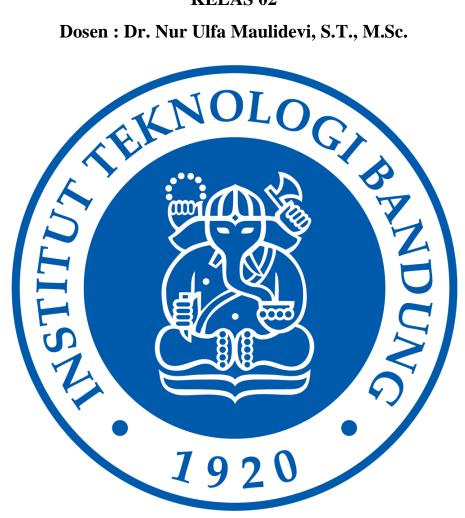
LAPORAN TUGAS KECIL I

"Penyelesaian Word Search Puzzle dengan Algoritma Brute Force" Laporan Ini Dibuat Untuk Memenuhi Tugas Perkuliahan

Mata Kuliah Strategi Algoritma (IF2211)

KELAS 02

Dosen: Dr. Nur Ulfa Maulidevi, S.T., M.Sc.



DISUSUN OLEH:

Yakobus Iryanto Prasethio (13520104)

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA INSTITUT TEKNOLOGI BANDUNG **SEMESTER II TAHUN 2021-2022**

Daftar Isi

| Daftar Isi | |
|------------|----|
| BAB I | 2 |
| BAB II | |
| BAB III | 14 |
| BAB IV | 23 |
| RAR V | 23 |

BAB I Langkah Algoritma *Brute Force*

Deskripsi Langkah Algoritma

Algoritma *Brute Force* adalah algoritma yang menggunakan pendekatan *straightforward* untuk menyelesaikan sebuah permasalahan. Algoritma *brute force* umumnya tidak "cerdas" dan tidak mangkus, karena program membutuhkan volume komputasi yang besar dan waktu yang lama dalam penyelesaiannya.

Langkah – Langkah algoritma untuk program Word Search Puzzle Solver:

- 1. Program membaca setiap elemen matriks puzzle untuk mencari huruf pertama dari kata yang ingin dicari. Misalnya EARTH, maka program akan mencari huruf "E" di dalam matriks puzzle. Apabila program tidak menemukan huruf tersebut di dalam matriks, maka kata tersebut tidak terdapat dalam puzzle.
- 2. Setelah huruf ditemukan, maka program akan mencari ke setiap arah (seperti arah mata angin) untuk mencari huruf kedua. Misalnya "E" dalam EARTH sudah ditemukan, maka program akan mencari huruf "A". Jika program menemukan huruf kedua di kanan dari huruf pertama, maka *flag* kanan akan menjadi true. Apabila program tidak menemukan huruf kedua di arah manapun, maka kata tidak terdapat dalam matriks itu, sehingga program akan melanjutkan pencarian ke huruf pertama selanjutnya.
- 3. Program akan mengecek *flag* yang berada dalam status *true* dan mengecek kecocokan huruf selanjutnya dengan kata yang dicari. Misalnya *flag* kiri dari kata EARTH menjadi *true*, maka program akan mengecek apabila setiap huruf selanjutnya sama. Apabila semua huruf sama dan panjang dari kata sama, maka program akan selesai dan mencetak matriks solusi.

BAB II

Source Program dalam Bahasa Java

```
import java.io.*;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.*;
public class wordPuzzle {
       // DECLARING COLOR SPACES
       public static final String ANSI RESET = "\033[0m";
       public static final String ANSI MAGENTA = "\033[95m";
       public static final String ANSI RED = "\033[31m";
       public static final String ANSI_GREEN = "\033[32m";
      public static final String ANSI_YELLOW = "\033[33m";
public static final String ANSI_BLUE = "\033[34m";
       public static final String ANSI PURPLE = "\033[35m";
       public static final String ANSI CYAN = "\033[36m";
       public static final String ANSI BRIGHT RED = "\033[91m";
       public static boolean is Within Range (int index, int lower Bound, int
       upperBound) {
             if ((index >= lowerBound) && (index < upperBound)) {</pre>
                    return true;
              } else {
                    return false;
       }
       public static void main (String[] args) {
              int i, j, k, x, y;
              int letterCounter, charIndexX, charIndexY, stepsTaken;
              int ansX, ansY;
             String testCase, chars, path1, path2, fullPath;
             Character currentChar;
             boolean markFound = false;
             boolean checkFlag;
             boolean up, rightup, right, rightdown, down, leftdown, left, leftup;
             boolean finalup, finalrightup, finalright, finalrightdown, finaldown,
              finalleftdown, finalleft, finalleftup;
             long startComp, endComp, compDur, startProg, endProg, progDur;
              float elapsedTime, fileDuration;
             // INITIATE BOOLEANS
             up = false;
             rightup = false;
             right = false;
             rightdown = false;
             down = false;
             leftdown = false;
             left = false;
             leftup = false;
              finalup = false;
              finalrightup = false;
              finalright = false;
              finalrightdown = false;
              finaldown = false;
              finalleftdown = false;
              finalleft = false;
              finalleftup = false;
             System.out.println("Please input your file name in the test folder:");
             path1 = "../test/";
              Scanner input = new Scanner(System.in);
             path2 = input.nextLine();
              fullPath = path1 + path2;
             startProg = System.nanoTime();
```

```
input.close();
try {
```

Blok kode dipisah agar dapat terbaca dengan baik

```
List<String> fullFile = Files.readAllLines(Paths.get(fullPath));
ArrayList<ArrayList<Character>> game = new
ArrayList<ArrayList<Character>>();
     // READING MATRIX
     for (String line : fullFile) {
          if (line.trim().isEmpty()) {
              break;
          } else {
              Scanner scanElmt = new Scanner (line);
              ArrayList<Character> col = new ArrayList<Character>();
              while (scanElmt.hasNext()) {
                  chars = scanElmt.next().trim();
                  col.add(chars.charAt(0));
              game.add(col);
              scanElmt.close();
         }
     }
// CREATE COLOR MATRIX
String[][] color codes = new String[game.size()][game.get(0).size()];
for (i=0;i<game.size();i++) {</pre>
       for(j=0;j<game.get(i).size();j++) {</pre>
       color_codes[i][j] = ANSI_RESET;
// CREATING ANSWER MATRIX
String[][] answer = new String[game.size()][game.get(0).size()];
for (i=0;i<game.size();i++) {</pre>
       for (j=0;j<game.get(i).size();j++) {</pre>
             answer[i][j] = game.get(i).get(j);
       }
}
// CREATE COLOR MATRIX
String[][] color codes = new String[game.size()][game.get(0).size()];
for (i=0;i<game.size();i++) {
       for(j=0;j<game.get(i).size();j++) {</pre>
              color codes[i][j] = ANSI RESET;
       }
}
// CREATING ANSWER MATRIX
String[][] answer = new String[game.size()][game.get(0).size()];
for (i=0;i<game.size();i++) {</pre>
      for (j=0;j<game.get(i).size();j++) {</pre>
              answer[i][j] = "-";
     }
// READ TEST CASES AND CALCULATING
for (String line : fullFile) {
       stepsTaken = 0;
       if (line.trim().isEmpty()) {
             markFound = true;
              continue;
       if (markFound) {
```

```
startComp = System.nanoTime();
Scanner readCase = new Scanner(line);
testCase = readCase.next();
for (i=0;i<game.size();i++) {</pre>
```

Blok kode dipisah agar dapat terbaca dengan baik

```
for (i=0;i<game.size();i++) {</pre>
       for (j=0;j<game.get(0).size();j++) {</pre>
              stepsTaken++;
              if (game.get(i).get(j) == testCase.charAt(0)) {
                     k = 0;
                      x = j; // Letak baris
                      y = i; // Letak kolom
                      // CHECK EVERY DIRECTION
                      // IF CURRENTLY SCANNING TOP LEFT CORNER
                      if ((i == 0) \&\& (j == 0)) {
                             // Arah kanan
                             if (testCase.charAt(k+1) == game.get(y).get(x+1)) {
                                    stepsTaken++;
                                    right = true;
                             }
                             // Arah kanan bawah
                             if (\text{testCase.charAt}(k+1) == \text{game.get}(y+1).\text{get}(x+1))  {
                                    stepsTaken++;
                                    rightdown = true;
                      }
                             // Arah bawah
                             if (testCase.charAt(k+1) == game.get(y+1).get(x)) {
                                    stepsTaken++;
                                    down = true;
                      // IF CURRENTLY SCANNING TOP RIGHT CORNER
                      } else if ((i == 0) \&\& (j == game.get(i).size()-1)) {
                             // Arah bawah
                             if (testCase.charAt(k+1) == game.get(y+1).get(x)) {
                                    stepsTaken++;
                                    down = true;
                             // Arah kiri bawah
                             if (\text{testCase.charAt}(k+1) == \text{game.get}(y+1).\text{get}(x-1))  {
                                    stepsTaken++;
                                    leftdown = true;
                             if (testCase.charAt(k+1) == game.get(y).get(x-1)) {
                                    stepsTaken++;
                                    left = true;
                      // IF CURRENTLY SCANNING BOTTOM LEFT CORNER
                      } else if ((i == game.size()-1) && (j == 0)) {
                             // Arah atas
                             if (\text{testCase.charAt}(k+1) == \text{game.get}(y-1).\text{get}(x))  {
                                    stepsTaken++;
                                    up = true;
                             // Arah kanan atas
                             if (\text{testCase.charAt}(k+1) == \text{game.get}(y-1).\text{get}(x+1))  {
                                    stepsTaken++;
                                    rightup = true;
                             // Arah kanan
                             if (testCase.charAt(k+1) == game.get(y).get(x+1)) {
                                    stepsTaken++;
                                    right = true;
                      // IF CURRENTLY SCANNING BOTTOM RIGHT CORNER
```

```
} else if ((i == game.size()-1)&&(j == game.get(i).size()-1)) {
       // Arah atas
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y-1).\text{get}(x)) {
               stepsTaken++;
              up = true;
       // Arah kiri atas
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y-1).\text{get}(x-1))  {
               stepsTaken++;
               leftup = true;
       // Arah kiri
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y).\text{get}(x-1)) {
               stepsTaken++;
               left = true;
// IF CURRENTLY SCANNING TOP ROW
} else if ((i == 0) \&\& (j != 0)) {
       // Arah kanan
       if (testCase.charAt(k+1) == game.get(y).get(x+1)) {
               stepsTaken++;
               right = true;
       }
       // Arah kanan bawah
       if (testCase.charAt(k+1) == game.get(y+1).get(x+1))  {
               stepsTaken++;
               rightdown = true;
       // Arah bawah
       if (testCase.charAt(k+1) == game.get(y+1).get(x)) {
               stepsTaken++;
               down = true;
       // Arah kiri bawah
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y+1).\text{get}(x-1))  {
               stepsTaken++;
               leftdown = true;
       // Arah kiri
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y).\text{get}(x-1)) {
               stepsTaken++;
               left = true;
// IF CURRENTLY SCANNING LEFT COLUMN
} else if ((i != 0) \&\& (j == 0)) {
       // Arah atas
       if (testCase.charAt(k+1) == game.get(y-1).get(x)) {
               stepsTaken++;
               up = true;
       // Arah kanan atas
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y-1).\text{get}(x+1))  {
               stepsTaken++;
               rightup = true;
       }
       // Arah kanan
       if (testCase.charAt(k+1) == game.get(y).get(x+1)) {
               stepsTaken++;
               right = true;
       // Arah kanan bawah
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y+1).\text{get}(x+1))  {
               stepsTaken++;
               rightdown = true;
       // Arah bawah
       if (testCase.charAt(k+1) == game.get(y+1).get(x)) {
               stepsTaken++;
```

```
down = true;
// IF CURRENTLY SCANNING BOTTOM ROW
} else if ((i == game.size()-1) && (j != 0)) {
       // Arah kiri
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y).\text{get}(x-1))  {
               stepsTaken++;
               left = true;
       // Arah kiri atas
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y-1).\text{get}(x-1))  {
               stepsTaken++;
               leftup = true;
       // Arah atas
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y-1).\text{get}(x)) {
               stepsTaken++;
               up = true;
       // Arah kanan atas
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y-1).\text{get}(x+1)) {
               stepsTaken++;
               rightup = true;
       // Arah kanan
       if (testCase.charAt(k+1) == game.get(y).get(x+1)) {
               stepsTaken++;
               right = true;
// IF CURRENTLY SCANNING RIGHT COLUMN
} else if ((i != 0) \&\& (j == game.get(i).size()-1)) {
       // Arah atas
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y-1).\text{get}(x))  {
               stepsTaken++;
               up = true;
       }
       // Arah kiri atas
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y-1).\text{get}(x-1))  {
               stepsTaken++;
               leftup = true;
       // Arah kiri
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y).\text{get}(x-1)) {
               stepsTaken++;
               left = true;
       // Arah kiri bawah
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y+1).\text{get}(x-1))  {
               stepsTaken++;
               leftdown = true;
       // Arah bawah
       if (testCase.charAt(k+1) == game.get(y+1).get(x)) {
               stepsTaken++;
               down = true;
       }
// IF CURRENTLY SCANNING THE MIDDLE OF THE MATRIX
} else {
       // Arah atas
       if (testCase.charAt(k+1) == game.get(y-1).get(x)) {
               stepsTaken++;
               up = true;
       // Arah kanan atas
       if (\text{testCase.charAt}(k+1) == \text{game.get}(y-1).\text{get}(x+1))  {
               stepsTaken++;
               rightup = true;
       }
```

```
// Arah kanan
        if (\text{testCase.charAt}(k+1) == \text{game.get}(y).\text{get}(x+1)) {
                stepsTaken++;
                right = true;
        // Arah kanan bawah
        if (\text{testCase.charAt}(k+1) == \text{game.get}(y+1).\text{get}(x+1))  {
                stepsTaken++;
                rightdown = true;
        // Arah bawah
        if (testCase.charAt(k+1) == game.get(y+1).get(x)) {
                stepsTaken++;
                down = true;
        // Arah kiri bawah
        if (\text{testCase.charAt}(k+1) == \text{game.get}(y+1).\text{get}(x-1))  {
                stepsTaken++;
                leftdown = true;
        // Arah kiri
        if (\text{testCase.charAt}(k+1) == \text{game.get}(y).\text{get}(x-1)) {
                stepsTaken++;
                left = true;
        }
        // Arah kiri atas
        if (\text{testCase.charAt}(k+1) == \text{game.get}(y-1).\text{get}(x-1))  {
                stepsTaken++;
                leftup = true;
        }
}
// ITERATE THE REST OF CHARACTERS BASED ON FLAG
checkFlag = true;
. . . . . .
```

Blok kode dipisah agar dapat terbaca dengan baik

```
// ITERATE THE REST OF CHARACTERS BASED ON FLAG
checkFlag = true;
k = 1;
letterCounter = 1;
charIndexX = 0;
charIndexY = 0;
while ((k < testCase.length()) && (checkFlag)) {</pre>
       if (up) {
             charIndexX = x;
             charIndexY = y-1;
             if ((isWithinRange(charIndexX, 0, game.get(i).size())) &&
             (isWithinRange(charIndexY, 0, game.size()))) {
                    if ((game.get(y-1).get(x) == testCase.charAt(k)) &&
                    (letterCounter != testCase.length())) {
                           stepsTaken++;
                           letterCounter++;
                           if (letterCounter == testCase.length()) {
                                  finalup = true;
                                  up = false;
                           }
                     } else {
                           x = j;
                           y = i;
                           k = 0;
                           letterCounter = 1;
                           up = false;
             } else {
                    x = j;
```

```
y = i;
             k = 0;
             letterCounter = 1;
             up = false;
      }
} else if (rightup) {
      charIndexX = x+1;
      charIndexY = y-1;
      if ((isWithinRange(charIndexX, 0, game.get(i).size())) &&
      (isWithinRange(charIndexY, 0, game.size()))) {
             if ((game.get(y-1).get(x+1) == testCase.charAt(k)) &&
             (letterCounter != testCase.length())) {
                    stepsTaken++;
                    letterCounter++;
                    x++;
                    y--;
if (letterCounter == testCase.length()) {
                           finalrightup = true;
                           rightup = false;
                    }
             } else {
                    x = j;
                    y = i;
                    k = 0;
                    letterCounter = 1;
                    rightup = false;
       } else {
             x = j;
             y = i;
             k = 0;
             letterCounter = 1;
             rightup = false;
      }
} else if (right) {
      charIndexX = x+1;
      charIndexY = y;
      if ((isWithinRange(charIndexX, 0, game.get(i).size())) &&
      (isWithinRange(charIndexY, 0, game.size()))) {
             if ((game.get(y).get(x+1) == testCase.charAt(k)) &&
             (letterCounter != testCase.length())) {
                    stepsTaken++;
                    letterCounter++;
                    x++;
                    if (letterCounter == testCase.length()) {
                           finalright = true;
                           right = false;
                    }
              } else {
                    x = j;
                    y = i;
                    k = 0;
                    letterCounter = 1;
                    right = false;
      } else {
             x = j;
             y = i;
             k = 0;
             letterCounter = 1;
             right = false;
      }
} else if (rightdown) {
      charIndexX = x+1;
      charIndexY = y+1;
      if ((isWithinRange(charIndexX, 0, game.get(i).size())) &&
      (isWithinRange(charIndexY, 0, game.size()))) {
```

```
if ((game.get(y+1).get(x+1) == testCase.charAt(k)) &&
             (letterCounter != testCase.length())) {
                    stepsTaken++;
                    letterCounter++;
                    x++;
                    y++;
                    if (letterCounter == testCase.length()) {
                           finalrightdown = true;
                           rightdown = false;
                    }
              } else {
                    x = j;
                    y = i;
                    k = 0;
                    letterCounter = 1;
                    rightdown = false;
       } else {
             x = j;
             y = i;
             k = 0;
             letterCounter = 1;
             rightdown = false;
      }
} else if (down) {
      charIndexX = x;
      charIndexY = y+1;
      if ((isWithinRange(charIndexX, 0, game.get(i).size())) &&
      (isWithinRange(charIndexY, 0, game.size()))) {
             if ((game.get(y+1).get(x) == testCase.charAt(k)) &&
             (letterCounter != testCase.length())) {
                    stepsTaken++;
                    letterCounter++;
                    y++;
                    if (letterCounter == testCase.length()) {
                           finaldown = true;
                           down = false;
             } else {
                    x = j;
                    y = i;
                    k = 0;
                    letterCounter = 1;
                    down = false;
             }
       } else {
             x = j;
             y = i;
             k = 0;
             letterCounter = 1;
             down = false;
      }
} else if (leftdown) {
      charIndexX = x-1;
      charIndexY = y+1;
      if ((isWithinRange(charIndexX, 0, game.get(i).size())) &&
       (isWithinRange(charIndexY, 0, game.size()))) {
             if ((game.get(y+1).get(x-1) == testCase.charAt(k)) &&
             (letterCounter != testCase.length())) {
                    stepsTaken++;
                    letterCounter++;
                    x--;
                    y++;
                    if (letterCounter == testCase.length()) {
                           finalleftdown = true;
                           leftdown = false;
             } else {
```

```
x = j;
                    y = i;
                    \hat{k} = 0;
                    letterCounter = 1;
                    leftdown = false;
       } else {
             x = j;
             y = i;
             k = 0;
             letterCounter = 1;
             leftdown = false;
      }
} else if (left) {
      charIndexX = x-1;
      charIndexY = y;
      if ((isWithinRange(charIndexX, 0, game.get(i).size())) &&
      (isWithinRange(charIndexY, 0, game.size()))) {
             if ((game.get(y).get(x-1) == testCase.charAt(k)) &&
             (letterCounter != testCase.length())) {
                    stepsTaken++;
                    letterCounter++;
                    x--;
                    if (letterCounter == testCase.length()) {
                           finalleft = true;
                           left = false;
             } else {
                    x = j;
                    y = i;
                    k = 0;
                    letterCounter = 1;
                    left = false;
             }
       } else {
             x = j;
             y = i;
             k = 0;
             letterCounter = 1;
             left = false;
      }
} else if (leftup) {
      charIndexX = x-1;
      charIndexY = y-1;
      if ((isWithinRange(charIndexX, 0, game.get(i).size())) &&
       (isWithinRange(charIndexY, 0, game.size()))) {
             if ((game.get(y-1).get(x-1) == testCase.charAt(k)) &&
             (letterCounter != testCase.length())) {
                    stepsTaken++;
                    letterCounter++;
                    x--;
                    y--;
                    if (letterCounter == testCase.length()) {
                           finalleftup = true;
                           leftup = false;
                    }
              } else {
                    x = j;
                    y = i;
                    k = 0;
                    letterCounter = 1;
                    leftup = false;
       } else {
             x = j;
             y = i;
             k = 0;
             letterCounter = 1;
```

```
leftup = false;
if (letterCounter == testCase.length()) {
      endComp = System.nanoTime();
      compDur = endComp - startComp;
      elapsedTime = (float)compDur / (float)1000000;
      System.out.println("======= Word Found! =======");
      System.out.println("Program took " + stepsTaken + " step(s) to find
      the word");
      System.out.print("Exited in ");
      System.out.printf("%.5f", elapsedTime);
      System.out.println(" ms");
      ansX = j;
      ansY = i;
      for (i=0;i<testCase.length();i++) {</pre>
             if (finalup) {
                    currentChar = testCase.charAt(i);
                    answer[ansY] [ansX] = String.valueOf(currentChar);
                    color_codes[ansY][ansX] = ANSI_BLUE;
                    ansY--;
              } else if (finalrightup) {
                    currentChar = testCase.charAt(i);
                    answer[ansY][ansX] = String.valueOf(currentChar);
                    color codes[ansY][ansX] = ANSI CYAN;
                    ansX++;
                    ansY--;
              } else if (finalright) {
                    currentChar = testCase.charAt(i);
                    answer[ansY] [ansX] = String.valueOf(currentChar);
                    color codes[ansY][ansX] = ANSI GREEN;
                    ansX++;
              } else if (finalrightdown) {
                    currentChar = testCase.charAt(i);
                    answer[ansY] [ansX] = String.valueOf(currentChar);
                    color codes[ansY][ansX] = ANSI MAGENTA;
                    ansX++;
                    ansY++;
              } else if (finaldown) {
                    currentChar = testCase.charAt(i);
                    answer[ansY] [ansX] = String.valueOf(currentChar);
                    color_codes[ansY][ansX] = ANSI_PURPLE;
                    ansY++;
              } else if (finalleftdown) {
                    currentChar = testCase.charAt(i);
                    answer[ansY] [ansX] = String.valueOf(currentChar);
                    color codes[ansY][ansX] = ANSI RED;
                    ansX--;
                    ansY++;
              } else if (finalleft) {
                    currentChar = testCase.charAt(i);
                    answer[ansY][ansX] = String.valueOf(currentChar);
                    color codes[ansY][ansX] = ANSI BRIGHT RED;
                    ansX--;
              } else if (finalleftup) {
                    currentChar = testCase.charAt(i);
                    answer[ansY] [ansX] = String.valueOf(currentChar);
                    color codes[ansY][ansX] = ANSI YELLOW;
                    ansX--:
                    ansY--;
             }
      }
      for (i=0;i<game.size();i++) {</pre>
             for (j=0;j<game.get(i).size();j++) {</pre>
```

Closing bracket tidak teratur akibat kode yang terpisah – pisah

```
    k++;

}

k++;

}

}

catch (IOException e) {
    System.out.println("File not found!");

endProg = System.nanoTime();

progDur = endProg - startProg;

fileDuration = (float)progDur/(float)1000000;

System.out.println("Program fully exited in " + fileDuration + " ms");

}
```

• Kode dapat terlihat lebih jelas dalam file wordPuzzle.java

BAB III

Screenshot Input dan Output

3.1.Ukuran kecil 1

| Gambar | Keterangan |
|--|--|
| K J V T O J B L P O E Q N R L R I Y A U U O R U L E R D O G E V N C V P O F K T Q X L W W S G I H X E K H S Y O E A V Y A X V U Y I E N B S Q M I L T R N M Y F P V G U X W L Y D F E D L F N V B N A H G R W M K W Y M D I S Y B X M K L F P Y L T J P S O C P Y V A Z Y I N E X F X H R L I N D K C N H Z G F C P D M O U S A F Z O K B D U H K Q D J W W S W F T P G B D I N T U J M B D O E E T U N B S S V P E N C I L R X P Q E O W T H R Z H R L O S S E F N Y Q Y W R Y I X J K L X C O B E S K A M U F M H O B T Y Q V BOOK ERASER HIGHLIGHTER PEN PENCIL | Isi file small1.txt. Tema dari word puzzle adalah nama beberapa alat tulis kantor. |
| PENCIL RULER SCISSORS Please input your file name in the test folder: small1.txt_ P Q E O W T H R Z H R L O S S E F N Y Q Y W R Y I X J K L X C O B E S K A M U F M H O B T Y Q V ========= Word Found! ======== Program took 119 step(s) to find the word Exited in 0.25860 ms K J V T O J B L P O E Q N R L R I Y A U U O R U L E R D O G E V N C V P O F K T Q X L W W S G I H X E K H S Y O E A V Y A X V U Y I E N B S Q M I L T R N M Y F P V G U X W L Y D F E D L F N V B N A H G R W M K W Y M D I S Y B X M K L F P Y L T J P S O C P Y V A Z Y I N E X F X H R L I N D K C N H Z G F C P D M O U S A F Z O K B D U H K Q D J W W S W F T P G B D I N T U J M B D O E E T U N B S S V P E N C I L R X P Q E O W T H R Z H R L O S S E F N Y Q Y W R Y I X J K L X C O B E S K A M U F M H O B T Y Q V Program fully exited in 139.85031 ms Press any key to continue | Input nama file ke dalam program Bagian akhir dari eksekusi program. Bagian atas tidak terlihat karena output program akan mencetak array setiap kali sebuah kata ditemukan. |

3.2.Ukuran kecil 2

| Gambar | Keterangan |
|--|--|
| W Z M G S L E C Z D Q U R V I S Y X G W D G A J Y I H A N P H T J I Z Q D B N T P S C U L Z Z Z W T F I I W C B N H A C O H Z J Z G R N Q V R P K W N C S C V Y B F E G V A G E Q A B O I L E R G T X U Q R Z Y G S D K E D A L S L I S N E T U H H O U V J G V L F H J Z R A H C E D G A K E V C U X Y X W P Q X R X V W S W A E W A A E B S U F K S V O B W G S V T Q W D I I Y D E R R Q W K H A O U H H N X N J E G C F S W P E Y T P N T N X K H S I F Q C S Z T R S K Q Y G C R K M D B U P R A Q Z O B I X K B G R J R B BOILER CABINET DISHWASHER FRIDGE MICROWAVE SINK STOVE | Isi file small2.txt. Tema dari word puzzle adalah nama objek yang berada di dapur. |
| Please input your file name in the test folder: small2.txt P E Y T P N T N X K H S I F Q C S Z T R S K Q Y G C R K M D B U P R A Q Z O B I X K B G R J R B ========= Word Found! ======== Program took 128 step(s) to find the word Exited in 0.25640 ms W Z M G S L E C Z D Q U R V I S Y X G W D G A J Y I H A N P H T J I Z Q D B N T P S C U L Z Z Z W T F I I W C B N H A C O H Z J Z G R N Q V R P K W N C S C V Y B F E G V A G E Q A B O I L E R G T X U Q R Z Y G S D K E D A L S L I S N E T U H H O U V J G V L F H J Z R A H C E D G A K E V C U X Y X W P Q X R X V W S W A E W A A E B S U F K S V O B W G S V T Q W D I I Y D E R R Q W K H A O U H H N X N J E G C F S W P E Y T P N T N X K H S I F Q C S Z T R S K Q Y G C R K M D B U P R A Q Z O B I X K B G R J R B Program fully exited in 162.4436 ms Press any key to continue | Input nama file ke dalam program Bagian akhir dari eksekusi program. Bagian atas tidak terlihat karena output program akan mencetak array setiap kali sebuah kata ditemukan. |

3.3.Ukuran kecil 3

| Gambar | Keterangan |
|---|---|
| E J F M V I P R P W T N I X K I G D W Z A B L V R F X Q J K O A B B N S S S Z G U I Y S K T X U Z A Q N J V H I Z B P P B V U A M Q J V I L X Q B B P I F Q Q X P U F T E C J P N R X M L A O K X C Y T A P H O T O S T A L K N X L I A P B D B W J E I G L O A F A K H Z R L Z Z K U M O T W W H G F M A O H E N I A B R I T G M D P W K R P A T F B K P E K K P H E F O N L P C T O D F T G I I R N R Z B E D H N I A T R U C R S R B Z Y D J A B H I E M R O G I G Q Y B C L I Z I N P S P M M H O B M O H Z R E S A Z Z Z H BED BLANKET CHAIR CURTAIN DRAWER LAMP MIRROR PHOTOS PILLOW RUG | Isi file small3.txt. Tema dari word puzzle adalah nama nama objek di kamar tidur. |
| Please input your file name in the test folder: small3.txt | Input nama file ke dalam program |
| R S R B Z Y D J A B H I E M R O G I G Q Y B C L I Z I N P S P M M H O B M O H Z R E S A Z Z Z H | Bagian akhir dari eksekusi program. Bagian atas tidak terlihat karena output program akan mencetak array setiap kali sebuah kata ditemukan. |

3.4.Ukuran sedang 1

| Gambar | Keterangan |
|--|--|
| D J Y X A H B Q J I Q E Y O G Y A K A R T A A G P Y Y K O Y R F N G E I U U Y M C Q U A T S M W A M G W J O V A M W C J Y C L J C X L N C P B A O G N A R E G G N A T U Y Z H B J I D S A T R L I R L A Z K O Z L N H E I V J A E A R X T G L F W I G C K Z I O A A M A X Y K T U K N Q X M N V G Z K P K Y K S L I X Z M A S A X A Z U I O G H B A R H H A H W Y Z R N R D Y X M V X N E R S L Y I Q W M O H C B A L T Z E B G K R D G H D E S U K E M C S M D I K A F T T B E P M I T S R X J L B Z E N J X Y O T J S H T M S A Z V B Z O C Q S K Y X P O A N F I T U L A N D I T K Z A M F Z J F U S G U E X B Z M W O P I I P X S L L M T H N P V F W I R V M M X G W U O R I N M X Y Y A E R A U I C K H W R T A G N N Q C D J D R G S T B Y G D D M F A F F Z X B D R L T X T O C T N U H D V B A N D U N G E Y R Q N I U U V E Y A P R R L T M Q R I P C K P D K G P M G T L F B D D V B Q J E O E W G Z H V D J L J R H J L Z X F W O B K S H O H S Q J K E G C Y K Q O C L N B O D H P G U D K BANDUNG BANTEN BOGOR DEPOK JAKARTA MADIUN SEMARANG SURABAYA TANGGERANG YOGYAKARTA | Isi file medium1.txt. Tema dari word puzzle adalah nama kota – kota di Indonesia. |
| Please input your file name in the test folder: medium1.txt D K G P M G T L F B D D V B Q J E O E W G Z H V D J L J R H J L Z X F W O B K S H O H S Q J K E G C Y K Q O C L N B O D H P G U D K ========== Word Found! ========= Program took 23 step(s) to find the word Exited in 0.32450 ms D J Y X A H B Q J I Q E Y O G Y A K A R T A A G P Y Y K O Y R F N G E I U U Y M C Q U A T S M W A M G W J O V A M W C J Y C L J C X L N C P B A O G N A R E G G N A T U Y Z H B J I D S A T R L I R L A Z K O Z L N H E I V J A E A R X T G L F W I G C K Z I O A A M A X Y K T U K N Q X M N V G Z K P K Y K S L I X Z M A S A X A Z U I O G H B A R H H A H W Y Z R N R D Y X M V X N E R S L Y I Q W M O H C B A L T Z E B G K R D G H D E S U K E M C S M D I K A F T T B E P M I T S R X J L B Z E N J X Y O T J S H T M S A Z V B Z O C Q S K Y X P O A N F I T U L A N D I T K Z A M F Z J F U S G U E X B Z M W O P I I P X S L L M T H N P V F W I R V M M X G W U O R I N M X Y Y A E R A U I C K H W R T A G N N Q C D J D R G S T B Y G D D M F A F F Z X B D R L T X T O C T N U H D V B A N D U N G E Y R Q N I U U V E Y A P R R L T M Q R I P C K P D K G P M G T L F B D D V B Q J E O E W G Z H V D J L J R H J L Z X F W O B K S H O H S Q J K E G C Y K Q O C L N B O D H P G U D K Program fully exited in 315.1875 ms Press any key to continue | Input nama file ke dalam program Bagian akhir dari eksekusi program. Bagian atas tidak terlihat karena output program akan mencetak array setiap kali sebuah kata ditemukan. |

3.5.Ukuran sedang 2

| Gambar | Keterangan |
|--|--|
| K V A O R V T Z J Z O B I W M W J I O V H C G J H Z N N I C K E L R R D Z N D L N C X M T M H N I C X L G F N P E A J R I N A I D I Y I D Y V Y O X R O M C M P H I D R R P C E T F F W T D M G G A I S G I P X B O P P T K W B N I C U Y W Q P J R S K N O Q V K M E P C I D H H H M Y H F I Q F W N R C O B A L T J J R T K V T A L B C K R N I X I R I E C Y F J W T F M M Z G G W G G I K K V M T H J R B R B C H Y B G M N P E D J O B C Q J F G M S C W T E K D N T J E N O C I L I S S U L V V J O Z G G D E D N S F M H Q M C H I N V S N S L C A I Q U I E C I H R C Z C L V L N Z K G L K H Y D R O G E N U X A B E F S L C U F A N H V D C H T O Q Q G M L H G C H H G N B A C H M V B Z I R V U T A S R T Q L K A R E T I U X N O F I T A F P M I O R R R S M L O G I U E B D Y Q I Z G K S U H P Z A F F S J S D Y F R S X P O N Q S A H M T Z J K S R J O K D X R K J A U O O H O M W B O Z Q S M S V S S I O J Q H R X J S J S S H W J M H D L Y S X F E S A P P C V N K D S X C G Q N C CALSIUM CARBON COBALT COPPER HELIUM HYDROGEN MAGNESTUM NICKEL | Isi file medium2.txt. Tema dari word puzzle adalah nama elemen di tabel periodik. |
| NITROGEN OXYGEN SILICON SODIUM Please input your file name in the test folder: medium2.txt J O K D X R K J A U O O H O M W B O Z Q S M S V S S I O J Q H R X J S J S S H W J M H D L Y S X F E S A P P C V N K D S X C G Q N C ======== Word Found! ======= Program took 461 step(s) to find the word Exited in 0.35040 ms K V A O R V T Z J Z O B I W M W J I O V H C G J H Z N N I C K E L R R D Z N D L N C X M T M H N I C X L G F N P E A J R I N A I D I Y I D Y V Y O X R O M C M P H I D R R P C E T F F W T D M G G A I S G I P X B O P P T K W B N I C U Y W Q P J R S K N O Q V K M E P C I D H H H M Y H F I Q F W N R C O B A L T J J R T K V T A L B C K R N I X I R I E C Y F J W T F M M Z G G W G G I K K V M T H J R B R B C H Y B G M N P E D J O B C Q J F G M S C W T E K D N T J E N O C I L I S S S U L V V J O Z G G D E D N S F M H Q M C H I N V S N S L C A I Q U I E C I H R C Z C L V L N Z K G L K H Y D R O G E N U X A B E F S L C U F A N H V D C H T O Q Q G M L H G C H H G N B A C H M V B Z I R V U T A S R T Q L K A R E T I U X N O F I T A F P M I O R R R S M L O G I U E B D Y Q I Z G K S U H P Z A F S J S D Y F R S X P O N Q S A H M T Z J K S R J O K D X R K J A U O O H O M W B O Z Q S M S V S S I O J Q H R X J S J S S H W J M H D L Y S X F E S A P P C V N K D S X C G Q N C Program fully exited in 363.8663 ms Press any key to continue | Input nama file ke dalam program Bagian akhir dari eksekusi program. Bagian atas tidak terlihat karena output program akan mencetak array setiap kali sebuah kata ditemukan. |

3.6.Ukuran sedang 3

| Gambar | Keterangan |
|--|--|
| J S P B C S H M X T C G X Z R V A P J V Z P C Z N Y U G B S I M T S D Z U T Z D I T N H E R Y Y J L K K L H Y X S L A G J V T C P G T A Q M G C Z E B V X U W P R B M C K B F K A F L E B J J H T A I P X I D K E E G J L M P V Y N P K C Z C C K L K Q F E A L K X A S S I Q M V E R P O S A C W G N L T D T O Q A Z N Z M V Z H P R Y L G C M J C B O R J H P H O Z Z W C Q J Z V W T X T B A A O K F O C J L S V A Z B M M A P K P S O R L N R E T Z U V R Q X G K Q Q J X S M I G F L I S P D E U X N T K L L Z N V S D P M Y N E I U K O N V Y C R I S P S T V D K N V K S I Y L O G W E E K H B K A R H W V N F U D E R L L G W L K C V H K E T A Q S D X Z C S A U N P O W T A W N G I H M M H A G C N S C W N Z B M B Z E C N K Q B V E M Q T T V O L A K G M A U T W U O A U K W T T Q A I O R H I N Z N I M D E O B R A J O L W C V Y J J B R R D A M G A C E G A T S A P W T S K P R F P J U Y T L C Q E K M M Y T A D H D D V M V R G K N A S B R J R P L L B S A P M U N D K N H J B K O I CANDY COOKIE CRISPS DUMPLING FRIES HAMBURGER HOTDOG MEATBALL NOODLE | Isi file medium3.txt. Tema dari word puzzle adalah nama makanan umum. |
| Please input your file name in the test folder: medium3.txt C E G A T S A P W T S K P R F P J U Y T L C Q E K M M Y T A D H D D V M V R G K N A S B R J R P L L B S A P M U N D K N H J B K O I ========= Word Found! ========= Program took 122 step(s) to find the word Exited in 0.23100 ms J S P B C S H M X T C G X Z R V A P J V Z P C Z N Y U G B S I M T S D Z U T Z D I T N H E R Y Y J L K K L H Y X S L A G J V T C P G T A Q M G C Z E B V X U W P R B M C K B F K A F L E B J J H T A I P X I D K E E G J L M P V Y N P K C Z C C K L K Q F E A L K X A S S I Q M V E R P O S A C W G N L T D T O Q A Z N Z M V Z H P R Y L G C M J C B O R J H P H O Z Z W C Q J Z V W T X T B A A O K F O C J L S V A Z B M M A P K P S O R L N R E T Z U V R Q X G K Q Q J X S M I G F L I S P D E U X N T K L L Z N V S D P M Y N E I U K O N V Y C R I S P S T V D K N V K S I Y L O G W E E K H B K A R H W V N F U D E R L L G W L K C V H K E T A Q S D X Z C S A U N P O W T A W N G I H M M H A G C N S C W N Z B M B Z E C N K Q B V E M Q T T V O L A K G M A U T W U O A U K W T T Q A I O R H I N Z N I M D E O B R A J O L W C V Y J J B R R D A M G A C E G A T S A P W T S K P R F P J U Y T L C Q E K M M Y T A D H D D V M V R G K N A S B R J R P L L B S A P M U N D K N H J B K O I Program fully exited in 337.95322 ms Press any key to continue | Input nama file ke dalam program Bagian akhir dari eksekusi program. Bagian atas tidak terlihat karena output program akan mencetak array setiap kali sebuah kata ditemukan. |

3.7.Ukuran besar 1

| Gambar | Keterangan |
|---|---|
| P | Isi file large1.txt. Tema dari word puzzle adalah nama negara di seluruh dunia. |
| THAILAND VIETNAM Please input your file name in the test folder: large1.txt | Input nama file ke dalam program |
| K M O H N I O U K A O N H T L C U U O P G S T F E Y L X A W M X J Q B T O N V M T W X C N G P E P A O J J T L X B F R M Z W E H R L E L C R H A U R Y W E Y Z Q K Y I D D X Y H B Z G C A O F N H E Y Y E """ """ """ """ """ """ | Bagian akhir dari eksekusi program. Bagian atas tidak terlihat karena output program akan mencetak array setiap kali sebuah kata ditemukan. |

3.8.Ukuran besar 2

| Gambar | Keterangan |
|---|--|
| L T K M F S A F M Q S E D X R Z V D S X S S L S H H J L C S R D Q Q G G G F M O V C Y G H L D K L Y U Y X U H M E J U B G D C G O Z A P C Y Y Y E K G L C F G G T P F E D S V Q J X A T Q I I K B O O K O H L H G F O V H S O C I T Y U K X I N W U N U D R V Q W X T Z U W T N P D U G Y O R E T W G M W Y I X M I D R U S H L W A P Y L U N J H H L J L C P O I H S I B U S T I M B F J R E Q V F X O Y U H G Y G R G X L K P O F C W T I Z Z B M T Z C H S A O E N X W W U C N A U R E N F O U K H F R I O L F R B B O B L Z M L I I O P F S U U E L M C M K H G J E O O K C M W H Y K L H G F X A I X Z W W W Q C I K E G D X X D V F C N M F N T P Y W V H N T P H Z U Y N P G S N M M P Q B C X D V F C N M F N T P Y W V H N T P H Z U Y N P G S N M M P Q B C Y K X Q S M C D N B K W P T K M T A F H U T T R C M H A V H R L Z R W K D W M P S A Q A I D S J N E B I T V Y U N S B U A N B H Y L G I W X H L T P C U V M I P D Y S O N U R G B C P L J V Y D I L Q L V Y W G L P N L Q B D K S X L S X P N P E T E A J F D H G T V F L Z C Z X N R M F I W A P R A C H W G Z A E Q R F R G Q D F T S N H W H G Z Z O S F I A I R R K P Q H W L E I H K A C P S D W L T S D E G Z P A N M E C S L O O U I R A R R E F E Y S O U V G Z Q M J E V Z U M A I E B C E L I N J T X S V C E B X E P K V O E R L G G U R F E C J S Z Q I S X T K I D D G L H D J T L T H F M R V L M Z Q O Y K U I Q H N I S S A N J T C U N J C D E T P F T R M B J L Y E L E I A X T Y T Z S M S W G X E K H T H T C N A A U L A D T Q E O E E V U B H Q R S J I R D M A S O C P G O X E G X A W A U H Q H T Y N D K F Y Y Q R L H O I B L F K Q G K B B N U J M F G R Z T U E M I X V A I V U V D Z B B A L X N O O L G T R J Y Z J E A T H U O N L T A N H V U N D F V O W C Q X G U R Q A D Z X V R D S C P G O X E G X A W A U S Q U R Q A D Z X V R D S S A D T C U N J M F G R Z T U E M I X V A I V U V D Z B B A L X N O O L G T R J Y Z J E A T H U O N L U A N H V U V D Z B B A L X N O O L G T R J Y Z J E A T H U O N L U A C F O Q S Z H M O Y A H K K A D A T Q F E K G B F O S C B F T J I V T A L N X P J V U T K L A O V | Isi file large2.txt. Tema dari word puzzle adalah nama brand mobil di seluruh dunia. |
| volvo Please input your file name in the test folder: large2.txt_ | Input nama file ke dalam program |
| N A Y A P K F N A C F O Q S Z H N O Y A H K K A D A T Q F E K G B F O S C B F T J I V Y A L M X P J Y U R T K I A O N V I R F Y O N U G F A A H U I L M R G B Y J M G D T U Y E Z Q T B M L V X D U E B F B ENTER HOR F TO M I S T S T S T S T S T S T S T S T S T S | Bagian akhir dari eksekusi program. Bagian atas tidak terlihat karena output program akan mencetak array setiap kali sebuah kata ditemukan. |

3.9.Ukuran besar 3

| Gambar | Keterangan |
|--|-------------------------------------|
| D | Isi file large3.txt. |
| S N K T D S H R E D N J L B A R S M Q M B R Q G Y M X Y R G G O K W M J K D A T S V Y I J I S S B C S L D A K B V D C W R S U C B F B K A B U R O P F F O J A H B V V T O K O E N J | Tema dari word puzzle adalah nama |
| R T A P A M B J A M J H Z I S B P G A L I X K B U N N M P D T W E O T U I Q Y P U E E W F X E L D I C E O H Q V F T O F | senjata di game Valorant. |
| I F F K C Q Y B Z A D O Z J Y T R A E Y M Y N V A R Y B O G J U D G E X H Z O G N Y J I J N A D Q C O W H N E Z B I Q O E A R J A J K U P L U S Z X G U T Y A H T N A N G D U L X G | |
| V J U U R J Z O P C G G H J G E O K E A N L Z X H G N W G Q T T N K Q W R E Y U I U D E C V R V L D A Y P N H G U I H J | |
| FFSLQQAGXWFHBPRZICNSHURUOLBCWI MEFJLMZMIIRMWSTRKZBEPEMBEVTWNM HMAPMFPAZIKJWWHDILMWGMLCPBAEIE | |
| 0 0 W T W T Z U K A G U Y B N F N F L N V H B W O X X X P U L V F M Q L H K G Y L R J G U A R D I A N O X H Z Z W Z M C | |
| A L L W M T U Y P N V G P N B E T T G X Y J P E B K U K W K G H X J I C Y Y R Z E B B M A Q S D O U M M W H C T M K Q D Z J U B Q Y K Z T D E Q Y F H I V O J T O U C Y A M V B O A | |
| Q C A E Q Y W V F A X O B L E Q Y T R U G Y A X Y E Y Z A Z B H G O P O D V C A N C D D A V A B S Y H N M B Q C M X Y O | |
| G E J V X U U Y U M H Z M X P D N N H E O B H S Z W T L D E O L K O B C A N E T V V I J M K N Y M R S R U B K T X Q M Y B A X B W T N G E B V T B C L S K A H T T J P L T F Y L J E | |
| U Z G D F M T P C U T S T V P C Q V V C N A A Y L C N K L A R Y H T T R C D K N K X W H U Y E G O E U K G V I D X V O E | |
| H X N I D O D M N V S F U B G P E G I P S P O G Z G O M U E I T Y K P L Y H L Z H W F U O V I B R S C I A A B J L G I S O O E C R H G X W X Y Z P V H J I F J N C Z T X N I L E R B | |
| ARES | |
| BUCKY BULLDOG CLASSIC | |
| FRENZY GHOST | |
| GUARDIAN JUDGE MARSHAL | |
| ODIN OPERATOR PHANTOM | |
| SHERRIF SHORTY | |
| SPECTRE STINGER VANDAL | |
| Please input your file name in the test folder: large3.txt | Input nama file ke dalam program |
| H X N I D O D M N V S F U B G P E G I P S P O G Z G O M U E I T Y K P L Y H L Z H W F U O V I B R S C I A A B J L G I S O Q E C R H G X W X Y Z P V H J I F J N C Z T X N I L E R B | Bagian akhir dari eksekusi program. |
| Program took 796 step(s) to find the word Exited in 0.34030 ms DDLMPSGMIVPUQDCRVYTROHSMIHFMZT | Bagian atas tidak terlihat karena |
| I Q Z O M A E Z Z F L B J T D I R F J H X S A O S F C G H R S N K T D S H R E D N J L B A R S M Q M B R Q G Y M X Y R G G O K W M J K D A T S V Y I J I S S B C S L D A K B V D C W R S U C B F B K A B U R O P F F O J A H B V V T O K O E N J | output program akan mencetak array |
| R T A P A M B J A M J H Z I S B P G A L I X K B U N N M P D T M E O T U I Q Y P U E E W F X E L D I C E O H Q V F T O F I F F K C Q Y B Z A D O Z J Y T R A E Y M Y N V A R Y B O G | setiap kali sebuah kata ditemukan. |
| EARJAJKUPLUSZXGUTYÄHTNANGDULXG VJUURJZOPCGGHJGEOKEANLZXHGNWGQ TTNKQWREYUIUDECVRVLDAYPNHGUIHJ | |
| FFSLQQAGXWFHBPRZICNSHURUOLBCWI MEFJLMZMIIRMWSTRKZBEPEMBEVTWNM HMAPMFPAZIKJWWHDILMWGMLCPBAEIE OOWTWTZUKAGUYBNFNFLNVHBWOXXXPU | |
| L V F M Q L H K G Y L R J G U A R D I A N O X H Z Z W Z M C A L L W M T U Y P N V G P N B E T T G X Y J P E B K U K W K G H X J I C Y Y R Z E B B M A Q S D O U M M W H C T M K Q D Z J U B Q Y K Z T D E Q Y F H I V O J T O U C Y A M V B O A | |
| Q C A E Q Y W V F A X Ô B L E Q Y T R U G Y A X Y E Y Z A Z B H G O P O D V C A N C D D A V A B S Y H N M B Q C M X Y O G E J V X U U Y U M H Z M X P D N N H E O B H S Z W T L D E O L K O B C A N E T V V I J M K N Y M R S R U B K T X O M Y | |
| BAXBWTNGEBVTBCLSKAHTTJPLTFYLJE UZGDFMTPCUTSTVPCQVVCNAAYLCNKLA RYHTTRCDKNKXWHUYEGOEUKGVIDXVOE | |
| H X N I D O D M N V S F U B G P E G I P S P O G Z G O M U E I T Y K P L Y H L Z H W F U O V I B R S C I A A B J L G I S O Q E C R H G X W X Y Z P V H J I F J N C Z T X N I L E R B Program fully exited in 838.94476 ms | |
| Press any key to continue | |

BAB IV Link Kode Program

Kode program dapat dilihat pada halaman github berikut https://github.com/YakobusIP/Tugas-Kecil-Stima-1-Word-Puzzle.git

BAB V Daftar Referensi

Munir, Rinaldi. 2022. Algoritma *Brute Force*. Diakses pada 24 Januari 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag1.pdf.