

III. MAPPING DAN RELATIONAL MODEL

DOKTER

Nama_dokter	gender	Nama_spesialisasi	<u>Id_dokter</u>	alamat	Nomer_telp
-------------	--------	-------------------	------------------	--------	------------

ICU

<u>Id_ICU</u>	Nama_ruang
---------------	------------

PERAWAT

<u>Id_perawat</u>	Nama_perawat	gender	Jam_jaga
-------------------	--------------	--------	----------

POLI

<u>Id_poli</u>	Nama_poli
----------------	-----------

IGD

<u>Id_IGD</u>	Nama_ruang	<u>Id_dokter</u>
---------------	------------	------------------

STAFF

<u>Id_staff</u>	Nama_staff	alamat	gender	divisi	No_telp_staff
-----------------	------------	--------	--------	--------	---------------

PASIEN

<u>Id_pasien</u>	Nama_pasien	No_telp_pasien	alamat	gender	<u>Id_icu</u>	<u>Id_perawat</u>	<u>Id_poli</u>
------------------	-------------	----------------	--------	--------	---------------	-------------------	----------------

RESEP

<u>Id_resep</u>	Tanggal_resep	<u>Id_dokter</u>	<u>Id_pasien</u>
-----------------	---------------	------------------	------------------

RADIOLOGI

<u>Id_radiologi</u>	Tindakan_radio	<u>Id_rekamMedis</u>
---------------------	----------------	----------------------

ADMINISTRASI

<u>Id_admin</u>	Tanggal_admin	<u>Id_dokter</u>	<u>Id_kamar</u>	<u>Id_resep</u>	<u>Id_pemeriksaan</u>	<u>Id_pasien</u>
-----------------	---------------	------------------	-----------------	-----------------	-----------------------	------------------

RAWAT_INAP

<u>Id_kamar</u>	Nama_kelas
-----------------	------------

PEMERIKSAAN

<u>Id_pemeriksaan</u>	Cat_pemeriksaan	Jenis_pemeriksaan	<u>Id_poli</u>	<u>Id_ICU</u>	<u>Id_IGD</u>
-----------------------	-----------------	-------------------	----------------	---------------	---------------

LABORATORIUM

<u>Id_lab</u>	Nama_ruang
---------------	------------

ASURANSI

<u>Id_asuransi</u>	rujukan	Nama_asuransi
--------------------	---------	---------------

REKAM_MEDIS

<u>Id_RekamMedis</u>	Cat_rekammedis	<u>Id_pasien</u>
----------------------	----------------	------------------

Memiliki (dokter → poli)

<u>Id_dokter</u>	<u>Id_poli</u>
------------------	----------------

Melayani (staff → pasien)

<u>Id_staff</u>	<u>Id_pasien</u>
-----------------	------------------

Memeriksa (dokter → pasien)

<u>Id_dokter</u>	<u>Id_pasien</u>
------------------	------------------

Bagian_dari (Radiologi → pemeriksaan)

<u>Id_radiologi</u>	<u>Id_pemeriksaan</u>
---------------------	-----------------------

Dibutuhkan (IGD → pasien)

<u>Id_igd</u>	<u>Id_pasien</u>
---------------	------------------

Meliputi (administrasi → asuransi)

<u>Id_adm</u>	<u>Id_asuransi</u>
---------------	--------------------

Memerlukan (pemeriksaan → laboratorium)

<u>Id_pemeriksaan</u>	<u>Id_lab</u>
-----------------------	---------------

NB : Hijau : hasil relasi mapping

IV. IMPLEMENTASI

```
66 public void tampilTabel(){
67     DefaultTableModel model = new DefaultTableModel();
68     model.addColumn("ID");
69     model.addColumn("Nama");
70     model.addColumn("Spesialisasi");
71     model.addColumn("Jenis Kelamin");
72     model.addColumn("Alamat");
73     model.addColumn("No.Telefon");
74     try {
75         String sql = "select * from dokter";
76         Connection con=DriverManager.getConnection("jdbc:mysql://localhost/rumah_sakit","root","");
77         java.sql.Statement stm=con.createStatement();
78         java.sql.ResultSet res=stm.executeQuery(sql);
79         while(res.next()){
80             model.addRow(new Object[]{res.getString(1),res.getString(2),res.getString(3),res.getString(4),res.getString(5),res.getString(6)});
81         }
82         table.setModel(model);
83     } catch (Exception e) {
84         e.printStackTrace();
85     }
86 }
87 }
```

Pada gambar diatas adalah method untuk membuat table. Sedangkan dibawahnya terdapat method try catch untuk sql yang akan menampilkan isi dari database dari table dokter.

```

159 JButton btnSimpan = new JButton("SIMPAN");
160 btnSimpan.addActionListener(new ActionListener()
161 {
162
163     @Override
164     public void actionPerformed(ActionEvent act){
165         // TODO Auto-generated method stub
166         Object obj = act.getSource();
167         if(obj == btnSimpan){
168             try{
169                 Class.forName("com.mysql.cj.jdbc.Driver");
170                 Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/rumah_sakit","root","");
171                 String sql = "insert into dokter values(?,?,?,?,?)";
172                 PreparedStatement prepare = con.prepareStatement(sql);
173
174                 prepare.setString(1, textField.getText());
175                 prepare.setString(2, textField_1.getText());
176                 prepare.setString(3, textField_2.getText());
177                 prepare.setString(4, textField_3.getText());
178                 prepare.setString(5, textField_4.getText());
179                 prepare.setString(6, textField_5.getText());
180                 prepare.executeUpdate();
181                 JOptionPane.showMessageDialog(null,"Data berhasil ditambahkan ke dalam database","Pesan",JOptionPane.INFORMATION_MESSAGE);
182                 tampilTabel();
183                 prepare.close();
184             }catch(Exception ex){
185                 ex.printStackTrace();
186                 JOptionPane.showMessageDialog(null,"Data gagal ditambahkan ke dalam database","Pesan",JOptionPane.ERROR_MESSAGE);
187             }
188         }
189     }
190
191 });
192 btnSimpan.setBounds(32, 415, 202, 25);
193 frame.getContentPane().add(btnSimpan);
194

```

Pada gambar diatas merupakan code suatu operasi dari button atas nama simpan, yaitu yang berisi action listener yang akan mengeksekusi perintah, saat user mengklik maka data akan tersimpan ke dalam database dengan menggunakan insert into dokter.

```

196 JButton btnhapus = new JButton("HAPUS");
197 btnhapus.addActionListener(new ActionListener(){
198
199     @Override
200     public void actionPerformed(ActionEvent act){
201         // TODO Auto-generated method stub
202         Object obj = act.getSource();
203         if(obj == btnhapus){
204             try{
205                 Class.forName("com.mysql.cj.jdbc.Driver");
206                 Connection con=DriverManager.getConnection(
207                     "jdbc:mysql://localhost:3306/rumah_sakit","root","");
208                 String sql = "delete from dokter where id_dokter = ?";
209                 PreparedStatement prepare = con.prepareStatement(sql);
210
211                 prepare.setString(1, textField.getText());
212                 prepare.executeUpdate();
213                 JOptionPane.showMessageDialog(null,"Data berhasil dihapus dari database","Pesan",JOptionPane.INFORMATION_MESSAGE);
214                 tampilTabel();
215                 prepare.close();
216             }catch(Exception ex){
217                 JOptionPane.showMessageDialog(null,"Data gagal dihapus dari database","Pesan",JOptionPane.ERROR_MESSAGE);
218             }
219         }
220     }
221
222 });
223 btnhapus.setBounds(246, 415, 202, 25);
224 frame.getContentPane().add(btnhapus);
225

```

Sedangkan gambar diatas adalah method operasi untuk menghapus database.

```

227 JButton btnUpdate = new JButton("UPDATE");
228 btnUpdate.addActionListener(new ActionListener() {
229
230     @Override
231     public void actionPerformed(ActionEvent act){
232         // TODO Auto-generated method stub
233         Object obj = act.getSource();
234         if(obj == btnUpdate){
235             try{
236                 Class.forName("com.mysql.cj.jdbc.Driver");
237                 Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/rumah_sakit","root","");
238                 String sql = "UPDATE dokter SET nama_dokter = '"+textField_1.getText()+"', nama_spesialisasi = '"+textField_2.getText()+"', gender = '"+textField_3.getText()+"'";
239                 PreparedStatement prepare = con.prepareStatement(sql);
240                 prepare.executeUpdate();
241                 JOptionPane.showMessageDialog(null,"Data berhasil diupdate","Pesan",JOptionPane.INFORMATION_MESSAGE);
242                 tampilTabel();
243                 prepare.close();
244             }catch(Exception ex){
245                 JOptionPane.showMessageDialog(null,"Data gagal diupdate","Pesan",JOptionPane.ERROR_MESSAGE);
246             }
247         }
248     }
249
250 });
251 btnUpdate.setBounds(459, 415, 202, 25);
252 frame.getContentPane().add(btnUpdate);
253

```

Gambar diatas berfungsi untuk mengupdate database, yaitu data dapat di update per bagian atau

perkolom sesuai primary keynya.

V. REFERENSI

PRACTO. (2015, February 6). *What is Hospital Management System?* Retrieved November 7, 2018, from doctors practo: <https://doctors.practo.com/hospital-management-system/>

Sappal, A. (2015). *E-R Diagram Hospital Management System (Entity Relationship Diagram)*. Retrieved November 7, 2018, from creately: <https://creately.com/diagram/example/in4w5ua22/E-R%20Diagram%20Hospital%20Management%20System>

VI. KELOMPOK

NIM	Nama/email	Role	Tanggungjawab	TTD
M0517019	Hakikid Amana	Ketua (Mapping)	Bertanggung jawab dalam mapping dari ERD , dan membuat view, serta melakukan quality control.	
M0517011	Bimo Adrian S.	Anggota (Survey)	Bertanggung jawab dalam mencari informasi/ survey	
M0517028	Maunab Galang E.	Anggota (Database)	Bertanggung jawab dalam membuat struktur database dan mengisi tuples pada database.	
M0517030	Mila Rosyida U.	Anggota (ERD)	Bertanggung jawab dalam membuat ERD dan membuat laporan	
M0517042	Rahmat Prasetyo N.	Anggota (Implementasi Aplikasi)	Bertanggung jawab membuat aplikasi java yang memiliki fungsi operasi Add, ead, Update, dan Delete, dan menghubungkan java ke database phpmyadmin.	