Kecerdasan Buatan

Pertemuan 3

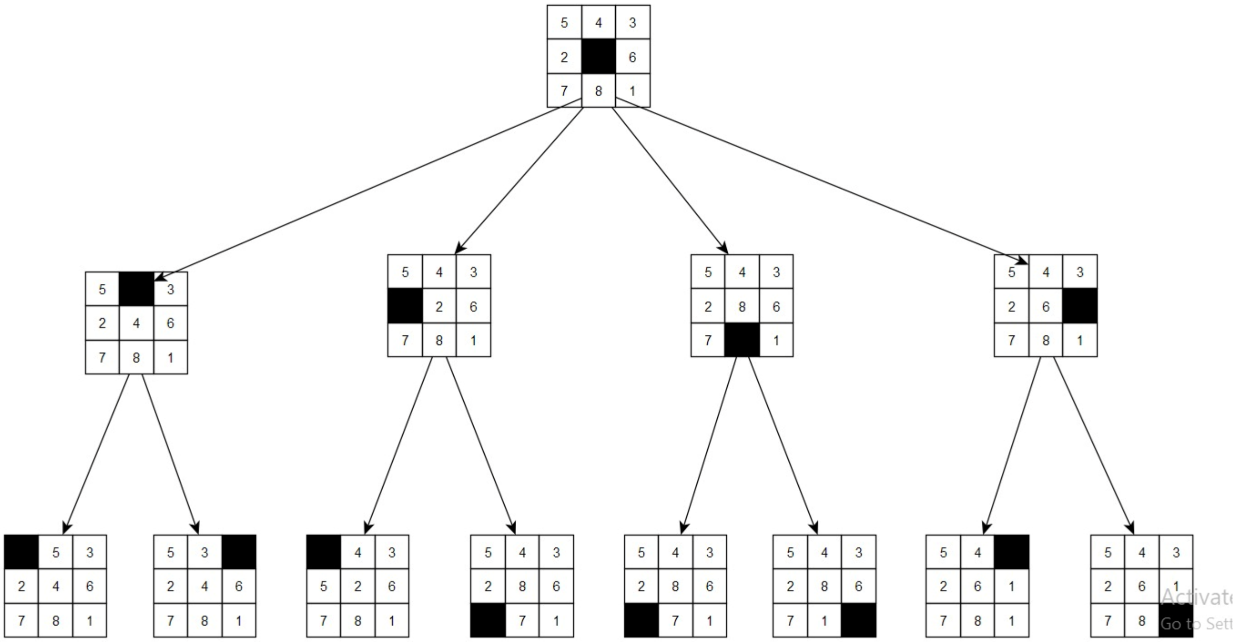
Rizky cahya zuliyanto (3122522018)
Yusuf Rahmadhani A (3122522030)
Tolak Ivandi Y (3122522020)
D3 PSDKU Sumenep

PRODI D3 TEKNIK INFORMATIKA

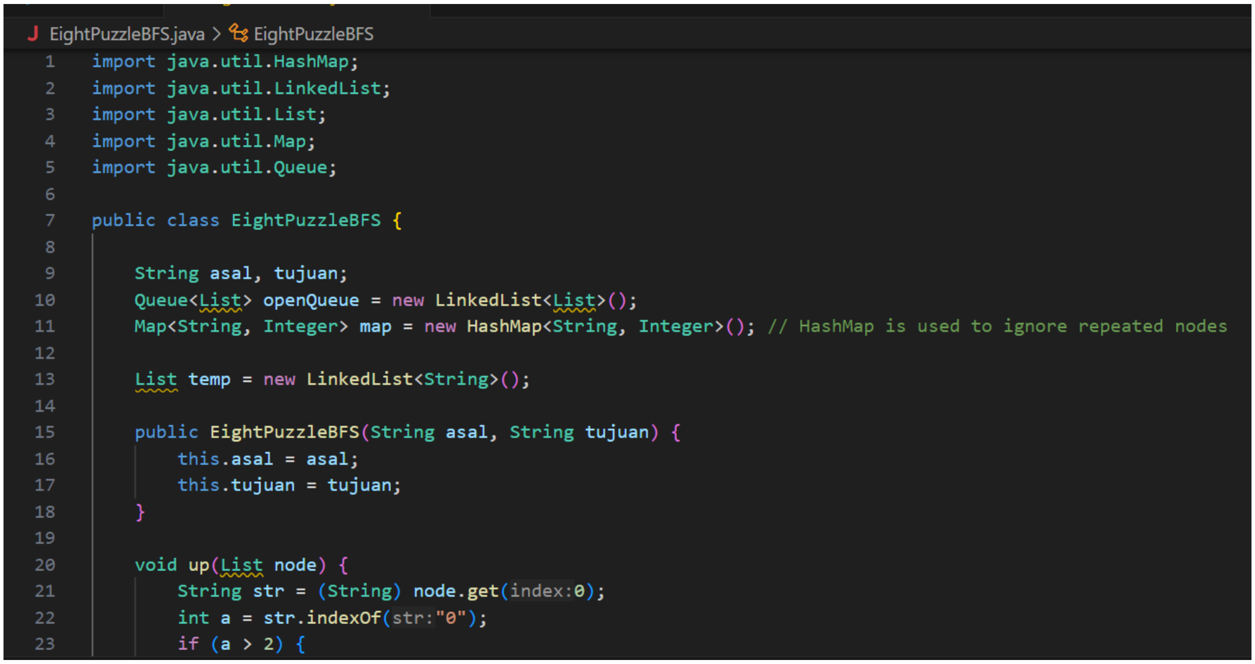
DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER

PENS PSDKU SUMENEP

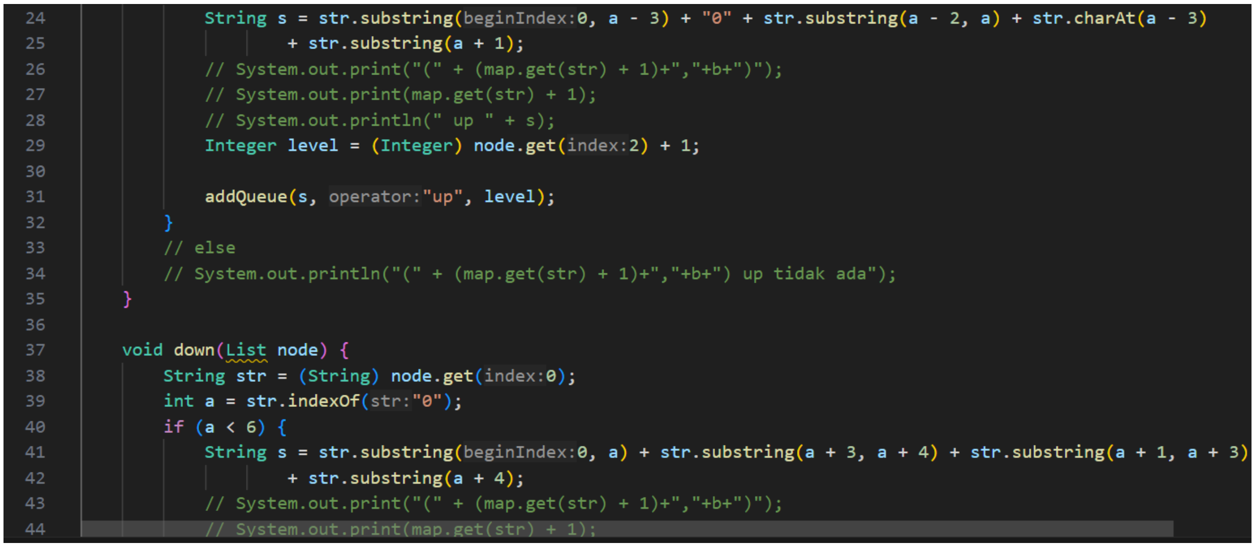
1.



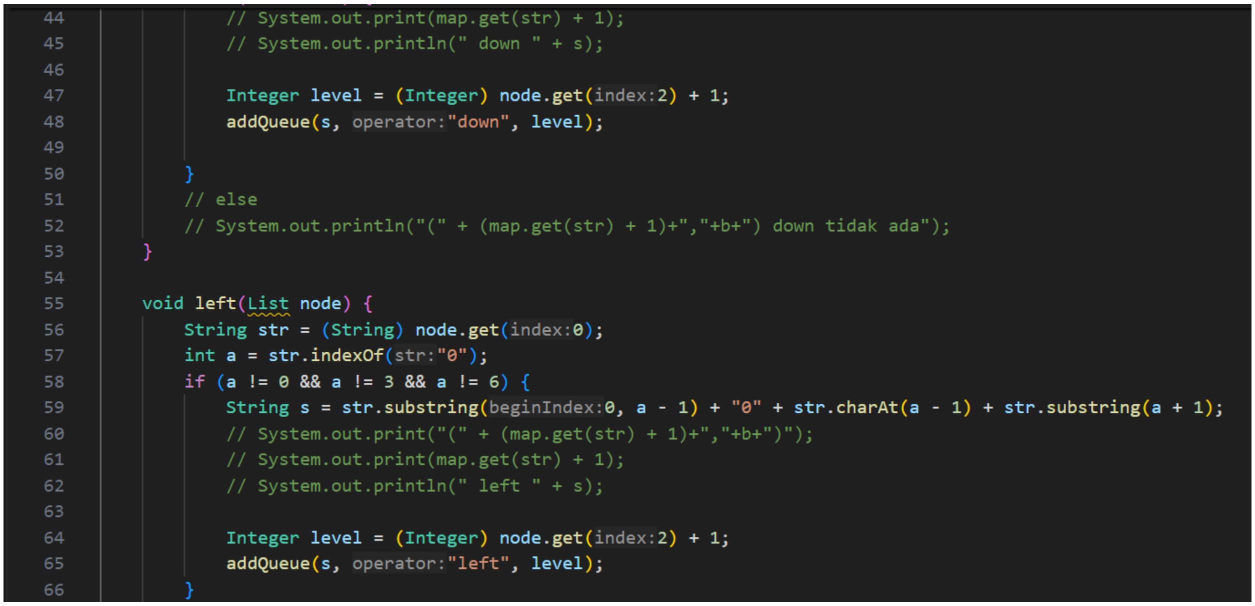
2.



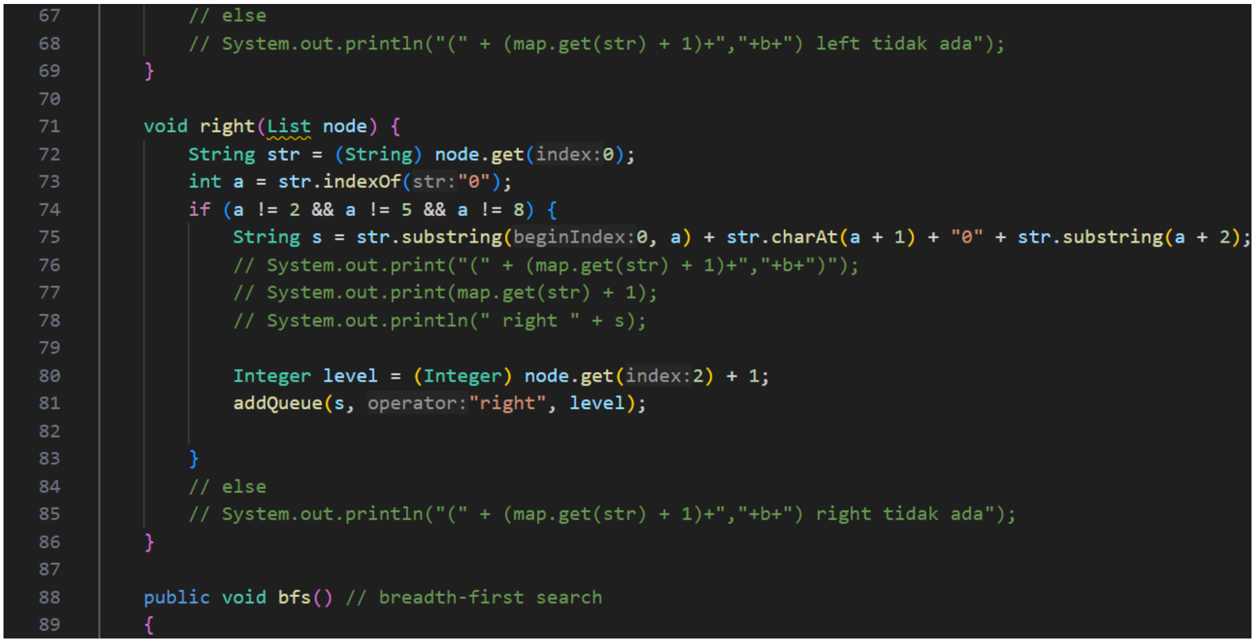
```java
import java.util.HashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import java.util.Queue;

public class EightPuzzleBFS {

    String asal, tujuan;
    Queue<List> openQueue = new LinkedList<List>();
    Map<String, Integer> map = new HashMap<String, Integer>(); // HashMap is used to ignore repeated nodes

    List temp = new LinkedList<String>();

    public EightPuzzleBFS(String asal, String tujuan) {
        this.asal = asal;
        this.tujuan = tujuan;
    }

    void up(List node) {
        String str = (String) node.get(index:0);
        int a = str.indexOf(str:"0");
        if (a > 2) {
```



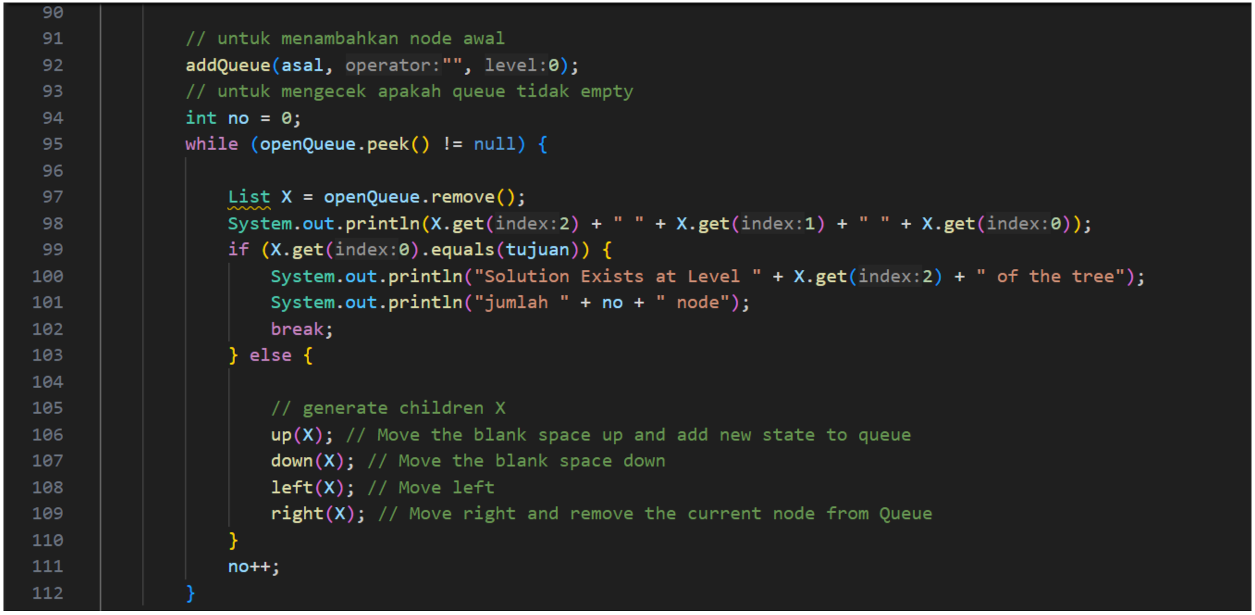
```java
            String s = str.substring(beginIndex:0, a - 3) + "0" + str.substring(a - 2, a) + str.charAt(a - 3)
                    + str.substring(a + 1);
            // System.out.print("(" + (map.get(str) + 1)+","+b+")");
            // System.out.print(map.get(str) + 1);
            // System.out.println(" up " + s);
            Integer level = (Integer) node.get(index:2) + 1;

            addQueue(s, operator:"up", level);
        }
        // else
        // System.out.println("(" + (map.get(str) + 1)+","+b+") up tidak ada");
    }

    void down(List node) {
        String str = (String) node.get(index:0);
        int a = str.indexOf(str:"0");
        if (a < 6) {
            String s = str.substring(beginIndex:0, a) + str.substring(a + 3, a + 4) + str.substring(a + 1, a + 3)
                    + str.substring(a + 4);
            // System.out.print("(" + (map.get(str) + 1)+","+b+")");
            // System.out.print(map.get(str) + 1);
```



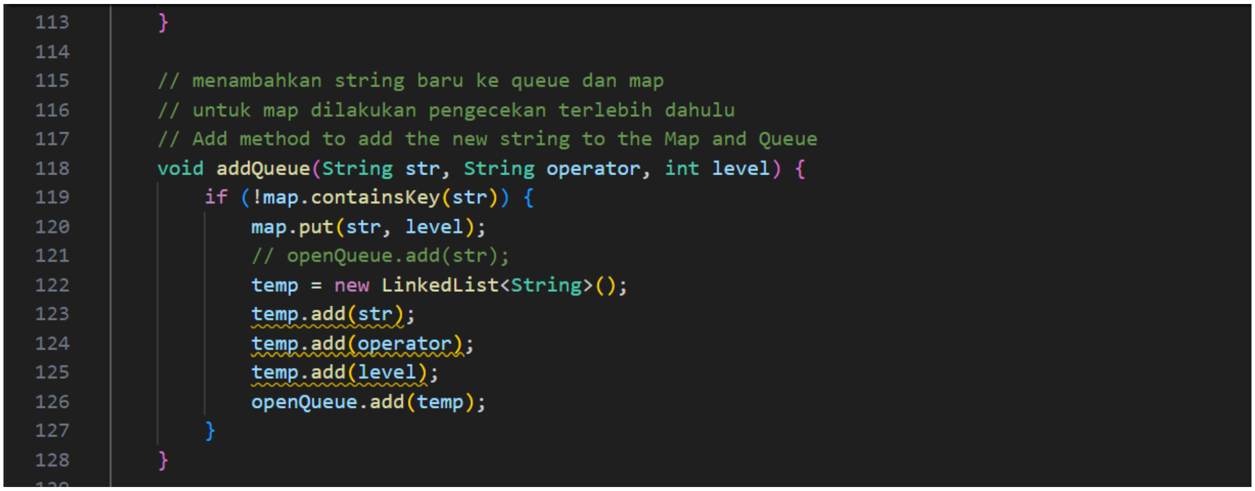
```java
            // System.out.print(map.get(str) + 1);
            // System.out.println(" down " + s);

            Integer level = (Integer) node.get(index:2) + 1;
            addQueue(s, operator:"down", level);

        }
        // else
        // System.out.println("(" + (map.get(str) + 1)+","+b+") down tidak ada");
    }

    void left(List node) {
        String str = (String) node.get(index:0);
        int a = str.indexOf(str:"0");
        if (a != 0 && a != 3 && a != 6) {
            String s = str.substring(beginIndex:0, a - 1) + "0" + str.charAt(a - 1) + str.substring(a + 1);
            // System.out.print("(" + (map.get(str) + 1)+","+b+")");
            // System.out.print(map.get(str) + 1);
            // System.out.println(" left " + s);

            Integer level = (Integer) node.get(index:2) + 1;
            addQueue(s, operator:"left", level);
        }
```



```
        // else
        // System.out.println("(" + (map.get(str) + 1)+","+b+") left tidak ada");
    }

    void right(List node) {
        String str = (String) node.get(index:0);
        int a = str.indexOf(str:"0");
        if (a != 2 && a != 5 && a != 8) {
            String s = str.substring(beginIndex:0, a) + str.charAt(a + 1) + "0" + str.substring(a + 2);
            // System.out.print("(" + (map.get(str) + 1)+","+b+")");
            // System.out.print(map.get(str) + 1);
            // System.out.println(" right " + s);

            Integer level = (Integer) node.get(index:2) + 1;
            addQueue(s, operator:"right", level);

        }
        // else
        // System.out.println("(" + (map.get(str) + 1)+","+b+") right tidak ada");
    }

    public void bfs() // breadth-first search
    {
```

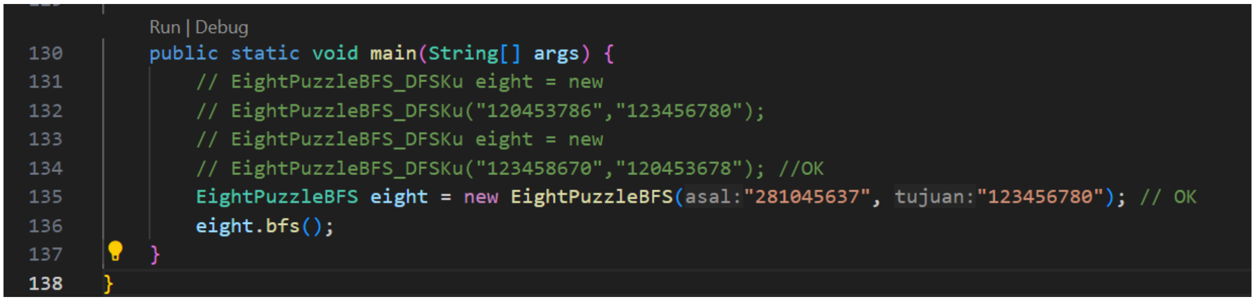


```

        // untuk menambahkan node awal
        addQueue(asal, operator:"", level:0);
        // untuk mengecek apakah queue tidak empty
        int no = 0;
        while (openQueue.peek() != null) {

            List X = openQueue.remove();
            System.out.println(X.get(index:2) + " " + X.get(index:1) + " " + X.get(index:0));
            if (X.get(index:0).equals(tujuan)) {
                System.out.println("Solution Exists at Level " + X.get(index:2) + " of the tree");
                System.out.println("jumlah " + no + " node");
                break;
            } else {

                // generate children X
                up(X); // Move the blank space up and add new state to queue
                down(X); // Move the blank space down
                left(X); // Move left
                right(X); // Move right and remove the current node from Queue
            }
            no++;
        }
```

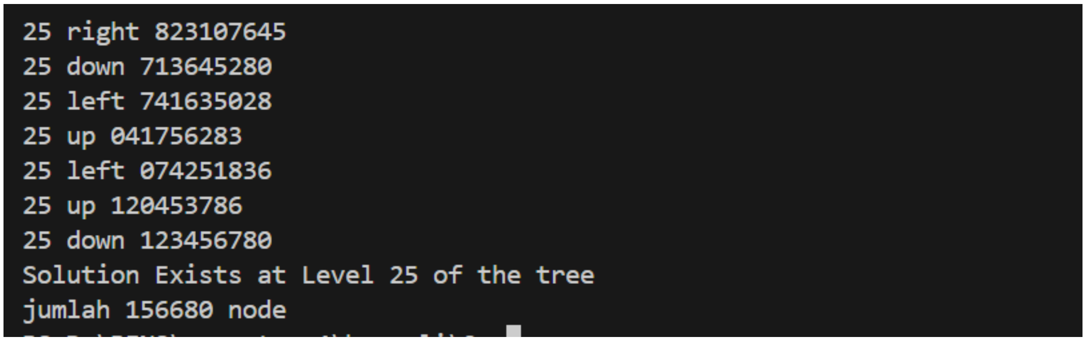


```java
    }

    // menambahkan string baru ke queue dan map
    // untuk map dilakukan pengecekan terlebih dahulu
    // Add method to add the new string to the Map and Queue
    void addQueue(String str, String operator, int level) {
        if (!map.containsKey(str)) {
            map.put(str, level);
            // openQueue.add(str);
            temp = new LinkedList<String>();
            temp.add(str);
            temp.add(operator);
            temp.add(level);
            openQueue.add(temp);
        }
    }
```

Percobaan ke 1

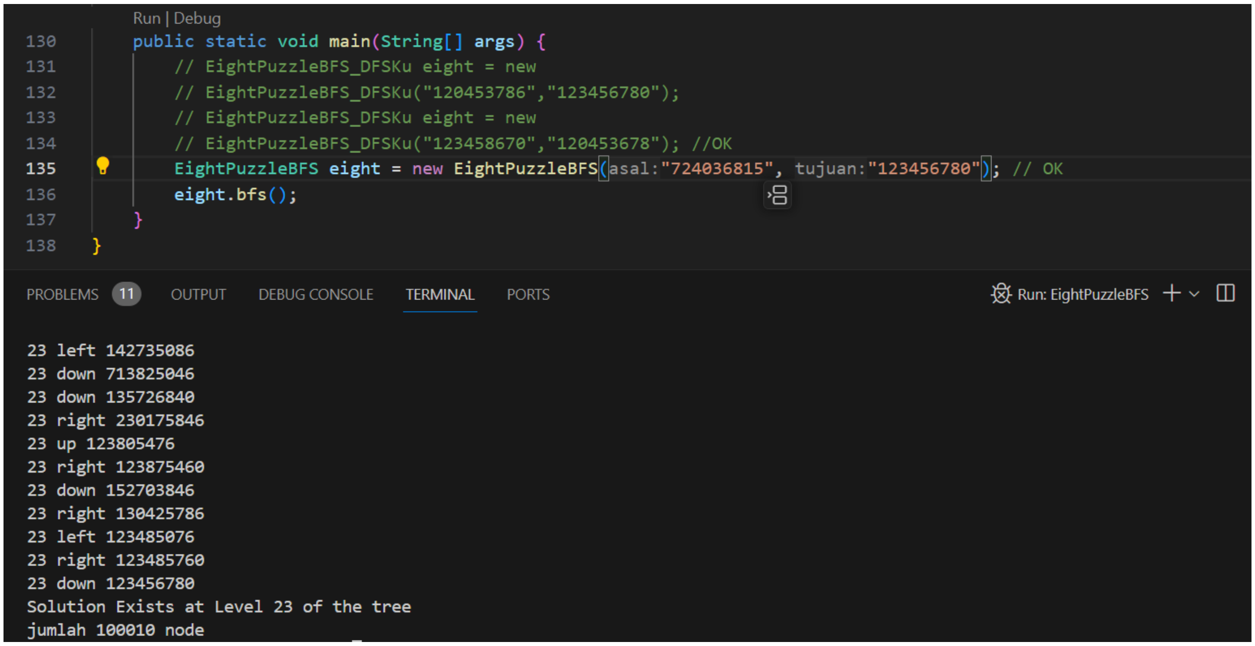


```java
    Run | Debug
    public static void main(String[] args) {
        // EightPuzzleBFS_DFSKu eight = new
        // EightPuzzleBFS_DFSKu("120453786","123456780");
        // EightPuzzleBFS_DFSKu eight = new
        // EightPuzzleBFS_DFSKu("123458670","120453678"); //OK
        EightPuzzleBFS eight = new EightPuzzleBFS(asal:"281045637", tujuan:"123456780"); // OK
        eight.bfs();
    }
}
```



```
25 right 823107645
25 down 713645280
25 left 741635028
25 up 041756283
25 left 074251836
25 up 120453786
25 down 123456780
Solution Exists at Level 25 of the tree
jumlah 156680 node
```

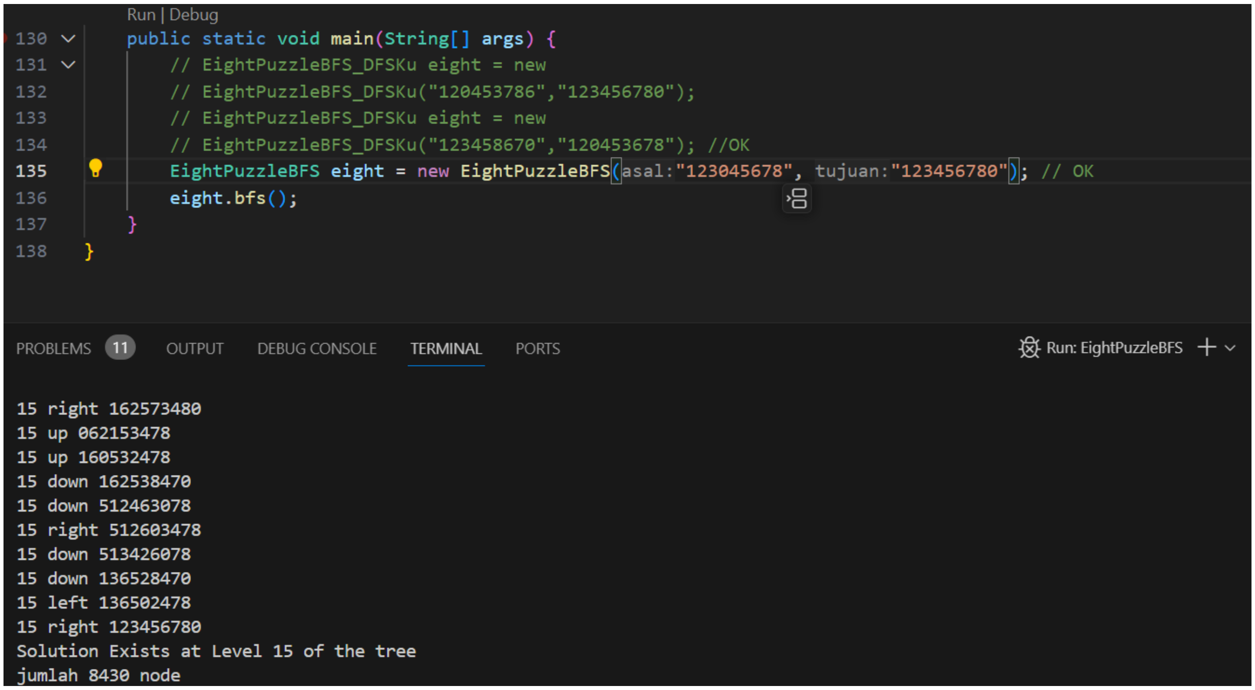
Percobaan ke 2



```
public static void main(String[] args) {
    // EightPuzzleBFS_DFSKu eight = new
    // EightPuzzleBFS_DFSKu("120453786","123456780");
    // EightPuzzleBFS_DFSKu eight = new
    // EightPuzzleBFS_DFSKu("123458670","120453678"); //OK
    EightPuzzleBFS eight = new EightPuzzleBFS(asal:"724036815", tujuan:"123456780"); // OK
    eight.bfs();
}
}
```

```
23 left 142735086
23 down 713825046
23 down 135726840
23 right 230175846
23 up 123805476
23 right 123875460
23 down 152703846
23 right 130425786
23 left 123485076
23 right 123485760
23 down 123456780
Solution Exists at Level 23 of the tree
jumlah 100010 node
```

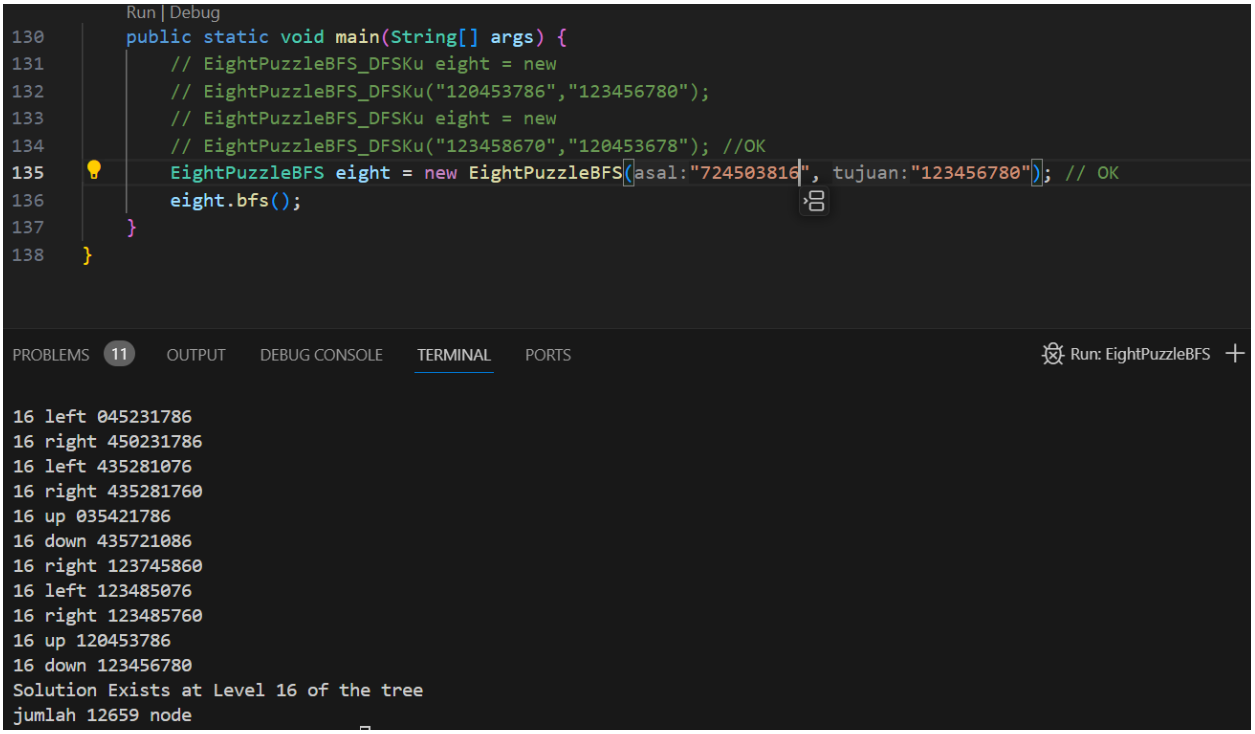
Percobaan ke 3



```
public static void main(String[] args) {
    // EightPuzzleBFS_DFSKu eight = new
    // EightPuzzleBFS_DFSKu("120453786","123456780");
    // EightPuzzleBFS_DFSKu eight = new
    // EightPuzzleBFS_DFSKu("123458670","120453678"); //OK
    EightPuzzleBFS eight = new EightPuzzleBFS(asal:"123045678", tujuan:"123456780"); // OK
    eight.bfs();
}
}
```

```
15 right 162573480
15 up 062153478
15 up 160532478
15 down 162538470
15 down 512463078
15 right 512603478
15 down 513426078
15 down 136528470
15 left 136502478
15 right 123456780
Solution Exists at Level 15 of the tree
jumlah 8430 node
```

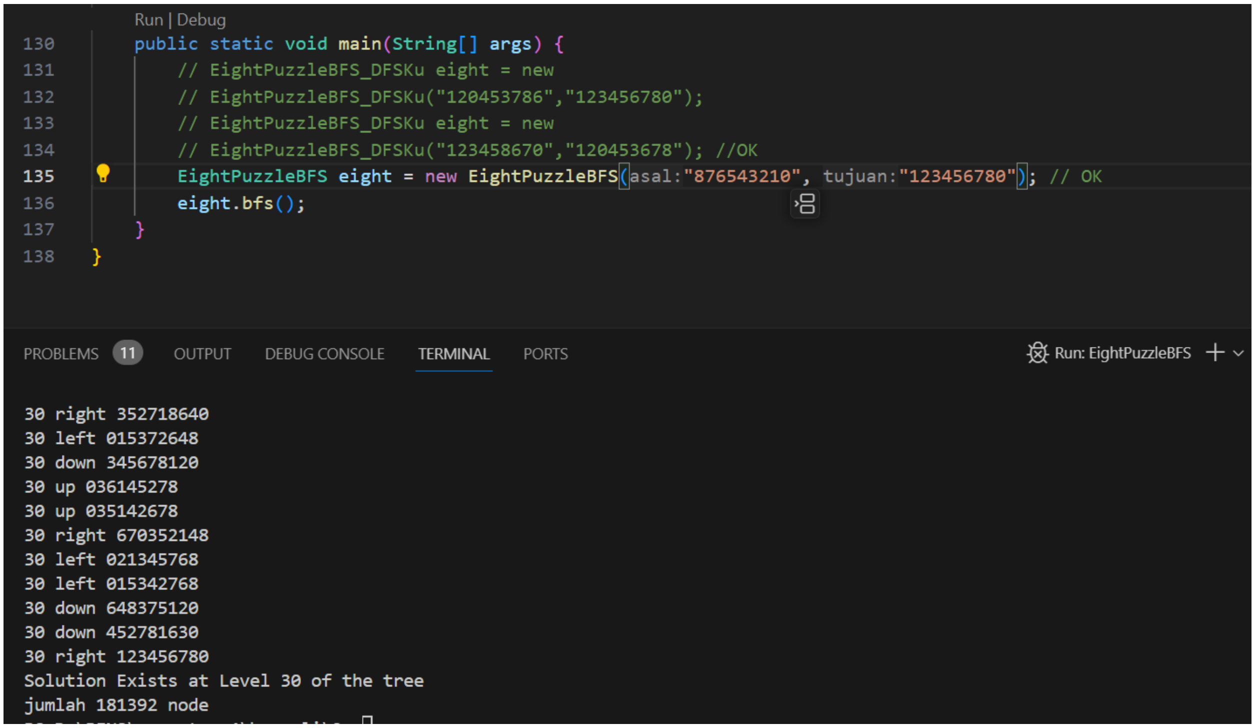
Percobaan ke 4



```java
    public static void main(String[] args) {
        // EightPuzzleBFS_DFSKu eight = new
        // EightPuzzleBFS_DFSKu("120453786","123456780");
        // EightPuzzleBFS_DFSKu eight = new
        // EightPuzzleBFS_DFSKu("123458670","120453678"); //OK
        EightPuzzleBFS eight = new EightPuzzleBFS(asal:"724503816", tujuan:"123456780"); // OK
        eight.bfs();
    }
}
```

```
16 left 045231786
16 right 450231786
16 left 435281076
16 right 435281760
16 up 035421786
16 down 435721086
16 right 123745860
16 left 123485076
16 right 123485760
16 up 120453786
16 down 123456780
Solution Exists at Level 16 of the tree
jumlah 12659 node
```

Percobaan ke 5



```java
    public static void main(String[] args) {
        // EightPuzzleBFS_DFSKu eight = new
        // EightPuzzleBFS_DFSKu("120453786","123456780");
        // EightPuzzleBFS_DFSKu eight = new
        // EightPuzzleBFS_DFSKu("123458670","120453678"); //OK
        EightPuzzleBFS eight = new EightPuzzleBFS(asal:"876543210", tujuan:"123456780"); // OK
        eight.bfs();
    }
}
```

```
30 right 352718640
30 left 015372648
30 down 345678120
30 up 036145278
30 up 035142678
30 right 670352148
30 left 021345768
30 left 015342768
30 down 648375120
30 down 452781630
30 right 123456780
Solution Exists at Level 30 of the tree
jumlah 181392 node
```