CHAPTER V
IMPLEMENTATION AND TESTING

5.1 Implementation

This application need several steps that must be done. for the first step that need to do is draw some pattern(vertexs and edges) that can be done by the user.

```
public boolean onTouchEvent(MotionEvent event) {
    switch (event.getAction()) {
    case MotionEvent.ACTION_DOWN:

        Point p = new Point();
        p.x = (int)event.getX();
        p.y = (int)event.getY();
        points.add(p);
        addNode(p.x,p.y);
        invalidate();



    case MotionEvent.ACTION_MOVE:
    case MotionEvent.ACTION_UP:
    case MotionEvent.ACTION_CANCEL: {
      break;
    }
    }
    invalidate();
    return true;

}
```

Code above have function to grab screen coordinates whenever user touch the screen, point p is a variable thatdeclared on the first program, and used to store coordinates.

```
    for(Point p: points)
      {
```


http://stackoverflow.com/questions/22791677/how-to-draw-a-new-circle-every-time-the-screen-is-touched
April 2014

```
        paint.setColor(Color.BLACK);
          canvas.drawCircle(p.x, p.y, 20, paint);
    }
```

Code above have the function to loop dependent on how many coordinates store on points, and send them to variable p. And then draw a circle depend on variable stored by user.

```
    for(String n: vertexName)
    {
      Typeface tf = Typeface.create("Helvetica",Typeface.BOLD);
      paintfont.setTypeface(tf);
      paintfont.setTextSize(20);
      int x = myGraph.koorX(n)+25;
      int y = myGraph.koorY(n)+25;

      paintfont.setColor(Color.BLUE);
        canvas.drawText(n, x, y, paintfont);
    }
```

After the circle appear, the user must know, the name of the circle that the user make, so code above print a circle or vertex name with font Helvetica and Type bold, so now every user touch the screen, right where user touch the screen will appear a circle, and right where user touch the screen +25 dpi, there will that circle name.

```
    for (int i=0; i<lines.size(); i++)
    {
        Rect currline= lines.get(i);
        canvas.drawLine(currline.left, currline.top,
currline.right, currline.bottom, paint);
    }
```

The user need to connect all the circle, so that is the function of creating a line whenever user fill edit text node1 and node 2, and connected them use the hubungkan buton.

```
      int r=0;
```


http://stackoverflow.com/questions/6304195/android-displaying-score-as-a-string-on-canvas-is-creating-a-new-string-per-dr
June 2014
http://developer.android.com/reference/android/graphics/Paint.html#setARGB(int,%20int,%20int,%20int
June 2014

```
        int g=0;
        int b=0;

      for (int i=0; i<postmanLines.size(); i++)
      {

        paint2.setARGB(255,r,g,b);
        paint2.setStrokeWidth(30);
          Rect currline= postmanLines.get(i);
          canvas.drawLine(currline.left, currline.top,
currline.right, currline.bottom, paint2);

          r=r+(255/postmanLines.size());
          g=g+(255/postmanLines.size());
          b=b+(255/postmanLines.size());
      }

      invalidate();
```

The next process is draw the pattern that sent by the tabu search algorithm r,g,b is a red green blue variable that needed to make a gradient colour so the user can see where the path go and where the path end. The path start with r,g,b 0 0 0(black) and end with r,g,b number near 255(near white). And invalidate is the last that run, invalidate method is canvas method that whenever the programmer finish the setup of canvas, invalidate will show the picture that the programmer willing to draw.

Next step is fill start vertex, and then press the start button, the code below

```
Button mybutton2;
EditText text3;
text3 = (EditText) findViewById(R.id.editText3);
mybutton2 = (Button) findViewById(R.id.button2);
mybutton2.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
if(!text3.getText().equals(null))
{
drawing.ChinesePostmanProblem(text3.getText().toString().toUpperCa
se());
};
}
});
```

EditText text3 is the field that the user need to fill , and Button mybutton2 is the button that user need to press , before the CPP process started. Button 3 named Start postman in the layout, text3 named start vertex in the layout. And chineseproblem method is method that make a list of CPP solution with start vertex that given by user.

Next step is processing the pattern into the right solution of postman problem

```
if(myGraph.cekVertex(start)&&myGraph.cekCycle(start))
        {
                myGraph.isEulerian();
                String [] tampung;
                tampung = myGraph.tabuSearch(start);
                int arrLength = tampung.length-1;
                int pos1=0;
                int pos2=1;

                for(int i=0;i<arrLength;i++)
                {
                        Rect tmp= new Rect();

                        tmp.left=myGraph.koorX(tampung[pos1]);
                        tmp.top=myGraph.koorY(tampung[pos1]);
                        tmp.right=myGraph.koorX(tampung[pos2]);
                        tmp.bottom=myGraph.koorY(tampung[pos2]);

                        pos1++;pos2++;
                        postmanLines.add(tmp);
                }
        }
```

Here the code above is to make a list of solution of CPP using tabusearch method, after a list of solution obtained,"for" method will store the coordinates of the list of vertex, to drawed latter. CekVertex method is boolean method to find if the vertex that inserted by user is a vertex that already declared, and cekCycle is a method to check if all vertex are connected, or there is another vertex that not connected to another, if there is no vertex is not connected, the

program will not run. IsEulerian is a method to find the vertex that have odd edges, and connect them.

```
public String[] tabuSearch(String start)
```

Now the tabusearch method, the method that return String , a list of solution, a list of vertex, that need to be draw latter.

```
Nodes tmp = headnodes;
while(tmp.getNext()!=null)
      {

            edges = edges+ (tmp.getHeadnode().getOrderVertex()/2);

            tmp = tmp.getNext();
      }
```

The very first step is to find how many edges contained in graph that user draw.

```
String [][] localSearch = new String[countAcak][edges];
localSearch[0][0] = start;

if((tmp.getHeadnode().getOrderVertex()/2)>1 &&
tmp.getHeadnode().getNamaVertex().equals(start) )
{
for(int j=0;j<forCounter;j++)
{
localSearch[0][counter] = tmp.getHeadnode().getNamaVertex();
}
}
localSearch[0][edges-1] = start;
```

Here, The code above, have function to generate very first list of string of solution CPP, at the first and end of the local search variable will fill the start vertex that defined by user. All the vertex will be included into this variable.

```
      String [] resetArray = new String[edges];
      String [] resetArray2 = new String[edges];
System.arraycopy( localSearch[0], 0,resetArray2, 0, edges);
```

There is 2 variable that needed to reset the list to list that have maximum score. Whenever the program find new maximum score or at least equal score, value of localsearch will changed to resetArray2.

```
checkerCon = hitungSkor(localSearch[0]);
```

Hitung Skor method, used to count the score of the localSearch , list of vertex.

```
while(checkerCon!=edges-1)
```

Hunt for the solution will not stop , before skor achieve the number of edges-1, edges -1 because if total edges on graph 8, the edges will be 9, because i will use the edges to define the length of array of vertex list, that is total edges on graph +1.

```
Random r = new Random();
posAwal = r.nextInt((edges-2) - 1) + 1;
posAcak = r.nextInt((edges-1) - posAwal) + posAwal;
```

2 int random variable needed to be used in scramble process. Whenever the scrambled process stuck, random variable will cover them. Random process variable posAwal will randomize a number between 1 and number of edges-2, and with posacak will generate random number between 1 and number of edges-1.

```
if(posAcak==edges-1){
posAwal++;
posAcak=posAwal+1;
System.arraycopy(resetArray, 0, localSearch[0], 0, edges);
}
```

Here is, the code above, whenever the scramble process reach the value of edges, this code above will reset it, but with pos awal++, and posacak start at posawal+1, so the scramble process will be restart again.

http://stackoverflow.com/questions/5785745/make-copy-of-array-java June 2014

```
for(int j=posAcak-1;j>posAwal-1;j--)
{
    localSearch[0][j+1] = localSearch[0][j];
}
```

And, the code above, will move the list at the position j to the right. So theres created a new combination of list

```
if(hitungSkor(listSearch[l])>=checkerCon)
```

The code above tell the tabu search algorithm, tabu search try to find something that equals or higher than the checkercon variable. So everytime hitungskor result reach or higher than checkerCon, value of checkerCon will replace with hitungskor result.

```
return resetArray2;
```

Whenever tabusearch method find a list of path with score equals edge-1, tabusearch method will return the value of that path. And so the tabu search process finished.

After the step above, the result is a list of vertex that need to be drawed at the canvas. There is process how to draw that list of vertex

```
public void ChinesePostmanProblem(String start)
{
      if(myGraph.cekVertex(start)&&myGraph.cekCycle(start))
      {
            myGraph.isEulerian();
            String [] tampung;
            tampung = myGraph.tabuSearch(start);
            int arrLength = tampung.length-1;
            int pos1=0;
            int pos2=1;

            for(int i=0;i<arrLength;i++)
            {
                  Rect tmp= new Rect();
```


http://stackoverflow.com/questions/7970857/shifting-elements-in-an-array
June 2014

```
                tmp.left=myGraph.koorX(tampung[pos1]);
                tmp.top=myGraph.koorY(tampung[pos1]);
                tmp.right=myGraph.koorX(tampung[pos2]);
                tmp.bottom=myGraph.koorY(tampung[pos2]);

                pos1++;pos2++;

                postmanLines.add(tmp);
            }

        }
}
```

```
tampung = myGraph.tabuSearch(start);
```

Come back to chinesePostmanProblem method so after aray of String tampung get the result that given by tabuSearch method,

```
int arrLength = tampung.length-1;
```

Program need to define how many loop needed for draw them all, so the program find the length of tampung variable -1, tampung.lentgh-1 because process will start from tampung[0] and tampung[1], and end at tampung[length-1] to tampung[length] so there is no null pointer error.

```
int pos1=0;
int pos2=1;
```

This two little variables , needed by program , to draw the lines we need x1,y1,x2,y2 and that is the functin of this two variables.

```
Rect tmp= new Rect();
```

Rect, a rectangle is used for drawing lines, rectangle have four point, because of that rectangle needed to draw this line consists of two variables x and two variables y.

```
tmp.left=myGraph.koorX(tampung[pos1]);
tmp.top=myGraph.koorY(tampung[pos1]);
tmp.right=myGraph.koorX(tampung[pos2]);
tmp.bottom=myGraph.koorY(tampung[pos2]);
```

There is how the rectangle filled, code above tell about rectangled fill by two coordinates x and y, so this rectangle was like, doesnt have a height.

```
postmanLines.add(tmp);
```

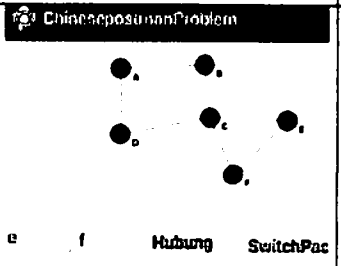
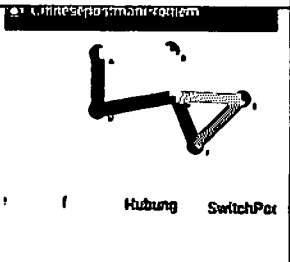
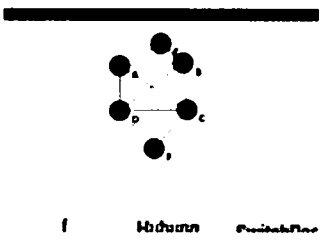
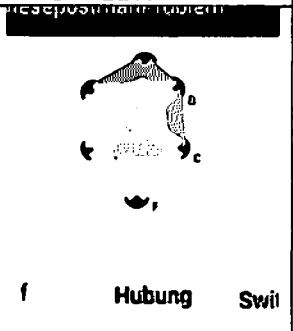
Here is the last process of chinesePostmanProblem method. Code above have a function to send tmp(Rect) to postmanLines(list of Rect), and draw the lines. Code to draw the line of rect will showed below.

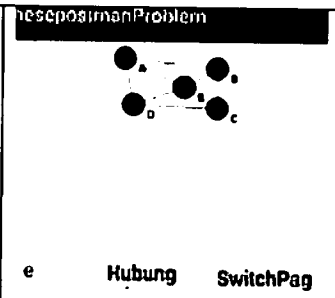
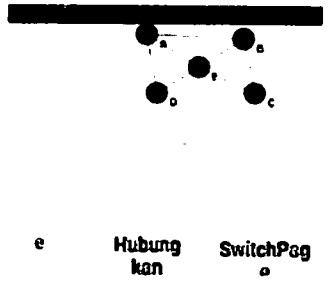
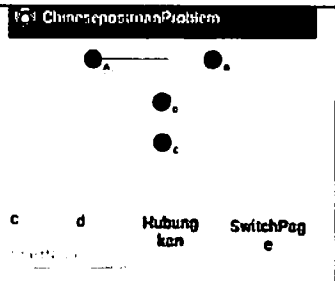
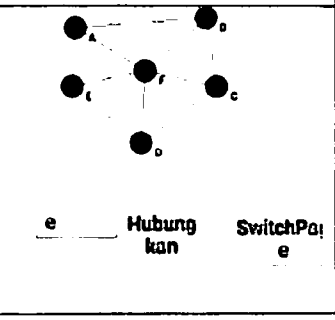
```
for (int i=0; i<postmanLines.size(); i++)
{

    paint2.setARGB(255,r,g,b);
    paint2.setStrokeWidth(30);
    Rect currline= postmanLines.get(i);
    canvas.drawLine(currline.left, currline.top, currline.right,
currline.bottom, paint2);

          r=r+(255/postmanLines.size());
          g=g+(255/postmanLines.size());
          b=b+(255/postmanLines.size());
      }
```

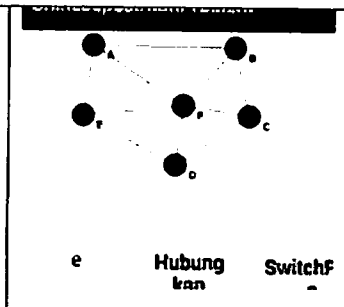
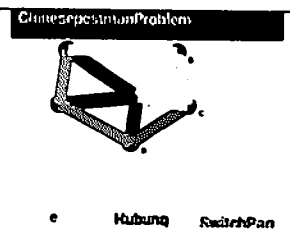
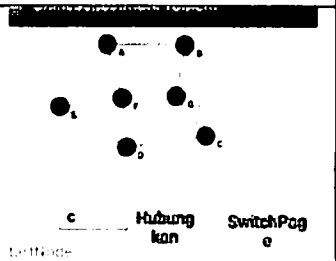
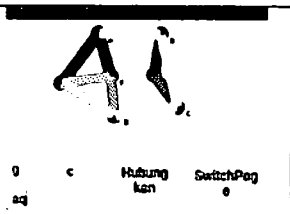
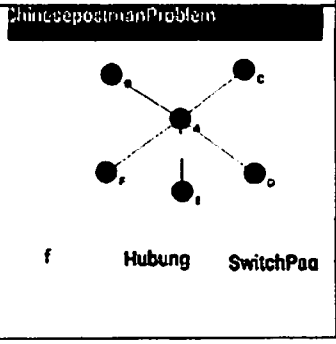
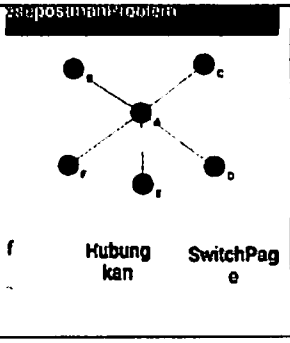
5.2 Testing

To know how far the tolerance of this application to different pattern, various pattern will be used and the test will show the path result,then, from path detected, the result with range from Success to failure will be showed. These are the testing result table:

Table 5.1 Testing Table

No	Pattern	Result Path	vertex/ Odd vertex	Note
1		 ADCFECBA	6/0	Success
2		 AEBCDBADFCA	6/0	Success

3		 ABECDEADCBA	5/4	Success
4		 Hubung SwitchPag AEDCBEABCDA	5/4	Path E C was not bypassed because the program glitched and then to create 3 more path instead of 2 path
5		 ACDABDCBA	4/4	Success
6			6/6	Success

7			6/6	Success
8			5/4	Path E C was not bypassed because the program glitched and then to create 3 more path instead of 2 path
9			7/6	Success
10			6/6	Looping proccess to find solution, was not ended, because solution was not found.

From testing result, we can see that pattern that have 5 vertex with 4 odd vertex sometimes contains not true solution for CPP. The rest of pattern works perfect, But at the last pattern seem no solution found, the program crashed.

5.3 Interface

5.3.1 Start interface

After launching application by touch application icon on android menu, user will face a simple layout.

Figure 5.3.1 Main Layout

After user touch the screen, a circle and circle name will appear on the screen, example screen shot can be view at Figure 5.3.2 below.

Figure 5.3.2 Result View

after user create more than one circle, user can connected them. to connect 2 circles is to fill two edit text node1 and node2 and then press button hubungkan.

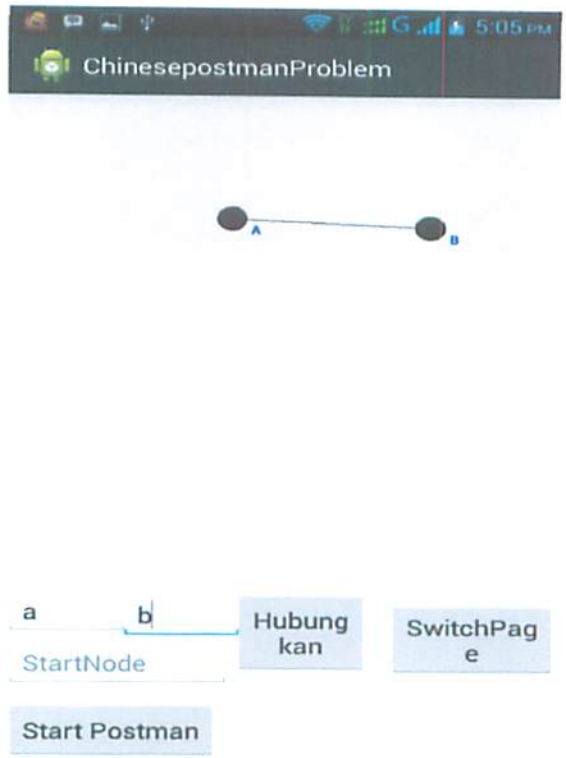


Figure 5.3.3 Result View Edge

5.3.2 Result

After creating a Cycle patern, user can start the CPP by choosing a vertex to be a start vertex, for example screen shot can be viewed at Figure 5.3.4 below

Figure 5.3.4 Result View CPP start

User can view the process of the CPP by clicking SwitchPage button, there will be some explanation to user, so user can be more understand about how the program works, the screen shot example can be viewed on Figure 5.3.5 below

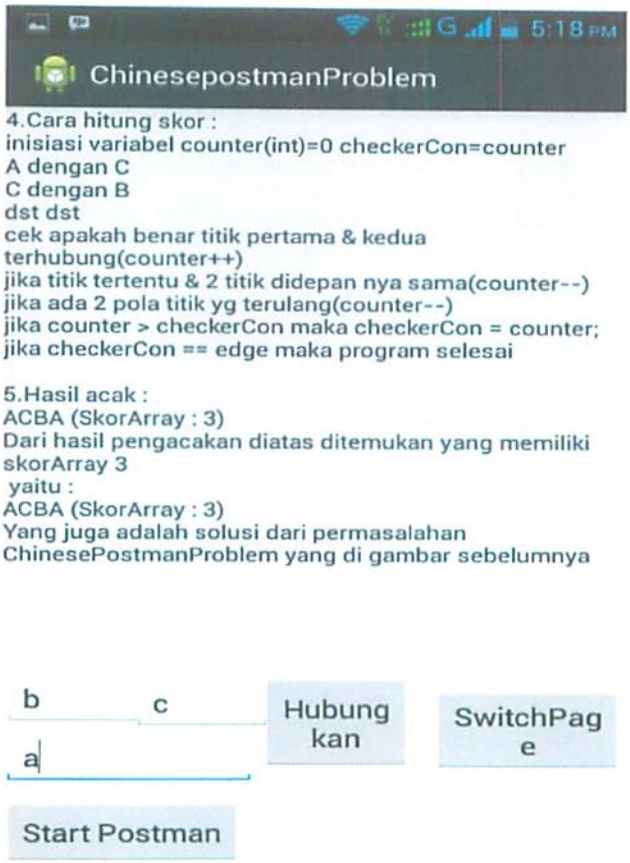


Figure 5.3.5 Result View User Page

http://stackoverflow.com/questions/17489 77/making-textview-scrollable-in-android July 2014