CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation

This project uses Java Programming Language and used three data structure. First is 2 linked list that store the room data used by user, and store the condition add in the program. Second Data Structure is Array list to store the data about the individuals that obtained in the processing. Last is Array that use for store the data layout used to ceking the fitness and show the optimum layout.

Below is the java programming contractor and Arraylist class that call in genetic algorithm process.

Row 1 is Arraylist class that call in genetic algorithm process. And row 2-12 is contractor that contain the store parameter that contains the values use in the program (row 5 panjang, 6 lebar tanah, 7 fitness terbesar, row 8-11 is the location of stairs).

Below is the method to add room and parameter use to add new room location that will save into linked list in genetic algorithm process.

Row 1 is Arraylist class that call in genetic algorithm process. And row 2-12 is contractor that contain the store parameter that contains the values use in the program (row 5 panjang, 6 lebar tanah, 7 fitness terbesar, row 8-11 is the location of stairs).
Row 1 is the name method and parameter use to add new condition. Row 2-15 is the process to place the new condition data. If the head is null, then it will create the head node. If it not null it will set on the next of the node.

Below is the method to generate 1 individuals and the parameter use in this method is the index location data will save:

1. public void setIndividu(int index){
2.  String namaR ="";
3.  int titik1 = 0;
4.  int titik2 = 0;
5.  curr = head;
6.  
7.  int letakX = 0;
8.  int letakY = 0;
9.  int xrandom = 0;
10. int yrandom = 0;
11. int randomLokasiX = 0;
12. int randomLokasiY = 0;
13. int cekPerulangan = 0;
14. String [][] layoutBidang = new String[pjngTanah*2][lbrTanah];
15. layoutBidang=setTangga();
16. while (curr != null){
17.  
18.  int x1 = 0;
19.  int y1 = 0;
20.  int x2 = 0;
int y2 = 0;
int bx = 0;
int by = 0;
int luasRuangan = 0;
int luasDicek = 0;
int xCetak = 0;
yCetak = 0;
int status = 0;
xLokasi = 0;
yLokasi = 0;
z = 0;
temp = 0;
batas = 0;
cekPerulangan = 0;
namaR = curr.getRuang();

bx = curr.getTitik1();
by = curr.getTitik2();
int pos = rn.nextInt(2);
if(pos==1){
    temp = bx;
    bx = by;
    by = temp;
}
if(namaR.contains("Garasi")){
    int max = Math.max(bx, by);
    int min = Math.min(bx, by);
    bx = max;
    by = min;
}
x1 = bx;
y1 = by;
luasRuangan = x1 * y1;

int set = rn.nextInt(2);
randomLokasiX = (set*pjngTanah)+rn.nextInt(pjngTanah);
randomLokasiY = rn.nextInt(lbrTanah);
if(set==0){
    for(int i = 0; i < 5; i++)
        if(cekPerulangan >= 4){
            randomLokasiX = 0;
            randomLokasiY = 0;
        }
    //System.out.println(pjngTanah*(set+1));
    for(int x = randomLokasiX; x < pjngTanah; x++)
        for(int y = randomLokasiY; y < lbrTanah; y++)
            //System.out.println("X nya =" + x + "$ Y nya =" +y);
        if(layoutBidang[x][y].equals("00000") && status == 0){
            if(x+x1<= pjngTanah){
                luasDicek = 0;
                if(x+x1<=pjngTanah){
for(int xr = x; xr < x+x1; xr++){
    if(y+y1<= lbrTanah){
        for(int yr = y; yr < y+y1; yr++){
            //System.out.println(xr + " "+yr);
            if(layoutBidang[xr][yr].equals("00000") ){
                luasDicek++;
                if(luasDicek==luasRuangan){
                    status = 1;
                    xLokasi = x;
                    yLokasi = y;
                    i=5;
                }
            }
        }
    }
    cekPerulangan++;
}
}
if(set==1){
    for(int i = 0; i < 5;i++)
    {
        if(cekPerulangan >= 4){
            randomLokasiX = 0;
            randomLokasiY = 0;
        }
        //System.out.println(pjngTanah*(set+1));
        for(int x = randomLokasiX; x < pjngTanah*2; x++){
            for(int y = randomLokasiY; y < lbrTanah ; y++){
                //System.out.println("X nya ="+ x + " Y nya = "+y);
                if(layoutBidang[x][y].equals("ZZZZZ") && status == 0){
                    if(x+x1<= pjngTanah*2){
                        luasDicek = 0;
                        if(x+x1==pjngTanah*2){

                    }

                }
            }
        }
    }
    //System.out.println(xr + " "+yr);
    if(layoutBidang[xr][yr].equals("ZZZZZ") ){
        luasDicek++;
    }
    luasDicek==luasRuangan){
        status = 1;
        xLokasi = x;
        yLokasi = y;
        i=5;
    }
}
}
Row 2-15 is generate the variable that use in the method. Row 16-58 is the random process using java default method, that will set the process the room will be placed on which floor. Row 69-100 is process the room will be placed First Floor, the location of the room will be randomized against length and width of land which will then be checked whether it is occupied or not and when it is occupied it will be randomized again(maximum randomized is 10x). Row 101-154 is the room will be placed in Second Floor, the location of the room will be randomized against length and width of land which will then be checked whether it is occupied or not and when it is occupied it will be randomized again (maximum randomized is 10 times).

Below is the method to processing the mutation of individuals

1. public void mutasi(int tempat,int index){
2. int jmlRuang=jumlahRuang()+1;
3. double[][] fit = new double[10][jmlRuang+1];
4. int[] melanggar = new int[jmlRuang+1];
5. int x=0;
6. int jumMelanggar = 0;
7. String ruang1 = "";
8. String ruang2 = "";
9. int ygDiubah = 0;
10. `String[] ruangan = new String[jmlRuang+1];`
11. `String namaR;`
12. `fit=dataFitnes(index,index);`
13. `for(int i = 0; i < jmlRuang;i++){`
14. `if(fit[0][i]!=1){`
15. `melanggar[x]=i+1;`
16. `x++;`
17. `}`
18. `}`
19. `ruangan=namaRuang();`
20. `if(x>0){`
21. `if(melanggar[0]>=0){`
22. `//System.out.print("vvvv "+melanggar[i]+" ");`
23. `ruang1 = ruangan[melanggar[0]-1];`
24. `}`
25. `}`
26. `}`
27. `}`
28. `if(x==1){`
29. `ygDiubah=rn.nextInt(jmlRuang-1);`
30. `ruang2 = ruangan[ygDiubah];`
31. `while(ruang2.equals(ruang1)){`
32. `ygDiubah=rn.nextInt(jmlRuang-1);`
33. `ruang2 = ruangan[ygDiubah];`
34. `}`
35. `else if(x>1){`
36. `ruang2 = ruangan[melanggar[1]-1];`
37. `}`
38. `}`
39. `}`
40. `tukarGen(tempat,index,ruang1,ruang2);`
41. `}`
42. `}

Row 1 is name method and parameter used the method. Row 2-12 id declare variable that contains the value of each variable used on the program. Row 13-19 is search the location of the violate room scan will start from index zero to last index (room total). Row 21-40 determination the room to be mutated it will process the room that violate, the 0 is for violate and 1 is for not violate. Row 41 is the process to save the individuals that clear mutated.

Below is the method to crossover the 2 difference individuals:

1. `public void crossover(int index,int index1, int index2){`
2. `String namaR = "";`
3. `int id1x1 = 0;`
4. `int id1x2 = 0;`
5. `int id1y1 = 0;`
6. `int id1y2 = 0;`
7. `int id2x1 = 0;`
8. `int id2x2 = 0;`
```
9. int id2y1 = 0;
10. int id2y2 = 0;
11. int tukar = 0;
12. int luasRuangan = 0;
13. int luasDicek = 0;
14. int status = 0;
15. int xLokasi = 0;
16. int yLokasi = 0;
17. int x1 = 0;
18. int x2 = 0;
19. int y1 = 0;
20. int y2 = 0;
21. int cekLokasi = 0;
22. curr = head;
23. String [][] layout = new String[pjngTanah*2][lbrTanah];
24. String [][] layout2 = new String[pjngTanah*2][lbrTanah];
25. int [] hasil = new int[4];
26. layout=setTangga();
27. layout2=setTangga();
28. while (curr != null){
29. namaR = curr.getRuang();
30. status = 0;
31. cekLokasi = 0;
32. 
33. int[] data1 = table.getDataByIndex(index1,namaR);
34. id1x1 = data1[0];
35. id1x2 = data1[1];
36. id1y1 = data1[2];
37. id1y2 = data1[3];
38. int[] data2 = table.getDataByIndex(index2,namaR);
39. id2x1 = data2[0];
40. id2x2 = data2[1];
41. id2y1 = data2[2];
42. id2y2 = data2[3];
43. tukar = rn.nextInt(2);
44. 
45. if(tukar==0){
46. id1x1 = data1[0];
47. id1x2 = data1[1];
48. id1y1 = data1[2];
49. id1y2 = data1[3];
50. 
51. id2x1 = data2[0];
52. id2x2 = data2[1];
53. id2y1 = data2[2];
54. id2y2 = data2[3];
55. }else{
56. id1x1 = data2[0];
57. id1x2 = data2[1];
58. id1y1 = data2[2];
59. id1y2 = data2[3];
60. }
61. id2x1 = data1[0];
62. id2x2 = data1[1];
```
Row 1 is the method name that using parameter of index location place the processed data will save and 2 index of individuals that will crossover. Row 2-31 is declare variable. Row 32-70 is the process determination room will be swap. Row 71-81 is the processing save new layout 1 into arraylist. Row 81-89 is the processing save new layout 2 into arraylist.

Below is the method to cek the fitness of individuals:

```java
public double cekFitnes(int noIndex, String namaRuang, String[][] layout){
    double fitness = 0;
    ListFitnes fcurr = fhead;
    double dFitnes = 0;
    double hasil = 0;
    fitness = fitness+cekLuasRuangan(layout,namaRuang);
    if(fitness>0){
        curr=curr.getNext();
    }
}
```
while (fcurr != null){
  if(namaRuang.contains(fcurr.getRuang1())){
    if(fcurr.getRuang2().contains("Depan")){
      fitness = fitness+cekTdkDepan(layout,namaRuang,fcurr.getJarak());
    }
    else if(fcurr.getRuang2().contains("Belakang")){
      fitness = fitness+cekTdkBelakang(layout,namaRuang,fcurr.getJarak());
    }else{
      fitness = fitness+cek2ruangan(layout,namaRuang,fcurr.getRuang2(),fcurr.getJarak());
      dFitnes = dFitnes+1;
    }
    fcurr=fcurr.getNext();
  }else {
    fitness = 0;
    dFitnes = dFitnes+1;
  }
}
hasil = (double)fitness/dFitnes;
if(namaRuang.contains("Mandi")){
  if(cekMandiTdkDepan(layout,namaRuang)<1){
    hasil=hasil-3;
  }
  if(cekMandiTidakSebelah1(layout,namaRuang)!=1){
    hasil=hasil-5;
  }
  if(cekMandiTidakSebelah2(layout,namaRuang)<1){
    hasil=hasil-5;
  }
}
if(namaRuang.contains("Dapur")){
  if(cekDapurTdkDepan(layout,namaRuang)!=1){
    hasil=hasil-5;
  }
  if(cekDapurTidakSebelah1(layout,namaRuang)!=1){
    hasil=hasil-5;
  }
}
if(namaRuang.contains("Tamu")){
  if(cekTamuDepan(layout,namaRuang)!=1){
    hasil=hasil-5;
  }
}
if(namaRuang.contains("Garasi")){
  if(cekGarasiDepan(layout,namaRuang)!=1){
    hasil=hasil-5;
  }
}
if(namaRuang.contains("Halam")){
  if(cekMandiTdkDepan(layout,namaRuang)!=1){
    hasil=hasil-5;
  }
}
Row 1 is name method and parameter used to process check the parameter. Row 3-6 is set variable. Row 8-27 is dynamic check fitness where the condition are added it will check every room and compare the condition of each room. Row 28-70 is checking the permanent fitness with weight value score, it will check the bathroom at row 34-48. Kitchen at row 44-47 kitchen is not locate in front. Row 48-50 is check the location the living room is set in front. Row 53-57 is check the location the garage is set in front. Row 58-60 is check the location yard room is not in middle. Row 63-65 is check the location the dining room is set in front.

Below is method to add new room place base on index location:

```java
1. public void tambah(int index, String ruang, int value, int value2, int valueX, int valueX2) {
2.     if (table[index] == null)
3.         table[index] = new LinkedHashEntry(ruang, value, value2, valueX, valueX2);
4.     else {
5.         LinkedHashEntry entry = table[index];
6.         while (entry.getNext() != null && !ruang.equals(entry.getRuang())){
7.             entry = entry.getNext();
8.         }
9.         entry.setNext(new LinkedHashEntry(ruang, value, value2, valueX, valueX2));
10.     }
11. }
12. }
```

Row 1 is method name add parameter that used in add new room. Row 2-3 is add the new room when the new room is first time add in the index. Row 4-12 is add new room when the previous room already exists.

### 5.2 Testing

The testing is divided into 3 section:

1. First Testing
Test the effect of the amount of data (number of room) and land space use to fitness obtained in the testing. Testing will do 15 times with fixed land space (length and width) and the number of rooms is a value which is changed in the process. And the result will recorded to be drawn conclusions.

2. Second Testing

Test the effect of the amount of data (number of room) and land space use to fitness and number of iteration obtained in the testing. Testing will do 15 times with fixed land space (length and width) and the number of rooms is a value which is changed in the process. And the maximum iteration use to test is 200 times iteration. And the result will recorded to be drawn conclusions.

3. Third Testing

The test effect of the amount of room condition to fitness gained. Testing will do 15 times with fixed land space (length and width) and the data is rooms that contain the different room number condition (number condition is a value which is changed in the process). And the result will recorded to be drawn conclusions.

Below is hardware use to test the program:

<table>
<thead>
<tr>
<th>No</th>
<th>Item Name</th>
<th>Vendor Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Processing Unit</td>
<td>Intel Core I5 4210U</td>
</tr>
<tr>
<td>2</td>
<td>Clock Speed</td>
<td>1.6 Ghz</td>
</tr>
<tr>
<td>3</td>
<td>Memory</td>
<td>16 GB</td>
</tr>
<tr>
<td>4</td>
<td>Core</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>OS</td>
<td>Ubuntu 14.04 LTS</td>
</tr>
<tr>
<td>6</td>
<td>HDD</td>
<td>Toshiba 500 GB</td>
</tr>
</tbody>
</table>

1. First Testing

Explain the effect of the amount of data (number of room) and land space use to fitness obtained in the testing. Testing will do several time with fixed land space (length and width) and the number of rooms is a value which is changed in the process. And the result of the experiments is below.
<table>
<thead>
<tr>
<th>Surface area</th>
<th>Condition</th>
<th>Remaining Area</th>
<th>Amount of Room</th>
<th>Size of Stairs</th>
<th>Fitness (test 1)</th>
<th>Fitness (test 2)</th>
<th>Fitness (test 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>800 (20x20)</td>
<td>7</td>
<td>398</td>
<td>10</td>
<td>3x2</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>800 (20x20)</td>
<td>7</td>
<td>302</td>
<td>14</td>
<td>3x2</td>
<td>100</td>
<td>97</td>
<td>98</td>
</tr>
<tr>
<td>800 (20x20)</td>
<td>7</td>
<td>242</td>
<td>16</td>
<td>3x2</td>
<td>100</td>
<td>93</td>
<td>97,9</td>
</tr>
<tr>
<td>800 (20x20)</td>
<td>7</td>
<td>182</td>
<td>18</td>
<td>3x3</td>
<td>89,3</td>
<td>93</td>
<td>93</td>
</tr>
<tr>
<td>800 (20x20)</td>
<td>7</td>
<td>122</td>
<td>20</td>
<td>3x3</td>
<td>83,7</td>
<td>82,5</td>
<td>83</td>
</tr>
</tbody>
</table>

From the test obtained, the addition of the amount of Surface Area and reduced of Remaining Area will affect the fitness received, and it decrease in fitness value.

2. Second Testing

Explain the effect of the amount of data (number of room) and land space use and number of iteration when get the maximum fitness obtained in the testing. Testing will do several time with fixed land space (length and width) and the number of rooms is a value which is changed in the process. And the maximum iteration use to test is 200x iteration. And the result will recorded to be drawn conclusions.
Table 5.3: Tabel Testing 2

<table>
<thead>
<tr>
<th>Surface Area</th>
<th>Condition</th>
<th>Remaining Area</th>
<th>Amount of Room</th>
<th>Fitness (test 1)</th>
<th>Fitness (test 2)</th>
<th>Fitness (test 3)</th>
<th>Iteration (test 1)</th>
<th>Iteration (test 2)</th>
<th>Iteration (test 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>800 (20 x 20)</td>
<td>7</td>
<td>398</td>
<td>10</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>180</td>
<td>186</td>
<td>192</td>
</tr>
<tr>
<td>800 (20 x 20)</td>
<td>7</td>
<td>302</td>
<td>14</td>
<td>100</td>
<td>97</td>
<td>98</td>
<td>183</td>
<td>98</td>
<td>114</td>
</tr>
<tr>
<td>800 (20 x 20)</td>
<td>7</td>
<td>242</td>
<td>16</td>
<td>100</td>
<td>93</td>
<td>97, 9</td>
<td>102</td>
<td>116</td>
<td>125</td>
</tr>
<tr>
<td>800 (20 x 20)</td>
<td>7</td>
<td>182</td>
<td>18</td>
<td>89, 3</td>
<td>93</td>
<td>93</td>
<td>90</td>
<td>124</td>
<td>114</td>
</tr>
<tr>
<td>800 (20 x 20)</td>
<td>7</td>
<td>122</td>
<td>20</td>
<td>83, 7</td>
<td>82, 5</td>
<td>83</td>
<td>127</td>
<td>130</td>
<td>107</td>
</tr>
</tbody>
</table>

From the test obtained the addition of the amount of Surface Area and reduced of Remaining Area arrest the maximum fitness gained in the range 83-130 for fitness that does not reach 100%. And for the fitness reach 100% get on iteration 183 from the process there is a match that can make the individual better, but can not be predicted because it only happens 1 times in 12 times testing.

3. Third Testing

Explain the effect of the amount of room condition to fitness gained. Testing will do several time with fixed land space (length and width) and the data is rooms that contain the different room number condition (number condition is a value which is changed in the process). And the result will recorded to be drawn conclusions.
From the test obtained the addition of the amount of condition arrest the maximum fitness gained. And more condition will it decrease maximum fitness value.
In the program user must input data to be processed, the input data is room condition start, length, width, position of stairs, and rooms that will be processed. If all data has been entered, process will start by clicking the button Process. The process will be done in several time. When the process is complete, it will display the layout below.

Illustration 5.1: Main Program