

# **LAPORAN TUGAS BESAR**

## **SISTEM INFORMASI RUMAH SAKIT**

Untuk:

Menyelesaikan Tugas Besar Mata Kuliah IF2242

Dosen:

Arief Ichwani, S.Kom., M.Cs

Disusun oleh:

Desi Togi Sintauli Tambunan (119140073)

Fathimatul Mahmudzah (119140168)

Makruf Alkarkhi (119140075)

Muhammad Asyroful Nur Maulana Yusuf (119140026)

Reza Kusuma (119140001)



**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNIK ELEKTRO, INFORMATIKA, DAN SISTEM FISIKA**  
**INSTITUT TEKNOLOGI SUMATERA**  
**LAMPUNG SELATAN**  
**2021**

## DAFTAR ISI

DAFTAR ISI.....	2
DAFTAR GAMBAR .....	3
DAFTAR TABEL.....	4
BAB I.....	5
PENDAHULUAN .....	5
BAB II.....	7
RANCANGAN SISTEM.....	7
2.1 Deskripsi Sistem .....	7
2.2 Use Case Diagram.....	7
2.3 Rancangan Basis Data.....	8
A. Rancangan ERD .....	8
B. Mapping ERD ke Relational Model.....	9
C. Perancangan Fisik .....	9
BAB III .....	12
IMPLEMENTASI DAN PEMBAHASAN.....	12
3.1 Implementasi dan Pembahasan .....	12
3.2 Penjelasa Fitur/Fungsi Aplikasi .....	16
BAB VI.....	49
KESIMPULAN DAN SARAN.....	49
4.1 Kesimpulan .....	49
4.2 Saran .....	49
DAFTAR PUSTAKA .....	50
LOG SHEET .....	51
BIODATA DAN LINGKUP KERJA ANGGOTA .....	55

## DAFTAR GAMBAR

<i>Gambar 1 Use Case Diagram.....</i>	<i>7</i>
<i>Gambar 2 Entity Relationship Diagram.....</i>	<i>8</i>
<i>Gambar 3 Relational Model .....</i>	<i>9</i>
<i>Gambar 4 SQL Tenaga Medis .....</i>	<i>12</i>
<i>Gambar 5 SQL Output Tenaga Medis .....</i>	<i>12</i>
<i>Gambar 6 SQL Dokter.....</i>	<i>12</i>
<i>Gambar 7 SQL Output Dokter.....</i>	<i>13</i>
<i>Gambar 8 SQL Perawat.....</i>	<i>13</i>
<i>Gambar 9 Output SQL Perawat .....</i>	<i>13</i>
<i>Gambar 10 SQL Obat.....</i>	<i>14</i>
<i>Gambar 11 Output SQL Obat.....</i>	<i>14</i>
<i>Gambar 12 SQL Pasien .....</i>	<i>14</i>
<i>Gambar 13 Output SQL Pasien.....</i>	<i>15</i>
<i>Gambar 14 SQL Rekam Medis .....</i>	<i>15</i>
<i>Gambar 15 Output SQL Rekam Medis .....</i>	<i>15</i>
<i>Gambar 16 SQL Penyakit .....</i>	<i>15</i>
<i>Gambar 17 Output SQL Penyakit.....</i>	<i>16</i>
<i>Gambar 18 Tampilan Utama Antarmuka Sistem Aplikasi Basis Data Rumah Sakit.....</i>	<i>16</i>
<i>Gambar 19 Tampilan Shortcut Icon Sub-Menu Aplikasi Rumah Sakit .....</i>	<i>17</i>
<i>Gambar 20 Tampilan Menu Rumah Sakit .....</i>	<i>17</i>
<i>Gambar 21 Tampilan Antarmuka Sub_Sistem Data Dokter.....</i>	<i>18</i>
<i>Gambar 22 Tampilan Antarmuka Sub-Sistem Data Obat .....</i>	<i>19</i>
<i>Gambar 23 Tampilan Antarmuka Sub-Sistem Data Pasien .....</i>	<i>20</i>
<i>Gambar 24 Tampilan Antarmuka Sub-Sistem Data Penyakit .....</i>	<i>21</i>
<i>Gambar 25 Tampilan Antarmuka Sub-Sistem Data Perawat.....</i>	<i>22</i>
<i>Gambar 26 Tampilan Antarmuka Sub-sistem Data Rekam Medis .....</i>	<i>23</i>
<i>Gambar 27 Tampilan Anatarmuka Sub-Sistem Data Tenaga Medis.....</i>	<i>24</i>
<i>Gambar 28 Tampilan Antarmuka Sub-Sistem Data Multivalue Dosis Obat.....</i>	<i>25</i>
<i>Gambar 29 Tampilan Antarmuka Sub-Sistem Data Multivalue No Telepon .....</i>	<i>26</i>
<i>Gambar 30 Tampilan Antarmuka Sub-Menu Relasi Perawat dan Pasien .....</i>	<i>27</i>
<i>Gambar 31 Tampilan Antarmuka Sub-Menu Tabel Join Pasien, Obat, dan Rekam Medis ....</i>	<i>27</i>
<i>Gambar 32 Tampilan Antarmuka Sub-Menu Data Chart Diagram.....</i>	<i>28</i>
<i>Gambar 33 Kode Program Koneksi .....</i>	<i>30</i>
<i>Gambar 34 Kode Program Menu Utama .....</i>	<i>31</i>
<i>Gambar 35 Kode Program Kolerasi Sub-Menu .....</i>	<i>32</i>
<i>Gambar 36 Kode Progrm Sub-Sistem Dengan Void .....</i>	<i>33</i>
<i>Gambar 37 Kode Program Kolerasi Sub-Menu .....</i>	<i>34</i>
<i>Gambar 38 Kode Program Untuk Menampilkan Data Terurut .....</i>	<i>36</i>
<i>Gambar 39 Kode Program Untuk Mengambil Data Pada Sistem Basis Data.....</i>	<i>36</i>

## DAFTAR TABEL

<i>Table 1 Tenaga Medis.....</i>	<i>9</i>
<i>Table 2 Entitas Dokter.....</i>	<i>10</i>
<i>Table 3 Entitas Perawat.....</i>	<i>10</i>
<i>Table 4 Obat .....</i>	<i>10</i>
<i>Table 5 Multivalue Atribut Obat.....</i>	<i>10</i>
<i>Table 6 Pasien.....</i>	<i>10</i>
<i>Table 7 Perawat - Pasien.....</i>	<i>10</i>
<i>Table 8 Multivalue Atribut No Telepon .....</i>	<i>10</i>
<i>Table 9 Rekam Medis.....</i>	<i>10</i>
<i>Table 10 Pasien - Rekam Medis.....</i>	<i>10</i>
<i>Table 11 Penyakit .....</i>	<i>11</i>

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Sistem Informasi Rumah Sakit (SIRS) adalah suatu tatanan yang berurusan dengan pengumpulan data, pengelolaan data, penyajian informasi, analisis dan penyimpulan informasi serta penyampaian informasi yang dibutuhkan untuk kegiatan rumah sakit. Sistem Informasi Rumah Sakit (SIRS) mencakup semua rumah sakit umum dan khusus baik yang dikelola secara publik maupun privat sebagaimana diatur dalam Undang-Undang Republik Indonesia Nomor 44 Tahun 2009 tentang Rumah Sakit dan Peraturan Menteri Kesehatan No. 82 tahun 2013 tentang Sistem Informasi Manajemen Rumah Sakit, dimana pasal 1 ayat 6 dikatakan bahwa fungsi SIRS adalah untuk meningkatkan efisiensi, efektivitas, profesionalisme, kinerja, serta akses dalam pelayanan. SIMRS atau Sistem Informasi Manajemen Rumah Sakit adalah sistem pengelolaan informasi seluruh kegiatan rumah sakit sehingga membantu setiap proses manajemennya.

Berdasarkan Peraturan Menteri Kesehatan Republik Indonesia Nomor 82 Tahun 2013 dijelaskan bahwa Setiap Rumah Sakit wajib menyelenggarakan SIMRS atau Sistem Informasi Manajemen Rumah Sakit. Aplikasi penyelenggaraan SIMRS yang dibuat oleh Rumah Sakit harus memenuhi persyaratan minimal yang ditetapkan oleh Menteri Kesehatan. Tata kelola sistem informasi yang baik harus selaras dengan fungsi, visi, misi dan strategi organisasi. Secara generik fungsi Rumah Sakit (menurut WHO tahun 1957), memberikan pelayanan kesehatan lengkap kepada masyarakat baik kuratif maupun rehabilitatif, dimana output layanannya menjangkau pelayanan keluarga dan lingkungan, Rumah Sakit juga merupakan pusat pelatihan tenaga kesehatan serta untuk penelitian biososial. Rumah sakit juga merupakan pusat pelayanan rujukan medik spesialis dan sub spesialis dengan fungsi utama menyediakan dan menyelenggarakan upaya kesehatan yang bersifat penyembuhan (kuratif) dan pemulihan (rehabilitasi pasien).

Studi kasus yang diambil pada tugas besar ini adalah sebagai berikut. Rumah Sakit Yusup terdapat tenaga medis yang terdiri dari perawat dan dokter. Kemudian, terdapat pasien yang memiliki id pasien, nama, tanggal lahir, id penyakit, no telepon, alamat, umur, dan ruangan pasien. Seorang pasien ditangani oleh dokter yang memiliki id dokter, nama, dan spesialisasi. Pada rumah sakit, pasien dirawat oleh beberapa perawat yang memiliki data id perawat, nama, dan ruangan pasien. Seorang dokter yang menangani seorang pasien, didampingi oleh beberapa perawat. Pasien wajib melakukan pemeriksaan terlebih dahulu melalui rekam medis yang memiliki data id rekam medis, nama pasien, tanggal lahir pasien, tanggal pemeriksaan, gejala penyakit. Kemudian, dari hasil rekam medis dapat diketahui penyakit yang dimiliki pasien. Rincian data dari rekam medis adalah nama penyakit, id obat, gejala penyakit. Setelah itu, dokter menentukan obat yang akan dikonsumsi oleh pasien. Lalu, perawat memberikan obat kepada pasien dengan rincian yakni dosis obat, tanggal kadaluarsa, id obat, harga obat, jenis obat, dan nama obat.

## **1.2 Rumusan Masalah**

Rumusan masalah yang terdapat pada tugas besar ini adalah:

1. Bagaimana cara merancang dan membuat SIRS?
2. Bagaimana cara membuat Use Case Diagram pada studi kasus SIRS?
3. Bagaimana merancang ERD pada studi kasus SIRS?

## **1.3 Tujuan Pengembangan Aplikasi**

Tujuan pengembangan aplikasi yang terdapat pada tugas besar ini adalah:

1. Mengetahui cara merancang dan membuat SIRS.
2. Mengetahui cara membuat Use Case Diagram pada studi kasus SIRS.
3. Mengetahui cara merancang ERD pada studi kasus SIRS.

## BAB II

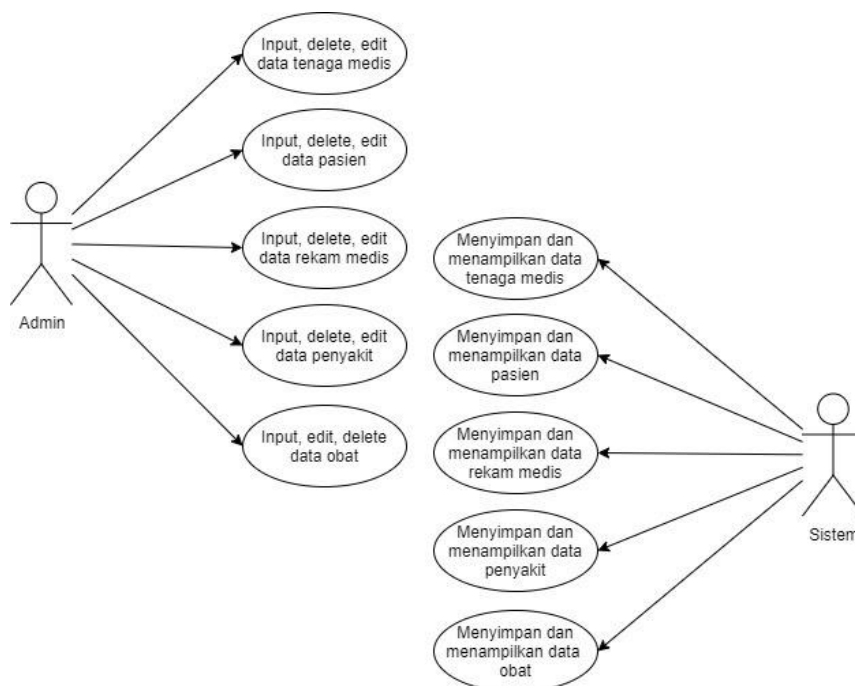
### RANCANGAN SISTEM

#### 2.1 Deskripsi Sistem

Sistem yang dirancang adalah suatu sistem Aplikasi Basis Data Rumah Sakit, yang dapat menampung segala penyimpanan data dan memasukan nya ke dalam sistem Basis Data yang sudah dibuat. Yang mana sistem Basis Data ini adalah Sistem yang memiliki kriteria Sebagai Berikut

1. Dapat menampilkan tampilan antarmuka aplikasi yang sederhana dan mudah dipahami oleh user.
2. Dapat menerima inputan data yang diinputkan oleh user pada tampilan antarmuka aplikasi.
3. Dapat menyimpan data hasil inputan dari user ke dalam sistem Basis Data yang sudah dirancang.
4. Dapat menampilkan data-data yang terdapat pada sistem Basis Data ke tampilan antarmuka aplikasi, supaya user bisa melihat dan mengecek data apa saja yang sudah ada dan sudah diinputkan.
5. Memungkinkan user untuk menambah, mengedit, dan menghapus data yang ada sistem Basis Data melalui tampilan antarmuka aplikasi yang ditampilkan pada user.

#### 2.2 Use Case Diagram

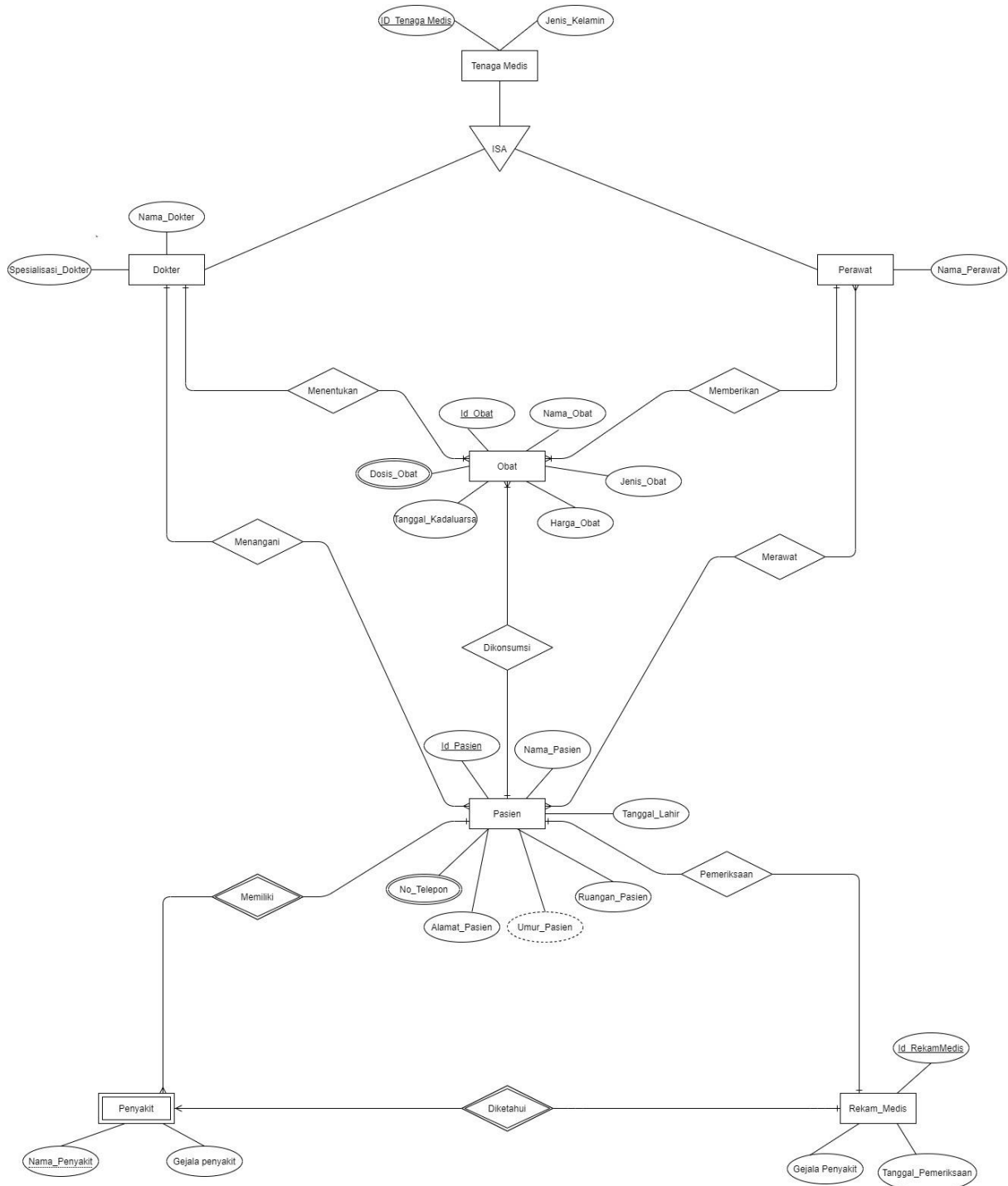


*Gambar 1 Use Case Diagram*

Use case diagram sistem operasi rumah sakit ini dibuat berdasarkan kebutuhan fungsional yang ada pada aplikasi ini, use case diagram dibuat untuk mendeskripsikan sebuah interaksi antara actor (admin) dengan sistem operasi rumah sakit.

## 2.3 Rancangan Basis Data

### A. Rancangan ERD



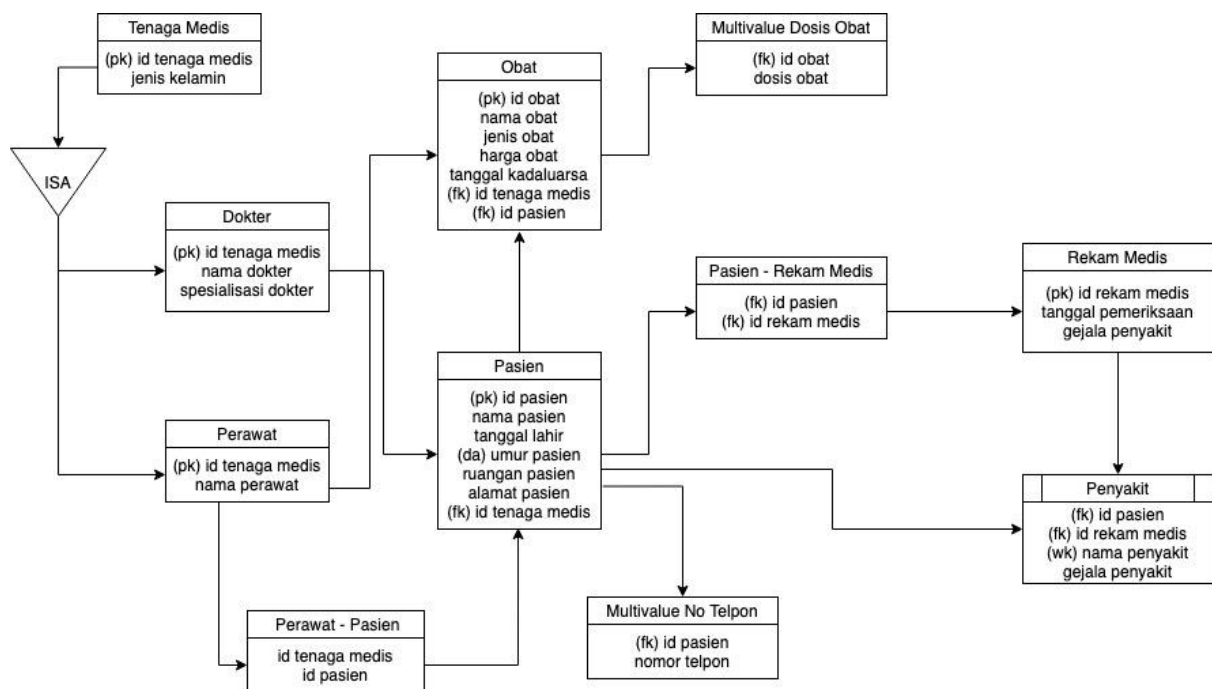
Gambar 2 Entity Relationship Diagram

Rumah sakit terdapat Tenaga medis yang memiliki id tenaga medis dan jenis kelamin, tenaga medis terdiri dari perawat dan dokter. Kemudian, ada pun pasien yang memiliki data id pasien, nama pasien, tanggal lahir, no telepon, alamat, umur, dan ruangan pasien. seorang pasien ditangani oleh dokter yang memiliki data nama dokter



dan spesialisasi dokter. dan pasien dirawat oleh beberapa perawat yang memiliki data nama perawat. Seorang dokter yang menangani seorang pasien, didampingi oleh beberapa perawat. Pasien wajib melakukan pemeriksaan terlebih dahulu melalui rekam medis yang memiliki data id rekam medis, tanggal pemeriksaan, dan gejala penyakit. kemudian, dari rekam medis dapat diketahui penyakit yang dimiliki pasien, dimana rinciannya yakni nama penyakit dan gejala penyakit. Setelah itu, dokter menentukan obat yang akan dikonsumsi oleh pasien. Lalu, perawat memberikan obat kepada pasien dengan rincian yakni dosis obat, tanggal kadaluarsa, id obat, harga obat, jenis obat, dan nama obat.

## B. Mapping ERD ke Relational Model



Gambar 3 Relational Model

Relational model diatas dibuat berdasarkan entity relationship diagram, database sistem rumah sakit ini dibangun dengan 11 tabel, dimana 6 tabel merupakan entity, 1 tabel weak entity, 2 tabel relasi, dan 2 tabel multivalue. Sebuah tabel terdiri beberapa kolom dan dilengkapi dengan beberapa baris yang merupakan data. Relational model dibuat untuk mengetahui alur dari database ini, yaitu dimulai dari tabel tenaga medis, dokter dan perawat, pasien, rekam medis, penyakit, hingga obat. Di setiap tabel entitas terdapat kunci primer atau primary key, dengan adanya primary key maka akan mencegah terjadinya duplikasi data atau data yang berulang-ulang.

## C. Perancangan Fisik

Table 1 Tenaga Medis

<b>Id_Tenaga_Medis</b>	<b>Jenis_Kelamin</b>

Table 2 Entitas Dokter

<b>Id_Tenaga_Medis</b>	Nama_Dokter	Spesialisasi Dokter

Table 3 Entitas Perawat

<b>Id_Tenaga_Medis</b>	Nama_Perawat

Table 4 Obat

<b>Id_Obat</b>	Nama_Obat	Jenis_Obat	Harga_Obat	Tanggal_Kadaluarsa	Id_Tenaga_Medis	Id_Pasien

Table 5 Multivalue Atribut Obat

<b>Id_Obat</b>	Dosis_Obat

Table 6 Pasien

<b>Id_Pasien</b>	Nama_Pasien	Tanggal_Lahir	Umur_Pasien	Ruangan_Pasien	Alamat_Pasien	Id_Tenaga_Medis

Table 7 Perawat - Pasien

<b>Id_Tenaga_Medis</b>	<b>Id_Pasien</b>

Table 8 Multivalue Atribut No Telepon

<b>Id_Pasien</b>	No_Telepon

Table 9 Rekam Medis

<b>Id_Rekam_Medis</b>	Tanggal_Pemeriksaan	Gejala_Penyakit

Table 10 Pasien - Rekam Medis

<b>Id_Pasien</b>	<b>Id_Rekam_Medis</b>

Table 11 Penyakit

<b>Id_Pasien</b>	<b>Id_Rekam_Medis</b>	Nama_Penyakit	Gejala_Penyakit

## BAB III

### IMPLEMENTASI DAN PEMBAHASAN

#### 3.1 Implementasi dan Pembahasan

##### Tabel Tenaga Medis

Pada tabel Tenaga Medis berisi Id\_TenagaMedis dan Jenis Kelamin dimana Id\_TenagaMedis merupakan Primary Key.

```
MariaDB [rumah_sakit]> DESC Tenaga_Medis;
+-----+-----+-----+-----+-----+-----+
| Field          | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Id_TenagaMedis | char(5) | NO   | PRI | NULL    |       |
| Jenis_Kelamin  | char(1) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.058 sec)
```

*Gambar 4 SQL Tenaga Medis*

Untuk mengisi tabel tersebut menggunakan syntax **INSERT INTO** Tenaga\_Medis **VALUES**

-> (ID001, 'L'),

-> (ID002, 'P'),

-> (ID003, 'P');

Sehingga menghasilkan output seperti dibawah.

```
MariaDB [rumah_sakit]> select * from tenaga_medis;
+-----+-----+
| Id_TenagaMedis | Jenis_Kelamin |
+-----+-----+
| ID001          | L              |
| ID002          | P              |
| ID003          | P              |
+-----+-----+
3 rows in set (0.000 sec)
```

*Gambar 5 SQL Output Tenaga Medis*

##### Entitas Dokter

Pada tabel Dokter berisi Id\_TenagaMedis, Nama\_Dokter, dan Spesialisasi\_Dokter dimana Id\_TenagaMedis merupakan Primary Key.

```
MariaDB [rumah_sakit]> DESC DOKTER;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Id_TenagaMedis | char(5)       | NO   | PRI | NULL    |       |
| Nama_Dokter    | varchar(30)   | YES  |     | NULL    |       |
| Spesialisasi_Dokter | varchar(20) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.089 sec)
```

*Gambar 6 SQL Dokter*

Untuk mengisi tabel tersebut menggunakan syntax **INSERT INTO** Dokter **VALUES**

-> (ID001, 'Dr. Yusuf', 'Jantung'),

-> (ID002, 'Dr. Eci', 'Paru-Paru'),

-> (ID002, 'Dr. Maqruf', 'Otak');

Sehingga menghasilkan output seperti dibawah.

```
MariaDB [rumah_sakit]> select * from dokter;
+-----+-----+-----+
| Id_TenagaMedis | Nama_Dokter | Spesialisasi_Dokter |
+-----+-----+-----+
| ID001          | Dr. Yusuf   | Jantung              |
| ID002          | Dr. Eci     | Paru-Paru            |
| ID003          | Dr. Maqruf  | Otak                  |
+-----+-----+-----+
3 rows in set (0.001 sec)
```

*Gambar 7 SQL Output Dokter*

## Entitas Perawat

Pada tabel Perawat berisi Id\_TenagaMedis dan Nama\_Perawat, yang mana Primary Key terdapat pada Id\_TenagaMedis.

```
MariaDB [rumah_sakit]> DESC Perawat;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Id_TenagaMedis | char(5)       | NO   | PRI | NULL    |       |
| Nama_Perawat   | varchar(30)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.067 sec)
```

*Gambar 8 SQL Perawat*

Untuk mengisi tabel tersebut menggunakan syntax **INSERT INTO Perawat VALUES**

-> (ID001, 'Yusuf');

Sehingga menghasilkan output seperti dibawah.

```
MariaDB [rumah_sakit]> select * from perawat;
+-----+-----+
| Id_TenagaMedis | Nama_Perawat |
+-----+-----+
| ID001          | Yusuf       |
+-----+-----+
1 row in set (0.052 sec)
```

*Gambar 9 Output SQL Perawat*

## Tabel Obat

Pada tabel Pasien berisi Id\_Obat, Nama\_Obat, Jenis\_Obat, Harga\_Obat, Tanggal\_Kadaluarsa, Id\_TenagaMedis dan Id\_Pasien. Primary Key pada Obat terdapat di Id\_Obat sedangkan Foreign Key terdapat pada Id\_TenagaMedis dan Id\_Pasien.

```
MariaDB [Rumah_Sakit]> DESC Obat;
```

Field	Type	Null	Key	Default	Extra
Id_Obat	char(5)	NO	PRI	NULL	
Nama_Obat	varchar(30)	YES		NULL	
Jenis_Obat	char(10)	YES		NULL	
Harga_Obat	char(9)	YES		NULL	
Tanggal_Kadaluarsa	date	YES		NULL	
Id_TenagaMedis	char(5)	YES	MUL	NULL	
Id_Pasien	char(5)	YES	MUL	NULL	

7 rows in set (0.147 sec)

*Gambar 10 SQL Obat*

Untuk mengisi tabel tersebut menggunakan syntax **INSERT INTO Obat VALUES**  
 -> (ID001, 'Antangin', 'Kaplet', 'Rp. 200,000', '12 MEI 2023', 'ID002', 'ID002'),  
 -> (ID003, 'Maag', 'Serbuk', 'Rp. 250,000', '12 MEI 2022', 'ID003', 'ID003');  
 Sehingga menghasilkan output seperti dibawah.

```
MariaDB [rumah_sakit]> select * from Obat;
```

Id_Obat	Nama_Obat	Jenis_Obat	Harga_Obat	Tanggal_Kadaluarsa	Id_TenagaMedis	Id_Pasien
ID002	Antangin	Kaplet	Rp. 200,000	12 MEI 2023	ID002	ID002
ID003	Maag	Serbuk	Rp. 250,000	12 Mei 2022	ID003	ID003

2 rows in set (0.000 sec)

*Gambar 11 Output SQL Obat*

## Tabel Pasien

Pada tabel Pasien berisi Id\_Pasien, Nama\_Pasien, Tanggal\_Lahir, Ruangan\_Pasien, Alamat\_Pasien, Id\_TenagaMedis, dan Umur. Pada tabel ini Primary Key terdapat pada Id\_Pasien sedangkan Foreign Key terdapat pada Id\_TenagaMedis.

```
MariaDB [rumah_sakit]> DESC Pasien;
```

Field	Type	Null	Key	Default	Extra
Id_Pasien	char(5)	NO	PRI	NULL	
Nama_Pasien	varchar(30)	YES		NULL	
Tanggal_Lahir	date	YES		NULL	
Ruangan_Pasien	char(5)	YES		NULL	
Alamat_Pasien	varchar(30)	YES		NULL	
Id_TenagaMedis	char(5)	YES	MUL	NULL	
Umur	char(8)	YES		NULL	

7 rows in set (0.067 sec)

*Gambar 12 SQL Pasien*

Untuk mengisi tabel tersebut menggunakan syntax **INSERT INTO Pasien VALUES**  
 -> (ID001, 'Yusuf', '12 Juli 2001', 'Kamar 01', 'Lampung', 'ID001', '20 tahun'),  
 -> (ID002, 'Habib', '12 Mei 2001', 'Kamar 02', 'BaLam', 'ID001', '20 tahun'),  
 -> (ID003, 'Enak', '12 Mei 2001', 'Kamar 05', 'BaLam', 'ID001', '20 tahun'),  
 -> (ID004, 'Gilang', '12 Juli 2008', 'Kamar 01', 'Padang', 'ID001', '12 tahun');  
 Sehingga menghasilkan output seperti dibawah.

```
MariaDB [rumah_sakit]> select * from pasien;
```

Id_Pasien	Nama_Pasien	Tanggal_Lahir	Ruangan_Pasien	Alamat_Pasien	Id_TenagaMedis	Umur
ID001	YUSUF	12 JULI 2001	Kamar 01	LAMPUNG	ID001	20 TAHUN
ID002	Habib	12 Mei 2001	Kamar 02	BaLam	ID001	20 Tahun
ID003	Enak	12 Mei 2001	Kamar 05	BaLAm	ID001	20 Tahun
ID004	Gilang	12 Juli 2008	Kamar 01	Padang	ID001	12 Tahun

```
4 rows in set (0.000 sec)
```

*Gambar 13 Output SQL Pasien*

### Tabel Rekam Medis

Pada tabel Rekam Medis berisi Id\_RekamMedis, Tanggal\_Pemeriksaan, dan Gejala\_Penyakit. Primary Key pada tabel ini terdapat Id\_RekamMedis.

```
MariaDB [rumah_sakit]> DESC Rekam_Medis;
```

Field	Type	Null	Key	Default	Extra
Id_RekamMedis	char(5)	NO	PRI	NULL	
Tanggal_Pemeriksaan	date	YES		NULL	
Gejala_Penyakit	varchar(20)	YES		NULL	

```
3 rows in set (0.087 sec)
```

*Gambar 14 SQL Rekam Medis*

Untuk mengisi tabel tersebut menggunakan syntax **INSERT INTO** Rekam Medis **VALUES**

-> (ID002, '3 Mei 2021', 'Batuk'),

-> (ID003, '12 Desember 2022', 'Otak');

Sehingga menghasilkan output seperti dibawah.

```
MariaDB [rumah_sakit]> select * from rekam_medis;
```

Id_RekamMedis	Tanggal_Pemeriksaan	Gejala_Penyakit
ID002	3 Mei 2021	Batuk
ID003	12 Desember 2022	Otak

```
2 rows in set (0.017 sec)
```

*Gambar 15 Output SQL Rekam Medis*

### Tabel Penyakit

Pada tabel Rekam Medis berisi Id\_Pasien, Nama\_Penyakit, Gejala\_Penyakit, dan Id\_RekamMedis. Pada tabel ini Foreign Key terdapat pada Id\_Pasien dan Id\_RekamMedis.

```
MariaDB [Rumah_Sakit]> DESC Penyakit;
```

Field	Type	Null	Key	Default	Extra
Id_Pasien	char(5)	YES	MUL	NULL	
Nama_Penyakit	varchar(30)	YES		NULL	
Gejala_Penyakit	varchar(20)	YES		NULL	
Id_RekamMedis	char(5)	YES	MUL	NULL	

```
4 rows in set (0.076 sec)
```

*Gambar 16 SQL Penyakit*



Untuk mengisi tabel tersebut menggunakan syntax **INSERT INTO Penyakit VALUES**

-> (ID002, 'Batuk Berdahak', 'Batuk', 'ID002'),

-> (ID003, 'Gigi Berlubang', 'Gigi', 'ID003');

Sehingga menghasilkan output seperti dibawah.

```
MariaDB [rumah_sakit]> select * from penyakit;
```

Id_Pasien	Nama_Penyakit	Gejala_Penyakit	Id_RekamMedis
ID002	Batuk Berdahak	Batuk	ID002
ID003	Gigi Berlubang	Gigi	ID003

2 rows in set (0.002 sec)

*Gambar 17 Output SQL Penyakit*

## 3.2 Penjelasa Fitur/Fungsi Aplikasi

### 3.2.1 Penjelasan Keseluruhan Fitur Dan Fungsi Aplikasi Dari Tampilan Antarmuka

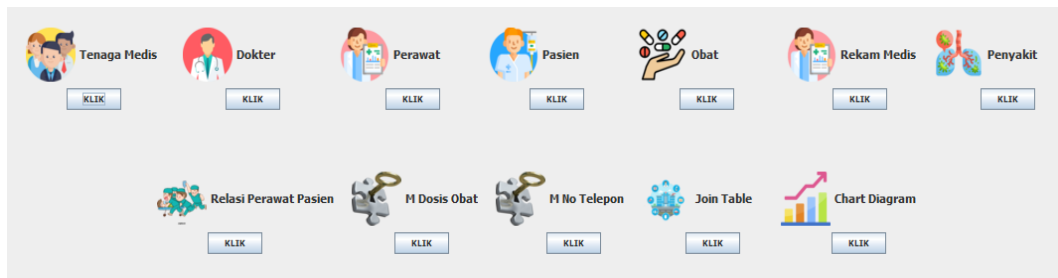
Tampilan utama antarmuka Sistem Aplikasi Basis Data Rumah Sakit.



*Gambar 18 Tampilan Utama Antarmuka Sistem Aplikasi Basis Data Rumah Sakit*

Pada tampilan antarmuka sistem aplikasi ini, sistem sudah bisa menampilkan segala tampilan yang sederhana dan mudah dipahami oleh user. Dimana untuk melakukan proses input data ataupun melihat hasil sub-sistem yang ada, user bisa memilih meng-klik menu pilihan yang ada di sebelah pojok kiri bagian atas. Atau dengan meng-klik shortcut icon/tampilan yang tersebar di bagian tengah antarmuka sistem.





*Gambar 19 Tampilan Shortcut Icon Sub-Menu Aplikasi Rumah Sakit*

Tampilan shortcut icon sub-menu sistem pada tampilan antarmuka sistem aplikasi yang ditampilkan kepada user. User bisa meng-klik tombol “klik” sebagai akses cepat untuk mengakses menu sub-sistem tanpa harus melalui menu bar yang ada di bagian pojok kiri sebelah atas layar tampilan.



*Gambar 20 Tampilan Menu Rumah Sakit*

Hasil dari meng-klik menu bar yang terletak di bagian pojok kiri atas tampilan antarmuka sistem. Maka akan memunculkan menu tambahan sub-sistem yang ada pada sistem utama Basis Data Rumah Sakit yang sudah dibuat. Dimana ketika user mengklik salah satu sub-menu yang ada, maka user akan bisa mengakses sub-sistem pada aplikasi ini.

### Tampilan Sub-Menu Data Dokter

ID Tenaga Medis	Nama Dokter	Spesialisasi Dokter
ID001	Dr. Yusuf	Jantung
ID002	Dr. Eci	Paru-Paru
ID003	Dr. Maqruf	Otak

*Gambar 21 Tampilan Antarmuka Sub\_Sistem Data Dokter*

Gambar diatas merupakan tampilan antarmuka Sub-Sistem Data Dokter. Yang mana sub-sistem ini adalah sistem yang menangani proses terkait aplikasi sistem basis data yang terkhusus untuk data data mengenai dokter pada sistem rumah sakit.

Fitur yang bisa digunakan:

1. Aplikasi Sub-Sistem bisa menyimpan dan menampilkan data hasil inputan User.
2. User bisa menambahkan data berupa Nama Dokter, dan spesialis Dokter. Kemudian menggunakan fitur “SIMPAN” untuk memberikan perintah menyimpan data kepada aplikasi, lalu aplikasi akan menampilkan kembali data yang sudah disimpan.
3. User bisa mengedit data yang sudah diinput, seperti mengganti data pada nama dokter atau mengganti data pada spesialisasi dokter dengan mengklik data pada tabel dan menggantinya, kemudian menggunakan fitur “EDIT”.
4. User bisa menghapus data yang sudah diinputkan, dengan mengklik data yang akan dihapus pada tabel, kemudian menggunakan fitur “HAPUS”. Namun bila ada data yang memiliki foreign key (ID Tenaga Medis) tidak bisa dihapus jika data di Tenaga Medis tidak dihapus terlebih dahulu. Lalu jika ingin membatalkan/tidak jadi mengedit data yang sudah ada, user bisa menggunakan fitur “BATALL”.
5. User bisa mencari data tertentu dengan cara mengetikkan kata atau huruf, lalu sistem akan mencarikan data yang sesuai, dengan mencocokkan nya

dengan nama dokter. Akses ini dapat dilakukan dengan menggunakan fitur “CARI”.

### **Tampilan Sub-Menu Data Obat**

ID Obat	Nama Obat	Jenis Obat	Harga Obat	Tanggal Kadaluarsa...	ID Tenaga Medis	ID Pasien
ID002	Antangin	Kaplet	Rp. 200,000	12 Mei 2023	ID002	ID002
ID003	Maag	Serbuk	Rp. 250,000	12 Mei 2022	ID003	ID003

*Gambar 22 Tampilan Antarmuka Sub-Sistem Data Obat*

Gambar diatas merupakan tampilan antarmuka Sub-Sistem Data Obat. Yang mana sub-sistem ini adalah sistem yang menangani proses terkait aplikasi sistem basis data yang terkhusus untuk data data mengenai Obat pada sistem rumah sakit.

Fitur yang bisa digunakan:

1. Aplikasi Sub-Sistem bisa menyimpan dan menampilkan data hasil inputan User.
2. User bisa menambahkan data berupa Nama Obat, Jenis Obat, Harga Obat, dan Tanggal Kadaluarsa Obat. User juga diminta memilih ID Pasien dan ID Tenaga Medis sebagai penghubung antara data Obat dengan data Pasien dan data Tenaga Medis yang dihubungkan dengan (Foreign Key).
3. Kemudian user menggunakan fitur “SIMPAN” untuk memberikan perintah menyimpan data pada aplikasi, lalu aplikasi akan menampilkan kembali data yang sudah disimpan.
4. User bisa mengedit data yang sudah diinput, seperti mengganti data pada nama, jenis, harga dan tanggal kadaluarsa obat dan memilih kembali ID Pasien dan ID Tenaga Medis sebagai Foreign Key. Akses tersebut dapat dilakukan dengan mengklik data pada tabel dan menggantinya, kemudian menggunakan fitur “EDIT”.
5. User bisa menghapus data yang sudah diinputkan, dengan mengklik data yang akan dihapus pada tabel, kemudian menggunakan fitur “HAPUS”. Tetapi jika ada data yang memiliki hubungan foreign key (ID Pasien dan ID Tenaga Medis) tidak akan bisa dihapus jika data yang berhubungan pada tabel Pasien dan tabel Tenaga Medis tidak dihapus terlebih dahulu. Lalu jika ingin membatalkan/tidak jadi mengedit data yang sudah ada, user bisa menggunakan fitur “BATAL”.
6. User bisa mencari data tertentu dengan cara mengetikkan kata atau huruf, lalu sistem akan mencari data yang sesuai, dengan mencocokkannya

dengan nama obat. Akses ini dapat dilakukan dengan menggunakan fitur “CARI”.

### Tampilan Sub-Menu Data Pasien

ID Pasien	Nama Pasien	Tanggal Lahir	Ruangan Pasien	Alamat Pasien	ID Tenaga Medis	Umur Pasien
ID001	YUSUF	12 JULI 2001	Kamar 01	LAMPUNG	ID001	20 TAHUN
ID002	Habib	12 Mei 2001	Kamar 02	BaLam	ID001	20 Tahun
ID003	Enak	12 Mei 2001	Kamar 05	BaLAm	ID001	20 Tahun
ID004	Gilang	12 Juli 2008	Kamar 01	Padang	ID001	12 Tahun

*Gambar 23 Tampilan Antarmuka Sub-Sistem Data Pasien*

ID Pasien	Nama Pasien	Alamat Pasien
ID004	Gilang	Padang

ID Pasien	Nama Pasien	Alamat Pasien
ID001	YUSUF	LAMPUNG
ID002	Habib	BaLam
ID003	Enak	BaLAm

Gambar diatas merupakan tampilan antarmuka Sub-Sistem Data Pasien. Yang mana sub-sistem ini adalah sistem yang menangani proses terkait aplikasi sistem basis data yang terkhusus untuk data data mengenai Pasien pada sistem rumah sakit.

Fitur yang bisa digunakan:

1. Aplikasi Sub-Sistem bisa menyimpan dan menampilkan data hasil inputan User.
2. User bisa menambahkan data berupa Nama Pasien, Tanggal Lahir Pasien, Umur Pasien, dan Alamat Pasien. User juga diminta memilih ID Tenaga Medis dan Ruangan Pasien sebagai penghubung antara data Pasien dan data Tenaga Medis yang dihubungkan dengan (Foreign Key).
3. Kemudian user menggunakan fitur “SIMPAN” untuk memberikan perintah menyimpan data pada aplikasi, lalu aplikasi akan menampilkan kembali data yang sudah disimpan.
4. User bisa mengedit data yang sudah diinput, seperti mengganti data pada nama, Tanggal Lahir, Umur, Alamat, dan memilih kembali ID Tenaga Medis dan Ruangan Pasien sebagai Foreign Key. Akses tersebut dapat dilakukan

dengan mengklik data pada tabel dan menggantinya, kemudian menggunakan fitur “EDIT”.

5. User bisa menghapus data yang sudah diinputkan, dengan mengklik data yang akan dihapus pada tabel, kemudian menggunakan fitur “HAPUS”. Tetapi jika ada data yang memiliki hubungan foreign key (ID Tenaga Medis) tidak akan bisa dihapus jika data yang berhubungan pada tabel Tenaga Medis tidak dihapus terlebih dahulu. Lalu jika ingin membatalkan/tidak jadi mengedit data yang sudah ada, user bisa menggunakan fitur “BATAL”.
6. User bisa mencari data tertentu dengan cara mengetikkan kata atau huruf, lalu sistem akan mencari data yang sesuai, dengan mencocokkannya dengan nama pasien. Akses ini dapat dilakukan dengan menggunakan fitur “CARI”.

### **Tampilan Sub-Menu Data Penyakit**

ID Pasien	Nama Penyakit	Gejala Penyakit	ID Rekam Medis
ID002	Batuk Berdahak	Batuk	ID002
ID003	Gigi Berlubah	Gigi	ID003

*Gambar 24 Tampilan Antarmuka Sub-Sistem Data Penyakit*

Gambar diatas merupakan tampilan antarmuka Sub-Sistem Data Penyakit. Yang mana sub-sistem ini adalah sistem yang menangani proses terkait aplikasi sistem basis data yang terkhusus untuk data data mengenai Penyakit pada sistem rumah sakit.

Fitur yang bisa digunakan:

1. Aplikasi Sub-Sistem bisa menyimpan dan menampilkan data hasil inputan User.
2. User bisa menambahkan data berupa Nama Penyakit dan Gejala Penyakit. User juga diminta memilih ID Rekam Medis sebagai penghubung antara data Pasien dan data Rekam Medis yang dihubungkan dengan (Foreign Key).
3. Kemudian user menggunakan fitur “SIMPAN” untuk memberikan perintah menyimpan data kepada aplikasi, lalu aplikasi akan menampilkan kembali data yang sudah disimpan.

4. User bisa mengedit data yang sudah diinput, seperti mengganti data pada nama penyakit dan gejala penyakit, kemudian memilih kembali ID Rekam Medis sebagai Foreign Key. Akses tersebut dapat dilakukan dengan mengklik data pada tabel dan menggantinya, kemudian menggunakan fitur “EDIT”. Lalu jika ingin membatalkan/tidak jadi mengedit data yang sudah ada, user bisa menggunakan fitur “BATAL”.
5. User bisa menghapus data yang sudah diinputkan, dengan mengklik data yang akan dihapus pada tabel, kemudian menggunakan fitur “HAPUS”. Tetapi jika ada data yang memiliki hubungan foreign key (ID Pasien dan ID Rekam Medis) tidak akan bisa dihapus jika data yang berhubungan pada tabel Rekam Medis tidak dihapus terlebih.

### **Tampilan Sub-Menu Data Perawat**



ID Tenaga Medis	Nama Perawat
ID001	Yusuf

*Gambar 25 Tampilan Antarmuka Sub-Sistem Data Perawat*

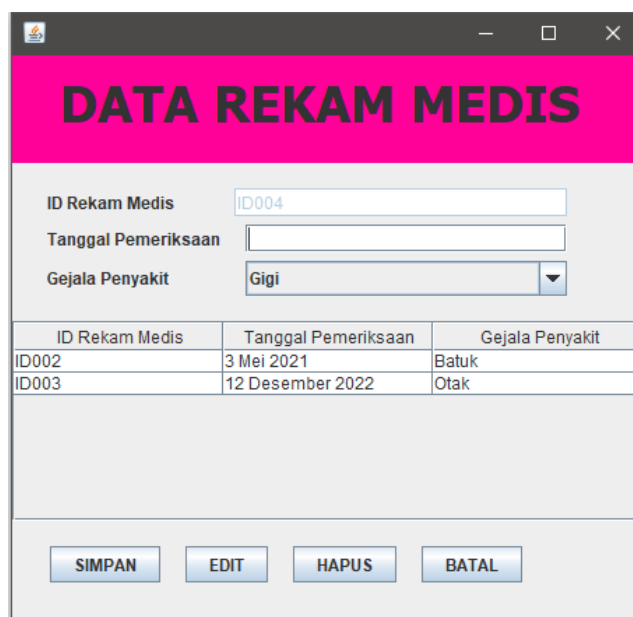
Gambar diatas merupakan tampilan antarmuka Sub-Sistem Data Perawat. Yang mana sub-sistem ini adalah sistem yang menangani proses terkait aplikasi sistem basis data yang terkhusus untuk data data mengenai Perawat pada sistem rumah sakit.

Fitur yang bisa digunakan:

1. Aplikasi Sub-Sistem bisa menyimpan dan menampilkan data hasil inputan User.
2. User bisa menambahkan data berupa Nama Perawat. Sistem akan menentukan ID Tenaga Medis secara otomatis sesuai urutan penginputan data. Yang mana ID Tenaga Medis sebagai penghubung antara data Tenaga Medis dengan data Perawat yang dihubungkan dengan (Foreign Key). Kemudian user menggunakan fitur “SIMPAN” untuk memberikan perintah menyimpan data pada aplikasi, lalu aplikasi akan menampilkan kembali data yang sudah disimpan
3. User bisa mengedit data yang sudah diinput, seperti mengganti data pada nama perawat. Akses tersebut dapat dilakukan dengan mengklik data pada tabel dan menggantinya, kemudian menggunakan fitur “EDIT”.

4. User bisa menghapus data yang sudah diinputkan, dengan mengklik data yang akan dihapus pada tabel, kemudian menggunakan fitur “HAPUS”. Tetapi jika ada data yang memiliki hubungan foreign key (ID Tenaga Medis) tidak akan bisa dihapus jika data yang berhubungan pada tabel Tenaga Medis tidak dihapus terlebih dahulu. Lalu jika ingin membatalkan/tidak jadi mengedit data yang sudah ada, user bisa menggunakan fitur “BATAL”.
5. User bisa mencari data tertentu dengan cara mengetikkan kata atau huruf, lalu sistem akan mencari data yang sesuai, dengan mencocokkannya dengan nama perawat. Akses ini dapat dilakukan dengan menggunakan fitur “CARI”.

### **Tampilan Sub-Menu Data Rekam Medis**



ID Rekam Medis	Tanggal Pemeriksaan	Gejala Penyakit
ID002	3 Mei 2021	Batuk
ID003	12 Desember 2022	Otak

*Gambar 26 Tampilan Antarmuka Sub-sistem Data Rekam Medis*

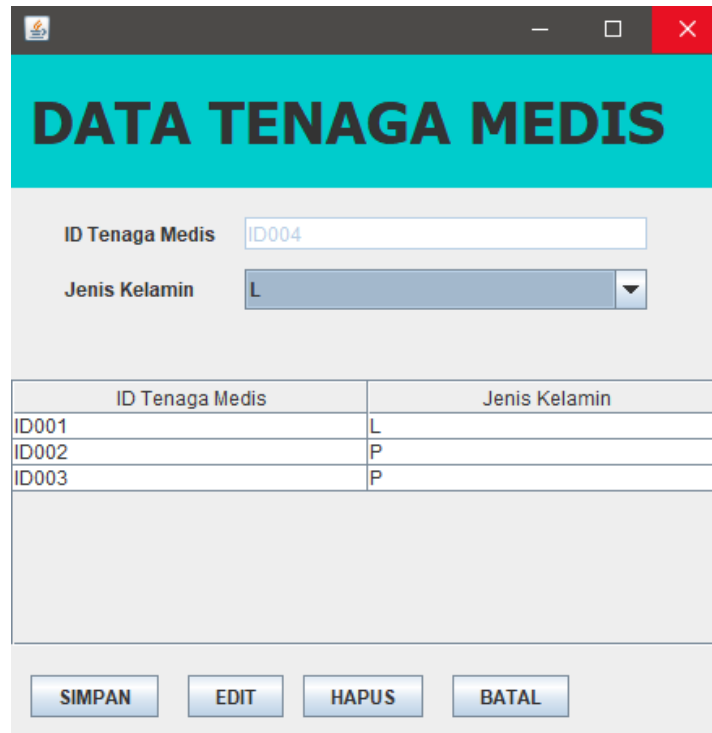
Gambar diatas merupakan tampilan antarmuka Sub-Sistem Data Rekam Medis. Yang mana sub-sistem ini adalah sistem yang menangani proses terkait aplikasi sistem basis data yang terkhusus untuk data data mengenai Rekam Medis pada sistem rumah sakit.

Fitur yang bisa digunakan:

1. Aplikasi Sub-Sistem bisa menyimpan dan menampilkan data hasil inputan User.
2. User bisa menambahkan data berupa Tanggal Pemeriksaan dan Gejala Penyakit. Sistem akan menentukan ID Rekam Medis secara otomatis sesuai urutan penginputan data. Kemudian user menggunakan fitur “SIMPAN” untuk memberikan perintah menyimpan data kepada aplikasi, lalu aplikasi akan menampilkan kembali data yang sudah disimpan.
3. User bisa mengedit data yang sudah diinput, seperti mengganti data pada tanggal pemeriksaan dan gejala penyakit. Akses tersebut dapat dilakukan dengan mengklik data pada tabel dan menggantinya, kemudian menggunakan fitur “EDIT”. Lalu jika ingin membatalkan/tidak jadi mengedit data yang sudah ada, user bisa menggunakan fitur “BATAL”.

4. User bisa menghapus data yang sudah diinputkan, dengan mengklik data yang akan dihapus pada tabel, kemudian menggunakan fitur “HAPUS”.

### **Tampilan Sub-Menu Data Tenaga Medis**



ID Tenaga Medis	Jenis Kelamin
ID001	L
ID002	P
ID003	P

*Gambar 27 Tampilan Anatarmuka Sub-Sistem Data Tenaga Medis*

Gambar diatas merupakan tampilan antarmuka Sub-Sistem Data Tenaga Medis. Yang mana sub-sistem ini adalah sistem yang menangani proses terkait aplikasi sistem basis data yang terkhusus untuk data data mengenai Tenaga Medis pada sistem rumah sakit.

Fitur yang bisa digunakan:

1. Aplikasi Sub-Sistem bisa menyimpan dan menampilkan data hasil inputan User.
2. User bisa menambahkan data berupa Tanggal Pemeriksaan dan Gejala Penyakit. Sistem akan menentukan ID Tenaga Medis secara otomatis sesuai urutan penginputan data. Kemudian user menggunakan fitur “SIMPAN” untuk memberikan perintah menyimpan data kepada aplikasi, lalu aplikasi akan menampilkan kembali data yang sudah disimpan.
3. User bisa mengedit data yang sudah diinput, seperti mengganti data Jenis Kelamin. Akses tersebut dapat dilakukan dengan mengklik data pada tabel dan menggantinya, kemudian menggunakan fitur “EDIT”. Lalu jika ingin membatalkan/tidak jadi mengedit data yang sudah ada, user bisa menggunakan fitur “BATALL”.
4. User bisa menghapus data yang sudah diinputkan, dengan mengklik data yang akan dihapus pada tabel, kemudian menggunakan fitur “HAPUS”.



### Tampilan Sub-Menu Data Multivalue Dosis Obat

**MULTIVALUE DOSIS OBAT**

ID Obat: ID001

Dosis Obat: Dosis Rendah

ID Obat	Dosis Obat
---------	------------

SIMPAN EDIT HAPUS BATALL

*Gambar 28 Tampilan Antarmuka Sub-Sistem Data Multivalue Dosis Obat*

Gambar diatas merupakan tampilan antarmuka Sub-Sistem Data Multivalue Dosis Obat. Yang mana sub-sistem ini adalah sistem yang menangani proses terkait aplikasi sistem basis data yang terkhusus untuk data data mengenai Multivalue Dosis Obat pada sistem rumah sakit.

Fitur yang bisa digunakan:

1. Aplikasi Sub-Sistem bisa menyimpan dan menampilkan data hasil inputan User.
2. User bisa menambahkan data berupa Dosis Obat ID Obat. Yang mana ID Obat merupakan foreign key yang menghubungkan data multivalue Dosis Obat dengan data Obat. Kemudian user bisa menggunakan fitur “SIMPAN” untuk memberikan perintah menyimpan data pada aplikasi, lalu aplikasi akan menampilkan kembali data yang sudah disimpan.
3. User bisa mengedit data yang sudah diinput, seperti mengganti data ID Obat dan Dosis Obat. Akses tersebut dapat dilakukan dengan mengklik data pada tabel dan menggantinya, kemudian menggunakan fitur “EDIT”. Lalu jika ingin membatalkan/tidak jadi mengedit data yang sudah ada, user bisa menggunakan fitur “BATALL”.
4. User bisa menghapus data yang sudah diinputkan, dengan mengklik data yang akan dihapus pada tabel, kemudian menggunakan fitur “HAPUS”.

### Tampilan Sub-Menu Data Multivalue No Telepon

ID Pasien	Nomor Telepon
ID002	087612345678
ID003	085367231278
ID004	085367231290

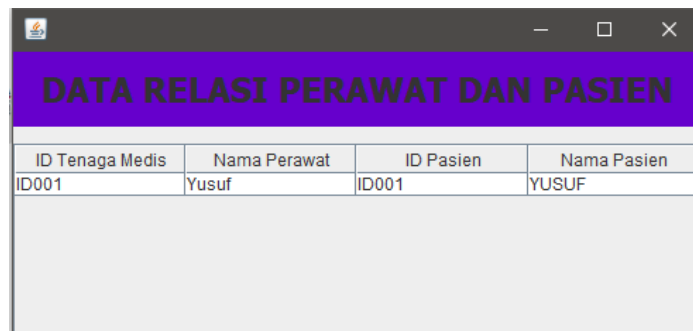
*Gambar 29 Tampilan Antarmuka Sub-Sistem Data Multivalue No Telepon*

Gambar diatas merupakan tampilan antarmuka Sub-Sistem Data Multivalue No Telepon. Yang mana sub-sistem ini adalah sistem yang menangani proses terkait aplikasi sistem basis data yang terkhusus untuk data data mengenai Multivalue No Telepon pada sistem rumah sakit.

Fitur yang bisa digunakan:

1. Aplikasi Sub-Sistem bisa menyimpan dan menampilkan data hasil inputan User.
2. User bisa menambahkan data berupa ID Pasien dan No Telepon. Yang mana ID Pasien merupakan foreign key yang menghubungkan data multivalue No Telepon dengan data Pasien. Kemudian user bisa menggunakan fitur “SIMPAN” untuk memberikan perintah menyimpan data pada aplikasi, lalu aplikasi akan menampilkan kembali data yang sudah disimpan.
3. User bisa mengedit data yang sudah diinput, seperti mengganti data ID Pasien dan No Telepon. Akses tersebut dapat dilakukan dengan mengklik data pada tabel dan menggantinya, kemudian menggunakan fitur “EDIT”. Lalu jika ingin membatalkan/tidak jadi mengedit data yang sudah ada, user bisa menggunakan fitur “BATAL”.
4. User bisa menghapus data yang sudah diinputkan, dengan mengklik data yang akan dihapus pada tabel, kemudian menggunakan fitur “HAPUS”.

### **Tampilan Sub-Menu Data Relasi Perawat dan Pasien**



ID Tenaga Medis	Nama Perawat	ID Pasien	Nama Pasien
ID001	Yusuf	ID001	YUSUF

*Gambar 30 Tampilan Antarmuka Sub-Menu Relasi Perawat dan Pasien*

Gambar diatas merupakan tampilan antarmuka Sub-Sistem Data hasil relasi dari 2 tabel. Yang mana sub-sistem ini adalah sistem menampilkan hasil penggabungan antara tabel Perawat dan tabel Pasien.

Fitur yang bisa digunakan:

1. Aplikasi Sub-Sistem hanya bisa menampilkan data hasil inputan User yang berupa penggabungan 2 tabel (Perawat dan Pasien). Yang mana hasil ini didapatkan dari menggabungkan setiap foreign key yang berhubungan dari tabel Perawat dan Pasien. Yang digabungkan menggunakan Natural Join.

### **Tampilan Sub-Menu Tabel Join Pasien, Obat, dan Rekam Medis**



ID Pasien	Nama Pasien	Nama Obat	Tanggal Pemeriks...	Harga Obat
ID002	Habib	Antangin	3 Mei 2021	Rp. 200,000
ID003	Enak	Maag	12 Desember 2022	Rp. 250,000

*Gambar 31 Tampilan Antarmuka Sub-Menu Tabel Join Pasien, Obat, dan Rekam Medis*

Gambar diatas merupakan tampilan antarmuka Sub-Sistem Data hasil relasi dari 3 tabel. Yang mana sub-sistem ini adalah sistem menampilkan hasil penggabungan antara tabel Pasien, Obat dan tabel Rekam Medis.

Fitur yang bisa digunakan:

1. Aplikasi Sub-Sistem hanya bisa menampilkan data hasil inputan User yang berupa penggabungan 3 tabel (Pasien, Obat, Rekam Medis). Yang mana hasil ini didapatkan dari menggabungkan setiap foreign key yang berhubungan dari tabel Pasien, Obat dan Rekam Medis. Yang digabungkan menggunakan Natural Join

### Tampilan Sub-Menu Chart Diagram

Pasien	Obat	Tenaga Medis
4	3	3

PASIENT

OBAT

TENAGA MEDIS

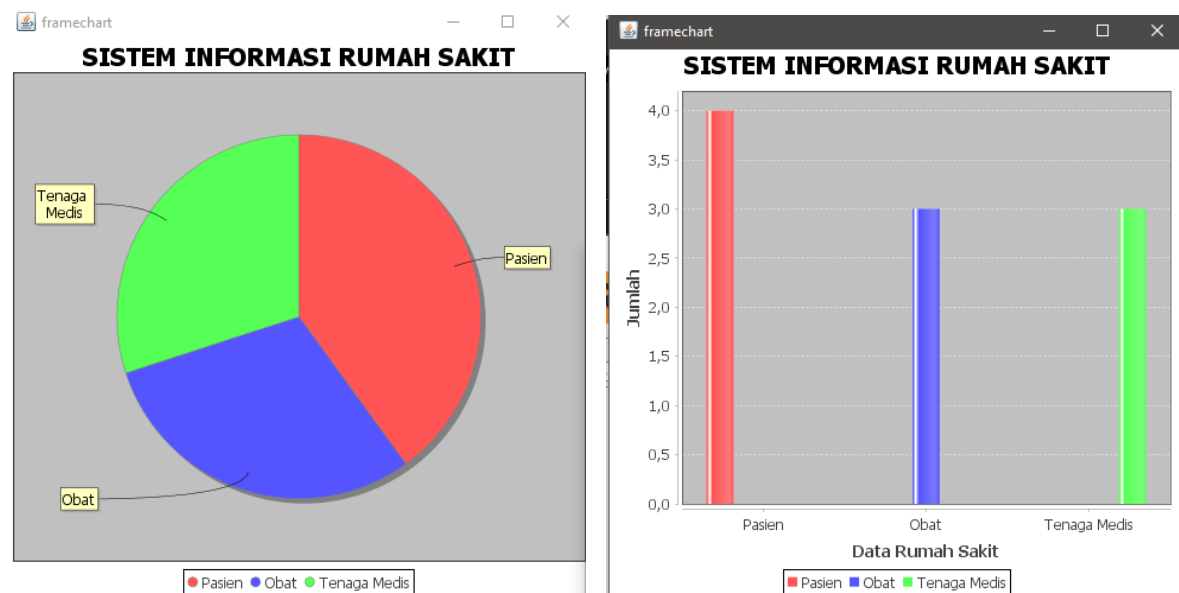
SIMPAN

HAPUS

GRAFIK PIE

GRAFIK BAR

Gambar 32 Tampilan Antarmuka Sub-Menu Data Chart Diagram



Gambar diatas merupakan tampilan antarmuka Sub-Menu Data Chart Diagram. Yang mana sub-sistem ini adalah sistem yang berfungsi untuk menampilkan hasil data data diantara pasien, obat, dan Tenaga Medis yang mana data yang ditampilkan nantinya berupa grafik.

Fitur yang bisa digunakan:

1. Pada menu sub-sistem ini user diberikan akses untuk menginputkan berapa banyak data Pasien, data Obat, dan data Tenaga Medis yang mana nantinya data akan ditampilkan dengan diagram.
2. User diberikan akses mengganti data yang akan diinputkan dengan cara menginputkan data baru lalu memerintahkan aplikasi untuk menyimpan dan menampilkan data kembali menggunakan diagram dengan menggunakan fitur “SIMPAN”.

3. User diberikan akses untuk menghapus data hasil inputkan, karena akan diganti dengan data yang baru dengan menggunakan fitur “Hapus”.
4. User diberikan akses untuk memilih metode penampilan data. Yaitu menggunakan grafik Pie atau grafik bar.

### **3.2.2 Penjelasan Keseluruhan Fitur dan Fungsi Aplikasi dari Proses Source Code (Kode Program) yang Menggunakan Bahasa Pemrograman “Java”**

Umumnya, secara keseluruhan pada pemrograman aplikasi Basis Data ini yang menggunakan bahasa pemrograman “java” hampir semua kode program memiliki kesamaan secara umum. Mungkin perbedaan yang mencolok adalah pemfokusan pada tahap pemrosesan string SQL/Query database yang akan dihubungkan dengan kode program “Java”.

Letak perbedaan secara umum diantaranya adalah sebagai berikut:

- a. Tahap pengkoneksian sistem database dengan kode program, yang menggunakan eksekusi SQL yang membutuhkan nama database.
- b. Setelah sistem database dan kode program “Java” sudah terhubung maka dibutuhkan kode program untuk mengakses data pada setiap tabel yang ada pada sistem database.
- c. Tahap pengambilan data pada setiap tabel yang ada pada sistem database akan diproses dengan kode program yang menggunakan eksekusi SQL yang membutuhkan nama dari setiap tabel (secara umum kode program nya sama, namun yang membedakan adalah saat tahap eksekusi string SQL untuk mengambil data nama dari setiap tabel).

Berikut ini adalah Lampiran Screenshoot Code Program “Java” dalam implementasi pembuatan aplikasi Sistem Basis Data yang bertemakan Sistem Informasi Rumah Sakit.

## Kode Pemrograman Koneksi

Kode program ini berfungsi untuk mengkoneksikan kode program “Java” dengan sistem basis data yang telah dibuat.

```
package TUBES;

import java.sql.DriverManager;

public class koneksi {
    private static java.sql.Connection koneksi;

    public static java.sql.Connection getkoneksi(){
        if (koneksi == null){
            try {
                String url = "jdbc:mysql://localhost:3306/rumah_sakit";
                String user = "root";
                String password = "";
                DriverManager.registerDriver(new com.mysql.jdbc.Driver());
                koneksi = DriverManager.getConnection(url, user, password);
                System.out.println("Berhasil Terhubung");
            } catch (Exception e) {
                System.out.println("Error");
            }
        }
        return koneksi;
    }

    public static void main(String args[]){
        getkoneksi();
    }
}
```

*Gambar 33 Kode Program Koneksi*

Kode program ini mengeksekusi string URL localhost PHPMyAdmin, untuk mengakses Sistem Database yang sudah kita rancang. Dengan memasukkan nama Database yaitu “rumah\_sakit” kedalam kode program pada bagian string URL.

```
String url = "jdbc:mysql://localhost:3306/rumah_sakit";
String user = "root";
String password = "";
```

Kode Program akan memproses string URL yang mengandung nama database yang sudah dibuat yaitu “rumah\_sakit”, dengan identitas user bernama “root”, dan dengan password yang kosong/tidak berpassword “”.

### Kode Program Menu Utama (Main Menu)

Kode program ini berfungsi untuk menjalin korelasi antar masing-masing Sub menu, sehingga user bisa melihat semua tampilan sub-menu yang ada secara keseluruhan pada satu layar tampilan.

```
package TUBES;

public class Menu extends javax.swing.JFrame {
    private String title;

    public Menu() {
        initComponents();
        setTitle ("Menu");
        setExtendedState (MAXIMIZED_BOTH);
    }
}
```

*Gambar 34 Kode Program Menu Utama*

Tampilan diatas merupakan kode program pendeklarasian dari sistem menu, yang mana program ini diberikan nama “Menu”. Selanjutnya program diisi oleh bagian inti dari program menu utama ini, yaitu proses korelasi dan pemanggilan terhadap setiap masing-masing program sub-menu, yang nantinya secara keseluruhan program sub-menu pada sistem aplikasi ini akan ditampilkan pada output interface dari program main menu.

Bagian ini berisi -----Program Inti Menu Utama-----yang akan dijelaskan di bagian halaman selanjutnya, diantaranya adalah:

1. Program yang bisa mengkorelasikan semua fungsi sub-sistem pada sistem “menu” ini, sehingga bisa diakses dengan cara meng-klik pilihan yang muncul di Menu Bar sistem yang berada di pojok kiri bagian atas.

```

@SuppressWarnings("unchecked")
Generated Code

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tbltenagamedis ().setVisible(true);
}

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tblpenyakit ().setVisible(true);
}

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tblperawat ().setVisible(true);
}

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tbldokter ().setVisible(true);
}

private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tblpasien ().setVisible(true);
}

private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tblobat ().setVisible(true);
}

private void jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tblrekammedis ().setVisible(true);
}

```

*Gambar 35 Kode Program Kolerasi Sub-Menu*

Kode program diatas adalah yang mengkorelasikan semua sub-menu agar bisa diakses melalui satu tampilan sekaligus, yaitu pada tampilan sistem “menu” pada menu bar. Seluruh sub sistem akan bisa diakses melalui sistem “menu” ini, dengan cara memanggil program sub-sistem tersebut melalui akses pemanggilan (void). Yang mana urutan akses pemanggilan yang didefinisikan pada program diatas adalah (Tabel Tenaga Medis, Tabel Penyakit, Tabel Perawat, Tabel Dokter, Tabel Pasien, Tabel Obat, Tabel Rekam Medis).

2. Program yang bisa mengkorelasikan semua fungsi sub-sistem pada sistem “menu” ini , sehingga bisa diakses dengan cara meng-klik pilihan yang muncul pada tiap Shortcut yang ada pada bagian tengah tampilan Sistem (pada baris pertama).



```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    new tbltenagamedis ().setVisible(true);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tbldokter ().setVisible(true);
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tblperawat ().setVisible(true);
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tblpenyakit ().setVisible(true);
}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tblpasien ().setVisible(true);
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tblobat ().setVisible(true);
}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tblrekammedis ().setVisible(true);
}

```

*Gambar 36 Kode Progrm Sub-Sistem Dengan Void*

ggil program sub-sistem tersebut melalui akses pemanggilan (void). Yang mana urutan akses pemanggilan yang didefinisikan pada program diatas adalah (Tabel Tenaga Medis, Tabel Dokter, Tabel Perawat, Tabel Penyakit, Tabel Pasien, Tabel Obat, Tabel Rekam Medis).

3. Program yang bisa mengkorelasikan semua fungsi sub-sistem tambahan/fitur tambahan pada sistem “menu” ini , sehingga bisa diakses dengan cara meng-klik pilihan yang muncul pada tiap Shortcut yang ada pada bagian tengah tampilan Sistem (pada baris kedua).

```

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tblperawat_pasien ().setVisible(true);
}

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tblmdosisobat ().setVisible(true);
}

private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new tblmnotelepon ().setVisible(true);
}

private void jButton11ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new jointabel ().setVisible(true);
}

private void jButton12ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new Grafik1 ().setVisible(true);
}

```

*Gambar 37 Kode Program Kolerasi Sub-Menu*

Kode program diatas adalah yang mengkorelasikan semua sub-menu agar bisa diakses melalui satu tampilan sekaligus, yaitu pada tampilan sistem “menu” melalui tombol/icon shortcut yang ada pada kategori fitur tambahan sistem (shortcut pada baris kedua).

Seluruh sub sistem akan bisa diakses melalui sistem “menu” ini, dengan cara memanggil program sub-sistem tersebut melalui akses pemanggilan (void). Yang mana urutan akses pemanggilan yang didefinisikan pada program diatas adalah (Tabel gabungan 2 tabel antara Perawat dan Pasien, Tabel Multivalue Dosis Obat, Tabel Muultivalue No Telepon, Tabel gabungan 3 tabel antara tabel Pasien, Obat, Rekam Medis dan terakhir adalah Tabel Grafik).

```

public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Menu().setVisible(true);
        }
    });
}

```

Tampilan diatas merupakan kode program penutup dari sistem menu, yang mana program ini diberikan nama “main” dengan tipe proses program pemanggilan (void).

## Penjelasan Kode Program Masing-masing Entitas Sub-Sistem

Pada umumnya, secara keseluruhan proses penulisan kode program adalah sama. Yang membedakan hanyalah pada pemrosesan string SQL yang menghubungkan antara kode program dengan sistem Basis Data. Jadi dari semua entitas Sub-Sistem yang ada, hanya menggunakan satu versi lengkap kode program saja. Namun ketika proses menkoneksi dan mengambil data dari kode program ke sistem Basis Data, menggunakan string SQL yang berbeda.

Sehingga pada umumnya perbedaan yang ada hanyalah pada pemrosesan string SQL yang menghubungkan antara kode program dengan sistem Basis Data. Yang lainnya secara umum menggunakan kode program yang sama.

Sebagai contoh dan penjelasan yang lebih lengkap, berikut ini akan ditampilkan salah satu kode program lengkap yang akan memperlihatkan secara lengkap dan terstruktur runtutan kode program yang dibuat untuk memproses suatu Entitas Sub-Sistem. Sampel kode program yang diambil adalah dari Tabel Dokter.

### Kode Program untuk menampilkan data secara terurut berdasarkan ID PRIMARY KEY, atau berdasarkan ID FOREIGN KEY

```
package TUBES;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class tblDokter extends javax.swing.JFrame {
    koneksi koneksi = new koneksi();

    private DefaultTableModel model;

    private void autonumber() {
        try {
            Connection c = koneksi.getkoneksi();
            Statement s = c.createStatement();
            String sql = "SELECT * FROM dokter ORDER BY Id_TenagaMedis DESC";
            ResultSet r = s.executeQuery(sql);
            if (r.next()) {
                String ID = r.getString("Id_TenagaMedis").substring(2);
                String Id_TenagaMedis = "" + (Integer.parseInt(ID) + 1);
                String Nol = "";

                if (Id_TenagaMedis.length() == 1) {
                    Nol = "00";
                } else if (Id_TenagaMedis.length() == 2) {
                    Nol = "0";
                } else if (Id_TenagaMedis.length() == 3) {
                    Nol = "";
                }

                txid.setText("ID" + Nol + Id_TenagaMedis);
            } else {
                txid.setText("ID001");
            }
            r.close();
            s.close();
        } catch (Exception e) {
            System.out.println("autonumber error");
        }
    }
}
```

*Gambar 38 Kode Program Untuk Menampilkan Data Terurut*

Gambar diatas adalah kode program awal pada Tabel Dokter yang berfungsi membuat penomoran otomatis pada data yang dimasukan diurutkan sesuai dengan data ID Tenaga Medis dimana ini merupakan FOREIGN KEY penghubung dari Tabel Tenaga Medis ke Tabel Dokter.

```
String sql = "SELECT * FROM dokter ORDER BY Id_TenagaMedis DESC";  
ResultSet r = s.executeQuery(sql);
```

Pada Gambar diatas terdapat string SQL yang dituliskan:

- String “Select \* FROM dokter” adalah command string untuk mengambil data dari Tabel Dokter
- String “..... ORDER BY Id\_TenagaMedis DESC” adalah command string untuk mengurutkan data yang sudah diambil dari tabel dokter. Diurutkan sesuai dengan urutan ID Tenaga Medis

Pada Entitas lain sebenarnya kode program yang dituliskan adalah sama, yang membedakan hanyalah pada penulisan string SQL nya saja. Yaitu nama Tabel apa yang data nya akan diambil dan dikoneksikan lalu di proses kedalam kode program.

### **Kode Program Untuk Meload/Mengambil Data Yang Ada Pada Sistem Basis Data**

```
7 public void clear(){  
8     txnama.setText("");  
9 }  
10 public void loadData(){  
11     model.getDataVector().removeAllElements();  
12     model.fireTableDataChanged();  
13  
14     try{  
15         Connection c = koneksi.getkoneksi();  
16         Statement s = c.createStatement();  
17  
18         String sql = "SELECT * FROM dokter";  
19         ResultSet r = s.executeQuery(sql);  
20  
21         while(r.next()){  
22             Object[] o = new Object[3];  
23             o [0] = r.getString("Id_TenagaMedis");  
24             o [1] = r.getString("Nama_Dokter");  
25             o [2] = r.getString("Spesialisasi_Dokter");  
26  
27             model.addRow(o);  
28         }  
29         r.close();  
30         s.close();  
31     }catch(SQLException e){  
32         System.out.println("terjadi kesalahan");  
33     }  
34 }  
35
```

*Gambar 39 Kode Program Untuk Mengambil Data Pada Sistem Basis Data*

Gambar diatas adalah kode program utama pada Tabel Dokter yang berfungsi meload/mengambil data yang tersimpan di sistem Basis Data.

```
String sql = "SELECT * FROM dokter";
ResultSet r = s.executeQuery(sql);

while(r.next()){
    Object[] o = new Object[3];
    o [0] = r.getString("Id_TenagaMedis");
    o [1] = r.getString("Nama_Dokter");
    o [2] = r.getString("Spesialisasi_Dokter");
}
```

Pada Gambar diatas terdapat string SQL yang dituliskan:

- String “Select \* FROM dokter” adalah command string untuk mengambil data dari Tabel Dokter
- String “..... r.getString(nama tiap field pada Tabel Dokter)” adalah command string untuk mengambil data pada field yang ada pada tabel Dokter. Data yang diambil akan disesuaikan dengan nama field yang dituliskan

```
public tblDokter() {
    initComponents();
    this.setLocationRelativeTo(null);

    txid.setEnabled(false);

    model = new DefaultTableModel();

    jTable2.setModel(model);

    model.addColumn("ID Tenaga Medis");
    model.addColumn("Nama Dokter");
    model.addColumn("Spesialisasi Dokter");

    loadData();
    autonumber();
}
```

Gambar diatas adalah kode program terakhir pada Tabel Dokter yang berfungsi untuk memvisualisasikan/menampilkan data-data yang sudah di load pada program sebelumnya dan ditampilkan ke tabel yang sudah dibuat pada interface Tabel Dokter.

Pada Tabel Entitas lain secara umum tetap menggunakan susunan kode yang sama seperti kode program sebelumnya yang screenshot nya sudah dilampirkan diatas. Namun memiliki perbedaan ketika akan mengakses field pada Tabel menggunakan string SQL, beberapa perbedaan penggunaan string SQL pada entitas Tabel lainnya diantara adalah sebagai berikut:

Berikut adalah screenshoot string SQL yang membedakan kode program antara masing-masing entitas Sub-Sistem:

- Sub-Menu Data Obat

```
String sql = "SELECT * FROM obat ORDER BY Id_Obat DESC";  
ResultSet r = s.executeQuery(sql);
```

Kode Program untuk pengurutan data yang akan telah diinputkan oleh user yang disimpan pada sistem database

```
String sql = "SELECT * FROM obat";  
ResultSet r = s.executeQuery(sql);  
  
while(r.next()){  
    Object[] o = new Object[7];  
    o [0] = r.getString("Id_Obat");  
    o [1] = r.getString("Nama_Obat");  
    o [2] = r.getString("Jenis_Obat");  
    o [3] = r.getString("Harga_Obat");  
    o [4] = r.getString("Tanggal_Kadaluarsa");  
    o [5] = r.getString("Id_TenagaMedis");  
    o [6] = r.getString("Id_Pasien");  
    model.addRow(o);  
}
```

Kode Program untuk meload/mengambil data dari Database

```
jTable4.setModel(model);  
  
model.addColumn("ID Obat");  
model.addColumn("Nama Obat");  
model.addColumn("Jenis Obat");  
model.addColumn("Harga Obat");  
model.addColumn("Tanggal Kadaluarsa");  
model.addColumn("ID Tenaga Medis");  
model.addColumn("ID Pasien");  
  
loadData();  
autonumber();
```

Kode Program untuk memvisualisasikan/menampilkan data dari database

- Sub-Menu Pasien

```
String sql = "SELECT * FROM pasien ORDER BY Id_Pasien DESC";  
ResultSet r = s.executeQuery(sql);
```

Kode Program untuk pengurutan data yang akan telah diinputkan oleh user yang disimpan pada sistem database.

```
String sql = "SELECT * FROM pasien";
ResultSet r = s.executeQuery(sql);

while(r.next()){
    Object[] o = new Object[7];
    o [0] = r.getString("Id_Pasien");
    o [1] = r.getString("Nama_Pasien");
    o [2] = r.getString("Tanggal_Lahir");
    o [3] = r.getString("Ruangan_Pasien");
    o [4] = r.getString("Alamat_Pasien");
    o [5] = r.getString("Id_TenagaMedis");
    o [6] = r.getString("Umur");
    model.addRow(o);
}
```

Kode Program untuk meload/mengambil data dari Database

```
model = new DefaultTableModel();

jTable2.setModel(model);

model.addColumn("ID Pasien");
model.addColumn("Nama Pasien");
model.addColumn("Tanggal Lahir");
model.addColumn("Ruangan Pasien");
model.addColumn("Alamat Pasien");
model.addColumn("ID Tenaga Medis");
model.addColumn("Umur Pasien");

loadData();
autonumber();
```

Kode Program untuk memvisualisasikan/menampilkan data dari database

- Sub Query (Umur Pasien Tertua)

```
String sql = "SELECT Id_Pasien, Nama_Pasien, Alamat_Pasien from pasien where Umur = (select max(Umur) from pasien)";
ResultSet r = s.executeQuery(sql);

while(r.next()){
    Object[] o = new Object[3];
    o [0] = r.getString("Id_Pasien");
    o [1] = r.getString("Nama_Pasien");
    o [2] = r.getString("Alamat_Pasien");

    model.addRow(o);
}
```

Kode Program untuk meload/mengambil data dari Database

```
jTable1.setModel(model);

model.addColumn("ID Pasien");
model.addColumn("Nama Pasien");
model.addColumn("Alamat Pasien");
```

Kode Program untuk memvisualisasikan/menampilkan data dari database

- Sub Query (Umur Pasien Termuda)

```
private DefaultTableModel model;
public void loadData(){
    model.getDataVector().removeAllElements();
    model.fireTableDataChanged();

    try{
        Connection c = koneksi.getkoneksi();
        Statement s = c.createStatement();

        String sql = "SELECT Id_Pasien, Nama_Pasien, Alamat_Pasien from pasien where Umur = (select min(Umur) from pasien)";
        ResultSet r = s.executeQuery(sql);

        while(r.next()){
            Object[] o = new Object[3];
            o[0] = r.getString("Id_Pasien");
            o[1] = r.getString("Nama_Pasien");
            o[2] = r.getString("Alamat_Pasien");

            model.addRow(o);
        }
    }
```

Kode Program untuk meload/mengambil data dari database

```
jTable1.setModel(model);

model.addColumn("ID Pasien");
model.addColumn("Nama Pasien");
model.addColumn("Alamat Pasien");

loadData();
}
```

Kode Program untuk memvisualisasikan/menampilkan data dari database

- Sub-Menu Penyakit

```
String sql = "SELECT * FROM penyakit ORDER BY Id_Pasien DESC";
ResultSet r = s.executeQuery(sql);
```



Kode Program untuk pengurutan data yang akan telah diinputkan oleh user yang disimpan pada sistem database

```
String sql = "SELECT * FROM penyakit";
ResultSet r = s.executeQuery(sql);

while(r.next()){
    Object[] o = new Object[4];
    o [0] = r.getString("Id_Pasien");
    o [1] = r.getString("Nama_Penyakit");
    o [2] = r.getString("Gejala_Penyakit");
    o [3] = r.getString("Id_RekamMedis");

    model.addRow(o);
}
```

Kode Program untuk meload/mengambil data dari Database

```
model = new DefaultTableModel();

jTable2.setModel(model);

model.addColumn("ID Pasien");
model.addColumn("Nama Penyakit");
model.addColumn("Gejala Penyakit");
model.addColumn("ID Rekam Medis");

loadData();
autonumber();
```

Kode Program untuk memvisualisasikan/menampilkan data dari database

- Sub-Menu Perawat

```
String sql = "SELECT * FROM perawat ORDER BY Id_TenagaMedis DESC";
ResultSet r = s.executeQuery(sql);
```

Kode Program untuk pengurutan data yang akan telah diinputkan oleh user yang disimpan pada sistem database

```
String sql = "SELECT * FROM perawat";
ResultSet r = s.executeQuery(sql);

while(r.next()){
    Object[] o = new Object[2];
    o [0] = r.getString("Id_TenagaMedis");
    o [1] = r.getString("Nama_Perawat");
    model.addRow(o);
}
```

Kode Program untuk meload/mengambil data dari database

```
model = new DefaultTableModel();

jTable3.setModel(model);

model.addColumn("ID Tenaga Medis");
model.addColumn("Nama Perawat");

loadData();
autonumber();
```

Kode Program untuk memvisualisasikan/menampilkan data dari database

- Sub-Menu Rekam Medis

```
String sql = "SELECT * FROM rekam_medis ORDER BY Id_RekamMedis DESC";
ResultSet r = s.executeQuery(sql);
```

Kode Program untuk pengurutan data yang akan telah diinputkan oleh user yang disimpan pada sistem database

```
String sql = "SELECT * FROM rekam_medis";
ResultSet r = s.executeQuery(sql);

while(r.next()){
    Object[] o = new Object[3];
    o[0] = r.getString("Id_RekamMedis");
    o[1] = r.getString("Tanggal_Pemeriksaan");
    o[2] = r.getString("Gejala_Penyakit");

    model.addRow(o);
}
```

Kode Program untuk meload/mengambil data dari Database

```
model = new DefaultTableModel();

jTable1.setModel(model);

model.addColumn("ID Rekam Medis");
model.addColumn("Tanggal Pemeriksaan");
model.addColumn("Gejala Penyakit");

loadData();
autonumber();
```

Kode Program untuk memvisualisasikan/menampilkan data dari database

- Sub-Menu Tenaga Medis

```
String sql = "SELECT * FROM tenaga_medis ORDER BY Id_TenagaMedis DESC";
ResultSet r = s.executeQuery(sql);
```

Kode Program untuk pengurutan data yang akan telah diinputkan oleh user yang disimpan pada sistem database

```
String sql = "SELECT * FROM tenaga_medis";
ResultSet r = s.executeQuery(sql);

while(r.next()){
    Object[] o = new Object[2];
    o [0] = r.getString("Id_TenagaMedis");
    o [1] = r.getString("Jenis_Kelamin");

    model.addRow(o);
}
```

Kode Program untuk meload/mengambil data dari Database

```
data.setSelectedIndex(1);

model = new DefaultTableModel();

jTable1.setModel(model);

model.addColumn("ID Tenaga Medis");
model.addColumn("Jenis Kelamin");

loadData();
autonumber();
```

Kode Program untuk memvisualisasikan/menampilkan data dari database

- Sub-Menu Multivalue Dosis Obat

```
String sql = "SELECT * FROM multivalue_dosisobat";
ResultSet r = s.executeQuery(sql);
```

Kode Program untuk meload/mengambil data dari Database

```
model = new DefaultTableModel();

jTable1.setModel(model);

model.addColumn("ID Obat");
model.addColumn("Dosis Obat");

loadData();
```

Kode Program untuk memvisualisasikan/menampilkan data dari database

- Sub-Menu Multivalue No Telepon

```
String sql = "SELECT * FROM multivalue_notelpon";
ResultSet r = s.executeQuery(sql);

while(r.next()){
    Object[] o = new Object[2];
    o [0] = r.getString("Id_Pasien");
    o [1] = r.getString("Nomor_Telpon");
}
```

Kode Program untuk meload/mengambil data dari Database

```
model = new DefaultTableModel();

jTable1.setModel(model);

model.addColumn("ID Pasien");
model.addColumn("Nomor Telepon");

loadData();
```

Kode Program untuk memvisualisasikan/menampilkan data dari database

- Sub-Menu Join Tabel/Penggabungan Tabel (2 tabel)

```
String sql = "SELECT perawat.Id_TenagaMedis, perawat>Nama_Perawat, pasien.Id_Pasien, pasien>Nama_Pasien FROM perawat INNER JOIN pasien";
ResultSet r = s.executeQuery(sql);

while(r.next()){
    Object[] o = new Object[4];
    o [0] = r.getString("Id_TenagaMedis");
    o [1] = r.getString("Nama_Perawat");
    o [2] = r.getString("Id_Pasien");
    o [3] = r.getString("Nama_Pasien");

    model.addRow(o);
}
```

Kode Program untuk meload/mengambil data dari Database

```
model = new DefaultTableModel();

jTable1.setModel(model);

model.addColumn("ID Tenaga Medis");
model.addColumn("Nama Perawat");
model.addColumn("ID Pasien");
model.addColumn("Nama Pasien");

loadData();
```

Kode Program untuk memvisualisasikan/menampilkan data dari database

- Sub-Menu Join Tabel/Penggabungan Tabel (3 Tabel)

```
String sql = "SELECT pasien.Id_Pasien, pasien>Nama_Pasien, obat>Nama_Obat, rekam_medis.Tanggal_Pemeriksaan, obat.Harga_Obat FROM pasien JOIN obat";
ResultSet r = s.executeQuery(sql);

while(r.next()){
    Object[] o = new Object[5];
    o [0] = r.getString("pasien.Id_Pasien");
    o [1] = r.getString("pasien>Nama_Pasien");
    o [2] = r.getString("obat>Nama_Obat");
    o [3] = r.getString("rekam_medis.Tanggal_Pemeriksaan");
    o [4] = r.getString("obat.Harga_Obat");

    model.addRow(o);
}
```

Kode Program untuk meload/mengambil data dari Database

```
jTable1.setModel(model);

model.addColumn("ID Pasien");
model.addColumn("Nama Pasien");
model.addColumn("Nama Obat");
model.addColumn("Tanggal Pemeriksaan");
model.addColumn("Harga Obat");

loadData();
}
```

Kode Program untuk memvisualisasikan/menampilkan data dari database

- Sub-Menu Diagram

```
public void loadData() {
    model.getDataVector().removeAllElements();
    model.fireTableDataChanged();

    try{
        Connection c = koneksi.getkoneksi();
        Statement s = c.createStatement();

        String sql = "SELECT * FROM grafikdatabase;";

        ResultSet a = s.executeQuery(sql);

        while(a.next()){
            Object[] o = new Object[3];
            o [0] = a.getString("Pasien");
            o [1] = a.getString("Obat");
            o [2] = a.getString("TenagaMedis");

            model.addRow(o);
        }
    }
}
```

Kode Program untuk meload/mengambil data dari Database

```
// TODO add your handling code here:  
  
jTable1.setModel(model);  
  
model.addColumn("Pasien");  
model.addColumn("Obat");  
model.addColumn("Tenaga Medis");
```

Kode Program untuk memvisualisasikan/menampilkan data dari database

### Penjelasan Kode Program Yang Mengoperasikan Setiap Tombol Tombol Yang Ada Pada Setiap Entitas Sub-Sistem Yang Ada

Pada umumnya, secara keseluruhan proses penulisan kode program pada penggunaan fungsi button/tombol/icon pada sistem juga adalah sama. antara masing-masing entitas Sub-Sistem. Yang membedakannya adalah pemrosesan string SQL yang menggunakan nama Tabel beserta nama-nama field yang ada pada tabel yang akan diambil data nya

Beberapa contoh kode program dari fungsi tombol yang digunakan diantaranya:

- Simpan

```
private void btnsimpanActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String Id_TenagaMedis = txid.getText();  
    String Nama_Dokter = txnama.getText();  
    String Spesialisasi_Dokter = (String)jspes.getSelectedItemAt();  
  
    try{  
        Connection c = koneksi.getkoneksi();  
        String sql = "INSERT INTO dokter VALUES (?, ?, ?)";  
        PreparedStatement p = c.prepareStatement(sql);  
        p.setString(1, Id_TenagaMedis);  
        p.setString(2, Nama_Dokter);  
        p.setString(3, Spesialisasi_Dokter);  
  
        p.executeUpdate();  
        p.close();  
        JOptionPane.showMessageDialog(null, "Data Tersimpan");  
    }catch(SQLException e){  
        System.out.println("Terjadi Kesalahan");  
    }finally{  
        loadData();  
        autonumber();  
        clear ();  
    }  
}
```

- Edit

```
private void btneditActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int i = jTable2.getSelectedRow();
    if(i == -1){
        return;
    }
    String Id_TenagaMedis = (String)model.getValueAt(i, 0);
    String Nama_Dokter = txnama.getText();
    String Spesialisasi_Dokter = (String)jspes.getSelectedItemAt();

    try{
        Connection c = koneksi.getkoneksi();
        String sql = "UPDATE dokter SET Nama_Dokter = ?, Spesialisasi_Dokter = ? WHERE Id_TenagaMedis = ?";
        PreparedStatement p = c.prepareStatement(sql);
        p.setString(1, Nama_Dokter);
        p.setString(2, Spesialisasi_Dokter);
        p.setString(3, Id_TenagaMedis);

        p.executeUpdate();
        p.close();
        JOptionPane.showMessageDialog(null, "Data Terubah");
        btnsimpan.setEnabled(true);
        clear();
    }catch(SQLException e){
        System.out.println("Update Error");
    }finally{
        loadData();
        autonumber();
    }
}
```

- Hapus

```
private void btnhapusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int i = jTable2.getSelectedRow();
    if(i == -1){
        return;
    }

    String Id_TenagaMedis = (String)model.getValueAt(i, 0);

    int question = JOptionPane.showConfirmDialog(null, "Yakin Data Akan Dihapus?", "Konfirmasi", JOptionPane.OK_CANCEL_OPTION,
        JOptionPane.QUESTION_MESSAGE);
    if(question == JOptionPane.OK_OPTION){
        try{
            Connection c = koneksi.getkoneksi();
            String sql = "DELETE FROM dokter WHERE Id_TenagaMedis = ?";
            PreparedStatement p = c.prepareStatement(sql);
            p.setString(1, Id_TenagaMedis);

            p.executeUpdate();
            p.close();
            JOptionPane.showMessageDialog(null, "Data Terhapus");
        }catch(SQLException e){
            System.out.println("Terjadi Kesalahab");
        }finally{
            loadData();
            autonumber();
            clear();
        }
    }
    if(question == JOptionPane.CANCEL_OPTION){
    }
}
```

- Batal

```
private void btnbatalActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    clear ();
    loadData();
    btnsimpan.setEnabled(true);
    autonumber();
}
```

- Cari

```
private void txcariKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    DefaultTableModel tabel = new DefaultTableModel();

    tabel.addColumn("Id_TenagaMedis");
    tabel.addColumn("Nama_Dokter");
    tabel.addColumn("Spesialisasi_Dokter");

    try{
        Connection c = koneksi.getkoneksi();
        String sql = "Select * from dokter where Nama_Dokter like '%" + txcari.getText() + "%'";
        Statement stat = c.createStatement();
        ResultSet rs = stat.executeQuery(sql);
        while(rs.next()){
            tabel.addRow(new Object[]{
                rs.getString(1),
                rs.getString(2),
                rs.getString(3),
            });
        }
        jTable2.setModel(tabel);
        loadData();
    }catch(Exception e){
        System.out.println("Cari Data Error");
    }finally{
    }
}
```

- Fungsi (Klik) pada tombol touchpad/mouse

```
private void jTable2MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    btnsimpan.setEnabled(false);
    int i = jTable2.getSelectedRow();
    if (i == -1){
        return;
    }

    String Id_TenagaMedis = (String)model.getValueAt(i, 0);
    txid.setText(Id_TenagaMedis);
    String Nama_Dokter = (String)model.getValueAt(i, 1);
    txnama.setText(Nama_Dokter);
    String Spesialisasi_Dokter = (String)model.getValueAt(i, 2);
    jspes.setSelectedItem(Spesialisasi_Dokter);
}
```



## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Kesimpulan dari tugas besar yang kami buat adalah:

1. Dalam membuat SIRS terlebih dahulu kita harus merancang kebutuhan apa saja yang diperlukan di dalam SIRS.
2. Selanjutnya kita menyusun kebutuhan tersebut ke dalam Use Case Diagram dan merancang ERD.
3. Setelah memiliki ERD kita menentukan bahasa pemrograman apa yang akan kita gunakan untuk mengaplikasikan sistem yang sudah direncanakan.

#### **4.2 Saran**

Saran yang kami dapatkan setelah mengerjakan tugas besar ini adalah untuk para teman kami semua yang sedang mengerjakan tugas tetaplah saling menyemangati. Terkhusus pada kondisi saat ini, sulitnya untuk bertemu tatap muka dan kendala teknis untuk melakukan pertemuan melalui dunia digital, membuat pengerjaan tugas besar menjadi hal yang perlu benar-benar dipersiapkan dengan pengkondisian yang baik.

## DAFTAR PUSTAKA

- Dimas, S., & 2, Z. (2020, November 07). Contoh ERD Sistem Informasi Rumah Sakit. Retrieved May 10, 2021, from <https://kelasprogrammer.com/contoh-erd-sistem-informasi-rumah-sakit/>
- Krakatau\_medika. (n.d.). Sistem Informasi Manajemen Rumah sakit (simrs). Retrieved May 10, 2021, from <https://krakataumedika.com/info-media/artikel/sistem-informasi-manajemen-rumah-sakit-simrs>
- Membuat Aplikasi CRUD di netbeans Dengan mysql database* [Video file]. (2020, November 13). Retrieved May 10, 2021, from <https://youtu.be/VLIDpw74hN0>
- Membuat menu utama di Netbeans - mdi form* [Video file]. (2020, December 04). Retrieved May 10, 2021, from <https://youtu.be/-xf0BfgEETo>
- Spesialisasi/Generalisasi relationship. (n.d.). Retrieved May 10, 2021, from <http://pbokelasc.blogspot.com/2012/06/spesialisasigeneralisasi-relationship.html>
- Teori LENGKAP tentang Sistem Informasi Rumah Sakit Menurut PARA Ahli dan Contoh TESIS Sistem Informasi Rumah Sakit. (n.d.). Retrieved May 10, 2021, from <https://idtesis.com/teori-lengkap-tentang-sistem-informasi-rumah-sakit-menurut-para-ahli-dan-contoh-tesis-sistem-informasi-rumah-sakit/#:~:text=Definisi%20Sistem%20Informasi%20Rumah%20Sakit,dibutuhkan%20untuk%20kegiatan%20rumah%20sakit.>

## LOG SHEET

### Log Sheet Activity

No	Tanggal	Nama	NIM	Kegiatan
1.	25 April 2021	Desi Togi Sintauli Tambunan	119140073	Merancang ERD
		Makruf Alkarkhi	119140075	Merancang ERD
		Muhammad Asyroful Nur Maulana Yusuf	119140026	Pembuatan code program dan design UI (Data Tenaga Medis, Data Dokter, Data Perawat)
2.	26 April 2021	Desi Togi Sintauli Tambunan	119140073	Merevisi ERD
		Makruf Alkarkhi	119140075	Merevisi ERD
		Muhammad Asyroful Nur Maulana Yusuf	119140026	Pembuatan code program dan design UI (Data Obat dan Data Pasien)
3.	27 April 2021	Desi Togi Sintauli Tambunan	119140073	Membuat query mysql dari database yang sudah dirancang
		Makruf Alkarkhi	119140075	Merancang Database Mapping ERD ke Relational Model
		Muhammad Asyroful Nur Maulana Yusuf	119140026	Pembuatan code program dan design UI (Data Penyakit, Data Rekam Medis)
4.	28 April 2021	Muhammad Asyroful Nur Maulana Yusuf	119140026	Merevisi code pada data-data yang sudah dibuat.
		Reza Kusuma Andya Nugraha	119140001	Membuat sebagian proses kode program koneksi
5.	29 April 2021	Muhammad Asyroful Nur Maulana Yusuf	119140026	Pembuatan code program dan design UI (Tabel Relasi, Join Table)
		Reza Kusuma Andya Nugraha	119140001	Membuat sebagian proses design dan kode program 1. relasi perawat – pasien 2. Relasi pasien - rekam medis
		Fathimatul Mahmudzah	119140168	Pembuatan Kerangka Laporan
6.	30 April 2021	Muhammad Asyroful Nur Maulana Yusuf	119140026	Pembuatan code program dan design UI (Grafik)
		Reza Kusuma Andya Nugraha	119140001	Membuat sebagian proses design dan kode program 1. Multivalue Dosis Obat 2. Multivalue No Telepon
		Fathimatul Mahmudzah	119140168	Pengisian BAB 1
7.	1 Mei 2021	Muhammad Asyroful Nur Maulana Yusuf	119140026	Pembuatan Design UI dan code program mengenai Tampilan utama Aplikasi

		Reza Kusuma Andya Nugraha	119140001	Membuat sebagian proses kode program grafik bar
8.	3 Mei 2021	Muhammad Asyroful Nur Maulana Yusuf	119140026	Pemrosesan Java ke Java Exe File
		Reza Kusuma Andya Nugraha	119140001	Mengecek dan memeriksa keseluruhan proses eksekusi program
9.	4 Mei 2021	Reza Kusuma Andya Nugraha	119140001	Membuat laporan pada bagian BAB 2 yg berfokus pada bagian pemrograman
10.	5 Mei 2021	Desi Togi Sintauli Tambunan	119140073	Membuat Use Case Diagram
		Makruf Alkarkhi	119140075	Membuat Use Case Diagram
		Muhammad Asyroful Nur Maulana Yusuf	119140026	Pengambilan video demo Aplikasi
		Reza Kusuma Andya Nugraha	119140001	Membuat laporan pada bagian BAB 3 yang berfokus pada bagian pemrograman (menyelesaikan tahapan penjelasan dari interface/ tampilan dari sistem program)
		Fathimatul Mahmudzah	119140168	Mengatur format laporan BAB 2 dan BAB 3
11.	6 Mei 2021	Desi Togi Sintauli Tambunan	119140073	Memasukkan data ke dalam database mysql
		Makruf Alkarkhi	119140075	Mengerjakan laporan bagian Bab II dari point 1.1 sampai 1.3 bagian B.
12.	7 Mei 2021	Desi Togi Sintauli Tambunan	119140073	Mengerjakan laporan bagian Bab II point 1.3 yang C dan Bab III bagian 3.1
13.	8 Mei 2021	Fathimatul Mahmudzah	119140168	Mengerjakan laporan BAB 4 dan BAB 5
14.	9 Mei 2021	Fathimatul Mahmudzah	119140168	Mengatur dan Memformat laporan dari cover hingga daftar pustaka
15.	10 Mei 2021	Fathimatul Mahmudzah	119140168	Mengatur dan Memformat data diri
16.	11 Mei 2021	Fathimatul Mahmudzah	119140168	Mengatur dan Memformat Log Sheet. Serta melakukan finishing laporan

## Bukti Pengadaan Meet Kelompok

19 April 2021 Merancang Basis Data (Desi, Fathimatul, Maqruf, Muhammad, Reza)

The screenshot shows a Google Meet interface. At the top, a banner indicates "04\_026\_Yusuf is presenting". The main window displays a presentation slide with the following text:

PASIENT -> REKAM MEDIS (PERIKSA)  
REKAM MEDIS -> PENYAKIT (MEMERIKSA)

Below the slide, there is a list of participants in the meeting:

- 04\_026\_Yusuf
- Reza Kusuma
- Makruf Alkarkhi
- Fathimatul Mahmudzah

At the bottom of the screen, there are controls for the meeting, including a "Meeting details" link, a "Turn on captions" button, and a "04\_026\_Yusuf is presenting" status indicator.

23 April 2021 Memfix-an ERD (Desi, Maqruf, Reza, Muhammad)

The screenshot shows a Google Meet interface. At the top, a banner indicates "MANM Yusuf is presenting". The main window displays an Entity-Relationship Diagram (ERD) with the following entities and relationships:

- Entities: **Rekam\_Medis** (Primary Key), **Pasien** (Primary Key), **Penyakit** (Primary Key), **Obat** (Primary Key), **Obat\_Resep** (Primary Key), **Obat\_Konsumsi** (Primary Key), **Obat\_Konsumsi\_Resep** (Primary Key).
- Relationships: **Rekam\_Medis** is connected to **Pasien** and **Penyakit**. **Pasien** is connected to **Obat**. **Penyakit** is connected to **Obat**. **Obat** is connected to **Obat\_Resep** and **Obat\_Konsumsi**. **Obat\_Resep** is connected to **Obat\_Konsumsi**.

Below the ERD, there is a list of participants in the meeting:

- Makruf Alkarkhi
- Reza Kusuma
- MANM Yusuf

At the bottom of the screen, there are controls for the meeting, including a "Meeting details" link, a "Turn on captions" button, and a "MANM Yusuf is presenting" status indicator.

## 26 April 2021 Menentukan Tabel Basis Data (Desi, Maqruf, Muhammad)

04\_026\_Yusuf is presenting

phpMyAdmin

Database: rumah\_sakit

Tabel: obat

#	Nama	Jenis	Penyortiran	Atribut	Tak Terbilang	Bawahan	Komentar	Ekstra	Tindakan
1	ID Obat	char(10)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
2	Nama Obat	text	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
3	Jenis Obat	char(10)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
4	Tanggal Kadaluarsa	date		Tidak	Tidak ada				Ubah Hapus Lainnya
5	Dosis Obat	char(10)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
6	Harga Obat	int(6)		Tidak	Tidak ada				Ubah Hapus Lainnya

Meeting details

Turn on captions

04\_026\_Yusuf is presenting

## 28 April 2021 Membahas Visualisasi Data Program (Muhammad, Reza)

MANM Yusuf sedang melakukan presentasi

APK TUBES BASDAT - NetBeans IDE 8.2

```
try {
    Connection c = koneksi.getKoneksi();
    String sql = "INSERT INTO dokter VALUES (?, ?, ?)";
    PreparedStatement p = c.prepareStatement(sql);
    p.setString(1, Id_TesngMedis);
    p.setString(2, Nama_Dokter);
    p.setString(3, Spesialisasi_Dokter);
    p.executeUpdate();
    p.close();
    JOptionPane.showMessageDialog(null, "Data Tersimpan");
} catch (SQLException e) {
    System.out.println("Terjadi Kesalahan");
} finally {
    loadData();
    autonumber();
    clear();
}
```

private void btneditActionPerformed(java.awt.event.ActionEvent evt) {
 // TODO add your handling code here:
 int i = jTable2.getSelectedRow();
 if (i == -1) {
 return;
 }
 String Id\_TesngMedis = (String)model.getValueAt(i, 0);
 String Nama\_Dokter = txtNama.getText();
 String Spesialisasi\_Dokter = txtSpesialisasi.getText();
}

Meeting details

MANM Yusuf sedang melakukan presentasi

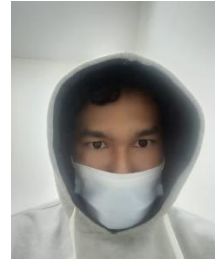
## BIODATA DAN LINGKUP KERJA ANGGOTA

### Ketua:

---

Nama : Muhammad Asyroful Nur Maulana Yusuf  
NIM : 119140026  
Prodi : Teknik Informatika  
Tempat, tanggal lahir : Seputih Mataram, 12 Juli 2001  
No WA : 085267228032  
Lingkup Kerja : Rancangan dan Program

---



### Anggota (Berdasarkan Abjad):

---

Nama : Desi Togi Sintauli Tambunan  
NIM : 119140073  
Prodi : Teknik Informatika  
Tempat, tanggal lahir : Bandar Lampung, 10 Desember 2000  
No WA : 0895323091987  
Lingkup Kerja : Database dan Laporan Lingkup Database

---



---

Nama : Fathimatul Mahmudzah  
NIM : 119140168  
Prodi : Teknik Informatika  
Tempat, tanggal lahir : Jati Indah, 18 April 2000  
No WA : 082371814033  
Lingkup Kerja : Kerangka Laporan dan Laporan Akhir

---



---

Nama : Makruf Alkarkhi  
NIM : 119140075  
Prodi : Teknik Informatika  
Tempat, tanggal lahir : Jakarta, 9 April 2001  
No WA : 119140075  
Lingkup Kerja : Database dan Laporan Lingkup Database

---



---

Nama : Reza Kusuma Andya Nugraha  
NIM : 119140001  
Prodi : Teknik Informatika  
Tempat, tanggal lahir : Gisting, 29 Agustus 2001  
No WA : 089625164963  
Lingkup Kerja : Rancangan dan Program

---

