Laporan

Tugas Kecil Strategi Algoritma

Penyelesaian Persoalan 15-Puzzle dengan Algoritma Branch and Bound

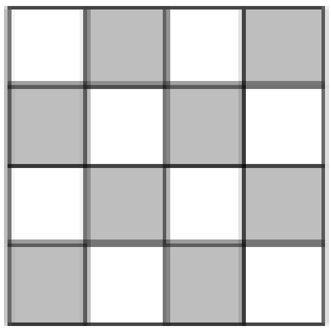
Disusun Oleh:
Fransiskus Davin Anwari/13520025

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2022

ALGORITMA BRANCH AND BOUND

Algoritma Branch and Bound adalah suatu algoritma pencarian solusi yang digunakan untuk persoalan optimisasi. Lebih rinci untuk meminimalkan atau memaksimalkan suatu fungsi objektif sambil tidak melanggar batasan persoalan. Algoritma memiliki suatu fungsi pembatasan yang berguna untuk "memangkas" jalur yang dianggap tidak mengarah ke suatu solusi.

Algoritma Branch and bound yang saya realisasikan mengimplementasikan suatu Priority Queue yang terdiri dari node berisi matrix, nilai c, dan langkah algoritma terakhir. Sebelum pencarian dimulai, algoritma menentukan terlebih dahulu jika 15-puzzle yang diberikan dapat diselesaikan. Hal ini dilakukan dengan menggunakan rumus Kurang(i), dimana pada setiap angka i pada matriks, akan dihitung berapa banyak angka yang kurang dari i yang tidak muncul sebelum sampai ke posisi i pada matriks. Rumus kurang(i) ini dilakukan 16 kali dari 1 sampai 15 dan 0 sebagai 16. Kemudian penjumlahan semua hasil kurang(i) akan dijumlahkan 1 jika angka 0 berada pada petak khusus yang digambarkan sebagai berikut.


Petak yang diarsir merupakan petak khusus untuk rumus kurang(i).

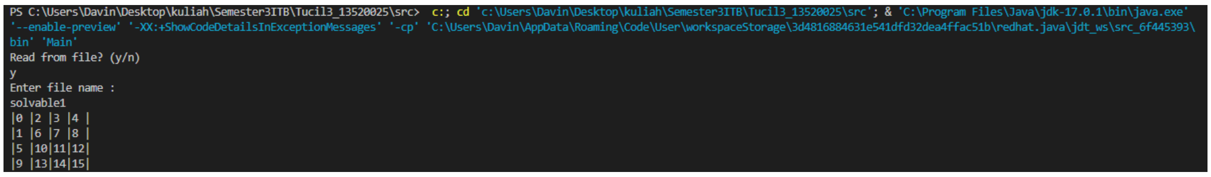
Jika dari penjumlahan semua rumus itu mendapatkan angka ganjil, maka puzzle tidak dapat diselesaikan sehingga program langsung diberhentikan. Jika mendapat angka genap, maka program lanjut ke tahap algoritma penyelesaian.

Pada saat algoritma mulai, program akan memasukan simpul akar ke dalam queue, lalu menelusuri anak dari simpul akar tersebut, kemudian simpul akar tersebut akan dipop. Simpul-simpul anak tersebut akan di enqueue secara berurut berdasarkan nilai c dari algoritma. Nilai c adalah cost dari simpul hidup yang ditelusuri. Cost ini dapat dihitung dengan menjumlahkan banyaknya petak tidak kosong yang letaknya belum sesuai dengan iterasi keberapa perjalanan algoritma.

Langkah selanjutnya adalah algoritma akan meneluri simpul anak yang memiliki nilai c paling kecil, dari sana algoritma akan menelusuri kembali simpul-simpul anak dari simpul yang ditelusuri tersebut. Hal ini akan terus diulangi hingga algoritma mencapai tujuan dimana simpul yang ditelusuri sudah sama dengan matrix solusi. Hal ini diketahui dengan mengecek apakah nilai fungsi g() = 0.

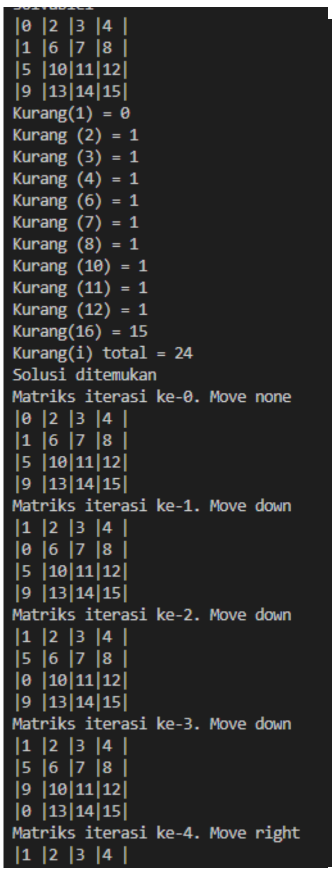
SCREENSHOT INPUT DAN OUTPUT

Puzzle Solvable

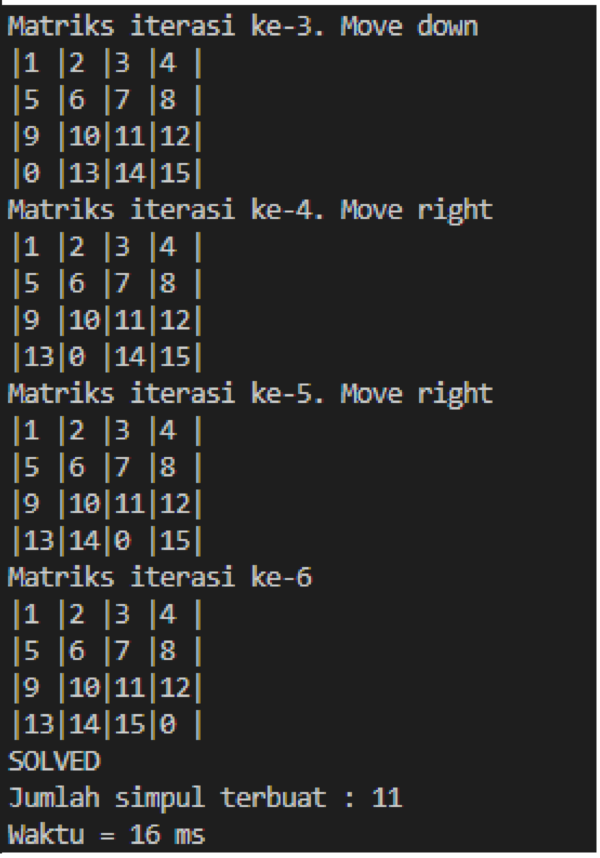


```
PS C:\Users\Davin\Desktop\kuliah\Semester3ITB\Tucil3_13520025\src>  c:; cd 'c:\Users\Davin\Desktop\kuliah\Semester3ITB\Tucil3_13520025\src'; & 'C:\Program Files\Java\jdk-17.0.1\bin\java.exe'
'--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Davin\AppData\Roaming\Code\User\workspaceStorage\3d4816884631e541dfd32dea4ffac51b\redhat.java\jdt_ws\src_6f445393\
bin' 'Main'
Read from file? (y/n)
y
Enter file name :
solvable1
|0 |2 |3 |4 |
|1 |6 |7 |8 |
|5 |10|11|12|
|9 |13|14|15|
```

Input puzzle solvable1

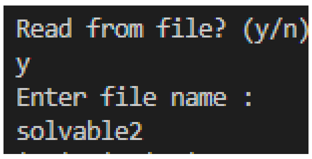


```
|0 |2 |3 |4 |
|1 |6 |7 |8 |
|5 |10|11|12|
|9 |13|14|15|
Kurang(1) = 0
Kurang (2) = 1
Kurang (3) = 1
Kurang (4) = 1
Kurang (6) = 1
Kurang (7) = 1
Kurang (8) = 1
Kurang (10) = 1
Kurang (11) = 1
Kurang (12) = 1
Kurang(16) = 15
Kurang(i) total = 24
Solusi ditemukan
Matriks iterasi ke-0. Move none
|0 |2 |3 |4 |
|1 |6 |7 |8 |
|5 |10|11|12|
|9 |13|14|15|
Matriks iterasi ke-1. Move down
|1 |2 |3 |4 |
|0 |6 |7 |8 |
|5 |10|11|12|
|9 |13|14|15|
Matriks iterasi ke-2. Move down
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|0 |10|11|12|
|9 |13|14|15|
Matriks iterasi ke-3. Move down
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|11|12|
|0 |13|14|15|
Matriks iterasi ke-4. Move right
|1 |2 |3 |4 |
```



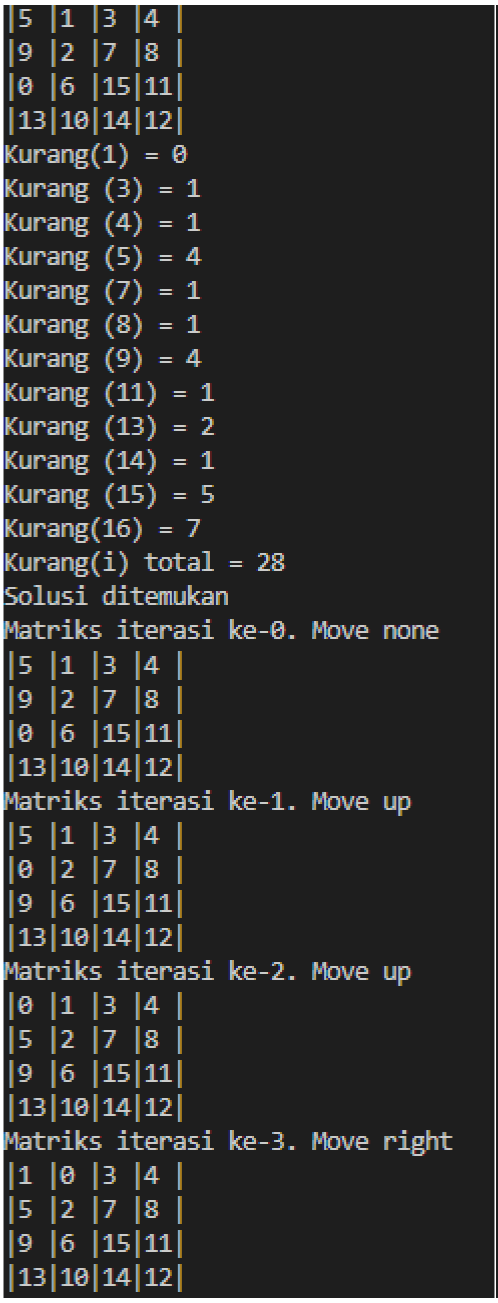
```
Matriks iterasi ke-3. Move down
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|11|12|
|0 |13|14|15|
Matriks iterasi ke-4. Move right
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|11|12|
|13|0 |14|15|
Matriks iterasi ke-5. Move right
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|11|12|
|13|14|0 |15|
Matriks iterasi ke-6
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|11|12|
|13|14|15|0 |
SOLVED
Jumlah simpul terbuat : 11
Waktu = 16 ms
```

Keluaran Solusi

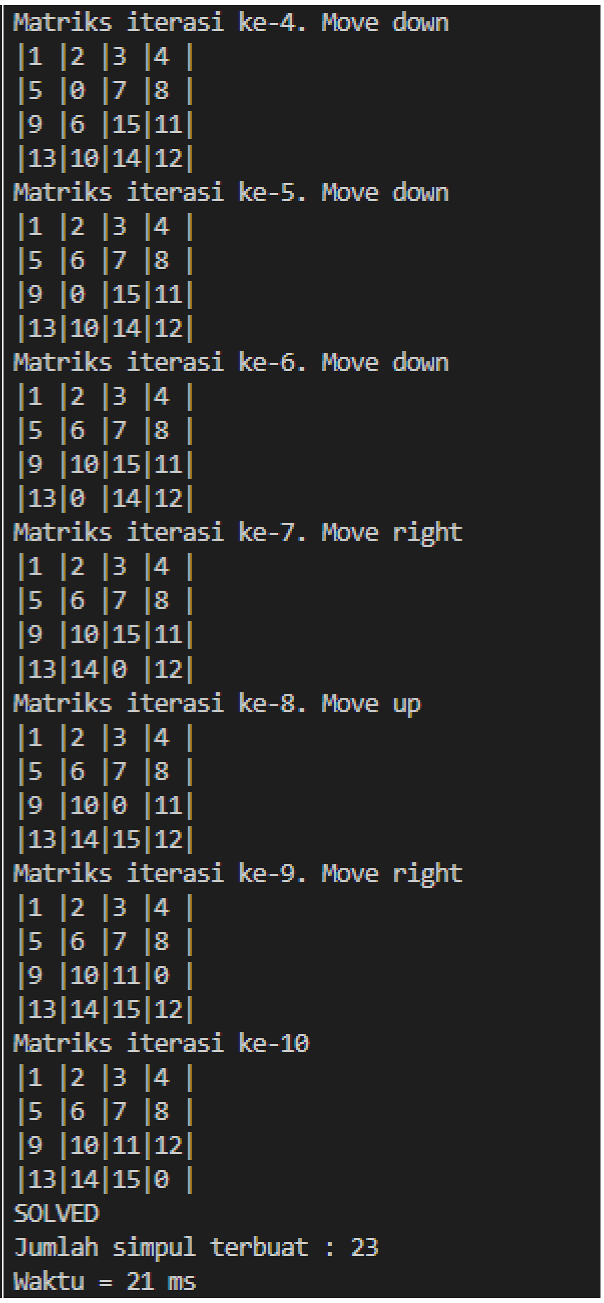


```
Read from file? (y/n)
y
Enter file name :
solvable2
```

Input puzzle solvable2

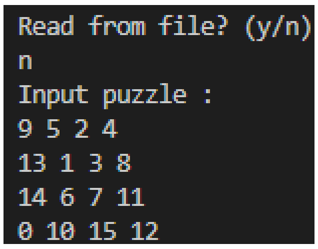


```
|5 |1 |3 |4 |
|9 |2 |7 |8 |
|0 |6 |15|11|
|13|10|14|12|
Kurang(1) = 0
Kurang (3) = 1
Kurang (4) = 1
Kurang (5) = 4
Kurang (7) = 1
Kurang (8) = 1
Kurang (9) = 4
Kurang (11) = 1
Kurang (13) = 2
Kurang (14) = 1
Kurang (15) = 5
Kurang(16) = 7
Kurang(i) total = 28
Solusi ditemukan
Matriks iterasi ke-0. Move none
|5 |1 |3 |4 |
|9 |2 |7 |8 |
|0 |6 |15|11|
|13|10|14|12|
Matriks iterasi ke-1. Move up
|5 |1 |3 |4 |
|0 |2 |7 |8 |
|9 |6 |15|11|
|13|10|14|12|
Matriks iterasi ke-2. Move up
|0 |1 |3 |4 |
|5 |2 |7 |8 |
|9 |6 |15|11|
|13|10|14|12|
Matriks iterasi ke-3. Move right
|1 |0 |3 |4 |
|5 |2 |7 |8 |
|9 |6 |15|11|
|13|10|14|12|
```



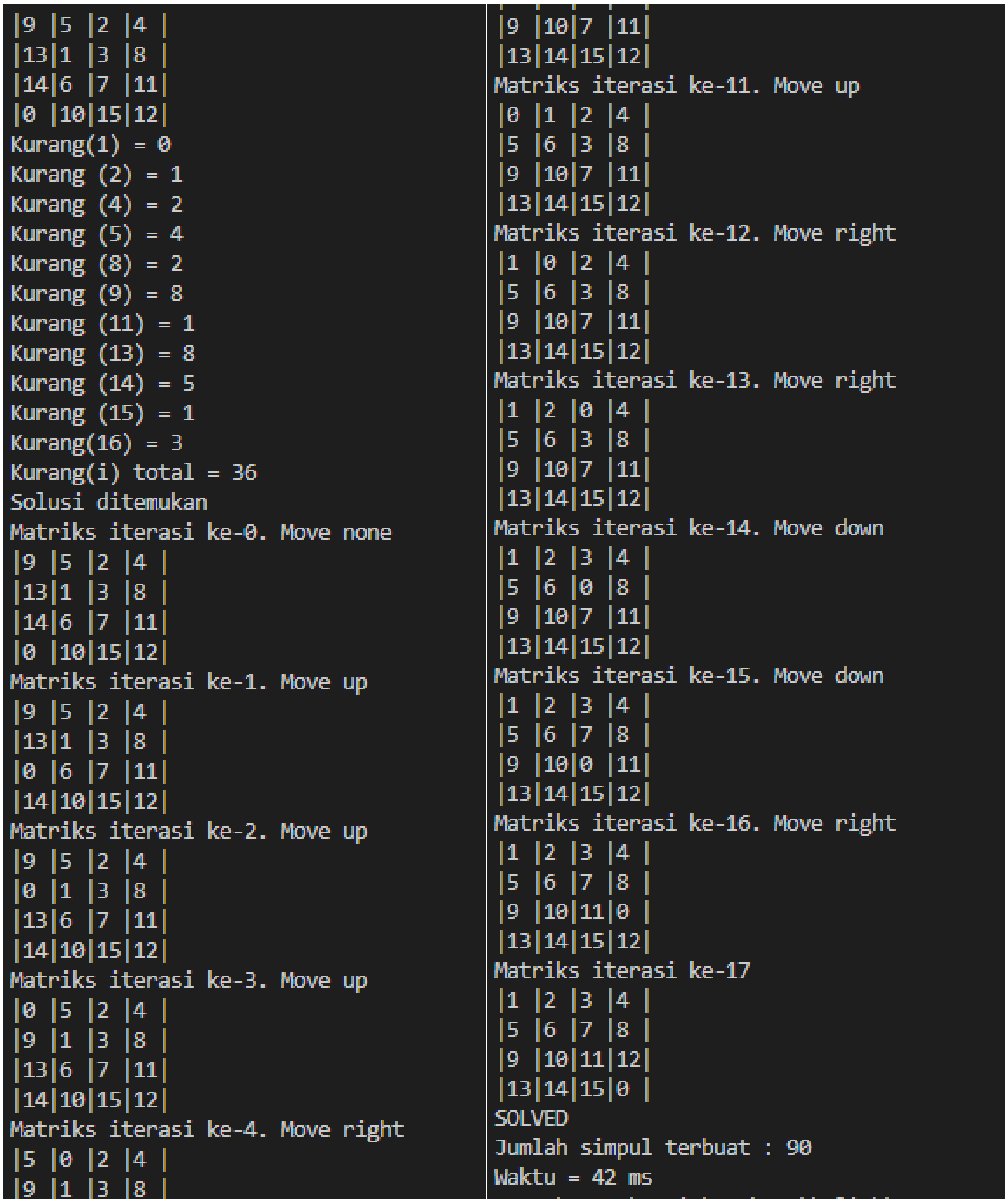
```
Matriks iterasi ke-4. Move down
|1 |2 |3 |4 |
|5 |0 |7 |8 |
|9 |6 |15|11|
|13|10|14|12|
Matriks iterasi ke-5. Move down
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |0 |15|11|
|13|10|14|12|
Matriks iterasi ke-6. Move down
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|15|11|
|13|0 |14|12|
Matriks iterasi ke-7. Move right
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|15|11|
|13|14|0 |12|
Matriks iterasi ke-8. Move up
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|0 |11|
|13|14|15|12|
Matriks iterasi ke-9. Move right
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|11|0 |
|13|14|15|12|
Matriks iterasi ke-10
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|11|12|
|13|14|15|0 |
SOLVED
Jumlah simpul terbuat : 23
Waktu = 21 ms
```

Keluaran solusi



```
Read from file? (y/n)
n
Input puzzle :
9 5 2 4
13 1 3 8
14 6 7 11
0 10 15 12
```

Input puzzle secara manual (solvable3)

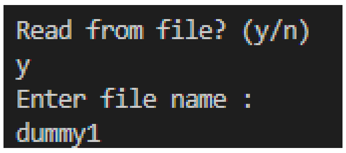


```
|9 |5 |2 |4 |
|13|1 |3 |8 |
|14|6 |7 |11|
|0 |10|15|12|
Kurang(1) = 0
Kurang (2) = 1
Kurang (4) = 2
Kurang (5) = 4
Kurang (8) = 2
Kurang (9) = 8
Kurang (11) = 1
Kurang (13) = 8
Kurang (14) = 5
Kurang (15) = 1
Kurang(16) = 3
Kurang(i) total = 36
Solusi ditemukan
Matriks iterasi ke-0. Move none
|9 |5 |2 |4 |
|13|1 |3 |8 |
|14|6 |7 |11|
|0 |10|15|12|
Matriks iterasi ke-1. Move up
|9 |5 |2 |4 |
|13|1 |3 |8 |
|0 |6 |7 |11|
|14|10|15|12|
Matriks iterasi ke-2. Move up
|9 |5 |2 |4 |
|0 |1 |3 |8 |
|13|6 |7 |11|
|14|10|15|12|
Matriks iterasi ke-3. Move up
|0 |5 |2 |4 |
|9 |1 |3 |8 |
|13|6 |7 |11|
|14|10|15|12|
Matriks iterasi ke-4. Move right
|5 |0 |2 |4 |
|9 |1 |3 |8 |
```

```
|9 |10|7 |11|
|13|14|15|12|
Matriks iterasi ke-11. Move up
|0 |1 |2 |4 |
|5 |6 |3 |8 |
|9 |10|7 |11|
|13|14|15|12|
Matriks iterasi ke-12. Move right
|1 |0 |2 |4 |
|5 |6 |3 |8 |
|9 |10|7 |11|
|13|14|15|12|
Matriks iterasi ke-13. Move right
|1 |2 |0 |4 |
|5 |6 |3 |8 |
|9 |10|7 |11|
|13|14|15|12|
Matriks iterasi ke-14. Move down
|1 |2 |3 |4 |
|5 |6 |0 |8 |
|9 |10|7 |11|
|13|14|15|12|
Matriks iterasi ke-15. Move down
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|0 |11|
|13|14|15|12|
Matriks iterasi ke-16. Move right
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|11|0 |
|13|14|15|12|
Matriks iterasi ke-17
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|11|12|
|13|14|15|0 |
SOLVED
Jumlah simpul terbuat : 90
Waktu = 42 ms
```

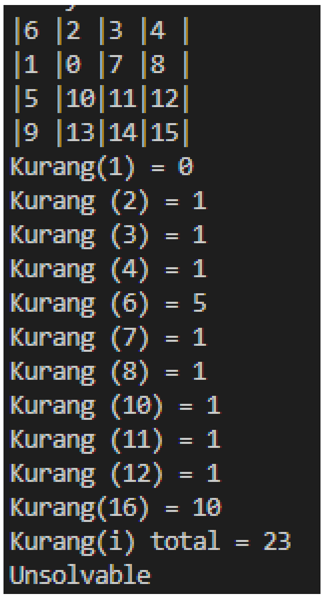
Keluaran Solusi

Puzzle Unsolvable



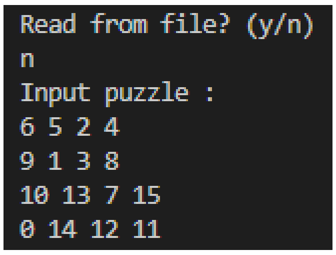
```
Read from file? (y/n)
y
Enter file name :
dummy1
```

Input puzzle dummy1



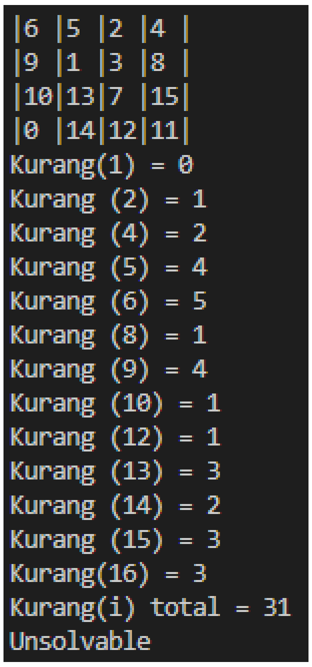
```
|6 |2 |3 |4 |
|1 |0 |7 |8 |
|5 |10|11|12|
|9 |13|14|15|
Kurang(1) = 0
Kurang (2) = 1
Kurang (3) = 1
Kurang (4) = 1
Kurang (6) = 5
Kurang (7) = 1
Kurang (8) = 1
Kurang (10) = 1
Kurang (11) = 1
Kurang (12) = 1
Kurang(16) = 10
Kurang(i) total = 23
Unsolvable
```

Keluaran dummy1



```
Read from file? (y/n)
n
Input puzzle :
6 5 2 4
9 1 3 8
10 13 7 15
0 14 12 11
```

Input puzzle secara manual (dummy2)



```
|6 |5 |2 |4 |
|9 |1 |3 |8 |
|10|13|7 |15|
|0 |14|12|11|
Kurang(1) = 0
Kurang (2) = 1
Kurang (4) = 2
Kurang (5) = 4
Kurang (6) = 5
Kurang (8) = 1
Kurang (9) = 4
Kurang (10) = 1
Kurang (12) = 1
Kurang (13) = 3
Kurang (14) = 2
Kurang (15) = 3
Kurang(16) = 3
Kurang(i) total = 31
Unsolvable
```

Hasil Convex Hull Perimeter vs Compactness

Kode Program

Github: https://github.com/fdavin/Tucil3_13520025
Google Drive: https://drive.google.com/drive/folders/1Uck_EKkTRX-nQ6KPhQ2VjhuEPcpJFjLm?usp=sharing

Cara menjalankan Program dapat dibaca pada README.md

Poin	Ya	Tidak
1. Pustaka berhasil dikompilasi	✓	
2. Program berhasil running	✓	
3. Program dapat menerima input dan menuliskan output	✓	
4. Luaran sudah benar untuk semua data uji	✓	
5. Bonus dibuat		✓

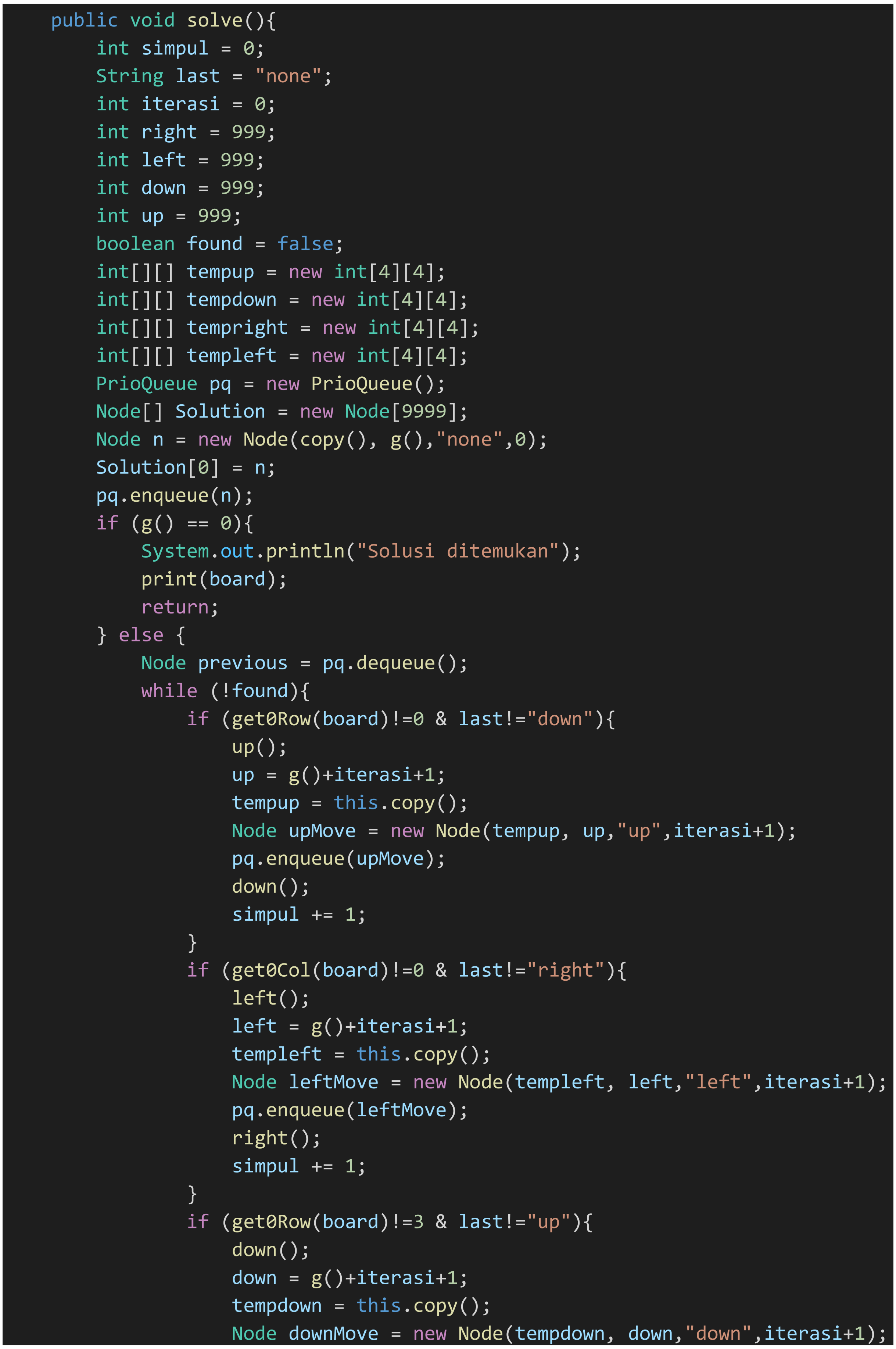
SOURCE CODE 15-PUZZLE PADA JAVA

Kode Puzzle.py (algoritma Branch and Bound)

```java
public class Puzzle {
    int[][] board = new int[4][4];
    int[][] solution = new int[][]{
        {1,2,3,4},
        {5,6,7,8},
        {9,10,11,12},
        {13,14,15,0}
    };
    public Puzzle(){
    }
    public Puzzle(int[][] board){
        this.board = board;
    }
    public int get0Row(int[][] currentboard){
        for(int i = 0; i < 4; i++){
            for(int j = 0; j < 4; j++){
                if(currentboard[i][j] == 0){
                    return i;
                }
            }
        }
        return -1;
    }
    public int get0Col(int[][] currentboard){
        for(int i = 0; i < 4; i++){
            for(int j = 0; j < 4; j++){
                if(currentboard[i][j] == 0){
                    return j;
                }
            }
        }
        return -1;
    }
    public int g(){
        int g = 0;
        for(int i = 0; i < 4; i++){
            for(int j = 0; j < 4; j++){
                if(board[i][j] != solution[i][j] & board[i][j] != 0){
                    g++;
                }
            }
        }
        return g;
    }
    public int distance(int FirstRow, int FirstCol, int SecondRow, int
SecondCol){
        int distanceRow = Math.abs(FirstRow - SecondRow);
        int distanceCol = Math.abs(FirstCol - SecondCol);
        return distanceRow+distanceCol;
    }
    public void change(int[][] result){
        for(int i = 0; i < 4; i++){
            for(int j = 0; j < 4; j++){
                board[i][j] = result[i][j];
            }
        }
    }
    public int[][] copy(){
        int[][] result = new int[4][4];
        for(int i = 0; i < 4; i++){
            for(int j = 0; j < 4; j++){
                result[i][j] = board[i][j];
            }
        }
        return result;
    }
    public void up(){
        int row = get0Row(board);
        int col = get0Col(board);
        if(row != 0){
            int temp = board[row-1][col];
            board[row-1][col] = 0;
            board[row][col] = temp;
        }
    }
    public void down(){
        int row = get0Row(board);
```

```java
        int col = get0Col(board);
        if(row != 3){
            int temp = board[row+1][col];
            board[row+1][col] = 0;
            board[row][col] = temp;
        }
    }
    public void right(){
        int row = get0Row(board);
        int col = get0Col(board);
        if(col != 3){
            int temp = board[row][col+1];
            board[row][col+1] = 0;
            board[row][col] = temp;
        }
    }
    public void left(){
        int row = get0Row(board);
        int col = get0Col(board);
        if(col != 0){
            int temp = board[row][col-1];
            board[row][col-1] = 0;
            board[row][col] = temp;
        }
    }
    public int kurang(){
        int kurang = 0;
        boolean loop = true;
        System.out.println("Kurang(1) = 0");
        for (int dicari=2;dicari<16;dicari++){
            while(loop){
                int temp = dicari-1;
                for (int i=0;i<4;i++){
                    for (int j=0;j<4;j++){
                        if (temp==0){
                            loop = false;
                            break;
                        }
                        if (board[i][j] < dicari & board[i][j] != 0){
                            temp-=1;
                        }
                        if (board[i][j]==dicari){
                            kurang += temp;
                            System.out.println("Kurang ("+dicari+") = "+temp);
                            loop = false;
                            loop = false;
                            break;
                        }
                    }
                    if (!loop){
                        break;
                    }
                }
            }
            loop = true;
        }
        int temp = 15;
        int count = 0;
        for (int i=0;i<4;i++){
            for (int j=0;j<4;j++){
                count = board[i][j];
                if (count < 16 & count != 0){
                    temp--;
                }
                if (count==0){
                    kurang += temp;
                    break;
                }
            }
            if (count==0){
                break;
            }
        }
        System.out.println("Kurang(16) = "+temp);
        if ((get0Col(board)+get0Row(board))%2!=0){
            kurang +=1;
        }
        System.out.println("Kurang(i) total = "+kurang);
        return kurang;
    }

    public void print(int[][] board){
        for(int i = 0; i < 4; i++){
            System.out.print("|");
            for(int j = 0; j < 4; j++){
                if (board[i][j]<10){
                    System.out.print(board[i][j] + " |");
                }
                else {
                    System.out.print(board[i][j] + "|");
                }
            }
            System.out.println();
        }
    }
```



```java
    public void solve(){
        int simpul = 0;
        String last = "none";
        int iterasi = 0;
        int right = 999;
        int left = 999;
        int down = 999;
        int up = 999;
        boolean found = false;
        int[][] tempup = new int[4][4];
        int[][] tempdown = new int[4][4];
        int[][] tempright = new int[4][4];
        int[][] templeft = new int[4][4];
        PrioQueue pq = new PrioQueue();
        Node[] Solution = new Node[9999];
        Node n = new Node(copy(), g(),"none",0);
        Solution[0] = n;
        pq.enqueue(n);
        if (g() == 0){
            System.out.println("Solusi ditemukan");
            print(board);
            return;
        } else {
            Node previous = pq.dequeue();
            while (!found){
                if (get0Row(board)!=0 & last!="down"){
                    up();
                    up = g()+iterasi+1;
                    tempup = this.copy();
                    Node upMove = new Node(tempup, up,"up",iterasi+1);
                    pq.enqueue(upMove);
                    down();
                    simpul += 1;
                }
                if (get0Col(board)!=0 & last!="right"){
                    left();
                    left = g()+iterasi+1;
                    templeft = this.copy();
                    Node leftMove = new Node(templeft, left,"left",iterasi+1);
                    pq.enqueue(leftMove);
                    right();
                    simpul += 1;
                }
                if (get0Row(board)!=3 & last!="up"){
                    down();
                    down = g()+iterasi+1;
                    tempdown = this.copy();
                    Node downMove = new Node(tempdown, down,"down",iterasi+1);
                    pq.enqueue(downMove);
                    up();
                    simpul += 1;
                }
                if (get0Col(board)!=3 & last!="left"){
                    right();
                    right = g()+iterasi+1;
                    tempright = this.copy();
                    Node rightMove = new Node(tempright,
right,"right",iterasi+1);
                    pq.enqueue(rightMove);
                    left();
                    simpul += 1;
                }
                previous = pq.dequeue();
                iterasi = previous.iterasi;
                Solution[iterasi] = previous;
                change(previous.node);
                if (g()==0){
                    found = true;
                    last = previous.last;
                    System.out.println("Solusi ditemukan");
                    int j=0;
                    while(Solution[j].node!=previous.node){
                        System.out.println("Matriks iterasi ke-
"+Solution[j].iterasi+". Move "+Solution[j].last);
                        print(Solution[j].node);
                        j++;
                    }
                    System.out.println("Matriks iterasi ke-"+j);
                    print(previous.node);
                } else {
                    last = previous.last;
                }
            }
        }if (iterasi==1000){
            System.out.println("Program tidak dapat sampai");
        } else {
               System.out.println("SOLVED");
        }
        System.out.println("Jumlah simpul terbuat : " + simpul);
    }

}
```

Kode Node.java

```java
public class Node {
    int[][] node = new int[4][4];
    int[][] solution = new int[][]{
        {1,2,3,4},
        {5,6,7,8},
        {9,10,11,12},
        {13,14,15,0}};
    int c;
    String last;
    int iterasi;
    Node(int[][] board, int c, String last, int iterasi){
        for (int i=0;i<4;i++){
            for (int j=0;j<4;j++){
                node[i][j] = board[i][j];
            }
        }
        this.c = c;
        this.last = last;
        this.iterasi = iterasi;
    }
}
```

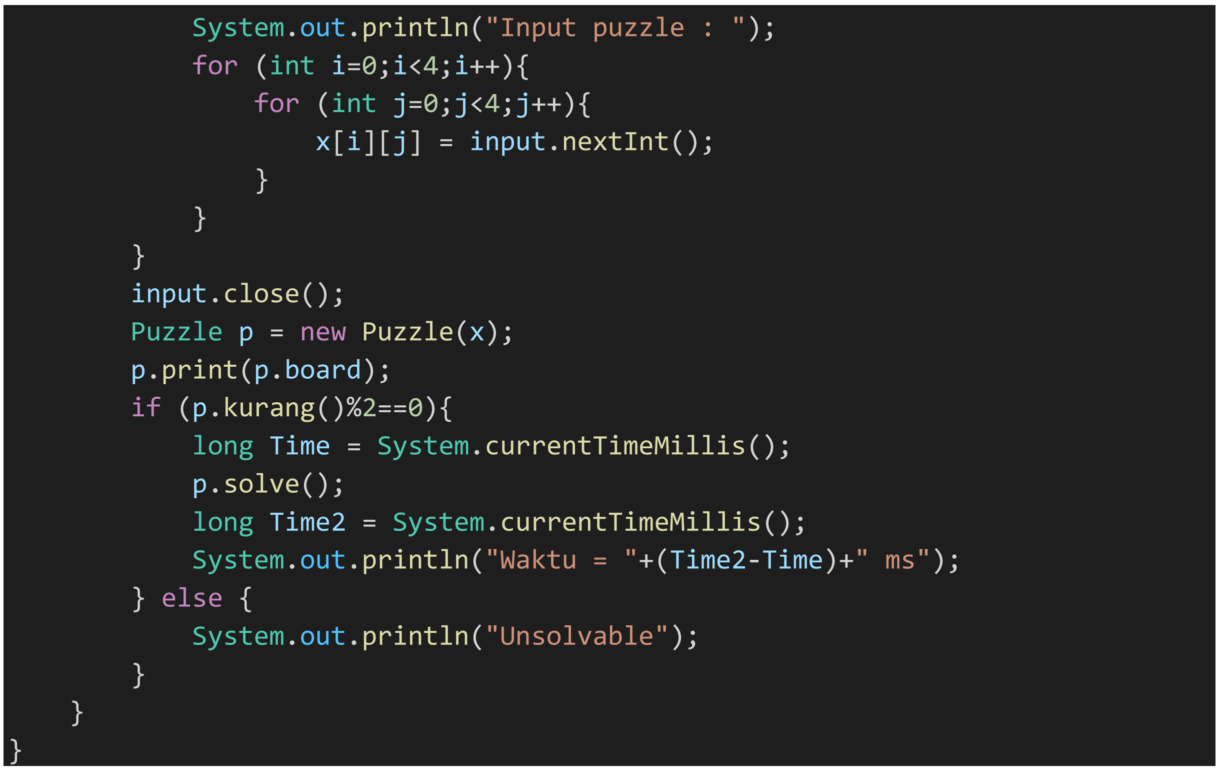
Kode PrioQueue.java

```java
public class PrioQueue {
    Node[] queue;
    int Neff;
    PrioQueue(){
        queue = new Node[9999];
    }
    public void enqueue(Node x){
        //enqueue node ke priority queue
        if (Neff==0){
            queue[Neff] = x;
            Neff++;
        } else {
            for (int i =0;i<Neff;i++){
                if (x.c<=queue[i].c){
                    for (int j=Neff;j>i;j--){
                        queue[j] = queue[j-1];
                    }
                    queue[i] = x;
                    Neff++;
                    break;
                }
            }
        }
    }
    public Node dequeue(){
        //dequeue node dari priority queue
        Node x = queue[0];
        queue[0] = queue[1];
        Neff--;
        for (int i = 1;i<Neff;i++){
            queue[i] = queue[i+1];
        }
        return x;
    }
}
```

Kode Main.java

```java
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class Main {
    public static int[][] ReadFile(String filePath) throws IOException {
        Scanner sc = new Scanner(new BufferedReader(new
FileReader("../test/"+filePath+".txt")));
          int rows = 4;
          int columns = 4;
          int [][] myArray = new int[rows][columns];
          while(sc.hasNextLine()) {
             for (int i=0; i<myArray.length; i++) {
                String[] line = sc.nextLine().trim().split(" ");
                for (int j=0; j<line.length; j++) {
                   myArray[i][j] = Integer.parseInt(line[j]);
                }
             }
          }
            return myArray;
        }
    public static void main(String[] args) throws IOException {
        Scanner input = new Scanner(System.in);
        int[][] x = new int[4][4];
        System.out.println("Read from file? (y/n)");
        String read = input.nextLine();
        if (read.equals("y")){
            System.out.println("Enter file name : ");
            String file = input.nextLine();
            x = ReadFile(file);
        } else if (read.equals("n")){
```



```java
            System.out.println("Input puzzle : ");
            for (int i=0;i<4;i++){
                for (int j=0;j<4;j++){
                    x[i][j] = input.nextInt();
                }
            }
        }
        input.close();
        Puzzle p = new Puzzle(x);
        p.print(p.board);
        if (p.kurang()%2==0){
            long Time = System.currentTimeMillis();
            p.solve();
            long Time2 = System.currentTimeMillis();
            System.out.println("Waktu = "+(Time2-Time)+" ms");
        } else {
            System.out.println("Unsolvable");
        }
    }
}
```

Test Case

1. solvable1.txt

0 2 3 4

1 6 7 8

5 10 11 12

9 13 14 15

2. solvable2.txt

5 1 3 4

9 2 7 8

0 6 15 11

13 10 14 12

3. solvable3.txt

9 5 2 4

13 1 3 8

14 6 7 11

0 10 15 12

4. dummy1.txt

6 2 3 4

```
1 0 7 8

5 10 11 12

9 13 14 15
```

5. dummy2.txt

```
6 5 2 4

9 1 3 8

10 13 7 15

0 14 12 11
```