LAPORAN TUGAS KECIL I

IF2211 STRATEGI ALGORITMA

Penyelesaian IQPuzzler dengan Algoritma Brute Force

Disusun oleh:

Ferdinand Gabe Tua Sinaga 13523051

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2025

Daftar Isi

BAGIAN I ALGORITMA BRUTE FORCE

Algoritma Brute Force adalah salah satu jenis algoritma yang hampir dapat diaplikasikan dalam semua permasalahan termasuk permasalahan seperti IQPuzzler. Algoritma ini bekerja dengan menguji setiap kemungkinan konfigurasi dari potongan puzzle, termasuk posisi, rotasi, dan refleksi masing-masing potongan, hingga menemukan susunan yang sesuai dengan aturan permainan. Dalam permainan ini Algoritma Brute Force akan melakukan beberapa langkah berikut:

Ambil satu potongan puzzle pertama dan coba tempatkan dalam grid permainan di posisi awal (0,0).

1. Jika tidak dapat ditempatkan (karena bertabrakan atau keluar dari grid), geser ke kanan satu langkah.
2. Jika mencapai batas kanan, geser ke baris berikutnya dan mulai dari kiri lagi.
3. Jika semua posisi telah dicoba dan tetap tidak bisa, lakukan rotasi dan ulangi proses dari awal.
4. Jika setelah semua rotasi tetap tidak bisa, lakukan pencerminan dan ulangi proses.
5. Jika potongan dapat ditempatkan, lanjutkan ke potongan berikutnya dan ulangi langkah-langkah di atas.
6. Jika seluruh potongan telah ditempatkan dengan benar, solusi ditemukan.
7. Jika suatu potongan tidak bisa ditempatkan setelah mencoba semua posisi, rotasi, dan pencerminan, kembalilah ke potongan sebelumnya dan cari posisi alternatif lain untuk potongan tersebut.
8. Ulangi proses ini hingga semua kemungkinan kombinasi telah diuji atau solusi ditemukan.

Dengan menggunakan pendekatan tersebut kita akan hampir pasti selalu mendapatkan solusi jika memang puzzle tersebut dapat diselesaikan.

BAGIAN II
SOURCE CODE

Pada Bab 1 sudah dijelaskan mengenai langkah-langkah untuk menyelesaikan puzzle tersebut, dalam program kali ini langkah langkah tersebut akan disebar ke beberapa file utama untuk meningkatkan modularitas. File-file tersebut diantaranya adalah file GameInfo.java, Solver.java, Point.java dan PuzzlePiece.java. Diantara ke 4 file tersebut mayoritas dari algoritma Solver,java. Sehingga source program yang akan ditampilkan hanya file tersebut saja.

Berikut Source Codenya pada kelas Solver:

```
public void trysolve(List<PuzzlePiece> listpuzzle) {
  if (done || isBoardFull()) {
      this.done = true;
      System.out.println(x:"Puzzle solved!");
      return;
  }

  if (listpuzzle.isEmpty()) {
      return;
  }

  PuzzlePiece head = listpuzzle.get(index:0);
  List<PuzzlePiece> tail = new ArrayList<>(listpuzzle.subList(fromIndex:1, listpuzzle.size()));

  for (int mirror = 0; mirror < 2; mirror++) {
      for (int rotation = 0; rotation < 4; rotation++) {
          head.normalize();


          for (int i = 0; i < solutionBoard.length; i++) {
              for (int j = 0; j < solutionBoard[0].length; j++) {
                  if (this.done) return;

                  head.shiftTo(i, j);

                  if (canPlace(head)) {
                      placePiece(head);
```

```
                        printSolution();
                        trysolve(tail);
                        if (done || isBoardFull()) return;
                        this.casesChecked++;
                        removePiece(head);
                    }
                }
            }
            head.rotate90(head.getPiece());
        }

        if (mirror == 0) {
            head.mirrorHorizontally();
        } else if (mirror == 1) {
            head.mirrorVertically();
        }
    }
}
```

```
private void placePiece(PuzzlePiece piece) {
  List<Point> puzzle = piece.getPiece();
  for (int i = 0; i < puzzle.size(); i++) {
      int row = puzzle.get(i).getPoint1();
      int column = puzzle.get(i).getPoint2();
      this.solutionBoard[row][column] = 1;
  }
}

private void removePiece(PuzzlePiece piece) {
  List<Point> puzzle = piece.getPiece();
  for (int i = 0; i < puzzle.size(); i++) {
      int row = puzzle.get(i).getPoint1();
      int column = puzzle.get(i).getPoint2();
      this.solutionBoard[row][column] = 0;
  }
}
```

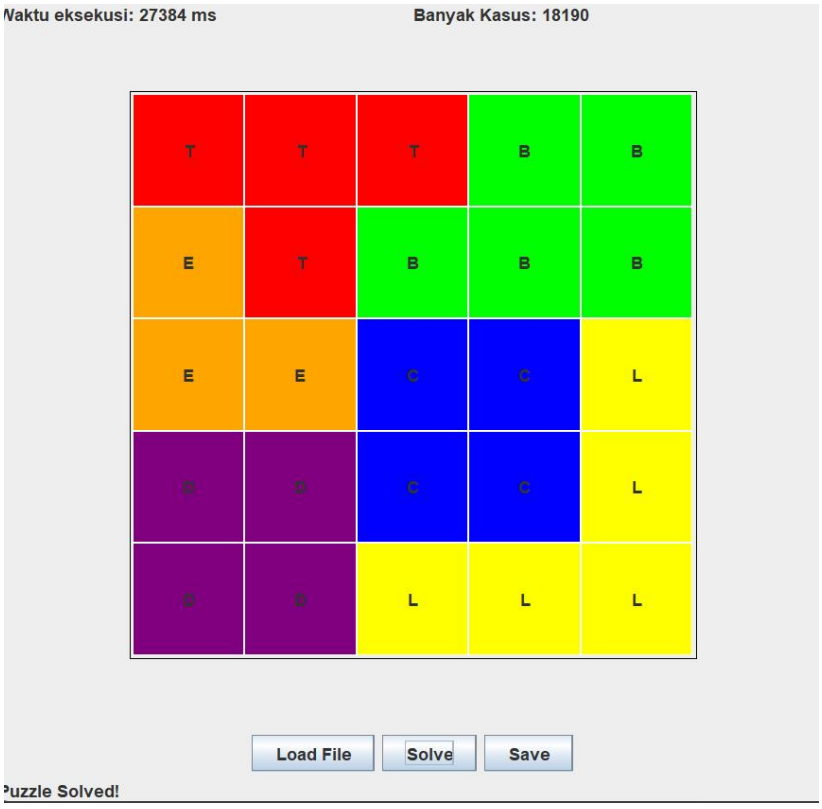
```java
private boolean canPlace(PuzzlePiece piece) {
  List<Point> puzzle = piece.getPiece();
  for (int i=0; i<puzzle.size(); i++){
    int row = puzzle.get(i).getPoint1();
    int column = puzzle.get(i).getPoint2();
    if (row < 0 || row >= solutionBoard.length || column < 0 || column >= solutionBoard[0].length) {
      return false;
    }
    if (solutionBoard[row][column] == 1 || solutionBoard[row][column] == -1){
        return false;
    }
  }
  return true;
}

private boolean isBoardFull() {
  for (int i = 0; i < this.solutionBoard.length; i++) {
      for (int j = 0; j < this.solutionBoard[0].length; j++) {
          if (solutionBoard[i][j] == 0 ) {
              return false;
          }
      }
  }

  return true;
```

BAGIAN III SCREENSHOT
HASIL TEST

TC1

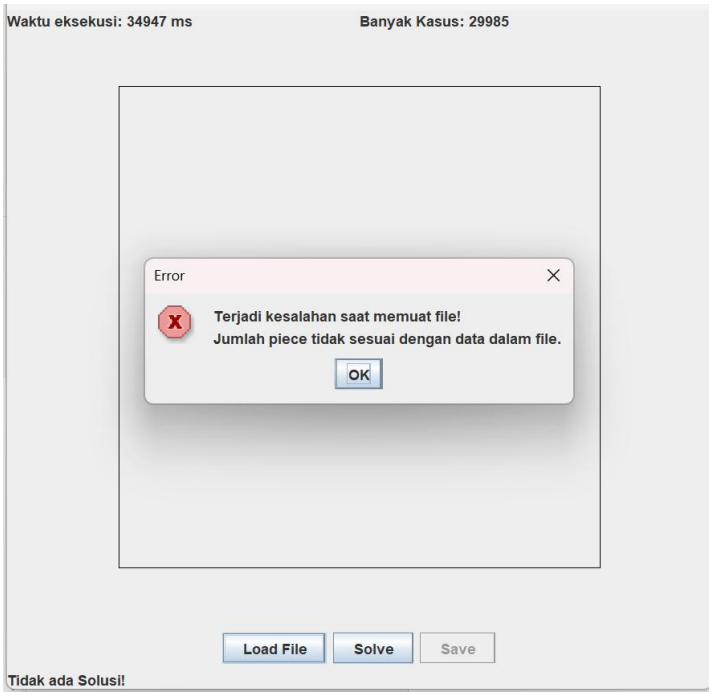


Input:
5 5 6
DEFAULT
TTT
 T
BB
BBB
CC
CC
L
L
LLL
E
EE
DD
DD

TC2

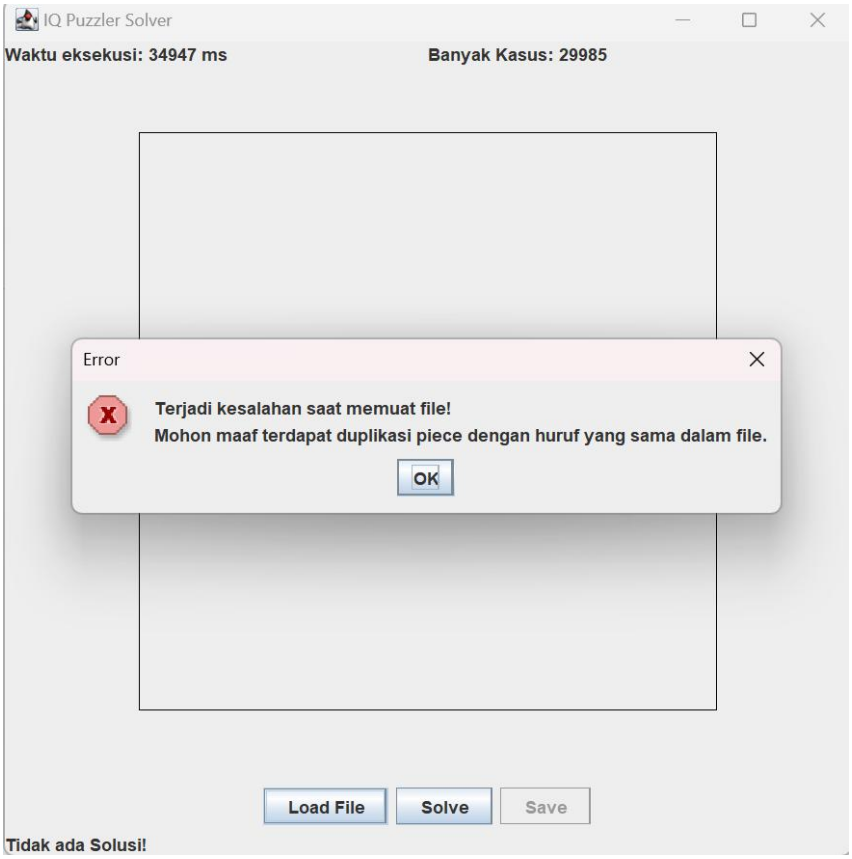
Inputnya:
4 4 7
DEFAULT
 T
TTT
 T
KK
BBBB
C
C
AA
D
E

TC3



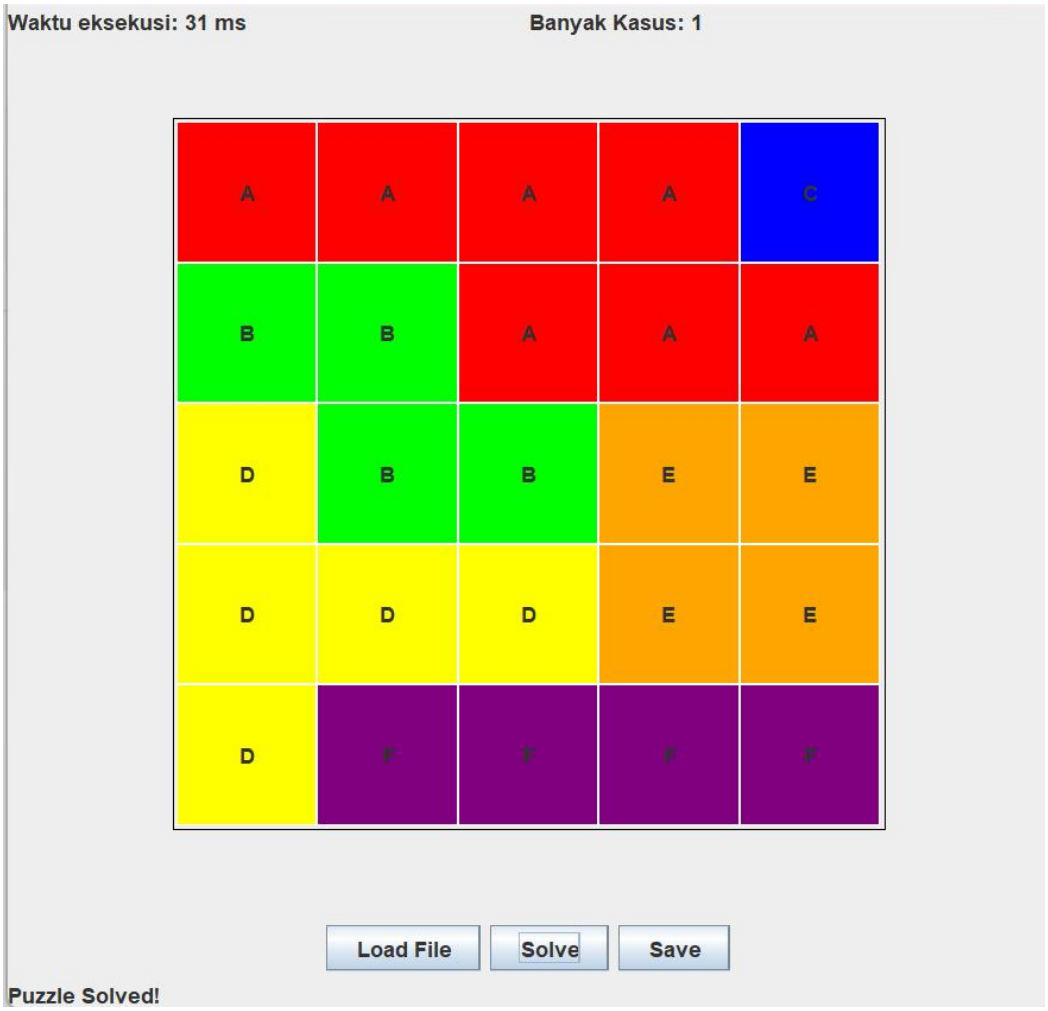
inputnya
5 5 7
DEFAULT
TTT
 T
BB
BBB
CC
CC
L
L
LLL
E
EE
DD
DD

TC4



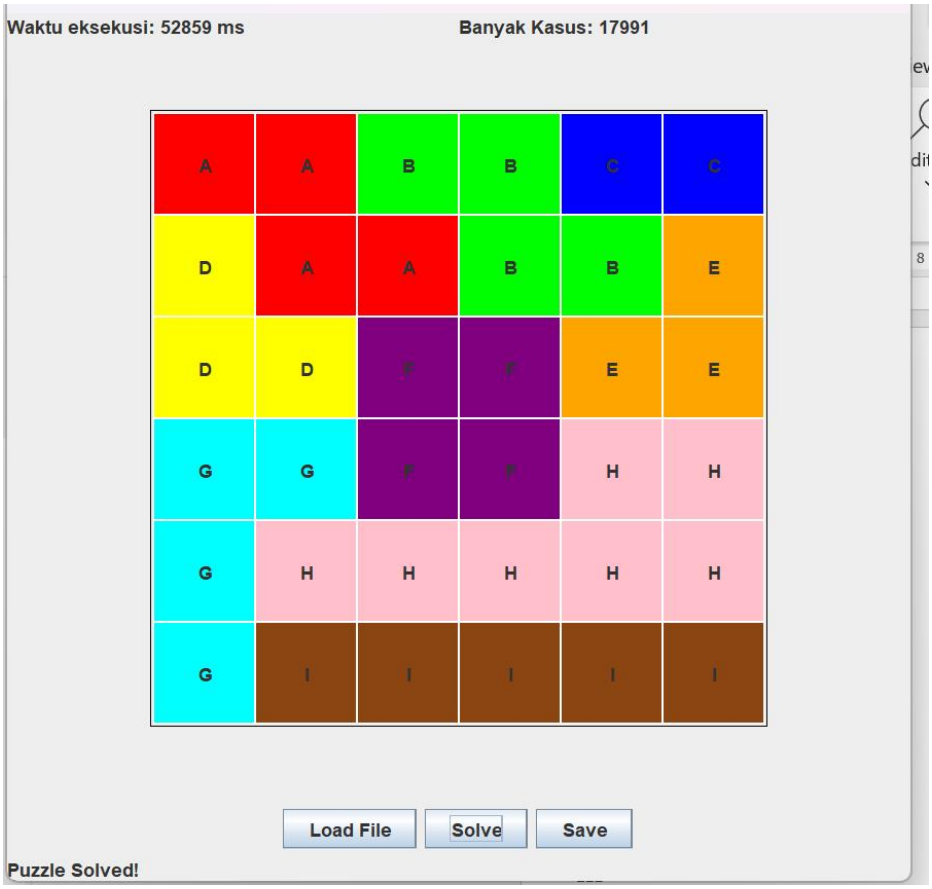
Inputnya:
6 5 5
DEFAULT
TTT
 T
BB
BBB
CC
CC
L
L
LLL
E
EE
CC
CC

TC 5



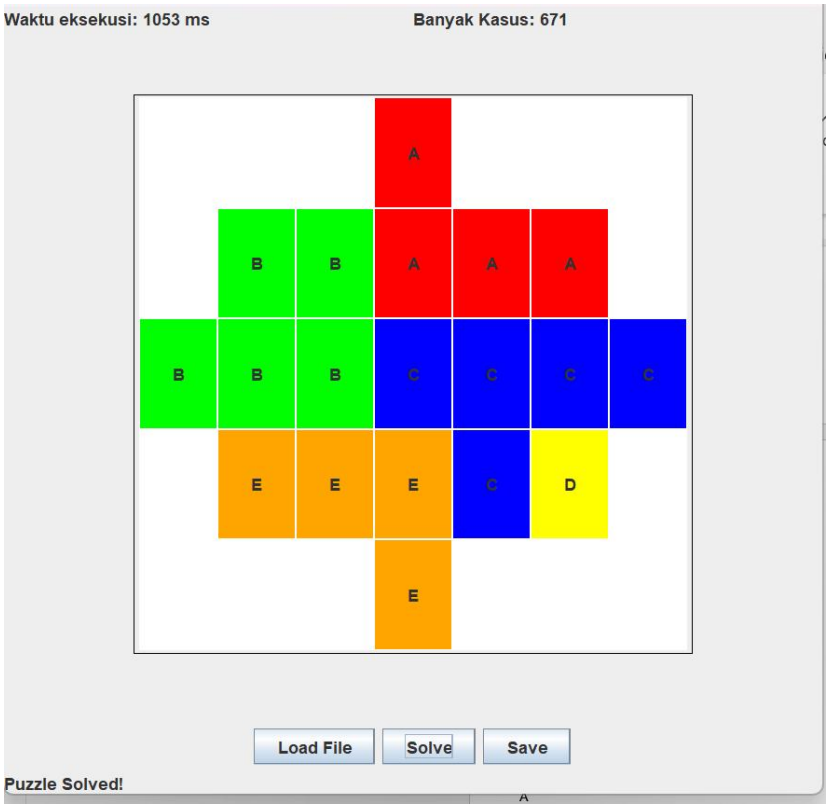
Inputnya:
5 5 6
DEFAULT
AAAA
 AAA
BB
 BB
C
D
DDD
D
EE
EE
FFFF

TC6



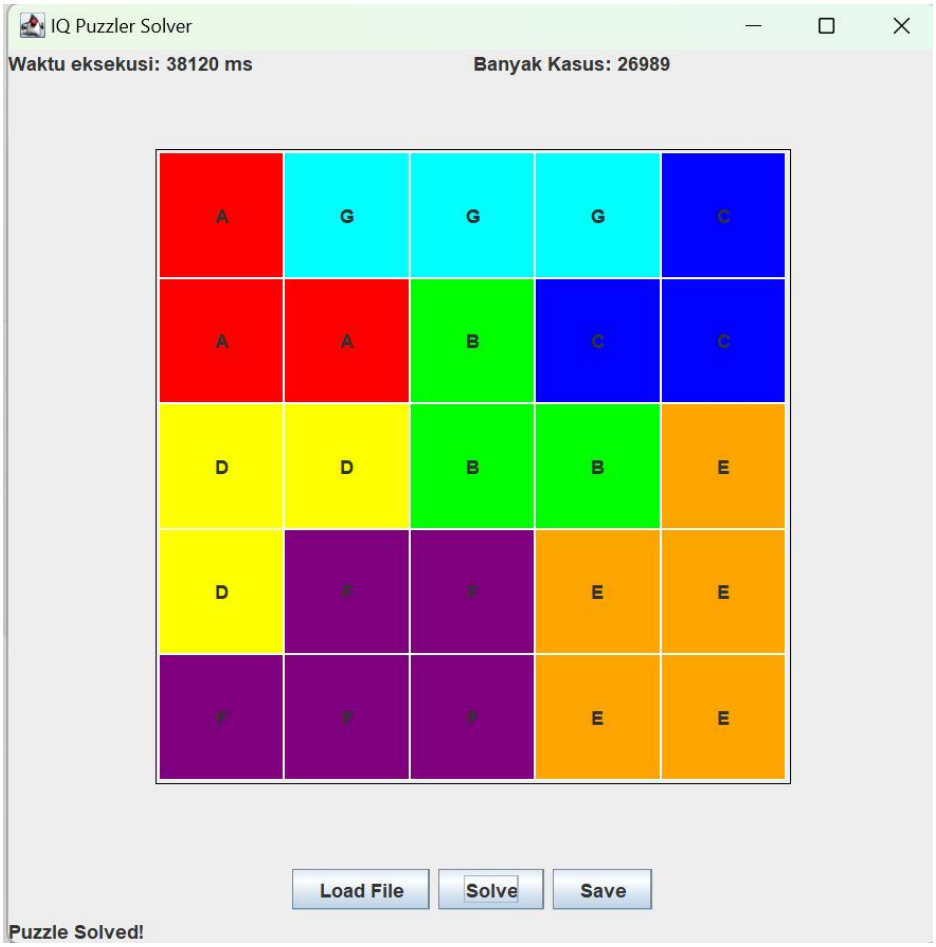
Inputnya:
6 6 9
DEFAULT
AA
 AA
BB
 BB
CC
D
DD
E
EE
FF
FF
GG
G
G
HHHHH
HH
IIIII

TC7



Inputnya:
5 7 5
CUSTOM
...X...
.XXXXX.
XXXXXXX
.XXXXX.
...X...
A
AAA
BB
BBB
CCCC
C
D
EEE
E

TC8



Inputnya:
5 5 7
DEFAULT
A
AA
B
BB
C
CC
D
DD
EE
EE
E
FF
FF
F
GGG

CHECKLIST

PRANALA GITHUB:
https://github.com/FerdinandGabe1805/Tucil1_13523051

No	Poin	Ya	Tidak
1	Program berhasil dikompilasi tanpa kesalahan	✓	
2	Program berhasil dijalankan	✓	
3	Solusi yang diberikan program benar dan mematuhi aturan permainan	✓	
4	Program dapat membaca masukan berkas .txt serta menyimpan solusi dalam berkas .txt	✓	
5	Program memiliki Graphical User Interface	✓	
6	Program dapat menyimpan solusi dalam bentuk file gambar		✓
7	Program dapat menyelesaikan kasus custom	✓	
8	Program dapat menyelesaikan Kasus konfigurasi piramida		✓
9	Program dibuat oleh saya sendiri	✓	