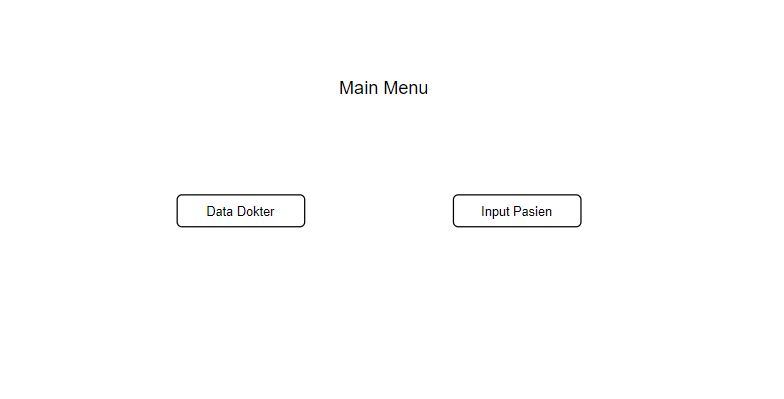


Gambar 3.6.1

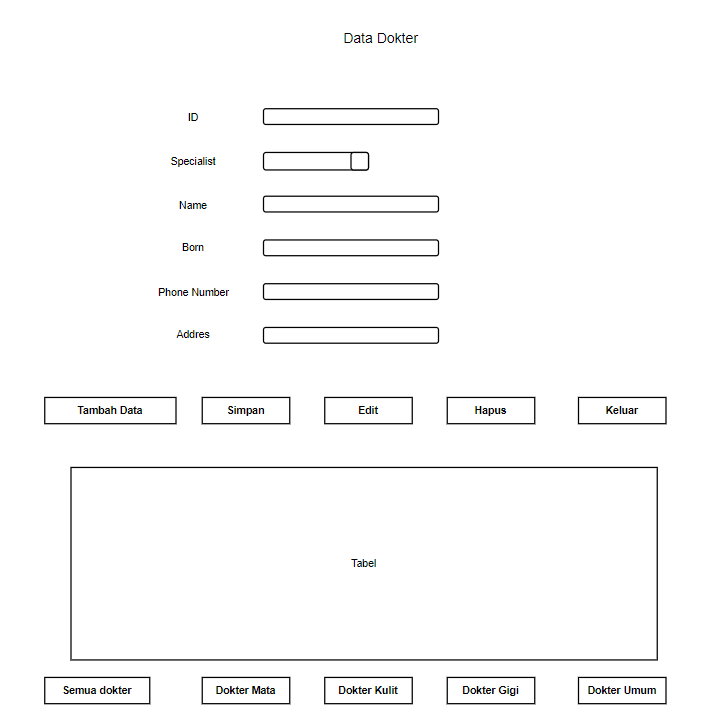
Terdapat 3 pilihan pada main menu, setelah melewati menu, user akan memilih salah satu antara membuka data dokter, data jam praktek, dan data pasien.

**3.7 Mock Up**

a. Main Menu

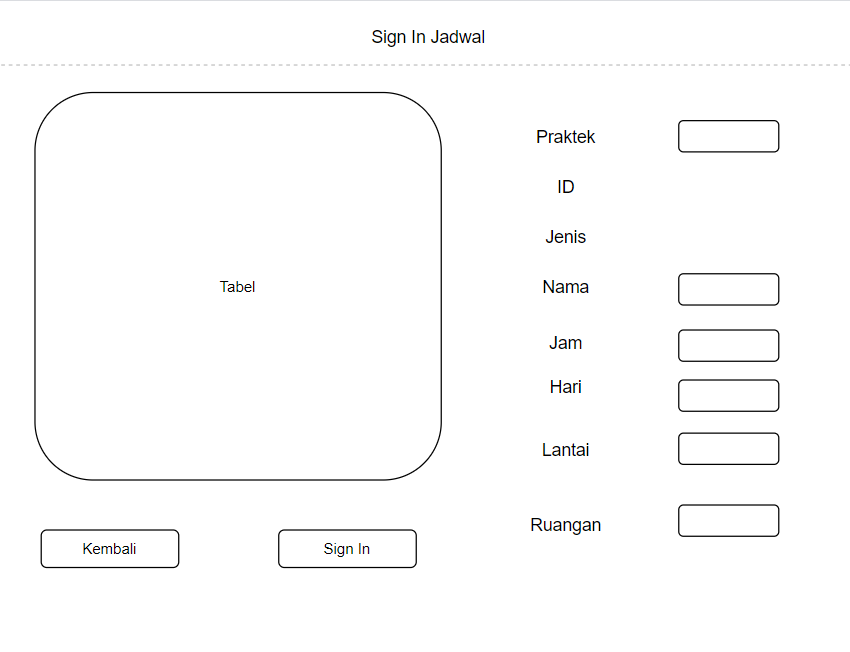


Gambar 3.7.1 Main Menu

b. Data Dokter

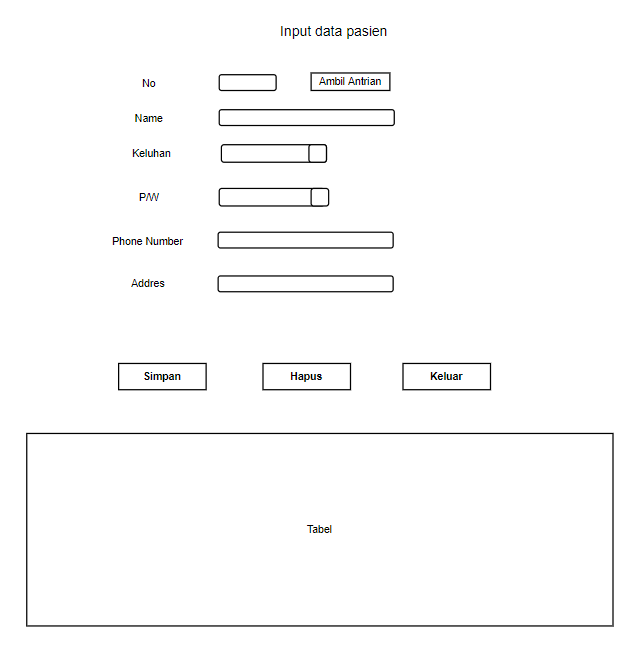
Gambar 3.7.2 Data Dokter

c. Sign Jadwal



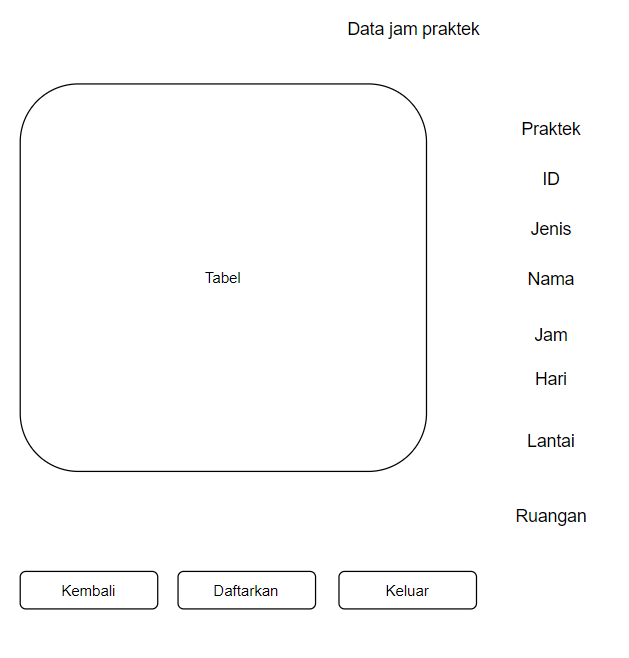
Gambar 3.7.3 Sign In Jadwal

d. Input Data Pasien



Gambar 3.7.4 Input Data Pasien

e. Data Jam Praktek



Gambar 3.7.5 Data Jam Praktek

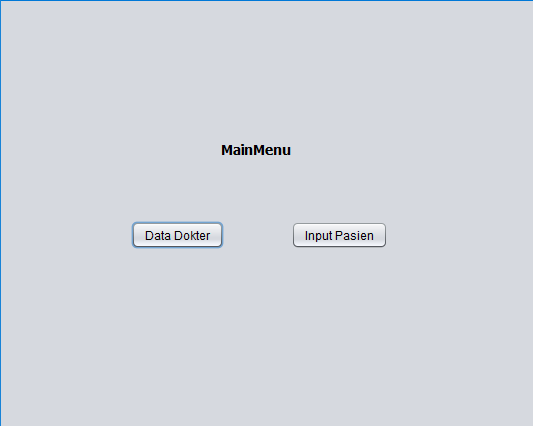
**BAB IV**

**IMPLEMENTASI**

* 1. **UI**

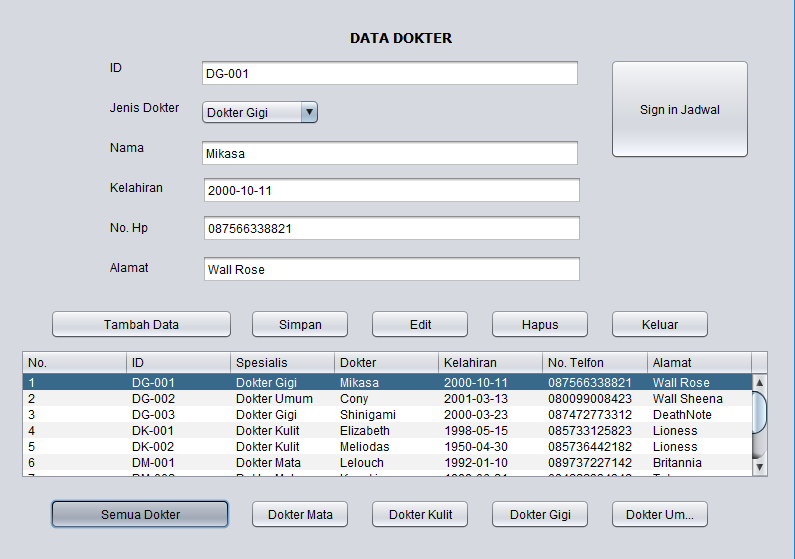
1. **Form main menu**

Pada form main menu ini terdapat dua pilihan untuk melanjutkan ke form selanjutnya yaitu *form data dokter* dan *form input pasien,* jika *user* menekan *button* *data dokter* maka halaman akan berpindah ke *form data dokter*, sebaliknya jika *user* menekan *button input pasien* maka halaman akan berpindah ke *form input pasien*.

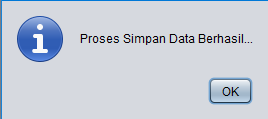


1. **Form dokter**

Pada *form* ini *user* dihadirkan sebuah rentetan komponen untuk mengisikan data diri dokter diantaranya id, jenis dokter, nama dokter, kelahiran, nomor handphone serta alamat. Selain itu juga terdapat *button* filter, simpan, edit, hapus, keluar yang fungsinya untuk *CRUD(create, read, update, delete)*



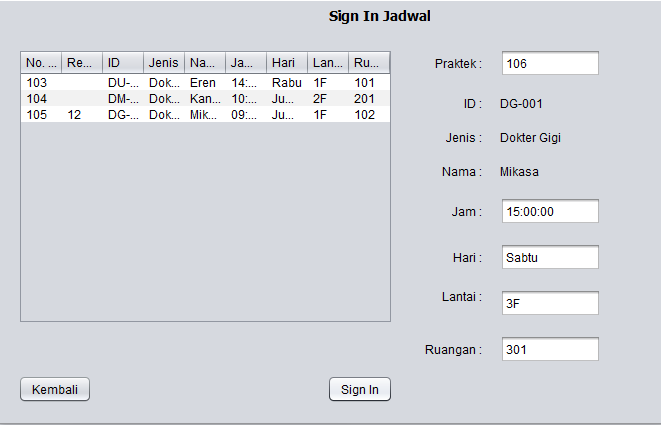
Untuk *button tambah* *data* berfungsi untuk menghapus semua isi nilaipada *textfield* yang terdapat pada *form data dokter* itu sendiri, *button simpan* berfungsi untuk menyimpan semua isi *textfield* ke dalam basis data “*dokter”, button edit* berfungsi untuk memanipulasi data pada semua *textfield* yang dapat diisi kembali isinya ketika *user* mengklik salah satu baris yang terdapat pada *table* yang berisikan data-data dokter yang terdapat pada basis data, selanjutnya terdapat *button keluar* yang berfungsi untuk menutup form dan kembali lagi ke form *main menu,* terakhir pada bagian bawah *table* terdapat rentetan *button* yang fungsinya untuk memfilter isi tabel sesuai dengan nilai jenis / spesialis dokter itu sendiri.



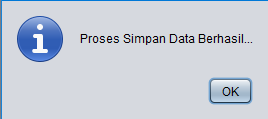
Setelah *user* menginput semua data diri dengan *textfield* serta *combobox* yang telah disediakan maka akan muncul *popup message* seperti pada gambar di atas, dan akan dialihkan menuju ke form selanjutnya dengan menekan *button sign in jadwal* nama formnya yaitu *form sign in jadwal,* sebagai contoh implementasi diambil sebuah data dokter bernama *Mikasa,* dengan spesialis dokter gigi, dan id DG-001*.*

1. **Form sign in jadwal**

Selanjutnya pada form ini *user* dihadirkan dengan form yang berfungsi untuk menginput sebuah jadwal ke dalam basis data “*jadwal”*, pada form sebelumnya *user* dapat memilih data dokter mana yang akan melakukan sebuah praktek, setelah memilih data maka pada form ini sistem akan otomatis menginput id, jenis dokter dan nama dokter sesuai dengan data dokter terpilihyang terdapat pada form sebelumnya.



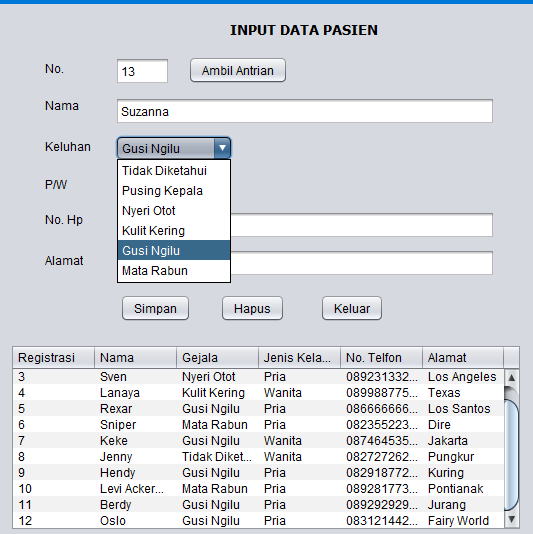
Sehingga pada form ini *user* hanya perlu menginput id praktek, jam, hari, lantai dan ruangan yang akan digunakan untuk dokter melakukan praktek, Ketika sudah menginput semua data yang diperlukan maka *user* diminta untuk menekan *button sign in* yang berfungsi untuk meninput semua data tersebut ke dalam basis data *“jadwal”*,



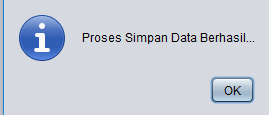
ketika data berhasil tersimpan maka akan muncul *pop up menu* sebagai berikut dan juga akan ditampilkan 1 baris semua data jadwal tanpa nomor *registrasi,* untuk field registrasi yang kosong sendiri berfungsi untuk mengidentifikasi bahwa jadwal tersebut masih tersedia atau belum ada pasien yang melakukan registrasi pada jadwal tersebut, sebaliknya jika field registrasi sudah terisi menandakan bahwa pada jadwal tersebut sudah terdapat pasien yang melakukan registrasi.

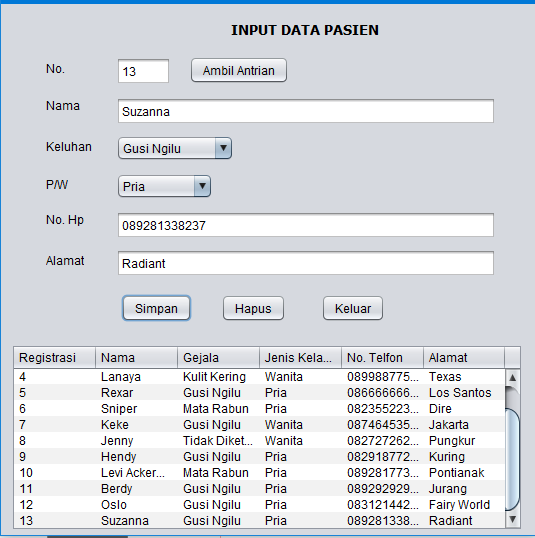
1. **Form input pasien**

Setelah kembali ke form *mainmenu* dan menekan *button input pasien* maka *user* akan memasuki form yang berfungsi untuk menginput data pasien, pada form ini *user* akan dihadapkan dengan rentetan *textfield, combobox, button-button dan sebuah tabel* yang fungsinya sendiri untuk menambahkan dan menghapus data pasien.



Ketika memasuki form ini *user* diminta untuk mengambil nomor antrian/registrasi dengan menekan *button ambil antrian* maka akan muncul nomor antrian yang berfungsi untuk mendaftarkan pasien ke jadwal dokter nantinyaa, *user* dapat mengisikan semua data pada *textfield* yang telah disediakan dan memilih salah satu keluhan yang pilihannya tersedia seperti gambar di atas, jika *user* memilih keluhan dengan spesifik tertentu maka pasien akan dialihkan ke dokter dengan spesialisasi tertentu juga, jika *user* memilih keluhan dengan spesifik tidak diketahui maka pasien akan dialihkan ke dokter umum.

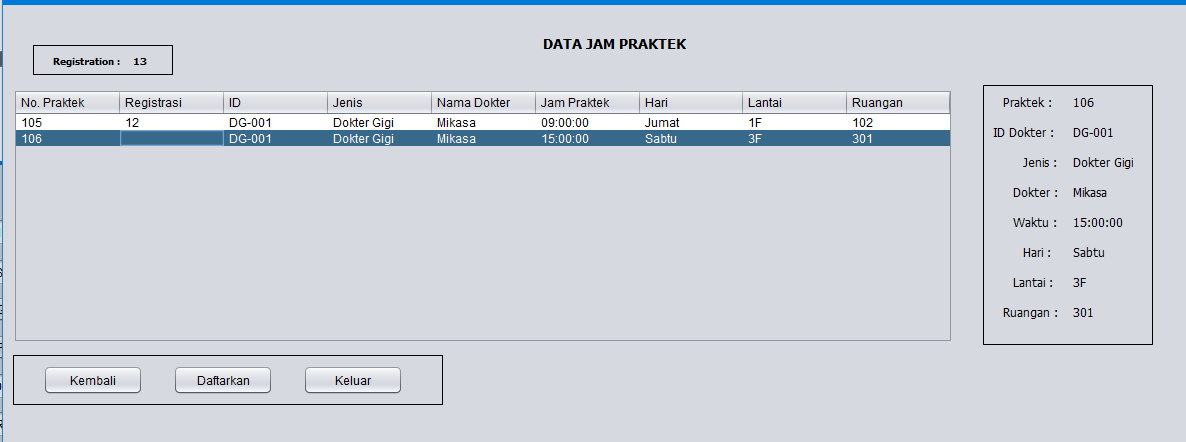




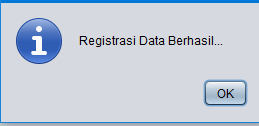
Setelah menginput semua data, maka *user* diminta untuk menekan *button simpan* yang berfungsi untuk menginput semua data pasien ke dalam basis data *“pasien”,* setelah itu akan muncul *popup message* dan data pasien yang telah diinput dapat dilihat pada tabel data pasien

1. **Form data jam praktek**

Setelah *user* menekan *button simpan* maka akan muncul *popup form data jam praktek* yang dimana form ini berfungsi untuk memilih salah satu list jadwal manakah yang diinginkan *user* untuk diinputkan dengan nomor registrasi pasien pada form sebelumnya



Pada form ini *user* diminta untuk mengklik pada salah satu baris pada tabel dan menekan *button daftarin* yang terdapat pada bagian form, pada bagian kanan form dapat dilihat spesifik data untuk jadwal praktek yang akan dipilih, dan bagian ujung kiri form terdapat nomor registrasi pasien sebagai contoh nomor registrasi 13 akan mengambil jadwal praktek 106.



Setelah menekan *button daftarin* maka nomor registrasi pasien akan diinputkan ke dalam database *“jadwal”* dan akan muncul *popup message* pada gambar di atas



Hasil penginputan nomor registrasi akan diupdate dan dapat dilihat pada tabel bahwa nomor register 13 telah terdaftar pada nomor praktek 106.

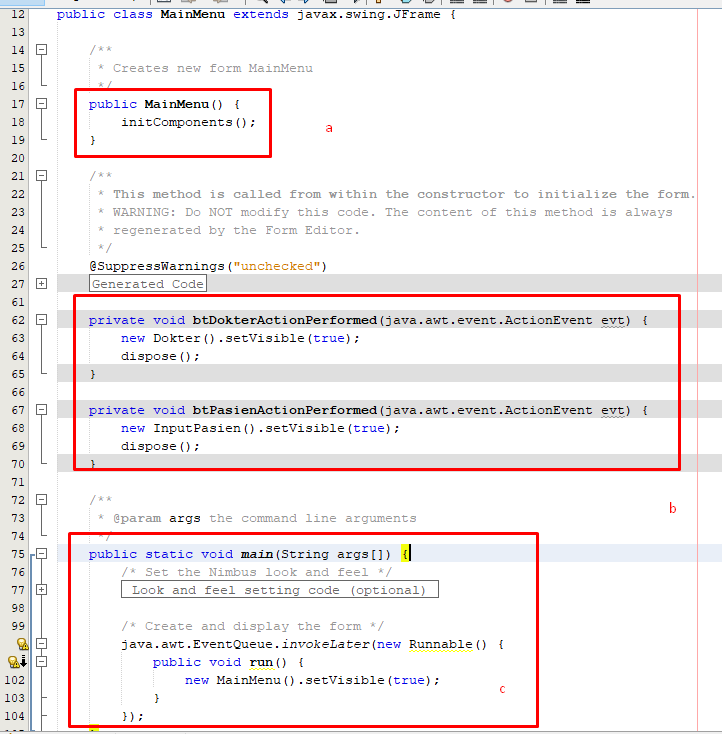
* 1. **Source Code**

1. **Konfig**



Khusus class konfig sendiri berfungsi untuk konfigurasi antara IDE dengan basis data yang akan dihubungkan nantinya, sehingga class ini akan dipanggil pada class lain untuk keperluan transaksi / keluar masuknya. Dapat dilihat pada baris 22-24 sudah diinisialisasi untuk masing-masing variable url berlaku sebagai untuk menunjuk tujuan basis data yang ingin diakses, serta user dan password berlaku sebagai hak akses basis datanya.

1. **Form main menu**

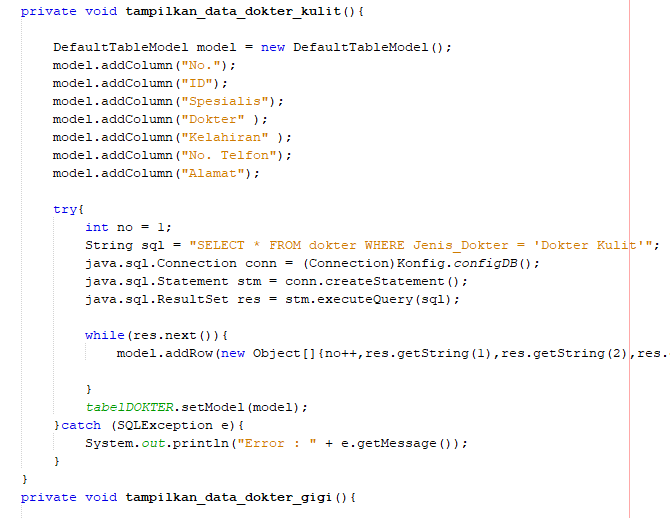


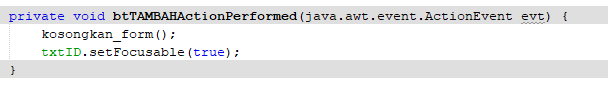
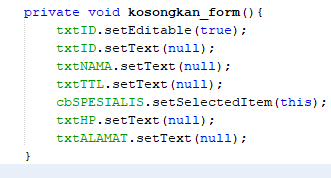
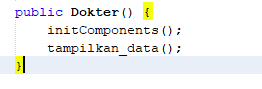
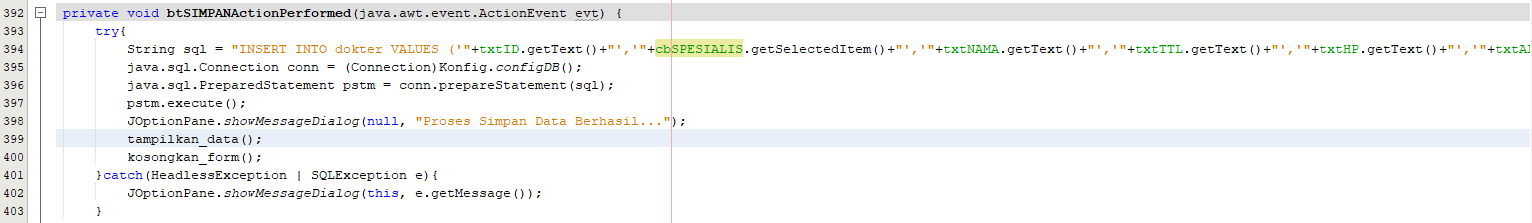
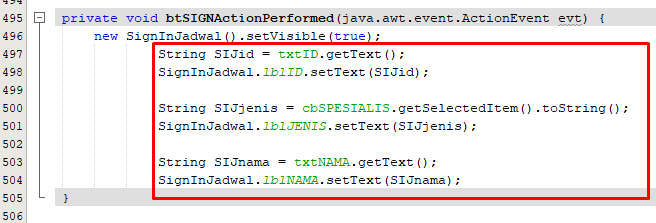
* 1. Merupakan sebuah constructor untuk *form main menu* yang berisikan method initComponents, initComponents sendiri berfungsi untuk menginisialisasi komponen-komponen JavaSwing secara default
  2. Merupakan method untuk akses *button data dokter* dan *button input pasien,* yang mana *setVisible* berfungsi untuk menampilkan form yang akan dipanggil
  3. Merupakan mainclass sehingga ketika proses *run* aplikasi akan menampilkan *form main menu*

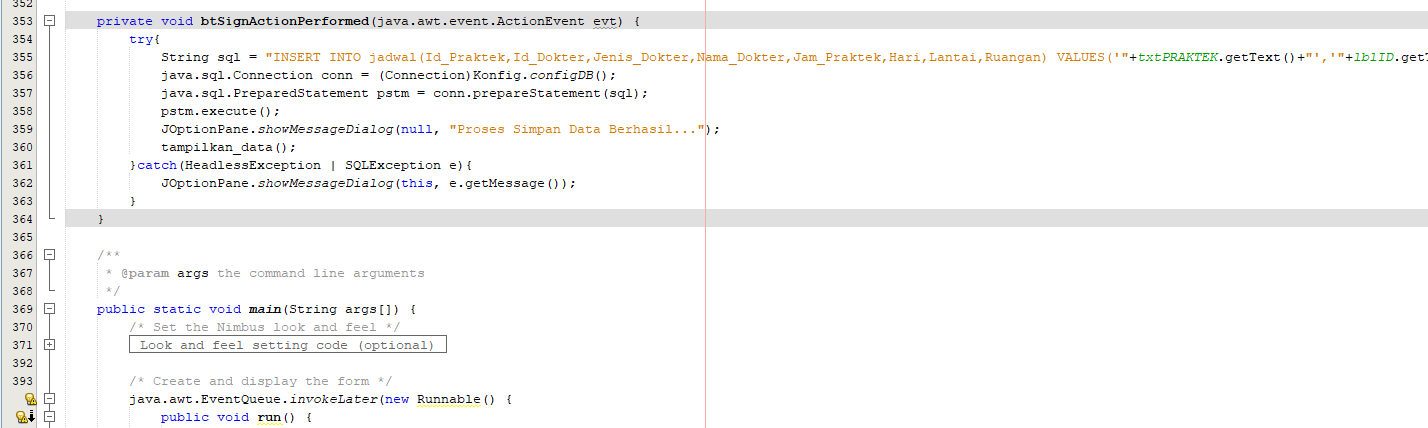
1. **Form data dokter**



1. Berfungsi untuk inisialisasi tabel beserta fieldnya
2. Berfungsi untuk menginisialisasi isi dari String sql dan menghubungkannya dengan class config untuk dihubungkan ke basis data, kemudian selanjutnya menambahkan format susunan yang terdapat pada basis data dengan field tabel yang sudah diinisialisasikan sebelumnya
3. Berfungsi untuk menampilkan *error message* bila terjadi kelasahan ketika menghubungkan dari IDE ke basis data

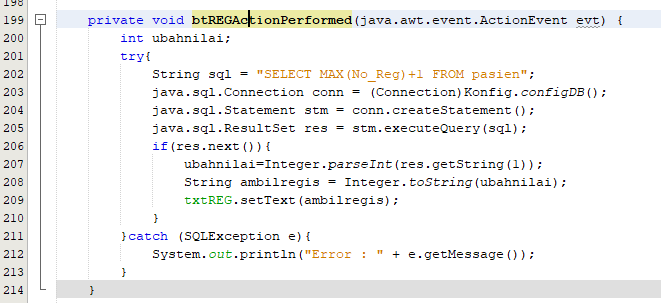


1. Untuk masing-masing jenis dokter juga dibuatkan methodnya dengan string sql yang berbeda, berfungsi untuk memfilter dokter-dokter tertentu  
     
   
2. Method pada *button tambah* terdapat sebuah method *kosongkan\_form()* yang berfungsi untuk mengosongkan setiap isi *textfield* pada form  
   
3. Pada constructor form ini terdapat method untuk menampilkan data pada tabel yaitu *tampilkan\_data();*
4. Source code di atas dapat dilihat pada baris 394 berfungsi untuk mengambil nilai yang terdapat pada setiap item seperti *textfield(obj.getText)* dan *combobox(obj.getSelectedItem)* ke dalam basis data dengan *query* yang tertera pada gambar di atas.  
   
5. Source code di atas merupakan fungsi ketika suatu tabel diklik maka isi-isi seperti *textfield* maupun *combobox* akan terisi sesuai dengan baris mana yang diklik *user,* pada baris 408 berfungsi untuk mengambil nilai baris yang terdapat pada tabel, yang nantinya pada baris 410 akan dipanggil variable tersebut untuk merepresentasikan baris ke-berapa serta *field* basis data yang disesuaikan dengan kolom yang terdapat pada tabel IDE  
   
6. Merupakan source code untuk *button update* yang mana sama saja seperti *button tambah* hanya saja *query* nya yang berbeda.
7. Merupakan source code untuk *button delete,* untuk *void* delete sendiri pada *query* nya hanya membutuhkan sebuah 1 field saja yaitu *primary key* untuk menghapus 1 deret baris record data pada tabel basis data.  
   
8. Berfungsi untuk memformat field *textfield* nama agar hanya dapat diisi oleh char huruf
9. Berfungsi untuk mematikan fungsi filter pada tabel
10. Berfungsi untuk memfilter semua record dokter yang berjenis dokter umum dengan *void* yang telah diinisialisasikan sebelumnya pada tabel
11. Berfungsi untuk memfilter semua record dokter yang berjenis dokter gigi dengan *void* yang telah diinisialisasikan sebelmnya pada tabel
12. Berfungsi untuk memfilter semua record dokter yang berjenis dokter kulit dengan *void* yang telah diinisialisasikan sebelumnya pada tabel
13. Berfungsi untuk memfilter semua record dokter yang berjenis dokter mata dengan *void* yang telah diinisialisasikan sebelumnya pada table  
    
14. Berfungsi untuk mengambil *value* dari *textfield* pada form dokter ke *label* pada form sign in jadwal.
15. **Form sign in jadwal**



Pada *button sign in jadwal* di form data dokter, dapat dilihat pada gambar di atas baris 355 dimana *query* yang digunakan kurang lebih hampir sama seperti yang digunakan untuk *button update* tetapi yang membuatnya berbeda disini tidak menginsert field register dikarenakan untuk keperluan mengosongkan field tersebut, sehingga *query* yang digunakanpun field spesifik.

1. **Form input pasien**



Pada *button registrasi* atau saat *user* mengambil nomor antrian *query* yang digunakanpun berbeda, yaitu dengan mengambil nilai maximal dari *field primary no\_reg* dan ditambahkan nilai 1, setelah itu dimasukkan *value* nya ke *textfield no\_reg.*

1. **Form data jam praktek**



Pada Constructor DataJamPraktek terdapat case dimana si sistem mengindentifikasi dari combobox keluhan yang terdapat pada form sebelumnya untuk diklasifikasikan. Jika *value* Tidak Diketahui, Pusing Kepala, Nyeri otot maka tampilka tabel jadwal dengan jenis dokter yang mana dokter umum, begitu juga yang lainnya.

**BAB IV**

**PENUTUP**

* 1. **KESIMPULAN**

Pada aplikasi “Poliklinik” ini masih terdapat beberapa kekurangan yang membuat pembuatan aplikasi menjadi tidak sempurna, namun aplikasi ini sudah dapat membantu mempermudah pasien yang ingin melakukan konsultasi di poliklinik maupun melakukan pemeriksaan (medical checkup), sasaran utama dalam pembuatan aplikasi ini adalah lebih dikhususkan untuk mempermudah petugas manajemen data pada poliklinik yang bertujuan untuk memberi informasi dan mengubah sebuah informasi dari data dokter, pasien, dan jam praktek melalui media aplikasi.

**5.2 Saran**

Dalam pembuatan aplikasi “Poliklinik” ini masih banyak yang harus diperbaiki baik dari alur kerja sistem hingga rancangan User Interfacenya, hal itu tentu akan menjadi pertimbangan untuk aplikasi “Poliklinik” ini sendiri bila aplikasi ini nantinya akan disebar luaskan lebih luas lagi sehingga dapat lebih banyak digunakan orang untuk mempermudah pekerjaan mereka dalam urusan pemeriksaan kesehatan.