Laporan Tugas Kecil 1 IF2211 Strategi Algoritma

Semester II tahun 2024/2025

Penyelesaian *IQ Puzzler Pro* dengan Algoritma Brute Force

Disusun oleh:

Raka Daffa Iftikhaar - 13523018

Program Studi S1 Teknik Informatika

Sekolah Teknik Elektro dan Informatika

2025

A. Algoritma *Brute Force* dalam Mencari Solusi *Puzzle*

Permainan IQ Puzzle Pro adalah permainan pencocokan sebuah bentuk puzzle pada sebuah bidang dengan ukuran tertentu. Permainan ini menuntut kemampuan untuk memecahkan puzzle. Permainan ini bisa diselesaikan dengan algoritma brute force. Algoritma brute force adalah sebuah algoritma untuk menyelesaikan sesuatu dan tidak akan berhenti berjalan sampai mendapatkan solusi akhir. Penerapan algoritma brute force pada program ini dilakukan dengan beberapa tahapan.

1. Program menerima masukan file dalam format .txt. File .txt itu berisi ukuran grid/wadah, jumlah bentuk/komponen, bentuk penyusunan, dan juga komponen dalam bentuk huruf. File tersebut akan diambil komponen pentingnya seperti yang sudah disebutkan diatas. Pemrosesan ini dilakukan pada file `PuzzleInput.java` dan untuk mengambil komponen-komponen pentingnya dilakukan dengan menggunakan fungsi `bacaPuzzle`. Fungsi tersebut akan mengembalikan panjang wadah, lebar wadah, dan bentuk-bentuk puzzle.
2. Terdapat dua objek yang sudah dibuat dalam program ini, yaitu objek Bentuk dalam file `Bentuk.java` dan juga objek Wadah dalam file `Wadah.java`.
3. Objek bentuk dapat menyimpan bentuk dalam file .txt menjadi sebuah koordinat dan disimpan dalam bentuk list. Objek ini juga bisa melakukan rotasi 90°. 180°, dan 270°. Objek ini juga bisa melakukan pencerminan bentuk secara horizontal dan vertikal. Rotasi dan pencerminan ini dibantu dengan satu file lainnya yaitu `UbahBentuk.java`. Penyimpanan hasil dari rotasi dan pencerminan bentuk tersebut dilakukan dengan menggunakan Set dalam fungsi `addIfUnique` dan `getUniqueKey`.
4. Objek wadah sedikit lebih rumit dari objek bentuk. Objek ini dapat melakukan banyak hal seperti print hasil grid pada fungsi `printWadah`, memberikan warna pada bentuk, mengecek tempat kosong di grid pada fungsi `cekBentuk` dan `sudahTerisi`, serta menambah dan menghapus bentuk pada fungsi `place` dan `hapus`.
5. Algoritma brute force utama berada pada file Bruteforce.java. Objek yang bernama Bruteforce dapat melakukan pengecekan tiap bentuk yang sesuai untuk mengisi grid/wadah yang ada. Proses brute dilakukan dengan beberapa tahap, diantaranya:
    a. Inisiasi objek dan variabel seperti list untuk menyimpan bentuk awal dan list dalam list untuk menyimpan variasi dari tiap bentuk.
    b. Proses brute force dilakukan dengan nested loop sebanyak tiga kali. Loop pertama dilakukan sebanyak variasi bentuk yang ada, loop kedua dan ketiga sesuai dengan ukuran grid/wadah untuk mengecek setiap kolom/koordinat.
    c. Bila saat melakukan pengecekan dan dihasilkan true dari fungsi cekBentuk, maka variasi tersebut akan ditempatkan pada grid/wadah, namun bila tidak maka bentuk akan dihapus dan dilanjutkan variasi bentuk lainnya.
    d. Proses tersebut dilakukan secara rekursif sampai semua variasi bentuk dicoba untuk ditempatkan. Hasil akhir dari algoritma ini adalah ditemukannya hasil yang sesuai atau tidak ditemukan hasil sama sekali.
    e. Hasil tersebut dicek dengan menghitung jumlah huruf yang berhasil ditempatkan di wadah/grid. Bila jumlah huruf sesuai dengan hasil kali dari kolom dan baris maka dapat ditentukan bahwa didapatkan hasil, namun bila tidak berarti hasil tidak didapatkan.

6. Pemrosesan terakhir terdapat pada file `Main.java`. Pada file ini, semua interaksi dengan pengguna akan dilakukan. Dimulai dengan memasukkan nama file, lalu pemanggilan objek Bruteforce, lalu penghitungan waktu proses. Semua hal tersebut akan ditampilkan bersama output berupa hasil puzzle yang telah diselesaikan, waktu proses, jumlah kasus, dan opsi untuk menyimpan hasil berupa file .txt ataupun gambar dengan format .png.

B. Source Code Program

1. File Bentuk.java

```java
import java.util.*;

public class Bentuk {
  private final List<int[]> koordinat;
  private final char huruf;

  public Bentuk(List<int[]> koordinatBaru, char huruf) {
    this.koordinat = new ArrayList<>(koordinatBaru.size());
    for (int[] k : koordinatBaru) {
      this.koordinat.add(new int[] { k[0], k[1] });
    }
    this.huruf = huruf;
  }

  public List<int[]> getKoordinat() {
    return koordinat;
  }

  public char getHuruf() {
    return huruf;
  }

  public List<Bentuk> buatVariasi() {
    Set<String> bentukUnique = new HashSet<>();
    List<Bentuk> variasi = new ArrayList<>();
    Bentuk reflectedX = UbahBentuk.cerminAtas(this);
    Bentuk reflectedY = UbahBentuk.cerminSamping(this);

    addIfUnique(this, bentukUnique, variasi);
    generateRotations(this, bentukUnique, variasi);
    generateRotations(reflectedX, bentukUnique, variasi);
```

```
    generateRotations(reflectedY, bentukUnique, variasi);

    return variasi;
  }

   private void generateRotations(Bentuk awal, Set<String> bentukUnique,
List<Bentuk> variasi) {
    if (!addIfUnique(awal, bentukUnique, variasi)) {
      return;
    }

    Bentuk current = awal;
    for (int i = 0; i < 3; i++) {
      current = UbahBentuk.putar90(current);
      addIfUnique(current, bentukUnique, variasi);
    }
  }

   private boolean addIfUnique(Bentuk bentuk, Set<String> bentukUnique,
List<Bentuk> variasi) {
    String key = getUniqueKey(bentuk);
    if (bentukUnique.add(key)) {
      variasi.add(bentuk);
      return true;
    }
    return false;
  }

  private String getUniqueKey(Bentuk bentuk) {
    StringBuilder sb = new StringBuilder();
    List<int[]> koordinat = new ArrayList<>(bentuk.getKoordinat());

    koordinat.sort((a, b) -> a[0] != b[0] ? a[0] - b[0] : a[1] - b[1]);

    for (int[] k : koordinat) {
      sb.append(k[0]).append(',').append(k[1]).append(';');
    }
    return sb.toString();
  }
```

```
}
```

2. File Wadah.java

```
import java.util.*;

public class Wadah {
  private Map<String, Character> mapWadah;
  private int baris, kolom;
  private static final String[] COLORS = {
      "\u001B[30m", // BLACK
      "\u001B[31m", // RED
      "\u001B[32m", // GREEN
      "\u001B[33m", // YELLOW
      "\u001B[34m", // BLUE
      "\u001B[35m", // MAGENTA
      "\u001B[36m", // CYAN
      "\u001B[37m", // WHITE
      "\u001B[90m", // BRIGHT BLACK (GRAY)
      "\u001B[91m", // BRIGHT RED
      "\u001B[92m", // BRIGHT GREEN
      "\u001B[93m", // BRIGHT YELLOW
      "\u001B[94m", // BRIGHT BLUE
      "\u001B[95m", // BRIGHT MAGENTA
      "\u001B[96m", // BRIGHT CYAN
      "\u001B[97m", // BRIGHT WHITE
      "\u001B[1;30m", // BOLD BLACK
      "\u001B[1;31m", // BOLD RED
      "\u001B[1;32m", // BOLD GREEN
      "\u001B[1;33m", // BOLD YELLOW
      "\u001B[1;34m", // BOLD BLUE
      "\u001B[1;35m", // BOLD MAGENTA
      "\u001B[1;36m", // BOLD CYAN
      "\u001B[1;37m", // BOLD WHITE
      "\u001B[3;30m", // ITALIC BLACK
      "\u001B[3;31m" // ITALIC RED
  };
  private static final String RESET = "\u001B[0m";
  private Map<Character, String> mapWarna;
```

```java
  public Wadah(int baris, int kolom) {
    this.baris = baris;
    this.kolom = kolom;
    this.mapWadah = new HashMap<>();
    this.mapWarna = new HashMap<>();
  }

  public void printWadah() {
    Set<Character> uniqueChars = new HashSet<>(mapWadah.values());
    int colorIdx = mapWarna.size();

    for (Character c : uniqueChars) {
      if (!mapWarna.containsKey(c)) {
        mapWarna.put(c, COLORS[colorIdx % COLORS.length]);
        colorIdx++;
      }
    }

    for (int i = 0; i < baris; i++) {
      for (int j = 0; j < kolom; j++) {
        char c = getCharAt(i, j);
        if (c != '.') {
          String color = mapWarna.get(c);
          System.out.print(color + c + RESET + " ");
        } else {
          System.out.print(c + " ");
        }
      }
      System.out.println();
    }
  }

  public boolean cekBentuk(Bentuk bentuk, int x, int y) {
    for (int[] k : bentuk.getKoordinat()) {
      int newX = x + k[0];
      int newY = y + k[1];
        if (newX < 0 || newX >= baris || newY < 0 || newY >= kolom ||
sudahTerisi(newX, newY)) {
        return false;
```

```
      }
    }
    return true;
  }

  public boolean sudahTerisi(int x, int y) {
    return mapWadah.containsKey(x + "," + y);
  }

  public void place(Bentuk bentuk, int x, int y) {
    for (int[] k : bentuk.getKoordinat()) {
      String pos = (x + k[0]) + "," + (y + k[1]);
      mapWadah.put(pos, bentuk.getHuruf());
    }
  }

  public void hapus(Bentuk bentuk, int x, int y) {
    for (int[] k : bentuk.getKoordinat()) {
      String pos = (x + k[0]) + "," + (y + k[1]);
      mapWadah.remove(pos);
    }
  }

  public int getRows() {
    return baris;
  }

  public int getCols() {
    return kolom;
  }

  public char getCharAt(int x, int y) {
    String pos = x + "," + y;
    return mapWadah.getOrDefault(pos, '.');
  }
}
```

3. File PuzzleInput.java

```
import java.io.*;
```

```java
import java.util.*;

public class PuzzleInput {
  public static class PuzzleData {
    public int baris, kolom;
    public List<Bentuk> bentukList;

    public PuzzleData(int baris, int kolom, List<Bentuk> bentukList) {
      this.baris = baris;
      this.kolom = kolom;
      this.bentukList = bentukList;
    }

    public int getCountBentuk() {
      Set<Character> uniqueChars = new HashSet<>();
      for (Bentuk bentuk : bentukList) {
        uniqueChars.add(bentuk.getHuruf());
      }
      return uniqueChars.size();
    }
  }

   public static PuzzleData bacaPuzzle(String fileName) throws IOException
{
    List<Bentuk> bentuk = new ArrayList<>();
    int panjang = 0;
    int lebar = 0;
    int jumlahBentuk = 0;

    try (BufferedReader br = new BufferedReader(new FileReader(fileName)))
{
      String baris1 = br.readLine();
      if (baris1 == null) {
        throw new IllegalArgumentException("Baris pertama kosong");
      }

      String[] ukuran = baris1.split(" ");
      if (ukuran.length != 3) {
```

```
            throw new IllegalArgumentException("Komponen baris pertama
kurang");
      }

      try {
        panjang = Integer.parseInt(ukuran[0]);
        lebar = Integer.parseInt(ukuran[1]);
        jumlahBentuk = Integer.parseInt(ukuran[2]);

        if (panjang <= 0 || lebar <= 0 || jumlahBentuk <= 0) {
                throw new IllegalArgumentException("Setiap angka harus
positif");
        }
      } catch (NumberFormatException e) {
        throw new IllegalArgumentException("Input Ukuran Salah");
      }

      String baris2 = br.readLine();
      if (baris2 == null || !baris2.equals("DEFAULT")) {
        throw new IllegalArgumentException("Hanya bisa DEFAULT");
      }

      List<int[]> listBentuk = new ArrayList<>();
      String line;
      char currentChar = ' ';
      int row = 0;

      while ((line = br.readLine()) != null) {
        if (line.trim().isEmpty()) {
          row++;
          continue;
        }

        char charBentuk = ' ';
        for (int i = 0; i < line.length(); i++) {
          if (line.charAt(i) != ' ') {
            charBentuk = line.charAt(i);
            break;
          }
```

```
        }

        if (charBentuk == ' ') {
          row++;
          continue;
        }

        if (charBentuk != currentChar && !listBentuk.isEmpty()) {
          bentuk.add(new Bentuk(listBentuk, currentChar));
          listBentuk = new ArrayList<>();
        }
        currentChar = charBentuk;

        for (int i = 0; i < line.length(); i++) {
          if (line.charAt(i) == currentChar) {
            listBentuk.add(new int[] { row, i });
          }
        }
        row++;
      }

      if (!listBentuk.isEmpty()) {
        bentuk.add(new Bentuk(listBentuk, currentChar));
      }

      PuzzleData puzzleData = new PuzzleData(panjang, lebar, bentuk);
      if (puzzleData.getCountBentuk() != jumlahBentuk) {
        throw new IllegalArgumentException("Jumlah bentuk tidak sesuai");
      }
    }

    return new PuzzleData(panjang, lebar, bentuk);
  }
}
```

4. File UbahBentuk.java

```
import java.util.*;

public class UbahBentuk {
```

```java
public static Bentuk putar90(Bentuk bentuk) {
  List<int[]> koordinatBaru = new ArrayList<>();
  for (int[] k : bentuk.getKoordinat()) {
    koordinatBaru.add(new int[] { -k[1], k[0] });
  }

  return new Bentuk(normalisasi(koordinatBaru), bentuk.getHuruf());
}

public static Bentuk putar180(Bentuk bentuk) {
  List<int[]> koordinatBaru = new ArrayList<>();
  for (int[] k : bentuk.getKoordinat()) {
    koordinatBaru.add(new int[] { -k[0], -k[1] });
  }

  return new Bentuk(normalisasi(koordinatBaru), bentuk.getHuruf());
}

public static Bentuk putar270(Bentuk bentuk) {
  List<int[]> koordinatBaru = new ArrayList<>();
  for (int[] k : bentuk.getKoordinat()) {
    koordinatBaru.add(new int[] { k[1], -k[0] });
  }

  return new Bentuk(normalisasi(koordinatBaru), bentuk.getHuruf());
}

public static Bentuk cerminAtas(Bentuk bentuk) {
  List<int[]> koordinatBaru = new ArrayList<>();
  for (int[] k : bentuk.getKoordinat()) {
    koordinatBaru.add(new int[] { k[0], -k[1] });
  }

  return new Bentuk(normalisasi(koordinatBaru), bentuk.getHuruf());
}

public static Bentuk cerminSamping(Bentuk bentuk) {
  List<int[]> koordinatBaru = new ArrayList<>();
  for (int[] k : bentuk.getKoordinat()) {
```

```
      koordinatBaru.add(new int[] { -k[0], k[1] });
    }

    return new Bentuk(normalisasi(koordinatBaru), bentuk.getHuruf());
  }

  private static List<int[]> normalisasi(List<int[]> koordinat) {
    int minX = Integer.MAX_VALUE, minY = Integer.MAX_VALUE;
    for (int[] k : koordinat) {
      minX = Math.min(minX, k[0]);
      minY = Math.min(minY, k[1]);
    }

    List<int[]> hasilNormalisasi = new ArrayList<>();
    for (int[] k : koordinat) {
      hasilNormalisasi.add(new int[] { k[0] - minX, k[1] - minY });
    }

    hasilNormalisasi.sort(Comparator.comparingInt((int[] k) -> k[0])
        .thenComparingInt(k -> k[1]));
    return hasilNormalisasi;
  }
}
```

5. File Bruteforce.java

```
import java.util.*;

public class Bruteforce {
  private Wadah wadah;
  private List<Bentuk> listBentuk;
  private List<List<Bentuk>> semuaVariasi;
  private int jumlahKasus;

  public Bruteforce(Wadah wadah, List<Bentuk> listBentuk) {
    this.wadah = wadah;
    this.listBentuk = listBentuk;
    this.semuaVariasi = new ArrayList<>();
    this.jumlahKasus = 0;
```

```
  for (Bentuk bentuk : listBentuk) {
    semuaVariasi.add(bentuk.buatVariasi());
  }
}

public boolean solved() {
  return solveBacktrack(0);
}

private boolean solveBacktrack(int bentukIdx) {
  if (bentukIdx == listBentuk.size()) {
    int totalWadah = wadah.getRows() * wadah.getCols();
    int wadahTerisi = 0;

    for (int i = 0; i < wadah.getRows(); i++) {
      for (int j = 0; j < wadah.getCols(); j++) {
        if (wadah.getCharAt(i, j) != '.') {
          wadahTerisi++;
        }
      }
    }

    return wadahTerisi == totalWadah;
  }

  List<Bentuk> variasiSekarang = semuaVariasi.get(bentukIdx);

  for (Bentuk variasi : variasiSekarang) {
    for (int x = 0; x < wadah.getRows(); x++) {
      for (int y = 0; y < wadah.getCols(); y++) {
        jumlahKasus++;

        if (wadah.cekBentuk(variasi, x, y)) {
          wadah.place(variasi, x, y);

          if (solveBacktrack(bentukIdx + 1)) {
            return true;
          }
```

```
            wadah.hapus(variasi, x, y);
          }
        }
      }
    }
    return false;
  }

  public int getJumlahKasus() {
    return jumlahKasus;
  }
}
```

6. File Main.java

```
import java.util.*;
import java.io.*;

public class Main {
  public static void main(String[] args) {
    if (System.getProperty("os.name").toLowerCase().contains("windows")) {
      try {
                                    new     ProcessBuilder("cmd",     "/c",
"color").inheritIO().start().waitFor();
      } catch (Exception e) {
      }
    }

    try {
      Scanner scanner = new Scanner(System.in);
      System.out.println("Masukkan nama file: ");
      String filename = scanner.nextLine();

      if (!filename.contains("/")) {
        filename = "test/input/" + filename;
      }

                                PuzzleInput.PuzzleData     puzzleData     =
PuzzleInput.bacaPuzzle(filename);
```

```
      int rows = puzzleData.baris;
      int cols = puzzleData.kolom;
      List<Bentuk> bentukList = puzzleData.bentukList;

      Wadah wadah = new Wadah(rows, cols);
      Bruteforce solver = new Bruteforce(wadah, bentukList);

      System.out.println("Mencari solusi...\n");
      long waktuMulai = System.currentTimeMillis();
      boolean found = solver.solved();
      long waktuSelesai = System.currentTimeMillis();
      long durasi = waktuSelesai - waktuMulai;

      if (found) {
        System.out.println("Solusi ditemukan!\n");
        wadah.printWadah();
        System.out.println("\nWaktu pencarian: " + durasi + " ms\n");
                System.out.println("Banyak kasus yang ditinjau: " +
solver.getJumlahKasus() + "\n");
               System.out.println("Apakah anda ingin menyimpan solusi?
(ya/tidak)");
        String jawaban = scanner.nextLine().toLowerCase();

        if (jawaban.equals("ya")) {
          saveSolution(wadah, filename);
        }
      } else {
        System.out.println("Tidak ada solusi yang ditemukan.");
        System.out.println("\nWaktu pencarian: " + durasi + " ms\n");
                System.out.println("Banyak kasus yang ditinjau: " +
solver.getJumlahKasus() + "\n");
      }

      scanner.close();

    } catch (Exception e) {
      System.out.println("Error: " + e.getMessage());
    }
```

```
  }

    private static void saveTextSolution(Wadah wadah, String filename)
throws IOException {
    try (PrintWriter writer = new PrintWriter(new FileWriter(filename))) {
      for (int i = 0; i < wadah.getRows(); i++) {
        for (int j = 0; j < wadah.getCols(); j++) {
          writer.print(wadah.getCharAt(i, j) + " ");
        }
        writer.println();
      }
    }
  }

  public static void saveSolution(Wadah wadah, String filename) {
    try {
      String inputfile = filename;
      if (inputfile.contains("/")) {
        inputfile = inputfile.substring(inputfile.lastIndexOf("/") + 1);
      }
      if (inputfile.contains(".")) {
        inputfile = inputfile.substring(0, inputfile.lastIndexOf("."));
      }

      String textFilename = "test/output/solusi_" + inputfile + ".txt";
      saveTextSolution(wadah, textFilename);

      String imageFilename = "test/output/solusi_" + inputfile + ".png";
      PrintGambar.saveImageSolution(wadah, imageFilename);

      System.out.println("Solusi berhasil disimpan di:");
      System.out.println("- Text: " + textFilename);
      System.out.println("- Image: " + imageFilename);
    } catch (IOException e) {
              System.out.println("Error saat menyimpan solusi: " +
e.getMessage());
    }
  }
}
```

C. Cara Menjalankan Program

Sebelum menjalankan program, pengguna harus memastikan telah menginstall java terlebih dahulu. Disarankan untuk menggunakan java versi terbaru demi performa yang lebih baik. Berikut adalah cara menjalankan program.

1. Anda dapat menggunakan file custom dengan menaruh file pada direktori `test/input`.
2. Buka terminal pada direktori program.
3. Silahkan ketik `java -cp bin Main`.
4. Ikuti arahan program.
5. File yang disimpan dapat dibuka pada folder `test/output`.

D. Hasil Eksekusi Program

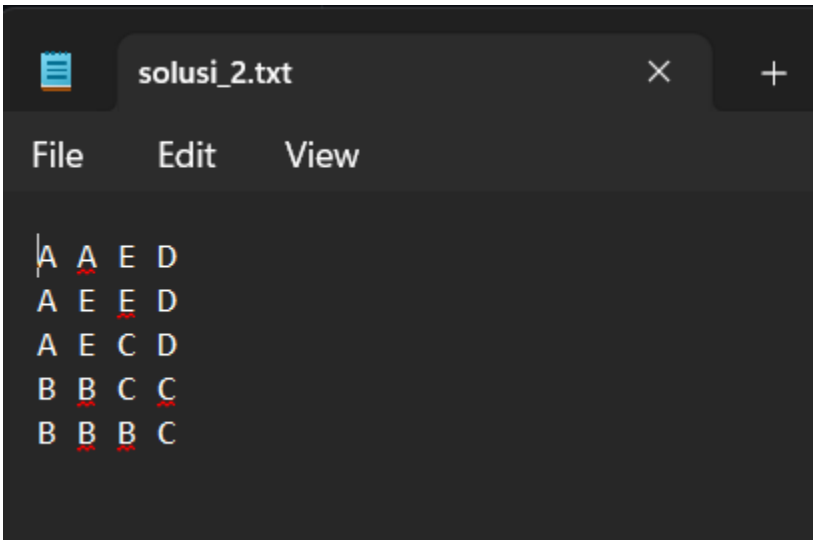
Berikut adalah hasil eksekusi program menggunakan sistem operasi Windows 11 dengan spesifikasi prosesor Ryzen 7 5825u serta RAM 16gb DDR4 3200Mhz.

```
PS C:\Users\rakad\OneDrive\Dokumen\ITB\Semester 4\Stima\Tucil1_13523018> java -cp bin Main
Masukkan nama file:
1.txt
Mencari solusi...

Solusi ditemukan!

A C C E E
A A C E E
B D D F E
B B D F F
G G G F F

Waktu pencarian: 73 ms

Banyak kasus yang ditinjau: 83280

Apakah anda ingin menyimpan solusi? (ya/tidak)
tidak
```

Gambar 4.1 Eksekusi program menggunakan file 1.txt tanpa menyimpan hasil.

```
PS C:\Users\rakad\OneDrive\Dokumen\ITB\Semester 4\Stima\Tucil1_13523018> java -cp bin Main
Masukkan nama file:
2.txt
Mencari solusi...

Solusi ditemukan!

A A E D
A E E D
A E C D
B B C C
B B B C

Waktu pencarian: 19 ms

Banyak kasus yang ditinjau: 519

Apakah anda ingin menyimpan solusi? (ya/tidak)
ya
Solusi berhasil disimpan di:
- Text: test/output/solusi_2.txt
- Image: test/output/solusi_2.png
```

Gambar 4.2 Eksekusi program menggunakan file 2.txt dengan menyimpan hasil.



Gambar 4.3 Hasil save file solusi_2.txt

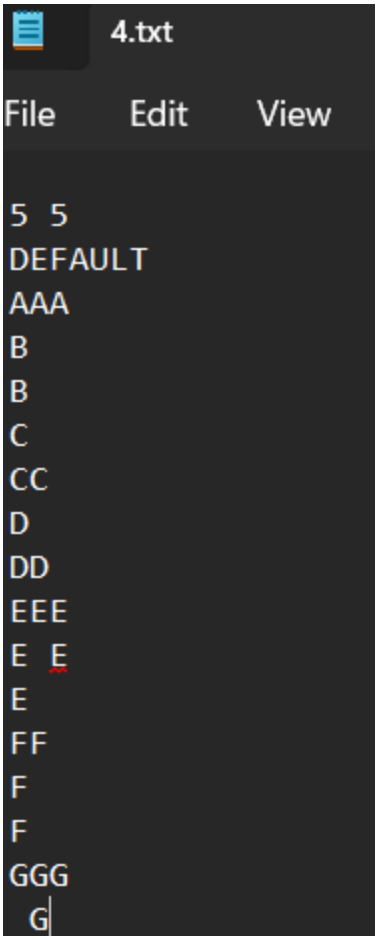
```
PS C:\Users\rakad\OneDrive\Dokumen\ITB\Semester 4\Stima\Tucil1_13523018> java -cp bin Main
Masukkan nama file:
3.txt
Mencari solusi...

Solusi ditemukan!

A A D D D
A E D B D
F E E B B
F E B B C
F F F C C

Waktu pencarian: 80 ms

Banyak kasus yang ditinjau: 136174

Apakah anda ingin menyimpan solusi? (ya/tidak)
ya
Solusi berhasil disimpan di:
- Text: test/output/solusi_3.txt
- Image: test/output/solusi_3.png
```

Gambar 4.4 Eksekusi program menggunakan file 3.txt dengan menyimpan hasil.

Gambar 4.5 Hasil save file solusi_3.png

```
PS C:\Users\rakad\OneDrive\Dokumen\ITB\Semester 4\Stima\Tucil1_13523018> java -cp bin Main
Masukkan nama file:
4.txt
Error: Komponen baris pertama kurang
```

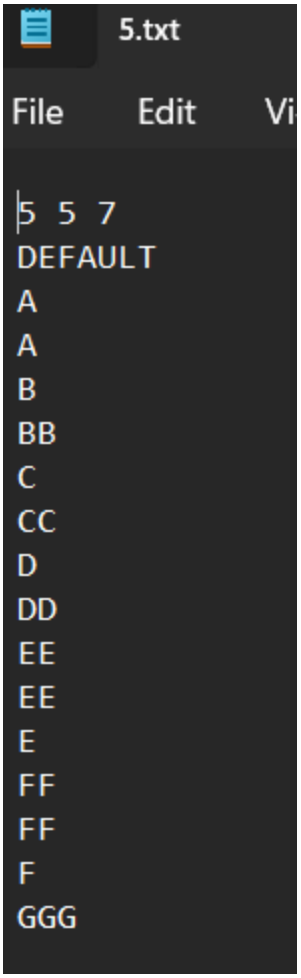
Gambar 4.6 Eksekusi file 4.txt dengan komponen baris pertama hanya dua buah



Gambar 4.7 File input 4.txt

```
PS C:\Users\rakad\OneDrive\Dokumen\ITB\Semester 4\Stima\Tucil1_13523018> java -cp bin Main
Masukkan nama file:
5.txt
Mencari solusi...

Tidak ada solusi yang ditemukan.

Waktu pencarian: 21317 ms

Banyak kasus yang ditinjau: 435990200
```

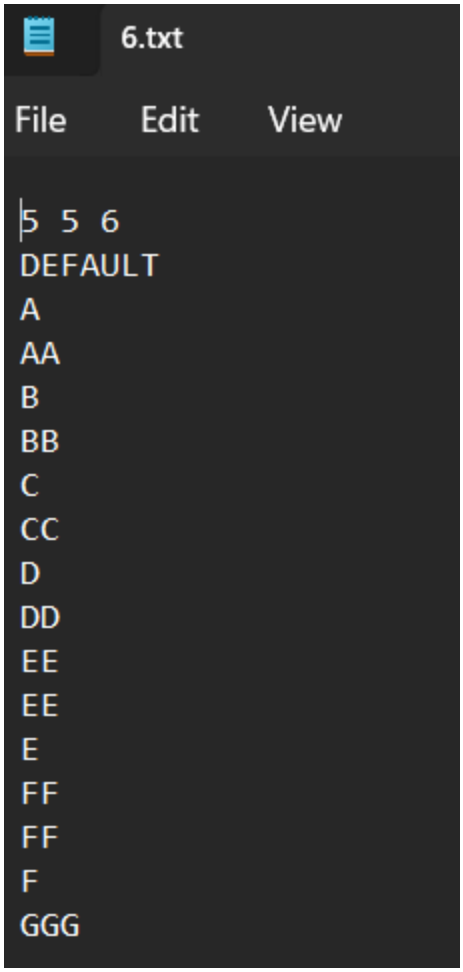
Gambar 4.8 Eksekusi file 5.txt dengan pengurangan satu huruf (bukan bentuk) pada test case



Gambar 4.9 File test case 5.txt (penghapusan satu huruf A)

```
PS C:\Users\rakad\OneDrive\Dokumen\ITB\Semester 4\Stima\Tucil1_13523018> java -cp bir
Masukkan nama file:
6.txt
Error: Jumlah bentuk tidak sesuai
```

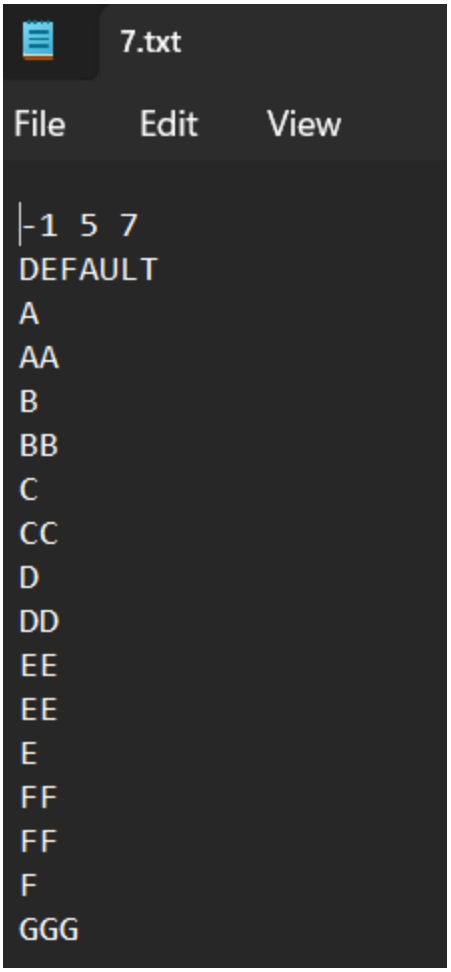
Gambar 4.10 Eksekusi file 6.txt dengan perbedaan jumlah bentuk pada file



Gambar 4.11 File test case 6.txt

```
PS C:\Users\rakad\OneDrive\Dokumen\ITB\Semester 4\Stima\Tucil1_13523018> java -cp bin Main
Masukkan nama file:
7.txt
Error: Setiap angka harus positif
```

Gambar 4.12 Hasil eksekusi file 7.txt dengan input tidak valid



Gambar 4.13 File 7.txt dengan input tidak valid

E. Lampiran

Tautan repository GitHub: https://github.com/rakdaf08/Tucil1_13523018

Tabel Spesifikasi Program

No	Poin	Ya	Tidak
1	Program berhasil dikompilasi tanpa kesalahan	V	
2	Program berhasil dijalankan	V	
3	Solusi yang diberikan program benar dan mematuhi aturan permainan	V	
4	Program dapat membaca masukan berkas .txt serta menyimpan solusi dalam berkas .txt	V	

5	Program memiliki *Graphical User Interface* (GUI)		V
6	Program dapat menyimpan solusi dalam bentuk file gambar	V	
7	Program dapat menyelesaikan kasus konfigurasi *custom*		V
8	Program dapat menyelesaikan kasus konfigurasi Piramida (3D)		V
9	Program dibuat oleh saya sendiri	V	