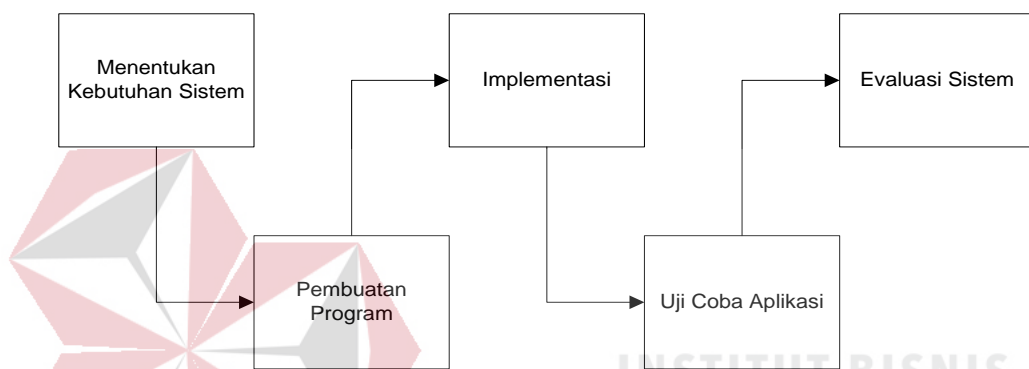


## BAB IV

### IMPLEMENTASI DAN EVALUASI

Pada bab ini akan menjelaskan tentang implementasi dan evaluasi dalam pengembangan aplikasi yang dibangun berbasis *mobile*. Adapun langkah-langkah pengerjaan yang dilakukan pada bab ini seperti pada gambar 4.1.



Gambar 4.1 Blok Diagram Implementasi dan Evaluasi

#### 4.1 Kebutuhan Sistem

Sistem ini membutuhkan media yaitu sebuah data pasien, data penyakit dan data kode titik bekam beserta lokasi titik bekam yang nantinya disimpan kedalam *smartphone*. Media *smartphone* nantinya untuk membantu terapis agar dapat mempermudah dalam penentuan lokasi titik bekam. Sistem nantinya akan mendeteksi berdasarkan histori pasien dengan data penyakit pada *smartphone*

Untuk dapat menjalankan sistem ini maka diperlukan perangkat keras untuk terapis, perangkat lunak untuk pengembangan dengan kondisi dan syarat tertentu agar sistem dapat berjalan dengan baik.

#### 4.1.1 Kebutuhan Perangkat Keras

Aplikasi pendukung penentuan titik bekam berbasis android dijalankan pada perangkat *mobile* Android. Spesifikasi minimum *handphone* yang dibutuhkan untuk menjalankan aplikasi ini adalah:

1. Minimum CPU *Single-core* 1 GHz *processor*.
2. Minimum *Internal Memory* 2GB.
3. Minimum RAM 512MB
4. Minimum Resolusi 1024x768 pixels

Spesifikasi rekomendasi *handphone* yang dibutuhkan untuk menjalankan aplikasi ini adalah:

1. Rekomendasi CPU *Single-core* 1 GHz *processor*.
2. Rekomendasi *Internal Memory* 2GB.
3. Rekomendasi RAM 1GB
4. Rekomendasi Resolusi 1024x768 *pixels*

#### 4.1.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan untuk mengembangkan Aplikasi *Mobile* paduan terapi bekam adalah:

- a. Kebutuhan Perangkat Lunak Pengguna

Perangkat lunak yang dibutuhkan untuk menjalankan aplikasi pendukung penentuan titik bekam pada *smartphone* sebenarnya tidak ada, karena sistem hanya membutuhkan *operating system* android 3.0. Semua yang dibutuhkan agar sistem ini berjalan menjadi satu dalam *operating system*.

b. Kebutuhan Perangkat Lunak Pengembangan

Kebutuhan perangkat lunak yang digunakan untuk mengembangkan aplikasi ini adalah sebagai berikut:

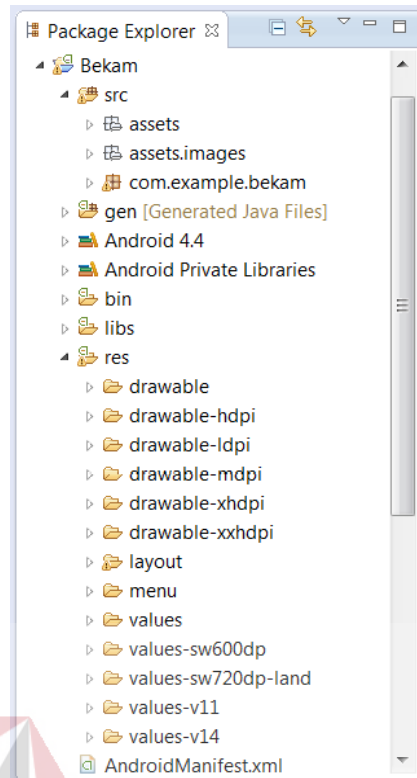
1. Sistem operasi Android 3.0.
2. Android SDK r11.
3. ADT Plugin *for Eclipse*.
4. Eclipse .
5. Sqlite.
6. *Bluestack* (Simulator Android) atau *handphone*.
7. Adobe Photoshop CS 4.

## 4.2 Pembuatan Program

Aplikasi *mobile* pendukung penentuan titik bekam dibangun menggunakan android SDK r11-linux dan *text editor* Eclips Helios yang didukung dengan ADT Plugin *for Eclipse*. Penulisan kode program pada *text editor* disimpan dalam file dengan ekstensi \*.java dan \*.xml. Android SDK akan meng-*compile file* berektensi \*.java dan \*.xml tersebut dan membuat package untuk aplikasi *mobile* pendukung penentuan titik bekam berekstensi \*.apk. Nantinya, file \*.apk tersebut yang akan digunakan untuk menjalankan aplikasi melalui *smartphone* android.

### 4.2.1 Struktur Bagian Android

Pada *project* android yang dibuat terdapat beberapa struktur atau bagian yang masing-masing mempunyai fungsi tersendiri. Berikut struktur bagian android dapat dilihat pada gambar 4.2.

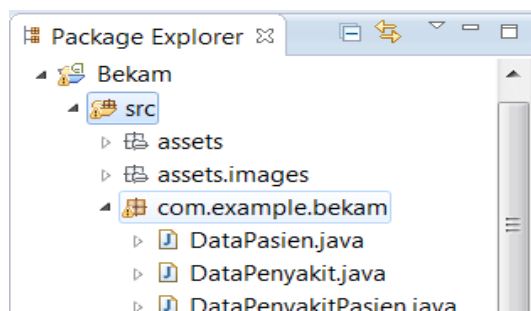


Gambar 4.2 Struktur Android

Pada *project* aplikasi pendukung penentuan titik bekam terdapat beberapa bagian struktur yaitu :

a. *Src*

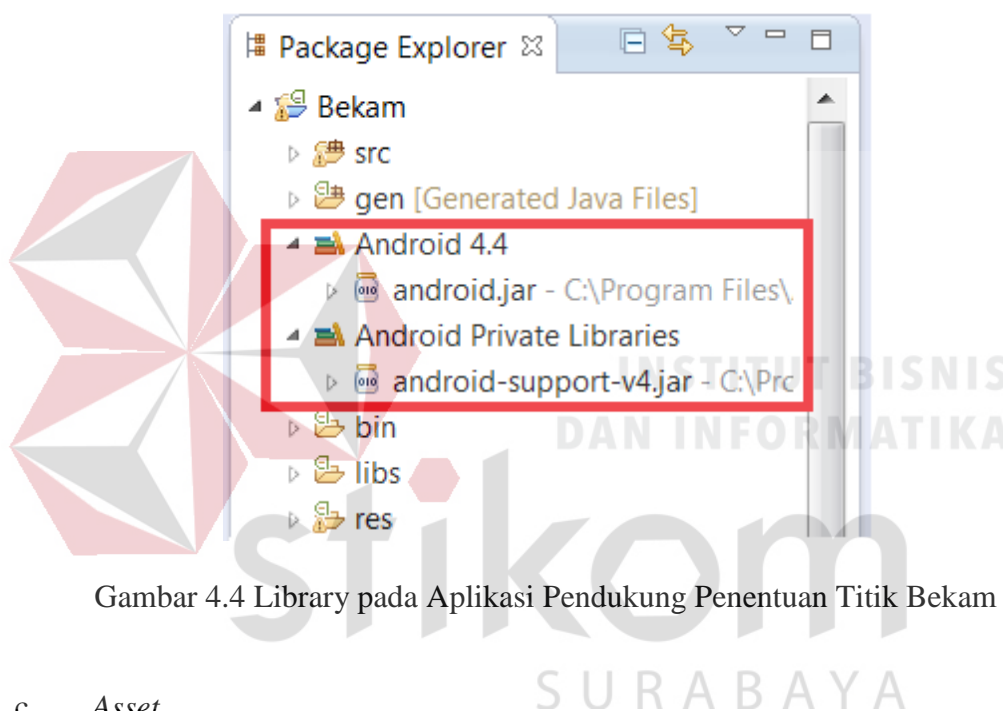
Pada *project* android terdapat *src* alias *source* pada bagian ini berfungsi sebagai folder utama yang berisi code java yang dibuat. Seperti terlihat pada gambar 4.3



Gambar 4.3 Main Activity Android

b. *Android 4.4, Android Dependences dan Referenced Libraries*

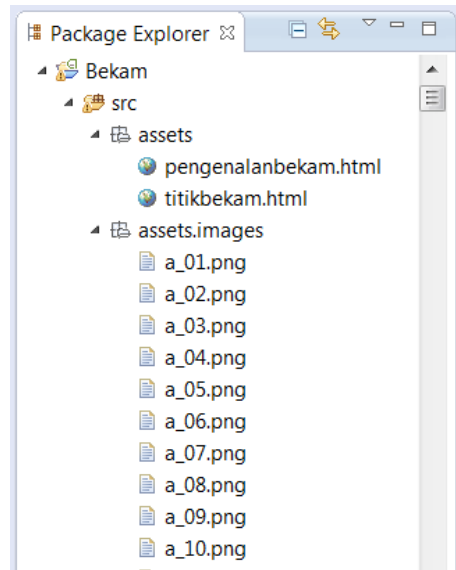
Pada *project* android terdapat *android 4.4* dan *Android Dependences* pada bagian ini berfungsi sebagai runtime untuk versi android target. Sedangkan *referenced libraries* dibuat dengan bahasa java, sebagian besar library java bisa dipakai di android. Syarat utamanya, library harus dibuat murni dengan java. Seperti terlihat pada gambar 4.4.



Gambar 4.4 Library pada Aplikasi Pendukung Penentuan Titik Bekam

c. *Asset*

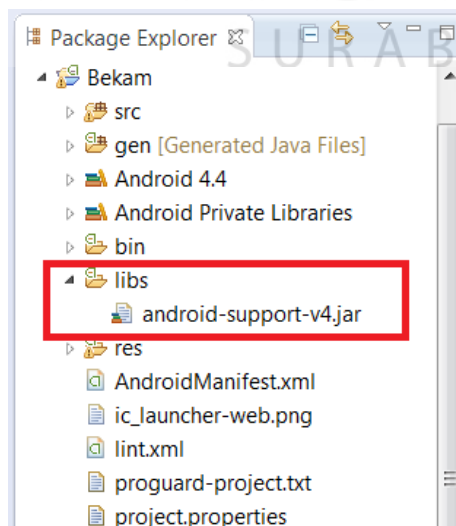
Pada aplikasi pendukung penentuan titik bekam terdapat bagian asset. Asset adalah folder yang bisa dimanfaatkan oleh programmer untuk menyimpan file audio, video, file html, gambar dan file bantu lainnya. Pada pembuatan aplikasi pendukung penentuan titik bekam, asset digunakan untuk menyimpan gambar-gambar yang dapat menunjang berjalannya aplikasi pendukung penentuan titik bekam. Berikut isi dari asset pada gambar 4.5.



Gambar 4.5 Isi Struktur Android pada Bagian Asset

d. *Libs*

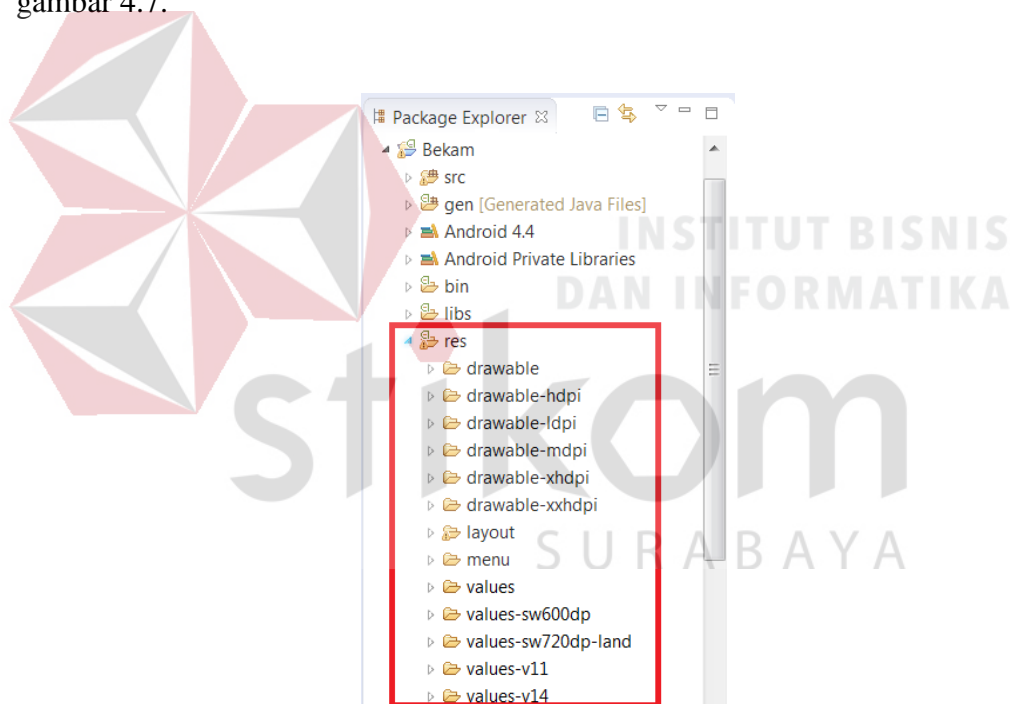
*Libs* merupakan *library* resmi dari Google atau android. *Library* yang digunakan disini adalah `android-support-v4.jar` yang dipakai untuk membuat tampilan interface ICS di android dengan versi yang lebih rendah seperti Froyo atau Gingerbread, berikut dapat terlihat pada gambar 4.6.



Gambar 4.6 Isi Struktur pada Bagian Libs

e. *Res*

Res atau resource adalah folder yang digunakan untuk menampung semua resource yang ada di program. Resource disini bisa berupa gambar, layout, menu dan values. Gambar akan diletakan di folder `drawable-*dpi`. File layout XML dari tampilan android diletakan di folder `layout`, menu di `actionbar` diletakan di folder `menu` dan folder `value` menampung `string.xml`, `dimens.xml` dan `style.xml`. Semua file ini nantinya dipanggil oleh kode java. Kesalahan dalam menulis file XML di `res` akan membuat source code menjadi error. Berikut dapat dilihat pada gambar 4.7.

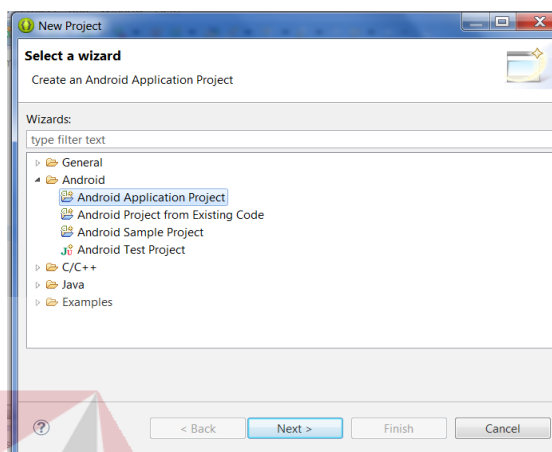


Gambar 4.7 Fitur Layout pada Android

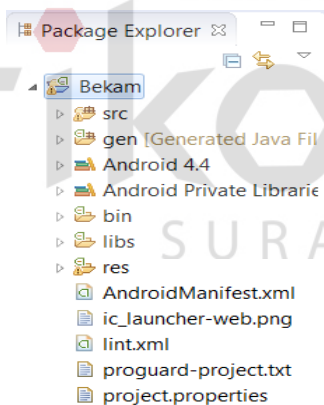
#### 4.2.2 Membuat Project Android

Membuat aplikasi pendukung penentuan titik bekam berbasis android dibuat pada *software* eclipse. Cara membuat project pada eclipse adalah *File – New – Project* kemudian pilih “Android Application Project” seperti pada gambar 4.8.

Setelah itu isi nama project beserta minimal requirement .sdk hal ini untuk menentukan minimal aplikasi dijalankan pada *operating system* android versi berapa. Kemudian menentukan bentuk *icon*, nama *activity* dan *activity main*. Berikut hasil pembuatan *project* android dapat dilihat pada gambar 4.9.



Gambar 4.8 Pembuatan Project Android



Gambar 4.9 Hasil Pembuat Project Android Baru

### 4.2.3 Menu Utama

Menu utama merupakan tampilan pembuka ketika terapis mulai memasuki tampilan aplikasi. Tampilan menu utama dapat dilihat pada gambar 4.10. Pada menu ini terdapat 5 macam pilihan menu yaitu menu pengenalan



bekam, menu pengenalan titik bekam, menu pengobatan bekam, menu data penyakit dan menu keluar apabila terapis ingin mengakhiri sesi pendukung penentuan titik bekam untuk keluar dari aplikasi.



Gambar 4.10 Menu Utama

Pada gambar 4.11 menjelaskan tentang proses menentukan menu button sesuai dengan ke 5 menu yang terdapat pada menu utama yaitu menu pengenalan bekam, menu pengenalan titik bekam, menu pengobatan bekam, menu data penyakit dan menu keluar apabila terapis ingin mengakhiri sesi pendukung penentuan titik bekam untuk keluar dari aplikasi.

```

*MenuUtama.java
String filename;
Context context;
MediaPlayer mp;

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_menu_utama);

    ImageButton btn1 = (ImageButton) findViewById(R.id.btn1);
    btn1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            // TODO Auto-generated method stub
            finish();
            //startActivity(new Intent(Menu.this,SettingPref.class));
            Intent myIntent = new Intent(view.getContext(),PengenalanBekam.class);
            startActivityForResult(myIntent, 0);
        }
    });

    ImageButton btn2 = (ImageButton) findViewById(R.id.btn2);
    btn2.setOnClickListener(new View.OnClickListener() {
        @Override
    
```

Gambar 4.11 Proses pada Menu Utama

#### 4.2.4 Menu Pengenalan Bekam

Menu pengenalan bekam merupakan menu yang menjelaskan tentang bekam untuk memberikan informasi tentang definisi bekam, sejarah bekam, macam-macam cara pengobatan nabawi, jenis-jenis bekam, prosedur melakukan pembekaman, beberapa hal yang harus diperhatikan dalam proses pembekaman, cara membekam, pengenalan tempat/titik bekam, titik terlarang untuk dibekam.

Hal tersebut di tujukan untuk mengenalkan tentang bekam dan mencegah tingkat resiko dalam melakukan pembekaman kepada pasien agar tidak terjadi kesalahan dalam proses pembekaman dan mencegah penyakit tersebut menular kepada terapis dan pasien yang lain. Tampilan menu persiapan bekam dapat dilihat pada Gambar 4.12.

*Source code* dibawah ini menjelaskan tentang pembuatan menu pengenalan bekam yang menggunakan format .html. Untuk lebih jelasnya dapat dilihat *Source code* pembuatannya pada gambar 4.13.



Gambar 4.12 Menu Pengenalan Bekam

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title></title>
</head>

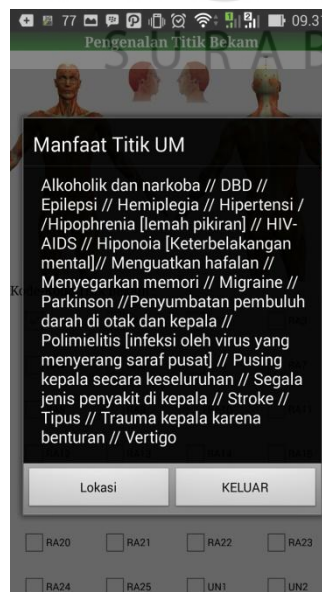
<body>
<p align="justify">&nbsp;</p>
<p align="justify">&nbsp;</p>
<p align="justify">PROSEDUR MELAKUKAN PEMBEKAMAN</p>
<p align="justify">&nbsp;</p>
<p align="justify">1. PERSIAPAN
<p align="justify">A. MENYIAPKAN ALAT, SARANA DAN RUANGAN
<p align="justify">1) Alat yang dipersiapkan: set kop/tabung penghisap, skapel, jarum, lancet pen, pisa
duk kain, sarung tangan, masker, mangkok/cawan, tempat sampah, meja dan kursi.</p>
<p align="justify">2) Bahan yang disiapkan: kassa, kapas/tissue, betadin, detol, sabun, zalf, alkohol,
minyak zaitun, minyak habbatussauda, al qusthul hindi, minyak urut hangat (misal
gandapura), minuman hangat, baik kalau disediakan madu dan susu.</p>
<p align="justify">3) Mensterilkan alat agar bebas kuman dan tidak menyebarkan penyakit, dengan cara:
tabung kop paling sedikit selama 30 menit setelah air mendidih terus menerus (karet dilepas
dulu). Sarung tangan, karet dan duk kain disterilkan dengan tablet formalin.</p>

```

Gambar 4.13 *Source Code* pada Menu Pengenalan Bekam

#### 4.2.5 Menu Pengenalan Titik Bekam

Menu pengenalan titik bekam merupakan menu yang menjelaskan tentang lokasi titik bekam berdasarkan kode titik bekam yang dipilih dan menjelaskan fungsi dari masing-masing titik bekam. Apabila gambar pada anatomi tubuh kurang jelas dapat dilakukan *zoom in* dan *zoom out* agar dapat dilihat lebih jelas. Tampilan menu pengenalan titik bekam dapat dilihat pada gambar 4.14.



Gambar 4.14 Menu Pengenalan Titik Bekam

Pada gambar 4.15 merupakan *source* yang digunakan untuk memanggil *function* gambar yang terdapat pada *titikbekam.html* dan untuk menampilkan lokasi titik bekam kemudian.

```

*PengenalanTitikBekam.java  titikbekam.html
public void checkRA20(View v) {
    if (((CheckBox) v).isChecked()) { myBrowser.loadUrl("javascript:ra20_a()");}
    else{ myBrowser.loadUrl("javascript:ra20_b()");}
}
public void checkRA21(View v) {
    if (((CheckBox) v).isChecked()) { myBrowser.loadUrl("javascript:ra21_a()");}
    else{ myBrowser.loadUrl("javascript:ra21_b()");}
}
public void checkRA22(View v) {
    if (((CheckBox) v).isChecked()) { myBrowser.loadUrl("javascript:ra22_a()");}
    else{ myBrowser.loadUrl("javascript:ra22_b()");}
}
public void checkRA23(View v) {
    if (((CheckBox) v).isChecked()) { myBrowser.loadUrl("javascript:ra23_a()");}
    else{ myBrowser.loadUrl("javascript:ra23_b()");}
}
public void checkRA24(View v) {
    if (((CheckBox) v).isChecked()) { myBrowser.loadUrl("javascript:ra24_a()");}
    else{ myBrowser.loadUrl("javascript:ra24_b()");}
}
public void checkRA25(View v) {
    if (((CheckBox) v).isChecked()) { myBrowser.loadUrl("javascript:ra25_a()");}
}

```

Gambar 4.15 *Source Code* Untuk Memanggil *Function* *titikbekam.html*

Sedangkan pada gambar 4.16 merupakan *source code* untuk melakukan *zoom in* dan *zoom out* pada gambar anatomi yang terdapat pada aplikasi pendukung penentuan titik bekam.

```

*PengenalanTitikBekam.java
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_pengenalan_titik_bekam);
    myBrowser = (WebView) findViewById(R.id.mybrowser);

    final MyJavaScriptInterface myJavaScriptInterface
        = new MyJavaScriptInterface(this);
    myBrowser.addJavaScriptInterface(myJavaScriptInterface, "AndroidFunction");

    myBrowser.getSettings().setBuiltInZoomControls(true);
    myBrowser.getSettings().setDisplayZoomControls(false);

    myBrowser.getSettings().setLoadWithOverviewMode(true);
    myBrowser.getSettings().setUseWideViewPort(true);

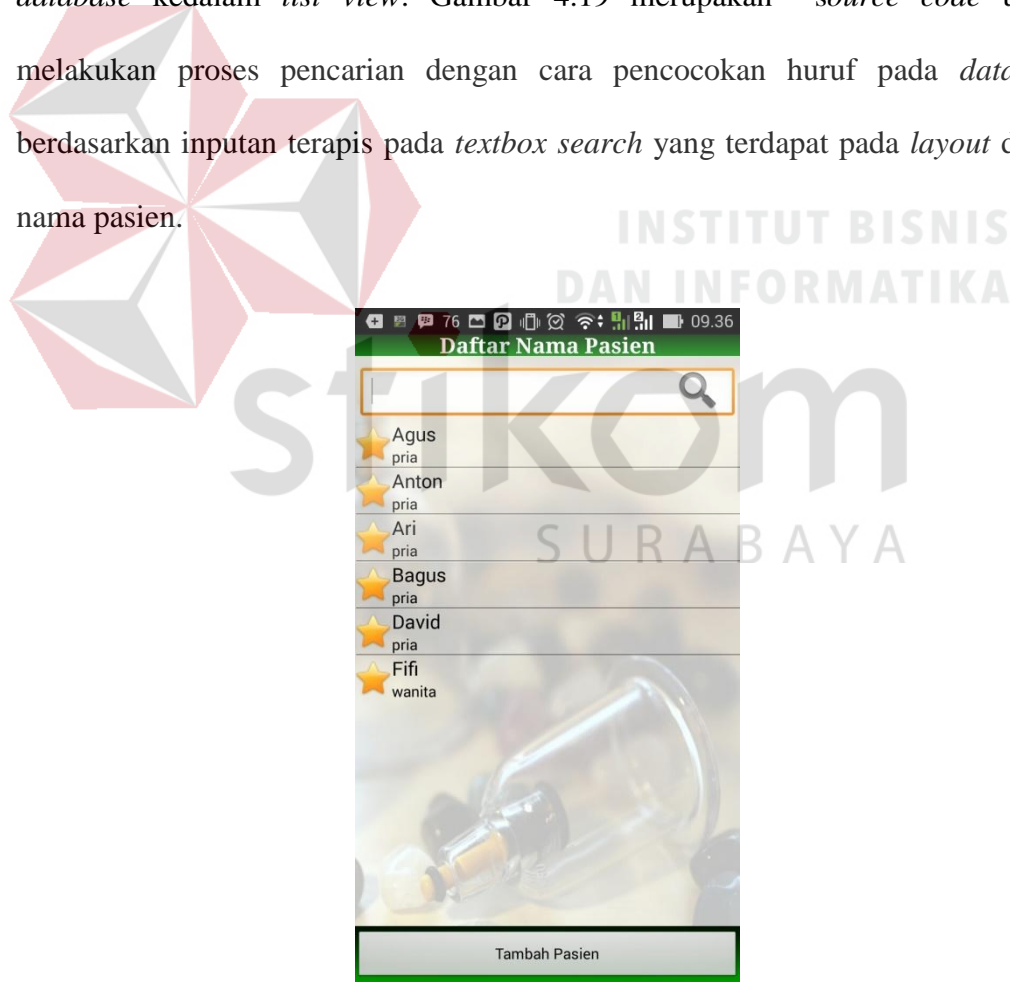
    myBrowser.getSettings().setJavaScriptEnabled(true);
    myBrowser.loadUrl("file:///android_asset/titikbekam.html");
}

```

Gambar 4.16 *Source Code* Untuk Melakukan *Zoom In* dan *Zoom Out*.

#### 4.2.6 Menu Pengobatan Bekam

Menu pengobatan bekam merupakan menu yang menjelaskan tentang lokasi mana saja yang akan dibekam. Terapis harus mencari nama pasien pada daftar nama pasien terlebih dahulu, tetapi jika nama pasien yang akan dibekam belum terdaftar maka terapis dapat mendaftarkan nama pasien terlebih dahulu seperti pada gambar 4.17 agar sistem dapat memproses ke tahap berikutnya. *Source code* menampilkan daftar nama pasien pada *list view* dapat dilihat pada gambar 4.18, dimana *source code* tersebut menampilkan daftar nama pasien dari *database* kedalam *list view*. Gambar 4.19 merupakan *source code* untuk melakukan proses pencarian dengan cara pencocokan huruf pada *database* berdasarkan inputan terapis pada *textbox search* yang terdapat pada *layout* daftar nama pasien.



Gambar 4.17 Form Daftar Nama Pasien

```

*ListPasien.java
Log.v("t", "1");
ix=0;
datap pasien = new DataPasien(this);
dbpasien = datap pasien.getWritableDatabase();
pasienCursor = dbpasien.rawQuery("SELECT id_pasien, nama_pasien, jenis_kelamin, tanggal_lahir, tanggal_terdaftar FROM pasien or
if (pasienCursor.moveToFirst()) {
    for (; !pasienCursor.isAfterLast(); pasienCursor.moveToNext()) {
        Log.v("t", "x");
        ix++;
        Log.v("t", pasienCursor.getString(0).toString());
        Log.v("t", pasienCursor.getString(1).toString());
        Log.v("t", pasienCursor.getString(2).toString());
        Log.v("t", pasienCursor.getString(3).toString());
        Log.v("t", pasienCursor.getString(4).toString());
        String ip = pasienCursor.getString(0).toString();
        String np = pasienCursor.getString(1).toString();
        String jk = pasienCursor.getString(2).toString();
        String tl = pasienCursor.getString(3).toString();
        String tt = pasienCursor.getString(4).toString();
        Log.v("t", "y");
        ListPasien.add(new Pasien(ip,np,jk,tl,tt));
    }
}
Log.v("t", "1");

if (!ListPasien.isEmpty()){
    PasienAdapter adapter = new PasienAdapter(ListPasien.this, ListPasien);
    list.setAdapter(adapter);
}

```

Gambar 4.18 *Source Code* Untuk Menampilkan Daftar Nama Pasien

```

*ListPasien.java
protected void search(String text) {
    // TODO Auto-generated method stub
    ListPasien.clear();
    //if (text.length()>0) {
        String query = "SELECT id_pasien, nama_pasien, jenis_kelamin, tanggal_lahir, tanggal_terdaftar FROM pasien where nama_pasie
        pasienCursor = dbpasien.rawQuery(query, null);
        if (pasienCursor.moveToFirst()) {
            for (; !pasienCursor.isAfterLast(); pasienCursor.moveToNext()) {
                Log.v("t", "x");
                ix++;
                String ip = pasienCursor.getString(0).toString();
                String np = pasienCursor.getString(1).toString();
                String jk = pasienCursor.getString(2).toString();
                String tl = pasienCursor.getString(3).toString();
                String tt = pasienCursor.getString(4).toString();
                Log.v("t", "y");
                ListPasien.add(new Pasien(ip,np,jk,tl,tt));
            }
        }
    }
}
PasienAdapter adapter = new PasienAdapter(ListPasien.this, ListPasien);

```

Gambar 4.19 *Source Code* Untuk Melakukan Proses Pencarian Nama Pasien

Pada form ini terapis dapat melakukankkan perubahan data diri pasien yang berfungsi bila terjadi kesalahan pada pengisian data diri pasien. Pada form *edit* pasien data yang dapat dirubah yaitu nama pasien, jenis kelamin pasien dan tanggal lahir pasien, yang kemudian dapat menekan *button* simpan untuk menyimpan kedalam *database* atau terapis dapat juga memilih *button* hapus untuk menghapus data pasien dari *database*, seperti terlihat dapa gambar 4.20.

Gambar 4.20 Form Edit Data Pasien

Pada gambar 4.21 dan gambar 4.22 menjelaskan *source code* menyimpan hasil *edit* data pasien dan menghapus data pasien dari *database* pada aplikasi. Dimana nantinya hasil *edit* data pasien digunakan oleh terapis melakukan perubahan pada data pasien sedangkan menghapus data pasien digunakan bila terapis merasa tidak lagi membutuhkan data tersebut.

```

*EditPasien.java
@Override
public void onClick(View view) {
    // TODO Auto-generated method stub
    if (view == findViewById(R.id.btnSimpan)){

        String jk = "";
        radioJK = (RadioGroup) findViewById(R.id.radioJK);
        int selectedId2 = radioJK.getCheckedRadioButtonId();
        if(selectedId2>0){
            radioJKButton = (RadioButton) findViewById(selectedId2);
            jk = radioJKButton.getText().toString();
        }
        //tanggal sekarang
        Calendar now = Calendar.getInstance();
        Date myDate = now.getTime();
        String myFormat = "yyyy-MM-dd";
        SimpleDateFormat sdf = new SimpleDateFormat(myFormat, Locale.US);
        String tglskrg = sdf.format(myDate);
        //tanggal dr datepicker
        Date tgl1lh = new Date(tanggalLahir.getYear() - 1900, tanggalLahir.getMonth(), tanggalLahir.getDayOfMonth());
        String myFormat2 = "yyyy-MM-dd";
        SimpleDateFormat sdf2 = new SimpleDateFormat(myFormat2, Locale.US);
        String tgl_lhr = sdf2.format(tgl1lh);

        String nm = txtNama.getText().toString();
        String tl = tgl_lhr;
        String tt = tglskrg;
    }
}

```

Gambar 4.21 *Source Code* Menyimpan Hasil Edit Data Pasien



```

else if (view == findViewById(R.id.btnHapus)){
    AlertDialog.Builder myDialog = new AlertDialog.Builder(EditPasien.this);
    myDialog.setTitle("Hapus Data");

    final TextView edit_tahun = new TextView(EditPasien.this);
    LayoutParams dialogC1_idLayoutParams = new LayoutParams(LayoutParams.MATCH_PARENT, LayoutParams.WRAP_CONTENT);
    edit_tahun.setLayoutParams(dialogC1_idLayoutParams);
    edit_tahun.setText("Apakah anda yakin ingin menghapus data ini?");

    LinearLayout layout = new LinearLayout(EditPasien.this);
    layout.setOrientation(LinearLayout.VERTICAL);
    layout.addView(edit_tahun);
    myDialog.setView(layout);

    myDialog.setPositiveButton("Proses", new DialogInterface.OnClickListener() {
        // do something when the button is clicked
        public void onClick(DialogInterface arg0, int arg1) {

            datapasien = new DataPasien(EditPasien.this);
            dbpasien = datapasien.getWritableDatabase();
            datapasien.delete_byID(dbpasien, id_pasien);
        }
    });
}

```

Gambar 4.22 Source Code Untuk Menghapus Data Pasien dari Database

Pada form ini terapis dapat menentukan langkah pengobatan yang akan dilakukan terhadap pasien, seperti pada *radio button* “Detail Data Pasien” berfungsi untuk mengetahui penyakit apa saja yang pernah ditangani dan pada tahap apa penanganan tersebut berjalan, bila pasien belum pernah melakukan pembekaman menggunakan aplikasi pendukung penentuan titik bekam terhadap penyakit apapun maka pada tahap tersebut sistem tidak dapat memproses permintaan tersebut. Maka terapis harus memilih *radio button* “Nama Penyakit Lain”. Pada *radio button* “Nama Penyakit Lain” berisi tentang daftar nama penyakit yang akan ditangani oleh terapis seperti pada gambar 4.23.

Pada gambar 4.24 menjelaskan *source code* penghitungan usia pasien pada aplikasi, dimana nantinya usia pasien digunakan oleh terapis untuk mengukur tekanan udara pada *cup* bekam.



Gambar 4.23 Form Detail Data Pasien

```

MemilihPenyakit.java  activity_memilih_penyakit.xml
Log.v("tgl", "2");
int month2 = myCal2.get(Calendar.MONTH);
int day2 = myCal2.get(Calendar.DATE);
int year2 = myCal2.get(Calendar.YEAR);

Calendar now= Calendar.getInstance();
Calendar tanggalahirr = Calendar.getInstance();
Log.v("tgl", "3");
tanggalahirr.set(year2, month2, day2);

int years = now.get(Calendar.YEAR) - tanggalahirr.get(Calendar.YEAR);
int months = now.get(Calendar.MONTH) - tanggalahirr.get(Calendar.MONTH);
int days = now.get(Calendar.DAY_OF_MONTH) - tanggalahirr.get(Calendar.DAY_OF_MONTH);
if (days < 0){
    months --;
    days +=now.getActualMaximum(Calendar.DAY_OF_MONTH);
}
if (months < 0){
    years --;
    months +=12;
}
Log.v("tgl", "4");
txtUmur.setText(" : "+years+" tahun ");
Log.v("tgl", "5");

```

Gambar 4.24 Source Code Menghitung Usia Pasien

*Source code* menampilkan data penyakit yang pernah ditangani pada *combobox* histori penyakit pasien dapat dilihat pada gambar 4.25, dimana *source code* tersebut menampilkan data berupa nama penyakit yang pernah ditangani dari *database* kedalam *combobox* histori penyakit pasien.

```

*MemilihPenyakit.java activity_memilih_penyakit.xml
ix=0;
datapenyakit = new DataPenyakit(this);
dbpenyakit = datapenyakit.getWritableDatabase();
if(jenis_kelamin.equalsIgnoreCase("pria")){
    penyakitCursor = dbpenyakit.rawQuery("SELECT penyakit.id_penyakit, nama_penyakit, pria, wanita, penyakit.status FROM penyakit,peny
}else{
    penyakitCursor = dbpenyakit.rawQuery("SELECT penyakit.id_penyakit, nama_penyakit, pria, wanita, penyakit.status FROM penyakit,peny
}
if (penyakitCursor.moveToFirst()) {
    for (; !penyakitCursor.isAfterLast(); penyakitCursor.moveToNext()) {
        ix++;
        String ip = penyakitCursor.getString(0).toString();
        String np = penyakitCursor.getString(1).toString();
        String p = penyakitCursor.getString(2).toString();
        String w = penyakitCursor.getString(3).toString();
        String s = penyakitCursor.getString(4).toString();
        ListPenyakit1a.add(ip);
        ListPenyakit1b.add(np);
    }
    ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item,ListPenyakit1b);
    dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spinner1.setAdapter(dataAdapter);
}

```

Gambar 4.25 Source Code Menampilkan Histori Penyakit Pasien

Source code menampilkan data nama penyakit pada *combobox* daftar nama penyakit dapat dilihat pada gambar 4.26, dimana *source code* tersebut menampilkan data berupa seluruh nama penyakit yang dimiliki oleh aplikasi dan data tersebut diambil dari *database* kedalam *combobox* daftar nama penyakit.

```

*MemilihPenyakit.java activity_memilih_penyakit.xml
ix=0;
datapenyakit = new DataPenyakit(this);
dbpenyakit = datapenyakit.getWritableDatabase();
if(jenis_kelamin.equalsIgnoreCase("pria")){
    penyakitCursor = dbpenyakit.rawQuery("SELECT id_penyakit, nama_penyakit, pria, wanita, status FROM penyakit where pria='ya' and per
}else{
    penyakitCursor = dbpenyakit.rawQuery("SELECT id_penyakit, nama_penyakit, pria, wanita, status FROM penyakit where wanita='ya' and p
}
if (penyakitCursor.moveToFirst()) {
    for (; !penyakitCursor.isAfterLast(); penyakitCursor.moveToNext()) {
        ix++;

        String ip = penyakitCursor.getString(0).toString();
        String np = penyakitCursor.getString(1).toString();
        String p = penyakitCursor.getString(2).toString();
        String w = penyakitCursor.getString(3).toString();
        String s = penyakitCursor.getString(4).toString();

        ListPenyakit2a.add(ip);
        ListPenyakit2b.add(np);
    }
}

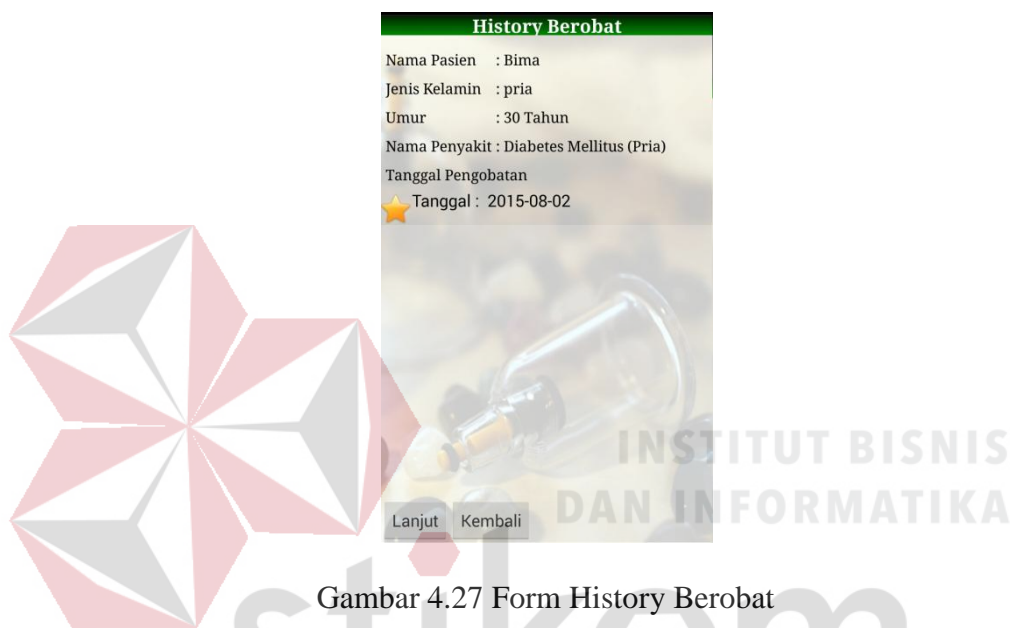
ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item,ListPenyakit2b);
dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinner2.setAdapter(dataAdapter);

```

Gambar 4.26 Source Code Menampilkan Daftar Nama Penyakit

Pada gambar 4.27 terapis dapat mengetahui riwayat pengobatan pada pasien tersebut. Pada form history pengobatan sistem dapat menampilkan data berupa

nama pasien, jenis kelamin, umur, nama penyakit dan tanggal pengobatan. Akan tetapi data tanggal pengobatan muncul bila sebelumnya pasien pernah melakukan terapi bekam dengan penyakit yang sama. Bila terapis melakukan kesalahan dalam memilih penyakit, terapis dapat kembali pada form sebelumnya dengan cara menekan *button* kembali, seperti terlihat pada gambar 4.27.



Gambar 4.27 Form History Berobat

*Source code* menampilkan data history pasien dapat dilihat pada gambar 4.28 dimana *source code* tersebut menampilkan data berupa seluruh nama pasien, jenis kelamin, umur, nama penyakit dan daftar tanggal terapi.

```

datap pasien = new DataPasien(this);
dbpasien = datap pasien.getWritableDatabase();
pasienCursor = dbpasien.rawQuery("SELECT id_pasien,nama_pasien,jenis_kelamin,tanggal_lahir,tanggal_tenda
if (pasienCursor.moveToFirst()) {
    Log.v("", "3");
    nama_pasien=pasienCursor.getString(1).toString();
    jenis_kelamin=pasienCursor.getString(2).toString();
    tanggal_lahir=pasienCursor.getString(3).toString();
    //tanggal_tenda=pasienCursor.getString(4).toString();
    Log.v("", "4");

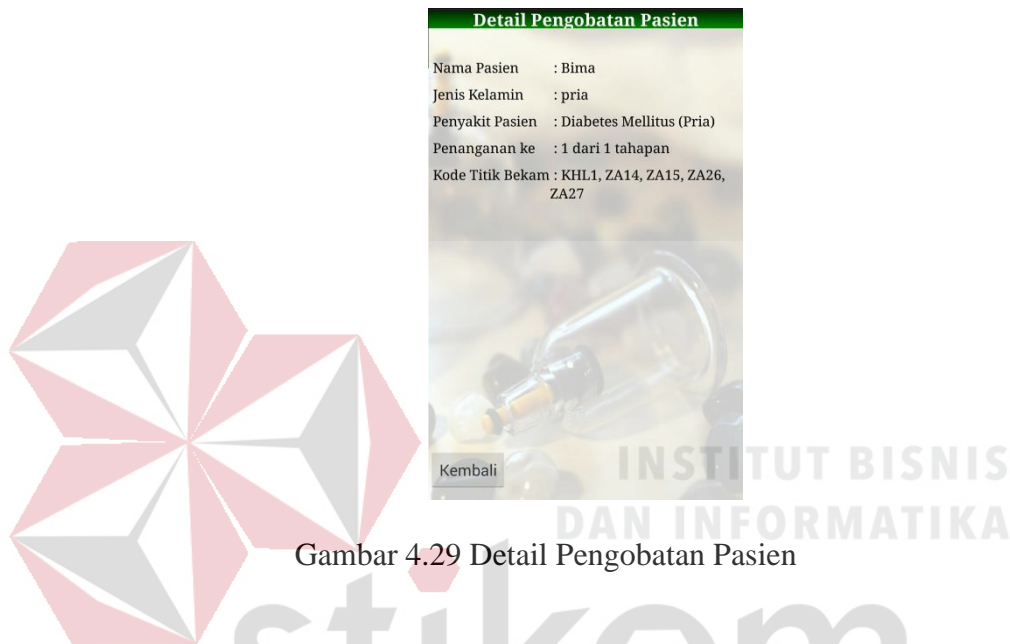
    txtNamaPasien.setText(" : "+nama_pasien);
    txtJenisKelamin.setText(" : "+jenis_kelamin);

    Log.v("tgl", "1");
    String myFormat2 = "yyyy-MM-dd";
    SimpleDateFormat sdf2 = new SimpleDateFormat(myFormat2, Locale.US);
    Date dateObj2;
    Calendar myCal2 = Calendar.getInstance();
    try {
        dateObj2 = sdf2.parse(tanggal_lahir);
    }

```

Gambar 4.28 *Source Code* Menampilkan Data Pasien pada Histori Berobat

Dari gambar 4.29 terapis dapat mengetahui data pasien tersebut dan dapat memantau perkembangannya karena terdapat pula proses penanganan keberapa dan kode titik bekam yang diambil pada pertemuan sebelumnya, seperti terlihat pada gambar 4.29.



Gambar 4.29 Detail Pengobatan Pasien

```

datatahapanpenyakit = new DataTahapanPenyakit(this);
dbtahapapanpenyakit = datatahapanpenyakit.getWritableDatabase();
tahapapanpenyakitCursor = dbtahapapanpenyakit.rawQuery("SELECT id_tahapan_penyakit,id_penyakit,tahapan_ke,1
if (tahapapanpenyakitCursor.moveToFirst() {
    tahapan_maks=tahapanpenyakitCursor.getString(2).toString();
}
}else{
    tahapan_maks="0";
}

txtPenangananKe.setText(" : "+tahapapan_ke+" dari "+tahapapan_maks+" tahapan ");

datap pasien = new DataPasien(this);
dbpasien = datap pasien.getWritableDatabase();
pasienCursor = dbpasien.rawQuery("SELECT id_pasien,nama_pasien,jenis_kelamin,tanggal_lahir,tanggal_ter
if (pasienCursor.moveToFirst() {
    Log.v("", "3.1");
    nama_pasien=pasienCursor.getString(1).toString();
    jenis_kelamin=pasienCursor.getString(2).toString();
    tanggal_lahir=pasienCursor.getString(3).toString();

    datatitikpenyakit = new DataTitikPenyakit(this);
    dbtitikpenyakit = datatitikpenyakit.getWritableDatabase();
    titikpenyakitCursor = dbtitikpenyakit.rawQuery("SELECT * FROM titik_penyakit where id_tahapan_penyakit
    if (titikpenyakitCursor.moveToFirst() {
        while(titikpenyakitCursor.isAfterLast() == false){
            if(namaTitik.equals("")) == false){
                namaTitik = namaTitik + ", ";
            }
            namaTitik = namaTitik + titikpenyakitCursor.getString(2);
            titikpenyakitCursor.moveToNext();
        }
        Log.e("", "4.2 titik");
    }
}
Log.e("abc","n "+namaTitik);
Log.e("aaaa","bbb");
txtKodeBekam.setText(" : "+namaTitik);

```

Gambar 4.30 *Source Code* Form Detail Pengobatan Pasien

*Source code* form detail pengobatan pasien dapat dilihat pada gambar 4.30 dimana *source code* tersebut menampilkan data berupa seluruh nama pasien, jenis kelamin, nama penyakit, penanganan ke, dan kode titik bekam.

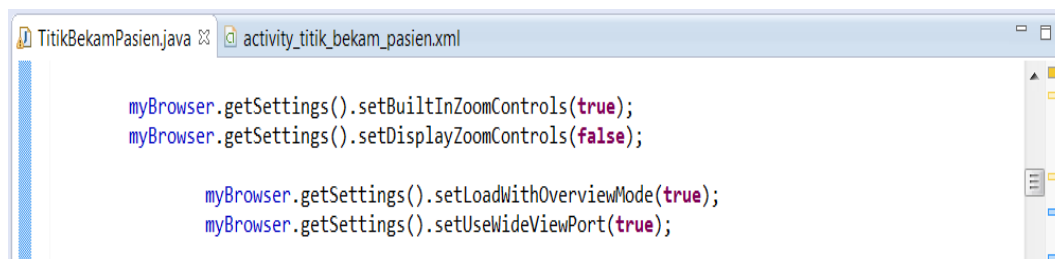
Menu titik bekam pasien merupakan menu yang menjelaskan tentang penentuan lokasi titik-titik yang akan dibekam berdasarkan nama penyakit dan tahapan penanganan. Gambar anatomi pada form ini dapat melakukan *zoom in* dan *zoom out*, seperti terlihat pada gambar 4.31.



Gambar 4.31 Form Titik Bekam Pasien

*Source code zoom in* dan *zoom out* pada gambar anatomi dapat dilihat pada gambar 4.32, dimana *source code* tersebut dapat melakukan *zoom in* dan *zoom out* untuk memperjelas gambar titik bekam yang ada pada aplikasi dan apa bila terapis menekan button lanjut sistem akan melakukan proses penambahan

jumlah penanganan yang disimpan kedalam *database* untuk mendukung pengobatann bekam pada sesi berikutnya, seperti pada gambar 4.33.



```

TitikBekamPasien.java activity_titik_bekam_pasien.xml
myBrowser.getSettings().setBuiltInZoomControls(true);
myBrowser.getSettings().setDisplayZoomControls(false);

myBrowser.getSettings().setLoadWithOverviewMode(true);
myBrowser.getSettings().setUseWideViewPort(true);

```

Gambar 4.32 *Source Code* Zoom In/Zoom Out



```

TitikBekamPasien.java activity_titik_bekam_pasien.xml
public void onClick(View view) {
    // TODO Auto-generated method stub
    if (view == findViewById(R.id.btnLanjut)){
        Log.v("insert", "1");
        if(tahapan_aktif==1){
            datapenyakitpasien = new DataPenyakitPasien(this);
            dbpenyakitpasien = datapenyakitpasien.getWritableDatabase();
            penyakitpasienCursor = dbpenyakitpasien.rawQuery("SELECT id_penyakit_pasien,id_pasien,id_penyakit,tahapan,status FROM peny
            if (penyakitpasienCursor.moveToFirst()) {
                id_penyakit_pasien=Integer.parseInt(penyakitpasienCursor.getString(0).toString()+1)+"";
            }
            if(id_penyakit_pasien.equalsIgnoreCase("")){
                id_penyakit_pasien="1";
            }
            Log.v("insert", "2");
            datapenyakitpasien = new DataPenyakitPasien(this);
            dbpenyakitpasien = datapenyakitpasien.getWritableDatabase();
            datapenyakitpasien.generateData(dbpenyakitpasien,id_penyakit_pasien,id_pasien,id_penyakit,tahapan_aktif+"","on");
        }else{
            datapenyakitpasien = new DataPenyakitPasien(this);
            dbpenyakitpasien = datapenyakitpasien.getWritableDatabase();
            datapenyakitpasien.updateId(dbpenyakitpasien,id_penyakit_pasien,tahapan_aktif+"");
        }
    }
}

```

Gambar 4.33 *Source Code* Penambahan Jumlah Penanganan

#### 4.2.7 Tampilan Menu Data Penyakit

Pada tampilan tersebut terdapat fitur *serch* yang ditujukan untuk mempermudah terapis dalam mencari nama penyakit. Kemudian terdapat daftar nama-nama penyakit yang dapat dipilih untuk melakukan *view detail data*, *edit data* dan *delete data*. Selain itu pada form ini terdapat *button* untuk menambahkan data nama penyakit sesuai dengan kebutuhan terapis, seperti pada gambar 4.34.



Gambar 4.34 Menu Data Penyakit

Gambar 4.35 merupakan *source code* menampilkan daftar nama penyakit pada *list view*. Dimana *source code* tersebut menampilkan daftar nama penyakit dari *database* kedalam *list view*.

```

ListDataPenyakit.java
ListPasien.java

ix=0;
datapenyakit = new DataPenyakit(this);
dbpenyakit = datapenyakit.getWritableDatabase();
penyakitCursor = dbpenyakit.rawQuery("SELECT id_penyakit, nama_penyakit, pria, wanita, status FROM penyakit order by id_penyakit");
if (penyakitCursor.moveToFirst()) {
    for (; !penyakitCursor.isAfterLast(); penyakitCursor.moveToNext()) {
        ix++;
        //datakata.updateIK(db,kataCursor.getString(0).toString(),Integer.toString(ix));
        String ip = penyakitCursor.getString(0).toString();
        String np = penyakitCursor.getString(1).toString();
        String p = penyakitCursor.getString(2).toString();
        String w = penyakitCursor.getString(3).toString();
        String s = penyakitCursor.getString(4).toString();

        ListPenyakit.add(new Penyakit(ip,np,p,w,s));
    }
}

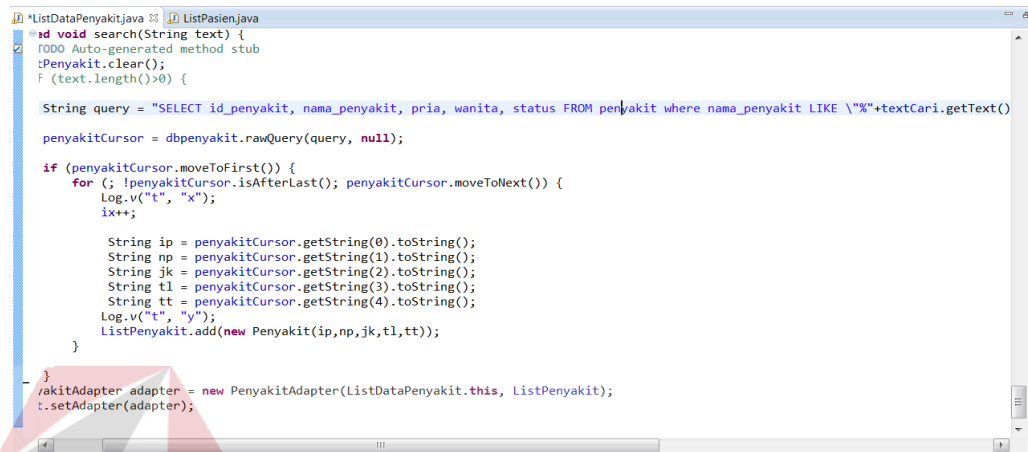
if (!ListPenyakit.isEmpty()){
    PenyakitAdapter adapter = new PenyakitAdapter(ListDataPenyakit.this, ListPenyakit);
    list.setAdapter(adapter);
}
else{
    Toast.makeText(ListDataPenyakit.this, "tidak ada data yang diterima", Toast.LENGTH_SHORT).show();
}

```

Gambar 4.35 Source Code Menampilkan Daftar Nama Penyakit

Gambar 4.36 merupakan *source code* untuk melakukan proses pencarian dengan cara pencocokan huruf pada *database* berdasarkan inputan terapis pada *textbox search* yang terdapat pada *list view* daftar nama pasien. *Source code* yang

terjadi pada *button* tambah penyakit adalah mengalihkan ke *form* penyakit baru yang nantinya data yang diinputkan tersebut akan disimpan kedalam *database*, dapat dilihat pada gambar 4.37



```

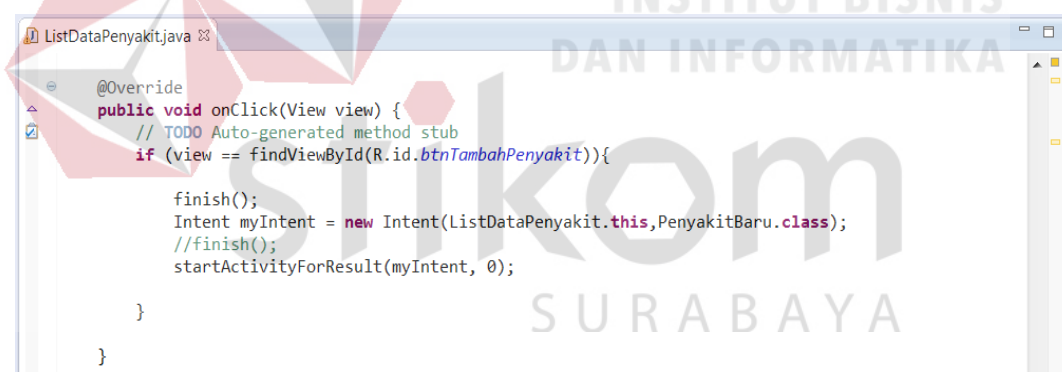
ListDataPenyakit.java ListPasienda.java
@Override
public void search(String text) {
    // TODO Auto-generated method stub
    Penyakit.clear();
    if (text.length() > 0) {
        String query = "SELECT id_penyakit, nama_penyakit, pria, wanita, status FROM penyakit where nama_penyakit LIKE \"%"+textCari.getText()+"%";
        penyakitCursor = dbpenyakit.rawQuery(query, null);

        if (penyakitCursor.moveToFirst()) {
            for (; !penyakitCursor.isAfterLast(); penyakitCursor.moveToNext()) {
                Log.v("t", "x");
                ix++;

                String ip = penyakitCursor.getString(0).toString();
                String np = penyakitCursor.getString(1).toString();
                String jk = penyakitCursor.getString(2).toString();
                String tl = penyakitCursor.getString(3).toString();
                String tt = penyakitCursor.getString(4).toString();
                Log.v("t", "y");
                ListPenyakit.add(new Penyakit(ip,np,jk,tl,tt));
            }
        }
    }
}
PenyakitAdapter adapter = new PenyakitAdapter(ListDataPenyakit.this, ListPenyakit);
this.setAdapter(adapter);

```

Gambar 4.36 Source Code Pencarian Nama Penyakit



```

ListDataPenyakit.java
@Override
public void onClick(View view) {
    // TODO Auto-generated method stub
    if (view == findViewById(R.id.btnTambahPenyakit)){
        finish();
        Intent myIntent = new Intent(ListDataPenyakit.this, PenyakitBaru.class);
        //finish();
        startActivityForResult(myIntent, 0);
    }
}

```

Gambar 4.37 Source Code Menghubungkan ke Form Penyakit Baru

Setelah memilih form tambah data penyakit maka terapis harus mengisi data nama dan jenis kelamin terlebih dahulu. Tetapi pada fitur jenis kelamin dibuat checkbox bertujuan untuk terapis dapat memilih salah satu jenis kelamin atau keduanya agar dapat ditangani. Karena beberapa penyakit tertentu hanya dapat disembuhkan berdasarkan jenis kelamin pasien, sebagai contoh pengobatan penyakit diabetes mellitus pada pria dan wanita memiliki lokasi titik bekam yang



berbeda. Tetapi ada juga penyakit pada pria maupun wanita memiliki lokasi titik bekam yang sama. Form tersebut dapat dilihat pada gambar 4.38.

Gambar 4.38 Form Tambah Data Penyakit

```

*PenyakitBaru.java
public void onClick(View view) {
    // TODO Auto-generated method stub
    if (view == findViewById(R.id.btnSimpan)){
        String idp = "";

        datapenyakit = new DataPenyakit(this);
        dbpenyakit = datapenyakit.getWritableDatabase();
        penyakitCursor = dbpenyakit.rawQuery("SELECT id_penyakit FROM penyakit order by id_penyakit desc limit 1", null);
        if (penyakitCursor.moveToFirst()) {
            idp=(Integer.parseInt(penyakitCursor.getString(0).toString()+1)+"");
        }else{
            idp="1";
        }
        String nm = txtNama.getText().toString();

        if(checkBoxJK1.isChecked()){
            pria="ya";
        }else{
            pria="tidak";
        }

        if(checkBoxJK2.isChecked()){
            wanita="ya";
        }else{
            wanita="tidak";
        }
    }
    if(nm.length()>0 && (wanita=="ya" || pria=="ya")){

        datapenyakit = new DataPenyakit(this);
        dbpenyakit = datapenyakit.getWritableDatabase();
        datapenyakit.generateData(dbpenyakit,idp,nm,pria,wanita,"user");

        Bundle bundle = new Bundle();
        bundle.putString("id_penyakit",idp);
        Intent myIntent = new Intent(PenyakitBaru.this,TahapanBekamBaru.class);
        finish();
        myIntent.putExtras(bundle);
        startActivityForResult(myIntent, 0);
    }else{
        Toast.makeText(PenyakitBaru.this, "inputan data salah", Toast.LENGTH_SHORT).show();
    }
}
}
}

```

Gambar 4.39 Source Code Simpan Data Penyakit Baru

Pada gambar 4.39 merupakan *Source code* tambah data penyakit, dimana berfungsi untuk menambahkan data penyakit kedalam *database* sesuai dengan kebutuhan terapis.

Pada form edit data penyakit terapis dapat melaksanakan perubahan data penyakit yang berfungsi bila terjadi kesalahan pada pengisian data penyakit. Pada form edit data penyakit yang dapat dirubah yaitu nama penyakit dan jenis kelamin pasien yang dapat ditangani yang kemudian dapat disimpan dengan menekan *button* simpan untuk menyimpan kedalam *database* atau terapis dapat juga memilih *button* hapus untuk menghapus data penyakit dari *database*, seperti terlihat pada gambar 4.40. *Source code* edit data penyakit dapat dilihat pada gambar 4.41, dimana berfungsi untuk mengubah data penyakit pada *database* jika terjadi kesalahan pengisian data penyakit dan proses hapus data penyakit dapat dilihat pada gambar 4.42, dimana berfungsi untuk menghapus data penyakit beserta pengobatan titik-titik bekam pada *database*.



The image shows a mobile application interface for editing disease data. The title bar is green and says 'Edit Data Penyakit'. The form contains a text input for 'Nama Penyakit' with the value 'Sembelit'. Below that, there are two radio buttons for 'Jenis Kelamin Yang Dapat Ditangani', with 'pria' and 'wanita' both selected. At the bottom, there are two buttons: 'Hapus' and 'Simpan'.

Gambar 4.40 Form Edit Data Penyakit

```

*EditPenyakit.java
public void onClick(View view) {
    // TODO Auto-generated method stub
    if (view == findViewById(R.id.btnSimpan)){
        String nm = txtNama.getText().toString();
        if(checkBoxJK1.isChecked()){
            pria="ya";
        }else{
            pria="tidak";
        }
        if(checkBoxJK2.isChecked()){
            wanita="ya";
        }else{
            wanita="tidak";
        }
        if(nm.length()>0 && (wanita=="ya" || pria=="ya")){
            datapenyakit = new DataPenyakit(this);
            dbpenyakit = datapenyakit.getWritableDatabase();
            datapenyakit.update(dbpenyakit,id_penyakit,nm,pria,wanita);

            Bundle bundle = new Bundle();
            bundle.putString("id_penyakit",id_penyakit);
            Intent myIntent = new Intent(EditPenyakit.this,DetailPenyakit.class);
            finish();
            myIntent.putExtras(bundle);
            startActivityForResult(myIntent, 0);
        }else{

```

Gambar 4.41 Source Code Simpan Hasil Edit Data Penyakit

```

*EditPenyakit.java
public void onClick(DialogInterface arg0, int arg1) {

    datatahapanpenyakit = new DataTahapanPenyakit(EditPenyakit.this);
    dbtahapanpenyakit = datatahapanpenyakit.getWritableDatabase();
    tahapanpenyakitCursor = dbtahapanpenyakit.rawQuery("SELECT id_tahapan_penyakit,id_penyakit,tahapan_ke,keterangan FROM tahapan_penyakit WHERE id_penyakit=?", new String[] {id_penyakit});
    if (tahapanpenyakitCursor.moveToFirst()) {
        for (; !tahapanpenyakitCursor.isAfterLast(); tahapanpenyakitCursor.moveToNext()) {

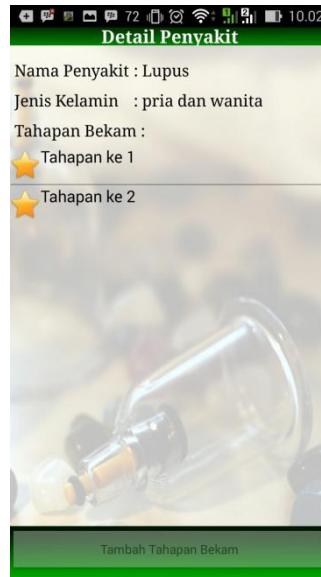
            String it = tahapanpenyakitCursor.getString(0).toString();
            datatitikpenyakit = new DataTitikPenyakit(EditPenyakit.this);
            dbtitikpenyakit = datatitikpenyakit.getWritableDatabase();
            datatitikpenyakit.delete_byTahapan(dbtitikpenyakit,it);
        }
    }

    datapenyakit = new DataPenyakit(EditPenyakit.this);
    dbpenyakit = datapenyakit.getWritableDatabase();
    datapenyakit.delete_byID(dbpenyakit,id_penyakit);
    datatahapanpenyakit = new DataTahapanPenyakit(EditPenyakit.this);
    dbtahapanpenyakit = datatahapanpenyakit.getWritableDatabase();
    datatahapanpenyakit.delete_byPenyakit(dbtahapanpenyakit,id_penyakit);
    Intent myIntent = new Intent(EditPenyakit.this,ListDataPenyakit.class);
    finish();
    startActivityForResult(myIntent, 0);
}

```

Gambar 4.42 Source Code Menghapus Data Penyakit dari Database

Pada gambar 4.43 merupakan tampilan dari detail data penyakit yang terdiri atas nama penyakit, jenis kelamin yang dapat ditangani, jumlah tahapan pengobatan dan *button* untuk menambah jumlah tahapan pengobatan bekam sesuai dengan kebutuhan terapis.



Gambar 4.43 Form Detail Data Penyakit

Proses menampilkan data kategori pada *listview* dapat dilihat pada gambar 4.44, dimana *source code* tersebut menampilkan data jumlah tahapan dari database kedalam *listview*.

```

DetailPenyakit.java
ix=0;
datatahapanpenyakit = new DataTahapanPenyakit(this);
dbtahapanpenyakit = datatahapanpenyakit.getWritableDatabase();
tahapapanpenyakitCursor = dbtahapanpenyakit.rawQuery("SELECT id_tahapan_penyakit,id_penyakit,tahapan_ke,keterangan FROM tahapan_penyakit
if (tahapapanpenyakitCursor.moveToFirst()) {
    for (; !tahapapanpenyakitCursor.isAfterLast(); tahapapanpenyakitCursor.moveToNext()) {
        ix++;
        //datakata.updateIK(db,kataCursor.getString(0).toString(),Integer.toString(ix));
        String it = tahapapanpenyakitCursor.getString(0).toString();
        String ip = tahapapanpenyakitCursor.getString(1).toString();
        String t = tahapapanpenyakitCursor.getString(2).toString();
        String k = tahapapanpenyakitCursor.getString(3).toString();

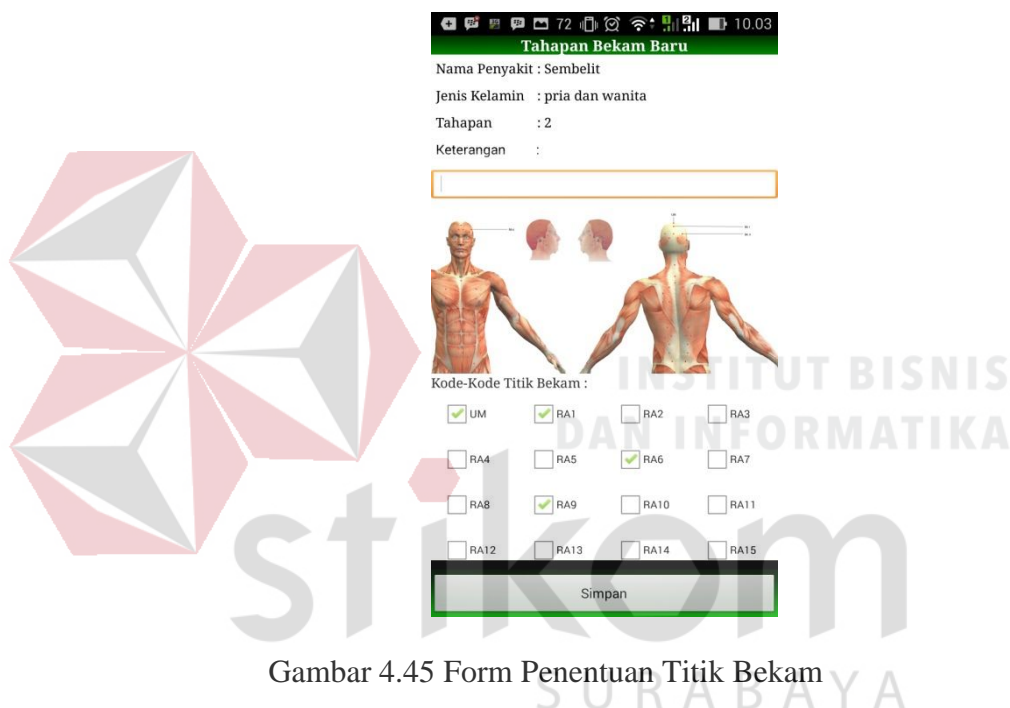
        ListTahapanPenyakit.add(new TahapanPenyakit(it,ip,t,k));
    }
}

```

Gambar 4.44 *Source Code* Menampilkan Jumlah Tahapan Kedalam Listview

Pada gambar 4.45 merupakan hasil dari penambahan tahapan pengobatan bekam pada form detail penyakit. Pada form tampilan dari detail data penyakit

terdiri atas nama penyakit, jenis kelamin yang dapat ditangani, jumlah tahapan pengobatan yang dipilih pada proses sebelumnya dan terapis dapat mengisi keterangan pada *textview* keterangan dan dapat menambahkan titik-titik bekam yang akan digunakan pada *checkbox* sesuai dengan kebutuhan terapis. *Button* untuk menambahkan jumlah tahapan pengobatan bekam sesuai dengan kebutuhan terapis.



Gambar 4.45 Form Penentuan Titik Bekam

*Source code* menampilkan nama penyakit, jenis kelamin dan jumlah tahapan bekam pada *textview* dapat dilihat pada gambar 4.46 dan gambar 4.47, dimana *source code* tersebut menampilkan data jumlah tahapan dari database kedalam *textview*. *Source code* menampilkan lokasi titik bekam pada *webview* dapat dilihat pada gambar 4.48, dimana *source code* tersebut menampilkan data jumlah tahapan dari *function javascript* kedalam *webview*.

```

TahapanBekamBaru.java
Log.v("", "2");
Bundle bundle = this.getIntent().getExtras();
id_penyakit = bundle.getString("id_penyakit");

datapenyakit = new DataPenyakit(this);
dbpenyakit = datapenyakit.getWritableDatabase();
penyakitCursor = dbpenyakit.rawQuery("SELECT id_penyakit, nama_penyakit, pria, wanita, status FROM penyakit where id_penyakit='"+id_pe
if (penyakitCursor.moveToFirst()) {
    Log.v("", "3");
    String ip = penyakitCursor.getString(0).toString();
    String np = penyakitCursor.getString(1).toString();
    String p = penyakitCursor.getString(2).toString();
    String w = penyakitCursor.getString(3).toString();
    String s = penyakitCursor.getString(4).toString();
    Log.v("", "4");

    txtNamaPenyakit.setText(" : "+np);
    Log.v("", "5");
    if(p.equalsIgnoreCase("ya") && w.equalsIgnoreCase("ya")){
        txtJenisKelamin.setText(" : "+p+" dan wanita");
    }else if(p.equalsIgnoreCase("ya") && w.equalsIgnoreCase("tidak")){
        txtJenisKelamin.setText(" : "+p+"pria");
    }else if(p.equalsIgnoreCase("tidak") && w.equalsIgnoreCase("ya")){
        txtJenisKelamin.setText(" : "+w+"wanita");
    }
}
}

```

Gambar 4.46 Source Code Nama Penyakit dan Jenis Kelamin

```

TahapanBekamBaru.java
datatahapanpenyakit = new DataTahapanPenyakit(this);
dbtahapanpenyakit = datatahapanpenyakit.getWritableDatabase();
tahapanpenyakitCursor = dbtahapanpenyakit.rawQuery("SELECT tahapan_ke FROM tahapan_penyakit where id_penyakit='"+id_penyakit+" order
if (tahapanpenyakitCursor.moveToFirst()) {
    tahapan_ke=(Integer.parseInt(tahapanpenyakitCursor.getString(0))+1)+"";
} else {
    tahapan_ke="1";
}

txtTahapan.setText(" : "+tahapan_ke);

```

Gambar 4.47 Source Code Jumlah Tahapan Pengobatan

```

TahapanBekamBaru.java
public void checkRA3(View v) {
    if (((CheckBox) v).isChecked()) { myBrowser.loadUrl("javascript:ra3_a()");}
    else{ myBrowser.loadUrl("javascript:ra3_b()");}
}
public void checkRA4(View v) {
    if (((CheckBox) v).isChecked()) { myBrowser.loadUrl("javascript:ra4_a()");}
    else{ myBrowser.loadUrl("javascript:ra4_b()");}
}
public void checkRA5(View v) {
    if (((CheckBox) v).isChecked()) { myBrowser.loadUrl("javascript:ra5_a()");}
    else{ myBrowser.loadUrl("javascript:ra5_b()");}
}
public void checkRA6(View v) {
    if (((CheckBox) v).isChecked()) { myBrowser.loadUrl("javascript:ra6_a()");}
    else{ myBrowser.loadUrl("javascript:ra6_b()");}
}
public void checkRA7(View v) {
    if (((CheckBox) v).isChecked()) { myBrowser.loadUrl("javascript:ra7_a()");}
    else{ myBrowser.loadUrl("javascript:ra7_b()");}
}
public void checkRA8(View v) {
    if (((CheckBox) v).isChecked()) { myBrowser.loadUrl("javascript:ra8_a()");}
    else{ myBrowser.loadUrl("javascript:ra8_b()");}
}
public void checkRA9(View v) {
    if (((CheckBox) v).isChecked()) { myBrowser.loadUrl("javascript:ra9_a()");}
    else{ myBrowser.loadUrl("javascript:ra9_b()");}
}

```

Gambar 4.48 Source Code Titik Bekam pada Gambar Anatomi

*Source code* menyimpan hasil inputan keterangan pada *textView* dan kode titik bekam pada *checkbox* dapat dilihat pada gambar 4.49, dimana *source code* tersebut menyimpan data pada form tahapan bekam baru kedalam *database* aplikasi.



```

public void onClick(View view) {
    // TODO Auto-generated method stub
    if (view == findViewById(R.id.btnSimpan)){
        String idt = "";
        datatahapanpenyakit = new DataTahapanPenyakit(this);
        dbtahapanpenyakit = datatahapanpenyakit.getWritableDatabase();
        tahapanpenyakitCursor = dbtahapanpenyakit.rawQuery("SELECT id_tahapan_penyakit FROM tahapan_penyakit order by id_tahapan_penyakit asc", null);
        if (tahapanpenyakitCursor.moveToFirst()) {
            idt=(Integer.parseInt(tahapanpenyakitCursor.getString(0).toString()+1)+"");
        }else{ idt="1"; }
        String kt = txtKeterangan.getText().toString();
        datatahapanpenyakit = new DataTahapanPenyakit(this);
        dbtahapanpenyakit = datatahapanpenyakit.getWritableDatabase();
        datatahapanpenyakit.generateData(dbtahapanpenyakit,idt,id_penyakit,tahapan_ke,kt);
        Log.v("insert", "1");
        Integer itp = 0;
        datatitikpenyakit = new DataTitikPenyakit(this);
        dbtitikpenyakit = datatitikpenyakit.getWritableDatabase();
        titikpenyakitCursor = dbtitikpenyakit.rawQuery("SELECT id_titik_penyakit FROM titik_penyakit order by id_titik_penyakit desc limit 1", null);
        if (titikpenyakitCursor.moveToFirst()) {
            itp=(Integer.parseInt(titikpenyakitCursor.getString(0).toString()+1));
        }else{ itp=1; }
        Log.v("insert", "2");
        datatitikpenyakit = new DataTitikPenyakit(this);
        dbtitikpenyakit = datatitikpenyakit.getWritableDatabase();
        Log.v("insert", "3");
        Log.v("insert", itp+"idt"+idt);
    }
}

```

Gambar 4.49 Source Code Button Simpan

### 4.3 Evaluasi Sistem

Setelah melakukan implementasi sistem, tahap selanjutnya adalah melakukan uji coba dan evaluasi sistem. Tujuan evaluasi ini adalah untuk mengetahui apakah aplikasi yang telah dibuat ini sudah berjalan dengan baik dan sesuai dengan tujuan atau output yang diharapkan. Uji coba ini akan dilakukan dengan menggunakan metode *black box testing*. Adapun evaluasi yang dilakukan adalah sebagai berikut:

1. Hasil uji coba aplikasi
2. Hasil uji coba aplikasi kepada pengguna
3. Analisa hasil uji coba system

### 4.3.1 Hasil Uji Coba Aplikasi

Proses uji coba dilakukan mengetahui apakah fungsi dari aplikasi pendukung penentuan titik bekam berbasis android telah berjalan sesuai yang diharapkan. Setiap fitur yang di sediakan akan diuji hasilnya sesuai dengan tabel *test case*. Hasil uji coba fungsi aplikasi adalah sebagai berikut:

#### A. Hasil Uji Coba Pengenalan Titik Bekam

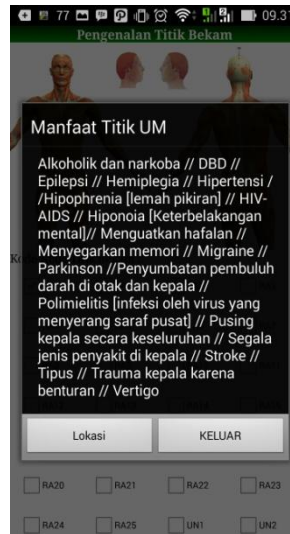
Pada menu pengenalan titik bekam berfungsi untuk mengetahui lokasi titik bekam dan manfaat dari titik bekam tersebut sesuai dengan kode titik bekam yang dipilih oleh terapi. Uji coba pengenalan titik bekam ini bertujuan untuk mengetahui apakah fungsi pengenalan titik bekam dapat berjalan dengan baik. Hasil uji coba tersebut dapat dilihat pada tabel 4.1

Tabel 4.1 Hasil Uji Coba Menampilkan Lokasi dan Manfaat Titik Bekam

Test Case ID	Tujuan	Input	Output Yang Diharapkan	Status
1.	Menampilkan lokasi titik-titik bekam dan manfaat dari titik bekam.	Pilih kode titik bekam pada checkbox daftar kode titik bekam	Pada gambar anatomi titik bekam tampil titik-titik bekam beserta kode titik bekam dan pada textbox manfaat kode titik bekam menampilkan nama-nama penyakit yang dapat disembuhkan.	Sukses

Pada tabel 4.1 merupakan pembuktian fungsi uji coba terhadap menampilkan lokasi titik bekam dan manfaat titik bekam. Pembuktian ini berstatus sukses karena dapat menampilkan lokasi titik bekam dan manfaat dari titik bekam yang dipilih. Pembuktian sukses terhadap menampilkan lokasi titik bekam dan manfaat titik bekam dapat dilihat pada gambar 4.50.





Gambar 4.50 Hasil Uji Coba Menampilkan Lokasi dan Manfaat Titik Bekam

#### B. Hasil Uji Coba Data Pasien

Halaman daftar pasien berfungsi untuk menambah nama pasien yang ditangani oleh terapis, kemudian data tersebut disimpan kedalam *database* aplikasi agar aplikasi dapat membuat histori dari nama pasien yang dipilih. Hasil uji coba tambah nama pasien dapat dilihat pada tabel 4.2.

Tabel 4.2 Hasil Uji Coba Menambah Nama Pasien

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
2.	Menambahkan nama pasien.	Pilih tambah pasien, kemudian masukkan nama pasien	Nama pasien baru tampil pada daftar nama pasien.	Sukses

Pada tabel 4.2 merupakan fungsi uji coba terhadap penambahan data pasien. Pembuktian ini berstatus sukses karena dapat menambahkan data pasien baru. Berikut pembuktian sukses dalam menambah data pasien pada gambar 4.51.



Gambar 4.51 Hasil Uji Coba Menambah Nama Pasien

Berikutnya pada tabel 4.3 adalah menampilkan uji daftar nama pasien yang sudah dibuat sebelumnya. Fitur menampilkan daftar nama pasien diuji apakah dapat menampilkan daftar nama pasien sesuai dengan data yang pernah diinputkan sebelumnya. Hasil uji coba menampilkan daftar nama pasien dapat dilihat pada tabel 4.3.

Tabel 4.3 Hasil Uji Coba Menampilkann Daftar Nama Pasien

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
3.	Menampilkan daftar nama pasien yang sudah tersimpan.	Masuk menu pengobatan bekam pada menu utama.	Tampil daftar nama pasien yang terdapat pada database aplikasi.	Sukses

Setelah dilakukan pembuktian pada tabel 4.3 ternyata hasilnya adalah sukses, nama pasien dapat masuk ke dalam daftar nama pasien sesuai dengan nama pasien yang telah dibuat sebelumnya. Berikut pembuktiannya pada gambar 4.52.



Gambar 4.52 Hasil Uji Coba Menampilkann Daftar Nama Pasien

Berikutnya adalah uji pencarian nama pasien yang sudah dibuat sebelumnya. Fitur pencarian nama pasien diuji apakah dapat melakukan proses

pencarian sesuai dengan data yang diinputkan. Berikut hasil uji coba pencarian nama pasien dapat dilihat pada tabel 4.4.

Tabel 4.4 Hasil Uji Coba Pencarian Nama Pasien

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
4.	Melakukan pencarian nama pasien.	Menginputkan nama pasien pada textbox search	Nama pasien yang dicari tampil pada daftar nama pasien.	Sukses

Setelah dilakukan pembuktian pada tabel 4.4 ternyata hasilnya adalah sukses, nama pasien yang dicari masuk ke dalam daftar nama pasien sesuai dengan nama pasien yang diinputkan. Berikut pembuktiannya pada gambar 4.53.



Gambar 4.53 Hasil Uji Coba Pencarian Nama Pasien

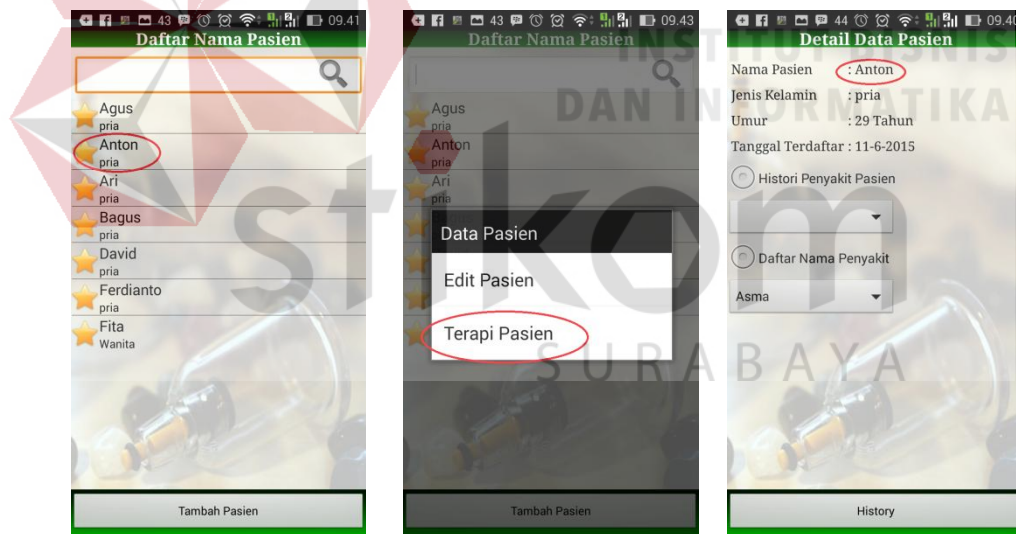
Uji berikutnya adalah ketika terapis memilih nama pasien yang akan ditangani. Setelah memilih nama pasien sistem menampilkan *pop-up* berupa pilihan untuk melakukan *edit* data pasien atau terapi pasien untuk melanjutkan ke

tahap terapi pasien. Berikut hasil uji coba masuk kemenu berikutnya dapat dilihat pada tabel 4.5.

Tabel 4.5 Hasil Uji Coba Masuk Kemenu Berikutnya

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
5.	Masuk nama pasien yang dipilih	Pilih nama pasien	Tampil nama pasien yang dipilih pada proses pengobatan pasien	Sukses

Setelah dilakukan pembuktian pada tabel 4.5 ternyata hasilnya adalah sukses, terapis dapat masuk ke dalam data pasien sesuai dengan nama pasien yang dipilih. Berikut pembuktiannya pada gambar 4.54.



Gambar 4.54 Hasil Uji Coba Masuk Kemenu Berikutnya

Berikutnya adalah uji edit data pasien jika terjadi kesalahan dalam pengisian data diri pasien yang sudah dibuat sebelumnya. Edit pasien yang telah dibuat, diuji apakah dapat melakukan perubahan data sesuai data yang diperlukan oleh terapis. Berikut pada tabel 4.6 hasil uji coba edit data diri pasien.

Tabel 4.6 Hasil Uji Coba Edit Data Diri Pasien

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
6.	Edit data diri pasien.	Memilih edit pasien pada pop-up nama pasien.	Data diri pasien sudah berubah sesuai dengan <i>inputan</i> baru	Sukses

Setelah dilakukan pembuktian pada tabel 4.6 ternyata hasilnya adalah sukses, data diri pasien dapat berubah sesuai kebutuhan terapis sehingga data diri pasien berubah sesuai inputan yang baru. Berikut pembuktiannya pada gambar 4.55.



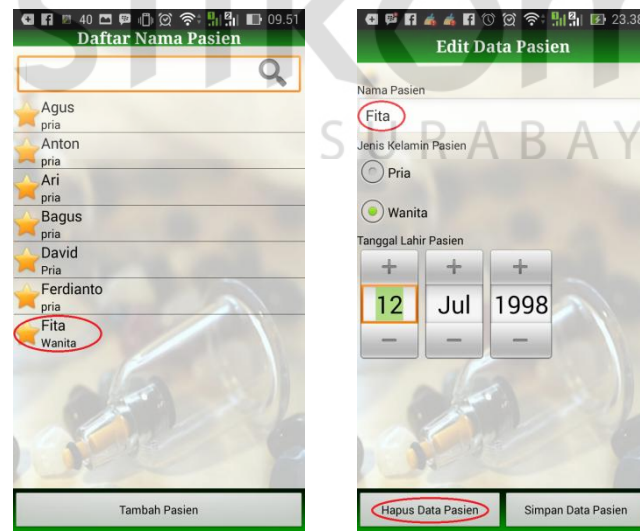
Gambar 4.55 Hasil Uji Coba Edit Data Diri Pasien

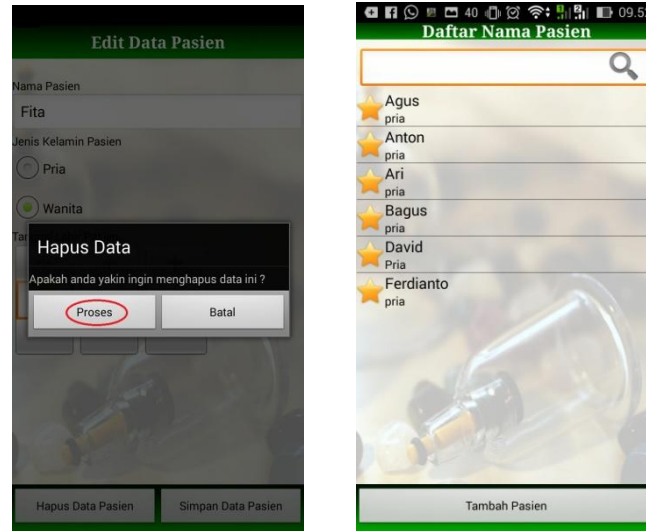
Berikutnya adalah uji hapus data pasien jika terjadi kesalahan data atau data tersebut tidak dibutuhkan, maka terapis dapat menghapus nama pasien tersebut. Hapus data pasien yang telah dibuat, diuji apakah dapat melakukan penghapusan data sesuai data pasien dari daftar nama pasien. Berikut pada tabel 4.7 hasil uji coba menghapus data pasien.

Tabel 4.7 Hasil Uji Coba Menghapus Data Pasien

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
7.	Menghapus data pasien	Memilih nama pasien pada daftar nama pasien untuk dihapus.	Nama pasien sudah terhapus dari daftar pasien	Sukses

Setelah dilakukan pembuktian pada tabel 4.7 ternyata hasilnya adalah sukses, data pasien dapat terhapus dari daftar nama pasien. Berikut pembuktiannya pada gambar 4.56.





Gambar 4.56 Hasil Uji Coba Menghapus Data Pasien

### C. Hasil Uji Coba Pengobatan Bekam

Halaman detail data pasien berfungsi untuk menampilkan seluruh data tentang pasien, nantinya data tersebut dapat digunakan untuk melakukan pengobatan bekam terhadap pasien. Desain uji detail data pasien dapat dilihat pada tabel 4.8 hasil uji coba menampilkan detail data pasien.

Tabel 4.8 Hasil Uji Coba Menampilkan Detail Data Pasien

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
8.	Menampilkan detail data pasien.	Pilih nama pasien	Tampil detail data diri pasien yang disertai histori penyakit pasien.	Sukses

Setelah dilakukan pembuktian pada tabel 4.8 ternyata hasilnya adalah sukses, terapis dapat melihat detail data pasien dan dapat memilih jenis pengobatan terhadap. Berikut pembuktiannya pada gambar 4.57.





Gambar 4.57 Hasil Uji Coba Menampilkan Detail Data Pasien

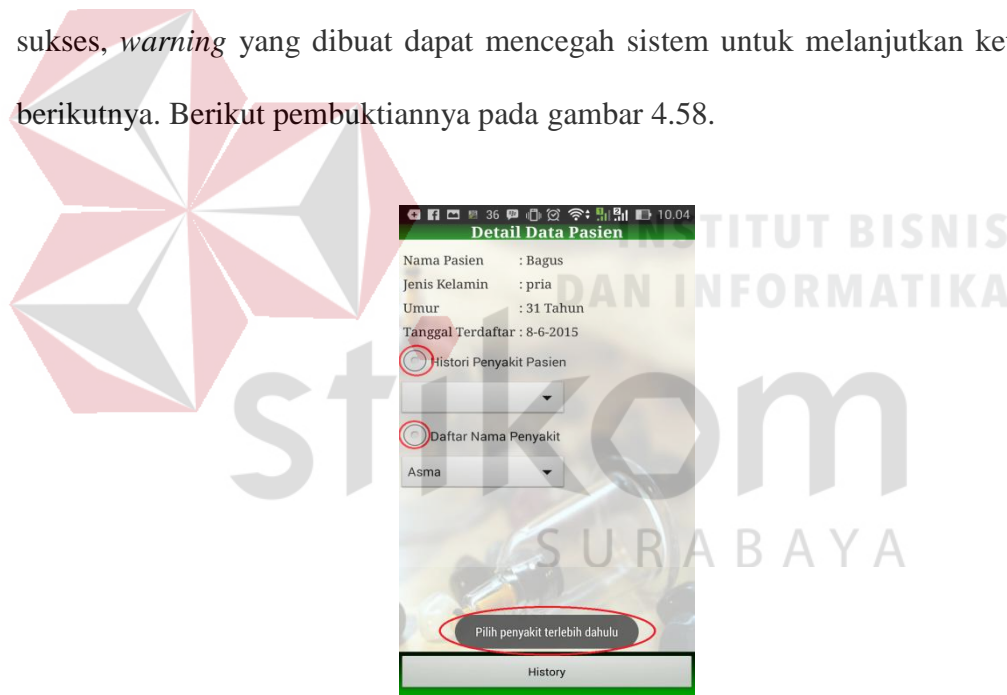
Pada tabel 4.9 adalah uji coba ketika terapis akan masuk ke tahap penentuan titik bekam tanpa memilih *radiobutton* histori nama penyakit pasien maupun daftar nama penyakit maka sistem akan memberi peringatan untuk memilih nama penyakit terlebih dahulu. Pada tahap ini *warning* yang dibuat, diuji apakah dapat mencegah sistem untuk melanjutkan ketahapan berikutnya bila terapis tidak memilih nama penyakit dan dianjurkan memilih salah satu *radiobutton* yang suda di sediakan. Berikut pada tabel 4.9 hasil uji coba *warning*

tanpa memilih nama penyakit untuk mencegah bila terapis tidak memilih nama penyakit dan *radiobutton*.

Tabel 4.9 Hasil Uji Coba *Warning* Tanpa Memilih Nama Penyakit

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
9.	<i>Warning</i> tanpa memilih nama penyakit.	Pilih menu lanjut pengobatan tanpa memilih salah satu radiobutton.	Tampil peringatan untuk memilih nama penyakit terlebih dahulu.	Sukses

Setelah dilakukan pembuktian pada tabel 4.9 ternyata hasilnya adalah sukses, *warning* yang dibuat dapat mencegah sistem untuk melanjutkan ketahap berikutnya. Berikut pembuktiannya pada gambar 4.58.



Gambar 4.58 Hasil Uji Coba *Warning* Tanpa Memilih Nama Penyakit

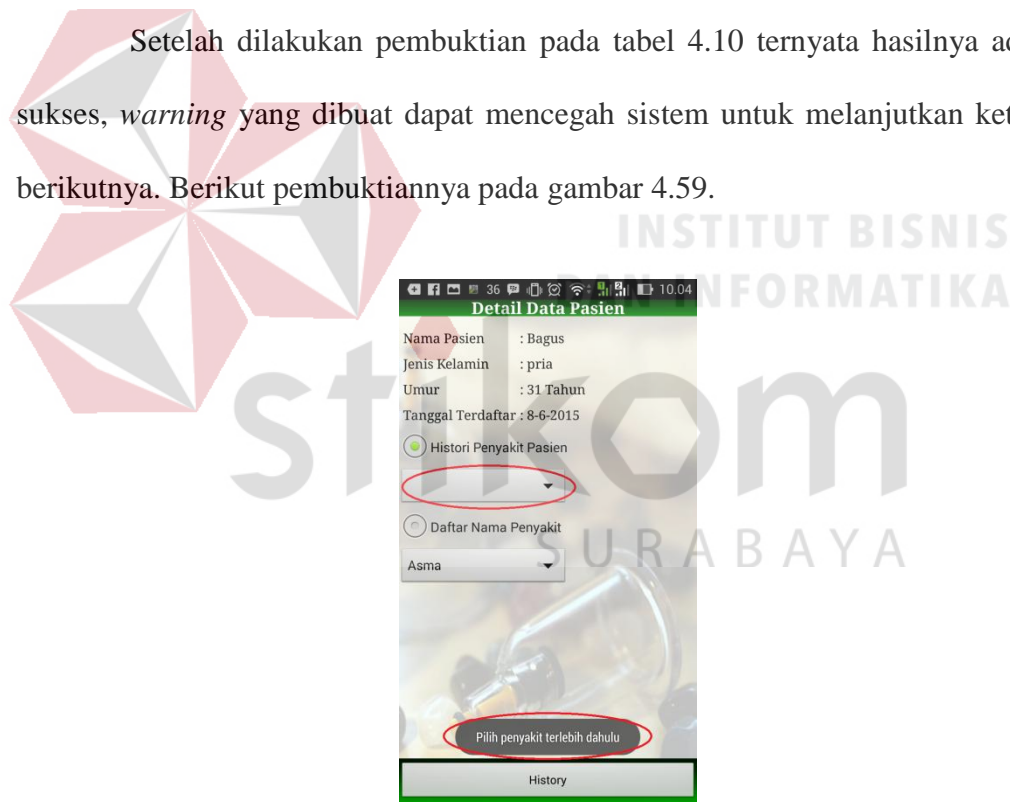
Pada tabel 4.10 adalah uji coba ketika terapis akan memilih nama penyakit pada histori penyakit pasien, tetapi pasien belum pernah berbekam. Maka tidak ada data penyakit pada histori penyakit pasien. Pada tahap ini *warning* yang dibuat, diuji apakah dapat mencegah sistem untuk melanjutkan ketahapan berikutnya bila histori penyakit pasien tidak terdapat nama penyakit. Berikut pada

tabel 4.10 hasil uji coba *warning* untuk mencegah bila histori penyakit pasien tidak terdapat nama penyakit.

Tabel 4.10 Hasil Uji Coba *Warning* Tidak Memiliki Histori Penyakit

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
10.	Histori penyakit tidak dapat tampil bila nama penyakit belum pernah melakukan bekam.	Pilih <i>radiobutton</i> histori penyakit pasien	Tampil <i>messagebox</i> untuk memilih penyakit terlebih dahulu pada <i>radiobutton</i> daftar nama penyakit.	Sukses

Setelah dilakukan pembuktian pada tabel 4.10 ternyata hasilnya adalah sukses, *warning* yang dibuat dapat mencegah sistem untuk melanjutkan ketahap berikutnya. Berikut pembuktiannya pada gambar 4.59.



Gambar 4.59 Hasil Uji Coba *Warning* Tidak Memiliki Histori Penyakit

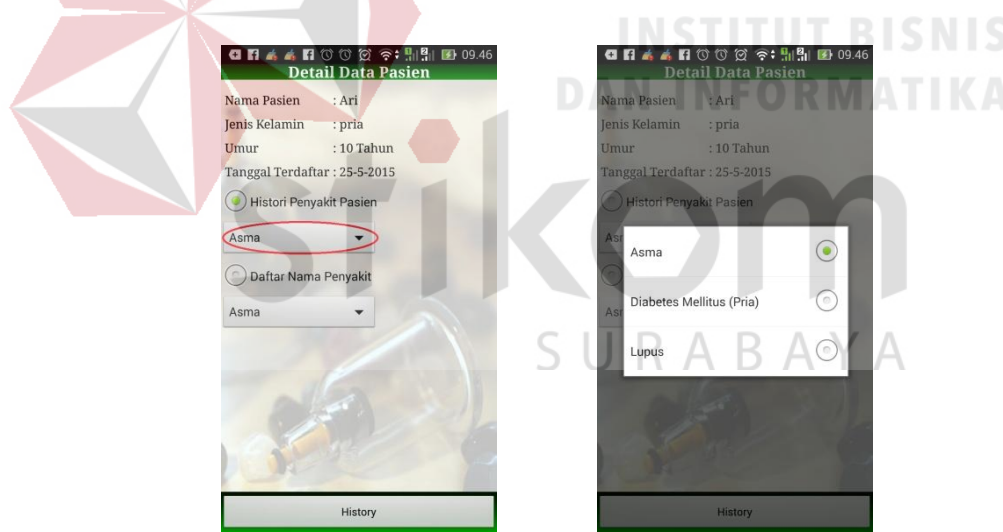
Berikutnya adalah uji coba untuk melihat daftar nama penyakit yang pernah di bekam pada *combobox* histori penyakit pasien. Histori penyakit pasien yang telah dibuat, diuji apakah dapat menampilkan daftar nama penyakit yang

pernah dibekam. Berikut pada tabel 4.11 hasil uji coba menampilkan histori nama penyakit yang pernah dibekam pada *combobox* histori penyakit pasien.

Tabel 4.11 Hasil Uji Coba Menampilkan Histori Nama Penyakit

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
11.	Menampilkan daftar nama penyakit yang pernah dibekam pada <i>combobox</i> histori penyakit pasien.	Pilih <i>combobox</i> histori penyakit pasien.	Tampil daftar nama penyakit yang pernah dibekam.	Sukses

Setelah dilakukan pembuktian pada tabel 4.11 ternyata hasilnya adalah sukses, *combobox* histori penyakit pasien dapat menampilkan daftar nama penyakit yang pernah dibekam. Berikut pembuktiannya pada gambar 4.60.



Gambar 4.60 Hasil Uji Coba Menampilkan Histori Nama Penyakit

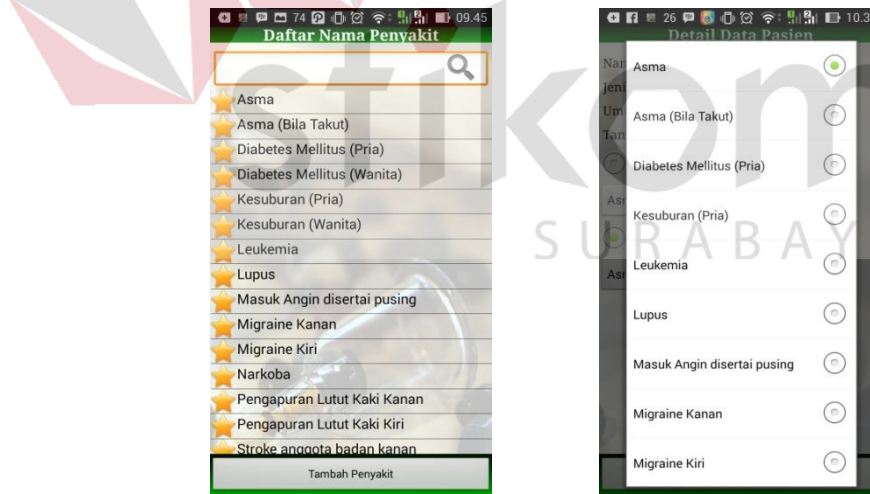
Berikutnya adalah uji coba untuk melihat daftar nama penyakit yang terdapat pada *combobox* daftar nama penyakit. *Combobox* daftar nama penyakit yang telah dibuat, diuji apakah dapat menampilkan daftar nama penyakit yang terdapat pada *master* data penyakit. Berikut pada tabel 4.12 hasil uji coba

menampilkan daftar nama penyakit yang diambil dari *master* data penyakit pada *combobox* daftar nama penyakit.

Tabel 4.12 Hasil Uji Coba Menampilkan Daftar Nama Penyakit

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
12.	Menampilkan daftar nama penyakit yang terdapat pada <i>combobox</i> daftar nama penyakit dengan daftar nama penyakit.	Pilih <i>combobox</i> daftar nama penyakit.	Tampil daftar nama penyakit yang terdata pada <i>master</i> data penyakit.	Sukses

Setelah dilakukan pembuktian pada tabel 4.12 ternyata hasilnya adalah sukses, *combobox* daftar nama penyakit dapat menampilkan daftar nama penyakit yang terdapat pada *master* data penyakit. Berikut pembuktiannya pada gambar 4.61.



Gambar 4.61 Hasil Uji Coba Menampilkan Daftar Nama Penyakit pada Pria

Berikutnya adalah uji coba agar tidak dapat menampilkan *listview* tanggal berobat yang terdapat pada form history pengobatan. *Listview* tanggal berobat yang telah dibuat, diuji apakah dapat mencegah menampilkan daftar

tanggal berobat bila pasien belum pernah berobat berdasarkan penyakit tersebut. Berikut pada tabel 4.13 hasil uji coba tidak dapat menampilkan *listview* tanggal berobat.

Tabel 4.13 Hasil Uji Coba Tidak Menampilkan *Listview* Tanggal Berobat

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
13.	<i>Listview</i> tanggal berobat tidak dapat tampil bila belum pernah berobat bekam.	Pilih nama penyakit pada daftar nama penyakit.	<i>Listview</i> tanggal berobat tidak dapat menampilkan apa-apa.	Sukses

Setelah dilakukan pembuktian pada tabel 4.13 ternyata hasilnya adalah sukses, *listview* pada form history berobat tidak dapat menampilkan daftar tanggal pengobatan karena belum pernah melakukan pengobatan pada penyakit tersebut. Berikut pembuktiannya pada gambar 4.62.



Gambar 4.62 Hasil Uji Coba Tidak Menampilkan *Listview* Tanggal Berobat

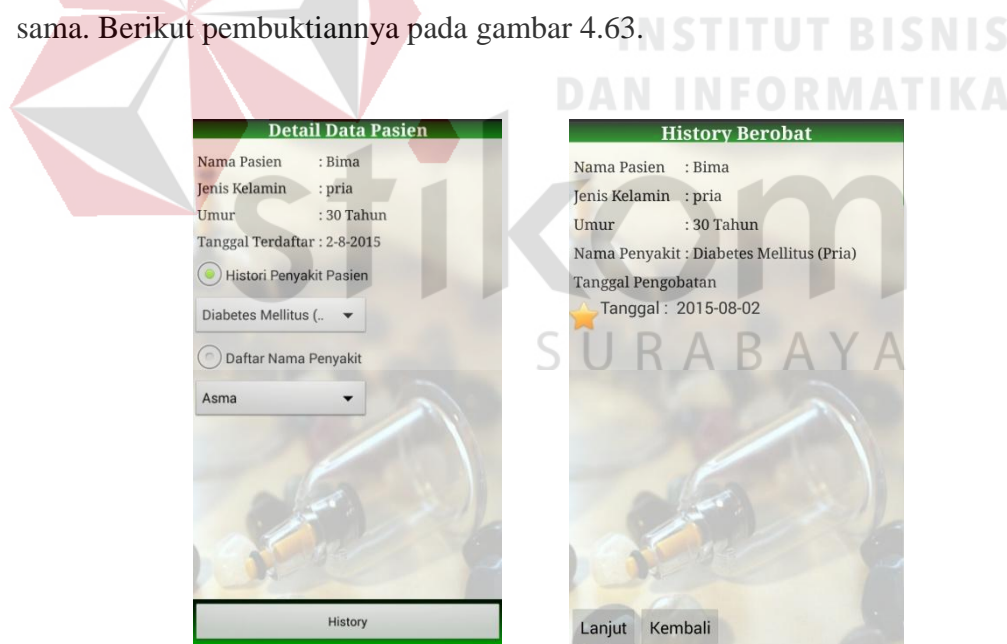
Berikutnya adalah uji coba untuk menampilkan *listview* tanggal berobat yang terdapat pada form history pengobatan. *Listview* tanggal berobat yang telah dibuat, diuji apakah dapat menampilkan daftar tanggal berobat bila sebelumnya

pasien pernah berobat berdasarkan penyakit tersebut. Berikut pada tabel 4.14 hasil uji coba dapat menampilkan *listview* tanggal berobat.

Tabel 4.14 Hasil Uji Coba Dapat Menampilkan *Listview* Tanggal Berobat

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
14.	<i>Listview</i> tanggal berobat dapat tampil bila pernah berobat bekam.	Pilih nama penyakit pada history penyakit pasien maupun daftar nama pasien.	<i>Listview</i> tanggal berobat akan muncul berdasarkan tanggal berobat.	Sukses

Setelah dilakukan pembuktian pada tabel 4.14 ternyata hasilnya adalah sukses, *listview* pada form history pengobatan dapat menampilkan daftar tanggal pengobatan karena sebelumnya pernah melakukan pengobatan pada penyakit yang sama. Berikut pembuktiannya pada gambar 4.63.



Gambar 4.63 Hasil Uji Coba Dapat Menampilkan *Listview* Tanggal Berobat

Berikutnya adalah uji coba *warning* untuk akses pada minggu berikutnya yang terdapat pada form history berobat. *Warning* ini dibuat untuk mencegah kesalahan pada terpis untuk melakukan bekam kurang dari seminggu pada

penyakit yang sama. Berikut pada tabel 4.15 hasil uji coba *warning* untuk akses pada minggu berikutnya berdasarkan tanggal terakhir berobat.

Tabel 4.15 Hasil Uji Coba *Warning* Untuk Akses pada Minggu Berikutnya

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
15.	Setelah berbekam pada penyakit yang sama, terapis tidak dapat mengakses form titik bekam bila kurang dari 7 hari dari pengobatan terakhir.	Pilih penyakit yang sama seperti pengobatan sebelumnya.	Tampil <i>messagebox</i> untuk “Datang lagi setelah seminggu dari terakhir pengobatan”	Sukses

Setelah dilakukan pembuktian pada tabel 4.15 ternyata hasilnya adalah sukses, *warning* untuk akses pada minggu berikutnya dapat muncul bila *user* melakukan pengobatan kembali pada penyakit yang sama kurang dari 7 hari. Berikut pembuktiannya pada gambar 4.64.



Gambar 4.64 Hasil Uji Coba *Warning* Untuk Akses pada Minggu Berikutnya

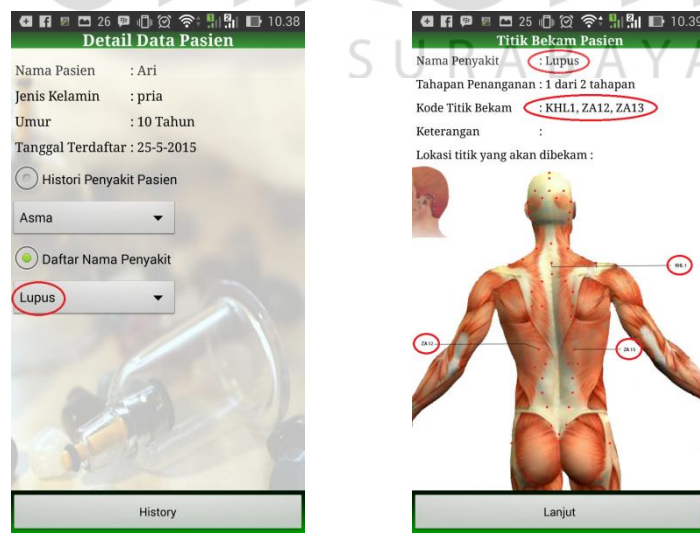


Berikutnya adalah uji coba menampilkan lokasi titik bekam berdasarkan nama penyakit yang dipilih oleh terapis. Penentuan lokasi titik bekam yang telah dibuat, diuji apakah dapat menampilkan lokasi titik bekam berdasarkan nama penyakit, jenis kelamin dan jumlah penanganan kepada pasien. Berikut pada tabel 4.16 hasil uji coba menampilkan lokasi titik bekam.

Tabel 4.16 Hasil Uji Coba Menampilkan Lokasi Titik-Titik Bekam

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
16.	Menampilkan lokasi titik-titik yang akan dibekam	Pilih nama penyakit yang akan ditangani	Tampil lokasi kode titik-titik bekam berdasarkan nama penyakit yang dipilih dan jumlah penanganan pasien	Sukses

Setelah dilakukan pembuktian pada tabel 4.16 ternyata hasilnya adalah sukses, penentuan titik bekam dapat berjalan sesuai inputan yang dimasukkan oleh terapis. Berikut pembuktiannya pada gambar 4.65.



Gambar 4.65 Hasil Uji Coba Menampilkan Lokasi Titik-Titik Bekam

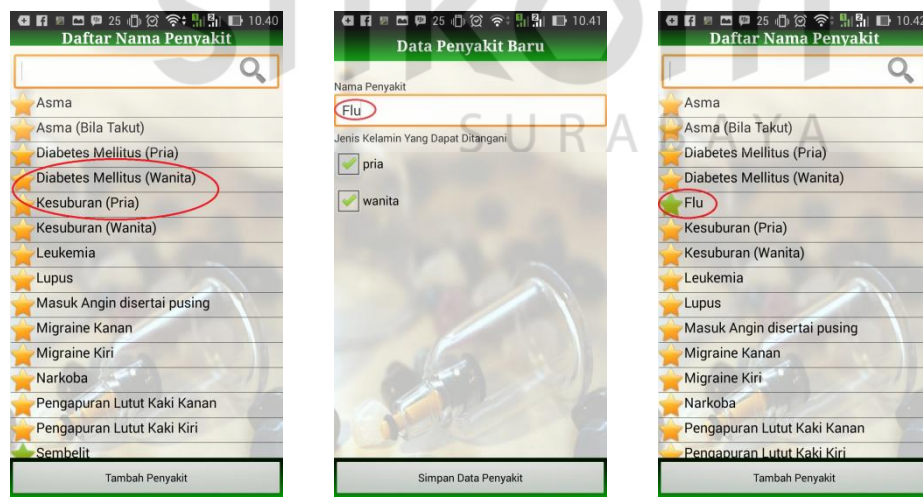
#### D. Hasil Uji Coba Data Penyakit

Halaman daftar pasien berfungsi untuk menambah nama penyakit, kemudian data tersebut disimpan ke dalam *database* aplikasi untuk mendukung proses pengobatan bekam. Hasil uji coba menambah nama penyakit dapat dilihat pada tabel 4.17.

Tabel 4.17 Hasil Uji Coba Menambah Nama Penyakit

Test Case ID	Tujuan	Input	Output Yang Diharapkan	Status
17.	Menambah data nama penyakit.	Pilih <i>button</i> tambah penyakit, kemudian masukkan data penyakit sesuai kebutuhan terapis.	Nama penyakit baru, tampil pada daftar nama penyakit.	Sukses

Setelah dilakukan pembuktian pada tabel 4.17 ternyata hasilnya adalah sukses, penambahan nama penyakit dapat tersimpan kedalam *database* sesuai inputan yang dimasukkan oleh terapis. Berikut pembuktiannya pada gambar 4.66.



Gambar 4.66 Hasil Uji Coba Menambah Nama Penyakit

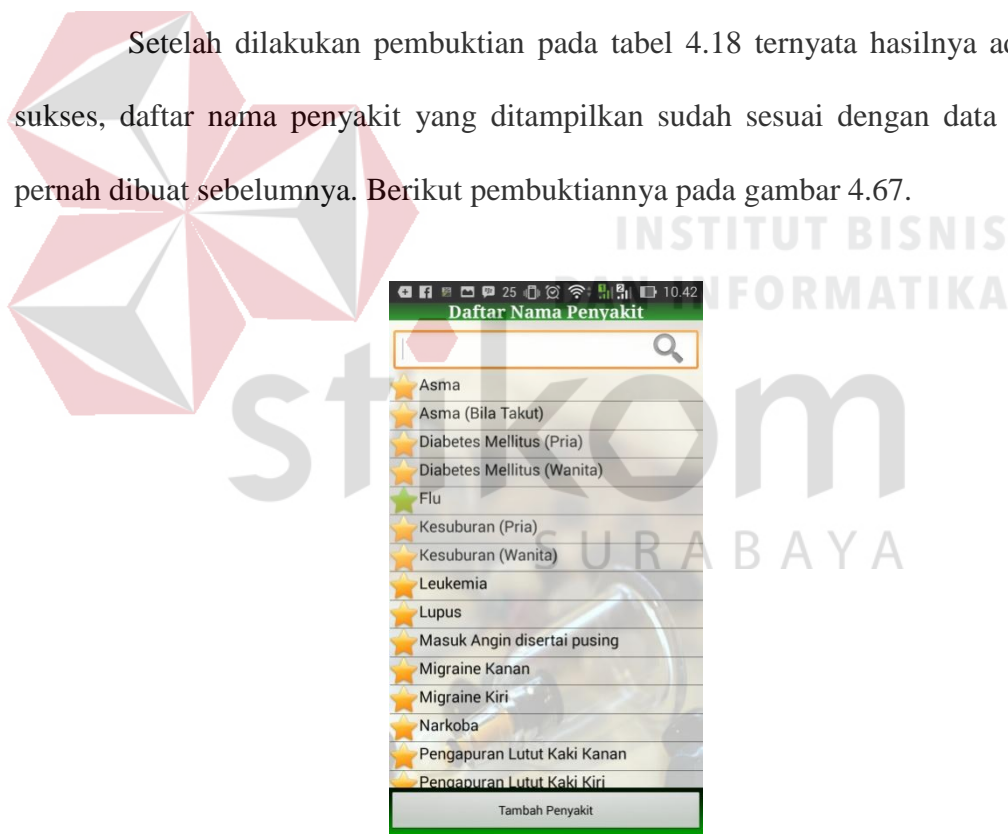
Berikutnya adalah uji menampilkan daftar nama penyakit yang sudah dibuat sebelumnya. Fitur menampilkan daftar nama pasien diuji apakah dapat

menampilkan daftar nama penyakit sesuai dengan data yang pernah diinputkan sebelumnya. Hasil uji coba menampilkan daftar nama penyakit dapat dilihat pada tabel 4.18.

Tabel 4.18 Hasil Uji Coba Menampilkan Daftar Nama Penyakit

Test Case ID	Tujuan	Input	Output Yang Diharapkan	Status
18.	Menampilkan daftar nama penyakit yang sudah tersimpan.	Masuk menu master data penyakit pada menu utama.	Tampil daftar nama penyakit yang terdapat pada <i>database</i> aplikasi.	Sukses

Setelah dilakukan pembuktian pada tabel 4.18 ternyata hasilnya adalah sukses, daftar nama penyakit yang ditampilkan sudah sesuai dengan data yang pernah dibuat sebelumnya. Berikut pembuktiannya pada gambar 4.67.



Gambar 4.67 Hasil Uji Coba Menampilkan Daftar Nama Penyakit

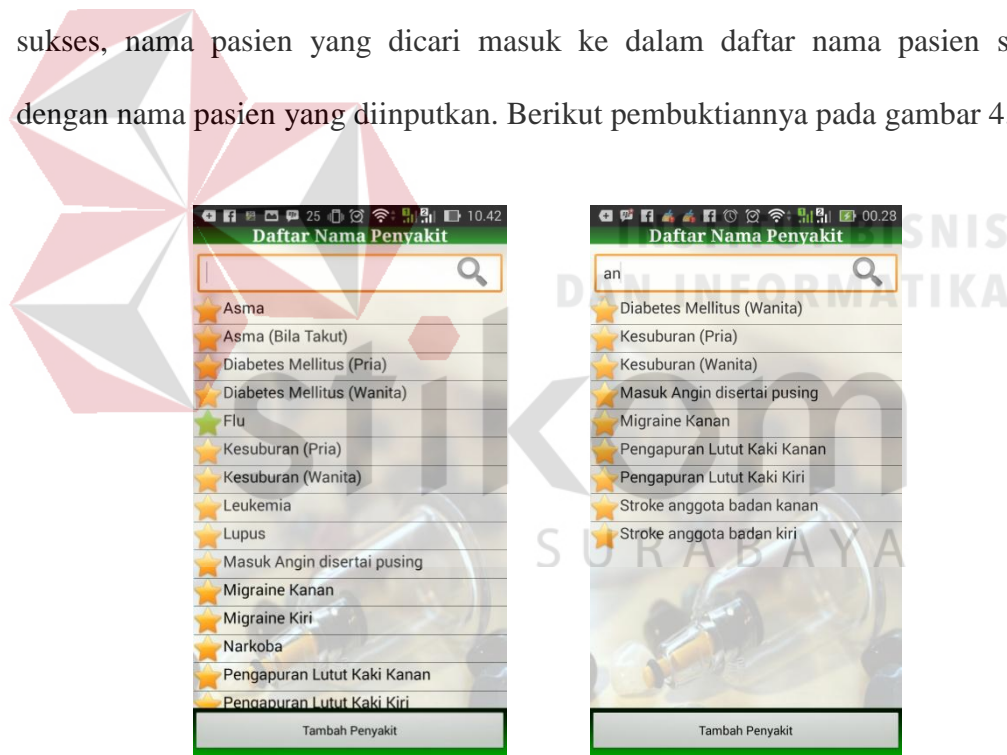
Berikutnya adalah uji pencarian nama penyakit yang sudah dibuat sebelumnya. Fitur pencarian nama penyakit diuji apakah dapat melakukan proses

pencarian sesuai dengan data yang diinputkan. Berikut hasil uji coba pencarian nama penyakit pada tabel 4.19.

Tabel 4.19 Hasil Uji Coba Pencarian Nama Penyakit

Test Case ID	Tujuan	Input	Output Yang Diharapkan	Status
19.	Melakukan pencarian nama penyakit.	Memasukkan nama penyakit yang akan dicari <i>pada textbox search</i> nama penyakit.	Nama penyakit yang dicari tampil pada daftar nama penyakit.	Sukses

Setelah dilakukan pembuktian pada tabel 4.19 ternyata hasilnya adalah sukses, nama pasien yang dicari masuk ke dalam daftar nama pasien sesuai dengan nama pasien yang diinputkan. Berikut pembuktiannya pada gambar 4.68.



Gambar 4.68 Hasil Uji Coba Pencarian Nama Penyakit

Uji berikutnya adalah ketika terapis memilih nama penyakit untuk melakukan perubahan data penyakit maupun melihat detail penyakit. Setelah memilih nama pasien sistem menampilkan *pop-up* berupa pilihan untuk

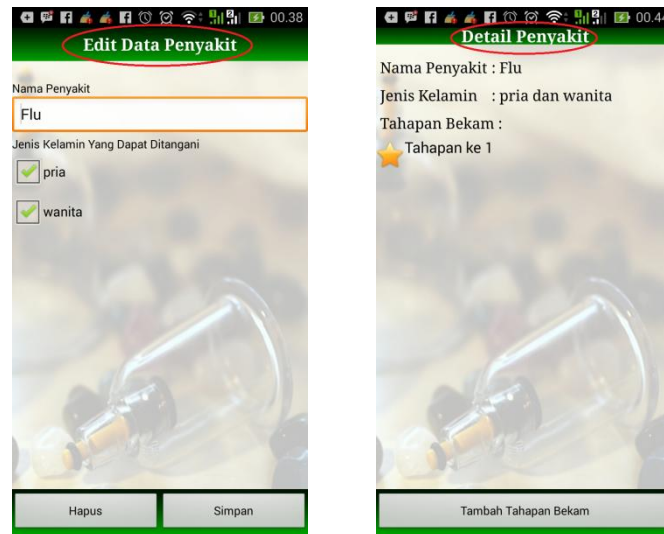
melakukan *edit* data penyakit atau melihat detail. Berikut hasil uji coba masuk tahap pemilihan nama penyakit pada tabel 4.20.

Tabel 4.20 Hasil Uji Coba Masuk Tahap Detail Penyakit

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
20.	Masuk tahap pemilihan nama penyakit.	Pilih nama penyakit	Tampil detail data penyakit berupa nama penyakit yang dipilih, jenis kelamin yang dapat ditangani dan jumlah tahapan pengobatan	Sukses

Setelah dilakukan pembuktian pada tabel 4.20 ternyata hasilnya adalah sukses, terapis dapat masuk ke dalam data penyakit sesuai dengan nama penyakit yang dipilih. Berikut pembuktiannya pada gambar 4.69.





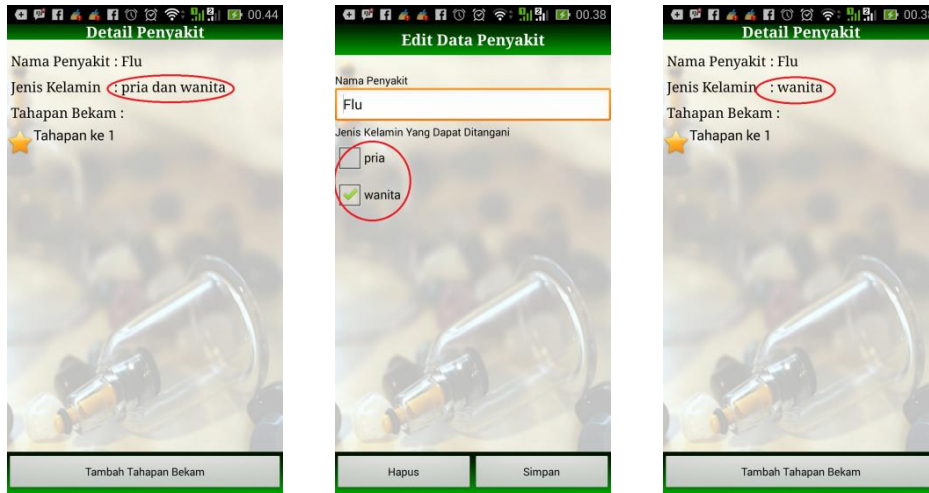
Gambar 4.69 Hasil Uji Coba Masuk Tahap Detail Penyakit

Berikutnya adalah uji edit data penyakit jika terjadi kesalahan dalam pengisian data penyakit yang sudah dibuat sebelumnya. Edit data penyakit yang telah dibuat, diuji apakah dapat melakukan perubahan data sesuai data yang diperlukan oleh terapis. Berikut pada tabel 4.21 hasil uji coba merubah data penyakit.

Tabel 4.21 Hasil Uji Coba Merubah Data Penyakit

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
21.	Merubah data penyakit.	Memilih nama penyakit yang akan dirubah pada daftar nama penyakit sesuai kebutuhan terapis.	Pada detail penyakit sudah berubah sesuai dengan data inputan baru.	Sukses

Setelah dilakukan pembuktian pada tabel 4.21 ternyata hasilnya adalah sukses, data penyakit dapat berubah sesuai kebutuhan terapis sehingga data penyakit berubah sesuai inputan yang baru. Berikut pembuktiannya pada gambar 4.70.



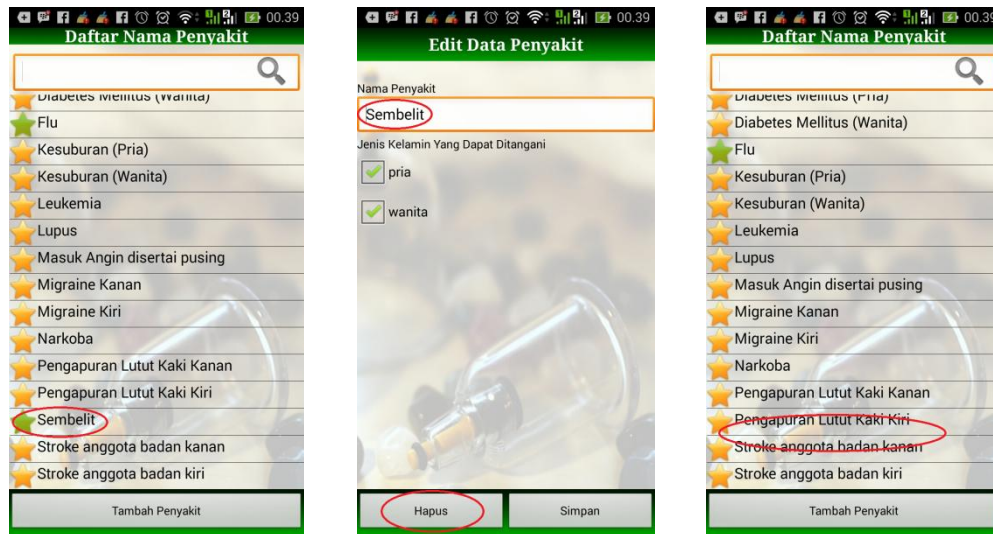
Gambar 4.70 Hasil Uji Coba Merubah Data Penyakit

Berikutnya adalah uji hapus data penyakit jika terjadi kesalahan data atau data tersebut tidak dibutuhkan, maka terapis dapat menghapus data penyakit tersebut. Hapus data penyakit yang telah dibuat, diuji apakah dapat melakukan penghapusan data sesuai data penyakit dari daftar nama penyakit. Berikut pada tabel 4.22 hasil uji coba menghapus data penyakit.

Tabel 4.22 Hasil Uji Coba Menghapus Data Penyakit

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
22.	Menghapus data penyakit.	Memilih nama penyakit pada daftar nama penyakit untuk dihapus.	Nama penyakit sudah terhapus dari daftar nama penyakit.	Sukses

Setelah dilakukan pembuktian pada tabel 4.22 ternyata hasilnya adalah sukses, data penyakit dapat terhapus dari daftar nama penyakit. Berikut pembuktiannya pada gambar 4.71.



Gambar 4.71 Hasil Uji Coba Menghapus Data Penyakit

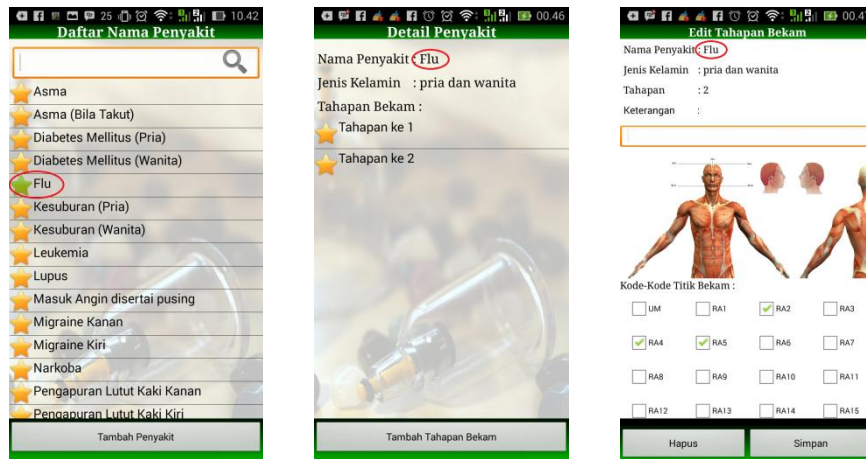
Halaman detail data penyakit berfungsi untuk menampilkan seluruh data tentang penyakit, nantinya data tersebut dapat digunakan untuk melakukan pengobatan bekam terhadap pasien. Desain uji detail data penyakit dapat dilihat pada tabel 4.23 hasil uji coba menampilkan detail data penyakit.

Tabel 4.23 Hasil Uji Coba Menampilkan Detail Data Penyakit

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
23.	Menampilkan detail data penyakit.	Pilih nama penyakit.	Tampil detail data penyakit berupa nama penyakit yang dipilih, jenis kelamin yang dapat ditangani dan jumlah tahapan pengobatan bekam.	Sukses

Setelah dilakukan pembuktian pada tabel 4.23 ternyata hasilnya adalah sukses, terapis dapat melihat detail data penyakit dan dapat mengakses setiap tahapan untuk melihat lokasi titik bekam. Berikut pembuktiannya pada gambar 4.72.





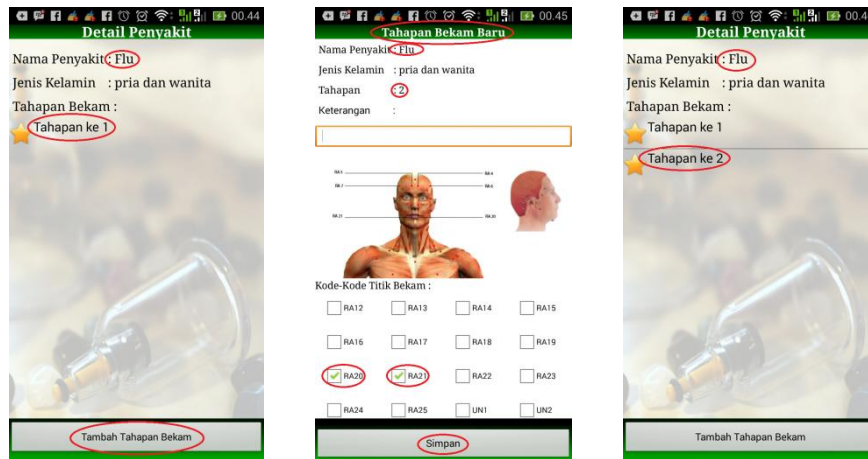
Gambar 4.72 Hasil Uji Coba Menampilkan Detail Data Penyakit

Halaman detail penyakit dapat juga untuk menambahkan tahapan bekam, kemudian data tersebut disimpan kedalam *database* aplikasi untuk mendukung proses pengobatan bekam. Hasil uji coba menambah data tahapan bekam dapat dilihat pada tabel 4.24.

Tabel 4.24 Hasil Uji Coba Menambah Data Tahapan dan Titik Bekam

Test Case ID	Tujuan	Input	Output Yang Diharapkan	Status
24.	Menambahkan data tahapan bekam dan titik bekam.	Pilih menu tambah tahapan bekam kemudian pilih beberapa kode titik bekam dan masukkan keterangan sesuai kebutuhan terapis.	Pada detail penyakit akan tampil tambahan data tahapan pengobatan sesuai data yang diinputkan oleh terapis	Sukses

Setelah dilakukan pembuktian pada tabel 4.24 ternyata hasilnya adalah sukses, penambahan data tahapan dan titik bekam dapat tersimpan kedalam *database* sesuai inputan yang dimasukkan oleh terapis. Berikut pembuktiannya pada gambar 4.73.



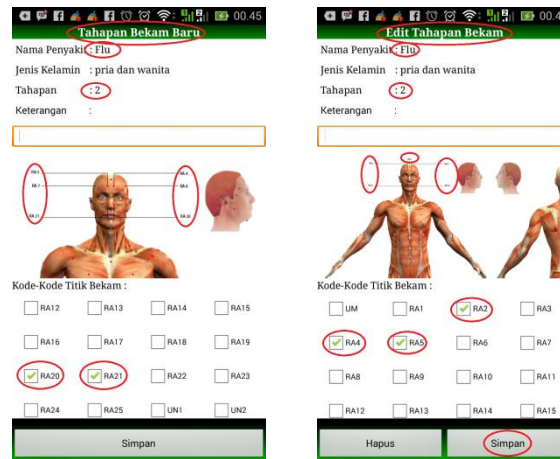
Gambar 4.73 Hasil Uji Coba Menambah Data Tahapan dan Titik Bekam

Berikutnya adalah uji edit data penyakit jika terjadi kesalahan dalam pengisian data tahapan bekam yang sudah dibuat sebelumnya. Edit data penyakit yang telah dibuat, diuji apakah dapat melakukan perubahan data sesuai yang diperlukan oleh terapis. Berikut pada tabel 4.25 hasil uji coba merubah data tahapan bekam.

Tabel 4.25 Hasil Uji Coba Merubah Data Tahapan Bekam

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
25.	Merubah data tahapan bekam.	Memilih fitur tahapan pengobatan yang akan dirubah data titik bekam dan keterangannya.	Lokasi titik bekam dan keterangan berubah sesuai dengan data yang diinputkan terapis.	Sukses

Setelah dilakukan pembuktian pada tabel 4.25 ternyata hasilnya adalah sukses, data tahapan bekam dapat berubah sesuai kebutuhan terapis sehingga data tahapan bekam berubah sesuai inputan yang baru. Berikut pembuktiannya pada gambar 4.74.



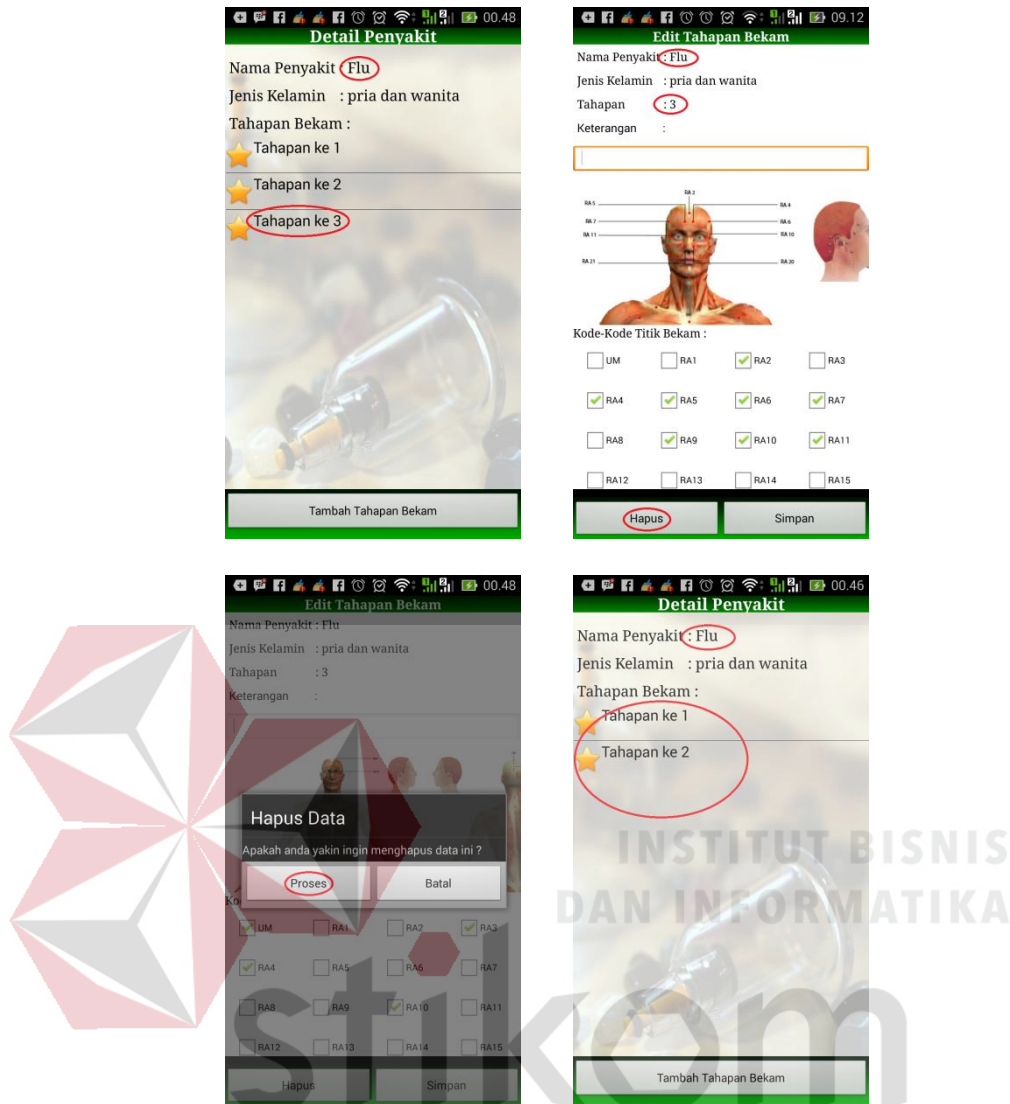
Gambar 4.74 Hasil Uji Coba Merubah Data Tahapan Bekam

Berikutnya adalah uji hapus data tahapan bekam jika terjadi kesalahan data atau data tersebut tidak dibutuhkan, maka terapis dapat menghapus data tahapan bekam tersebut. Hapus data tahapan bekam yang telah dibuat, diuji apakah dapat melakukan penghapusan data tahapan bekam dari detail penyakit. Berikut pada tabel 4.26 hasil uji coba menghapus data tahapan bekam.

Tabel 4.26 Hasil Uji Coba Menghapus Data Tahapan Bekam

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
26.	Menghapus data tahapan bekam.	Memilih tahapan bekam yang akan dihapus pada detail penyakit.	Tahapan yang telah dihapus tidak ada pada detail penyakit.	Sukses

Setelah dilakukan pembuktian pada tabel 4.26 ternyata hasilnya adalah sukses, data tahapan bekam dapat terhapus dari detail penyakit. Berikut pembuktiannya pada gambar 4.75.



Gambar 4.75 Hasil Uji Coba Menghapus Data Tahapan Bekam

### 4.3.2 Hasil Uji Coba Aplikasi Kepada Pengguna

Uji coba berikut yaitu uji coba pada pengguna yang terdiri atas 10 (sepuluh) responden selaku terapis bekam. Kesepuluh responden diberikan kesempatan untuk menjalankan sistem yang telah dikembangkan. Data yang didapat dari kesepuluh responden kemudian digunakan untuk mengevaluasi sistem. Tabel 4.27 adalah hasil dari rekap hasil kuesioner terapis bekam.

Tabel 4.27 Rekap Hasil Kuesioner Terapis Bekam

Pertanyaan	5 (Sangat Baik)	4 (Baik)	3 (Cukup Baik)	2 (Kurang)	1 (Sangat Kurang)	Total	Skor
1		12	18	2		32	64%
2		20	15			35	70%
3		36	3			39	78%
4	5	24	9			38	76%
5		24	12			36	72%
6	5	24	9			38	76%
7	5	12	18			35	70%
8	5	20	12			37	74%
9		28	9			37	74%
10	5	20	12			37	74%
11	10	28	3			41	82%
Rata-Rata Skor							73.88%

Rumus untuk menghitung nilai kolom 'Total' dan 'Skor' adalah sebagai berikut :

$$\text{Total} = \sum (\text{Jumlah Orang} * \text{Bobot Nilai})$$

Jumlah orang disini berarti beberapa responden yang memilih nilai tertentu dari pertanyaan yang diajukan. Sedangkan bobot nilai adalah rentang nilai antara 5 – 1, dimana 5 adalah sangat baik dan 1 adalah sangat kurang. Hasil dari jumlah orang dikali dengan bobot nilai akan menghasilkan nilai total, dimana total tersebut akan digunakan untuk menghitung skor akhir.

$$\text{Nilai Max} = (\text{Bobot Nilai Tertinggi} * \text{Jumlah Orang})$$

Jumlah nilai max disini berarti untuk mencari nilai sempurna dari pertanyaan yang diajukan kepada responden. Hasil dari jumlah bobot nilai tertinggi dikali dengan jumlah orang akan menghasilkan nilai maximal dimana nilai max tersebut akan digunakan untuk menghitung skor akhir.

$$\text{Skor} = (\text{Total/Nilai Max}) * 100 \%$$

Skor digunakan untuk mengetahui sejauh mana sebuah pertanyaan yang diajukan untuk menghasilkan nilai balik antara sangat lemah sampai sangat kuat. Contoh, untuk pertanyaan nomor 1 bagi terapis “ Menurut anda, apakah tampilan *text* pada aplikasi ini jelas dan mudah dibaca? ”, jika skor akhir dari pertanyaan tersebut adalah 64% termasuk kedalam interpretasi baik sesuai dengan tabel kriteria interpretasi skor pada tabel 4.25, yang berarti bahwa rata-rata responden beranggapan bahwa “ Menurut anda, apakah tampilan *text* pada aplikasi ini jelas dan mudah dibaca? ” terbilang baik, sesuai dengan skor akhir yang dihasilkan dari hasil rekap kuesioner.

Tabel 4.28 Kriteria Interpretasi Skor

Jangkaun Skor	Interpretasi
Antara 0% - 20%	Tidak Baik
Antara 21% - 40%	Kurang Baik
Antara 41% - 60%	Sedang
Antara 61% - 80%	Baik
Antara 81% - 100%	Sangat Baik

Tabel 4.28 menjelaskan interpretasi dari tiap nilai pada kolom ‘Skor’ pada tabel 4.27. Berikut adalah kesimpulan dari kuesioner ini:

Tabel 4.29 Kesimpulan Kuesioner Terapis Bekam

No	PERTANYAAN	SKOR	Intepretasi
	<i>Learnability</i>		
1.	Menurut anda, apakah tampilan <i>text</i> pada aplikasi ini jelas dan mudah dibaca?	64%	Baik
2.	Menurut anda, bagaimana tentang posisi tata letak (menu, gambar, <i>text</i> ) pada aplikasi ini?	70%	Baik
3.	Menurut anda, apakah proses berjalannya aplikasi dari layar satu ke layar berikutnya mudah dipahami?	78%	Baik
<b>Total <i>Learnability</i></b>		<b>70,7%</b>	<b>Baik</b>

<i>Efficiency</i>			
4.	Menurut anda, apakah halaman aplikasi ini dapat ditemukan dengan mudah dan cepat?	76%	Baik
5.	Menurut anda, apakah menu-menu yang dipilih dari aplikasi dapat ditemukan dengan mudah dan cepat?	72%	Baik
<b>Total Efficiency</b>		74%	Baik
<i>Memorability</i>			
6.	Menurut anda, apakah aplikasi ini masih mudah diingat ketika anda sudah lama tidak membukanya?	76%	Baik
<b>Total Memorability</b>		76%	Baik
<i>Error</i>			
7.	Menurut anda, apakah di dalam aplikasi ini mampu meminimalisir kesalahan yang dilakukan oleh pengguna?	70%	Baik
8.	Menurut anda, apakah terdapat pesan yang jelas apabila terjadi kesalahan?	74%	Baik
<b>Total Error</b>		72%	Baik
<i>Satisfaction</i>			
9.	Menurut anda, apakah anda akan menggunakan aplikasi ini kembali?	74%	Baik
10.	Menurut anda, adakah penentuan titik bekam yang didapatkan sesuai dengan yang anda cari?	74%	Baik
11.	Menurut anda, apakah aplikasi ini membantu anda dalam pencarian titik bekam?	82%	Sangat Baik
<b>Total Satisfaction</b>		76,7%	Baik

Dari kesimpulan pada tabel 4.29 dapat disimpulkan secara garis besar, rata-rata skor untuk tabel 4.29 pada terapis adalah 73,88% yang berarti responden memiliki interpretasi yang baik untuk sistem yang digunakan berarti sudah berjalan dengan sesuai.

### 4.3.3 Analisa Hasil Uji Coba Sistem

Analisa hasil uji coba dari keseluruhan uji yang dilakukan akan menentukan kelayakan fitur dasar sistem berdasarkan desain yang telah ditetapkan. Dengan melihat pengujian fitur-fitur dasar sistem seperti tampak pada uji coba no. 1 sampai dengan uji coba no. 26 serta evaluasi lapangan yang telah

dilakukan, dapat disimpulkan bahwa sistem telah berjalan dengan baik dan tidak ditemukan kesalahan fungsi dan ditambah hasil indikator keberhasilan yang baik dan hasil kuesioner total sebesar 73,88% berarti sistem sudah berjalan dengan baik. Dengan demikian dapat disimpulkan bahwa sistem yang dibuat sudah sesuai dengan apa yang diinginkan.

