

**IMPLEMENTASI ALGORITMA *BRUTE FORCE* DI DALAM  
PERMAINAN WORD SEARCH GENERATOR**

**Laporan Tugas Kecil I**

**Disusun sebagai syarat kelulusan mata kuliah  
IF2211/Strategi Algoritma**

**Oleh**

**HANA FATHIYAH**

**NIM : 13520047**



**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO & INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
Januari 2022**

## I. Algoritma *Brute-Force*

### a. Definisi

Algoritma *Brute-Force* merupakan algoritma dengan pendekatan yang lempang (*straightforward*) untuk memecahkan suatu persoalan. Algoritma *Brute-Force* biasanya didasarkan pada pernyataan yang terdapat di dalam persoalan (*problem statement*) dan definisi atau konsep yang dilibatkan di dalamnya. Dalam algoritma *Brute-Force*, persoalan dipecahkan dengan sangat sederhana, langsung, jelas caranya, dan langsung untuk dilakukan.

Salah satu implementasi dari algoritma *Brute-Force* ini adalah pencocokan string (*string matching*). Implementasi ini lebih diperluas lagi melalui permainan *Word Search Puzzle*. Dalam hal ini, pencocokan string dilakukan dengan delapan arah mata angin (*west, northwest, north, northeast, east, southeast, south, southwest*) hingga ditemukan kata yang dicari.

### b. Garis Besar Program

Pengerjaan *Word Search Puzzle* ini dimulai dari pembacaan file dengan format yang diberikan pada spesifikasi tugas besar, yakni suatu matriks karakter yang berisi alphabet acak dan kata-kata yang hendak dicari dengan dibatasi satu buah baris kosong. Matriks alphabet acak ini merupakan matriks 2 x 2 yang pada setiap baris terdapat jarak antarhuruf berupa spasi. Pada program yang saya buat, matriks alphabet acak ini dinamakan *game\_matrix* dengan pengalokasian ukuran awal sebesar 255 x 255. Selanjutnya, matriks alphabet ini berfungsi untuk menyimpan karakter-karakter yang terdapat di dalam file dengan mengabaikan tanda spasi (" "). Pembacaan kata demi kata dilakukan dengan cara yang sama, yaitu sebuah matriks *word\_list* yang berisi kata demi kata dengan indeks baris menyatakan urutan kata tersebut dan indeks kolom menyatakan huruf demi huruf yang terdapat di setiap kata.

Setelah pembacaan *file* berhasil dilakukan, tahap selanjutnya adalah pencocokan *string* yang dilakukan tahap demi tahap. Pencocokan *string* ini dilakukan pada setiap kata berdasarkan urutannya. Tahap selanjutnya adalah pewarnaan *string*

apabila ditemukan kata yang sesuai dengan sistem pewarnaan yang berbeda pada setiap katanya.

### c. Struktur Program

Keseluruhan program yang dibuat disimpan di dalam *folder src*. Di dalam *folder src* tersebut terdapat beberapa *folder* dan *file* dengan utilitas tertentu.

#### 1. *File type.h*

*File type.h* ini merupakan *file header* yang berisi *struct* Position (menandakan posisi terkini pada matriks) dan Size (menandakan baris dan kolom matriks), serta *enum* Direction (menyimpan 8 arah mata angin) untuk mempermudah penulisan program

#### 2. *Folder finder*

*Folder finder* berisi *file move.c* dan *file move.h*. *File move.h* merupakan *file header*; sedangkan *file move.c* merupakan implementasinya. Pengoperasian kedua *file* ini dibantu juga oleh *file type.h* yang berisi *struct* dan *enum* untuk mempermudah penulisan program. Fungsi *move* yang terdapat pada *move.c* mengembalikan *struct Position* berupa posisi penunjukan huruf setelah berpindah satu langkah berdasarkan delapan arah mata angin. Penggunaan *switch-case* dipilih untuk mempermudah penulisan program. Selanjutnya, pada *file move.c* ini pun terdapat suatu boolean (dalam hal ini dituliskan dengan bentuk *integer* yang memiliki keluaran 0 atau 1) untuk menandakan apakah posisi yang dicapai ketika bergerak satu langkah tersebut merupakan posisi yang valid dari segi angka dan ukuran matriks yang tersisa.

#### 3. *File main.c*

Program utama pada tugas ini ditulis di dalam *file main.c*. Di dalamnya, terdapat beberapa fungsi dan prosedur yang esensial untuk melakukan pencarian kata dalam matriks. Di dalam fungsi utama (*main*) terdapat perintah pembacaan *file*. Elemen-elemen dalam *file* tersebut dialokasikan ke dalam dua matriks berbeda. Matriks yang berisi karakter acak dimasukkan ke dalam matriks *game\_matrix*, sedangkan kata-kata yang akan dicari dimasukkan ke dalam *word\_list*.

Selain itu, terdapat variabel *display* berupa matriks 2x2 untuk membantu pewarnaan file. Matriks ini diinisialisasi dengan nilai -1 yang kemudian diubah apabila suatu kata ditemukan.

Pencarian kata dilakukan secara bertahap kata demi kata. Pengecekan kata dilakukan melalui prosedur *checkWord* yang memiliki parameter berupa kata (*word\_list[i]*), matriks alphabet acak (*game\_matrix*), dan variabel *matrix* untuk menandakan posisi. Dilakukan traversal hingga semua kata berhasil ditemukan.

Pada prosedur *checkWord* ini terdapat traversal yang digunakan untuk menginisialisasi matriks hasil (*result*) menjadi nilai *false*. Variabel *strlen* digunakan untuk menyimpan panjang kata. Kemudian dilakukan traversal kembali untuk melakukan perpindahan posisi. Pengecekan 8 arah mata angin dilakukan menggunakan traversal variabel *k* dengan fungsi boolean *isPositionMatch* yang apabila bernilai *true* melakukan pemanggilan prosedur *setTableFlag*.

Prosedur *isPositionMatch* memiliki parameter berupa kata (disimpan di dalam variabel *str*), posisi (disimpan di dalam variabel *pos*), arah mata angin (disimpan di dalam variabel *k*), submatriks (disimpan di dalam variabel *s*), dan matriks alphabet (disimpan di dalam variabel *matrix*). Di dalamnya terdapat pengecekan *string* secara satu persatu dengan keluaran berupa *boolean true* atau *false*.

Prosedur *setTableFlag* memiliki parameter berupa panjang kata (*length*), *Position* (*pos*) dalam bentuk *struct*, dan *direction* (*dir*) dalam bentuk *enum*. Posisi terkini disimpan di dalam variabel *currentPos*. Setelah itu, dilakukan *looping* berupa pemanggilan fungsi *move* selama *i* lebih kecil dari panjang kata. matriks hasil (*result*) diubah menjadi *true*, kemudian posisi terkini dipindahkan menggunakan fungsi *move* yang terdapat di dalam *folder finder*.

#### 4. Folder pcolor

Di dalam *folder* ini terdapat dua buah *file* untuk mempermudah pewarnaan. Digunakan 6 buah warna dengan pola pewarnaan menggunakan urutan kata dimodulus 6.

#### d. Implementasi *Brute-Force* dalam Program

Algoritma *Brute-Force* menjadi alat untuk pengimplementasian program ini. *Brute-force* digunakan di dalam program utama, yakni *main.c*. Dilakukan *brute-force* berupa pengecekan kata demi kata menggunakan prosedur *checkWord* dengan cara pengecekan 8 arah mata angin, apakah dapat berpindah atau tidak. Apabila dapat berpindah, dilakukan pemindahan posisi sepanjang banyak huruf dalam kata tersebut.

Jika ternyata diperoleh hasil *true* di sepanjang prosesnya, maka kata berhasil ditemukan.

## II. *Source Code* (dalam Bahasa C)

### a. *File type.h*

```
#pragma once

typedef struct Position {
    int row;
    int col;
} Position;

typedef struct Size {
    int rows;
    int cols;
} Size;

typedef enum Direction {
    NORTH,
    NORTHWEST,
    WEST,
    SOUTHWEST,
    SOUTH,
    SOUTHEAST,
    EAST,
    NORTHEAST
} Direction;
```

### b. *Folder finder*

#### 1. *move.h*

```
#pragma once

#include "../type.h"

Position move(Position currentPos, Direction arah);
```

```
int isPositionValid(Position pos, Size matrixSize);
```

## 2. *move.c*

```
#include "move.h"

Position move(Position currentPos, Direction arah)
{
    Position posisiBaru;

    switch (arah)
    {
        case WEST:
            posisiBaru.row = currentPos.row;
            posisiBaru.col = currentPos.col - 1;
            break;
        case NORTHWEST:
            posisiBaru.row = currentPos.row - 1;
            posisiBaru.col = currentPos.col - 1;
            break;
        case NORTH:
            posisiBaru.row = currentPos.row - 1;
            posisiBaru.col = currentPos.col;
            break;
        case NORTHEAST:
            posisiBaru.row = currentPos.row - 1;
            posisiBaru.col = currentPos.col + 1;
            break;
        case EAST:
            posisiBaru.row = currentPos.row;
            posisiBaru.col = currentPos.col + 1;
            break;
        case SOUTHEAST:
            posisiBaru.row = currentPos.row + 1;
            posisiBaru.col = currentPos.col + 1;
            break;
        case SOUTH:
            posisiBaru.row = currentPos.row + 1;
            posisiBaru.col = currentPos.col;
```

```

        break;
    case SOUTHWEST:
        posisiBaru.row = currentPos.row + 1;
        posisiBaru.col = currentPos.col - 1;
        break;
    default:
        posisiBaru = currentPos;
        break;
    }

    return posisiBaru;
}

int isPositionValid(Position pos, Size matrixSize)
{
    return pos.row >= 0 && pos.col >= 0
           && matrixSize.rows > pos.row && matrixSize.cols >
pos.col;
}

```

c. *File main.c*

```

// originally made by Hana Fathiyah 13520047

#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
#include "finder/move.h"
#include "pcolor/pcolor.h"

bool result[255][255];

bool isPositionMatch(char* str, Position pos, Direction dir,
Size s, char matrix[255][255]){
    Position currentPos = pos;
    int length = 0;
    int stringLength = strlen(str) - 1;

```

```

        while(isPositionValid(currentPos, s)) {
            if(str[length] ==
matrix[currentPos.row][currentPos.col]){
                length++;
            }else{
                return false;
            }

            if(length == stringLength){
                return true;
            }

            currentPos = move(currentPos, dir);
        }

        return false;
    }

void setTableFlag(int length, Position pos, Direction dir){
    Position currentPos = pos;
    int i = 0;

    while(i < length) {
        result[currentPos.row][currentPos.col] = true;
        i++;
        currentPos = move(currentPos, dir);
    }
}

void checkWord(char* str, char matrix[255][255], Size s) {
    for(int i = 0; i < s.rows; i++){
        for(int j = 0; j < s.cols; j++){
            result[i][j] = false;
        }
    }

    int strlength = strlen(str);

```



```

        for(int i = 0; i < s.rows; i++){
            for(int j = 0; j < s.cols; j++){
                Position pos;
                pos.row = i;
                pos.col = j;

                for(int k = 0; k < 8; k++){
                    if(isPositionMatch(str, pos, k, s, matrix)) {
                        setTableFlag(strlength-1, pos, k);
                    }
                }
            }
        }
    };

int main () {
    printf("WORD SEARCH GAME ENGINE\n");
    printf("originally made by Hana Fathiyah\n\n");
    char file_names[255];
    printf("insert file's name: ");
    scanf("%s",file_names);

    FILE *game_file = fopen(file_names, "r");

    if(!game_file){
        printf("The file that you want to open is not exist.\n");
        return false;
    }

    char word_list[255][255];
    char game_matrix[255][255];
    char line[255];
    bool next_line_of_matrix = true;
    int game_row = 0;
    int game_column = 0;
    int word_count = 0;

    while(next_line_of_matrix) {

```

```

fscanf(game_file,"%[^\\n]\\n",line);

if(line[1] == ' ') {
    game_column = 0;
    for(int i = 0; i < (int) strlen(line); i++) {
        if(line[i] != ' ') {
            game_matrix[game_row][game_column] = line[i];
            game_column++;
        }
    }

    game_row++;
} else {
    next_line_of_matrix = false;
    strcpy(word_list[word_count], line);
    word_count++;
}
}

while(!feof(game_file)) {
    fscanf(game_file,"%[^\\n]\\n",line);
    strcpy(word_list[word_count],line);
    word_count++;
}

Size matrix;
matrix.cols=game_column - 1;
matrix.rows=game_row;
fclose(game_file);

int display[255][255];
for(int i = 0; i < matrix.rows; i++) {
    for(int j = 0; j < matrix.cols; j++) {
        display[i][j] = -1;
    }
}

int game_coloring = 0;
for(int i = 0; i < word_count; i++){
    checkWord(word_list[i], game_matrix, matrix);
}

```

```

        for(int i = 0; i < matrix.rows; i++){
            for(int j=0; j < matrix.cols; j++){
                if(result[i][j]){
                    display[i][j] = game_coloring;
                    //printf("%c", game_matrix[i][j]);
                } else {
                    //printf("-");
                }
                if (j < matrix.cols - 1) {
                    //printf(" ");
                }
            }
            //printf("\n");
        }
        //printf("\n\n");
        game_coloring++;
    }

```

```

for(int i = 0; i < matrix.rows; i++) {
    for(int j = 0; j < matrix.cols; j++) {
        if (display[i][j] >= 0) {
            if (display[i][j] % 6 == 0) {
                print_red(game_matrix[i][j]);
            } else if (display[i][j] % 6 == 1) {
                print_green(game_matrix[i][j]);
            } else if (display[i][j] % 6 == 2) {
                print_yellow(game_matrix[i][j]);
            } else if (display[i][j] % 6 == 3) {
                print_blue(game_matrix[i][j]);
            } else if (display[i][j] % 6 == 4) {
                print_magenta(game_matrix[i][j]);
            } else if (display[i][j] % 6 == 5) {
                print_cyan(game_matrix[i][j]);
            }
        } else {
            printf("%c", game_matrix[i][j]);
        }
        if (j < matrix.cols - 1) {
            printf(" ");
        }
    }
}

```

```

    }
}
printf("\n");
}
}

```

d. *Folder pcolor*

1. *pcolor.h*

```

#define NORMAL "\x1B[0m"
#define RED "\x1B[31m"
#define GREEN "\x1B[32m"
#define YELLOW "\x1B[33m"
#define BLUE "\x1B[34m"
#define MAGENTA "\x1B[35m"
#define CYAN "\x1B[36m"

void print_red(char c);
void print_green(char c);
void print_yellow(char c);
void print_blue(char c);
void print_magenta(char c);
void print_cyan(char c);

```

2. *pcolor.c*

```

#include <stdio.h>
#include "pcolor.h"

void print_red(char c) {
    printf("%s%c", RED, c);
    printf("%s", NORMAL);
}

void print_green(char c) {
    printf("%s%c", GREEN, c);
    printf("%s", NORMAL);
}

```

```
void print_yellow(char c) {
    printf("%s%c", YELLOW, c);
    printf("%s", NORMAL);
}

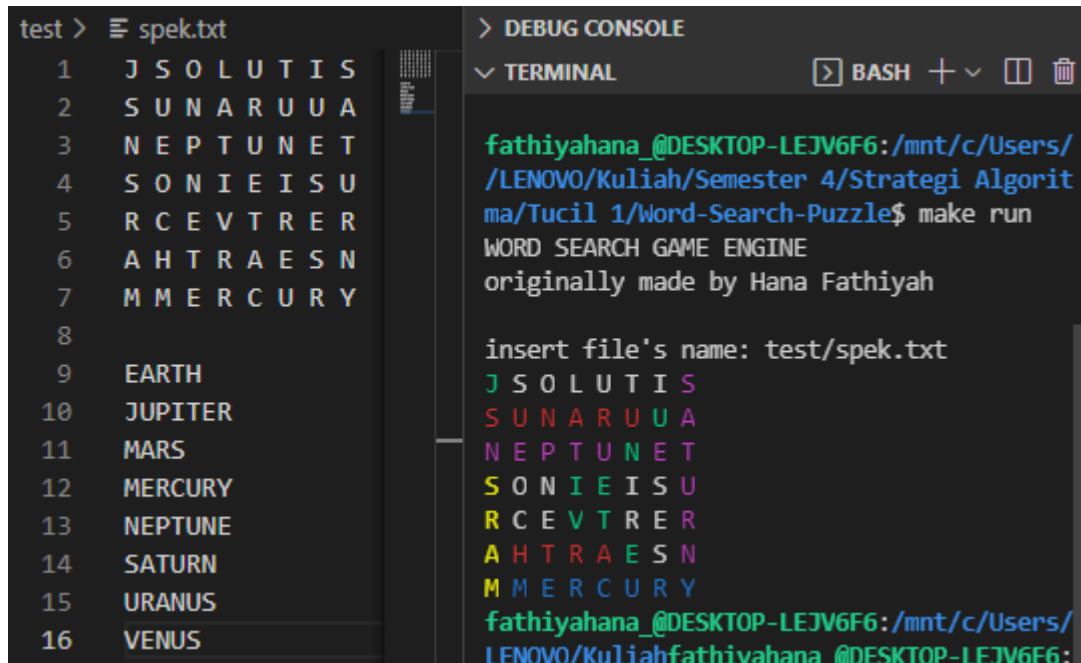
void print_blue(char c) {
    printf("%s%c", BLUE, c);
    printf("%s", NORMAL);
}

void print_magenta(char c) {
    printf("%s%c", MAGENTA, c);
    printf("%s", NORMAL);
}

void print_cyan(char c) {
    printf("%s%c", MAGENTA, c);
    printf("%s", NORMAL);
}
```

### III. Screenshot Program

- a. *input: spek.txt* (berdasarkan spesifikasi tugas kecil)



The screenshot shows a program interface with a list of planets on the left and a debug console on the right. The list of planets is as follows:

test >	spek.txt
1	J S O L U T I S
2	S U N A R U U A
3	N E P T U N E T
4	S O N I E I S U
5	R C E V T R E R
6	A H T R A E S N
7	M M E R C U R Y
8	
9	E A R T H
10	J U P I T E R
11	M A R S
12	M E R C U R Y
13	N E P T U N E
14	S A T U R N
15	U R A N U S
16	V E N U S

The debug console output is as follows:

```
> DEBUG CONSOLE
v TERMINAL
fathiyahana_@DESKTOP-LEJV6F6:/mnt/c/Users/
/LENOVO/Kuliah/Semester 4/Strategi Algorit
ma/Tucil 1/Word-Search-Puzzle$ make run
WORD SEARCH GAME ENGINE
originally made by Hana Fathiyah

insert file's name: test/spek.txt
J S O L U T I S
S U N A R U U A
N E P T U N E T
S O N I E I S U
R C E V T R E R
A H T R A E S N
M M E R C U R Y
fathiyahana_@DESKTOP-LEJV6F6:/mnt/c/Users/
LENOVO/Kuliahfathiyahana_@DESKTOP-LEJV6F6:
```

Waktu eksekusi program:

real 0m19.966s

user 0m0.000s

sys 0m0.033s

Jumlah total perbandingan huruf:

```
M M E R C U R Y
Jumlah perbandingan: 3820
```

b. *input: small1.txt* (14 x 12)

```
test > small1.txt
1 J A N U A R Y V F J N L
2 I X I Y A D N U S Y S K
3 C H O Y Y A D Y N A O E
4 Q C T O D A Y U M T M N
5 U R U T P G C G A U E R
6 S A G F E N U J Y E D L
7 Q M H H S F G Z D S A L
8 U G T S U G U A N D Y Y
9 I Y L L U R D P P A J A
10 J J S I L V A I D Y G D
11 O U P J R Z N I J V V N
12 D I L R D P R P P A U O
13 V F Q Y P F A G S L S M
14 P Q X S C R E B O T C O
15
16 ANYDAY
17 APRIL
18 AUGUST
19 FRIDAY
20 JANUARY
21 JULY
22 JUNE
23 MARCH
24 MAY
25 MONDAY
26 OCTOBER
27 SOMEDAY
28 SUNDAY
29 TODAY
30 TUESDAY

> DEBUG CONSOLE
v TERMINAL BASH + - [ ] [X]

fathiyahana_@DESKTOP-LEJV6F6:/mnt/c/Users/LENOVO/Kuliah/Semester 4/Strategi Algoritma/Tucil 1/Word-Search-Puzzle$ make run
WORD SEARCH GAME ENGINE
originally made by Hana Fathiyah

insert file's name: test/small1.txt
J A N U A R Y V F J N L
I X I Y A D N U S Y S K
C H O Y Y A D Y N A O E
Q C T O D A Y U M T M N
U R U T P G C G A U E R
S A G F E N U J Y E D L
Q M H H S F G Z D S A L
U G T S U G U A N D Y Y
I Y L L U R D P P A J A
J J S I L V A I D Y G D
O U P J R Z N I J V V N
D I L R D P R P P A U O
V F Q Y P F A G S L S M
P Q X S C R E B O T C O

fathiyahana_@DESKTOP-LEJV6F6:/mnt/c/Users/LENOVO/Kuliah/Semester 4/Strategi Algoritma/Tucil 1/Word-Search-Puzzle$
fathiyahana_@DESKTOP-LEJV6F6:/mnt/c/Users/LENOVO/Kuliah/Semester 4/Strategi Algoritma/Tucil 1/Word-Search-Puzzle$
fathiyahana_@DESKTOP-LEJV6F6:/mnt/c/Users/LENOVO/Kuliah/Semester 4/Strategi Algoritma/Tucil 1/Word-Search-Puzzle$
```

Waktu eksekusi program:

real 0m8.018s

user 0m0.002s

sys 0m0.044s

Jumlah total perbandingan huruf:

```
P Q X S C R E B O T C O
Jumlah perbandingan: 21137
fathiyahana_@DESKTOP-LEJV6F6:/mnt/c/Users/LENOVO/Kuliah/Semester 4/Strategi Algoritma/Tucil 1/Word-Search-Puzzle$
```

c. input: *small2.txt* (15 x 15)

```
test > small2.txt
1  M B F L E X I T A R I A N E E
2  G J E N O I T A N I M I L E C
3  J L S R W V Y U U H S F F B E
4  V F N I J S T V Z L K H R N V
5  O M T W T E D R E H H B W E P
6  I S I S U H I P Y J J V G Z E
7  U T A U C L Z Z S C M E O W X
8  U F S D I U Q I L E T K V Z A
9  B A L W Z S F K C A O T W T E
10 J Y U I H G S X R S Z M K J X
11 A G V T V S W I H D E I C L X
12 S G R B D K A E T U N R R Z N
13 Z O V C M N R D D S E P L Q S
14 G I R E X K G X F Z S T O W J
15 X A E N O Z M N A G E V N Z A
16
17  ATKINS
18  DASH
19  ELIMINATION
20  FAST
21  FLEXITARIAN
22  KOSHER
23  LIQUID
24  VEGAN
25  VEGETARIAN
26  ZONE

> DEBUG CONSOLE
v TERMINAL
fathiyahana_@DESKTOP-LEJV6F6:/mnt/c/Users/LENOVO/Kuliah/Semester 4/Strategi Algoritma/Tucil 1/Word-Search-Puzzle$ make run
WORD SEARCH GAME ENGINE
originally made by Hana Fathiyah

insert file's name: test/small2.txt
M B F L E X I T A R I A N E E
G J E N O I T A N I M I L E C
J L S R W V Y U U H S F F B E
V F N I J S T V Z L K H R N V
O M T W T E D R E H H B W E P
I S I S U H I P Y J J V G Z E
U T A U C L Z Z S C M E O W X
U F S D I U Q I L E T K V Z A
B A L W Z S F K C A O T W T E
J Y U I H G S X R S Z M K J X
A G V T V S W I H D E I C L X
S G R B D K A E T U N R R Z N
Z O V C M N R D D S E P L Q S
G I R E X K G X F Z S T O W J
X A E N O Z M N A G E V N Z A
fathiyahana_@DESKTOP-LEJV6F6:/mnt/c/Users/LENOVO/rs/LEfathiyahana_@DESKTOP-LEJV6F6:/mnt/c/Useellfathiyfa
fathiyahana_@DESKTOP-LEJV6F6:/mnt/c/Useirs/LENOVO/rs/rs/LENOVO/Kuliah/Semester 4/Strategi Algoritma/Tucil 1/Word-Search-Puzzle$
```

Waktu eksekusi program:

real 0m12.986s

user 0m0.010s

sys 0m0.072s

Jumlah total perbandingan huruf:

```
X A E N O Z M N A G E V N Z A
Jumlah perbandingan: 18786
```



d. input: *small3.txt* (18 x 16)

```
test > small3.txt
1  D M M V H Q M O X R F V E L B G
2  V I N H Q S D M T A A C O B S P
3  C X S I Q B R E A D P S S J E C
4  C A K P A S R E X I M T S V J S
5  S W R B L H S F R Y I N H J V W
6  N Q M R Y A X F G U F A J N Y C
7  I D W I O Y Y A C S K S H F E F
8  F I O M L U X S H A M S H Y A T
9  F I B U Y K I F W S S I X K S F
10 U A C P G B L J E O J O R D T T
11 M P J O C H O W B O L R N V G G
12 I W L Y O Q N W O M L C B K I R
13 N H X O E K S U Q K B R O C W Y
14 P A O H A G I G T V K U A R L V
15 Q Z I L Y V G E G S R K R G F X
16 Y Z R K I I E Z S E E I U U U H
17 P C Y D S K O S E M Q Y F Y E S
18 V Q S Q C T J N T L J Y R J A M
19
20 BISCUITS
21 BREAD
22 COOKIES
23 CROISSANTS
24 DISPLAYS
25 DOUGHNUTS
26 EGGS
27 LOAVES
28 MILK
29 MIXERS
30 MUFFINS
31 SUGAR
32 YEAST

> DEBUG CONSOLE
> TERMINAL

fathiyahana_@DESKTOP-LEJVF6: /mnt/c/Use
e run
WORD SEARCH GAME ENGINE
originally made by Hana Fathiyah

insert file's name: test/small3.txt
D M M V H Q M O X R F V E L B G
V I N H Q S D M T A A C O B S P
C X S I Q B R E A D P S S J E C
C A K P A S R E X I M T S V J S
S W R B L H S F R Y I N H J V W
N Q M R Y A X F G U F A J N Y C
I D W I O Y Y A C S K S H F E F
F I O M L U X S H A M S H Y A T
F I B U Y K I F W S S I X K S F
U A C P G B L J E O J O R D T T
M P J O C H O W B O L R N V G G
I W L Y O Q N W O M L C B K I R
N H X O E K S U Q K B R O C W Y
P A O H A G I G T V K U A R L V
Q Z I L Y V G E G S R K R G F X
Y Z R K I I E Z S E E I U U U H
P C Y D S K O S E M Q Y F Y E S
V Q S Q C T J N T L J Y R J A M

fathiyahana_@DESKTOP-LEJVF6: /mnt/c/Use
rs/LENOVO/Kuliah/Semester 4/Strategi Al
goritma/Tucil 1/Word-Search-Puzzle$
fathiyahana_@DESKTOP-LEJVF6: /mnt/c/Use
rs/LENOVO/Kuliah/Sfathiyahana_@DESKTOP-
LEJVF6: /mnt/c/Useeee-Search-Puzzle$
fathiyahana_@DESKTOP-LEJVF6: /mnt/c/Use
rs/LENOVO/Kuliah/Semester 4/Strategi
fathiyahana_@DESKTOP-LEJVF6: /mnt/c/Use
rs/LENOVO/Kuliah/Semester 4/Strategi Al
```

Waktu eksekusi program:

real 0m6.507s

user 0m0.005s

sys 0m0.001s

Jumlah total perbandingan huruf:

```
V Q S Q C T J N T L J Y R J A M
Jumlah perbandingan: 31154
```

e. input: *medium1.txt* (20 x 18)

The screenshot shows a word search game engine interface. On the left, a 20x18 grid of letters is displayed, with some letters highlighted in different colors. On the right, a list of words found is shown, including: ABOLISH, AWAKING, BIRDSONG, DIABOLIC, ELASTIC, ELOQUENCE, EXEMPLARY, FLICKER, IMPURE, LICHEE, PICKER, and PROTECTION. The interface also includes a terminal window showing the command 'make run' and the output 'WORD SEARCH GAME ENGINE originally made by Hana Fathiyah'.

Waktu eksekusi program:

real 0m6.446s

user 0m0.009s

sys 0m0.010s

Jumlah total perbandingan huruf:

LLBXJVAJDPEPPETSSSK  
Jumlah perbandingan: 60655

f. input: *medium2.txt* (21 x 21)

```
test > medium2.txt
1 QDCBBVFWDERCASSAMGEKR
2 CBEAPKUITBXROTINOMNMJ
3 MHTPDEUJGZBGNIWARDKRG
4 OIQOQZVLOIDLYVPKUOCU
5 RPP LUMAAYJJMRTKLVLPE
6 AWR LBSJDLQCKBISCDHLCU
7 LVCULOQFI FALASZEHQEA O
8 IJZTEJUHVIBAQBIZDXD TO
9 SNLENWRHDQLRVNFHLLDJE
10 TRMRQWDEBAAATXYKDFEOBV
11 IVPFKUIZSCWCRFXEMTWAC
12 CXNLNLFBDPCTEUKXTEOGSG
13 XFFEIVQENTPPBDTANGBKX
14 QDACDCJMJKQSOJSLITJEH
15 ZNUBUQBYXHLZADHUXYOTN
16 DRDWOTVSTEQD TYRCTPERT
17 CBZZEJCTGEQGS IHRQUEYK
18 NZQTSWLIYKOESHPIABSCR
19 YKYQDMEFFSORJDXCICZIR
20 SUNNEDVYEGNINAOLDETA F
21 MZRXPBNNPNEMSEKOPSYQV
22
23 BASKETRY
24 CIRCULATE
25 CRUCIFIED
26 DEMYSTIFY
27 DOUBLE
28 DRAWING
29 DRIZZLE
30 ENROLLED
31 LEAN
32 LOANING
33 MASSACRED

> DEBUG CONSOLE
> TERMINAL

fathiyahana_@DESKTOP-LEJV6F6:/mnt/c/Users/LEN
OVO/Kuliah/Semester 4/Strategi Algoritma/Tuci
l 1/Word-Search-Puzzle$ make run
WORD SEARCH GAME ENGINE
originally made by Hana Fathiyah

insert file's name: test/medium2.txt
QDCBBVFWDERCASSAMGEKR
CBEAPKUITBXROTINOMNMJ
MHTPDEUJGZBGNIWARDKRG
OIQOQZVLOIDLYVPKUOCU
RPP LUMAAYJJMRTKLVLPE
AWR LBSJDLQCKBISCDHLCU
LVCULOQFI FALASZEHQEA O
IJZTEJUHVIBAQBIZDXD TO
SNLENWRHDQLRVNFHLLDJE
TRMRQWDEBAAATXYKDFEOBV
IVPFKUIZSCWCRFXEMTWAC
CXNLNLFBDPCTEUKXTEOGSG
XFFEIVQENTPPBDTANGBKX
QDACDCJMJKQSOJSLITJEH
ZNUBUQBYXHLZADHUXYOTN
DRDWOTVSTEQD TYRCTPERT
CBZZEJCTGEQGS IHRQUEYK
NZQTSWLIYKOESHPIABSCR
YKYQDMEFFSORJDXCICZIR
SUNNEDVYEGNINAOLDETA F
MZRXPBNNPNEMSEKOPSYQV
fathiyahana_@DESKTOP-LEJV6F6:/mnt/c/Users/LEN
OVO/Kuliah/Semester 4/Strategi Algoritma/Tuci
l 1/Word-Search-Puzzle$
fathiyahana_@DESKTOP-LEJV6F6:/mnt/c/Users/LEN
OVO/Kuliah/Semester 4/Strategi Algoritma/Tuci
l 1/Word-Search-Puzzle$
```

Waktu eksekusi program:

real 0m7.215s

user 0m0.000s

sys 0m0.027s

Jumlah total perbandingan huruf:

```
MZRXPBNNPNEMSEKOPSYQV
Jumlah perbandingan: 73714
```

g. input: *medium3.txt* (24 x 22)

```
test > medium3.txt
1  N E L C A B E D O A O H A Y L O F T U R B Y
2  D L A C I N A H C E M B M X F S D E O X N T
3  X C D T L J S P N E U G C K M C E Q P J A I
4  A L O B A Z S O A P P L A U D E D K Z I E W
5  L T A U Q D U W Y G Z I N K K Q B V R W S F
6  E I K L O G N I D O O L F X E G H K T X O X
7  E C J C G U R W F M R R M T E O U A N R Z N
8  H U R F B X Y E O U L K I D G U F F E O F T
9  W P E E S C U R A E J F G D U E Y G N T N S
10 L E O T C V E H L E S J N E O J O U N H K I
11 Z T F E I O G U L O H N I B W F L Q T R V L
12 H H O X P N U I A L Q F S M L F M I N O Q A
13 T Y E P D B D N Q P M B N I E J Y P K W F I
14 F L X U F M O U T T K B E L T R T X R B N C
15 J S E I S O B S C S I P P C G B H Y T A Q O
16 J G G R Z C R N T I Y Q S E I H H A D C C S
17 K Z N D Y O R C Q O N V I I K K F J A K T P
18 O I I I F R V I I K N G D F J V W C R O L M
19 I C W O Y K B Q B B Y X O O Q R X A A D Q E
20 A L O R I Y G S F B L E V O T W A N M M F Q
21 R B M V I F P N L J L Y E D Z M J C R O G Y
22 T N P X G E S C G A P E P M P F O E Z W B S
23 Q Y G O Z R Y C I L G Y F F J K V R X O V P
24 W D E H S H B O C U U L A E Y M L V T A U F
25
26 APPLAUDED
27 BOSTON
28 CANCER
29 CLIMBED
30 DEBACLE
31 DISPENSING
32 FLOODING
33 FOAL
```

```
> DEBUG CONSOLE
v TERMINAL

/Semester 4/Strategi Algoritma/Tucil 1/Word-Search-Puzz
le$ make run
WORD SEARCH GAME ENGINE
originally made by Hana Fathiyah

insert file's name: test/medium3.txt
N E L C A B E D O A O H A Y L O F T U R B Y
D L A C I N A H C E M B M X F S D E O X N T
X C D T L J S P N E U G C K M C E Q P J A I
A L O B A Z S O A P P L A U D E D K Z I E W
L T A U Q D U W Y G Z I N K K Q B V R W S F
E I K L O G N I D O O L F X E G H K T X O X
E C J C G U R W F M R R M T E O U A N R Z N
H U R F B X Y E O U L K I D G U F F E O F T
W P E E S C U R A E J F G D U E Y G N T N S
L E O T C V E H L E S J N E O J O U N H K I
Z T F E I O G U L O H N I B W F L Q T R V L
H H O X P N U I A L Q F S M L F M I N O Q A
T Y E P D B D N Q P M B N I E J Y P K W F I
F L X U F M O U T T K B E L T R T X R B N C
J S E I S O B S C S I P P C G B H Y T A Q O
J G G R Z C R N T I Y Q S E I H H A D C C S
K Z N D Y O R C Q O N V I I K K F J A K T P
O I I I F R V I I K N G D F J V W C R O L M
I C W O Y K B Q B B Y X O O Q R X A A D Q E
A L O R I Y G S F B L E V O T W A N M M F Q
R B M V I F P N L J L Y E D Z M J C R O G Y
T N P X G E S C G A P E P M P F O E Z W B S
Q Y G O Z R Y C I L G Y F F J K V R X O V P
W D E H S H B O C U U L A E Y M L V T A U F

fathiyahana_@DESKTOP-LEJW6F6:/mnt/c/Users/LE
NOVO/Kuliah/Semester 4/Strategi Algoritma/Tu
cil 1/Word-Search-Puzzle$
```

Waktu eksekusi program:

real 0m5.152s

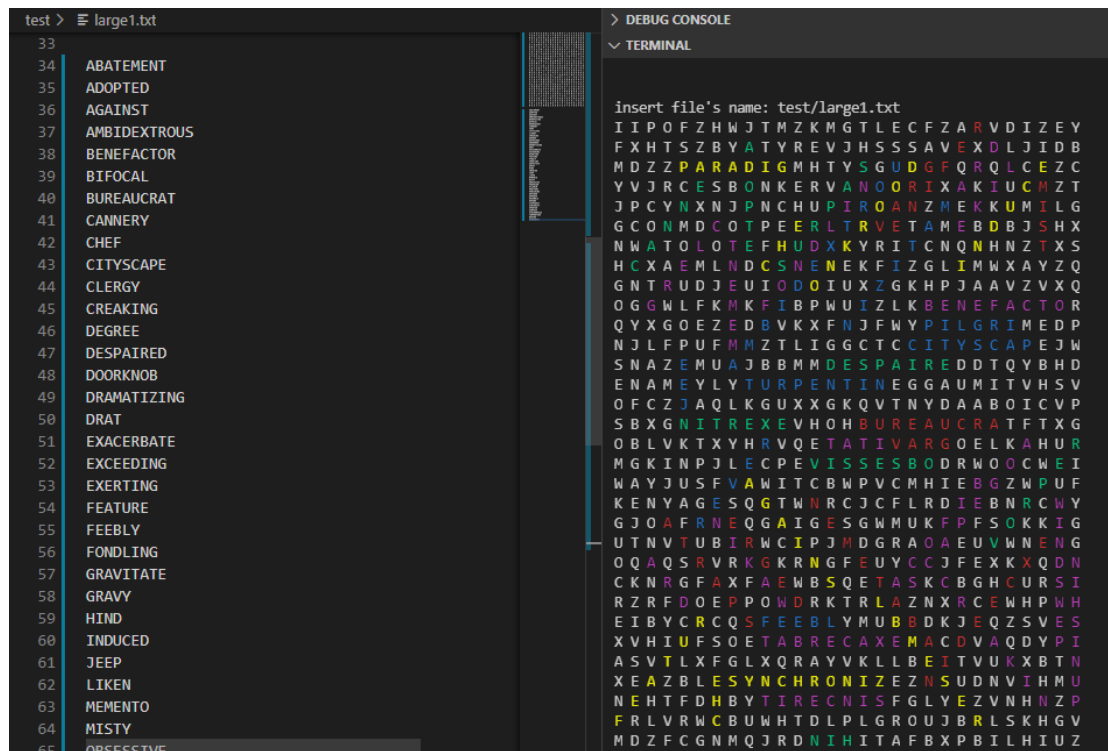
user 0m0.006s

sys 0m0.010s

Jumlah total perbandingan huruf:

```
W D E H S H B O C U U L A E Y M L V T A U F
Jumlah perbandingan: 88043
```

h. input: *large1.txt* (32 x 30)



The screenshot shows a debugger interface. On the left, a list of words is displayed, including ABATEMENT, ADOPTED, AGAINST, AMBIDEXTROUS, BENEFACITOR, BIFOCAL, BUREAUCRAT, CANNERY, CHEF, CITYSCAPE, CLERGY, CREAKING, DEGREE, DESPAIRED, DOORKNOB, DRAMATIZING, DRAT, EXACERBATE, EXCEEDING, EXERTING, FEATURE, FEBBLY, FONDLING, GRAVITATE, GRAVY, HIND, INDUCED, JEEP, LIKEN, MEMENTO, MISTY, and OBSESSIVE. On the right, a large grid of letters is shown, with some letters highlighted in various colors (red, green, blue, yellow, orange) to indicate matches or specific characters within the puzzle.

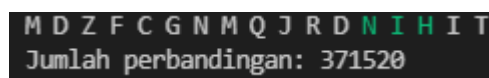
Waktu eksekusi program:

real 0m4.838s

user 0m0.002s

sys 0m0.017s

Jumlah total perbandingan huruf:



The screenshot shows a terminal window with the text "MDZFCGNMQJRDNIHIT" on the first line and "Jumlah perbandingan: 371520" on the second line.

i. input: *large2.txt* (31 x 31)

```
test > large2.txt
24  E F M S Y W C X Q Z I T Z F C T F J A D E U S E N D
25  N B I A K I S D E S M H S O M A H T W G K H R J Q Q
26  I B R B G J R Z T S C N F G R N N C O D A C S Q B M
27  L E Y X F E B T Q D O U O T N I O X H M Z S L R Q I
28  N K U R H Q J Y N U N Q N C R I F V E E E K R Y F B
29  A C T P S N A K E D S N F P X A R D N R P I F R I G
30  M F E G L R D J I T U T L G S I B A D L X N K A M G
31  M H K G V U X E U U L O T O Z V D D E K R R M P E G
32  S P X V D G M B T S T I K L V H A F V H V V Y V O H
33  R K P T F S U C L L I N E A R G T C R W X S G C Y G
34
35  ABEYANCE
36  ADDRESS
37  ANISEED
38  ASHAMED
39  ASSIMILATE
40  BAPTIZING
41  BASIN
42  BIDEED
43  CHART
44  CONDIMENT
45  CONSULT
46  CONTAINMENT
47  CRIME
48  DETERMINIST
49  DOWNSTAGE
50  EMPIRICIST
51  EXPERT
52  FEED
53  FRIGID
54  GERMANY
55  HEARING
```

Waktu eksekusi program:

real 0m6.002s

user 0m0.009s

sys 0m0.010s

Jumlah total perbandingan huruf:

```
KKPTFSUCLLINEAR
Jumlah perbandingan: 437071
```

j. *input: large3.txt (36 x 34)*

```
test > large3.txt
22 T Z N G U W R I N K L E D K J C D C V W Q T T T E
23 K J D L T M B I N R O H N E E R G P E R F F T V Q
24 R V S S E N R E H T E G O T O Y I Q W A E D S U N
25 J L D U S Z D D K N E Y L E T A C I L E D W T A M
26 U G N I T C O C N O C A W K R N B U H J R L T H S
27 W F M E F A G A B X L W J A L A O K T Y L O T D E
28 F C N C W A U H X I C J P P X J M M S Y F A O P L
29 W H A H O A U D T Q B Q K U C N L U R N F F L X Y
30 M O V I K C F R A M E D E Z R N H S T K G I R W J
31 B Q U C Y X N P R I G E B B K I W P Q R H N Q U F
32 V D T J D L U S Y N B X Y G W C T Q J L V G Q F D
33 R V L F R J N P F K X J P F Q W A Y Q F O L O Y W
34 E P E P I T T I N G N T D F O K U V S H T N E V E
35 T H C I X D K A F T E R C A R E M Y L E R U T A M
36 C N N V N H J S X P I Z N X W R E S T L E R K T R
37
38 ABOLITIONIST
39 AFTERCARE
40 BARGAINING
41 BRIEFLY
42 CHIC
43 CHOLESTEROL
44 CLUBS
45 CONCOCTING
46 DELICATELY
47 DIVISION
48 ELEVENTH
49 EXISTED
50 EXPLOITING
51 FERRY
52 FRAMED
53 GLIDER
54 GREENHORN

> DEBUG CONSOLE
✓ TERMINAL

fathiyahana_@DESKTOP-LE3V6F6: /mnt/c/Users/LENOVO/Kuliah/Semester 4/St
ucil 1/Word-Search-Puzzle$ make run
WORD SEARCH GAME ENGINE
originally made by Hana Fathiyah

insert file's name: test/large3.txt
X L W L J U B T H O R R I B L E H M S E T A R E P U C E R G H G V W
X E S S P P X Q G U W M D V B A S K I N B H E W E A Q H H V G N W K
V W B L F V T X D S R E D I L G A C E D A E T A B J A M W I S Y L A
M H C P S W Q O S Z T H A N K F U L N E S S Y B P H O R O F V T I R
D I D U G Q M U N T Y I T H D S B P I L L A G E S E J G O Z X C H U
W L S B M M F R O T A L S N A R T D A F A R T J S E A N C E Q E V Z
P V E D R B D S D D V Y E L B Y U P F C Z F F A Y J O X T N R G E E
G T R C W W T Y S H O Y S T O L I D F Q L W V F E R R Y C T O F R G
B D V G E S P D H I E X P L O I T I N G Z C X A Y X C C B T K R R N
G E I O X M I T I I G B N W B Q D F E S J R D H A S C N C M G H C I
N P C W T G O H L A B Y F S X Y A C U L D E T A R U T A S F C L F N
W Q I K K S W R E L D E J S T H M B A S E E P E D Q J W A Z U N I I
L I N Z I Y H I O K I T Q I E F E R O X F Y C Y B L U G U B X T F A
X S G C A S J V L O O Q L K X K C H O L E S T E R O L N S U O G E G
Q B X B K M S V R E C A F R D F D D W C I P K T L Z Z Z F L D N F R
G H S D M O W E H N S F V B E T O P F M X T H E E D F V Q K V K Y A
C P W Q O D G O S R R O P A T C F X W T E D T X M Q I O B L F M T B
N M G M O O L L E F V R R K S B K M K D Y C G O I T L A S B B A V H
J Y E L G O X V W D N M S D I A P X G P U K H K N I M F L E M R L K
O P O B H B I V H A P T G M X T H N F N Q P E T U I S K U P N S U C
Z U O N J N T Q A Z C K R H E E T C Q I O T Z S E S S V E J Z H D B
T Z N G U W R I N K L E D K J C D C V W Q T T T E K R T M U Y M A I
K J D L T M B I N R O H N E E R G P E R F F T V Q K C U I F E A V E
R V S S E N R E H T E G O T O Y I Q W A E D S U N U O A U Z J L P E
J L D U S Z D D K N E Y L E T A C I L E D W T A M O P V R C C L M D
U G N I T C O C N O C A W K R N B U H J R L T H S F I D O S Q O W R
W F M E F A G A B X L W J A L A O K T Y L O T D E H J S T R K W E A
F C N C W A U H X I C J P P X J M M S Y F A O P L P B A I E P B K O
```

Waktu eksekusi program:

real 0m8.793s

user 0m0.000s

sys 0m0.021s

Jumlah total perbandingan huruf:

```
C N N V N H J S X P I Z N X W R E
Jumlah perbandingan: 490590
```

#### IV. *Checklist* Penyelesaian Tugas

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

#### V. Link Repository

<https://github.com/hanafathiyah/Tugas-Kecil-Stima-1>