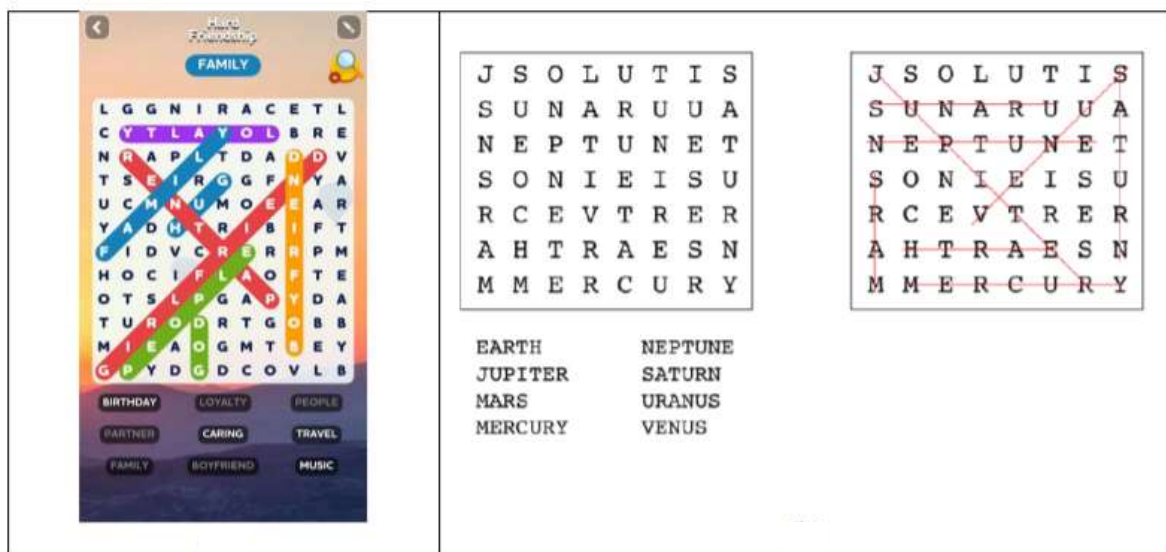


LAPORAN TUGAS KECIL 1

Mata Kuliah IF2211 Strategi Algoritma



Nama Penulis:

Aji Andhika Falah 13520012

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2022

BAB I

Algoritma *Brute Force*

Algoritma *brute force* adalah algoritma yang lempang (*straightforward*) dalam menyelesaikan suatu persoalan. Algoritma ini biasanya didasarkan pada pernyataan masalah (*problem statement*) dan definisi konsep yang dilibatkan. Algoritma *brute force* memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (*obvious way*). Untuk menyelesaikan persoalan permainan *word search*, langkah-langkah algoritma yang digunakan adalah:

1. *Word search* yang akan diselesaikan dimasukan ke sebuah matriks dengan tambahan info di tiap sel akan di print “-“ atau di print huruf di sel tersebut. Untuk awal-awal ditandai akan di print “-“.
2. kata yang akan dicari, dipisah menjadi per huruf dan dimasukan ke matriks (sebenarnya array cukup).
3. Dicek sel sel matriks *word search* sampai ditemukan sel yang memiliki huruf yang sama dengan huruf pertama kata yang akan dicari, dan sel tersebut ditandai akan di print huruf di sel tersebut.
4. Cek sel sel disekeliling sel yang memiliki huruf sama tersebut. Jika di salah satu arah memiliki huruf yang sama dengan huruf kedua dari kata yang dicari, maka sel tersebut juga ditandai akan di print huruf di sel tersebut.
5. Cek terus ke arah tersebut. Jika ditemukan ada huruf yang tidak sama atau ternyata *out of bounds*, maka mundur. Jika ditemukan semua huruf sama, maka di print matriks *word search* tersebut.
6. Kembalikan sel yang ditandai akan di print huruf di sel tersebut menjadi akan di print “-“.
7. Lakukan dari langkah 2 lagi sampai semua kata ditemukan.

BAB 2

Source Program

Untuk program, saya menggunakan bahasa Java. Ada 3 file yang telah dibuat, driver.java, Matriks.java, dan pair.java. File driver.java adalah sebagai driver program ini, file Matriks.java adalah yang mengolah dari file.txt menjadi matriks sampai menampilkan hasil, dan file pair.java berfungsi untuk mengolah sebuah tuple.

File driver.java

```
package bin;

import java.io.FileNotFoundException;
import java.util.Scanner;

public class driver {

    static Scanner keyboard = new Scanner(System.in);

    public static void main(String[] args)
    {
        String f;

        System.out.print("Enter file : ");
        f = keyboard.nextLine();
        try
        {
            Matriks.bacaFileMatriks(f);
            System.out.println();
        }
        catch(FileNotFoundException e)
        {
            keyboard.close();
            System.out.println("There is no such file");
            return;
        }
        System.out.println();
    }
}
```

File Matriks.java

```
package bin;
```

```

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class Matriks {
    int Brs,Kol;
    String [][][] matriks;

    /*Konstruktor Matriks*/
    public Matriks (int Brs, int Kol){
        this.Brs = Brs;
        this.Kol = Kol;
        this.matriks = new String [Brs][Kol][2];
    }

    /*Selektor nilai elemen pada Matriks[i][j]*/
    public String Elmt (int i, int j, int k){
        return matriks[i][j][k];
    }

    /*Mengganti nilai elemen pada Matriks[i][j] menjadi val*/
    public void SetElmt (int i, int j, int k, String val){
        matriks[i][j][k] = val;
    }

    /*Mereturn banyak baris Matriks*/
    public int Baris(){
        return Brs;
    }

    /*Mereturn banyak kolom Matriks*/
    public int Kolom(){
        return Kol;
    }

    /*Membaca Matriks dari file Eksternal dan mengolahnya*/
    public static void bacaFileMatriks (String namaFile) throws
FileNotFoundException{
        int Brs = 0;
        int wordBrs = -1;
        int Kol = 0;
        int wordKol = 0;
        int compare = 0;
        int i,ii,j;
        Matriks M = null;
        boolean found = false;
        boolean inputMode = true;

```

```

pair result;

File myFile = new File(namaFile);
Scanner fileReader = new Scanner(myFile);
String fileLine;

while (fileReader.hasNextLine()){
    fileLine = fileReader.nextLine();
    if (fileLine == "")
    {
        inputMode = false;
    }
    if (inputMode)
    {
        Kol = fileLine.split(" ").length;
        Brs++;
    }
    else
    {
        wordBrs++;
    }
}
fileReader.close();

fileReader = new Scanner(myFile);
M = new Matriks(Brs, Kol);
for (i=0; i<Brs; i++){
    for (j=0; j<Kol; j++){
        M.SetElmt(i, j, 0, fileReader.next());
        M.SetElmt(i, j, 1, "-");
    }
}

long startTime = System.nanoTime();
for (i=Brs+1; i<Brs+wordBrs+1;i++)
{
    fileLine = fileReader.next();
    System.out.printf("\n++ Word %s +=\n", fileLine);
    wordKol = fileLine.split("").length;
    Matriks word = new Matriks(1,wordKol);
    for (j=0; j<wordKol; j++)
    {
        word.SetElmt(0, j, 0, fileLine.substring(j, j+1));
    }

    outerloop:
    for (ii=0; ii<Brs; ii++)
    {

```

```

        for (j=0; j<Kol; j++)
        {
            compare++;
            if (M.Elmt(ii, j, 0).equals(word.Elmt(0, 0, 0)))
            {
                M.SetElmt(ii, j, 1, "a");
                result = M.cekAround(word, ii, j, wordKol);
                compare = compare + result.getComp();
                found = result.getBoolean();
                M.SetElmt(ii, j, 1, "-");
                if (found)
                {
                    break outerloop;
                }
            }
        }
    }

    if(!found)
    {
        System.out.printf("Kata tidak ditemukan\n");
    }
}

fileReader.close();
long endTime = System.nanoTime();
System.out.format("\nElapsed Time for this program is %d ms", (endTime
- startTime)/1000000);
System.out.format("\nTotal comparison is %d time(s)", compare);
}

/*Menuliskan suatu Matriks pada layar user*/
public void tulisMatriks(){
    int i,j;
    for(i=0; i < this.Brs; i++){
        for(j=0; j < this.Kol; j++){
            if (!"-".equals(matriks[i][j][1])){
                System.out.printf("%S", this.matriks[i][j][0]);
            }
            else{
                System.out.printf("-");
            }
            if(j < this.Kol - 1){
                System.out.printf(" ");
            }
        }
        if(i < this.Brs - 1) System.out.println("");
    }
    System.out.printf("\n");
}

```

```

}

/* Pengecekan kata pada word search */
public pair cekAround(Matriks word, int i, int j, int wordKol){
    pair result;
    int compare = 0;

    result = this.cekKanan(word, i, j, wordKol);
    compare = result.getComp();
    if (!result.getBoolean())
    {
        result = this.cekBawah(word, i, j, wordKol);
        compare = compare + result.getComp();
    }
    if (!result.getBoolean())
    {
        result = this.cekKiri(word, i, j, wordKol);
        compare = compare + result.getComp();
    }
    if (!result.getBoolean())
    {
        result = this.cekAtas(word, i, j, wordKol);
        compare = compare + result.getComp();
    }
    if (!result.getBoolean())
    {
        result = this.cekKananBawah(word, i, j, wordKol);
        compare = compare + result.getComp();
    }
    if (!result.getBoolean())
    {
        result = this.cekKiriBawah(word, i, j, wordKol);
        compare = compare + result.getComp();
    }
    if (!result.getBoolean())
    {
        result = this.cekKananAtas(word, i, j, wordKol);
        compare = compare + result.getComp();
    }
    if (!result.getBoolean())
    {
        result = this.cekKiriAtas(word, i, j, wordKol);
        compare = compare + result.getComp();
    }
    return new pair (compare,result.getBoolean());
}

public pair cekKiri(Matriks word, int i, int j, int wordKol){

```

```

        boolean found = false;
        int wordNum = 1;
        int compare = 0;

        j--;
        while(!found && j>=0)
        {
            compare++;
            if (this.Elmt(i, j, 0).equals(word.Elmt(0, wordNum, 0)))
            {
                this.SetElmt(i, j, 1, "a");
                j--;
                wordNum++;
                if (wordNum==wordKol)
                {
                    found = true;
                    this.tulisMatriks();
                }
            }

            else
            {
                break;
            }
        }

        while(wordNum != 1)
        {
            j++;
            this.SetElmt(i, j, 1, "-");
            wordNum--;
        }
        return new pair(compare,found);
    }

    public pair cekKanan(Matriks word, int i, int j, int wordKol){
        boolean found = false;
        int wordNum = 1;
        int compare = 0;

        j++;
        while(!found && j<this.Kol)
        {
            compare++;
            if (this.Elmt(i, j, 0).equals(word.Elmt(0, wordNum, 0)))
            {
                this.SetElmt(i, j, 1, "a");
                j++;
                wordNum++;
            }
        }
    }

```



```

        if (wordNum==wordKol)
        {
            found = true;
            this.tulisMatriks();
        }
    }

    else
    {
        break;
    }
}

while(wordNum != 1)
{
    j--;
    this.SetElmt(i, j, 1, "-");
    wordNum--;
}
return new pair(compare,found);
}

public pair cekAtas(Matriks word, int i, int j, int wordKol){
    boolean found = false;
    int wordNum = 1;
    int compare = 0;

    i--;
    while(!found && i>=0)
    {
        compare++;
        if (this.Elmt(i, j, 0).equals(word.Elmt(0, wordNum, 0)))
        {
            this.SetElmt(i, j, 1, "a");
            i--;
            wordNum++;
            if (wordNum==wordKol)
            {
                found = true;
                this.tulisMatriks();
            }
        }

        else
        {
            break;
        }
    }
}

```

```

        while(wordNum != 1)
        {
            i++;
            this.SetElmt(i, j, 1, "-");
            wordNum--;
        }
        return new pair(compare,found);
    }

    public pair cekBawah(Matriks word, int i, int j, int wordKol){
        boolean found = false;
        int wordNum = 1;
        int compare = 0;

        i++;
        while(!found && i<this.Brs)
        {
            compare++;
            if (this.Elmt(i, j, 0).equals(word.Elmt(0, wordNum, 0)))
            {
                this.SetElmt(i, j, 1, "a");
                i++;
                wordNum++;
                if (wordNum==wordKol)
                {
                    found = true;
                    this.tulisMatriks();
                }
            }

            else
            {
                break;
            }
        }

        while(wordNum != 1)
        {
            i--;
            this.SetElmt(i, j, 1, "-");
            wordNum--;
        }
        return new pair(compare,found);
    }

    public pair cekKiriAtas(Matriks word, int i, int j, int wordKol){
        boolean found = false;

```

```

int wordNum = 1;
int compare = 0;

j--;
i--;
while(!found && j>=0 && i>=0)
{
    compare++;
    if (this.Elmt(i, j, 0).equals(word.Elmt(0, wordNum, 0)))
    {
        this.SetElmt(i, j, 1, "a");
        j--;
        i--;
        wordNum++;
        if (wordNum==wordKol)
        {
            found = true;
            this.tulisMatriks();
        }
    }

    else
    {
        break;
    }
}

while(wordNum != 1)
{
    j++;
    i++;
    this.SetElmt(i, j, 1, "-");
    wordNum--;
}
return new pair(compare,found);
}

public pair cekKiriBawah(Matriks word, int i, int j, int wordKol){
    boolean found = false;
    int wordNum = 1;
    int compare = 0;

    j--;
    i++;
    while(!found && j>=0 && i<this.Brs)
    {
        compare++;
        if (this.Elmt(i, j, 0).equals(word.Elmt(0, wordNum, 0)))

```

```

        {
            this.SetElmt(i, j, 1, "a");
            j--;
            i++;
            wordNum++;
            if (wordNum==wordKol)
            {
                found = true;
                this.tulisMatriks();
            }
        }

        else
        {
            break;
        }
    }

    while(wordNum != 1)
    {
        j++;
        i--;
        this.SetElmt(i, j, 1, "-");
        wordNum--;
    }
    return new pair(compare,found);
}

public pair cekKananAtas(Matriks word, int i, int j, int wordKol){
    boolean found = false;
    int wordNum = 1;
    int compare = 0;

    j++;
    i--;
    while(!found && j<this.Kol && i>=0)
    {
        compare++;
        if (this.Elmt(i, j, 0).equals(word.Elmt(0, wordNum, 0)))
        {
            this.SetElmt(i, j, 1, "a");
            j++;
            i--;
            wordNum++;
            if (wordNum==wordKol)
            {
                found = true;
                this.tulisMatriks();
            }
        }
    }
}

```

```

        }
    }

    else
    {
        break;
    }
}

while(wordNum != 1)
{
    j--;
    i++;
    this.SetElmt(i, j, 1, "-");
    wordNum--;
}
return new pair(compare,found);
}

public pair cekKananBawah(Matriks word, int i, int j, int wordKol){
    boolean found = false;
    int wordNum = 1;
    int compare = 0;

    j++;
    i++;
    while(!found && j<this.Kol && i<this.Brs)
    {
        compare++;
        if (this.Elmt(i, j, 0).equals(word.Elmt(0, wordNum, 0)))
        {
            this.SetElmt(i, j, 1, "a");
            j++;
            i++;
            wordNum++;
            if (wordNum==wordKol)
            {
                found = true;
                this.tulisMatriks();
            }
        }

        else
        {
            break;
        }
    }
}

```

```

        while(wordNum != 1)
        {
            j--;
            i--;
            this.SetElmt(i, j, 1, "-");
            wordNum--;
        }
        return new pair(compare,found);
    }
}

```

File pair.java

```

package bin;

public class pair {
    int first;
    boolean second;

    public pair(int first, boolean second) {
        this.first = first;
        this.second = second;
    }

    public int getComp() {
        return first;
    }

    public boolean getBoolean() {
        return second;
    }
}

```

BAB 3

Screenshot input dan output

1. small1.txt

input:

```
SNIFFUMCUEWDHSE
OMTESTZAKQREBEG
HRVOJSUAGQDGFITG
ZHATDACBAISJQKS
BREADDEIKSSTKCOO
SIRWEYNPTONBMOG
LWMFCOLULSAUWCW
HBFWQANTLVSTWXX
NOILYHCJOVSLEAL
CRUSGBMCRKIRUEO
DBMUCIISTAOZQBA
ZQOIXUULEARGKRV
UDAEELGIVEDCQOZE
MFRHAKUTWIMPCZS
TSDRLSZUSQQYRHO

BISCUITS
BREAD
COFFEECAKE
COOKIES
CROISSANTS
DISPLAYS
DOUGHNUTS
EGGS
LOAVES
MILK
MIXERS
MUFFINS
SUGAR
SWEETROLL
YEAST
```

output:

Enter file : test/small1.txt

== Word BISCUITS ==

```
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- B - - - - -
- I - - - - -
- S - - - - -
- C - - - - -
- U - - - - -
- I - - - - -
- T - - - - -
- S - - - - -
```

== Word BREAD ==

```
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
B R E A D - - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
```

== Word COFFEECAKE ==

```
- - - - - E - - - -
- - - - - K - - - -
- - - - - A - - - -
- - - - - C - - - -
- - - E - - - -
- - E - - - -
- F - - - -
- F - - - -
- O - - - -
C - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
```


== Word COOKIES ==

== Word CROISSANTS ==

== Word DISPLAYS ==

==+ Word DOUGHNUTS ==+

== Word EGGS ==

== Word LOAVES ==

M I L K

S
R
E
X
I
M

SNIFFUM - - - - -

```
Elapsed Time for this program is 317 ms
Total comparison is 2436 time(s)
```

2. small2.txt

input:

```
W K B N X M S C H T H G I E
W R U Q J W J I A V T Y K W
F G L D Q X R O K U F K L C
I C B E P F R A C T I O N K
C E I N T T E N T H F P N G
J D T O C H T X I S E U Q N
T N R M D A C U Y P M Z J S
R K K I W R D U C E X Y E Q
B V V N H P L H R U U V C Q
Z C N A Q T B A T H E C B T
C U Y T G E T S F N A H M D
Y B B O Z O F P T R I L M M
P Q W R R L X H B A E N F O
L Z V Q R E T R A U Q K Y C
```

```
DENOMINATOR
HALF
SEVENTH
EIGHTH
NINTH
SIXTH
FIFTH
NUMERATOR
TENTH
FRACTION
QUARTER
|THIRD
```

output:

```
Enter file : test/small2.txt
```

```
++ Word DENOMINATOR ++
- - - - -
- - - - -
- - - D - - - - -
- - - E - - - - -
- - - N - - - - -
- - - O - - - - -
- - - M - - - - -
- - - I - - - - -
- - - N - - - - -
- - - A - - - - -
- - - T - - - - -
- - - O - - - - -
- - - R - - - - -
- - - - -
```

Word SEVENTH

[illegible]

==+ Word SIXTH ==+

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - H T X I S - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

==+ Word FIFTH ==+

- - - - - H - - -
- - - - - T - - -
- - - - - F - - -
- - - - - I - - -
- - - - - F - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

==+ Word NUMERATOR ==+

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


```

+=+ Word THIRD +=+
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- D - - - - -
- - R - - - - -
- - - I - - - - -
- - - - H - - - - -
- - - - - T - - - - -
- - - - -
- - - - -
- - - - -
- - - - -

Elapsed Time for this program is 199 ms
Total comparison is 1583 time(s)

```

3. small3.txt

input:

```

B B R F T S R F Q D Q G S M S
I N E N A K I T I P U G C B O
D C E M N I N B O O T S A W C
K E E Y K R G V A C O T R U K
D L B S T T S Y E O O D F V S
F L R U O C R E D I T C A R D
L P A N P H V J E W E L E R Y
I H C G S F L O W E R P I N S
P O E L S M S A Z X P C A S H
F N L A H A I R R I B B I N S
L E E S I K D U G C O A T S O
O I T S R E P O S S P U R S E
P Q S E T U T Z U X S H A T S
S K O S I P A H C S H O E S N
S N E A K E R S D R E S S F I

BOOTS
FLOWERPINS
SHIRT
BRACELETS
HAIRRIBBINS
SHOES
CASH
HATS
SKIRT
CELLPHONE
JEWELERY
SNEAKERS
COATS
MAKEUP
SOCKS
CREDITCARD
PURSE
SUNGLASSES
DRESS
RINGS
TANKTOPS
FLIPFLOPS
SCARF

```

output:

```
Enter file : test/small3.txt
```

```
+==+ Word BOOTS +==+
```

```
- - - - -  
- - - - -  
- - - - - B O O T S - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -
```

```
+==+ Word FLOWERPINS +==+
```

```
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - - F L O W E R P I N S  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -
```

```
+==+ Word SHIRT +==+
```

```
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - - S - - - - -  
- - - - - H - - - - -  
- - - - - I - - - - -  
- - - - - R - - - - -  
- - - - - T - - - - -  
- - - - -  
- - - - -
```

== Word BRACELETS ==

== Word HAIRRIBBINS ==

== Word SHOES ==

=== Word CASH ===

=== Word HATS ===

=== Word SKIRT ===

== Word RINGS ==


```

+== Word FLIPFLOPS +==
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
F - - - - -
L - - - - -
I - - - - -
P - - - - -
F - - - - -
L - - - - -
O - - - - -
P - - - - -
S - - - - -
- - - - -

+== Word SCARF +==
- - - - - S - -
- - - - - C - -
- - - - - A - -
- - - - - R - -
- - - - - F - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

Elapsed Time for this program is 508 ms
Total comparison is 3324 time(s)

```

4. file medium1.txt

input:

C R I N G E E K K R E S C I N D L M L E
I O R R E H N S H N E T S A H E A A H L
Y R E N V F S U E D T W B J H Q C Z E B
H E S M E A C O A G W Y V T T H E O D I
A T U O I R O I D E R L I E R A R B O S
U S R N R O N M S Z V L D Y B M A F N U
G U T E P H C E T M B I M S I Z T U I A
H L S V E T E T R L U O T S L I I S S L
T K B N R E R S O M S A E C K E O C M P
I C A E M L L B N E I J M S E R N A B L
N A N R Y P H A G N H A P S M J O T B A
E L E L B A T I D E R C E Y C F B E A C
S F N A M S I L A T P Z R N G H D O B E
S E P G B L U N D E R B U S S N U C R R
J L G E N U N C I A T I O N O W R I A A
V B L A T A N T E T A R U D B O A N S T
N A C R E D U L O U S T F G Q N T O I I
L I L Q X P I B L I G H T E D E E C V O
U L C D R D U O R H S N E S I R U A E N
O P N R A L U C S U P E R C J J U L M C

ABRASIVE
ENSCONCE
OBDURATE
ABSTAIN
ENSHROUD
OBFUSCATE
ABSTEMIOUS
ENUNCIATION
OBJECTIVE
ABSTRUSE
ENVENOM
PLAUSIBLE
BILK
HASTEN
PLETHORA
BLATANT
HAUGHTINESS
PLIABLE
BLIGHTED
HEADSTRONG
RENOWN
BLITHE
HEDONISM
REPRIEVE
BLUNDERBUSS
LACERATION
RESCIND
CREDITABLE
LACERATIONOBDURATE
TALISMAN
CREDULOUS
LACHRYMOSE
TEDIUM
CREPUSCULAR
LACKLUSTER
TEMPER
CRINGE
LACONIC

output:

```
Enter file : test/medium1.txt
```

== Word ABRASIVE ==

== Word ENSCONCE ==

== Word OBDURATE ==

sampai

[illegible]

5. medium2.txt

input:

C I F I C A P H T U O S D W Y I S A X D
R E R U S A E R T K N B P R S E I D A L
R D N E W S L E T T E R A M U S K Q F E
Y E M A R G O R P E S T S R E T E E R G
N A N S T N E D I S E R P E C I V S I P
T R D N H A D X M R L I O E N P L E E I
W G O S I U R I C X U P C G G R S L N H
R O H N E W L E V V R S E E F A Y Y D S
A E P S E N S D E T A D T S O P S T S R
V R H R H V D K N E V E N T S E H S H E
A E Q P E T N E D I S E R P T S A P E B
P G F S A S H O W H L C W W X L Q S D M
U U R N G R I W C Z W C L A M Q L N L E
B L E M G N G D H I S T O R I A N O A M
L A W E N K I O E K K P W F G R A I N S
I T O H I I W T T N W N E M F G V T O E
C I L T T F E T E O T U H A P E V I I C
I O F N I G N I R E H T A G K T E S T O
T N K A R S Q S O O M P A S Z E I O A N
Y S D H W E N O T S H T R I B X R P N D

ANTHEM
MEETINGS
RULES
BIRTHSTONE
MEMBERSHIP
SECOND
COFFEE
MESSAGE
SECRETARY
CONVENOR
NATIONAL
SOUTHPACIFIC
DUES
NEWSLETTER
SPEAKER
EVENTS
PASTPRESIDENT
STYLES
FEES
PHOTOGRAPHER
TREASURER
FLOWER
POSITIONS
TRIPS
FRIENDS
POSTDATED
VICEPRESIDENT
GATHERING
PRESIDENT
WEDNESDAY
GREETERS
PROGRAM
WINNER
HISTORIAN
PUBLICITY
WRITING
LADIES
REGULATIONS

output:

```
Enter file : test/medium2.txt
```

== Word ANTHEM ==

== Word MEETINGS ==

== Word RULES ==

== Word BIRTHSTONE ==

== Word MEMBERSHIP ==

== Word SECOND ==

sampai

```
+==+ Word REGULATIONS +==+
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- R - - - - -
- E - - - - -
- G - - - - -
- U - - - - -
- L - - - - -
- A - - - - -
- T - - - - -
- I - - - - -
- O - - - - -
- N - - - - -
- S - - - - -
```

```
Elapsed Time for this program is 1146 ms
Total comparison is 11270 time(s)
```

6. medium3.txt

input:

E W I S T N E M U G R A U N D E R S T A N D
O S S O R X W W R U N R A C E S Y M V Y F E
Q O I P C D T P E D E T A R T S U R F O A D
L V T O L O Y T L I E E R G A S I D P W H U
Y R O J X A M C B E A W D O C B M O T A T M
S T K C W X S P A G U A R G U E B Y P H O P
N Y A L F F E H R C A L M L Y E X P L A I N
H S Y E A F Q E E O P B A G S K Y L T S R W
U H T V V T E U P D M H R T T Z A D E N E S
Q D O U Z L Y B E D R I F P U A I L S I N P
D A A S O J O D S S X R S T T F D R V O Z E
Z F R C W P R S N M I F D E F D E F I V T U
V T G A S F M R I E O G B E U V O N E H I D
X M U R Y E H A N Y E P R P O T I B G X K I
J P E Y M E K D C T L E D P N P B I W C L H
Z C T S I I S I H K N U E I O D F E U V L U
I L V T H A C E B T M E O L R G E T K R J Y
L I Y O J H R M I E L P X S I K R P Y I Q K
A S T R T Q S D L S D E A B K U R R M F M G
D T B I F W E A W U V I R H H R K A N O R U
Z E A E Z A I H E J T H R L K V V W M G T J
Q N U S S K U P S E T R N U X L Y R G N A S

AGREE
FRUSTRATED
RIDE BIKES
ANGRY
HAPPY
RUN RACES
ARGUE
HURT
SCARY STORIES
ARGUMENTS
INSEPERABLE
SLEEPOVERS
BEST FRIENDS
SOLVE
BIG FIGHT
LISTEN
SPLASHED
CALMLY EXPLAIN
MARK
STOMPED
CAMP OUTS
MIKE
TALK
COMPROMISE
MUD PUDDLES
TOGETHER
DIFFERENT IDEAS
OPINIONS
UNDERSTAND
DISAGREE
UPSET

output:

```
Enter file : test/medium3.txt
```

== Word AGREE ==

== Word FRUSTRATED ==

```

+==+ Word RIDEBIKES +==+

```

== Word ANGRY ==

sampai


```
+== Word DISAGREE +==  
- - - - - E E R G A S I D - - - - -  
  
+== Word UPSET +==  
- - - - - U P S E T - - - - -
```

Elapsed Time for this program is 1195 ms
Total comparison is 10672 time(s)

7. large1.txt

input:

YQKGELECTROENCEPHALOGrameeBSJAKPBNQ
SECNEGILLETNIRETNUOCENVLTUFESEOVHJQ
NGNIZILANOITUTITSNIEDLEDYBZNNZDBRPG
JQNHNCIHPARGONITERORTCELELQCOMQGYDT
SESSENSUOENAROPMETXETJRZEVOITDCNADQ
SLKIESTSIGOLOISYHPORTCELEMEATEOYZKX
YEUUVKMZYZUKCSCJPVOVCQXGPTYYPNLSBN
GCCGHPFCKSSSMFBXSPXOJEBAZLJSZAVLNT
ETWNZOHSANBEJCEPHJURKKRALHEXIREAODP
LRJIEINSEXOQCWRSUNCSENTYAQNVELTNCIIS
EOETPCBBPDIWZNYRTPEOMCCFIKXEIMTITSI
CPKSLXSLCIIIZVSEEFSLLELIYMSPCNBEITATN
THMIVZJEFRTICRRSINNTYADRZGHONOUZIW
ROKDOGNINJYOODOEETSSSYSTECDRHMTNEINH
OTLARAYSXILSEENUAFTIOASECHJCRAAPLGS
OOMRMAMZCOMMTLELNLRSRDSZOTCJIELLAAUN
CGPTPWASGHOULAICATTNIDUZFOMOTIIRIO
URVNZLNIFNLBZLIUIEOANBQSALYNZZERSI
LAVOPSCJSFIOAOTLNNNRTREXLFQWUAAHTHT
OPICTATTGSLTRNRLLOERNTYPVVBOTTTSA
GHZVLYRKNOIGEYOYTEGRBWEBRKOJHCIIOUBI
RIAQWAFEXOBTGHFSCDRAIJVCEPGBKOOMDIT
AEPHTIHINYSENTLSLEETIAHRIOSTPQKNNENLN
PSZONEMEMINEFNMOWLWPSYOLUNJZSSSHIE
HVRGRWEYXTMREYOIKUYEMHIXZURUBVLCETR
ISEPHGLEVIUSJNQTWRRHFHIYOJTGOYJXDIE
EVMHQMCRDNSRSAXGHLYOXPLCOEHICCAQIEF
SOMBCHYKPEUTPHOUQHSSMODEAWDYOOSTXS
FCOUNTERSURVEILLANCESEGTYLXUXNRUOHI
OXDEYSXQOAVMWQYYHDZZXQTWOHLILVAWYED
VWNOITARTSNOMEDRETNUOCVHISXYLLYRATO
GIXJWOZIFNDEHYDROCHLORINATIONSSGYXT
AUZABONSTITDRACOYMO LAHPECNESPWULZKY
PZTIEGELECTROCARDIOGRAPHICERCPTKGNC
JAPROQSGNINOITIDNOCRETNUOCASZISQIIN

COMPREHENSIBLENESSES
CRYSTALLOGRAPHICALLY
ELECTROOCULOGRAPHIES
CONTRADISTINGUISHING
CYTODIFFERENTIATIONS
ELECTROPHOTOGRAPHIES
CONVENTIONALIZATIONS
DEHYDROCHLORINATIONS
ELECTROPHYSIOLOGICAL
COUNTERCONDITIONINGS
DEINDUSTRIALIZATIONS
ELECTROPHYSIOLOGISTS
COUNTERDEMONSTRATING
DEINSTITUTIONALIZING
ELECTRORETINOGRAPHIC
COUNTERDEMONSTRATION
DEOXYRIBONUCLEOTIDES
ENCEPHALOMYOCARDITIS
COUNTERDEMONSTRATORS
DEPARTMENTALIZATIONS
EXISTENTIALISTICALLY
COUNTERINTELLIGENCES
DIMETHYLNITROSAMINES
EXPRESSIONLESSNESSES
COUNTERMOBILIZATIONS
DISTINGUISHABILITIES
EXTEMPORANEOUSNESSES

output:

```
Enter file : test/large1.txt
```

== Word COMPREHENSIBLENESSES ==

S
 E
 S
 E
 N
 E
 L
 B
 I
 S
 N
 E
 H
 E
 R
 P
 M
 O
 C

==> Word CRYSTALLOGRAPHICALLY ==>

C

R

Y

S

T

A

L

L

O

G

R

A

P

H

I

C

A

L

L

Y

E

[illegible]

sampai

[illegible]

```
Elapsed Time for this program is 2516 ms
Total comparison is 22054 time(s)
```

8. large2.txt

input:

R Q C B Z B P K F S W E D E N B O R G I A N I S M P T S H Q U S A
 E R L A C I R T E M I R A H C C A S G Y H X S J T A P E S E S Q M
 T G N O I T A Z I N I N H C Y R T S H N G Z K C D Z Z S U C I T G
 E T Y I Y L L A C I P O C S O E R E T S M U A F H K U Q L N L H L
 M B U D Z X P U S Y L S Y N C A T E G O R E M A T I C U P E I G J
 O N T A M S M Q S L S S E N S U O I L I C R E P U S G I H C C U S
 L O L C T A R J F L G E Y F I N Q N X C Z Q F H D V X C A S I O U
 O I R Y K G R B S A D S M S Z X E S O W L X Q U O A H E N E C N B
 B T S L G D B G D N T S H I B Y F S U I X O A Q L O X N T R A D C
 O A C A B E S J O O R E U S V S T O U B L A M Q Y Z I T I C L A A
 R D N C K T N U L I E N I B S E E I F P C L N V A B K E M X C E R
 T N T I I E O S T S L D L W C A R T L G E A U D T V P N O E A R B
 C U O R X R I P M A I E T E H O F T T A A R R G B S D N N R R D O
 E C S T A U T H D C O D H L R K M H I T N S I T E Y G I I E E R N
 P E S E A H A Y J C Y N A O H Z J M S C A O L N I D V A O P O E I
 S F E M I P C G Y O U I D F R M F O I C I S I S S L R L U U U P F
 L R N O M L I M G I N M V D O T V X C S P L I R S T A E S S S U E
 Y E S I E U F O G M K E K E I M C H T E S P L U T N I G B N R S R
 L P U H C S I M A E C L C C Y R A E C U H I P A E N K T I B P P O
 L U O C C R B A U S P P L U K R R T P O A E O Y T V E X U N U I U
 A S E I O E A N N D K M H B I E R E N S R X P N Q E T T E T O L S
 C J N O C P L O N G Q I P F O O R O N C J K V K E A W F P Q I U S
 I V A T O U L M B L T S I P C S S K O U D I H K O R W J H E D O S
 P K T S L S Y E F S W C H H E T R N L M A N Q S H E S G I Q S N N
 O W L G Y I S T O U A O E R O G S X L A S V G F M M P A B Z I L T
 C J U B H M R R T T N M V M S E I G O L O T A M O T P M Y S H P G
 S H M S P E T Y I I I I A B Q B W G A D M D Z V H M I V V I A O M
 O L I Z A T M O C S C T N U G M N A B L G X X O H I S K B W L A C
 B M S Y T I N A T E O Z E S C L E R E N C H Y M A T O U S W T R K
 O S Z J S U L R A U Q N H B V G G I J N K B Z K P F T X K D L H H
 R K Z V I L Y B S Y C O Y Z E R Q Y K F P Y M T C T Q A R K I R Z
 T P N Q Y K L E X E G Q B R E T E M O N A M O M G Y H P S Q E O J
 S L S Z B E R S K B N T C F C S E M I P E R M E A B I L T Y O Q M

|
 SUPEREXCRESCENCE
 SEMIPERMEABILITY
 STAPHYLOCOCCEMIA
 SUPERFECUNDATION
 SEMIVERTICILLATE
 STEREOPHONICALLY
 SUPERINSTITUTION
 SEPTENTRIONALITY
 STEREOSCOPICALLY
 SUPERSERVICEABLE
 SESQUICENTENNIAL
 STOICHIOMETRICAL
 SUPERSULPHURETED
 SILICICALCAREOUS
 STROBOSCOPICALLY
 SWEDENBORGIANISM
 SIMPLEMENTEDNESS
 STRYCHNINIZATION
 SYLLABIFICATIONS
 SIMULTANEOUSNESS
 SUBCARBONIFEROUS
 SYMPTOMATOLOGIES
 SIPHONOSTOMATOUS
 SUBCARTILAGINOUS
 SYNCATEGOREMATIC
 SLUBBERDEGULLION
 SUBCOMMISSIONERS
 SPECTROBOLOMETER
 SULPHANTIMONIOUS

output:

[illegible]

== Word SEMIPERMEABILITY ==

[illegible]

sampai

== Word SULPHANTIMONIOUS ==

S
 U
 L
 P
 H
 A
 N
 T
 I
 M
 O
 N
 I
 O
 U
 S

```
Elapsed Time for this program is 2464 ms
Total comparison is 24863 time(s)
```

BAB 4

***GitHub* program**

Untuk program saya, bisa dilihat di link berikut:

https://github.com/Enderageous/Tucil1_13520012

Lampiran

| 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. | √ | |