

Penyelesaian Word Search Puzzle dengan Menggunakan Algoritma Brute Force

Laporan Tugas Kecil 1 IF2211 Strategi Algoritma

Semester II Tahun 2021/2022



Disusun oleh:

Grace Claudia 13520078

I. PENJELASAN ALGORITMA

A. Algoritma Brute Force

Brute Force merupakan salah satu pendekatan penyelesaian masalah yang memiliki ciri-ciri khusus yaitu penyelesaian yang bersifat **straightforward**. Penyelesaian ini dilakukan dengan **sangat sederhana** dengan mengorbankan jumlah langkah penyelesaian masalah. Dalam program, pengaruh penggunaan algoritma Brute Force adalah durasi pengeksekusian program. Dengan pertimbangan ini, brute force kerap kali kita temukan hanya pada **testcase yang kecil**. Program dengan metode Brute Force cenderung memakan waktu yang sangat lama untuk *testcase* yang besar.

B. Algoritma Program secara General

1. Pembacaan input teks yang puzzlenya akan diubah kedalam bentuk matrix NxN bernama **puzzle**, kata yang dicari dikumpulkan dalam sebuah vector bertipe string dengan nama vector **points**
2. Untuk pengubahan puzzle ke matrix, akan dilakukan **iterasi satu-satu terhadap line** yang ada pada file dan berakhir jika line dalam file tersebut **tidak memiliki elemen (“”)** yang berarti akan dilanjutkan dengan proses pemasukan kata yang dicari ke vector
3. Untuk pemasukan kata ke vector, akan **dipisahkan terlebih dahulu puzzle** yang berada di atas text file dengan penggunaan loop while, lalu **sampai end of file, kata-kata yang dicari akan di push** ke vector.
2. Mengiterate kata-kata yang terdapat dalam vector dan satu-satu dicari sesuai urutan kemunculan dalam file dengan algoritma brute force lalu mengembalikan vector berisi struct point yang berisi koordinat letak x dan y huruf pada kata yang ditemukan
3. Mendisplay character yang ditunjuk pada vector berisi point, selain itu diisi “-”

C. Penjelasan langkah-langkah algoritma brute force pada program

Penggunaan algoritma Brute Force terdapat pada fungsi **findKata()** yang menerima parameter berupa matrix dari puzzle dan sebuah string dari vector kata yang akan dicari. Fungsi ini mengembalikan **vector point yang berisi koordinat letak ditemukannya huruf dari kata** yang dicari yang nantinya digunakan untuk mendisplay. Langkah-langkah algoritma brute force adalah sebagai berikut:

1. Mengassign:

- lenkata dengan panjang dari kata yang dicari
- idxMin untuk pencarian koordinat x dan y yaitu 0
- idxJMax yaitu untuk pencarian maksimal koordinat y batasnya sesuai banyaknya kolom pada matrix puzzle
- idxIMax yaitu untuk pencarian maksimal koordinat x batasnya sesuai banyaknya baris pada matrix puzzle

- Counter untuk menghitung banyaknya perbandingan yang telah dilakukan untuk mencari kata tersebut
2. Menggunakan ukuran baris dan kolom dari puzzle untuk looping pengecekan kata yang dicari
 3. Mengiterasi satu per satu character dalam puzzle dengan mengecek tempat dimana **letak huruf pertama** dari kata yang dicari
 4. Jika ditemukan tempat dari huruf pertama tersebut akan dilakukan **pengecekan apakah panjang dari kata pada titik tersebut memenuhi** untuk dicari ukurannya dalam matrix agar tidak terjadi index out of bounds. Pengecekan ini membutuhkan value idxMin, idxJMax, dan idxImax.
 5. Selanjutnya apabila panjang kata memenuhi ukuran matrix maka akan dilakukan **pengecekan dari huruf kedua sampai terakhir** dengan urutan atas, bawah, kiri, kanan, diagonal ke atas kiri, diagonal ke atas kanan, diagonal ke bawah kiri, diagonal ke bawah kanan (tentunya pengecekan dilakukan hanya jika memenuhi syarat batas ukuran matrix, jika tidak maka akan skip pengecekan).
 6. Prosedur pengecekan:
 1. Jika sudah memenuhi kriteria maka akan dilanjutkan dengan **membuat vector points yang kosong**
 2. Vector points ini berguna jika index ke-x dari sebuah string sama dengan elemen matrix[i][j] yang diiterate, maka koordinat i j akan **di push ke vector points**
 3. Apabila **banyak elemen dari vector points sama** dengan panjang kata, maka dipastikan koordinat-koordinat tersebut adalah jawabannya, jika tidak maka akan **dilanjutkan pengecekan** ke arah selanjutnya.
 7. Jika pengecekan tersebut benar dan ditemukan koordinat masing-masing huruf yang sesuai maka akan di store ke suatu struct point.

II. SOURCE PROGRAM

Struct Point, Function newMatrix dan newWord

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  struct Point
5  // struct ini berisi koordinat x dan y yang menampung letak huruf yang telah ditemukan
6  {
7      int x, y;
8  };
9
10 vector< vector<char> > newMatrix(const char* filename) {
11     // fungsi ini digunakan untuk membuat matrix sesuai input file user
12     vector< vector<char> > mat;
13     string line;
14     ifstream file;
15     file.open(filename);
16     if (file.is_open()) {
17         getline(file, line);
18         while(line != "") {
19             vector<char> temp;
20             for (int i = 0; i < line.length(); i+=2) {
21                 temp.push_back(line[i]);
22             }
23             mat.push_back(temp);
24             getline(file, line);
25         }
26         file.close();
27     } else {
28         cout << "failed to open the file" << endl;
29     }
30
31     return mat;
32 }
33
34 vector<string> newWord(const char* filename) {
35     // fungsi ini digunakan untuk membuat vector dari kata kata yang akan dicari
36     vector<string> words;
37     string line;
38     ifstream file;
39     file.open(filename);
40     if (file.is_open()) {
41         vector<char> temp;
42         getline(file, line);
43         while(line != "") {
44             getline(file, line);
45         }
46
47         while(getline(file, line)) {
48             words.push_back(line);
49         }
50         file.close();
51     } else {
52         cout << "failed to open the file" << endl;
53     }
54
55     return words;
56 }
```

Procedure displayMatrix dan displayAnsMatrix

```
1 void displayMatrix(vector< vector<char> > mat){
2     // fungsi ini digunakan pada saat testing agar dapat mengecek inputan matrix yang dibaca
    dari file yang diinput
3     for (int i=0; i< mat.size(); i++){
4         for (int j=0; j < mat[i].size(); j++){
5             cout << mat[i][j] <<" ";
6         }
7         cout << "\n";
8     }
9 }
10
11 void displayAnsMatrix(vector< vector<char> > mat, vector<Point> p){
12     // fungsi ini digunakan untuk mendisplay matrix jawaban dari penyelesaian suatu kata
13     vector< vector<char> > mAns;
14     // membuat matrix berisikan "-"
15     for (int i=0; i< mat.size(); i++){
16         vector<char> temp;
17         for (int j=0; j < mat[i].size(); j++){
18             temp.push_back('-');
19         }
20         mAns.push_back(temp);
21     }
22     // memunculkan huruf yang ditemukan
23     for (int i=0; i<p.size();i++){
24         mAns[p[i].x][p[i].y] = mat[p[i].x][p[i].y];
25     }
26
27     displayMatrix(mAns);
28
29 }
```

Function FindKata

```
1 vector<Point> findKata(vector< vector<char> > m,string kata){
2     // fungsi ini digunakan untuk melakukan pencarian koordinat letak suatu kata ditemukan
3     int lenKata = kata.length();
4     int idxMin = 0;
5     int idxJMax = m[0].size();
6     int idxIMax = m.size();
7     int counter = 0;
8     for (int i=0; i<m.size(); i++){
9         for (int j=0; j<m[0].size();j++){
10             counter++;
11             // mengecek apakah elemen pertama dari kata yang dicari sesuai dengan elemen matrix
12             if (m[i][j] == kata[0]){
13                 // mengecek lurus ke atas
14                 if (i-(lenKata-1)>=idxMin){
15                     vector<Point> Points;
16                     for(int x = 0; x<lenKata; x++) {
17                         if (m[i-x][j] != kata[x]) {
18                             break;
19                         } else {
20                             Point P;
21                             P.x = i-x;
22                             P.y = j;
23                             Points.push_back(P);
24                         }
25                     }
26                     if (Points.size() == lenKata) {
27                         cout <<"Number of Comparisons: "<<counter<<"\n";
28                         return Points;
29                     }
30                 }
31                 // mengecek lurus ke bawah
32                 if (i+(lenKata-1)<=idxIMax){
33                     vector<Point> Points;
34                     for(int x = 0; x<lenKata; x++) {
35                         if (m[i+x][j] != kata[x]) {
36                             break;
37                         } else {
38                             Point P;
39                             P.x = i+x;
40                             P.y = j;
41                             Points.push_back(P);
42                         }
43                     }
44                     if (Points.size() == lenKata) {
45                         cout <<"Number of Comparisons: "<<counter<<"\n";
46                         return Points;
47                     }
48                 }
49             }
50         }
51     }
```

```

1      // mengecek lurus ke kiri
2      if (j-(lenKata-1)>=idxMin){
3          vector<Point> Points;
4          for(int x = 0; x<lenKata; x++) {
5              if (m[i][j-x] != kata[x]) {
6                  break;
7              } else {
8                  Point P;
9                  P.x = i;
10                 P.y = j-x;
11                 Points.push_back(P);
12             }
13         }
14         if (Points.size() == lenKata) {
15             cout <<"Number of Comparisons: "<<counter<<"\n";
16             return Points;
17         }
18     }
19     // mengecek lurus ke kanan
20     if (j+(lenKata-1)<=idxJMax){
21         vector<Point> Points;
22         for(int x = 0; x<lenKata; x++) {
23             if (m[i][j+x] != kata[x]) {
24                 break;
25             } else {
26                 Point P;
27                 P.x = i;
28                 P.y = j+x;
29                 Points.push_back(P);
30             }
31         }
32         if (Points.size() == lenKata) {
33             cout <<"Number of Comparisons: "<<counter<<"\n";
34             return Points;
35         }
36     }
37     // mengecek diagonal ke arah atas kiri
38     if (j-(lenKata-1)>=idxMin && i-(lenKata-1)>=idxMin){
39         vector<Point> Points;
40         for(int x = 0; x<lenKata; x++) {
41             if (m[i-x][j-x] != kata[x]) {
42                 break;
43             } else {
44                 Point P;
45                 P.x = i-x;
46                 P.y = j-x;
47                 Points.push_back(P);
48             }
49         }
50         if (Points.size() == lenKata) {
51             cout <<"Number of Comparisons: "<<counter<<"\n";
52             return Points;
53         }
54     }

```

```

1      // mengecek diagonal ke arah atas kanan
2      if (j+(lenKata-1)<=idxJMax && i-(lenKata-1)>=idxMin){
3          vector<Point> Points;
4          for(int x = 0; x<lenKata; x++) {
5              if (m[i-x][j+x] != kata[x]) {
6                  break;
7              } else {
8                  Point P;
9                  P.x = i-x;
10                 P.y = j+x;
11                 Points.push_back(P);
12             }
13         }
14         if (Points.size() == lenKata) {
15             cout <<"Number of Comparisons: "<<counter<<"\n";
16             return Points;
17         }
18     }
19     // mengecek diagonal ke arah bawah kiri
20     if (j-(lenKata-1)>=idxMin && i+(lenKata-1)<=idxIMax){
21         vector<Point> Points;
22         for(int x = 0; x<lenKata; x++) {
23             if (m[i+x][j-x] != kata[x]) {
24                 break;
25             } else {
26                 Point P;
27                 P.x = i+x;
28                 P.y = j-x;
29                 Points.push_back(P);
30             }
31         }
32         if (Points.size() == lenKata) {
33             cout <<"Number of Comparisons: "<<counter<<"\n";
34             return Points;
35         }
36     }
37     // mengecek diagonal ke arah bawah kanan
38     if (j+(lenKata-1)<=idxJMax && i+(lenKata-1)<=idxIMax){
39         vector<Point> Points;
40         for(int x = 0; x<lenKata; x++) {
41             if (m[i+x][j+x] != kata[x]) {
42                 break;
43             } else {
44                 Point P;
45                 P.x = i+x;
46                 P.y = j+x;
47                 Points.push_back(P);
48             }
49         }
50         if (Points.size() == lenKata) {
51             cout <<"Number of Comparisons: "<<counter<<"\n";
52             return Points;
53         }
54     }
55 }
56 }
57 }
58 }

```


Function Main

```

1 int main () {
2
3     // variable file untuk menampung input user
4     string file;
5
6     cout <<"8   8   8                               8eeee8                               " <<endl;
7     cout <<"8   8 8 eeeee eeeee eeeee   8           eeee eeeee eeeee eeee e   e " <<endl;
8     cout <<"8e  8 8 8 88 8   8 8 8   8 8eeee8 8   8   8 8 8   8 8 8 8 8 " <<endl;
9     cout <<"88  8 8 8   8 8eee8e 8e 8           88 8eee 8eee8 8eee8e 8e   8eee8 " <<endl;
10    cout <<"88  8 8 8   8 88   8 88 8   e   88 88   88  8 88   8 88   88 8 " <<endl;
11    cout <<"88ee8ee8 8eee8 88   8 88ee8   8eee88 88ee 88  8 88   8 88e8 88 8 \" <<endl;
12
13    // melakukan input nama file
14    cout << "File Name Input (please include .txt when input) : "; cin >> file;
15    cout << "\n";
16
17    // penambahan directory terhadap input file
18    string filename = "../test/" + file;
19
20    // pembuatan matrix bernama PUZZLE sebagai matrix yang akan dicari
21    vector< vector<char> > puzzle = newMatrix(filename.c_str());
22
23    // pembuatan vector untuk kata-kata yang akan dicari
24    vector<string> words = newWord(filename.c_str());
25
26    // proses pencarian kata
27    // assign waktu dimulai pencarian
28    clock_t start, end;
29    start = clock();
30    // mengiterate vector bernama WORDS agar mendapat vector POINTS, letak koordinat ditemu
31    kannya kata
32    for (int i=0;i<words.size();i++){
33        cout << words[i] << " ";
34        cout << "\n";
35        vector<Point> points = findKata(puzzle, words[i]);
36        displayAnsMatrix(puzzle, points);
37        cout << "\n";
38    }
39    // waktu selesai pencarian
40    end = clock();
41    double time_taken = double(end - start) / double(CLOCKS_PER_SEC);
42    cout << "Time taken by program is : " << fixed
43        << time_taken << setprecision(5);
44    cout << " secs " << endl;
45    return 0;
46 }

```

III. SCREENSHOT INPUT DAN OUTPUT

Awal Program Berjalan:

```
 8  8  8                               8eeee8
 8  8  8 eeeee eeeee eeeee           8      eeee eeeee eeeee eeee e  e
8e 8  8 8 88 8  8 8  8               8eeee 8  8  8 8  8  8 8 8  8
88 8  8 8 8  8 8eee8e 8e  8           88 8eee 8eee8 8eee8e 8e  8eee8
88 8  8 8 8  8 88  8 88  8           e  88 88  88  8 88  8 88  88  8
88ee8ee8 8eee8 88  8 88ee8           8eee88 88ee 88  8 88  8 88e8 88  8

File Name Input (please include .txt when input) : █
```

Small case (2)

Small Case - 1 (10x10, 10 words) → 0,084 secs

Input: small3.txt

```
test > ≡ small3.txt
 1  A N N A H M A P S J
 2  G R F P A U R O R A
 3  C T S E N L I C A S
 4  M E R I D A E A P M
 5  T T R D S N L H U I
 6  I E B E L L E O N N
 7  H S Y A O T A N Z E
 8  S N O W W H I T E O
 9  A G O S O D X A L H
10  I S R S T U H S U H
11
12  ANNA
13  ARIEL
14  AURORA
15  BELLE
16  JASMINE
17  MERIDA
18  MULAN
19  POCAHONTAS
20  RAPUNZEL
21  SNOWWHITE
```

total comparison: $1 + 7 + 15 + 53 + 10 + 31 + 6 + 8 + 19 + 71 = 221$

```
ANNA
Number of Comparisons: 1
```

```
ARIEL
Number of Comparisons: 7
```

AURORA
Number of Comparisons: 15

BELLE
Number of Comparisons: 53

```
JASMINE
Number of Comparisons: 10
```

[illegible]

```
MERIDA
Number of Comparisons: 31
```

M E R I D A

MULAN
Number of Comparisons: 6

M
U
L
A
N

POCAHONTAS
Number of Comparisons: 8

-	-	-	-	-	-	P	-
-	-	-	-	-	-	O	-
-	-	-	-	-	-	C	-
-	-	-	-	-	-	A	-
-	-	-	-	-	-	H	-
-	-	-	-	-	-	O	-
-	-	-	-	-	-	N	-
-	-	-	-	-	-	T	-
-	-	-	-	-	-	A	-
-	-	-	-	-	-	S	-

RAPUNZEL
Number of Comparisons: 19

RAPUNZEL

SNOWWHITE
Number of Comparisons: 71

S N O W W H I T E

Time taken by program is : 0.084000 sec

Small Case - 2 (10x10, 17 words) → 0,147 secs

Input: small1.txt

```
test > ≡ small1.txt
 1  P E A C H A P P L E
 2  F A B K I W I S H T
 3  R P A P P L U M I A
 4  G R N E C I N M D N
 5  R U A A T M U F I G
 6  A N N R N E T A L E
 7  P E A O R A N G E R
 8  E A P R I C O T M I
 9  W A T E R M E L O N
10  P E R S I M M O N E
11
12  APPLE
13  APRICOT
14  BANANA
15  FIG
16  GRAPE
17  KIWI
18  LEMON
19  LIME
20  MELON
21  ORANGE
22  PEACH
23  PEAR
24  PERSIMMON
25  PLUM
26  PRUNE
27  TANGERINE
28  WATERMELON
```

total comparison: $6 + 72 + 13 + 48 + 31 + 14 + 59 + 26 + 86 + 64 + 1 + 24 + 91 + 25 + 22 + 20 + 81 = 683$

[illegible]

```

WATERMELON
Number of Comparisons: 81
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
W A T E R M E L O N
- - - - -

Time taken by program is : 0.147000 sec

```

Medium case (3)

Medium Case - 1 (15x15, 26 words) → 0,714 secs

Input: medium1.txt

```
test >  medium1.txt
 1  C M N O R B K G C N T C F N T
 2  L H R T I D N I A H R T S R I
 3  L I I R C A N V O I E A O T E
 4  O F D M T O R T O I S E T W T
 5  T L E E P I D E E R R R T R L
 6  H Y D S W A L L O W N O E A T
 7  C H I C K E N L S R A I R A H
 8  W O L F L S C Z K E P A N D A
 9  O L H E N O Q A E W A L R U S
10  C A I A W N N U T E E L N E E
11  T O I Z M G H K I P L R R D A
12  O L R R A S M E T D E P S D D
13  P L V R T R T S S P I D E R O
14  U D O A E H D E Y G T S L E U
15  S O C T E S P A R R O W I E S
16
17  BIRD
18  CAT
19  CHEETAH
20  CHICKEN
21  CHIMPANZEE
22  COW
23  DEER
24  FLY
25  HAMSTER
26  KANGAROO
27  LIZARD
28  OCTOPUS
29  OTTER
30  OWL
31  PANDA
32  PIG
33  RAT
34  SEAL
35  SNAIL
36  SPARROW
37  SPIDER
38  SQUID
39  SWALLOW
40  TORTOISE
41  WALRUS
42  WOLF
```

Total comparison: $6 + 112 + 9 + 91 + 1 + 112 + 67 + 47 + 123 + 114 + 122 + 121 + 43 + 43 + 116 + 177 + 26 + 99 + 111 + 216 + 189 + 111 + 79 + 50 + 130 + 106 = 2421$

BIRD Number of Comparisons: 6	CHICKEN Number of Comparisons: 91	DEER Number of Comparisons: 67	KANGAROO Number of Comparisons: 114
- B - - I - - R - - D -	CHICKEN - A - - N - - G - - R - - O -	D E E R - A - - N - - G - - R - - O -	A K N G R O
CAT Number of Comparisons: 112	CHIMPANZEE Number of Comparisons: 1	FLY Number of Comparisons: 47	LIZARD Number of Comparisons: 122
- C - - A - - T -	H I M P A N Z E E	F L Y	L I Z A R D
CHEETAH Number of Comparisons: 9	COW Number of Comparisons: 112	HAMSTER Number of Comparisons: 123	OCTOPUS Number of Comparisons: 121
- C - - H - - E - - E - - T - - A - H	C O W	H A M S T E R	O C T O P U S
OCTOPUS Number of Comparisons: 121	PANDA Number of Comparisons: 116	SPARRROW Number of Comparisons: 216	SMALLOW Number of Comparisons: 79
- O - - C - - T - O P U S	P A N D A	S P A R R O W	S M A L L O W
OTTER Number of Comparisons: 43	PIG Number of Comparisons: 177	SNAIL Number of Comparisons: 111	SPIDER Number of Comparisons: 109
- O - - T - - T - - E - - R -	P I G	S N A I L	S P I D E R
OWL Number of Comparisons: 43	RAT Number of Comparisons: 26	SQUID Number of Comparisons: 111	WALRUS Number of Comparisons: 130
- O - - W - - L -	R A T	S Q U I D	W A L R U S
			Time taken by program is : 0.714000 sec PS D:\IF\Semester4\stima\tuc11> []

Medium Case - 2 (15x15, 18 words) → 0,511 secs

Input: medium2.txt

```
test > ≡ medium2.txt
 1  T N N F S T R A W B E R R Y C
 2  Y C O R O H T T E P Y A Y E O
 3  B D O E E N Y C H E R R Y F O
 4  I H M N F E T D L A E E R B K
 5  Y M O C H A S B M C N O R L I
 6  W P N H W W A E E H E R F U E
 7  M A A V N N E U S N A A U E S
 8  I B L A A H M E E C P S D B A
 9  S M A N G O H O O O O P G E N
10  U A A I U N N C A F L B E R D
11  M O A L I T O M M F I E E R C
12  F I T L C N H E I E T R B Y R
13  I S S A U A T L L E A R E V E
14  B U T T E R P E C A N Y H T A
15  P R A L I N E P E C A N W V M
16
17  BANANA
18  BLUEBERRY
19  BUTTERPECAN
20  CHERRY
21  COCONUT
22  COFFEE
23  COOKIESANDCREAM
24  FRENCHVANILLA
25  FUDGE
26  MANGO
27  MOCHA
28  NEAPOLITAN
29  PEACH
30  PRALINEPECAN
31  RASPBERRY
32  REESES
33  STRAWBERRY
34  WALNUT
```


Total comparison: $68 + 59 + 196 + 38 + 115 + 115 + 15 + 4 + 88 + 122 + 62 + 71 + 25 + 211 + 87 + 19 + 5 + 76 = 1376$

[illegible]

```

NEAPOLITAN
Number of Comparisons: 71
-----
N
E
A
P
O
L
I
T
I
T
A
N
-----

PEACH
Number of Comparisons: 25
-----
P
E
A
C
H
-----

PRALINEPECAN
Number of Comparisons: 211
-----
P
R
A
L
I
N
E
P
E
C
A
N
-----

RASPBERRY
Number of Comparisons: 87
-----
R
A
S
P
B
E
R
R
Y
-----

RESEES
Number of Comparisons: 19
-----
R
E
S
E
S
-----

STRAWBERRY
Number of Comparisons: 5
-----
S
T
R
A
W
B
E
R
R
Y
-----

WALNUT
Number of Comparisons: 76
-----
W
A
L
N
U
T
-----

Time taken by program is : 0.511000

```

Medium Case - 3 (15x15, 18 words) → 0,471 secs

Input : medium3.txt

```
test > ≡ medium3.txt
 1  E S P O C B L O O D H O U N D
 2  B U P O L T I F C D O S R M A
 3  U O B I I I F S O H D I C B L
 4  L C O O T N N H R S U R O E L
 5  L T O M X Z T E G I T I C H H
 6  D P A B B E Y E I A C S K O A
 7  O H O G A O R P R D E H E I S
 8  G K N M S N T D S A W S R D A
 9  B O R D E R C O L L I E S O A
10  W S M C N R E G W M D T P B P
11  R E R N J S A O S A A T A E S
12  A E O E I H I N L T W E N R O
13  O E H G I L L E I I I R I M M
14  E F S H I H T Z U A H T E A T
15  D A C H S H U N D N N N L N E
16
17  AIREDALE
18  BASENJI
19  BLOODHOUND
20  BORDERCOLLIE
21  BOXER
22  BULLDOG
23  COCKERSPANIEL
24  CORGI
25  DACHSHUND
26  DALMATIAN
27  DOBERMAN
28  IRISHSETTER
29  LHASAAPSO
30  POINTER
31  POMERANIAN
32  SHEEPDOG
33  SHIHTZU
34  SPITZ
```

Total comparison: $90 + 80 + 6 + 121 + 33 + 16 + 43 + 24 + 211 + 100 + 119 + 42 + 60 + 3 + 77 + 38 + 198 + 2 = 1263$

[illegible]

Large case (3)

Large Case - 1 (20x20, 28 words) → 1,2230 secs

Input: large1.txt

```
test > ≡ large1.txt
 1 S B S U Y M E R S S R A U T U N T I I F
 2 E O R T L S E R O M E N G N I D N I F N
 3 L O S O I X C V E R M A O C S T A E I A
 4 B S H T O N H O A T E D T O E O T S L V
 5 I M S M E H P U O R L N E C A H A K O I
 6 D U G A F E W H T C B H T O E M A C N L
 7 E C M R N S L I N K Y D O G A T E S E L
 8 R B S L T M I K E W A Z O W S K I W C U
 9 C U T I E F N T L C U O O D C D S N R S
10 N Z H N Y C P E W L D N Y L E N E E U N
11 I Z Y B A E T W D D E R S O A L L H S D
12 E L A R G S B B I N O S U N L L R I H B
13 H I S H O E D N H D A T S A I I S R R T
14 T G T L A T O D G H I M W U E B B U A V
15 I H I I T S S N E I L N O S R I C D E T
16 H T R N A A I Y Y A O T T D E E E I T N
17 T Y E U S D E Y O S A I M A E N T D S A
18 L E R I N M O N S T E R S I N C E A V E
19 T A S I S T S T A V D C I E R A U T M H
20 N R F N S H E R I F F W O O D Y R L A U
21
22 BOO
23 BRAVE
24 BRUCE
25 BUZZLIGHTYEAR
26 CARS
27 COCO
28 CRUSH
29 DUG
30 EDNAMODE
31 FINDINGDORY
32 FINDINGNEMO
33 INSIDEOUT
34 MARLIN
35 MATER
36 MIKEWAZOWSKI
37 MONSTERSINC
38 RATATOUILLE
39 REMY
40 REX
41 RUSSELL
42 SHERIFFWOODY
43 SLINKYDOG
44 SULLIVAN
45 THEGOODDINOSAUR
46 THEINCREDIBLES
47 TOYSTORY
48 UP
49 WALLE
```

Total comparison: $2 + 111 + 240 + 142 + 186 + 54 + 159 + 101 + 180 + 383 + 39 + 100 + 84 + 379 + 146 + 346 + 388 + 8 + 8 + 295 + 385 + 126 + 180 + 77 + 261 + 350 + 88 + 273 = 5091$

<div>ROO Number of Comparisons: 2 - B - O - O</div>	<div>BRAVE Number of Comparisons: 248 - R - U - C - E</div>	<div>CARS Number of Comparisons: 186 - A - R - S - C - D - C - O</div>	<div>CRUSH Number of Comparisons: 159 - C - R - U - N - H - O - D - E</div>	<div>EDWARD Number of Comparisons: 188 - E - D - N - A - H - O - D - E</div>
<div>BRAVE Number of Comparisons: 111 - E - V - A - R - B</div>	<div>BZZLIGHTYEAR Number of Comparisons: 342 - B - U - Z - L - T - G - H - T - Y - E - A - R</div>	<div>COCO Number of Comparisons: 54 - C - O - C - O</div>	<div>OUG Number of Comparisons: 181 - O - U - G - Y - B - O - G - N - I - D - N - F</div>	<div>FINDINGDORY Number of Comparisons: 383 - Y - B - O - G - N - I - D - N - F</div>
<div>FINDINGREMO Number of Comparisons: 39 - O - M - E - N - G - N - I - D - N - I - F</div>	<div>MARLIN Number of Comparisons: 84 - M - A - R - L - I - N</div>	<div>MIKHAZOWSKI Number of Comparisons: 346 - M - I - K - H - A - Z - O - W - S - K - I</div>	<div>RATATOUILLE Number of Comparisons: 388 - R - A - T - A - T - O - U - I - L - L - E - T - U - O - T - A - T - R</div>	<div>REX Number of Comparisons: 8 - R - X</div>
<div>INSIDEOUT Number of Comparisons: 180 - I - N - S - I - D - E - O - U - T</div>	<div>PATER Number of Comparisons: 379 - P - A - T - E - R</div>	<div>MONSTERSINC Number of Comparisons: 346 - M - O - N - S - T - E - R - S - I - N - C</div>	<div>REMY Number of Comparisons: 8 - R - E - M - Y</div>	<div>RUSSELL Number of Comparisons: 295 - R - U - S - S - E - L</div>

Number of Comparisons: 385

[illegible]

SULLIVAN

Number of Comparisons: 180

```
THEGOODINDOOR
Number of Comparisons: 77

- - - - -
- - - - - T
- - - - - H
- - - - - E
- - - - - G
- - - - - O
- - - - - D
- - - - - D
- - - - - I
- - - - - N
- - - - - O
- - - - - S
- - - - - A
- - - - - D
- - - - - R
```

THE INCREDIBLES

Number of Comparisons: 261

[illegible]

UP

Number of Comparisons: 88

Time taken by program is : 1.223000 sec

Large Case - 2 (20x20, 31 words) → 1,46 secs

Input : large2.txt

```
test > large2.txt
1  Y T N T P Y E T I H W W O N S L D Y S O
2  P G N E A S M L E N F E A E I S L A H M
3  E N N E R R V S N M U H R R F D T O D H
4  E I E L C O A T H D K F D E I N I Y O E
5  L L L Y E I Y E T E A A E A O E F A O V
6  S R I T E P F E R W T H N H C E L S H H
7  E A S H E U D E E E U B A M B I S P N A
8  S D T L E W H I L Y H C S U C O C Y I K
9  U N R T I S I N I A O O B E A K P E B C
10 O H A S H E U N N P M N L E O M E D O U
11 M O E A R S T O N L A T R O U I L A R D
12 Y J H R T E S T M I T O H R T I N B K C
13 E O F E V I L Q U E E N G T L O E A C M
14 K I O W M Y P P A H I T N P S L A S U E
15 C A N A S T A S I A T N H T L S K H D G
16 I E E O Y Y I R T R S O N E S G E F Y O
17 M H E M T A D P I A I A B I P A E U S O
18 B S U E T U A N N L O H F M M O H L I R
19 H M Q A E C L U E O U E N U E E O H A C
20 N M O W G L I P N W T D N M M A S H D S
21
22  ALICE
23  ANASTASIA
24  ARIEL
25  BAMBI
26  BASHFUL
27  BELLE
28  CAPTAINHOOK
29  DAISYDUCK
30  DEWEY
31  EEYORE
32  EVILQUEEN
33  GRUMPY
34  HAPPY
35  HUEY
36  JOHNDARLING
37  MALEFICENT
38  MICKEYMOUSE
39  MINNIEMOUSE
40  MOWGLI
41  MUFASA
42  PLUTO
43  POCAHONTAS
44  QUEENOFHEARTS
45  ROBINHOOD
46  SCROOGE MCDUC
47  K
48  SHEREKHAN
49  SLEEPY
50  SNOWWHITE
51  WENDY
52  WINNIETHEPOOH
```

Output

Total comparison: $98 + 282 + 33 + 132 + 238 + 238 + 366 + 399 + 70 + 129 + 244 + 253 + 270 + 147 + 222 + 191 + 321 + 355 + 382 + 395 + 368 + 190 + 363 + 219 + 400 + 71 + 166 + 101 + 15 + 390 + 146$
= 7194

<div><div>Alice</div><div>Number of Comparisons: 98</div><div><div><div>A</div><div>L</div><div>C</div><div>E</div></div></div></div>	<div><div>Ariel</div><div>Number of Comparisons: 33</div><div><div><div>A</div><div>R</div><div>E</div><div>L</div></div></div></div>	<div><div>Bashful</div><div>Number of Comparisons: 238</div><div><div><div>B</div><div>A</div><div>S</div><div>H</div><div>F</div><div>U</div><div>L</div></div></div></div>	<div><div>Captain Jack</div><div>Number of Comparisons: 366</div><div><div><div>K</div><div>O</div><div>H</div><div>N</div><div>A</div><div>T</div><div>P</div><div>A</div><div>C</div></div></div></div>	<div><div>Deity</div><div>Number of Comparisons: 70</div><div><div><div>D</div><div>E</div><div>M</div><div>E</div><div>V</div></div></div></div>
<div><div>Anastasia</div><div>Number of Comparisons: 282</div><div><div><div>A</div><div>N</div><div>A</div><div>S</div><div>T</div><div>A</div><div>S</div><div>T</div><div>A</div></div></div></div>	<div><div>Bambi</div><div>Number of Comparisons: 132</div><div><div><div>B</div><div>A</div><div>M</div><div>B</div><div>I</div></div></div></div>	<div><div>Belle</div><div>Number of Comparisons: 238</div><div><div><div>B</div><div>E</div><div>L</div><div>L</div><div>E</div></div></div></div>	<div><div>Daisy Duck</div><div>Number of Comparisons: 399</div><div><div><div>K</div><div>C</div><div>U</div><div>V</div><div>S</div><div>T</div><div>A</div><div>D</div></div></div></div>	<div><div>Eeyore</div><div>Number of Comparisons: 129</div><div><div><div>E</div><div>R</div><div>D</div><div>Y</div><div>E</div><div>E</div></div></div></div>
<div><div>Evil Queen</div><div>Number of Comparisons: 244</div><div><div><div>E</div><div>V</div><div>I</div><div>L</div><div>Q</div><div>U</div><div>E</div><div>E</div><div>N</div></div></div></div>	<div><div>Happy</div><div>Number of Comparisons: 270</div><div><div><div>Y</div><div>P</div><div>P</div><div>A</div><div>H</div></div></div></div>	<div><div>Johndarling</div><div>Number of Comparisons: 222</div><div><div><div>G</div><div>N</div><div>I</div><div>L</div><div>R</div><div>A</div><div>O</div><div>N</div><div>H</div><div>O</div><div>J</div><div>C</div><div>L</div></div></div></div>	<div><div>Mickey Mouse</div><div>Number of Comparisons: 321</div><div><div><div>E</div><div>S</div><div>U</div><div>O</div><div>M</div><div>Y</div><div>E</div><div>K</div><div>C</div><div>I</div><div>M</div></div></div></div>	<div><div>Mogli</div><div>Number of Comparisons: 382</div><div><div><div>M</div><div>O</div><div>G</div><div>L</div><div>I</div></div></div></div>
<div><div>Grumpy</div><div>Number of Comparisons: 253</div><div><div><div>Y</div><div>P</div><div>H</div><div>U</div><div>R</div><div>G</div></div></div></div>	<div><div>Huey</div><div>Number of Comparisons: 147</div><div><div><div>Y</div><div>E</div><div>U</div><div>H</div></div></div></div>	<div><div>Maleficent</div><div>Number of Comparisons: 191</div><div><div><div>T</div><div>N</div><div>E</div><div>C</div><div>I</div><div>F</div><div>E</div><div>L</div><div>A</div><div>H</div></div></div></div>	<div><div>Himmlerhouse</div><div>Number of Comparisons: 355</div><div><div><div>E</div><div>S</div><div>U</div><div>O</div><div>H</div><div>E</div><div>I</div><div>N</div><div>A</div><div>I</div><div>M</div></div></div></div>	<div><div>Mufasa</div><div>Number of Comparisons: 395</div><div><div><div>A</div><div>S</div><div>A</div><div>F</div><div>U</div><div>H</div></div></div></div>

Large Case - 3 (20x20, 29 words) → 1,359 secs

Input

test > ≡ large3.txt

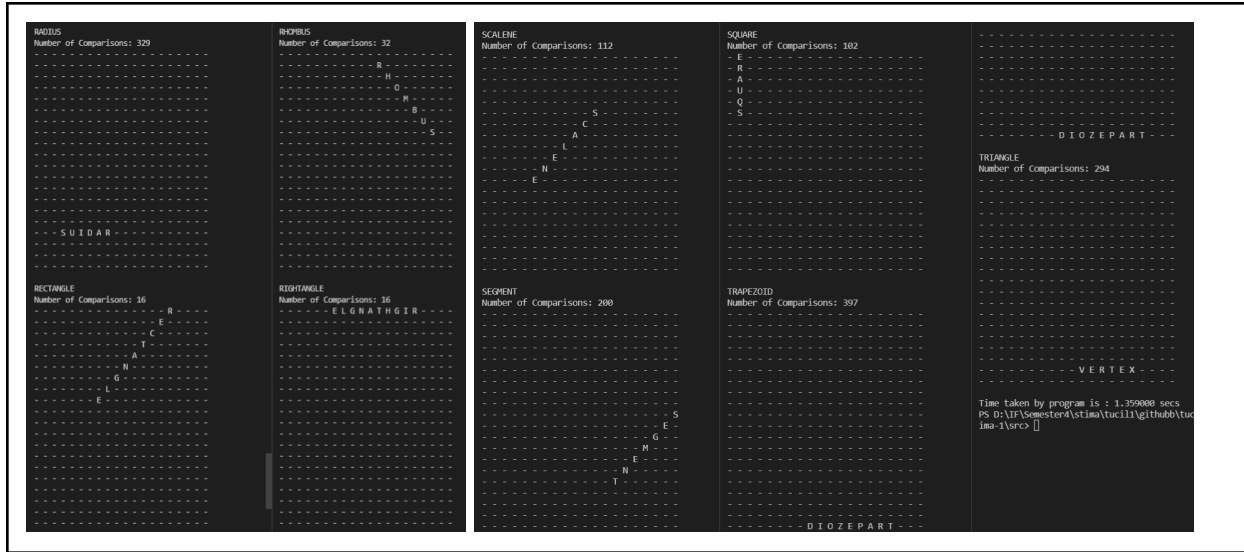
```
1  L E N I P O E L G N A T H G I R I Y A P
2  P R E E T E R O A Y D R A A E T O N N O
3  M A E T I R A D A U I E H C I N T I A H
4  P U N E S N R R M P B I T O H R I F T T
5  E Q I E O N W O L N Y A U I M E H L O O
6  R S U T S O S H A A N S A O E B O C C E
7  P N G W C S T C E G C C W I A C U T E T
8  E P N T E Q U I L A T E R A L L A S L E
9  N O I G L O S E L M U E S A W G T N A A
10 D I T C E S O E T E N T T W O G C H R S
11 I N C F S U N R D E T E C N D A A A E I
12 C T E C N E R E F M U C R I C Y B G T A
13 U N S H L W E D I A M E T E R T M N A Y
14 L O R H O B T U S E V R U C D E S O L C
15 A G E N N G E L G N A I R T N T R G I E
16 R A T O R L O E L C R I C T V D I A R N
17 E T N S U I D A R N E V S H L R O X D S
18 F N I T S H V N O I A A B V V L D E A D
19 T E U I S O R G H S V E R T E X S H U M
20 I P T R N E N C D I O Z E P A R T H Q M
```

```
21
22  ACUTE
23  ARC
24  CHORD
25  CIRCLE
26  CIRCUMFERENCE
27  CLOSED CURVE
28  DIAMETER
29  EQUILATERAL
30  HEXAGON
31  INTERSECTING
32  ISOSCELES
33  LINE
34  OBTUSE
35  OCTAGON
36  OVAL
37  PENTAGON
38  PERPENDICULAR
39  POINT
40  QUADRILATERAL
41  RADIUS
42  RECTANGLE
43  RHOMBUS
44  RIGHTANGLE
45  SCALENE
46  SEGMENT
47  SQUARE
48  TRAPEZOID
49  TRIANGLE
50  VERTEX
```

Output

$$135 + 174 + 128 + 313 + 235 + 280 + 248 + 145 + 378 + 343 + 45 + 98 + 265 + 100 + 366 + 382 + 61 + 142 + 399 + 329 + 16 + 32 + 16 + 112 + 200 + 182 + 397 + 294 + 371 = 6186$$

<div>ACUTE</div> <div>Number of Comparisons: 135</div>	<div>CHORD</div> <div>Number of Comparisons: 128</div>	<div>CIRCUMFERENCE</div> <div>Number of Comparisons: 235</div>	<div>DIAMETER</div> <div>Number of Comparisons: 248</div>	<div>HEXAGON</div> <div>Number of Comparisons: 378</div>
<div>- A C U T E -</div>	<div>- D - B - O - H - C</div>	<div>- E C N E R E F M U C R I C -</div>	<div>- D I A M E T E R -</div>	<div>- N - O - G - A - X - F - H</div>
<div>ARC</div> <div>Number of Comparisons: 174</div>	<div>CIRCLE</div> <div>Number of Comparisons: 313</div>	<div>CLOSEDCURVE</div> <div>Number of Comparisons: 280</div>	<div>EQUILATERAL</div> <div>Number of Comparisons: 145</div>	<div>INTERSECTING</div> <div>Number of Comparisons: 343</div>
<div>- C - N - A</div>	<div>- E L C R I C -</div>	<div>- E V R U C D E S O L C -</div>	<div>- E Q U I L A T E R A L -</div>	<div>- G - N - T - T - C - E - S - R - E - T - N - I</div>
<div>ISOSCELES</div> <div>Number of Comparisons: 45</div>	<div>OBTUSE</div> <div>Number of Comparisons: 265</div>	<div>OCTAGIN</div> <div>Number of Comparisons: 100</div>	<div>PENTAGON</div> <div>Number of Comparisons: 382</div>	<div>POINT</div> <div>Number of Comparisons: 142</div>
<div>- I - S - O - S - C - E - L - E - S</div>	<div>- O B T U S E -</div>	<div>- O - C - T - A - G - O - N</div>	<div>- N - O - G - A - T - N - E - P</div>	<div>- P - O - I - N - T</div>
<div>LINE</div> <div>Number of Comparisons: 98</div>	<div>OCTAGON</div> <div>Number of Comparisons: 100</div>	<div>ONAL</div> <div>Number of Comparisons: 366</div>	<div>PERPENDICULAR</div> <div>Number of Comparisons: 61</div>	<div>QUADRILATERAL</div> <div>Number of Comparisons: 399</div>
<div>- E - N - I - L</div>	<div>- O - C - T - A - G - O - N</div>	<div>- A - B - P - E - N - D - I - C - U - L - A - B - V - O</div>	<div>- P - E - P - E - N - D - I - C - U - L - A - B - V - O</div>	<div>- L - R - E - T - A - L - I - R - D - A - A - A - U</div>



IV. ALAMAT DRIVE KODE PROGRAM

[Click here](#)

V. CHECK LIST

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