

BAB III

ANALISA DAN PERANCANGAN *SOFTWARE*

3.1 Analisa Kebutuhan *Software* .

3.1.1 Identifikasi Masalah

Identifikasi permasalahan pada perancangan aplikasi elektronika dasar dijabarkan sebagai berikut:

1. Banyak pelajaran baik di sekolah kejuruan dan mata kuliah di universitas yang menjurus ke dalam ilmu elektronika yang membahas tentangnya ilmu elektronika
2. Sulitnya menghafal kode-kode warna pada komponen resistor, kode-kode pada komponen kapasitor dalam menentukan suatu nilai hambatan dan muatan.

3.1.2 Rumusan Masalah

Untuk rumusan masalah pada pembuatan skripsi ini sebagai berikut:

1. Aplikasi ini dibangun untuk membantupelajar mahasiswa jurusan elektronika industri, maupun mahasiswa jurusan teknik komputer dalam pembelajaran elektronika.
2. Menggunakan tools *eclipse indigo* untuk membuat aplikasi berbasis android.

3.1.3 Analisa Kebutuhan

Pada aplikasi pembelajaran elektronika dasar, saya menganalisa kebutuhan baik perangkat keras dan perangkat lunaknya yaitu sebagai berikut :.

1. Perangkat Keras

Komputer yang digunakan memiliki spesifikasi sebagai berikut.

- a. Tipe : ASUS 1015BXO
- b. HDD : 320 GB
- c. RAM : 2 GB
- d. *Proccesor* : *AMD C-60 APU*
- e. *Graphic* : *Radeon™ HD Graphics1.0GHz*

2. Komponen *Software*

Sedangkan perangkat lunak yang dibutuhkan sebagai berikut.

- a. *Windows 7 32 bit*
- b. *Eclipse Indigo*
- c. *Android SDK*
- d. *Android ADT*
- e. *Java JDK*

3.2 Desain

3.2.1 Rancangan Algoritma pada kasus

Rancangan algoritma pada kasus sebagai berikut:

Aplikasi ini, penulis mengambil satu sampel kalkulator saja yaitu kalkulator resistor, yaitu Berikut algoritmanya:

$$\Omega = (N1 + N2) * N3 \pm N4$$

Gambar III. 1. Algoritma Menghitung Resistor

Keterangan :

Ω = Nilai Hambatan (Ω)
 $N1$ = Nilai Pada Gelang 1
 $N2$ = Nilai Pada Gelang 2
 $N3$ = Nilai Pada Gelang 3
 $N4$ = Nilai Pada Gelang 4

Diketahui jika resistor 4 warna memiliki data sebagai berikut

$G1 = 5$, $G2 = 0$, $G3 = 10^2$, $G4 = 5\%$

$$\Omega = (5 + 0) * 10^2 \pm 5\%$$

$$\Omega = (50) * 100 \pm 5\%$$

$$\Omega = 5000 \pm 5\% = 250$$

Nilai resistansi maksimum = $5000 + 250 = 5250 \Omega$

Nilai resistansi minimum = $5000 - 250 = 4750 \Omega$

3.2.2 Software Architecture

1. Pseudocode Algoritma

Untuk Pseudocode algoritmanya dapat ditunjukkan seperti dibawah ini:

```

if (items[arg1].toString().equalsIgnoreCase("hitam"))
{
imageViewgel14dgt.setImageResource(R.drawable.gelreshitam);
n1=0;
HasilMin4dgt.setText(String.valueOf("Berhasil"));
}
elseif (items[arg1].toString().equalsIgnoreCase("cokelat"))
{
imageViewgel14dgt.setImageResource(R.drawable.gelrescokelat);
n1=10;
}
elseif (items[arg1].toString().equalsIgnoreCase("merah"))
{
imageViewgel14dgt.setImageResource(R.drawable.gelresmerah);
n1=20;
}
elseif (items[arg1].toString().equalsIgnoreCase("orange"))
{
imageViewgel14dgt.setImageResource(R.drawable.gelresorange);
n1=30;
}
elseif (items[arg1].toString().equalsIgnoreCase("kuning"))
{
imageViewgel14dgt.setImageResource(R.drawable.gelreskuning);
n1=40;
}
elseif (items[arg1].toString().equalsIgnoreCase("hijau"))
{
//      a=90;
//      hs.setText(String.valueOf(a));
imageViewgel14dgt.setImageResource(R.drawable.gelreshijau);
n1=50;
}
elseif (items[arg1].toString().equalsIgnoreCase("biru"))
{
imageViewgel14dgt.setImageResource(R.drawable.gelresbiru);
n1=60;
}
elseif (items[arg1].toString().equalsIgnoreCase("ungu"))
{
imageViewgel14dgt.setImageResource(R.drawable.gelresungu);
n1=70;
}
elseif (items[arg1].toString().equalsIgnoreCase("abu-abu"))
{

```

```

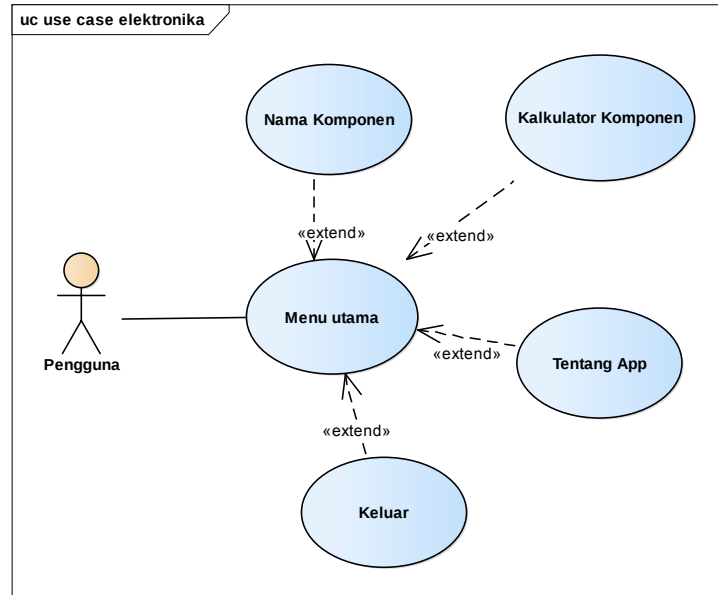
imageViewgel114dgt.setImageResource(R.drawable.gelresabuabu);
n1=80;
}
else
{
imageViewgel114dgt.setImageResource(R.drawable.gelresputih);
n1=90;
}
}
});

AlertDialog alertDialog=builder.create();
alertDialog.show();
}
publicvoid actionDialogList4(){
AlertDialog.Builder builder=new AlertDialog.Builder(context);
builder.setTitle("pilih warna");
builder.setItems(items4, new DialogInterface.OnClickListener() {
@Override
publicvoid onClick(DialogInterface arg0, int arg1) {
// TODO Auto-generated method stub
Toast.makeText(context, items4[arg1], Toast.LENGTH_LONG).show();
if (items4[arg1].toString().equalsIgnoreCase("emas"))
{
imageViewgel144dgt.setImageResource(R.drawable.gelresemas);
n4=0.05;
}
else
{
imageViewgel144dgt.setImageResource(R.drawable.gelresperak);
n4=0.1;
{
}
});
});
AlertDialog alertDialog=builder.create();
alertDialog.show();
}

```

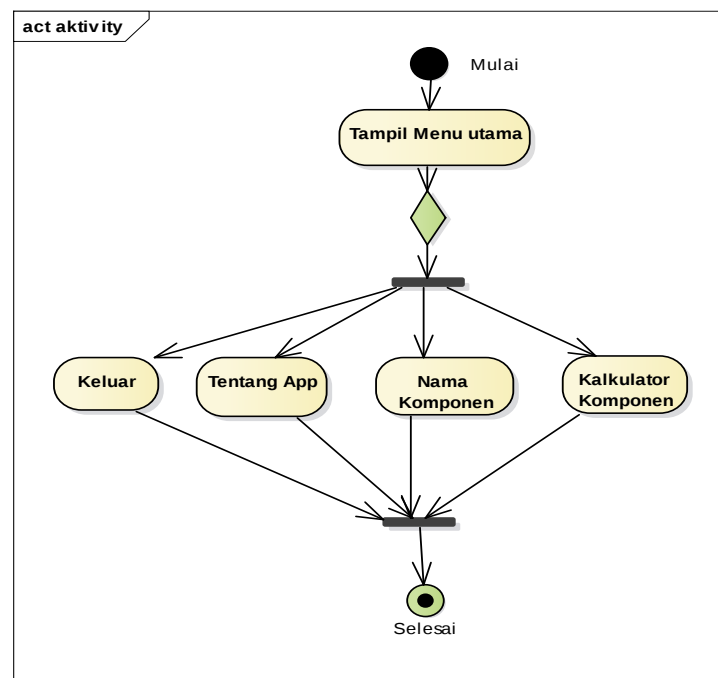
2. Pemodelan UML

a. Diagram Use Case Aplikasi elektronika elektronika dasar



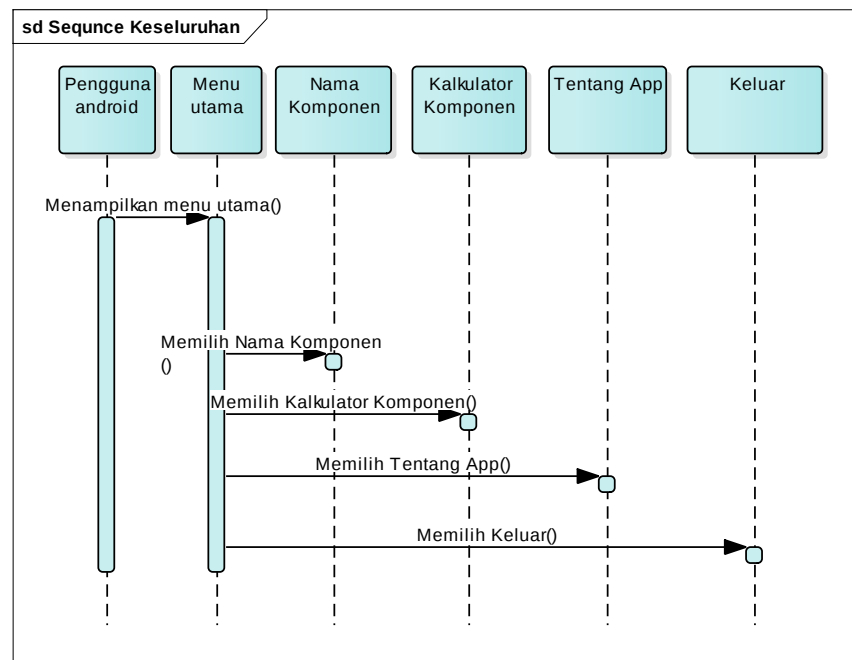
Gambar III. 2. Use Case Aplikasi elektronika elektronika dasar

b. Diagram Activity Pembelajaran Elektronika Dasar



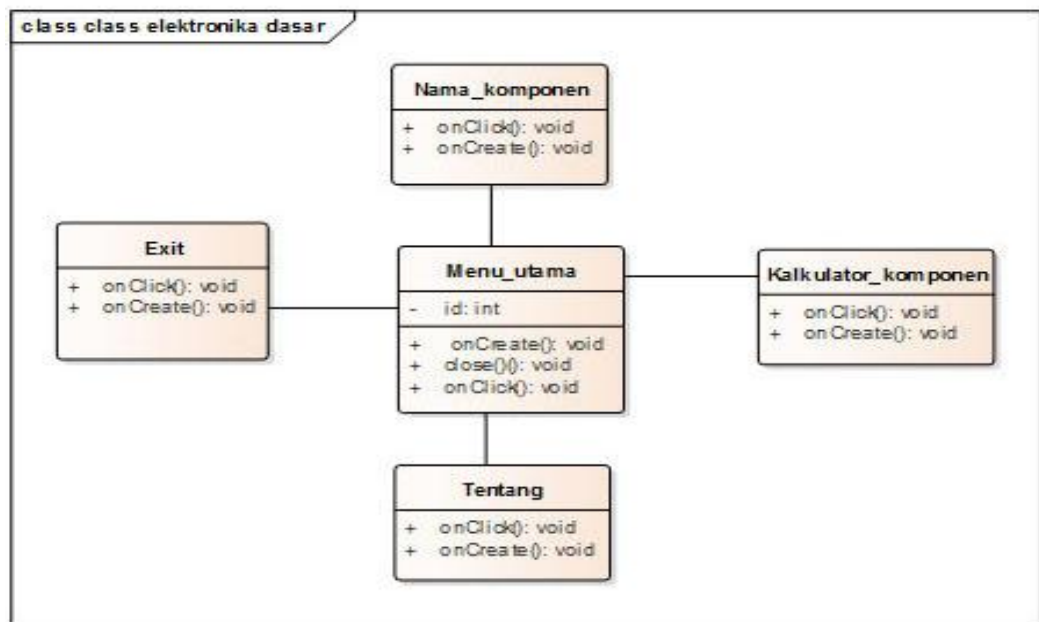
Gambar III. 3. Activity Aplikasi elektronika elektronika dasar

c. Diagram Sequence Keseluruhan



Gambar III. 4. Diagram Sequence Keseleluruhan

d. Diagram Class Aplikasi pembelajaran elektronika dasar

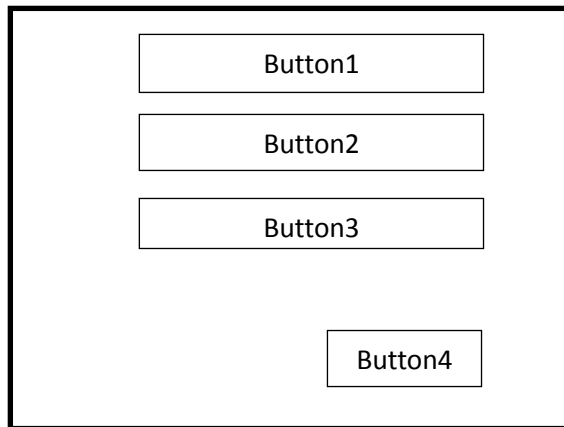


Gambar III. 5. Diagram Class Aplikasi pembelajaran elektronika dasar

3.2.3 User Interface

User Interface pada perancangan aplikasi Aplikasi elektronika elektronika dasarditunjukkan sebagai berikut:

1. Menu Utama



Gambar III. 6. Interface Menu Utama

a. Button1

Button berisi text “Nama Komponen”.

b. Button2

Button berisi text “Kalkulator Komponen”.

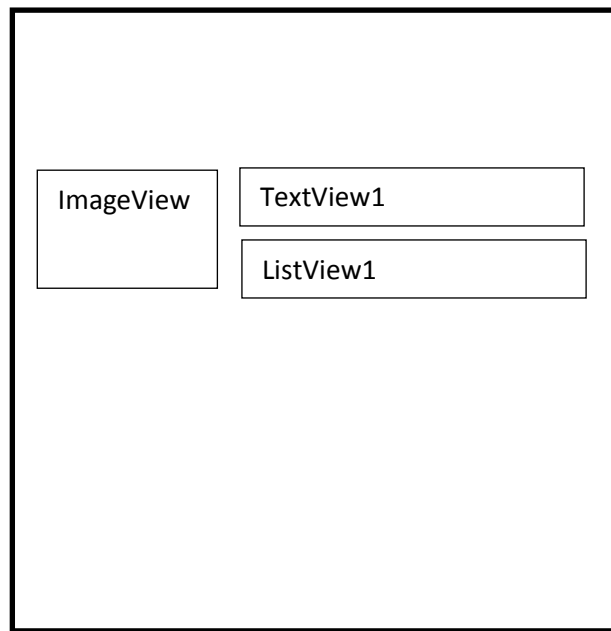
c. Button3

Button berisi text “Tentang App”.

d. Button4

Button berisi text “Keluar”.

2. Menu Nama Komponen



Gambar III. 7. Interface Menu Nama Komponen

a. ImageView

Image yang berisi gambar komponen.

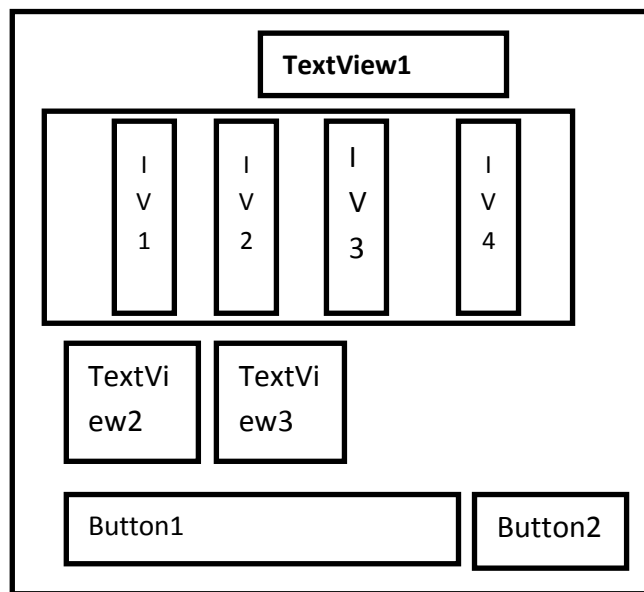
b. TextView

TextView berisi text “Nama Komponen”.

c. ListView

ListView berisi text “Isi Komponen”

3. Menu Kalkulator resistor



Gambar III. 8. Interface Kalkulator Resistor

a. TextView1

TextView1 berisi text “Resistor 4 digit”.

b. ImageView1

ImageView1 yaitu gelang warna 1

c. ImageView2

ImageView2 yaitu gelang warna 2.

d. ImageView3

ImageView3 yaitu gelang warna 3

e. ImageView4

ImageView4 yaitu gelang warna 4.

f. TextView2

TextView2 berisi text “Hasil Minimum”.

g. TextView3

TextView3 berisi text “Hasil Maximum”

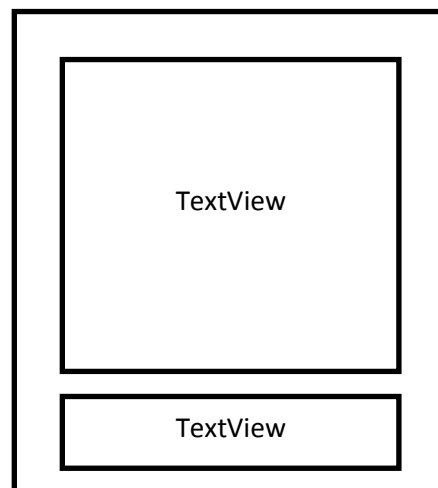
h. Button1

Button1 untuk menghitung nilai hambatan

i. Button2

Button2 untuk menghapus

4. Menu Tentang aplikasi



Gambar III. 9. Menu Tentang

a. TextView1

TextView1 berupa teks tentang aplikasi

b. TextView2

TextView2 berupa teks tentang Nama Pembuat

3.3 Implementasi

Dalam pengimplementasian saya menggunakan tool Eclipse Indigo yang terdapat tools untuk membuat desain. Berikut tampilan yang sudah diimplementasikan di *tools eclipse indigo*.

1. Desain Menu Utama



Gambar III. 10. Desain Menu Utama

2. Desain Kalkulator Komponen



Gambar III.11. Desain Nama Komponen

3. Desain Kalkulator Resistor



Gambar III. 12. Desain Kalkulator Resistor

4. Desain Tentang Aplikasi



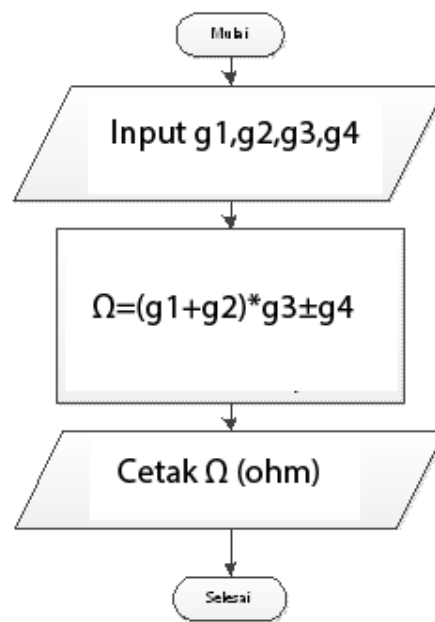
Gambar III. 13. Desain Petunjuk Penggunaan Aplikasi

3.4 Testing

Untuk melakukan pengujian atau *testing*, saya menggunakan menggunakan black box dan white box.

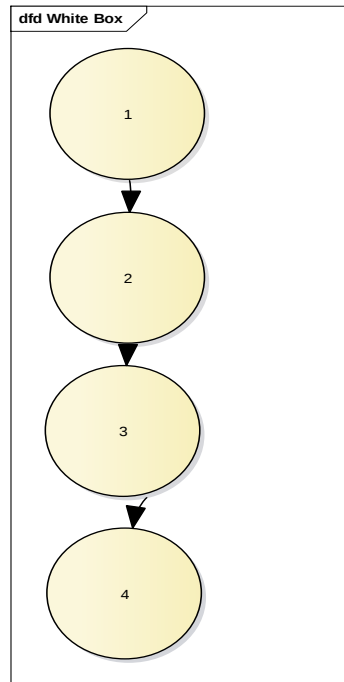
3.4.1. Pengujian White Box Kalkulator Resistor

Dalam pengujian White Box , sebelum diterapkan di white box, maka dibuat flowchartnya terlebih dahulu seperti ini:



Gambar III. 7. FlowChart Kalkulator Resistor

Untuk pengujian White Box seperti dibawah ini sesuai dengan flowchartnya dijelaskan dibawah ini.



Gambar III. 15. White Box Kalkulator Resistor

Adapun script pada kalkulator resistor sebagai berikut.

```

Publicclass Menu_utama extends Exit {
@Override
publicvoid onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
        setContentView(R.layout.design_menu_utama);
        registerBaseActivityReceiver();

@Override
publicvoid onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
        setContentView(R.layout.design_nama_komponen);
        registerBaseActivityReceiver();

publicvoid onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
        setContentView(R.layout.design_kalk_resistor);
        registerBaseActivityReceiver();
publicvoid actionDialogList(){

AlertDialog.Builder builder=new AlertDialog.Builder(context);
builder.setTitle("pilih warna");
builder.setItems(items, new DialogInterface.OnClickListener() {
@Override
publicvoid onClick(DialogInterface arg0, int arg1) {
// TODO Auto-generated method stub
Toast.makeText(context, items[arg1], Toast.LENGTH_LONG).show();
if (items[arg1].toString().equalsIgnoreCase("hitam"))
{
imageViewgel14dgt.setImageResource(R.drawable.gelreshitam);
n1=0;
}
elseif (items[arg1].toString().equalsIgnoreCase("cokelat"))
{
imageViewgel14dgt.setImageResource(R.drawable.gelrescokelat);
n1=10;
}
elseif (items[arg1].toString().equalsIgnoreCase("merah"))
{
imageViewgel14dgt.setImageResource(R.drawable.gelresmerah);
n1=20;
}
elseif (items[arg1].toString().equalsIgnoreCase("orange"))
{
imageViewgel14dgt.setImageResource(R.drawable.gelresorange);
n1=30;
}
elseif (items[arg1].toString().equalsIgnoreCase("kuning"))
{
imageViewgel14dgt.setImageResource(R.drawable.gelreskuning);
n1=40;
}
elseif (items[arg1].toString().equalsIgnoreCase("hijau"))

```

1

2

3

4

```

{
    //      a=90;
    //      hs.setText(String.valueOf(a));
    imageViewgel14dgt.setImageResource(R.drawable.gelreshijau);
    n1=50;
}
elseif (items[arg1].toString().equalsIgnoreCase("biru"))
{
    imageViewgel14dgt.setImageResource(R.drawable.gelresbiru);
    n1=60;
}
elseif (items[arg1].toString().equalsIgnoreCase("ungu"))
{
    imageViewgel14dgt.setImageResource(R.drawable.gelresungu);
    n1=70;
}
elseif (items[arg1].toString().equalsIgnoreCase("abu-abu"))
{
    imageViewgel14dgt.setImageResource(R.drawable.gelresabuabu);
    n1=80;
}
else
{
    imageViewgel14dgt.setImageResource(R.drawable.gelresputih);
    n1=90;
}
});
AlertDialog alertDialog=builder.create();
alertDialog.show();
}

```

Setelah itu saya menghitung Kompleksitas grafik white box, dapat diperoleh

nilainya dengan cara berikut:

$$V(G) = E - N + 2$$

Dimana:

E = Jumlah Edge yang ditentukan gambar panah

N = Jumlah simpul grafik alir ditentukan dengan gambar lingkaran

$$V(G) = 3 - 4 + 2 = 1$$

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

Setelah aplikasi dijalankan tertuju pada kalkulator resistor, terlihat bahwa satu set baris yang dihasilkan adalah 1-2-3-4.

3.4.2. Pengujian Blackbox

Sedangkan pengujian blackbox pada aplikasi kalkulator resistor sebagai berikut.

Tabel III. 1. Pengujian Blackbox Resistor

No.	Skenario Uji	Test Case	Hasil yang diharapkan	Valid
1	Memilih warna gelang1	ActionDia logList1()	Menampilkan list gelang warna1	Valid
2	Memilih warna gelang2	ActionDia logList2()	Menampilkan list gelang warna2	Valid
3	Memilih warna gelang3	ActionDia logList3()	Menampilkan list gelang warna3	Valid
4	Memilih warna gelang4	ActionDia logList4()	Menampilkan list gelang warna4	Valid
5	Mengklik Button “Hitung Hambatan”	Klik Button1	Menampilkan nilai hambatan maksimum dan minimum	Valid
6	Mengklik Button “Hapus”	Klik Button2	Menghapus ActionDialogList1() sampai dengan	Valid

			ActionDialogList4() dan menghapus TextView1 dan TextView2	
--	--	--	---	--

3.5.Support

Pada aplikasi elektronika dasar ini dibutuhkan hardware dan software yang mendukung. Spesifikasi hardware dan software aplikasi elektronika dasar ini yaitu sebagai berikut:

1. Hardware

a. Spesifikasi Komputer

- 1). Processor 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