

LAPORAN TUGAS KECIL I

PENYELESAIAN *WORD SEARCH PUZZLE*

DENGAN ALGORITMA *BRUTE FORCE*

Laporan dibuat untuk memenuhi salah satu tugas mata kuliah

IF2211 Strategi Algoritma



Disusun oleh:

Hilda Carissa Widelia 13520164

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2022

DAFTAR ISI

DAFTAR ISI	1
Algoritma Brute Force	2
Source Program	2
Screenshot Input dan Output	2
Link Drive Kode Program	2
Checklist	2
Daftar Referensi	2

Algoritma Brute Force

Dalam pengerjaan tugas kecil ini, algoritma *brute force* sangat diutamakan. Dalam program yang dibuat penulis, setelah menerima input kata yang harus dicari, penulis langsung mencari indeks dari huruf pertama kata yang dicari. Setelah menemukan indeks tersebut, dipanggil fungsi yang berfungsi untuk mencari arah, dapat dibilang fungsi ini juga yang menjadi fungsi utama untuk mencocokkan kata yang dicari dengan huruf dalam puzzle.

Memasuki fungsi ini, atau yang dalam program disebut fungsi `findArah`, pertama ditentukan dulu *value* dari variabel arah. Disini arah direpresentasikan dengan angka 1 hingga 8, dimana angka 1 adalah arah atas, 2 arah kanan atas, 3 arah kanan, dan seterusnya mengikuti arah jarum jam. Jika didapat bahwa ternyata huruf kedua yang ada di puzzle tidak sesuai dengan huruf kedua dari kata yang dicari, maka nilai arah ditambah 1. Sedangkan jika huruf kedua cocok, maka variabel penghitung panjang huruf akan ditambah 1. Saat nilai variabel penghitung panjang huruf tersebut sudah sama dengan panjang kata yang dicari, maka kata sudah ditemukan, dan langsung masuk ke fungsi untuk mencetak jawaban. Namun, jika semua arah sudah dicek dan masih tidak dapat ditemukan katanya, maka akan kembali ke fungsi utama dan mencari indeks lain untuk huruf pertama kata yang dicari.

Setelah kata ditemukan, akan masuk ke fungsi yang bernama `printMatrixJawab`, disini jawaban akan dicetak sesuai dengan arah yang sudah didapatkan sebelumnya. Untuk setiap arah, terdapat batasan yang berbeda agar kata yang dimaksud yang di print. Jika indeks yang sedang diproses tidak sesuai dengan batasan, akan diprint karakter “-”.

Source Program

```
1 package src;
2 import java.io.*;
3 import java.util.*;
4 import java.nio.file.Path;
5 import java.nio.file.WatchEvent;
6
7 public class App{
8     public static void main(String[] args){
9         Scanner sc = new Scanner(System.in);
10         int row=-1, col=-1; //word length counter and counter for word comparison
11         int[] find = {0,0}; // 1 0 for found, counter per kata
12         char[][] matrix;
13         String search;
14         String namaf;
15         System.out.println("Welcome to Word Search Solver!\nPlease insert the puzzle file name : ");
16         namaf = sc.nextLine();
17         try{
18             namaf = (Path.of("../test", namaf)).toString();
19             File myFile = new File(namaf);
20             Scanner scnn = new Scanner(myFile);
21             int m = 0; //total row
22             int n = 0; //total col
23             String data=scnn.nextLine();
24             while(data.length() != 0){
25                 String array[] = data.split(" ");
26                 n = 0;
27                 for (String i : array){
28                     n++;
29                 }
30                 m++;
31                 data = scnn.nextLine();
32             }
33             scnn.close();
34 }
```

Gambar 1.1 Main program dan menghitung baris dan kolom puzzle

```
34
35         File myFile2 = new File(namaf);
36         Scanner scn = new Scanner(myFile2);
37         matrix = new char[m][n];
38         int i = 0;
39         String data1=scn.nextLine();
40         while(data1.length()!=0){
41             String rows = data1.replaceAll("\\s","");
42             char[] array2 = rows.toCharArray();
43             for (int j = 0; j < n; j++){
44                 matrix[i][j] = array2[j];
45             }
46             i++;
47             data1 = scn.nextLine();
48         }
```

Gambar 1.2 Memasukkan puzzle dari file ke matrix

```

49     long startAllTime = System.nanoTime();
50     while(scen.hasNextLine()){
51         search = scen.nextLine();
52         long startTime = System.nanoTime();
53         System.out.println("Sedang mencari " + search);
54         find[0] = 0;
55         find[1] = 0;
56         // cari huruf pertama
57         for(int k = 0; k < m; k++){
58             for(int l = 0; l < n; l++){
59                 if (matrix[k][l] == search.charAt(0)){
60                     row = k;
61                     col = l;
62                     find = findArah(matrix, m, n, row, col, search, find);
63                     if(find[0] == 1){
64                         break;
65                     }else{
66                         row = -1;
67                         col = -1;
68                     }
69                 }else{
70                     find[1]++;
71                 }
72             }
73             if(find[0] == 1){
74                 break;
75             }
76         }
77         long elapsedTime = System.nanoTime()-startTime;
78         System.out.println("Total execution time in milis: " + elapsedTime/1000000);
79     }
80     scen.close();
81     long finalTime = System.nanoTime()-startAllTime;
82     System.out.println("Total all execution time in milis: " + finalTime/1000000);
83 } catch (FileNotFoundException e){
84     System.out.println("File not found");
85 }
86 sc.close();

```

Gambar 1.3 Mencari kata di puzzle dan setelah menemukan menuliskan total waktu eksekusi

```

90 public static void printMatrixJawab(char[][] matrix, int row, int col, int arah, int n, int m, int wlength, int attCounter){
91     if(arah == 1){ // atas
92         for(int i = 0; i < m; i++){
93             for(int j = 0; j < n; j++){
94                 if(j == col && (i >= row-wlength+1 && i <= row)){
95                     System.out.print(matrix[i][j] + " ");
96                 }else{
97                     System.out.print("- ");
98                 }
99             }
100             System.out.println();
101         }
102     }else if(arah == 2){ // kanan atas
103         int wc = wlength;
104         for(int i = 0; i < m; i++){
105             for(int j = 0; j < n; j++){
106                 if((i <= row && i >= row - wlength+1) && (j >= col && j <= col+wlength) && i+j == row+col && wc>0){
107                     System.out.print(matrix[i][j] + " ");
108                     wc--;
109                 }else{
110                     System.out.print("- ");
111                 }
112             }
113             System.out.println();
114         }
115     }else if(arah == 3){ // kanan
116         for(int i = 0; i < m; i++){
117             for(int j = 0; j < n; j++){
118                 if(i == row && (j >= col && j <= col+wlength-1)){
119                     System.out.print(matrix[i][j] + " ");
120                 }else{
121                     System.out.print("- ");
122                 }
123             }
124             System.out.println();
125         }

```

Gambar 1.4 Print matrix jawaban

```

126         }else if(arah == 4){ // kanan bawah
127             int wc = 0;
128             for(int i = 0; i < m; i++){
129                 for (int j = 0; j < n; j++){
130                     if(i >= row && j >= col && i-j == row-col && wc<=wlength-1){
131                         System.out.print(matrix[i][j] + " ");
132                         wlength--;
133                     }else{
134                         System.out.print("- ");
135                     }
136                 }
137                 System.out.println();
138             }
139
140         }else if(arah == 5){ // bawah
141             for(int i = 0; i < m; i++){
142                 for (int j = 0; j < n; j++){
143                     if(j == col && (i >= row && i <= row+wlength-1)){
144                         System.out.print(matrix[i][j] + " ");
145                     }else{
146                         System.out.print("- ");
147                     }
148                 }
149                 System.out.println();
150             }
151         }else if(arah == 6){ // kiri bawah
152             for(int i = 0; i < m; i++){
153                 for(int j = 0; j < n; j++){
154                     if(i + j == row+col && i >= row && wlength > 0 ){
155                         System.out.print(matrix[i][j] + " ");
156                         wlength--;
157                     }else{
158                         System.out.print("- ");
159                     }
160                 }
161                 System.out.println();
162             }

```

Gambar 1.5 Print matrix jawaban

```

163         }else if(arah == 7){ // kiri
164             for(int i = 0; i < m; i++){
165                 for(int j = 0; j < n; j++){
166                     if(i == row && (j >= col-wlength+1 && j <= col)){
167                         System.out.print(matrix[i][j] + " ");
168                     }else{
169                         System.out.print("- ");
170                     }
171                 }
172                 System.out.println();
173             }
174         }else if(arah == 8){ // kiri atas
175             int wc = wlength;
176             for(int i = 0; i < m; i++){
177                 for(int j = 0; j < n; j++){
178                     if(i >= row-wlength+1 && j >= col-wlength+1 && i-j == row-col && wc>0){
179                         System.out.print(matrix[i][j] + " ");
180                         wc--;
181                     }else{
182                         System.out.print("- ");
183                     }
184                 }
185                 System.out.println();
186             }
187         }
188         System.out.println("Banyak percobaan : " + attCounter);
189     }

```

Gambar 1.6 Print matrix jawaban

```

191 public static int[] findArah(char[][] matrix, int m, int n, int row, int col, String search, int[] find){
192     int wordlength = search.length()-1;
193     if(row>-1 && col>-1){
194         int wlc = 1, arah = 1, o = row, p = col;
195         // o p counter buat index satu satu
196         while(wlc < wordlength+1 && arah < 9){
197             if(arah == 1){ //keatas
198                 if(row < wordlength){
199                     arah++;
200                 }else{
201                     o--;
202                     if(matrix[o][p]==search.charAt(wlc)){
203                         wlc++;
204                     }else{
205                         wlc = 1;
206                         arah++;
207                         o = row;
208                         p = col;
209                     }
210                 }
211                 find[1]++;
212             }else if(arah == 2){ //ke kanan atas
213                 if(n-col < wordlength | row < wordlength){
214                     arah++;
215                 }else{
216                     o--;
217                     p++;
218                     if(matrix[o][p]==search.charAt(wlc)){
219                         wlc++;
220                     }else{
221                         wlc = 1;
222                         arah++;
223                         o = row;
224                         p = col;
225                     }
226                 }
227                 find[1]++;

```

Gambar 1.7 Fungsi mencocokkan kata

```

228             }else if(arah == 3){ //ke kanan
229                 if(n-col < wordlength){
230                     arah++;
231                 }else{
232                     p++;
233                     if(matrix[o][p]==search.charAt(wlc)){
234                         wlc++;
235                     }else{
236                         wlc = 1;
237                         arah++;
238                         o = row;
239                         p = col;
240                     }
241                 }
242                 find[1]++;
243             }else if(arah == 4){ // kanan bawah
244                 if(n-col < wordlength | m-row < wordlength){
245                     arah++;
246                 }else{
247                     o++;
248                     p++;
249                     if(matrix[o][p]==search.charAt(wlc)){
250                         wlc++;
251                     }else{
252                         wlc = 1;
253                         arah++;
254                         o = row;
255                         p = col;
256                     }
257                 }
258                 find[1]++;

```

Gambar 1.8 Fungsi mencocokkan kata


```

259         }else if(arah == 5){ // bawah
260             if(m-row < wordlength){
261                 arah++;
262             }else{
263                 o++;
264                 if(matrix[o][p]==search.charAt(wlc)){
265                     wlc++;
266                 }else{
267                     wlc = 1;
268                     arah++;
269                     o = row;
270                     p = col;
271                 }
272             }
273             find[1]++;
274         }else if(arah == 6){ // bawah kiri
275             if(m-row < wordlength | col < wordlength){
276                 arah++;
277             }else{
278                 o++;
279                 p--;
280                 if(matrix[o][p]==search.charAt(wlc)){
281                     wlc++;
282                 }else{
283                     wlc = 1;
284                     arah++;
285                     o = row;
286                     p = col;
287                 }
288             }
289             find[1]++;

```

Gambar 1.9 Fungsi mencocokkan kata

```

290         }else if(arah == 7){ // kiri
291             if(col < wordlength){
292                 arah++;
293             }else{
294                 p--;
295                 if(matrix[o][p]==search.charAt(wlc)){
296                     wlc++;
297                 }else{
298                     wlc = 1;
299                     arah++;
300                     o = row;
301                     p = col;
302                 }
303             }
304             find[1]++;
305         }else if(arah == 8){ // kiri atas
306             if(col < wordlength | row < wordlength){
307                 arah++;
308             }else{
309                 p--;
310                 o--;
311                 if(matrix[o][p]==search.charAt(wlc)){
312                     wlc++;
313                 }else{
314                     wlc = 1;
315                     arah++;
316                     o = row;
317                     p = col;
318                 }
319             }
320             find[1]++;
321         }

```

Gambar 1.10 Fungsi mencocokkan kata

```

322     }
323     if(arah > 8){
324         find[0] = 0;
325     }else{
326         printMatrixJawab(matrix, row, col, arah, n, m, search.length(), find[1]);
327         find[0] = 1;
328     }
329 }
330 return find;
331 }
332 }

```

Gambar 1.11 Mencetak matrix jika kata ditemukan

Screenshot Input dan Output

```
Welcome to Word Search Solver!  
Please insert the puzzle file name :  
Small1.txt
```

Sedang mencari CARRY

CARRY

```
Banyak percobaan : 177
Total execution time in milis: 48
Sedang mencari CLEAN
```

[illegible]

```
Banyak percobaan : 276
Total execution time in milis: 25
```

Sedang mencari DRAW

A 10x10 grid of dots. The letters W, A, R, and D are placed at the bottom left of the grid, starting from the first column and moving right across the rows.

```
Banyak percobaan : 289
Total execution time in milis: 142
Sedang mencari DRINK
```

A 10x10 grid of dots. The letters are placed at the following intersections (row, column) starting from the top-left:

- D: (9, 1)
- R: (9, 2)
- I: (8, 3)
- N: (7, 4)
- K: (6, 5)

```
Banyak percobaan : 272
Total execution time in milis: 124
```

```
Total execution time in milis: 156
```

A 20x20 grid of dots on a black background. The letters K, E, E, and P are placed at specific grid intersections. K is at row 14, column 7. The first E is at row 12, column 10. The second E is at row 10, column 12. P is at row 8, column 14.

```
Total execution time in milis: 110
```

```
Total execution time in millis: 151
```

MUCH

```
Total execution time in milis: 125
```

A 20x20 grid of dots. The letters are placed at the following intersections (row, column) starting from the top-left:

- M: (18, 10)
- Y: (17, 12)
- S: (16, 14)
- E: (15, 16)
- L: (14, 18)
- F: (13, 20)

```
Total execution time in milis: 112
```

P I C K

```
Total execution time in millis: 72
```

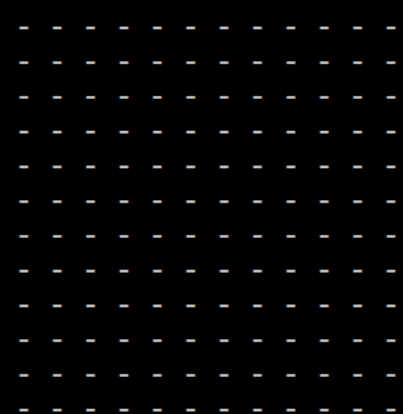
A 10x10 grid of dots. The letters are placed at the following intersections (row, column) starting from the top-left:

- N: (9, 3)
- E: (9, 4)
- V: (8, 5)
- E: (7, 5)
- S: (6, 6)

```
Total execution time in milis: 146
```

[illegible]

```
Total execution time in millis: 105
```



Y
A
D
O
T

A 20x20 grid of dots. The letters Y, R, and T are placed at the intersections of the grid lines. Y is at row 5, column 10. R is at row 6, column 11. T is at row 7, column 12.

13 - IF2211 - Strategi Algoritma

Sedang mencari ALBANY

[illegible]

A 10x10 grid of dots. The word "ANSWER" is written diagonally across the grid, starting from the bottom-left and moving towards the top-right. The letters are positioned as follows: 'A' at (row 10, col 1), 'N' at (row 9, col 2), 'S' at (row 8, col 3), 'W' at (row 7, col 4), 'E' at (row 6, col 5), 'R' at (row 5, col 6).

Sedang mencari AUSTIN

AUSTIN

[illegible]

14 - IF2211 - Strategi Algoritma

[illegible]

Sedang mencari CHARLESTON

```
Total execution time in milis: 39
```

A 10x10 grid of dots. The letters R, E, and V are placed at the following intersections (row, column): R at (4, 3), E at (5, 4), and V at (6, 5).

Sedang mencari HARTFORD

A large rectangular area filled with a uniform grid of small black dots, intended for students to write their answers.

Banyak percobaan : 315

```
Total execution time in millis: 47
```


- Y T I C N O S R E F F E J -

```
Total execution time in millis: 64
```

A 20x20 grid of dots. The word "GNISAL" is spelled out by removing dots in a specific pattern. The letters are formed by the following dot coordinates (row, column) where (0,0) is the top-left dot:

- G: (5,1), (5,2), (5,3), (5,4), (5,5), (5,6), (5,7), (5,8), (5,9), (5,10), (5,11), (5,12), (5,13), (5,14), (5,15), (5,16), (5,17), (5,18), (5,19), (5,20), (5,21), (5,22), (5,23), (5,24), (5,25), (5,26), (5,27), (5,28), (5,29), (5,30), (5,31), (5,32), (5,33), (5,34), (5,35), (5,36), (5,37), (5,38), (5,39), (5,40), (5,41), (5,42), (5,43), (5,44), (5,45), (5,46), (5,47), (5,48), (5,49), (5,50), (5,51), (5,52), (5,53), (5,54), (5,55), (5,56), (5,57), (5,58), (5,59), (5,60), (5,61), (5,62), (5,63), (5,64), (5,65), (5,66), (5,67), (5,68), (5,69), (5,70), (5,71), (5,72), (5,73), (5,74), (5,75), (5,76), (5,77), (5,78), (5,79), (5,80), (5,81), (5,82), (5,83), (5,84), (5,85), (5,86), (5,87), (5,88), (5,89), (5,90), (5,91), (5,92), (5,93), (5,94), (5,95), (5,96), (5,97), (5,98), (5,99), (5,100), (5,101), (5,102), (5,103), (5,104), (5,105), (5,106), (5,107), (5,108), (5,109), (5,110), (5,111), (5,112), (5,113), (5,114), (5,115), (5,116), (5,117), (5,118), (5,119), (5,120), (5,121), (5,122), (5,123), (5,124), (5,125), (5,126), (5,127), (5,128), (5,129), (5,130), (5,131), (5,132), (5,133), (5,134), (5,135), (5,136), (5,137), (5,138), (5,139), (5,140), (5,141), (5,142), (5,143), (5,144), (5,145), (5,146), (5,147), (5,148), (5,149), (5,150), (5,151), (5,152), (5,153), (5,154), (5,155), (5,156), (5,157), (5,158), (5,159), (5,160), (5,161), (5,162), (5,163), (5,164), (5,165), (5,166), (5,167), (5,168), (5,169), (5,170), (5,171), (5,172), (5,173), (5,174), (5,175), (5,176), (5,177), (5,178), (5,179), (5,180), (5,181), (5,182), (5,183), (5,184), (5,185), (5,186), (5,187), (5,188), (5,189), (5,190), (5,191), (5,192), (5,193), (5,194), (5,195), (5,196), (5,197), (5,198), (5,199), (5,200), (5,201), (5,202), (5,203), (5,204), (5,205), (5,206), (5,207), (5,208), (5,209), (5,210), (5,211), (5,212), (5,213), (5,214), (5,215), (5,216), (5,217), (5,218), (5,219), (5,220), (5,221), (5,222), (5,223), (5,224), (5,225), (5,226), (5,227), (5,228), (5,229), (5,230), (5,231), (5,232), (5,233), (5,234), (5,235), (5,236), (5,237), (5,238), (5,239), (5,240), (5,241), (5,242), (5,243), (5,244), (5,245), (5,246), (5,247), (5,248), (5,249), (5,250), (5,251), (5,252), (5,253), (5,254), (5,255), (5,256), (5,257), (5,258), (5,259), (5,260), (5,261), (5,262), (5,263), (5,264), (5,265), (5,266), (5,267), (5,268), (5,269), (5,270), (5,271), (5,272), (5,273), (5,274), (5,275), (5,276), (5,277), (5,278), (5,279), (5,280), (5,281), (5,282), (5,283), (5,284), (5,285), (5,286), (5,287), (5,288), (5,289), (5,290), (5,291), (5,292), (5,293), (5,294), (5,295), (5,296), (5,297), (5,298), (5,299), (5,300), (5,301), (5,302), (5,303), (5,304), (5,305), (5,306), (5,307), (5,308), (5,309), (5,310), (5,311), (5,312), (5,313), (5,314), (5,315), (5,316), (5,317), (5,318), (5,319), (5,320), (5,321), (5,322), (5,323), (5,324), (5,325), (5,326), (5,327), (5,328), (5,329), (5,330), (5,331), (5,332), (5,333), (5,334), (5,335), (5,336), (5,337), (5,338), (5,339), (5,340), (5,341), (5,342), (5,343), (5,344), (5,345), (5,346), (5,347), (5,348), (5,349), (5,350), (5,351), (5,352), (5,353), (5,354), (5,355), (5,356), (5,357), (5,358), (5,359), (5,360), (5,361), (5,362), (5,363), (5,364), (5,365), (5,366), (5,367), (5,368), (5,369), (5,370), (5,371), (5,372), (5,373), (5,374), (5,375), (5,376), (5,377), (5,378), (5,379), (5,380), (5,381), (5,382), (5,383), (5,384), (5,385), (5,386), (5,387), (5,388), (5,389), (5,390), (5,391), (5,392), (5,393), (5,394), (5,395), (5,396), (5,397), (5,398), (5,399), (5,400), (5,401), (5,402), (5,403), (5,404), (5,405), (5,406), (5,407), (5,408), (5,409), (5,410), (5,411), (5,412), (5,413), (5,414), (5,415), (5,416), (5,417), (5,418), (5,419), (5,420), (5,421), (5,422), (5,423), (5,424), (5,425), (5,426), (5,427), (5,428), (5,429), (5,430), (5,431), (5,432), (5,433), (5,434), (5,435), (5,436), (5,437), (5,438), (5,439), (5,440), (5,441), (5,442), (5,443), (5,444), (5,445), (5,446), (5,447), (5,448), (5,449), (5,450), (5,451), (5,452), (5,453), (5,454), (5,455), (5,456), (5,457), (5,458), (5,459), (5,460), (5,461), (5,462), (5,463), (5,464), (5,465), (5,466), (5,467), (5,468), (5,469), (5,470), (5,471), (5,472), (5,473), (5,474), (5,475), (5,476), (5,477), (5,478), (5,479), (5,480), (5,481), (5,482), (5,483), (5,484), (5,485), (5,486), (5,487), (5,488), (5,489), (5,490), (5,491), (5,492), (5,493), (5,494), (5,495), (5,496), (5,497), (5,498), (5,499), (5,500), (5,501), (5,502), (5,503), (5,504), (5,505), (5,506), (5,507), (5,508), (5,509), (5,510), (5,511), (5,512), (5,513), (5,514), (5,515), (5,516), (5,517), (5,518), (5,519), (5,520), (5,521), (5,522), (5,523), (5,524), (5,525), (5,526), (5,527), (5,528), (5,529), (5,530), (5,531), (5,532), (5,533), (5,534), (5,535), (5,536), (5,537), (5,538), (5,539), (5,540), (5,541), (5,542), (5,543), (5,544), (5,545), (5,546), (5,547), (5,548), (5,549), (5,550), (5,551), (5,552), (5,553), (5,554), (5,555), (5,556), (5,557), (5,558), (5,559), (5,560), (5,561), (5,562), (5,563), (5,564), (5,565), (5,566), (5,567), (5,568), (5,569), (5,570), (5,571), (5,572), (5,573), (5,574), (5,575), (5,576), (5,577), (5,578), (5,579), (5,580), (5,581), (5,582), (5,583), (5,584), (5,585), (5,586), (5,587), (

```
Total execution time in millis: 40
```

MONTGOMERY

```
Total execution time in millis: 47
```

NASHVILLE

```
Total execution time in millis: 61
```


[illegible]

Sedang mencari CHESS

CHES

```
Total execution time in milis: 137
```

CONK

Sedang mencari COVER

REVOUC

```
Total execution time in milis: 153
```

[illegible]

Sedang mencari DISTINCTION

A 20x20 grid of dots. The words 'DUST', 'TIN', 'COT', and 'ION' are arranged diagonally from the bottom-left to the top-right. 'DUST' is at the bottom-left, 'TIN' is above it, 'COT' is above that, and 'ION' is at the top-right. Each word is formed by dots, with some dots also forming the letters of other words in the diagonal.

```
Total execution time in milis: 170
```

A 10x10 grid of dots. The word "DOLAR" is written vertically in the center of the grid, with each letter occupying one row and one column. The letters are D, O, L, A, R, O, L, A, R, D from top to bottom.

Sedang mencari DOSES

A 20x20 grid of dots. The word "DESOS" is written diagonally across the grid, starting from the bottom-left and moving towards the top-right. The letters are positioned as follows: 'D' at row 15, column 10; 'E' at row 14, column 11; 'S' at row 13, column 12; 'O' at row 12, column 13; and 'S' at row 11, column 14.

```
Total execution time in millis: 125
```



DUMFOUND

N
 O
 I
 T
 R
 O
 T
 X
 E

Y
T
L
U
A
F


```
- - - - - E V I T A M R I F F A -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
Banyak percobaan : 79  
Total execution time in milis: 78
```

[illegible][illegible]

```
Sedang mencari COURSEOF - - - - -  
- - - - -  
- - - - -  
C O U R S E O F - - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
Banyak percobaan : 111  
Total execution time in milis: 96
```


R
E
V
E
L

```
Banyak percobaan : 467
Total execution time in milis: 91
```

F
O
N
A
M

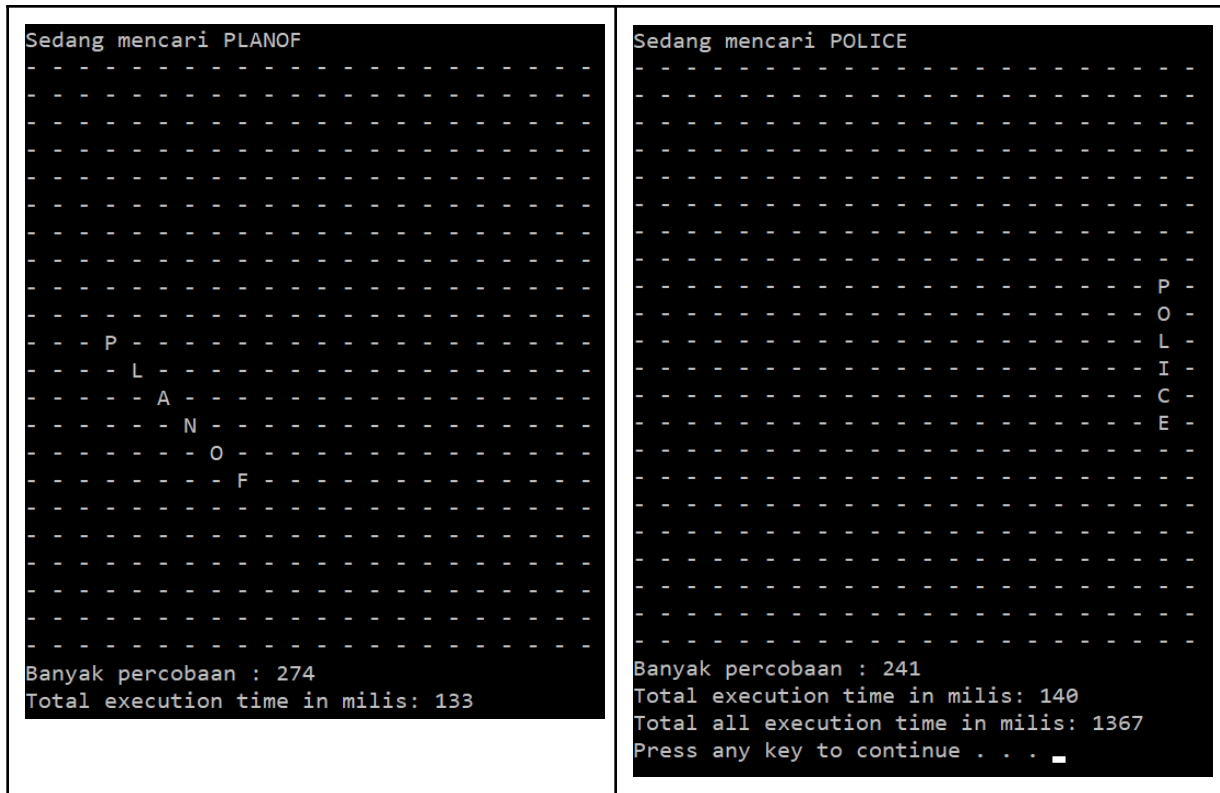
```
Banyak percobaan : 380
Total execution time in milis: 112
```

A 20x20 grid of dots. The letters M, O, and B are placed at the following intersections (row, column): M at (16, 4), O at (17, 5), and B at (18, 6).

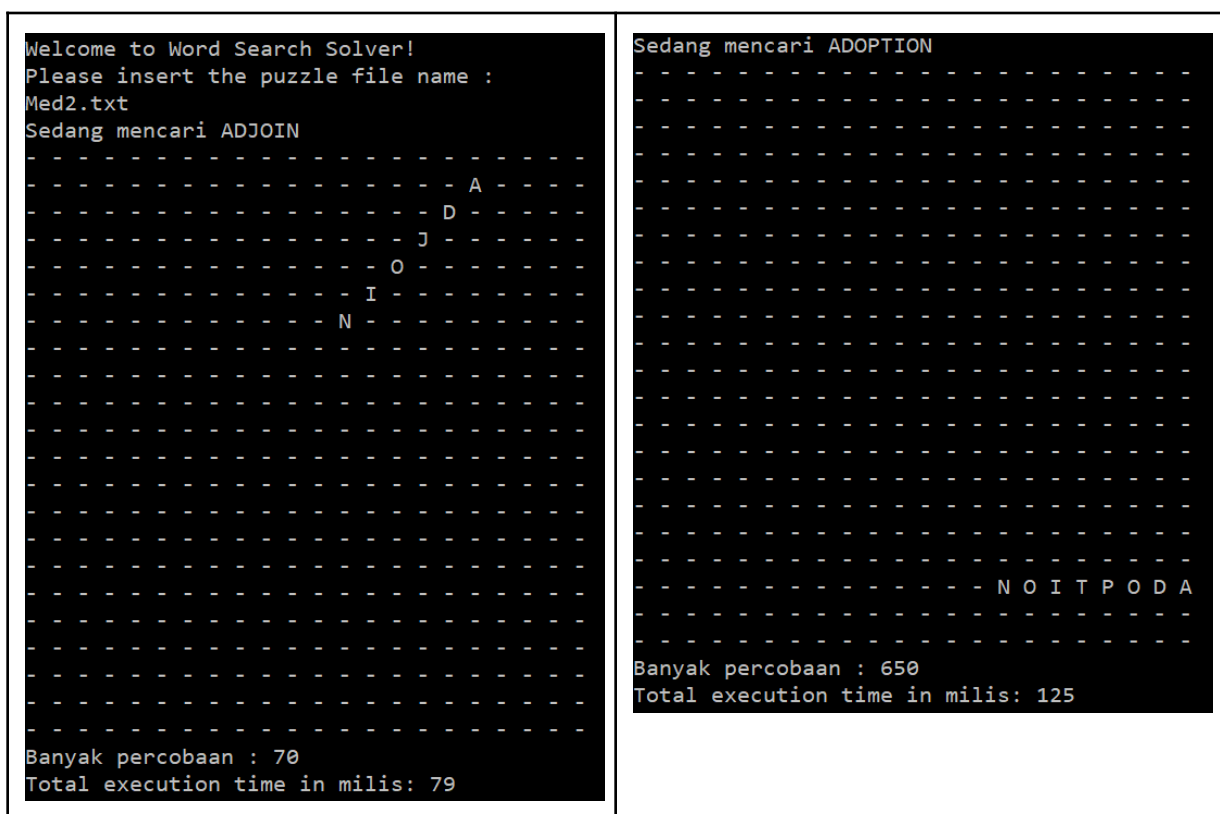
```
Banyak percobaan : 442
Total execution time in milis: 84
```

PIANO

```
Banyak percobaan : 395
Total execution time in milis: 97
```



Gambar 2.4 Input dan Output Test Case Medium 1



[illegible]

A 20x20 grid of dots with a solid black vertical bar on the left side. The letters H, C, A, O, R, K, C, O, and U are placed at specific grid intersections.

RO TUB I R T N O C

A 20x20 grid of dots on a black background. The letters 'S', 'L', 'O', and 'C' are formed by removing dots in specific patterns. 'S' is in the top right, 'L' is below it, 'O' is to the right of 'L', and 'C' is to the right of 'O'.

28 - IF2211 - Strategi Algoritma

REH SURC

U
C
R
A
T
O
R

DEAL

DISOBEDIENCE

29 - IF2211 - Strategi Algoritma

A 10x10 grid of dots. The word 'YOUNG' is written vertically in the first column, and the word 'EARTH' is written horizontally in the first row. The letters are white and stand out against the black background of the grid.

THIS IS ROF

A 20x20 grid of dots on a black background. The letters 'ENIGMA' are formed by the absence of dots in specific rows and columns. 'E' is at (10,10), 'N' at (10,11), 'I' at (10,12), 'G' at (10,13), 'M' at (10,14), 'A' at (10,15), and 'R' at (10,16).

A large grid of 100 small squares, each containing a single letter 'E'. The grid is arranged in 10 rows and 10 columns. The letters are white and centered within each square. The background is black.

30 - IF2211 - Strategi Algoritma

H
O
L
O
G
R
A
M

A 20x20 grid of dots. The letters E, L, A, and M are placed at the following coordinates (row, column): E at (10, 10), L at (10, 11), A at (10, 12), and M at (10, 13).

31 - IF2211 - Strategi Algoritma

[illegible][illegible]

32 - IF2211 - Strategi Algoritma


```
Welcome to Word Search Solver!  
Please insert the puzzle file name :  
Med3.txt  
Sedang mencari ACADEME
```

[illegible]

Sedang mencari ANNEX

```
A
N
N
E
X
```

Banyak percobaan : 345
Total execution time in milis: 116

Sedang mencari BACKPACKING

[illegible]

Sedang mencari BALLROOM

```
- - - - M O O R L L A B - - - -
```

Banyak percobaan : 48
Total execution time in milis: 114

LANRAC

C
 H
 E
 R
 F
 U
 L

EDULCINOC

ENOSITROC

35 - IF2211 - Strategi Algoritma

Sedang mencari DANGLING

- - - - - D A N G L I N G - - - - -

Banyak percobaan : 273

Total execution time in milis: 113

Sedang mencari DOUBLED

- - - - - D O U B L E D - - - - -

Banyak percobaan : 595

Total execution time in milis: 134

Sedang mencari DUPLICITY

- Y T I C I L P U D - - - - -

Banyak percobaan : 454

Total execution time in milis: 131

Sedang mencari ECTOPLASM

M
S
A
L
P
O
T
C
E

Banyak percobaan : 732

Total execution time in milis: 127

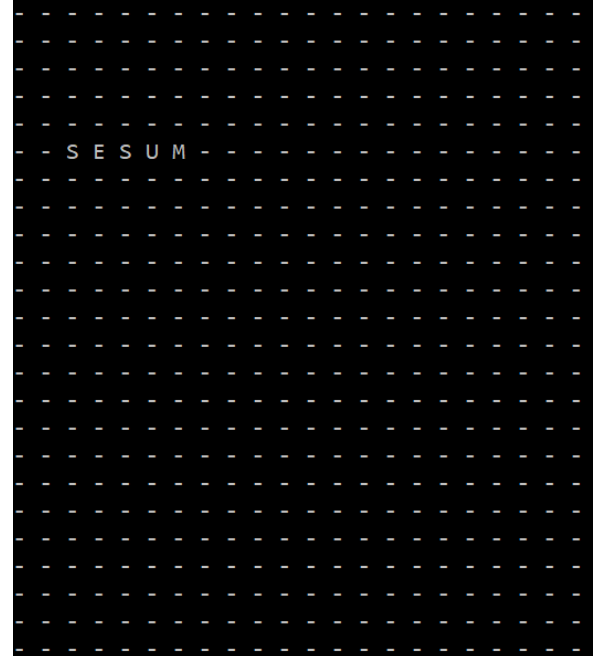
S R E H T A E F

FLEX

S U O I R O L G

37 - IF2211 - Strategi Algoritma

S U O T N E M O M



S E S U M

A 10x10 grid of small black squares on a white background. The squares are arranged in a regular pattern, with 10 squares in each row and 10 squares in each column, totaling 100 squares. The squares are small and evenly spaced, creating a grid-like pattern.

<pre> Sedang mencari BESTOFME - E - - M - - F - - O - - T - - S - - E - - B - Banyak percobaan : 299 Total execution time in milis: 132 </pre>	<pre> Sedang mencari BLOODSWEATANDTEARS - S R A E T D N A T A E W S D O O L B - Banyak percobaan : 437 Total execution time in milis: 128 </pre>
<pre> Sedang mencari BOYINLUV - V U L N I Y O B - Banyak percobaan : 758 Total execution time in milis: 182 </pre>	<pre> Sedang mencari BUTTERFLY - B U T T E R F L Y - Banyak percobaan : 773 Total execution time in milis: 179 </pre>

[illegible]

```
Banyak percobaan : 570
Total execution time in millis: 179
```

```

Sedang mencari DIMPLE
E L P M I D

```

```
Banyak percobaan : 16
Total execution time in millis: 179
```

```
A -  
N -  
D -
```

Banyak percobaan : 99
Total execution time in millis: 197

```
Banyak percobaan : 99
Total execution time in millis: 197
```

```

- - - - E P O D - - - -

```

Banyak percobaan : 291

```
Banyak percobaan : 291
Total execution time in millis: 185
```

[illegible]

```
Sedang mencari GOGO
```

```
G  
O  
G  
O
```

```
Banyak percobaan : 1067  
Total execution time in millis: 190
```

```

- - - - - HOUSE OF CARDS - - - - -

```

Banyak percobaan : 667
Total execution time in millis: 181

```

      U
      D
      E
      E
      N
      I

```

Sedang mencari JUSTONEDAY

J
U
S
T
O
N
E
D
A
Y

Banyak percobaan : 64
Total execution time in milis: 169

Sedang mencari LETMEKNOW

W
O
N
K
E
M
T
E
L

Banyak percobaan : 884
Total execution time in milis: 189

Sedang mencari LIE

L
I
E

Banyak percobaan : 686
Total execution time in milis: 196

Sedang mencari LOST

L
O
S
T

Banyak percobaan : 320
Total execution time in milis: 187


```
Y A D O T T O N
```

Banyak percobaan : 610
Total execution time in millis: 204

```

- - - - - N O I T C E L F E R - - - - -
Banyak percobaan : 888
Total execution time in millis: 206

```

```
R  
U  
N
```

Banyak percobaan : 36
Total execution time in millis: 202

```
Banyak percobaan : 1217
Total execution time in millis: 208
```


Sedang mencari SERENDIPITY

- - - S E R E N D I P I T Y - - -

Banyak percobaan : 1062
Total execution time in milis: 187

Sedang mencari SILVERSPoon

S
I
L
V
E
R
S
P
O
O
N

Banyak percobaan : 150
Total execution time in milis: 160

Sedang mencari SPINEBREAKER

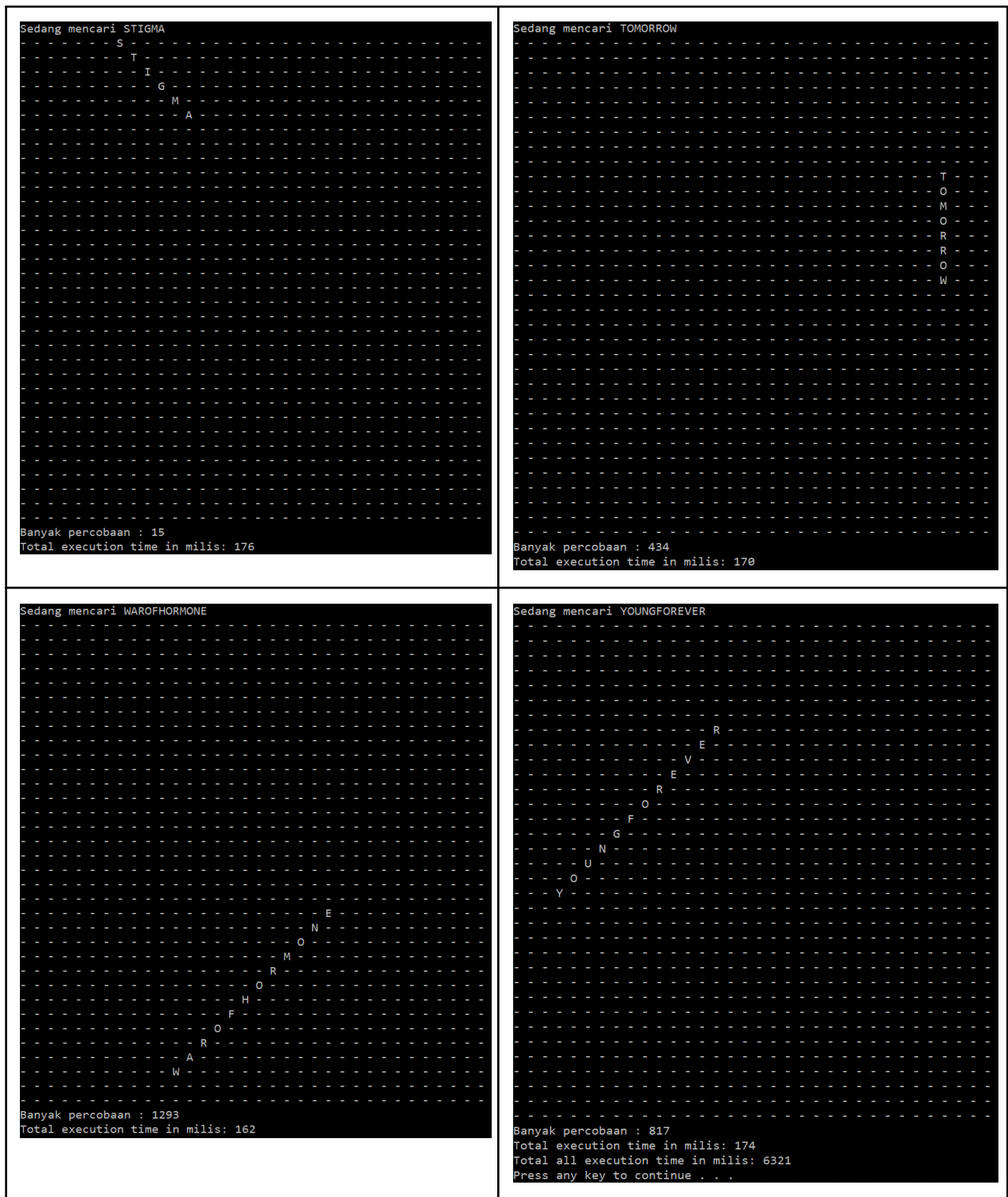
S
P
I
N
E
B
R
E
A
K
E
R

Banyak percobaan : 966
Total execution time in milis: 176

Sedang mencari SPRINGDAY

S
P
R
I
N
G
D
A
Y

Banyak percobaan : 29
Total execution time in milis: 187



Gambar 2.7 Input dan Output Test Case Big 1

```
Welcome to Word Search Solver!
Please insert the puzzle file name :
Big2.txt
Sedang mencari AFRESH
```

```
H S E R F A
```

```
Banyak percobaan : 177
Total execution time in milis: 132
```

```
Sedang mencari AVALANCHING
```

```
G
N
I
H
C
N
A
L
A
V
A
```

```
Banyak percobaan : 959
Total execution time in milis: 193
```

```
Sedang mencari BACKREST
```

```
T
S
E
R
K
C
A
B
```

```
Banyak percobaan : 298
Total execution time in milis: 199
```

```
Sedang mencari BAGHDAD
```

```
D
A
D
H
G
A
B
```

```
Banyak percobaan : 981
Total execution time in milis: 251
```

Sedang mencari CHILLED

- - - - C H I L L E D - - - -

Banyak percobaan : 900

Total execution time in milis: 203

Sedang mencari COCKY

- C -
- O -
- C -
- K -
- Y -

Banyak percobaan : 1105

Total execution time in milis: 154

Sedang mencari CONQUERABLE

C
O
N
Q
U
E
R
A
B
L
E

Banyak percobaan : 645

Total execution time in milis: 202

Sedang mencari CORDS

- C
- O
- R
- D
- S

Banyak percobaan : 1187

Total execution time in milis: 199

```

- - Y T I E D - -
Banyak percobaan : 1236
Total execution time in milis: 171

```

```
Banyak percobaan : 1533
Total execution time in milis: 127
```

```

- - - - - D
- - - - - N
- - - - - I
- - - W -
- N -
W -
O -
D O
D -

```

Banyak percobaan : 1099
Total execution time in millis: 148

```

- - - - - E M P T I N E S S - - - - -
Banyak percobaan : 898
Total execution time in millis: 147

```

Sedang mencari FINDINGS

F
I
N
D
I
N
G
S

Banyak percobaan : 495
Total execution time in milis: 167

Sedang mencari FOUNDER

F
O
U
N
D
E
R

Banyak percobaan : 249
Total execution time in milis: 170

Sedang mencari GREEDILY

G
R
E
E
D
I
L
Y

Banyak percobaan : 501
Total execution time in milis: 180

Sedang mencari GRUFF

F F U R G

Banyak percobaan : 669
Total execution time in milis: 168

Sedang mencari GUARANTEE

E
E
T
N
A
R
A
U
G

Banyak percobaan : 1347
Total execution time in milis: 189

Sedang mencari HYPERBOLE

E
L
O
B
R
E
P
Y
H

Banyak percobaan : 705
Total execution time in milis: 172

Sedang mencari INTERVIEWED

D
E
W
E
I
V
R
E
T
N
I

Banyak percobaan : 742
Total execution time in milis: 178

Sedang mencari LAUGHED

LAUGHED

Banyak percobaan : 611
Total execution time in milis: 175

```

M
I
S
E
R
Y

```

```
Banyak percobaan : 1102
Total execution time in milis: 167
```

```

      N
    U - 
   M - 
  B - 
 I - 
N - 
 G - 
 L - 
Y - 

Banyak percobaan : 536
Total execution time in milis: 191

```

```

- - - - - L A T N E M A N R O - - - - -

```

Banyak percobaan : 243
Total execution time in milis: 166


```
L O S A R A P
```

Banyak percobaan : 409
Total execution time in milis: 190

```

- S -
- Y -
- O -
- B -
- Y -
- A -
- L -
- P -

```

Banyak percobaan : 310
Total execution time in milis: 179

```

- - - - -
- P R E Y E D -
- - - - -

Banyak percobaan : 662
Total execution time in millis: 167

```

```

Banyak percobaan : 835
Total execution time in millis: 176

```

Sedang mencari REAPER

R
E
A
P
E
R

Banyak percobaan : 144
Total execution time in milis: 178

Sedang mencari RENAL

R
E
N
A
L

Banyak percobaan : 35
Total execution time in milis: 178

Sedang mencari REPENTANT

T N A T N E P E R

Banyak percobaan : 1126
Total execution time in milis: 179

Sedang mencari SCRIPTURAL

L
A
R
U
T
P
I
R
C
S

Banyak percobaan : 618
Total execution time in milis: 157

Sedang mencari SHRANK

K
N
A
R
H
S

Banyak percobaan : 271
Total execution time in milis: 182

Sedang mencari SMEARED

D
E
R
A
E
M
S

Banyak percobaan : 1032
Total execution time in milis: 172

Sedang mencari SPECIALTY

SPECIALTY

Banyak percobaan : 918
Total execution time in milis: 164

Sedang mencari SPINSTER

S
P
I
N
S
T
E
R

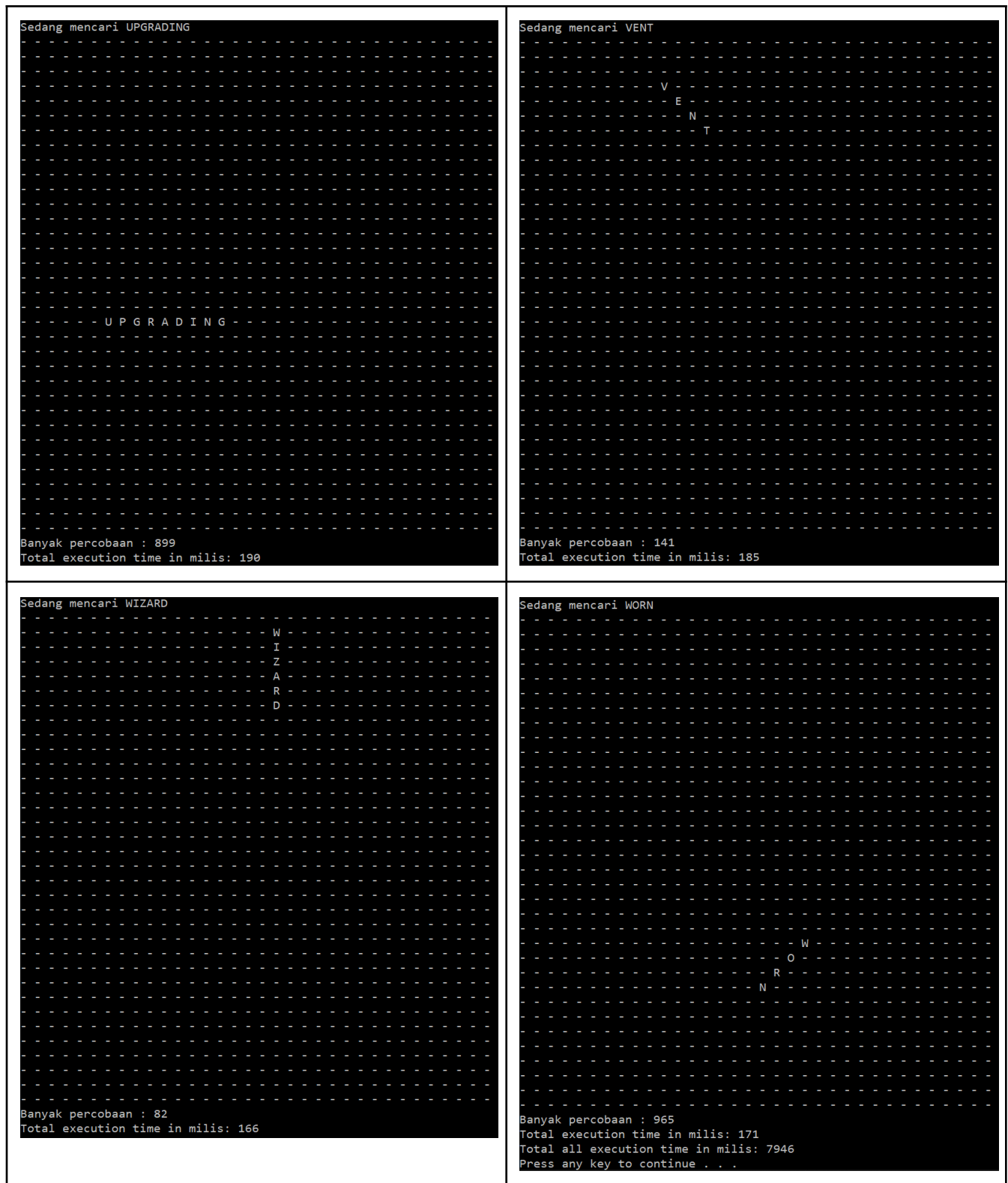
Banyak percobaan : 1154
Total execution time in milis: 188

A 20x20 grid of dots. The word "DONUTS" is spelled out by removing dots in a specific pattern. The letters are formed by the following dot coordinates (row, column) where the dot is missing:

- D**: (10, 15), (11, 15), (12, 15), (13, 15), (14, 15), (15, 15), (16, 15), (17, 15), (18, 15), (19, 15)
- O**: (10, 16), (11, 16), (12, 16), (13, 16), (14, 16), (15, 16), (16, 16), (17, 16), (18, 16), (19, 16)
- N**: (10, 17), (11, 17), (12, 17), (13, 17), (14, 17), (15, 17), (16, 17), (17, 17), (18, 17), (19, 17)
- U**: (10, 18), (11, 18), (12, 18), (13, 18), (14, 18), (15, 18), (16, 18), (17, 18), (18, 18), (19, 18)
- T**: (10, 19), (11, 19), (12, 19), (13, 19), (14, 19), (15, 19), (16, 19), (17, 19), (18, 19), (19, 19)
- S**: (10, 20), (11, 20), (12, 20), (13, 20), (14, 20), (15, 20), (16, 20), (17, 20), (18, 20), (19, 20)

SDUTS

S
S
E
R
G
I
T



Gambar 2.8 Input dan Output Test Case Big 2

Link Drive Kode Program

https://github.com/hcarissa/Tucil1_13520164.git

Checklist

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan (no syntax error)	✓	
2. Program berhasil running	✓	
3. Program dapat membaca file masukan dan menuliskan luaran.	✓	
4. Program berhasil menemukan semua kata di dalam puzzle	✓	

Daftar Referensi

Slide Kuliah IF2211 Strategi Algoritma - Algoritma Brute Force Bagian 1 -

[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-\(2022\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag1.pdf)

Perbandingan Algoritma Brute Force dan Backtracking dalam Permainan Word Search Puzzle

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2016-2017/Makalah2017/Makalah-IF2211-2017-077.pdf>