

## CHAPTER 5

### IMPLEMENTATION AND TESTING

#### 5.1 Implementation

Implementation of this project design is using Java programming language in Android Studio, with array data structure. In the project implementation can run on Android API 19 or above. Below are the steps when the program is running :

##### 5.1.1 Input Total Task

Data that use in this project is the data from user input. User will input total task.

```
public void onclick()
{
    go.setOnClickListener((v) -> {
        number = Integer.parseInt(etTask.getText().toString());
        if(number <27)
        {
            for (int i = 0; i < number; i++)
            {
                cp = new SetGet();
                cp.setTask(Abjad[i]);
                cp.setDurasi(0);
                cp.setPredecessor("");
                cp.setKet("");
                arrayList.add(cp);
            }

            adapter = new ItemListAdapter(MainActivity.this, R.layout.baris, arrayList);
            ListView listView = (ListView) findViewById(R.id.listview);
            listView.setAdapter(adapter);
            Toast.makeText(MainActivity.this, "OK", Toast.LENGTH_SHORT).show();
        }
    });
}
```

Illustration 5.1: Code to show row based on user input

After that user can click on Add button then it will appear rows that contain the fields based on the number from user input. The fields are to filled up by activity name, duration, and predecessor.

##### 5.1.2 Data Submit

Data that have been inputted by user will be saved into array. Data from array will be sent to another page by using intent function. In first activity create new intent for send value.

1. `String Task = stringTask+ arrayList.get(i).getTask()+":";`
2. `Intent intent = new Intent(MainActivity.this, diagram.class);`

```

3. Intent.putExtra(STASK, stringTask);
4.
5. startActivity(intent);

```

In other activity, data can retrieve with this code, and save in array :

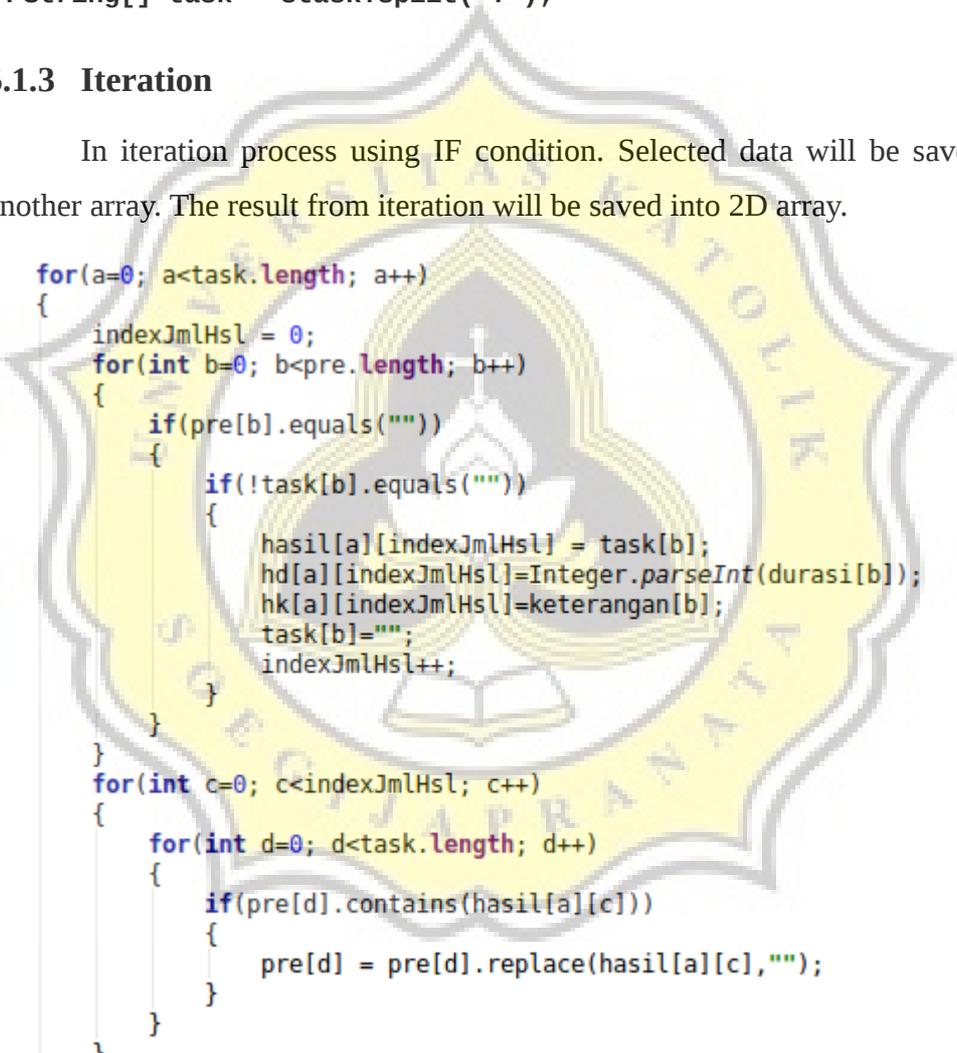
```

6. String stask =
   getIntent().getStringExtra(MainActivity.STASK);
7. String[] task = stask.split(":");

```

### 5.1.3 Iteration

In iteration process using IF condition. Selected data will be saved into another array. The result from iteration will be saved into 2D array.



```

for(a=0; a<task.length; a++)
{
    indexJmlHsl = 0;
    for(int b=0; b<pre.length; b++)
    {
        if(pre[b].equals(""))
        {
            if(!task[b].equals(""))
            {
                hasil[a][indexJmlHsl] = task[b];
                hd[a][indexJmlHsl]=Integer.parseInt(durasi[b]);
                hk[a][indexJmlHsl]=keterangan[b];
                task[b]="";
                indexJmlHsl++;
            }
        }
    }
    for(int c=0; c<indexJmlHsl; c++)
    {
        for(int d=0; d<task.length; d++)
        {
            if(pre[d].contains(hasil[a][c]))
            {
                pre[d] = pre[d].replace(hasil[a][c],"");
            }
        }
    }
}

```

Illustration 5.2: Iteration Code

### 5.1.4 Making circle in network diagram

In the iteration process, obtained result that will be determine the coordinate or position in network diagram.

```

float koor[][][] = new float [task.length][task.length][2];
for(int z=0; z<indexHasil; z++)
{
    for(int yy=0; yy<jmlHsl[z]; yy++)
    {
        koor[z][yy][0]=w;
        koor[z][yy][1]=p;
        canvas.drawCircle(w,p,20,lingkaran);
        canvas.drawText(hasil[z][yy]+","+hd[z][yy],(w-10),p,paint2);
        p+=80;
    }
    w+=80;
    p=20;

    if(z==indexHasil-1)
    {
        canvas.drawCircle(w,p,20,lingkaran2);
        canvas.drawText("FINISH",(w-15),p,Label);
    }
}

```

Illustration 5.3: Code to make circle

### 5.1.5 Forward Pass

Data that will be taken from the loop with some IF condition is the maximum value from an activity. The result from forward pass of every activity will be saved in variable.

```

String temp = Character.toString(cPre[t].charAt(cP));
if (temp.equals(hasil[z][yy])) {
    for (int tt = 0; tt < task.length; tt++) {
        if (cTask[tt].equals(temp)) {
            max = hasilHitung[z][yy][0] + Integer.parseInt(durasi[tt]);
            if (max > hasilHitung[z][yy][0]) {
                hasilHitung[z][yy][0] = max;
            }
        }
    }
}
}

```

Illustration 5.4: Code to find maximum value

### 5.1.6 Backward Pass

To do backward pass must loop all data, with some condition and find the minimum result of every activity. The result from backward pass will be saved in variable.

```

String temp = Character.toString(pasca[aa].charAt(bb));
if(temp.equals(hasil[xxx][yyy]))
{
    if(min>hasilHitung[xxx][yyy][1])
    {
        min = hasilHitung[xxx][yyy][1];
    }
}
}

```

Illustration 5.5: Code to find minimum value

### 5.1.7 Making Line

To make line with canvas in Android Studio. Line connect between an activity with its predecessor.

```
for(int jt=0;jt<task.length;jt++)
{
    if(cPre[jt].equals(""))
    {
        canvas.drawLine(20,20,xa,ya,garis);
    }
    ya+=80;
}
```

Illustration 5.6: Code to make a line

### 5.1.8 Determine and Draw the Critical Path

From the forward pass and backward pass, in every result of activity will be saved in variable. If the difference between the result of forward pass and backward pass is equal to zero, then the activity is the critical path.

```
8. hasilHitung[xx][yy][2] = hasilHitung[xx][yy][0] -
   hasilHitung[xx][yy][1];
9.   if(hasilHitung[xx][yy][2]==0)
10.      critical_path = critical_path+hasil[xx][yy];
```

So that the critical path can be seen clearly on the diagram then the circle which is the critical path given another color with this code:

### 5.1.9 Making gantt chart

To make gantt chart on canvas in Android Studio, using square as the starting point is the start time of an activity, and the ending point is the finish time of an activity. Time of gantt chart is adjusted with the result from project completion time. Every unit of time is quotient from maximum wide and maximum value of project completion time.

```

float maxWidth = 500;
float satuan;
satuan = maxWidth/fMaju;
float lA = 30 + ww;
for(int ee=0;ee<cTask.length;ee++)
{
    canvas.drawText("TASK",10,(10+ww),label);
    canvas.drawText(cTask[ee],10,lA,label);
    lA+=20;
    koorG[ee][1]=(ee+1)*20;
    for(int xx=0;xx<indexHasil;xx++)
    {
        for(int yy=0;yy<jmlHsl[xx];yy++)
        {
            if(hasil[xx][yy].equals(cTask[ee]))
            {
                koorG[ee][0]=(hasilHitung[xx][yy][0]*satuan);
            }
        }
    }
    float xA = koorG[ee][0];
    float yA = koorG[ee][1] + ww;
    canvas.drawRect((xA+20),yA,(xA+20+(Integer.parseInt(durasi[ee])*satuan)),(yA+20),paint);
    canvas.drawText(cTask[ee],(xA+40),(yA+10),paint3);
}

```

Illustration 5.7: Code to draw gantt chart

## 5.2 Testing

To know if the program will running well or not, the program should be tested. The final result which is expected to know the critical path of an activity, count the maximum project time that corresponding to CPM and it's displayed in network diagram and gantt chart. To do the test, user will input total activity in the available column.



Then click add button, then will appear input display (activity name, duration, and predecessor).

10

ADD

KODE	TASK	DURASI	PREDECESSOR
E		0	
F		0	
G		0	
H		0	
I		0	
J		0	

TAMPILKAN DATA PROSES

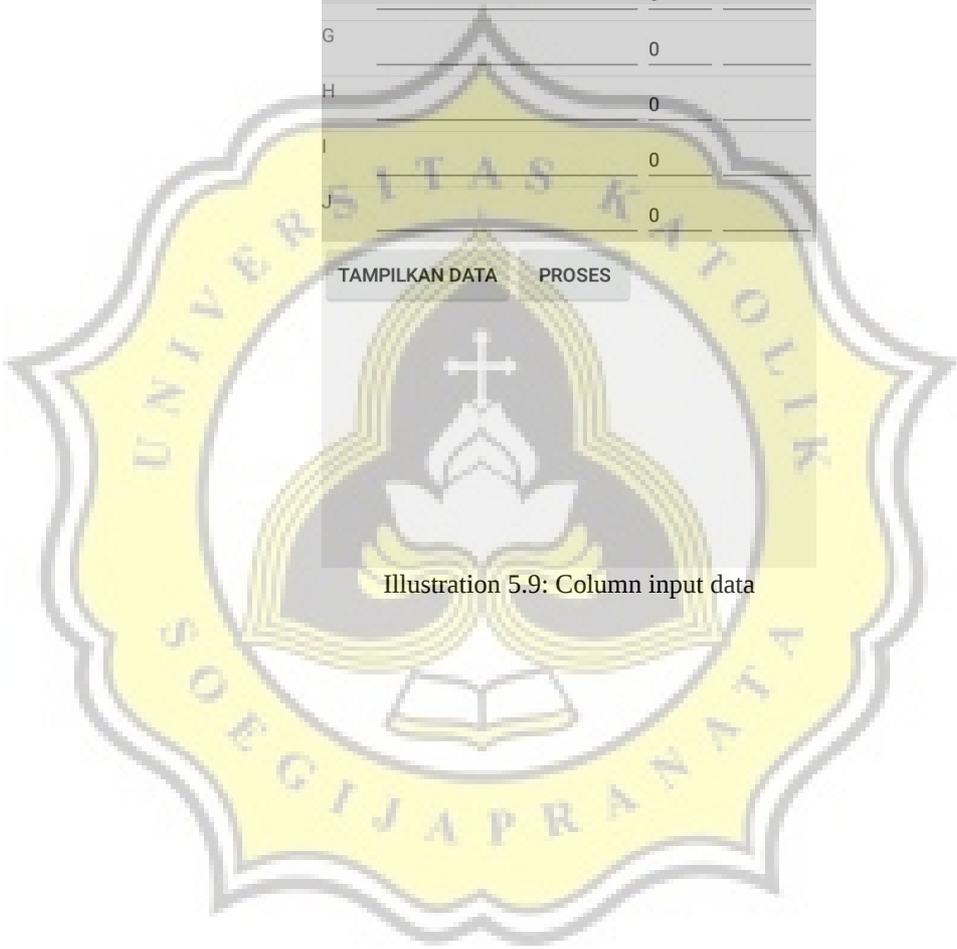


Illustration 5.9: Column input data

10

ADD

KODE	TASK	DURASI	PREDECESSOR
A	Product designing	5	
B	Basic research	1	
C	Product analysis	2	A
D	Product model	3	A
E	Sales brochure	2	A
F	Cost analysis	3	C

TAMPILKAN DATA PROSES

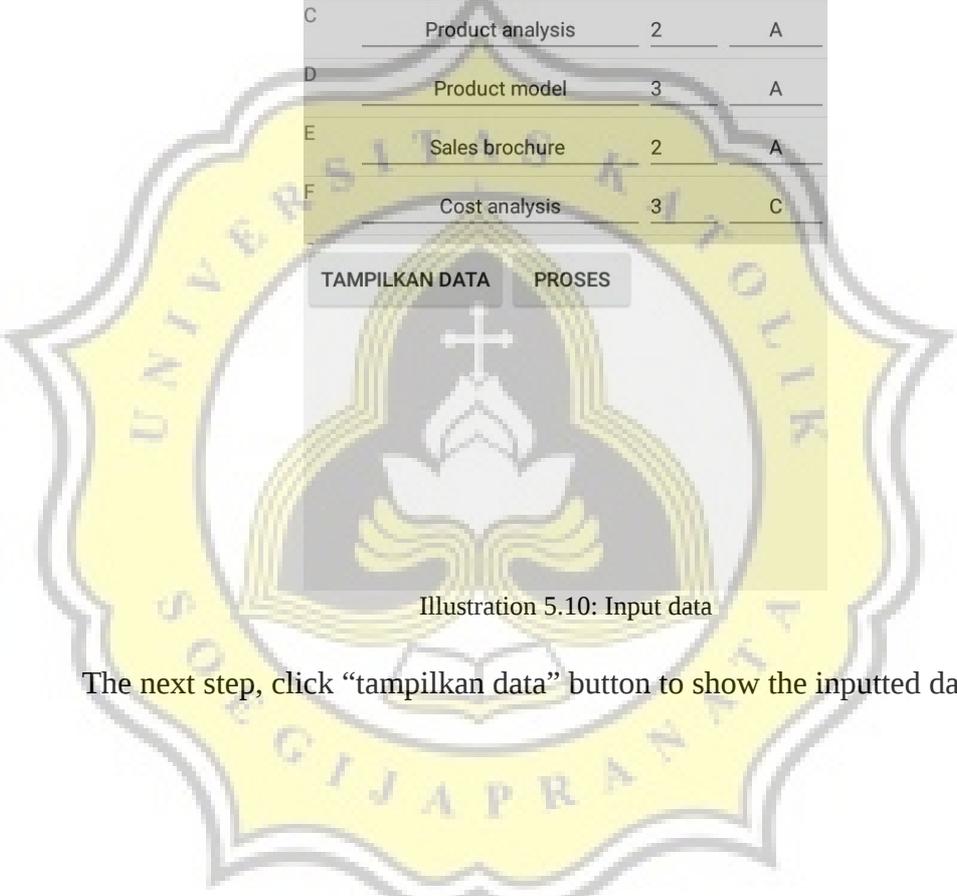


Illustration 5.10: Input data

The next step, click “tampilkan data” button to show the inputted data.

10

ADD

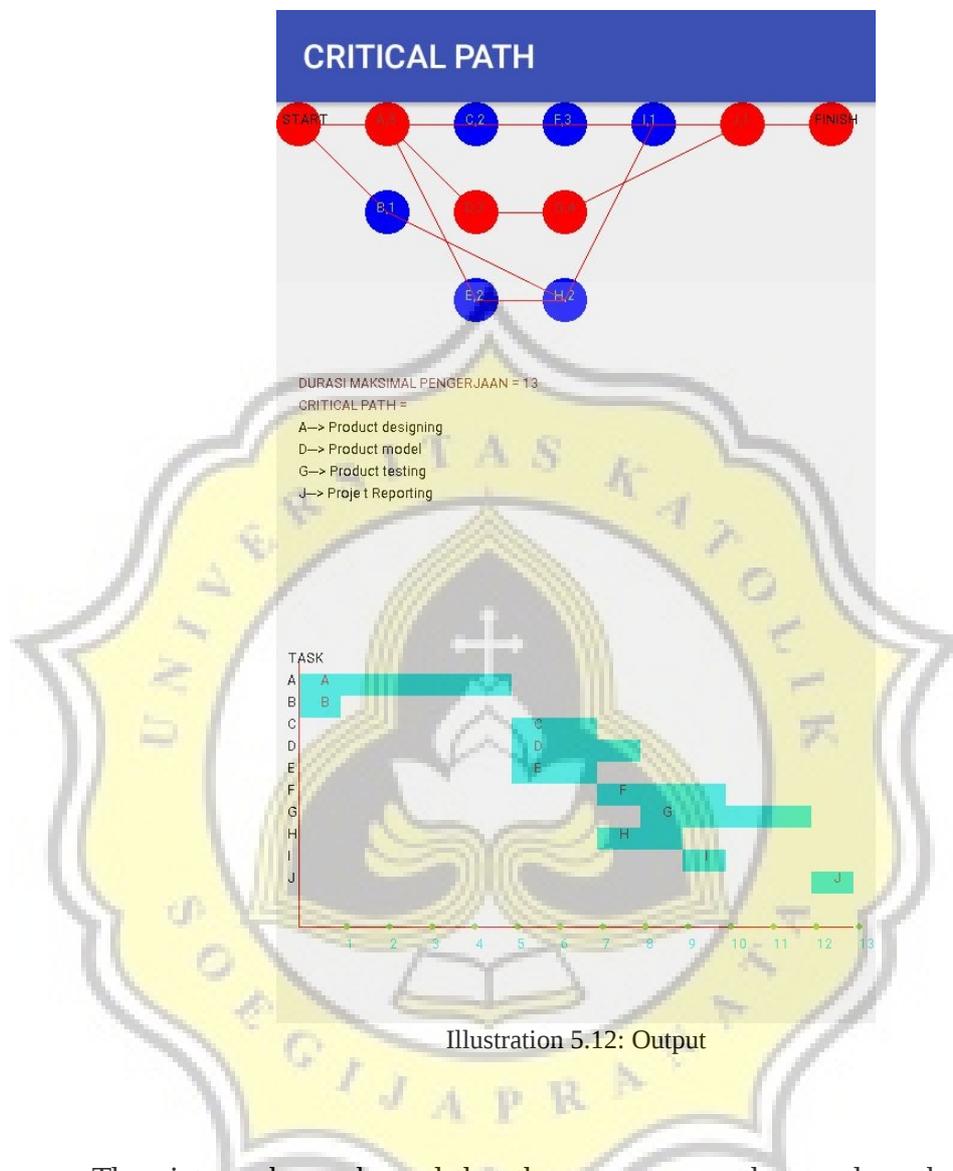
KODE	TASK	DURASI	PREDECESSOR
E	Sales brochure	2	A
F	Cost analysis	3	C
G	Product testing	4	D
H	Sales training	2	B,E
I	Proce fixing	1	H
J	Proje t Reporting	1	F,G,I

TAMPILKAN DATA	PROSES
A	Product designing 5
B	Basic research 1
C	Product analysis 2 A
D	Product model 3 A
E	Sales brochure 2 A
F	Cost analysis 3 C
G	Product testing 4 D
H	Sales training 2 B,E
I	Proce fixing 1 H
J	Proje t Reporting 1 F,G,I

Illustration 5.11: Display data

Then click “proses” button to run the program that has been made using CPM.



The picture above showed that the program run that produce the critical path that is A,D,G,J with activity name product designing, product model, product testing, project reporting and maximum time to finish the project is 13.