

BAB III

ANALISA DAN PERANCANGAN *SOFTWARE*

3.1 Analisa Kebutuhan *Software*

Pada bab ini akan disampaikan tahapan-tahapan analisis kebutuhan perangkat lunak yang merupakan langkah awal dalam pembuatan aplikasi pembelajaran perhitungan matriks dan trigonometri.

3.1.1. Identifikasi Masalah

Salah satu proses penelitian yang bisa dikatakan paling penting diantara proses lain yaitu indentifikasi permasalahan. Secara umum masalah penelitian bisa kita temukan lewat studi literature atau lewat pengamatan lapangan (observasi, survey, dsb). Pada penulisan skripsi permasalahan yang akan diteliti dalam pembuatan aplikasi pembelajaran perhitungan matriks dan trigonometri dengan perangkat lunak android berbasis mobile dan tablet yang mampu memberikan informasi tentang bagaimana cara perhitungan rumus matriks dan trigonometri itu sendiri.

3.1.2. Analisa Kebutuhan

Tahapan analisa kebutuhan mencakup hardware, software, aplikasi, dan output yang digunakan antara lain sebagai berikut:

1. Komponen *Hardware*

Komponen *hardware* yang digunakan penulis memiliki *standart* spesifikasi sebagai berikut:

1. Tipe Laptop : Inspiron N4110
2. HDD : 500 GB
3. RAM : 4 GB
4. *Proccesor* : *Core i3-2310M*
5. *Graphic* : *HD Grafis*

Dalam pembuatan aplikasi *android* minimal *Dual core* karena akan berpengaruh terhadap pembuatan *virtual android* yang akan digunakan.

2. komponen *software*

Komponen perangkat lunak yang digunakan dalam pembuatan sisitem tersebut antara lain:

a. *Eclipse*

Eclipse merupakan tempat untuk melakukan pembuatan proyek *android* dan ada beberapa *device* yang harus di install di *eclipse* diantaranya:

- 1) *Android SDK*
- 2) *Android ADT*

b. *Java JDK*

Java JDK digunakan untuk *plugin* bahasa pemograman *java*.

c. *Adobe PhotoShop 11.0*

Software yang digunakan dalam mendesain tampilan berupa gambar gambar yang akan digunakan dalam program *android*.

3. Aplikasi

Aplikasi yang digunakan merupakan aplikasi yang berbasis *android* sehingga program tersebut dapat digunakan untuk menjalankan fungsinya.

4. *input/ouput*

Input/output yang digunakan antara lain input penggunaan dari *interface android* itu sendiri yang menghasilkan *output* berdasarkan dari *input* yang dimasukan.

3.2 Desain

3.2.1. Rancangan Algoritma

Seperti yang dijelaskan pada bab II, bahwa algoritma yang digunakan adalah algoritma dari rumus matriks dan trigonometri itu sendiri. Dalam perhitungan rumus perbandingan trigonometri pada segitiga siku-siku memiliki penjelasan dan beberapa rumus dalam penyelesaiannya.

Sebagai contoh perhitungan Sin pada segitiga siku- siku:

Algoritma : { dibaca sisi samping (a) , sisi depan (b) dan, sisi miring (c). Tentukan nilai Sin pada segitiga siku-siku. Sin dihitung dengan rumusnya $\text{Sin} = b/c$ (sisi depan dibagi sisi miring). Nilai Sin dicetak di peiranti lunak}.

Deskripsinya dalam pemograman Android:

```
sin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        inper_sisi_samping=Double.parseDouble(sisi_samping.getText().toString());
        inper_sisi_depan=Double.parseDouble(sisi_depan.getText().toString());
        inper_sisi_miring=Double.parseDouble(sisi_miring.getText().toString());

        outper_Hasil=inper_sisi_depan/inper_sisi_miring;

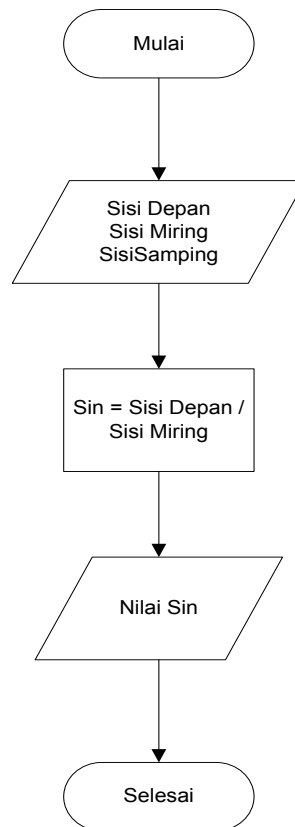
        Hasil.setText(String.valueOf(outper_Hasil));
    }
});
```

```

    });
}

```

Dengan Flowchart:



Gambar III. 1 Diagram Flowchart Perhitungan Sin

Adapun dalam perhitungan matriks memiliki beberapa jenis perhitungan

dan rumus. Sebagai contoh perhitungan matriks determinant sebagai berikut :

Algoritma : { input nilai “A”, “B”, ”C”, dan “D”. Hitung nilai determinant $(A \cdot D) - (B \cdot C)$. Nilai determinant diparanti lunak}.

Deskripsinya dalam pemograman Android:

```

Determinant.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        inper_A=Double.parseDouble(nilai_A.getText().toString());
    }
});

```

```

inper_B=Double.parseDouble(nilai_B.getText().toString());
inper_C=Double.parseDouble(nilai_C.getText().toString());
inper_D=Double.parseDouble(nilai_D.getText().toString());

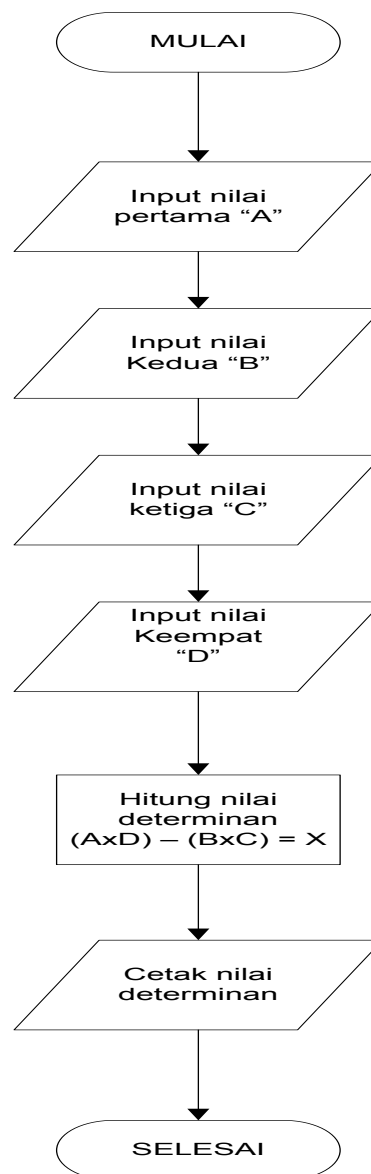
outper_Hasil=(A*D)-(B*C);

Hasil.setText(String.valueOf(outper_Hasil));

    }
});

```

Dengan Flowchart:



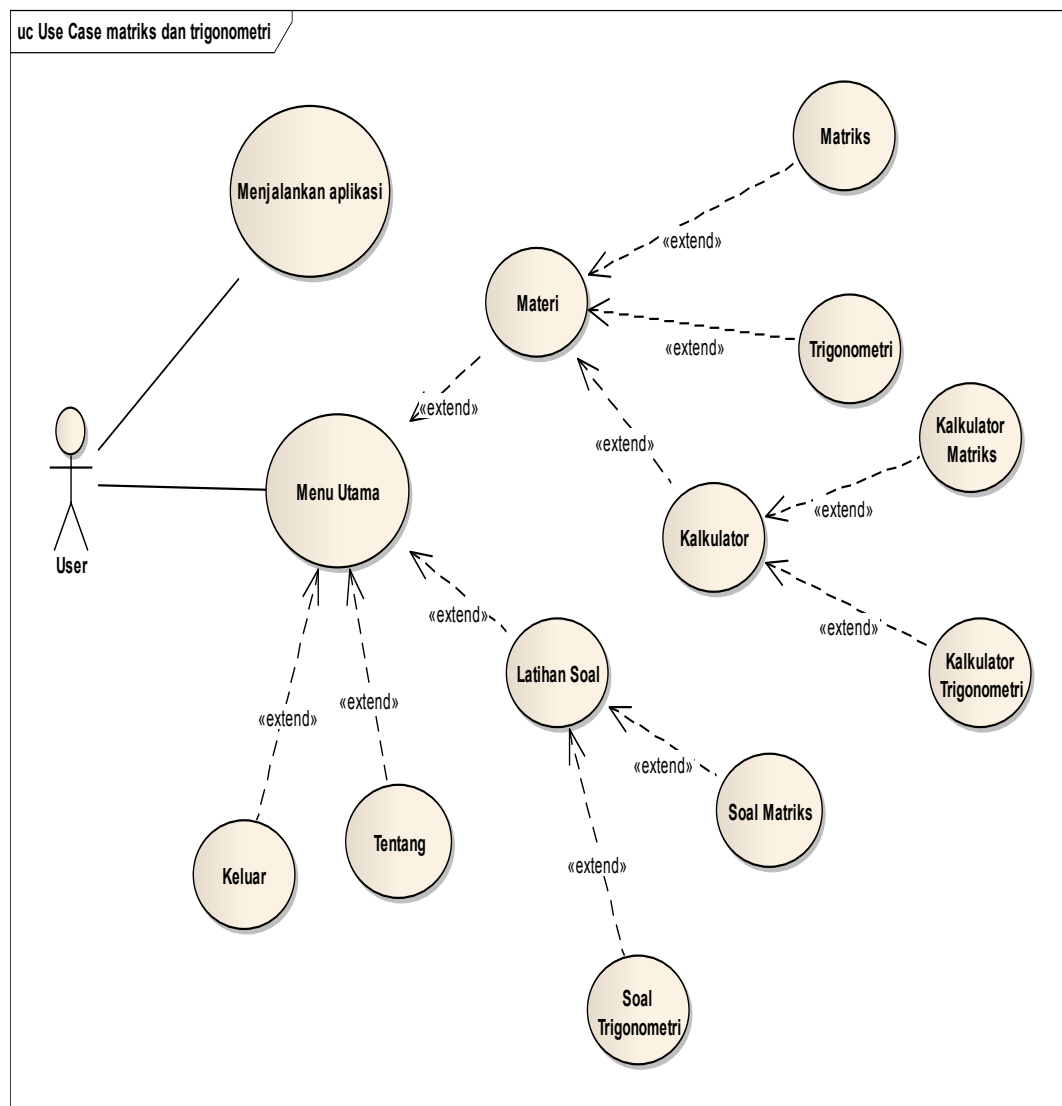
Gambar III. 2 Diagram Flowchart Contoh Perhitungan Determinan

3.2.2. Software Arcithecture

Rekayasa perangkat lunak merupakan suatu disiplin ilmu yang membahas semua aspek produk perangkat lunak, mulai dari tahapan awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, desain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan.

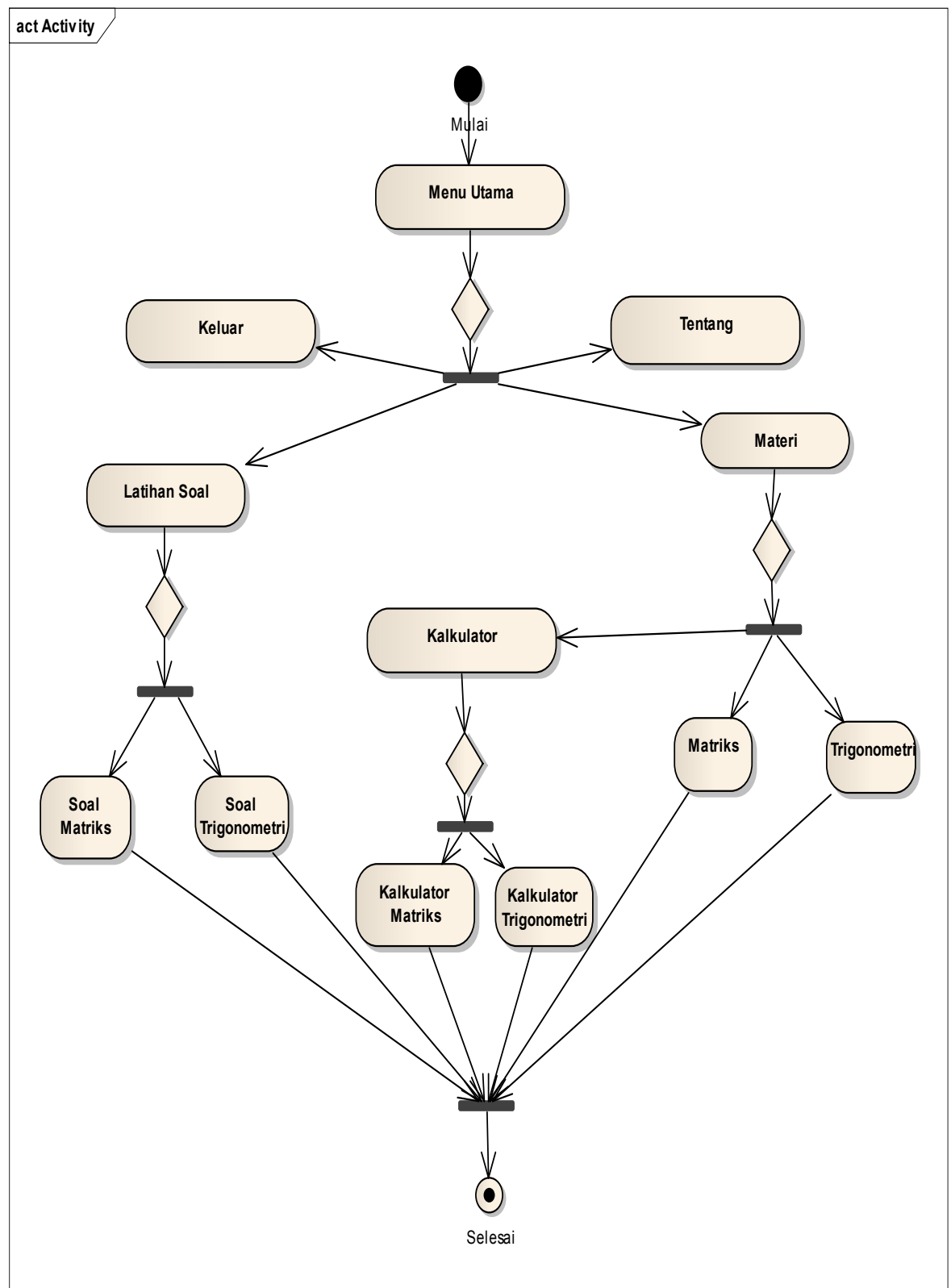
1. Diagram use case

Use Case merupakan pemodelan untuk kegiatan sistem informasi yang akan dibuat.



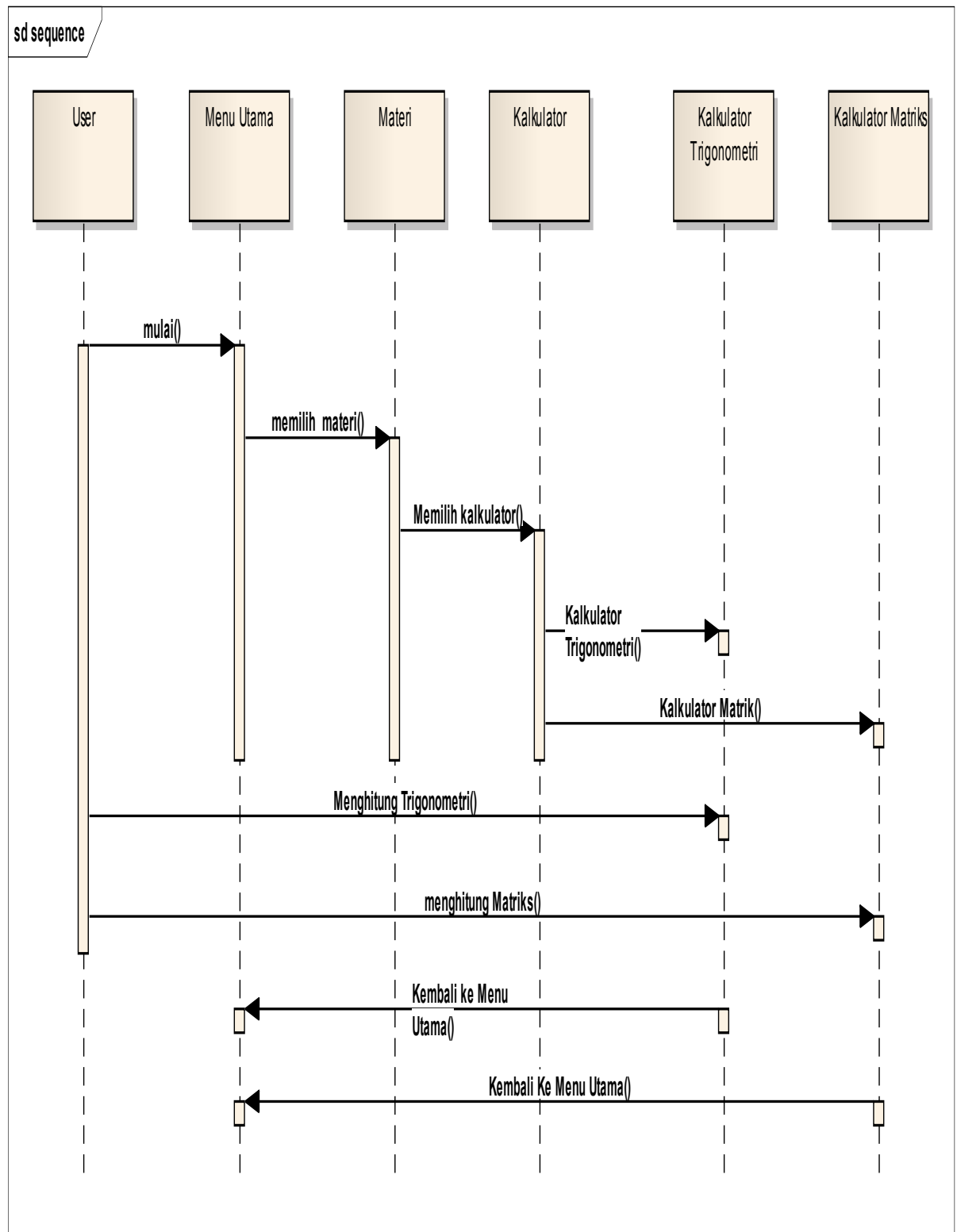
Gambar III. 3 Diagram Use Case Aplikasi Matriks dan Trigonometri

2. Diagram Activity Aplikasi Matriks dan Trigonometri



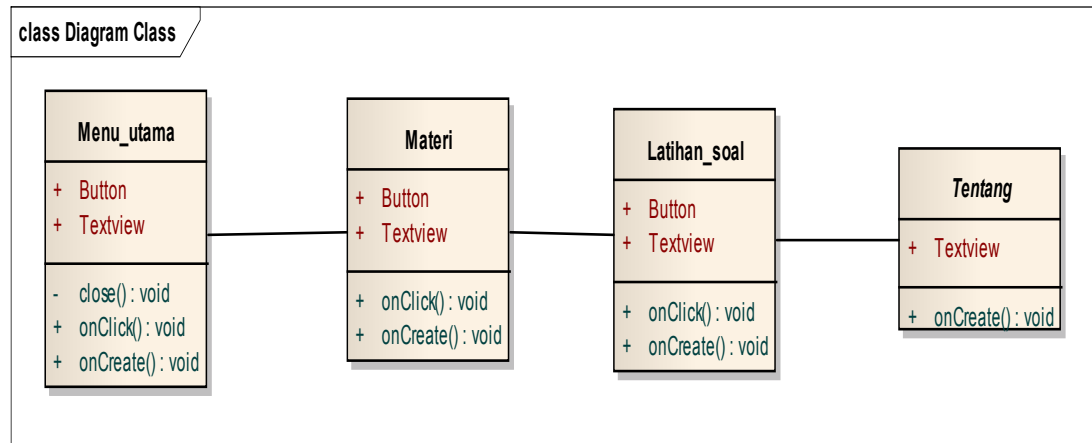
Gambar III. 4 Diagram Activity Aplikasi Matriks Dan Trigonometri

3. Diagram Sequence kalkulator Matriks Dan trigonometri



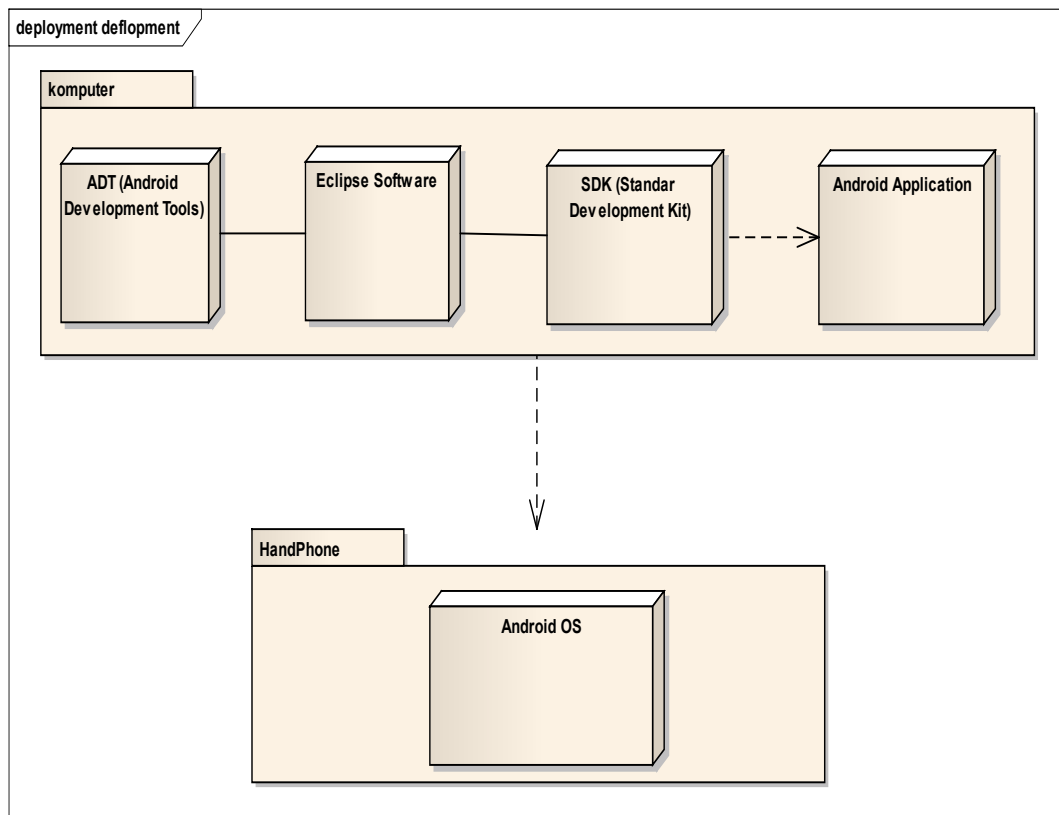
Gambar III. 5 Diagram Sequence Kalkulator Matriks Dan Trigonometri

4. Diagram Class Aplikasi Matriks Dan Trigonometri



Gambar III. 6 Diagram Class Aplikasi Matriks Dan Trigonometr

5. Diagram Deployment



Gambar III. 7 Diagram Deployment

3.2.3. User Interface

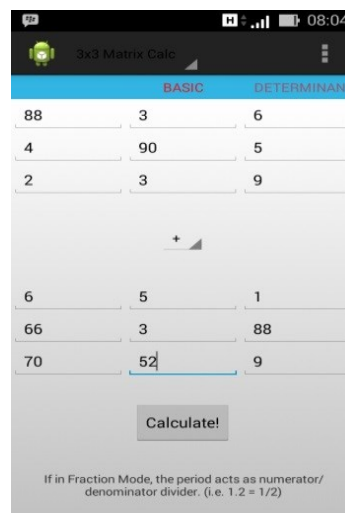
Arsitek *User Interface* pada aplikasi android ini terdiri dari beberapa widget pada layout android yang diberi nama XML file. XML file sendiri berada pada lokasi *res/layout/filename.xml*. Adapun dalam aplikasi ini menggunakan interface sebagai berikut

1. Menu Utama



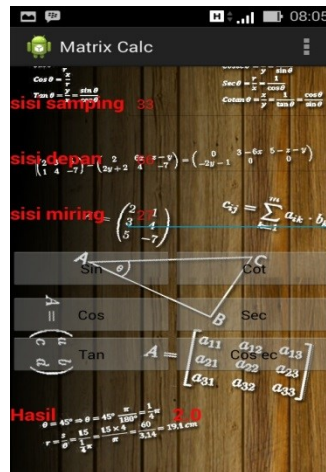
Gambar III. 8 Interface Menu Utama

2. Tampilan Kalkulator Matriks



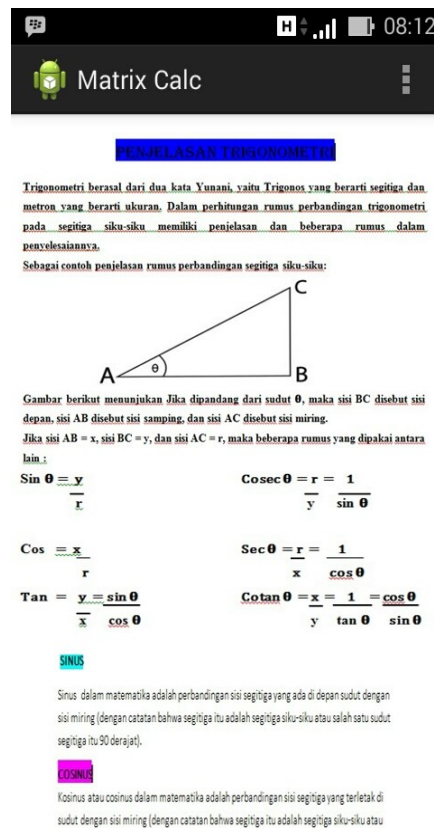
Gambar III. 9 Interface Menu Kalkulator Matriks

3. Tampilan Kalkulator Trigonometri



Gambar III. 10 Interface Menu Kalkulator Trigonometri

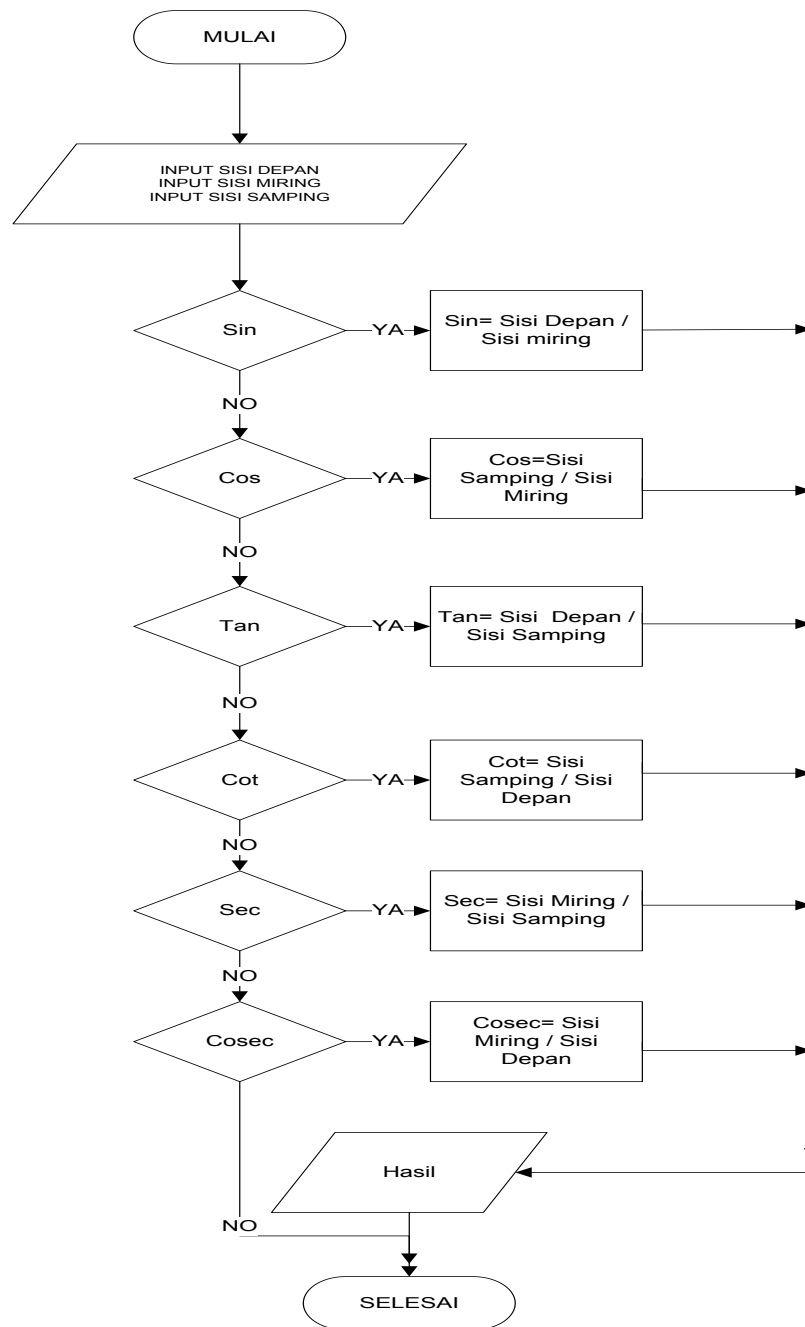
4. Tampilan Materi



Gambar III.11 Interface Tampilan Materi

3.3. Implementasi

Flowchart perhitungan matrik dan trigonometri menggunakan bahasa pemograman java yang merupakan bahasa pemograman dasar dari aplikasi perhitungan matriks dan trigonometri, berikut adalah flowchart dari aplikasi perhitungan matriks dan trigonometri.

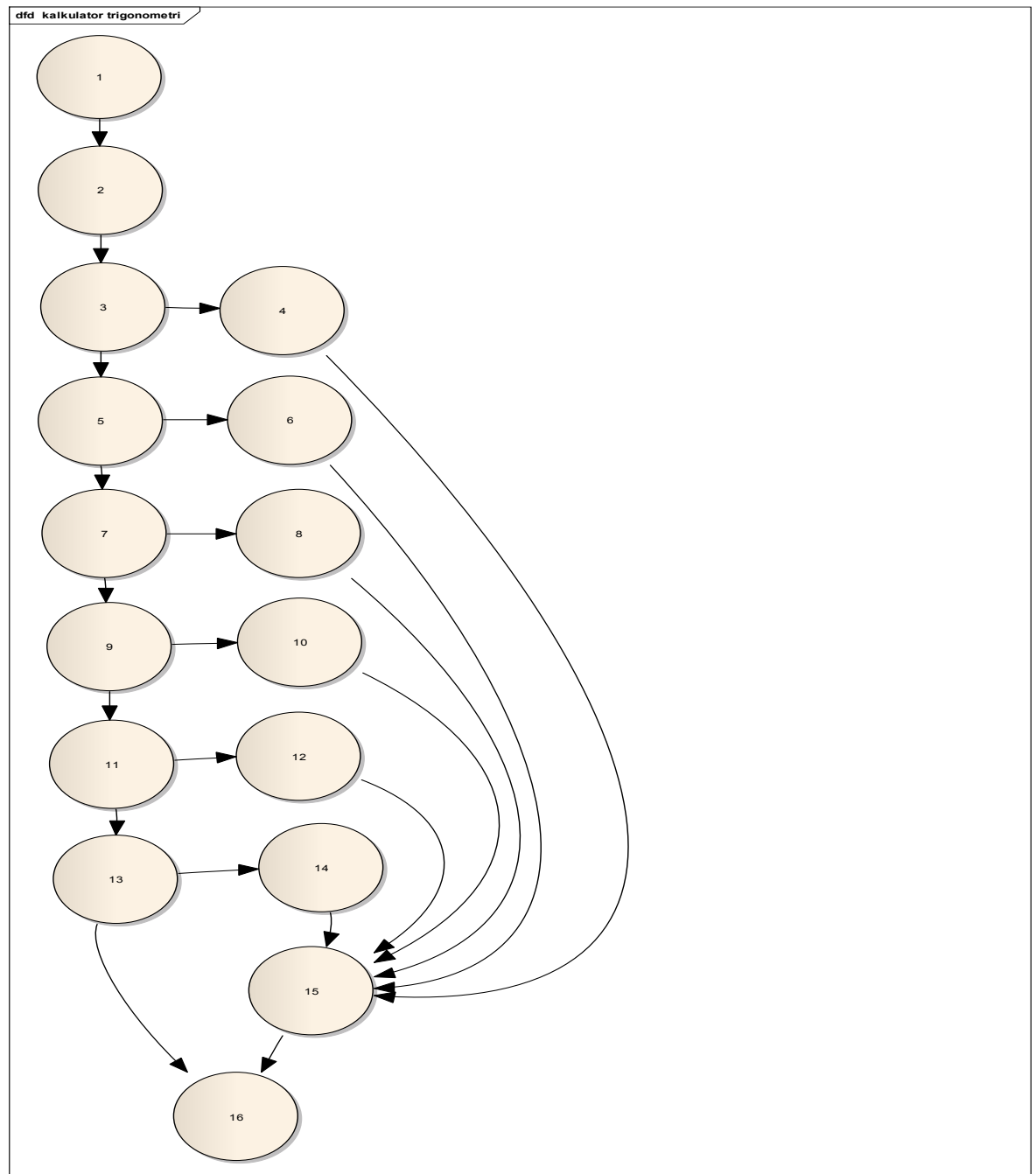


Gambar III. 14 Flowchart Kalkulator Trigonometri

3.4. Testing

Testing menggunakan *white box* dan *black box* untuk pengujian *white box* dengan menggunakan skema diagram alir, berikut ini merupakan diagram alir dari aplikasi pembelajaran perhitungan matriks dan trigonometri berbasis android.

1. White Box Testing



Gambar III. 15 Diagram alir pengujian white box

Scrip perhitungan pada kalkulator Trigonometri

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.layout_trigonometri_kalkulator);
    registerBaseActivityReceiver()
```

1

```
sisi_samping = (EditText) findViewById(R.id.editText1);
sisi_depan = (EditText) findViewById(R.id.editText2)
sisi_miring = (EditText) findViewById(R.id.editText3);
```

2

```
sin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
```

3

```
outper_Hasil=inper_sisi_depan/inper_sisi_miring;
```

4

```
cos.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
```

5

```
outper_Hasil=inper_sisi_samping/inper_sisi_miring;
```

6

```
Tan.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
```

7

```
outper_Hasil=inper_sisi_depan/inper_sisi_samping;
```

8

```
Cot.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
```

9

```
outper_Hasil=inper_sisi_samping/inper_sisi_depan;
```

10

```
sec.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
```

11

```
outper_Hasil=inper_sisi_miring/inper_sisi_samping;
```

12

```
cos ec.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
```

13

```
outper_Hasil=inper_sisi_miring/inper_sisi_samping;
```

14

```
Hasil.setText(String.valueOf(outper_Hasil));
```

15

```
protected void closeAllActivities(){
sendBroadcast(new Intent(FINISH_ALL_ACTIVITIES_ACTIVITY_ACTION));
}
```

16

Kompleksitas sikomatis dari garfik alir white box dapat diperoleh dengan perhitungan:

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

Dimana:

$V(G)$ = Jumlah langkah yang di hasilkan

E = Jumlah Edge yang ditentukan gambar panah

N = Jumlah simpul grafik alir di tentukan dengan gambar lingkaran

Sehingga didapat :

$$V(G) = 21 - 16 + 2 = 7$$

$V(G) = < 10$ berarti memenuhi syarat kekompleksitasi siklomatisnya.

Baris set yang dihasilkan dari jalur independent adalah sebagai berikut

- a. 1-2-16
- b. 1-2-3-4-15-16
- c. 1-2-3-4-5-6-15-16
- d. 1-2-3-4-5-6-7-8-15-16
- e. 1-2-3-4-5-6-7-8-9-10-15-16
- f. 1-2-3-4-5-6-7-8-9-10-11-12-15-16
- g. 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16
- h. ketika aplikasi dijalankan, maka terlihat bahwa satu set baris yang dihasilkan adalah 1-2-16-1-2-3-4-15-16-1-2-3-4-5-6-15-16-1-2-3-4-5-6-7-8-15-16-1-2-3-4-5-6-7-8-9-10-15-16-1-2-3-4-5-6-7-8-9-10-11-12-15-16-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16 dan terlihat bahwa simpul telah dieksekusi satu kali.

2. **Black Box Testing**

Adapun dari segi pengujian blackbox testing pada aplikasi perhitungan matriks dan trigonometri antara lain sebagai berikut.

Tabel III. 1. Pengujian Blackbox kalkulator Matriks

Input	Test Case	Output	Validasi
Menginput kolom matriks A	Input EditText matriks A	Terinput nilai matriks A	Sukses
Menginput kolom matriks B	Input EditText matriks B	Terinput nilai matriks B	Sukses

Menentukan jenis operasi hitung	Klik listview pilih jenis operasi hitung	Menampilkan jenis operasi hitung	Sukses
Memproses hasil hitung	Klik Button Hitung	Memproses hasil hitung matriks A, tampil nilai perhitungan	Sukses

Tabel III. 2. Pengujian Blackbox kalkulator Trigonometri

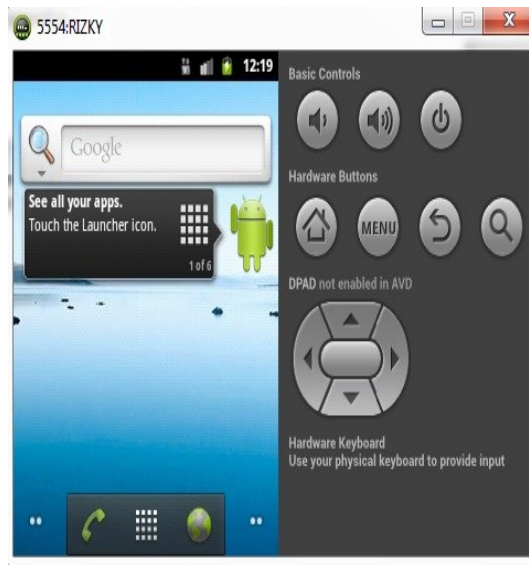
Input	Test Case	Output	Validasi
Menginput nilai sisi samping (cm)	Input EditText sisi samping	Terinput nilai sisi samping	Sukses
Menginput nilai sisi depan (cm)	Input EditText sisi depan	Terinput nilai sisi depan	Sukses
Menginput nilai sisi miring (cm)	Input EditText sisi miring	Terinput nilai sisi miring	Sukses
Memproses hasil hitung Sin	Mengklik Button Sin	Memproses hasil hitung dan menampilkan nilai Sin	Sukses
Memproses hasil hitung Cos	Mengklik Button Cos	Memproses hasil hitung dan menampilkan nilai Cos	Sukses

Memproses hasil hitung Tan	Mengklik Button Tan	Memproses hasil hitung dan menampilkan nilai Tan	Sukses
Memproses hasil hitung Cot	Mengklik Button Cot	Memproses hasil hitung dan menampilkan nilai Cot	Sukses
Memproses hasil hitung Sec	Mengklik Button Sec	Memproses hasil hitung dan menampilkan nilai Sec	Sukses
Memproses hasil hitung Cos ec	Mengklik Button Cos ec	Memproses hasil hitung dan menampilkan nilai Cos ec	Sukses

3.1 *Support*

Dalam menjalankan aplikasi yang dibuat penulis menggunakan perangkat keras yang mendukung kinerja Android seperti mobile, gadget, tablet, dan *computer*. Untuk device dibawah ini penulis menggunakan pengujian dengan menggunakan tools bernama Android Development Tools atau biasa disebut dengan singkatan ADT.

1. Tampilan Andorid Developmetn Tools



Gambar III. 16 Android Development Tools