**LAPORAN TUGAS KECIL 1**

Mata Kuliah IF2211 Strategi Algoritma

Calendar

Description automatically generated

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 ditandakan 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 tersebbut 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:

Text

Description automatically generated with medium confidence

output:

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text

Description automatically generated with medium confidence

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text

Description automatically generated

Text

Description automatically generated with medium confidence

**2. small2.txt**

input:

A picture containing text

Description automatically generated

output:

A picture containing calendar

Description automatically generated

Text

Description automatically generated with medium confidence

A screen shot of a computer

Description automatically generated with low confidence

A screen shot of a computer

Description automatically generated with low confidence

A picture containing text

Description automatically generated

**3. small3.txt**

input:

Text

Description automatically generated

output:

A screen shot of a computer

Description automatically generated with low confidence

A picture containing graphical user interface

Description automatically generated

Text

Description automatically generated with low confidence

A screen shot of a computer

Description automatically generated with low confidence

A screen shot of a computer

Description automatically generated with low confidence

A screen shot of a computer

Description automatically generated with low confidence

A screen shot of a computer

Description automatically generated with low confidence

Text

Description automatically generated with low confidence

**4. file medium1.txt**

input:

Text

Description automatically generated

output:

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated with low confidence

A picture containing text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

sampai

Text

Description automatically generated with low confidence

**5. medium2.txt**

input:

Text

Description automatically generated

output:

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screen shot of a computer

Description automatically generated with low confidence

sampai

Text

Description automatically generated with medium confidence

**6. medium3.txt**

input:

Text

Description automatically generated

output:

Text

Description automatically generated with medium confidence

A picture containing text

Description automatically generated

A picture containing text

Description automatically generated

sampai

A picture containing text

Description automatically generated

**7. large1.txt**

input:

A picture containing background pattern

Description automatically generated

output:

A screenshot of a computer

Description automatically generated with low confidence

A picture containing text

Description automatically generated

Background pattern

Description automatically generated

A picture containing text, appliance, kitchen appliance

Description automatically generated

sampai

Background pattern

Description automatically generated

**8. large2.txt**

input:

A picture containing text

Description automatically generated

output:

Background pattern

Description automatically generated

Background pattern

Description automatically generated

Background pattern

Description automatically generated with low confidence

Background pattern

Description automatically generated with medium confidence

sampai

Background pattern

Description automatically generated

# 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. | √ |  |