

**Laporan**  
**Tugas Kecil 1 IF2211 Strategi Algoritma**  
**Penyelesaian *Word Search Puzzle* dengan Algoritma *Brute Force***



Disusun Oleh:

Suryanto      13520059

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2022**

## DAFTAR ISI

DAFTAR ISI.....	2
ALGORITMA BRUTE FORCE.....	3
SOURCE CODE.....	4
SCREENSHOT INPUT OUTPUT .....	16
SOURCE CODE FILE .....	28

## ALGORITMA BRUTE FORCE

Tugas ini diprogram dengan bahasa pemrograman C dengan memanfaatkan beberapa library yang telah tersedia. *Library* yang digunakan adalah `stdio.h` untuk keperluan I/O program, `time.h` digunakan untuk menghitung waktu eksekusi, dan `string.h` untuk mengkonkat string (digunakan untuk menggabungkan nama file dan direktori file test case).

Program dijalankan dengan algoritma *brute force* yang mengikuti tahapan-tahapan berikut ini:

1. Program dimulai dengan pengecekan huruf pada matriks dengan indeks (0,0),
2. Jika huruf pada matriks dengan indeks (i,j) sesuai dengan huruf pertama pada kata yang dicek maka program akan melakukan pencocokan ke delapan arah dimulai dari vertikal atas dan dilanjutkan dengan arah lain sesuai arah jarum jam,
3. Apabila kedelapan arah telah diperiksa dan tidak ditemukan arah yang membentuk kata yang diuji, maka pencarian gagal dan program kembali ke tahap nomor 1 (tetapi dengan pencarian indeks kolom selanjutnya atau baris selanjutnya apabila telah mencapai kolom terakhir),
4. Jika ditemukan kata yang cocok, program akan mencatat arah, titik awal, dan titik akhir dari kata untuk digunakan pada keluaran solusi (pencarian berhasil).

## SOURCE CODE

Bahasa Pemrograman: C

```
#include <stdio.h>
#include <time.h>
#include <string.h>

typedef struct {
    char contents[100];
    int length;
    int arah;
    int xAwal;
    int yAwal;
    int xAkhir;
    int yAkhir;
} word;

//KAMUS FILE//
char currentChar;
static FILE * tapeFile;
char puzzle[100][100];
word kata[100];

#define True 1
#define False 0
#define newLine '\n'

void adv(){
    /* KAMUS LOKAL */
    static int * retval;

    /* ALGORITMA */
    retval = fscanf(tapeFile,"%c",&currentChar);
}

int startFile(char *fileName){

    /* Algoritma */
    tapeFile = fopen(fileName,"r");
    if (tapeFile == NULL){
        printf("Error: could not open file\n");
        return False;
    }
    else{
        adv();
        return True;
    }
}
```

```

    }
}

void readMatrix(int *row,int *col){
    /* KAMUS LOKAL */
    int i,j;

    /* ALGORITMA */
    i = 0;
    while (currentChar != newLine){
        j = 0;
        while (currentChar != newLine){
            puzzle[i][j] = currentChar;
            j++;
            adv();
            if (currentChar == ' ') adv();
        }
        adv();
        i++;
    }
    *row = i;
    *col = j;
}

void readKata(word solusi[100], int *banyakKata){
    /* KAMUS LOKAL */
    int i,j;

    /* ALGORITMA */
    adv();
    i = 0;
    while ((!feof(tapeFile))){
        j=0;
        while (currentChar != newLine && (!feof(tapeFile))){
            solusi[i].contents[j] = currentChar;
            adv();
            j++;
        }
        solusi[i].length=j;
        i++;
        adv();
    }
    *banyakKata = i;
}

void tulisKata(word sample){
    /* KAMUS LOKAL */
    int i;

```

```

    /* ALGORITMA */
    for (i = 0; i < sample.length; i++){
        printf("%c", sample.contents[i]);
    }
    printf("\n");
}

void tulisMatriks(word sample,int row, int col){
    /* KAMUS LOKAL */
    int i,j,k;

    /* ALGORITMA */
    tulisKata(sample);

    //vertikal-atas
    if (sample.arah == 1){
        k = sample.length-1;
        for (i = 0; i < row; i++){
            for ( j = 0; j < col; j++){
                if (j == sample.xAwal && i == sample.yAkhir){
                    printf("%c ", sample.contents[k]);
                    k--;
                    if (k >=0) sample.yAkhir++;
                }
                else{
                    printf("- ");
                }
            }
            printf("\n");
        }
    }

    //menyamping atas-kanan
    else if (sample.arah == 2){
        k = sample.length-1;
        for (i = 0; i < row; i++){
            for ( j = 0; j < col; j++){
                if (j == sample.xAkhir && i == sample.yAkhir){
                    printf("%c ", sample.contents[k]);

                    k--;
                    if (k >=0){
                        sample.yAkhir++;
                        sample.xAkhir--;
                    }
                }
            }
        }
    }
}

```

```

        else{
            printf("- ");
        }
    }
    printf("\n");
}

//mendatar kanan
else if (sample.arah == 3){
    k = 0;
    for (i = 0; i < row; i++){
        for (j = 0; j < col; j++){
            if (j == sample.xAwal && i == sample.yAwal){
                printf("%c ", sample.contents[k]);
                k++;
                if (k < sample.length){
                    sample.xAwal++;
                }
            }
            else{
                printf("- ");
            }
        }
        printf("\n");
    }
}

//menyamping kanan-bawah
else if (sample.arah == 4){
    k = 0;
    for (i = 0; i < row; i++){
        for (j = 0; j < col; j++){
            if (j == sample.xAwal && i == sample.yAwal){
                printf("%c ", sample.contents[k]);
                k++;
                if (k < sample.length){
                    sample.xAwal++;
                    sample.yAwal++;
                }
            }
            else{
                printf("- ");
            }
        }
        printf("\n");
    }
}

```

```

}

//vertikal bawah
else if (sample.arah == 5){
    k = 0;
    for (i = 0; i < row; i++){
        for (j = 0; j < col; j++){
            if (j == sample.xAwal && i == sample.yAwal){
                printf("%c ", sample.contents[k]);
                k++;
                if (k < sample.length){
                    sample.yAwal++;
                }
            }
            else{
                printf("- ");
            }
        }
        printf("\n");
    }
}

//menyamping kiri-bawah
else if (sample.arah == 6){
    k = 0;
    for (i = 0; i < row; i++){
        for (j = 0; j < col; j++){
            if (j == sample.xAwal && i == sample.yAwal){
                printf("%c ", sample.contents[k]);
                k++;
                if (k < sample.length){
                    sample.xAwal--;
                    sample.yAwal++;
                }
            }
            else{
                printf("- ");
            }
        }
        printf("\n");
    }
}

//mendatar kiri
else if (sample.arah == 7){
    k = sample.length-1;
    for (i = 0; i < row; i++){
        for (j = 0; j < col; j++){

```



```

        if (j == sample.xAkhir && i == sample.yAwal){
            printf("%c ", sample.contents[k]);
            k--;
            if (k >=0 ){
                sample.xAkhir++;
            }
        }
        else{
            printf("- ");
        }
    }
    printf("\n");
}

//menyamping atas-kiri
else if (sample.arah == 8){
    k = sample.length-1;
    for (i = 0; i< row; i++){
        for ( j =0; j< col; j++){
            if (j == sample.xAkhir && i == sample.yAkhir){
                printf("%c ", sample.contents[k]);
                k--;
                if (k >=0 ){
                    sample.xAkhir++;
                    sample.yAkhir++;
                }
            }
            else{
                printf("- ");
            }
        }
        printf("\n");
    }
}

}

int solver(word sample, int row, int col){
    /* KAMUS LOKAL */
    int i, j, k,ii,jj,found,temp,counter;

    /* ALGORITMA */
    //pengecekan searah jarum jam dimulai dari atas-kanan
    found = False;
    i = 0;
    counter = 0;
    while ((!found) && (i < row || j < col)){
        j = 0;

```

```

while (j<col && !found){
    temp = sample.contents[0] == puzzle[i][j];
    counter++;
    if (temp){

        sample.xAwal = j;
        sample.yAwal = i;

        k = 1;
        if (i>=sample.length-1){
            ii = i-1;
            while (temp && !found){
                counter++;
                temp = (sample.contents[k] == puzzle[ii][j]);
                k++;
                if (k == sample.length && temp){
                    found = True;
                    sample.xAkhir = j;
                    sample.yAkhir = ii;
                    sample.arah = 1;
                }
                ii--;
            }

            //menyamping atas-kanan
            if (!temp){ //apabila pencarian menurun sudah
dilakukan dan gagal mendapatkan solusi
                ii = i-1;
                k = 1;

                if (j<= col -sample.length-1){
                    temp = True;
                    // printf("hello %d", j);
                    jj = j+1;

                    while (temp && !found){
                        counter++;
                        temp = (sample.contents[k] == puzzle[ii][jj]);
                        k++;
                        if (k == sample.length && temp){
                            found = True;
                            sample.xAkhir = jj;
                            sample.yAkhir = ii;
                            sample.arah = 2;
                        }
                        ii--;
                        jj++;

```

```

    }
    }
}

//menyamping atas-kiri
if (!temp){ //apabila pencarian menurun sudah
dilakukan dan gagal mendapatkan solusi
    ii = i-1;
    k = 1;

    if (j>= sample.length-1){

        temp = True;
        jj = j-1;

        while (temp && !found){
            counter++;
            temp = (sample.contents[k] == puzzle[ii][jj]);
            k++;

            if (k == sample.length && temp){
                found = True;
                sample.xAkhir = jj;
                sample.yAkhir = ii;
                sample.arah = 8;
            }
            ii--;
            jj--;
        }
    }
}

}

//cek datar kanan
if (!found && (j<= col -sample.length))
{
    jj = j+1;
    k = 1;
    temp = True;
    while (temp && !found){
        counter++;
        temp = (sample.contents[k] == puzzle[i][jj]);
        k++;

        if (k == sample.length && temp){
            found = True;
            sample.xAkhir = jj;

```

```

        sample.yAkhir = i;
        sample.arah = 3;
    }
    jj++;
}

}

//cek datar kiri
if (!found && (j>=sample.length-1))
{
    temp = True;
    jj = j-1;
    k = 1;

    while (temp && !found){
        counter++;
        temp = (sample.contents[k] == puzzle[i][jj]);
        k++;

        if (k == sample.length && temp){
            found = True;
            sample.xAkhir = jj;
            sample.yAkhir = i;
            sample.arah = 7;
        }
        jj--;
    }

}

//cek menurun-bawah, menyamping bawah-kanan, menyamping bawah-
kiri
if (!found && i<=row - sample.length){
    ii = i+1;
    k = 1;
    temp = True;
    //menurun-bawah
    while (temp && !found){
        counter++;
        temp = (sample.contents[k] == puzzle[ii][j]);
        k++;

        if (k == sample.length && temp){
            found = True;
            sample.xAkhir = j;
            sample.yAkhir = ii;

```

```

        sample.arah = 5;
    }
    ii++;
}

//menyamping bawah-kanan
if (!temp){           //apabila pencarian menurun sudah
dilakukan dan gagal mendapatkan solusi
    ii = i+1;
    k = 1;
    if (j<= col -sample.length){
        temp = True;
        jj = j+1;

        while (temp && !found){
            counter++;
            temp = (sample.contents[k] == puzzle[ii][jj]);
            k++;

            if (k == sample.length && temp){
                found = True;
                sample.xAakhir = jj;
                sample.yAakhir = ii;
                sample.arah = 4;
            }
            ii++;
            jj++;
        }
    }
}

//menyamping bawah-kiri
if (!temp){           //apabila pencarian menurun sudah
dilakukan dan gagal mendapatkan solusi

    ii = i+1;
    k = 1;

    if (j>= sample.length-1){

        temp = True;
        jj = j-1;

        while (temp && !found){
            counter++;
            temp = (sample.contents[k] == puzzle[ii][jj]);
            k++;

```



```

    readMatrix(&row,&col);

    readKata(kata, &banyakKata);

    start = clock();
    sumPerbandingan = 0;
    for ( i = 0; i <banyakKata; i++){
        sumPerbandingan += solver(kata[i], row,col);
        printf("\n\n");
    }

    end = clock();
    cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;
    printf("Waktu Eksekusi program : %lf (s)\n", cpu_time_used);
    printf("Total Perbandingan : %d\n", sumPerbandingan);

}
else{
    // do nothing
}
}

```

## SCREENSHOT INPUT OUTPUT

1. small1.txt

[illegible]

## 2. small2.txt



### 3. small3.txt



[illegible][illegible][illegible]

GLANCE

G  
L  
A  
N  
C  
E

Banyak perbandingan huruf : 57

[illegible][illegible][illegible][illegible]

DECOMPRESS

S  
 S  
 E  
 R  
 P  
 M  
 O  
 C  
 E  
 D

Banyak perbandingan huruf : 299

[illegible]

```

LEFTY
      Y
      T
      F
      E
      L

Banyak perbandingan huruf : 366

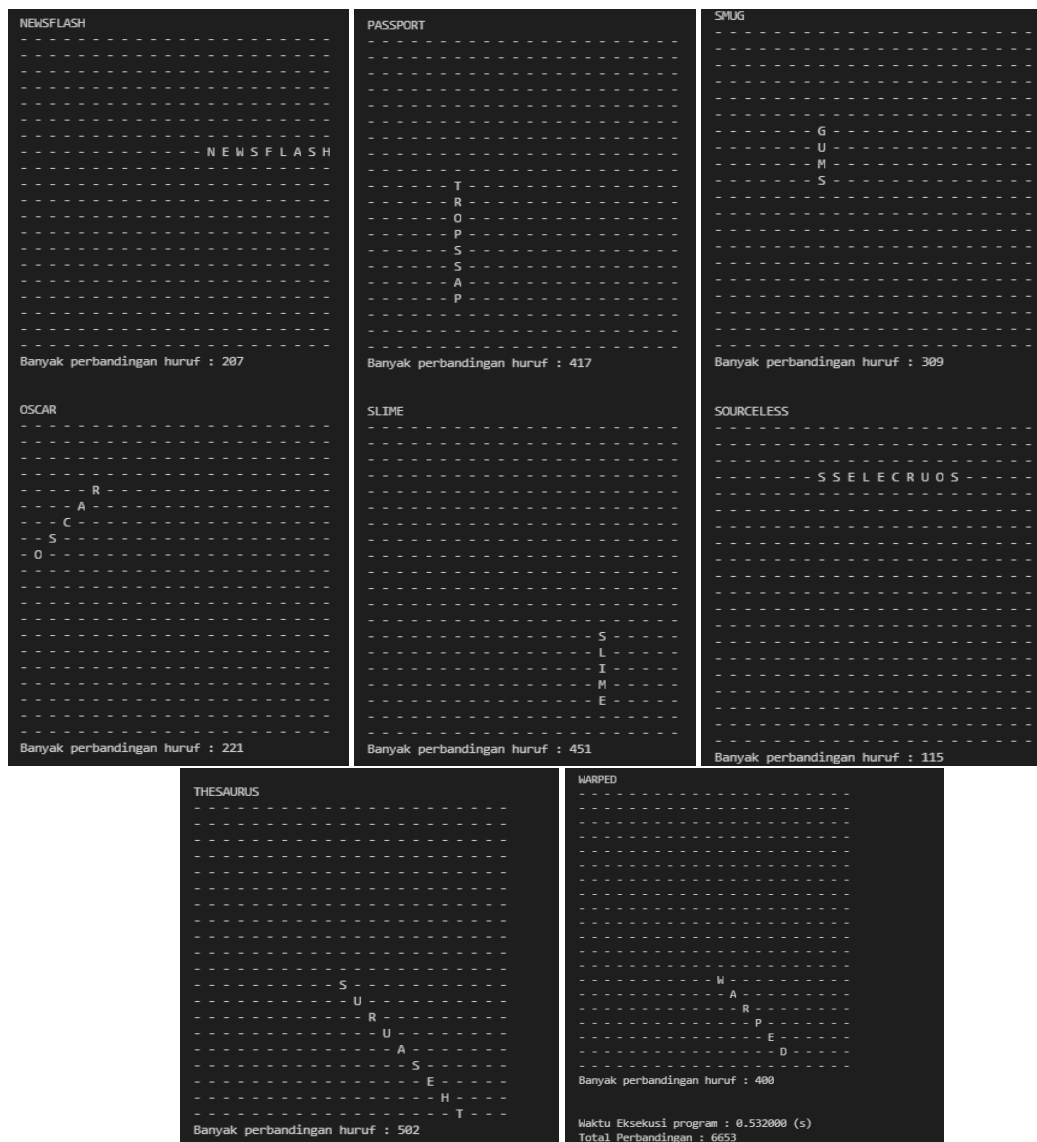
```

```

MIGHT
      M
      I
      G
      H
      T

Banyak perbandingan huruf : 295

```



## 5. medium2.txt

**BLASPHEMER**

CUBIC

## CUTLERY

DEFLECTION

DEPOSITED

## FUSION

GARMENT

**GENTLEFOLK**

6. medium3

<p>medium3.txt</p> <p>ADHERENCE</p> <p>- E -</p> <p>- C -</p> <p>- N -</p> <p>- E -</p> <p>- R -</p> <p>- E -</p> <p>- H -</p> <p>- D -</p> <p>- A -</p> <p>Banyak perbandingan huruf : 336</p> <p>AMUSE</p> <p>- E S U M A -</p> <p>Banyak perbandingan huruf : 463</p> <p>BIBLICAL</p> <p>- B -</p> <p>- I -</p> <p>- B -</p> <p>- L -</p> <p>- I -</p> <p>- C -</p> <p>- A -</p> <p>- L -</p> <p>Banyak perbandingan huruf : 358</p>	<p>BRAID</p> <p>- D -</p> <p>- I -</p> <p>- A -</p> <p>- R -</p> <p>- B -</p> <p>Banyak perbandingan huruf : 285</p> <p>BRINY</p> <p>- B R I N Y</p> <p>Banyak perbandingan huruf : 27</p> <p>DISHONEST</p> <p>- T S E N O H S I D - - - - -</p> <p>Banyak perbandingan huruf : 538</p>	<p>FINAL</p> <p>- L -</p> <p>- A -</p> <p>- N -</p> <p>- I -</p> <p>- F -</p> <p>Banyak perbandingan huruf : 273</p> <p>INERTNESS</p> <p>- I N E R T N E S S -</p> <p>Banyak perbandingan huruf : 118</p> <p>INSCRIBED</p> <p>- D -</p> <p>- E -</p> <p>- B -</p> <p>- I -</p> <p>- R -</p> <p>- C -</p> <p>- S -</p> <p>- N -</p> <p>- I -</p> <p>Banyak perbandingan huruf : 359</p>
---	---	--

```

PRESCRIBED
-----
P R E S C R I B E D
-----
Banyak perbandingan huruf : 258

IRREGULAR
-----
I R R E G U L A R
-----
Banyak perbandingan huruf : 42

SHOWER
-----
S H O W E R
-----
R E N O H S
-----
Banyak perbandingan huruf : 415

RECOVER
-----
R E C O V E R
-----
Banyak perbandingan huruf : 78

JUNE
-----
J U N E
-----
Banyak perbandingan huruf : 261

SOLID
-----
S O L I D
-----
Banyak perbandingan huruf : 88

SACKING
-----
S A C K I N G
-----
Banyak perbandingan huruf : 528

PERJURY
-----
P E R J U R Y
-----
Banyak perbandingan huruf : 391

SPATULA
-----
S P A T U L A
-----
Banyak perbandingan huruf : 468

TOTALLY
-----
T O T A L L Y
-----
Banyak perbandingan huruf : 448

URUGUAY
-----
U R U G U A Y
-----
Banyak perbandingan huruf : 335

Waktu Eksekusi program : 0.629000 (s)
Total Perbandingan : 6021

```

## 7. large1.txt



Banyak perbandingan huruf : 790

Waktu Eksekusi program : 3.120000 (s)

Total Perbandingan : 36302

8. large2.txt



YULE

Y  
U  
L  
E

Banyak perbandingan huruf : 509

Waktu Eksekusi program : 3.304000 (s)

Total Perbandingan : 35884

## SOURCE CODE FILE

<https://github.com/SurTan02/WordSearch-Puzzle-Solver>

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