BAB III

ANALISA DAN PERANCANGAN *SOFTWARE*

3.1 Analisa Kebutuhan *Software*

Pada bab ini penulisamenganalisa kebutuhan perangkat lunak yang merupakan langkah pertama dalam perancangan aplikasi contaner.

3.1.1 Identifikasi Masalah

Adapun identifikasi permasalahan pada pembuatan aplikasi ini diuraikan sebagai berikut:

1. Banyaknya perusahaan gudang contaner yang masih menerapkan sistem manual dalam pencatatan nomor contaner.
2. Belum adanya teknologi android untuk memudahkan perusahaan yang bergerak di bidang contaner untuk pencatatan nomor contaner

3.1.2 Rumusan Masalah

Untuk rumusan masalah pada pembuatan skripsi ini sebagai berikut:

1. Bagaimana membangun sistem untuk pencatatan nomor contaner pada perusahaan gudang contaner?
2. Bagaimana membangun sistem yang terintegrasi dengan android untuk pencatatan nomor contaner?

3.1.3 Analisa Kebutuhan

Untuk membuat perancangan aplikasi contaner, penulis menganalisa kebutuhan baik itu berupa hardware dan softwarenya, adapun sebagai berikut :.

1. Komponen *Hardware.*

   Komputer yang digunakan penulis mempunyai spesifikasi sebagai berikut.

   1) Tipe : Intel pentium..

   2) HDD : 500 GB

   3) RAM : 10 GB

   4) *Proccesor* : *Inte1 Core I3 1,6 GHZ*

   5) *Graphic* : *Radeon (tm) HD Graphic 2.0 GHz*

2. Komponen *Software*

   Software yang digunakan penulis pada laptop sebagai berikut.

   1) Eclipse

   Eclipse merupakan tempat kita mebuat projek aplikasi *android* dan ada beberapa device yang yang harus diinstall dieclips diantaranya.

   a) *Android* SDK

   b) *Android* ADT

   2) *Java* JDK

   Java JDK digunakan untuk plugin bahasa pemrograman java.

3.2 Desain

3.2.1 Rancangan Algoritma pada kasus

Adapun penerapan algortima pada kasus yaitu sebagai berikut:

1. Menentukan code contaner

   Contoh: KMA

2. Menglik button search contaner

3.2.2 *SoftwareArchitecture*

1. Pseudocode Algoritma

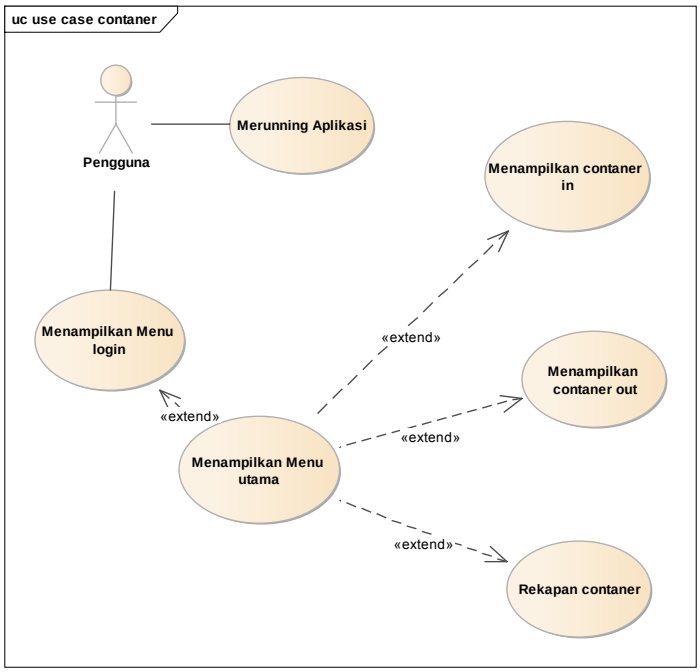
   Pseudo code algoritma aplikasi contaner dapat ditunjukkan sebagai berikut:

```
/** Class filter untuk melakukan filter (pencarian) */
      private class BarangFilter extends Filter{


            @Override
            protected FilterResults performFiltering(CharSequence
constraint) {
                  List<Contaner> filteredData = new
ArrayList<Contaner>();
                  FilterResults result = new FilterResults();
                  String filterString =
constraint.toString().toLowerCase();
                  for(Contaner brg: list){
                        if(brg.getNoContaner()
.toLowerCase().contains(filterString) || brg.getNamaPT()
.toLowerCase().contains(filterString)){
                              filteredData.add(brg);
                        }
                  }
```

```
                result.count = filteredData.size();
                result.values =  filteredData;
                return result;
            }

            @Override
            protected void publishResults(CharSequence constraint,
FilterResults results) {
                filterd = (List<Contaner>) results.values;
                notifyDataSetChanged();
            }
```

2. Pemodelan UML

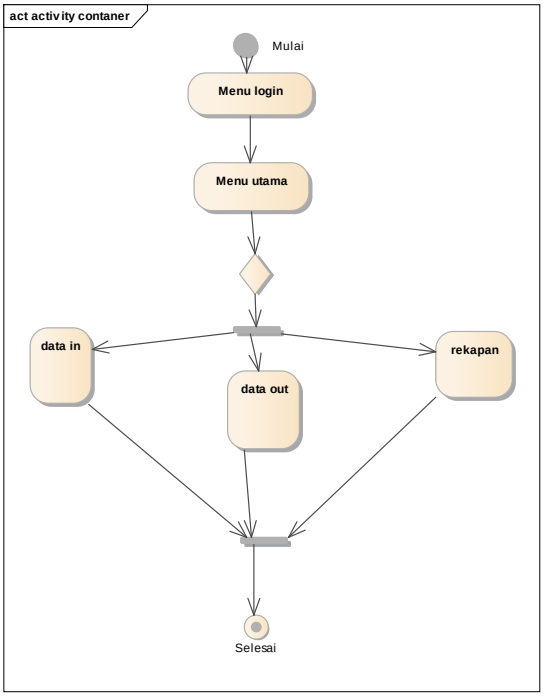
a. Diagram Use Case Aplikasi contaner



Sumber: Pribadi

Gambar III. 1.Diagam Use Case Aplikasi container

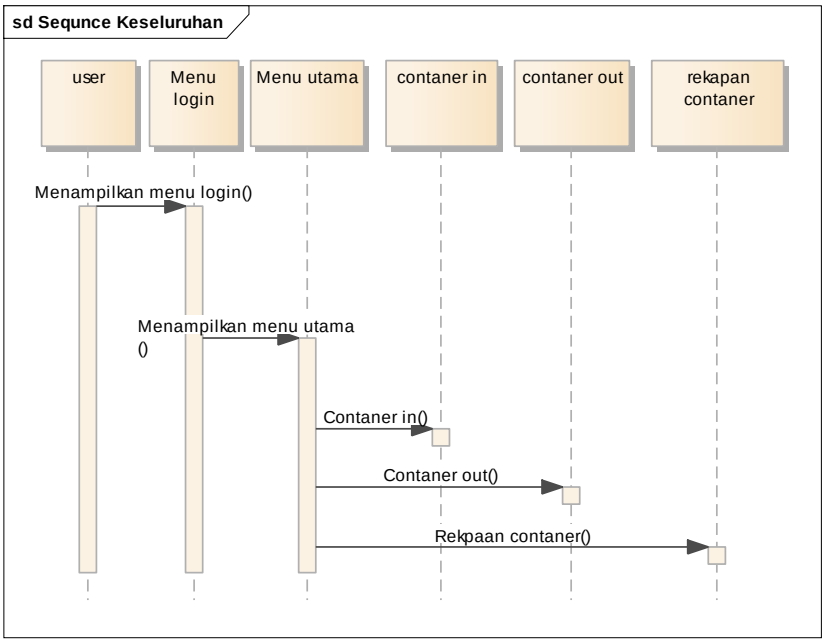
b. Diagram ActivityAplikasi contaner



Sumber: Pribadi

Gambar III. 2. Diagram Activity Aplikasi container

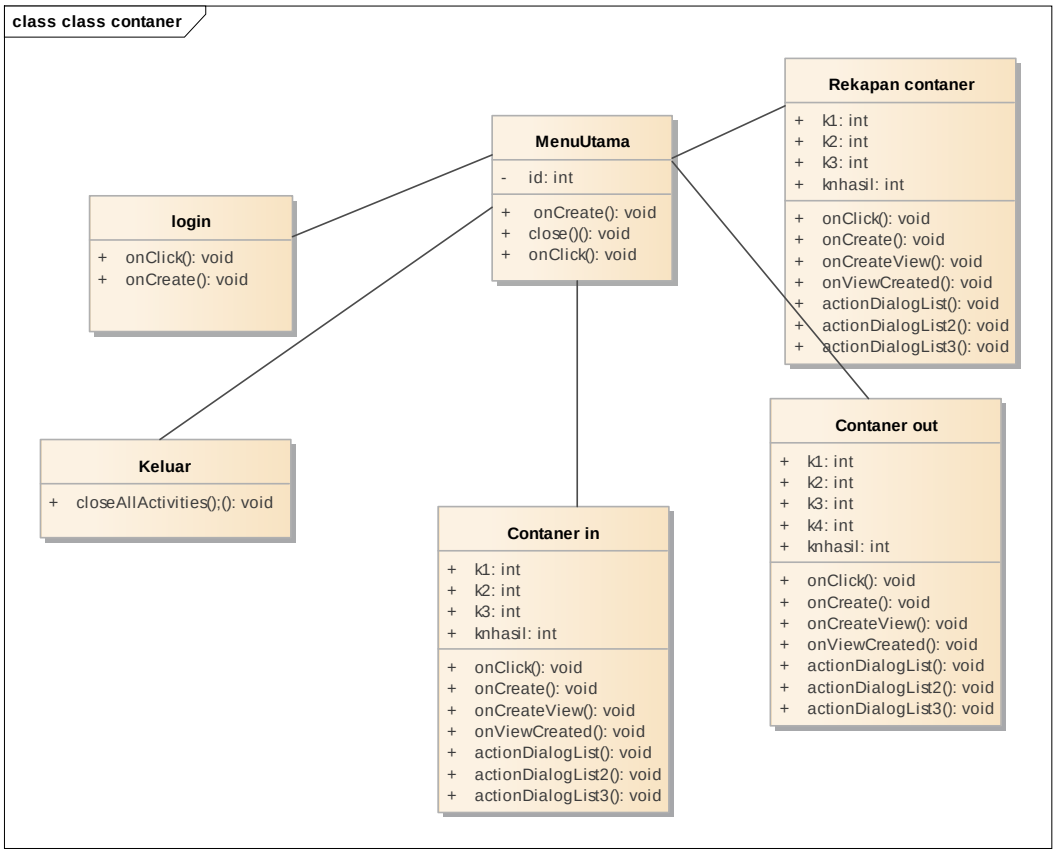
c. Diagram Sequence Keseluruhan



Sumber: Pribadi

Gambar III. 3. Diagram Sequence Keseleluruhan

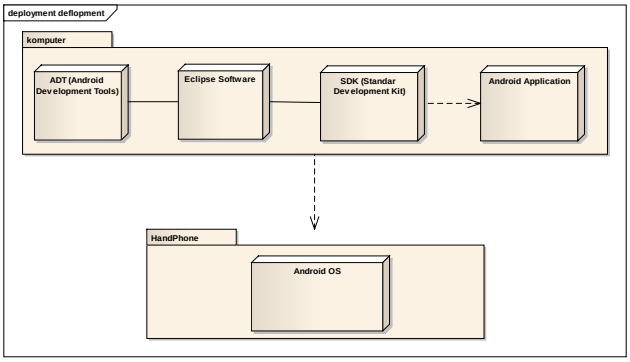
d. Diagram Class Aplikasi contaner



Sumber: Pribadi

Gambar III. 4.Diagram Class Aplikasi container

e. Diagram Deployment



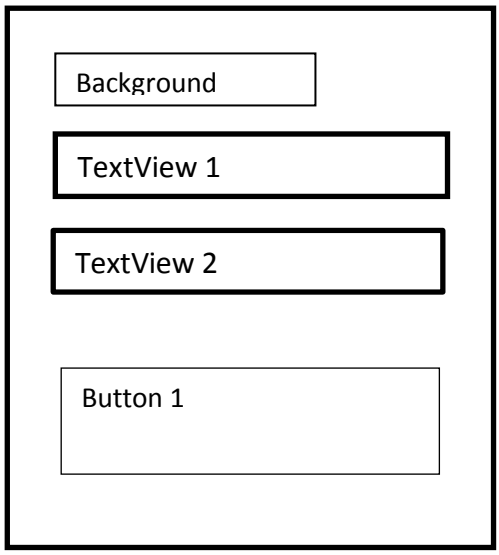
Sumber: pribadi

Gambar III. 5. Diagram Deployment

3.2.3 *User Interface*

Untuk perancangan *User Interface* pada aplikasi contanerbisa dilihat dibawah ini.

1. Menu Awal



Sumber: pribadi

Gambar III. 6. Interface Menu Login

a. Background

   Background menggunakan gambar.jpg.

b. TextView1

   TextView yang bertuliskan “"user”.

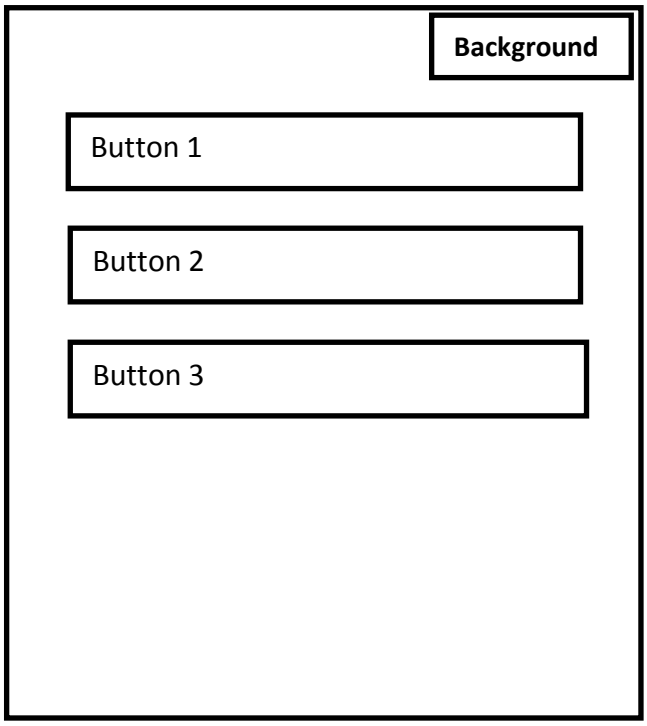
c. TextView2

   TextView yang bertuliskan “"password"”.

d. Button 1

   Button adalah button proses

2. Menu Utama



Sumber: Pribadi

Gambar III. 7. Interface Menu Utama

a. Background

   Background menggunakan background.png

b. Button1

   TextView yang bertuliskan “Contaner in”.

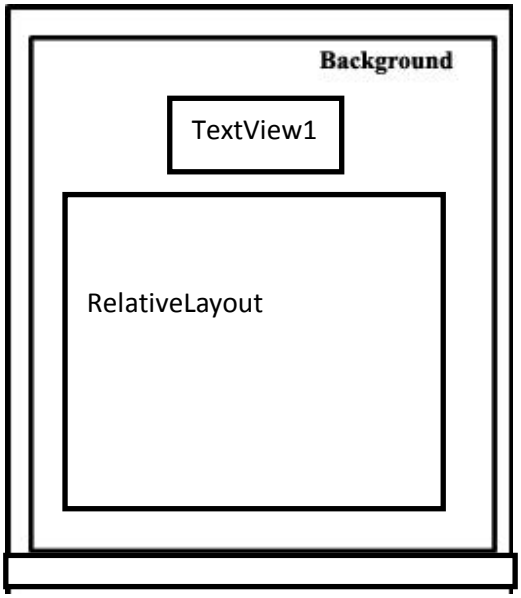
c. Button2

   TextView yang bertuliskan “Contaner out”.

d. Button3

   Button rekapan

3. Menu contaner in



Sumber: pribadi

Gambar III. 8. Menu container in

a. Background

   Background menggunakan gambar.png.

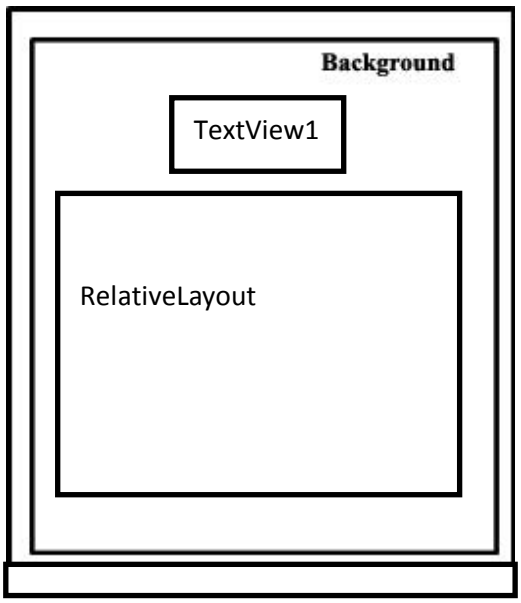
b. TextView1

   TextView yang bertuliskan “Data in”

c. RelativeLayout

   Form input

4. Menu contaner out



Sumber: pribadi

Gambar III. 9. Menu container out

a. Background

   Background menggunakan gambar.png yang berfungsi sebagai penghias latar belakang.

b. TextView1

   TextView yang bertuliskan “Contaner out”.

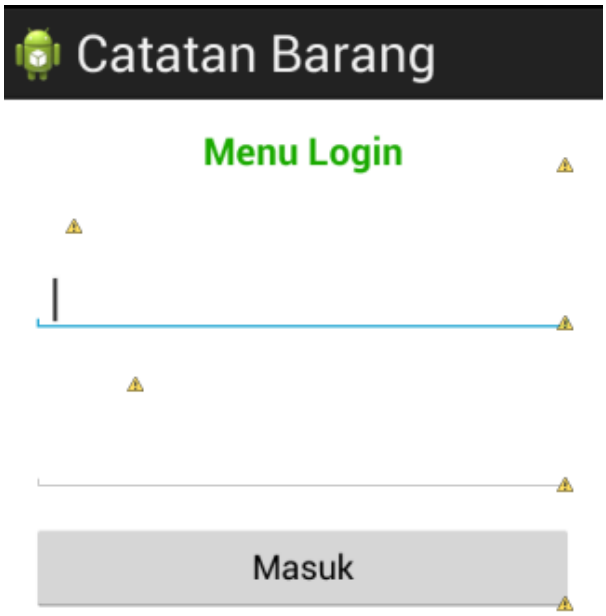
c. RelativeLayout

   Layout yang form contaner out

3.3 Implementasi

Setelah*user interface* dibuat, maka setelah itu penulis mengimplementasikan UI yang dibuat ke dalam tool *eclipse indigo* dengan membuat *layout-layout* sesuai di *user interface.* Berikut tampilan yang sudah diimplementasikan ke dalam *XMLlayout* di *tools eclipse indigo.*

1. Desain Menu login



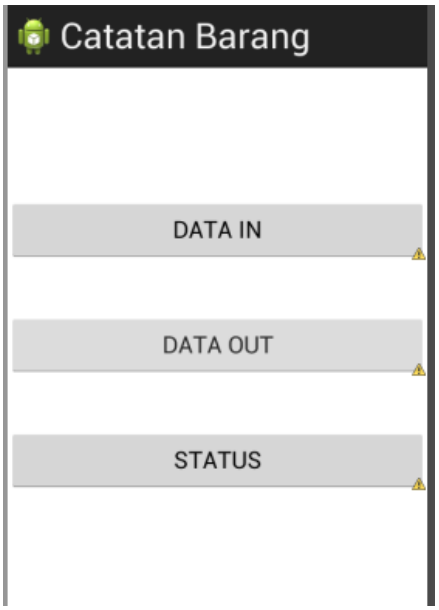
Sumber: pribadi

Gambar III. 10. Desain Menu login

Dengan listing inti XML sebagai berikut:

```
protectedvoid onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.desain_login);
```

2. Desain contaner in



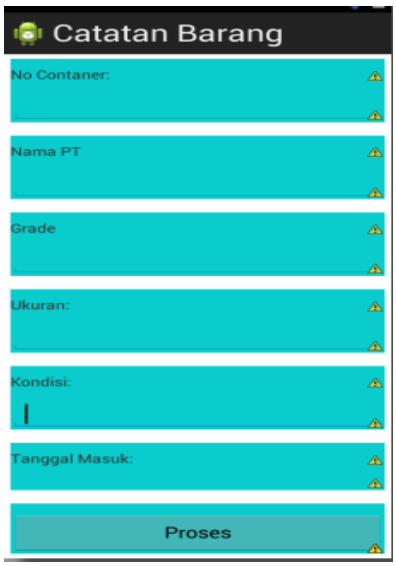
Sumber: pribadi

Gambar III. 11. Desain menu utama

Dengan listing inti XML sebagai berikut:

```
protectedvoid onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.desain_menu_utama);
```

3. Desain contaner in



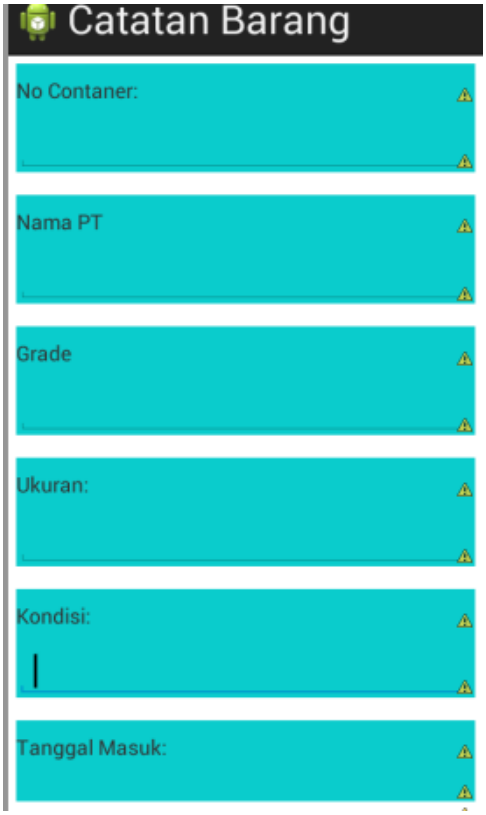
Sumber: pribadi

Gambar III. 12. Desain container in

Dengan listing inti XML sebagai berikut:

```
protectedvoid onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.desain_contaner_in);
```

4. Desain contaner out



Sumber: pribadi

Gambar III. 13. Desain Tabel hasil perhitungan subnetting

Dengan listing inti XML sebagai berikut:

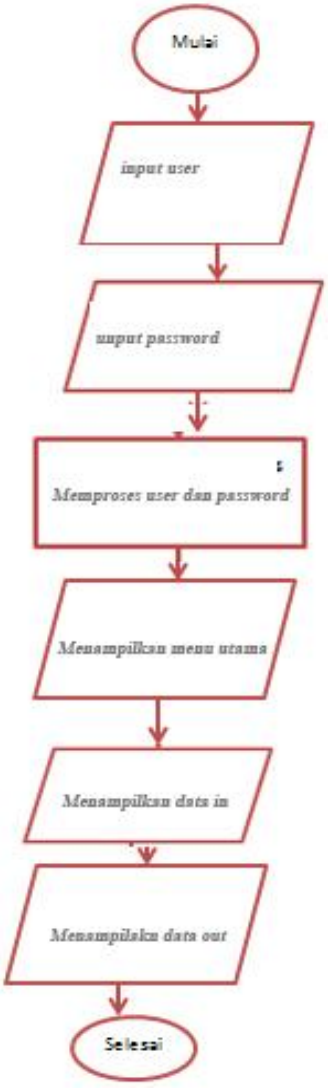
```
protectedvoid onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.desain_contaner_out);
```

3.4 Testing

Adapun untuk Testing menggunakan white box dan black box. Untuk pengujian white box dengan menggunakan skema diagram alir,

3.4.1. Flowchart contaner

Berikut adalah flowchart contaner

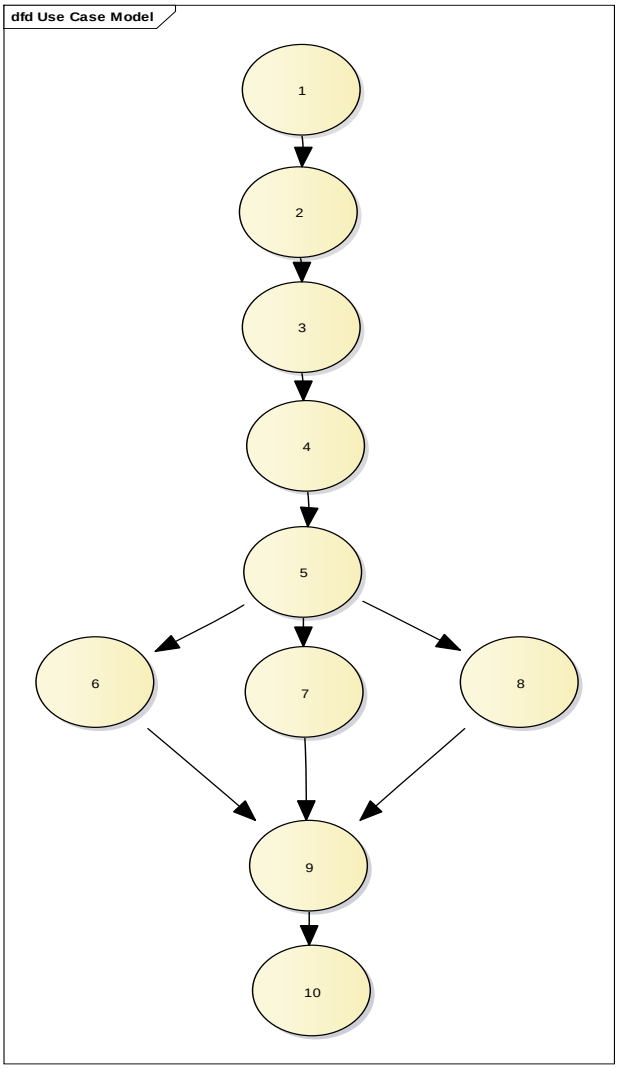


Sumber: pribadi

Gambar III. 14. Flowchart aplikasi contaner

3.4.2. Pengujian White Box

Berikut ini adalah White Box dari aplikasicontaner.



Sumber: pribadi

Gambar III. 15. White Box aplikasi container

Sedangkan script pada aplikasi contaner adalah sebagai berikut.

```
@Override
protectedvoid onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menu_awal);
```

1

.....

```
                  if (json != null) {
                        try {
                              JSONObject jsonObj = new
JSONObject(json);
                              boolean error =
jsonObj.getBoolean("error");
                              // checking for error node in json
                              if (!error) {
                                    // new category created
successfully
                                    isNewCategoryCreated = true;
                              } else {
                                    Log.e("Create Category Error:
", "> " + jsonObj.getString("message"));
                              }

                        } catch (JSONException e) {
                              e.printStackTrace();
                        }

                  } else {
                        Log.e("JSON Data", "Didn't receive any
data from server!");
                  }


      returnnull;.....

..

protected String doInBackground(String... params) {
                  if(params[0] == "delete"){
                        Server_req.sendGetRequest(
```

2

3

```
Koneksi.urlDelete+"?id="+selectedList.getId().toString());
                    }else{
                          /** Mengirimkan request ke server
danmemproses JSON response */
                          String response =
Server_req.sendGetRequest(Koneksi.urlSelectAll);
                          list = processResponse(response);
                    }
                    returnnull;
              }

              @Override
        protectedvoid onPostExecute(String result) {
              progressDialog.dismiss();
              runOnUiThread(new Runnable() {
                          @Override
                          publicvoid run() {
                                populateListView();
                          }
                    });
        }

     }.show();
et1.setText("");
}
}
.................
et4 = (EditText) findViewById(R.id.et4);
et4.setOnKeyListener(new OnKeyListener() {

@Override
publicboolean onKey(View arg0, int arg1, KeyEvent arg2) {
// TODO Auto-generated method stub
if (et4.length() == 3) {
if (Integer.parseInt(et4.getText().toString()) > 255) {
Toast.makeText(getApplicationContext(),
"Input antara 0-255", Toast.LENGTH_SHORT)
.show();
et4.setText("");
} else
bsubnet.requestFocus();
}
returnfalse;
}

});

................................................
```

4

```
@Override
publicvoid onClick(View v) {
privatevoid initView(){
            id_nocontaner = (EditText)
findViewById(R.id.id_nocontaner);
            id_namapt = (EditText) findViewById(R.id.id_namapt);
            id_grade = (EditText) findViewById(R.id.id_grade);
            id_ukuran = (EditText) findViewById(R.id.id_ukuran);
            id_kondisi = (EditText) findViewById(R.id.id_kondisi);

            //
            id_tanggalmasuk = (TextView) findViewById
(R.id.id_tanggalmasuk);



            String id = getIntent().getStringExtra("id");
            String nocontaner =
getIntent().getStringExtra("nocontaner");
            String namapt = getIntent().getStringExtra("namapt");
            String grade =
getIntent().getStringExtra("grade");
            String ukuran =
getIntent().getStringExtra("ukuran");
            String kondisi =
getIntent().getStringExtra("kondisi");
            //
            String tanggalmasuk =
getIntent().getStringExtra("tanggalmasuk");


            id_nocontaner.setText(nocontaner);
            id_namapt.setText(namapt);
            id_grade.setText(grade);
            id_ukuran.setText(ukuran);
            id_kondisi.setText(kondisi);
            //


     id_tanggalmasuk.setText(tanggalmasuk);.....................


};
      ..............................

.................................................

.............................................................
```

5

6

7

8

9

10

}

Setelah dijabarkan diatas, penulis menghitung Kompleksitas siklomatis grafik alir white box, dapatdiperoleh dengan cara sebagai berikut:

V(G) = E – N + 2

Dimana:

E = Jumlah Edge yang ditentukan gambar panah

N = Jumlah simpul grafik alir ditentukan dengan gambar lingkaran

V(G) = 11- 10 + 2 = 3

V(G) < 10 berarti memenuhi syarat kekomplesitasi siklomatisnya.Baris set yang dihasilkan dari jalur independent adalah sebagai berikut:

a. 1-2-3-4-5-6-9-10
b. 1-2-3-4-5-7-9-10
c. 1-2-3-4-5-8-9-10
d. Setelah aplikasi dijalankan, terlihat bahwa satu set baris yang dihasilkan adalah 1-2-3-4-5-6-9-10-1-2-3-4-5-7-9-10-1-2-3-4-5-8-9-10 dan terlihat bahwa simpul telah dieksekusi satu kali.

3.4.3. Pengujian Blackbox

Adapun pengujian blackbox testing pada perancangan aplikasi contanersebagai berikut.

Tabel III. 1. Pengujian Blackboxaplikasi contaner

No.	Skenario Uji	Test Case	Hasil yang diharapkan	Valid
1	Menginput Button contaner in	Klik Button1	Menampilkan form contaner in	Valid
2	Menginput Button contaner out	Klik Button 2	Menampilkan form contaner out	Valid
3	Menginput Button Rekapan	Klik Button 3	Menampilkan form rekapan	Valid
4	Menginput form input contaner	Klik Button proses 4	Terisi ke database	Valid
5	Menginput form output contaner	Klik Button proses 5	Terisi ke database	Valid

Sumber: Pribadi

3.5.Support

Dalam perancangan aplikasi contaner ini dibutuhkan hardware dan software yang support. Adapun spesifikasi hadrware dan software yang support sebagai berikut:

1. Hardware
    a. Spesifikasi Komputer
        1). Prosessor minimum Corei3
        2). RAM 2 GB
        3). VGA Card 512 MB
    b. Spesifikasi Smartphone
        1) Processordual core
        2) RAM 1024 MB
2. Software
    a. Spesifikasi Komputer
        1). Windows 7 32 bit
        2). Windows 8 32 bit
        3). Windows 10 32 bit
    b. Spesifikasi Smartphone
        1). OS Froyo
        2). OS Ginger Beard
        3). OS Jelly Bean 4.2.2