

**LAPORAN TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA**  
***WORD SEARCH PUZZLE***



**PUTRI NURHALIZA**  
**13520066**

**TEKNIK INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**SEMESTER II 2021/202**

## Daftar Isi

Daftar Isi .....	i
A. Deskripsi Algoritma <i>Brute Force</i> .....	1
B. <i>Source Code</i> Program (Bahasa Pemograman C++) .....	3
1. <i>Parsing</i> File ke <i>Class</i> .....	3
2. <i>Puzzle</i> .....	4
3. <i>Words</i> .....	7
4. <i>Search</i> .....	8
5. <i>Main</i> .....	14
C. Input dan Output Hasil Eksekusi .....	16
D. <i>Repository</i> Program (Github) .....	31
E. Rangkuman Keberjalanan Program .....	31

## A. Deskripsi Algoritma *Brute Force*

Penyelesaian *Word Search Puzzle* diimplementasi kan menggunakan algoritma *brute force* dengan langkah-langkah sebagai berikut.

1. Telusuri setiap elemen matriks secara traversal dengan iterasi baris dan kolom untuk mengetahui huruf yang ada pada posisi tersebut. Misal, matriks memiliki m baris dan n kolom. Kita dapat menempatkan posisi mulai dari matriks[0][0] dan berpindah ke posisi selanjutnya hingga matriks[m-1][n-1] jika kata belum ditemukan.
2. Cek apakah huruf pada posisi sekarang sama dengan huruf pertama pada kata yang akan dicari.
3. Jika sama, cek huruf kedua pada posisi setelahnya berdasarkan salah satu arah mata angin. Misal pada arah horizontal ke kanan, akan dicek apakah huruf kedua pada kata sama dengan huruf yang ada di kanan posisi sekarang.
  - Jika masih sama, maka teruskan penelusuran pada arah tersebut hingga huruf terakhir pada kata.
  - Jika ditemukan ketidaksesuaian, maka penelusuran pada arah tersebut dihentikan.

Lalu lakukan hal yang sama pada arah yang lain (horizontal ke kiri, vertikal ke atas, vertikal ke bawah, diagonal ke kanan atas, diagonal ke kanan bawah, diagonal ke kiri atas, dan diagonal ke kiri bawah) untuk mengecek kata dari awal. Jika kedelapan arah mata angin sudah ditelusuri namun kata belum ditemukan, maka ulangi dari poin 2 pada posisi selanjutnya di matriks.

4. Jika huruf pada posisi sekarang di matriks tidak sesuai dengan huruf pertama pada kata yang akan dicari, maka pindah ke posisi selanjutnya di matriks dan ulangi poin 2. Memungkinkan mencapai indeks terakhir pada matriks.

Singkatnya,

- Berhenti jika terdapat ketidaksesuaian dalam penelusuran alur tersebut. Lanjutkan pengecekan dari awal kata pada alur penelusuran yang lain.
- Berhenti mencari begitu kata sudah berhasil ditemukan.

Kita dapat memperkirakan apakah penelusuran akan *out of bounds* dengan memperhitungkan panjang kata dan posisi huruf pertama pada matriks untuk melihat apakah akan cukup atau tidak pada arah tersebut. Jika tidak, kita tidak perlu melakukan penelusuran hingga *out of bounds*.

Berikut adalah ilustrasinya, misal kita ingin mencari kata “stima”.

Cari huruf pertama yang sesuai

d	a	s	r	c	v	r	d	a	s	r	c	v	r
y	d	m	l	i	a	f	y	d	m	l	i	a	f
q	w	d	i	a	x	b	q	w	d	i	a	x	b
f	g	h	d	t	x	m	f	g	h	d	t	x	m
d	h	g	t	q	s	n	d	h	g	t	q	s	n

“d” tidak sesuai, geser kotak biru ke “a” namun juga tidak sesuai, lalu geser ke “s”.

```
d a s r c v r   d a s r c v r
y d m l i a f   y d m l i a f
q w d i a x b   q w d i a x b
f g h d t x m   f g h d t x m
d h g t q s n   d h g t q s n
```

Ditemukan posisi huruf pertama yang sesuai. Lalu cek ke 8 arah yang mungkin untuk huruf kedua yaitu “t”, namun tidak ada yang sesuai. Pindah kotak biru ke posisi berikutnya, hingga akhirnya mencapai “s” lagi.

```
d a s r c v r
y d m l i a f
q w d i a x b
f g h d t x m
d h g t q s n
```

Ditemukan “s”, lalu cek ke 8 arah yang mungkin dengan kotak hijau.

```
d a s r c v r   d a s r c v r
y d m l i a f   y d m l i a f
q w d i a x b   q w d i a x b
f g h d t x m   f g h d t x m
d h g t q s n   d h g t q s n
```

Pada arah horizontal ke kiri dan vertikal ke atas ditemukan huruf yang tidak sesuai dengan huruf kedua, pada 5 arah lain *out of bounds*. Sementara pada arah diagonal ke kiri atas,

```
d a s r c v r   d a s r c v r   d a s r c v r   d a s r c v r
y d m l i a f   y d m l i a f   y d m l i a f   y d m l i a f
q w d i a x b   q w d i a x b   q w d i a x b   q w d i a x b
f g h d t x m   f g h d t x m   f g h d t x m   f g h d t x m
d h g t q s n   d h g t q s n   d h g t q s n   d h g t q s n
```

Ditemukan kesesuaian mulai dari huruf kedua hingga akhir. Sehingga kata ditemukan di indeks matriks[4][5] dengan arah diagonal ke kiri atas.

## B. Source Code Program (Bahasa Pemograman C++)

Source code program dibuat secara modular, terdiri atas empat kelompok (*header* dan *driver*) serta satu file *main*.

### 1. Parsing File ke Class

Berfungsi untuk mem-*parsing* inputan dari file (.txt) dan menyimpannya ke struktur data *Puzzle* dan *Words*.

fileToData.h

```
#ifndef __FILETODATA_H
#define __FILETODATA_H

#include "puzzle.h"
#include "words.h"

void fileToData(string filename, Puzzle *p, Words *w);

#endif
```

fileToData.cpp

```
#include <iostream>
#include <stdlib.h>
#include <string>
#include <bits/stdc++.h>
#include "fileToData.h"
using namespace std;

void fileToData(string filename, Puzzle *p, Words *w)
{
    ifstream file(filename.c_str());
    string line;
    char letter;
    bool flag = false;
    int rows = 0, cols = 0, nWord=0;
    int i, j=0;

    // parsing puzzle baris pertama untuk menemukan banyak kolom
    getline(file, line);
    istringstream strline(line);
    while (strline>>letter)
    {
        p->setElmt(rows, cols, letter);
        cols++;
    }
    rows++;
```

```

// parsing puzzle baris kedua sampai akhir puzzle
while (!flag) {
    getline(file, line);
    istringstream strline(line);
    rows++;
    while (strline>>letter)
    {
        if (j >= cols) j = 0;
        i = rows-1;
        p->setElmt(i, j, letter);
        j++;
    }
    if (file.peek() == '\n') flag = true;    // parse puzzle done
}

// untuk ngeskip blank line pemisah puzzle dan words
getline(file, line);

// parsing words yang mau dicari
while (getline(file, line)) {
    // cout << line << endl;
    w->setWord(nWord, line);
    nWord++;
}

p->cols = cols;
p->rows = rows;
w->total = nWord;
}

```

## 2. Puzzle

Berisi atribut serta *method* yang diperlukan pada *puzzle* (matriks)

puzzle.h

```

#ifndef __PUZZLE_H
#define __PUZZLE_H

#include <string>
using namespace std;

class Puzzle {
public:
    int rows, cols;    // jumlah baris dan kolom efektif
    char ** matrix;    // matrix untuk menyimpan puzzle

    //constructor

```

```

    Puzzle();
    Puzzle(int x, int y);

    //setter isi Puzzle
    void setElmt(int x, int y, char val);
    void setDefault();
    void setResult(int frow, int fcol, string found, string w);

    //getter
    char getElmt(int x, int y);

    //display
    void display();
};

#endif

```

puzzle.cpp

```

#include <iostream>
#include <stdlib.h>
#include <string>
#include <bits/stdc++.h>
#include "puzzle.h"
using namespace std;

Puzzle::Puzzle()
{
    rows = 0;
    cols = 0;
}

Puzzle::Puzzle(int x, int y)
{
    rows = x;
    cols = y;
    matrix = new char *[rows];
    for (int i=0; i<rows; i++)
    {
        matrix[i] = new char[cols];
    }
}

void Puzzle::setElmt(int x, int y, char val)
{
    matrix[x][y] = val;
}

void Puzzle::setDefault()

```

```

{
    for (int i=0; i<rows; i++)
    {
        for (int j=0; j<cols; j++)
        {
            setElmt(i,j,'-');
        }
    }
}

char Puzzle::getElmt(int x, int y)
{
    return matrix[x][y];
}

void Puzzle::display()
{
    for (int i=0; i<rows; i++)
    {
        for (int j=0; j<cols; j++) cout << matrix[i][j] << " ";
        cout << endl;
    }
}

// SET BERDASAR HASIL PENCARIAN
void Puzzle::setResult(int frow, int fcol, string found, string w)
{
    setDefault();

    if (found == "LR") for (int j=fcol; j<fcol+w.length(); j++)
setElmt(frow,j,w[j-fcol]);

    else if (found == "RL") for (int j=fcol-w.length()+1; j<fcol+1; j++)
setElmt(frow,j,w[fcol-j]);

    else if (found == "UD") for (int i=frow; i<frow+w.length(); i++)
setElmt(i,fcol,w[i-frow]);

    else if (found == "DU") for (int i=frow-w.length()+1; i<frow+1; i++)
setElmt(i,fcol,w[frow-i]);

    else if (found == "ULDR") for (int i=frow; i<frow+w.length(); i++)
setElmt(i,fcol-frow+i,w[i-frow]);

    else if (found == "DRUL") for (int j=fcol-w.length()+1; j<fcol+1; j++)
setElmt(frow-fcol+j,j,w[fcol-j]);

```



```

        else if (found == "URDL") for (int i=frow; i<frow+w.length(); i++)
setElmt(i,fcol+frow-i, w[i-frow]);

        else if (found == "DLUR") for (int j=fcol; j<fcol+w.length(); j++)
setElmt(frow+fcol-j,j,w[j-fcol]);

        else cout << "\nKata tidak ditemukan pada puzzle";
}

```

### 3. Words

Berisi atribut dan *method* yang diperlukan pada *array* kata yang akan dicari.

words.h

```

#ifndef __WORD_H
#define __WORD_H

#include <string>
using namespace std;

class Words{
public:
    int total;
    string * words;

    //constructor
    Words();
    Words(int new_total);

    //setter
    void setWord(int i, string newWord);

    //getter
    string getWord(int i);

    //display
    void display();
};

#endif

```

words.cpp

```

#include <iostream>
#include <stdlib.h>
#include <string>
#include "words.h"
using namespace std;

```

```

Words::Words()
{
    total = 0;
}

Words::Words(int new_total)
{
    total = new_total;
    words = new string[total];
}

void Words::setWord(int i, string newWord)
{
    words[i] = newWord;
}

string Words::getWord(int i)
{
    return words[i];
}

void Words::display()
{
    for (int i=0; i<total; i++) cout << words[i] << endl;
}

```

#### 4. Search

Berisi atribut dan *method* yang diperlukan untuk algoritma *brute force* pencarian kata pada puzzle.

search.h

```

#ifndef __SEARCH_H
#define __SEARCH_H

#include <string>
#include "puzzle.h"
#include "words.h"
using namespace std;

class Search{
public:
    string found;           // "none" jika kata belum ditemukan, indikasi arah
    // jika kata sudah ditemukan
    int frow,fcol;         // indeks penunjuk huruf pertama kata yang ditemukan
    // pada matrix

```

```

        int count;           // jumlah total perbandingan huruf untuk pencarian 1
kata pada puzzle

        //constructor
        Search();

        void find(Puzzle *p, string w);
        void horizontal_LR(Puzzle *p, string w, int cRow, int cCol);
        void horizontal_RL(Puzzle *p, string w, int cRow, int cCol);
        void vertical_UD(Puzzle *p, string w, int cRow, int cCol);
        void vertical_DU(Puzzle *p, string w, int cRow, int cCol);
        void diagonal_ULDR(Puzzle *p, string w, int cRow, int cCol);
        void diagonal_DRUL(Puzzle *p, string w, int cRow, int cCol);
        void diagonal_URDL(Puzzle *p, string w, int cRow, int cCol);
        void diagonal_DLUR(Puzzle *p, string w, int cRow, int cCol);
};

#endif

```

### search.cpp

```

#include <iostream>
#include <stdlib.h>
#include <string>
#include "puzzle.h"
#include "words.h"
#include "search.h"
#include <bits/stdc++.h>
using namespace std;

//constructor
Search::Search()
{
    found = "none";
    count = 0;
}

// ALGORITMA BRUTEFORCE TRAVERSAL MATRIX
// Untuk pencarian huruf pertama yang sesuai
void Search::find(Puzzle *p, string w)
{
    int i=0;
    int j;
    while (found == "none" && i<p->rows)
    {
        j=0;
        while (found == "none" && j<p->cols)
        {

```

```

        count++;
        if (p->getElmt(i,j) == w[0])
        {
            horizontal_LR(p, w, i, j);
            horizontal_RL(p, w, i, j);
            vertical_UD(p, w, i, j);
            vertical_DU(p, w, i, j);
            diagonal_ULDR(p, w, i, j);
            diagonal_DRUL(p, w, i, j);
            diagonal_URDL(p, w, i, j);
            diagonal_DLUR(p, w, i, j);
        }
        j++;
    }
    i++;
}
frow = i-1;
fcol = j-1;
}

// Jika ditemukan huruf pertama yang sesuai, bruteforce akan dilanjutkan ke arah
berikut

// Pencarian pada arah horizontal ke kanan
void Search::horizontal_LR(Puzzle *p, string w, int cRow, int cCol)
{
    // batasan tidak out of bounds
    if (cCol < p->cols - w.length() + 1)
    {
        int k=1;
        bool foundtmp = true;
        while (foundtmp && k<w.length())
        {
            count++;
            if (p->getElmt(cRow, cCol+k) == w[k])
            {
                if (k==w.length()-1) found = "LR";
            }
            else foundtmp = false;
            k++;
        }
    }
}

// Pencarian pada arah horizontal ke kiri
void Search::horizontal_RL(Puzzle *p, string w, int cRow, int cCol)
{
    // batasan tidak out of bounds

```

```

    if (cCol >= w.length() - 1)
    {
        int k=1;
        bool foundtmp = true;
        while (foundtmp && k<w.length())
        {
            count++;
            if (p->getElmt(cRow, cCol-k) == w[k])
            {
                if (k==w.length()-1) found = "RL";
            }
            else foundtmp = false;
            k++;
        }
    }
}

// Pencarian pada arah vertikal ke bawah
void Search::vertical_UD(Puzzle *p, string w, int cRow, int cCol)
{
    // batasan tidak out of bounds
    if (cRow < p->rows - w.length() + 1)
    {
        int k=1;
        bool foundtmp = true;
        while (foundtmp && k<w.length())
        {
            count++;
            if (p->getElmt(cRow+k, cCol) == w[k])
            {
                if (k==w.length()-1) found = "UD";
            }
            else foundtmp = false;
            k++;
        }
    }
}

// Pencarian pada arah vertikal ke atas
void Search::vertical_DU(Puzzle *p, string w, int cRow, int cCol)
{
    // batasan tidak out of bounds
    if (cRow >= w.length() - 1)
    {
        int k=1;
        bool foundtmp = true;
        while (foundtmp && k<w.length())
        {

```

```

        count++;
        if (p->getElmt(cRow-k, cCol) == w[k])
        {
            if (k==w.length()-1) found = "DU";
        }
        else foundtmp = false;
        k++;
    }
}

// Pencarian pada arah diagonal ke kanan bawah
void Search::diagonal_ULDR(Puzzle *p, string w, int cRow, int cCol)
{
    // batasan tidak out of bounds
    if ((cCol < p->cols - w.length() + 1) && (cRow < p->rows - w.length() + 1))
    {
        int k=1;
        bool foundtmp = true;
        while (foundtmp && k<w.length())
        {
            if (p->getElmt(cRow+k, cCol+k) == w[k])
            {
                count++;
                if (k==w.length()-1) found = "ULDR";
            }
            else foundtmp = false;
            k++;
        }
    }
}

// Pencarian pada arah diagonal ke kiri atas
void Search::diagonal_DRUL(Puzzle *p, string w, int cRow, int cCol)
{
    // batasan tidak out of bounds
    if ((cRow >= w.length() - 1) && (cCol >= w.length() - 1))
    {
        int k=1;
        bool foundtmp = true;
        while (foundtmp && k<w.length())
        {
            count++;
            if (p->getElmt(cRow-k, cCol-k) == w[k])
            {
                if (k==w.length()-1) found = "DRUL";
            }
            else foundtmp = false;
        }
    }
}

```

```

        k++;
    }
}

// Pencarian pada arah diagonal ke kiri bawah
void Search::diagonal_URDL(Puzzle *p, string w, int cRow, int cCol)
{
    // batasan tidak out of bounds
    if ((cRow < p->rows - w.length() + 1) && (cCol >= w.length() - 1))
    {
        int k=1;
        bool foundtmp = true;
        while (foundtmp && k<w.length())
        {
            count++;
            if (p->getElmt(cRow+k, cCol-k) == w[k])
            {
                if (k==w.length()-1) found = "URDL";
            }
            else foundtmp = false;
            k++;
        }
    }
}

// Pencarian pada arah diagonal ke kanan atas
void Search::diagonal_DLUR(Puzzle *p, string w, int cRow, int cCol)
{
    // batasan tidak out of bounds
    if ((cRow >= w.length() - 1) && (cCol < p->cols - w.length() + 1))
    {
        int k=1;
        bool foundtmp = true;
        while (foundtmp && k<w.length())
        {
            count++;
            if (p->getElmt(cRow-k, cCol+k) == w[k])
            {
                if (k==w.length()-1) found = "DLUR";
            }
            else foundtmp = false;
            k++;
        }
    }
}

```

## 5. Main

File utama pada program yang berfungsi untuk:

- Menerima inputan nama file
- Menggunakan *method* fileToData
- Melakukan pencarian pada setiap *keywords* di *puzzle* menggunakan algoritma *brute force*
- Menghitung waktu eksekusi algoritma *brute force*
- Menghitung total perbandingan huruf
- Menampilkan output

main.cpp

```
#include <iostream>
#include <stdlib.h>
#include <string>
#include <bits/stdc++.h>
#include "puzzle.h"
#include "words.h"
#include "search.h"
#include "fileToData.h"
using namespace std;

int main()
{
    Puzzle* puzzle1 = new Puzzle(50,50);
    Words* words1 = new Words(20);
    double durationTotal = 0;
    int countTotal = 0;

    string filename;
    cout << "Masukkan nama file: ";
    cin >> filename;
    filename = "../test/" + filename;
    fileToData(filename, puzzle1, words1);

    for (int i=0; i<words1->total; i++)
    {
        Search* s = new Search();
        string str = words1->getWord(i);
        Puzzle* out = new Puzzle(puzzle1->rows,puzzle1->cols);
        chrono::steady_clock sc;

        auto start = sc.now();

        // ALGORITMA BRUTEFORCE PENCARIAN KATA
        // menghasilkan indeks ditemukannya huruf pertama kata dan arah
        // yang kemudian akan di set ke matrix hasil
        s->find(puzzle1, str);
```



```

    out->setResult(s->frow,s->fcol, s->found, str);
    // cout << endl << s->frow << " " << s->fcol << " " << s->found;

    auto end = sc.now();

    cout << endl << str << endl;
    out->display();
    cout << "Perbandingan huruf: " << s->count << endl;

    auto duration = static_cast<std::chrono::duration<double>>(end-start);
    durationTotal += duration.count();
    countTotal += s->count;
}

cout << "\n-----\n";
cout << "Total durasi: " << durationTotal << " detik\n";
cout << "Total perbandingan huruf keseluruhan: " << countTotal << "\n";

string dummy;
cout << "\nPress anything to exit (then enter): ";
cin >> dummy;

return 0;
}

```

### C. Input dan Output Hasil Eksekusi

Berikut adalah hasil eksekusi 9 *test case* yang terdapat pada folder /test. Total durasi yang kurang stabil disebabkan kecepatan eksekusi yang juga dipengaruhi beberapa faktor lain saat program sedang berjalan, sehingga bisa saja didapatkan total durasi yang berbeda untuk *test case* yang sama.)

Masukkan nama file: small1.txt

1MEJOGFNNKISO  
2EDPRESABATAD  
3VYIENWKJKORI  
4IIWSRWYZREAF  
5TPZOWCJANUQF  
6ASOLQUEEIECUE  
7IKOUZLANEZUR  
8TCSTURCLTGGE  
9INGITTJJQATAZN  
10NMYOIOZRRLHGC  
11IPANPVIDEOSE  
12TANALYSISAKI  
13IJRPKDUPATMR  
14TJZZCTHROATE  

THROAT

RESOLUTION

INITIATIVE

PERCENTAGE

DIFFERENCE

VIDEO

ANALYSIS

TRAINER

WEALTH

DATABASE

THROAT

Perbandingan huruf: 207

RESOLUTION

Perbandingan huruf: 25

INITIATIVE

Perbandingan huruf: 148

PERCENTAGE

Perbandingan huruf: 26

DIFFERENCE

Perbandingan huruf: 38

VIDEO

Perbandingan huruf: 139

ANALYSIS

Perbandingan huruf: 169

TRAINER

Perbandingan huruf: 127

WEALTH

Perbandingan huruf: 70

DATABASE

Perbandingan huruf: 35

Total durasi: 0.000999 detik  
Total perbandingan huruf keseluruhan: 984

Masukkan nama file: small2.txt

18 HOTEL  
19 DATA  
20 EQUIPMENT  
21 REACTION  
22 OPINION  
23 BUYER  
24 DECISION  
25 STORY  
26 RESPONSE  
27 INITIATIVE  
28 INTRODUCTION

EQUIPMENT

Perbandingan huruf: 220

```

- - - - -
- - - - -
- - - - -
- T - - -
- N - - -
- E - - -
- M - - -
- P - - -
- I - - -
- U - - -
- Q - - -
- E - - -
- - - - -
Perbandingan huruf: 226
```

BUYER

Perbandingan huruf: 19

Perbandingan huruf: 158

## RESPONSE

[illegible]

```

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - R - -
- - E - -
- - S - -
- - P - -
- - O - -
- - N - -
- - S - -
- - E - -
Perbandingan huruf: 157
```

## INTRODUCTION

```

      N - - -
      O - - -
      I - - -
      T - - -
      C - - -
      U - - -
      D - - -
      O - - -
      R - - -
      T - - -
      N - - -
      I - - -

```

Perbandingan huruf: 187

17

small3.txt (18x16 15 kata)

Masukkan nama file: small3.txt

```
1 J C P T C X M X M G E S A C H I
2 C C F D E E R B U J S G C R W K
3 G K U P P W Z W L X N Z M K T P
4 U V D I V E R T E Z O G R N W J
5 E U N I T N O C J W P A U T H K
6 F E A T U R E A L L S A C W A T
7 P W I A Y N O C Q N E I N A D P
8 H F P J I O Z C G E R D I C D Z
9 R J G W F I P U T O T T W C E V
10 O T H C G T V M K O K S U V V J
11 T C M G F C B U Y D G E I I I G
12 I E A A S U J L F S G T T Z R F
13 N F I V K D T A K J A Z T O R T
14 O F G G B O F T G I N G H U A T
15 M E X F W R Y E T D S O T K T F
16 J S T A R T P I S S P I L L W N
17 W E Z K W N N P K I D C V R V O
18 U V O U H I C X F B H R Z D J I
19
20 CONTINUE
21 EFFECT
22 INCUR
23 DIVERT
24 SPILL
25 ACCUMULATE
26 MONITOR
27 ARRIVE
28 AID
29 FEATURE
30 BREED
31 START
32 RESPONSE
33 INITIATIVE
34 INTRODUCTION
```

CONTINUE

E U N I T N O C

Perbandingan huruf: 96

EFFECT

T  
C  
E  
F  
F  
E

Perbandingan huruf: 296

INCUR

R  
U  
C  
N  
I

Perbandingan huruf: 162

DIVERT

D I V E R T

Perbandingan huruf: 60

SPILL

S P I L L

Perbandingan huruf: 308

ACCUMULATE

A  
C  
C  
U  
M  
U  
L  
A  
T  
E

Perbandingan huruf: 106

MONITOR

R  
O  
T  
I  
N  
O  
M

Perbandingan huruf: 255

ARRIVE

E  
V  
I  
R  
R  
A

Perbandingan huruf: 288

AID

A  
I  
D

Perbandingan huruf: 127

FEATURE

F E A T U R E

Perbandingan huruf: 90

BREED

D E E R B

Perbandingan huruf: 31

START

S T A R T

Perbandingan huruf: 291

RESPONSE

E

S

N

O

P

S

E

R

Perbandingan huruf: 148

INITIATIVE

E

V

I

T

A

I

T

I

N

I

Perbandingan huruf: 334

INTRODUCTION

N

O

I

T

C

U

D

O

R

T

N

I

Perbandingan huruf: 318

Total durasi: 0.002001 detik

Total perbandingan huruf keseluruhan: 2910

medium1.txt (20x18 13 kata)

Masukkan nama file: medium1.txt

1TYQNJJKOAUVVOWXOOE

2SUBSTANTIALDBSDMZM

3CMBSDZHHWOZIFYKDSO

4ISMDFWGCUDUQODGLFLS

5GMXZNPNUZOIZGXSNPH

6LCOSWELTERINGBWZCT

7AYWTGRJRSRVYFVMMWCO

8TBBGKFJNECESSARYOO

9SVKRHEHWWKHFCSDWAFT

10OHLKBBCPSPWWHITEDIC

11NTBWKTAHJKNSZTONOY

12SEVLDFCGIVXIRIARVL

13DSSWCUEVORWBNUZRBX

14ZTPMPFSZGVMAMFPZRI

15RYDVBXBELLIGERENTE

16IHVSCINTILLATINGJT

17BYOBDNYEAZPGMFYHEO

18OHDXGVVVPYJNYDOOLB

19UTC SJCKOKQKORVGCQGR

20SZYP PINMZBRWCTKMQW

21

22TOOTH SOME

23SWELTERING

24TESTY

25BLOODY

26NECESSARY

27PERFECT

28NOSTALGIC

29WHITE

30BELLIGERENT

31SUBSTANTIAL

32SCINTILLATING

33NIPPY

34ERRATIC

TOOTH SOME

E

M

O

S

H

T

O

O

T

Perbandingan huruf: 193

SWELTERING

E

V

I

T

A

I

T

I

N

I

Perbandingan huruf: 125

TESTY

T

E

S

T

Y

Perbandingan huruf: 233

BLOODY

E

V

I

T

A

I

T

I

N

I

Perbandingan huruf: 391

NECESSARY

E

V

I

T

A

I

T

I

N

I

Perbandingan huruf: 158

PERFECT

P

E

R

F

E

C

T

Perbandingan huruf: 85

NOSTALGIC

C

I

G

L

A

T

S

O

N

Perbandingan huruf: 208

WHITE

E

V

I

T

A

I

T

I

N

I

Perbandingan huruf: 247

BELLIGERENT

E

V

I

T

A

I

T

I

N

I

Perbandingan huruf: 299

19

The image shows a terminal window with a black background and white text. It displays a word search game interface. At the top, there are three word search grids. The first grid is for the word 'SUBSTANTIAL' and contains 30 letters. The second grid is for the word 'SCINTILLATING' and contains 330 letters. The third grid is for the word 'NIPPY' and contains 454 letters. Below these grids, there is a fourth grid for the word 'ERRATIC' which contains 337 letters. At the bottom of the terminal, there is a summary of the game: 'Total durasi: 0.002002 detik' and 'Total perbandingan huruf keseluruhan: 3090'.

```
Total durasi: 0.002002 detik
Total perbandingan huruf keseluruhan: 3090
```

Masukkan nama file: medium2.txt

ABANDONED

QUIZZICAL

LUDICROUS

ABANDONED

QUIZZICAL

LUDICROUS

ATTRACTIVE

INTERNAL

TRUCULENT

ABANDONED

QUIZZICAL

LUDICROUS

Perbandingan huruf: 165

Perbandingan huruf: 439

Perbandingan huruf: 144

Perbandingan huruf: 284

Perbandingan huruf: 351

Perbandingan huruf: 121

	<div><div>HYPNOTIC</div><div>C I T O N P Y H</div><div>Perbandingan huruf: 40</div></div>	<div><div>OMNISCIENT</div><div>O M N I S C I E N T</div><div>Perbandingan huruf: 120</div></div>	<div><div>DECOROUS</div><div>D E C O R O U S</div><div>Perbandingan huruf: 278</div></div>
	<div><div>HANDSOME</div><div>E M O S D N A H</div><div>Perbandingan huruf: 466</div></div>	<div><div>FREEZING</div><div>G N I Z E E R F</div><div>Perbandingan huruf: 275</div></div>	<div><div>DEPRESSED</div><div>D E P R E S S E D</div><div>Perbandingan huruf: 521</div></div>
	<div><div>RESOLUTE</div><div>E T U L O S E R</div><div>Perbandingan huruf: 335</div></div>	<div><div>Total durasi: 0.000999 detik</div><div>Total perbandingan huruf keseluruhan: 3539</div></div>	
	medium3.txt (24x22 14 kata)		
Masukkan nama file: medium3.txt	<div><div>AVAILABLE</div><div>E L B A L I A V A</div><div>Perbandingan huruf: 518</div></div>	<div><div>CHIVALROUS</div><div>C H I V A L R O U S</div><div>Perbandingan huruf: 241</div></div>	<div><div>BENEFICIAL</div><div>L A I C I F E N E B</div><div>Perbandingan huruf: 372</div></div>





Masukkan nama file: large1.txt

```

1  Q3RHRCPWGSDDSDSUWZHVRHLNHUHOEOUNZ
2  BANYQYLVRGLLSUJZQJZJZEHUBUPHTU
3  ISDRFFDLFKUOUOJXHXUSPKRLOULNAXKJ
4  PEPEYMRHEHDDPAZSRSNAXTFRRJSLRHS
5  YLZVAKTKOBTZQZKNKCLLIZRZUMMRVOX
6  QBORCJUNMTMDHCYVVRVZVAYPLZMTUHW
7  OVAEBHKGKSGVSKURJEMOJ3VNIPIVMXFLSR
8  QGRGHUQNWYHDXDUBDOUPOEWEGCTPVXHS
9  UORJ3J3J7YAFVEHENGLPMSIHEVQONF
10 EYI2JZJRHYNHEOGKNGRXNXPNNYIVSES
11 H2OXCHMDDEDOG3BTKEPERBMCOCVOMQOQ
12 CEVOE2PVELJZHFZPDDXVPECBTORKFH
13 GAEJ3JGRTWQEFQDHHVNRITB5MMHBLMMZM
14 MSJUMPORHTXRAVHRRN0QTAJNOKIGHMI
15 RKMVLDSERBZDS5PCJFRH5WRJXMXPSKRD
16 KOFBMTGBQCMANAPRRTCKXBKEUQOTMAE
17 GACSBURMTKTPVDCDBBKEFYCQDDICTFYZ
18 HTQCA0BUDHCPBFLNVTPEGISAEELRWV
19 XNSMLKNITPKGEFDELGTFLUYHRMNPZ
20 OFFYYVJUYBEUKWILNPAHAC0SUYEFCPZ
21 BGNMLWJBAYFZFNFKKBXFDHGDGSRCKFDF
22 VCATMBLTYTBRLGAJVLNFKYJ3NETEKC
23 RCRCRNWNXCS3PQACM3XKTE3PCZBBH
24 CEZENOFHKKFNIR0SYRHCAJLNUK0UGH
25 DVVXSQATISPFLYWNVEFRRTMAJHKB86G
26 TRRXZTQOFSKXTKTYGOEJBYPHCFQRQE
27 QANAPPYJLLPRIZFPVPYQDGKSCUPANQ
28 FGTQXNFLTMMFETDZTWWCSC0JSHMVXHTI
29 FLTXCQAWEQOQR0PUZQNYLGBQCIFYQVY
30 J0KFEQNOVDLT0E0VTQZNSADNLJXCX53
31 PTIOI0AGPYDGVVYUUSINLXZUKNDIXUFN
32 OSASAASKNHKPIWISKR55VKTPUOFTPO
33
34 UNRELEASABLE
35 RESTYLEY
36 SPONSAL
37 SOULPENCE
38 MOONIZE
39 SCROUNGED
40 COCAMA
41 PROSTRIKE
42 CALENDAL
43 FANGS
44 WHISPEROUS
45 ASCENDING
46 LIPOCERE
47 CHRYSORIN

```

RESTYLED

Perbandingan huruf: 286

Perbandingan huruf: 889

**SOULPENANCE**

Perbandingan huruf: 1019

Perbandingan huruf: 159

SCROUNGED

Perbandingan huruf: 488

Perbandingan huruf: 141

COCAMA

- A M A C O C

Perbandingan huruf: 726

PROSTRIKE

- P  
- R  
- O  
- S  
- T  
- R  
- I  
- K  
- E

Perbandingan huruf: 401

CALENDAR

Perbandingan huruf: 958

FANGS

- F
- A
- N
- G
- S

Perbandingan huruf: 1029

WHISPEROUS

Perbandingan huruf: 775

ASCENDING

Perbandingan huruf: 470



The image displays a 3x2 grid of word search puzzles. Each puzzle consists of a title, a grid of letters, and a word count at the bottom.

**Puzzle 1 (Top Left):** Title: WRETCHOCK. Grid: A 20x20 grid of letters. Word count: Perbandingan huruf: 1269.

**Puzzle 2 (Top Right):** Title: NONASSISTANCE. Grid: A 20x20 grid of letters. Word count: Perbandingan huruf: 1288.

**Puzzle 3 (Middle Left):** Title: TAXABILITY. Grid: A 20x20 grid of letters. Word count: Perbandingan huruf: 231.

**Puzzle 4 (Middle Right):** Title: FLAGELLATORY. Grid: A 20x20 grid of letters. Word count: Perbandingan huruf: 647.

**Puzzle 5 (Bottom Left):** Title: JAPERIES. Grid: A 20x20 grid of letters. Word count: Perbandingan huruf: 200.

**Puzzle 6 (Bottom Right):** Title: BILBY. Grid: A 20x20 grid of letters. Word count: Perbandingan huruf: 387.

HESPERIA

Perbandingan huruf: 376

UNMUTABLE

- U N M U T A B L E -

Perbandingan huruf: 817

INITIATORILY

Perbandingan huruf: 1135

REFULGENCY

Perbandingan huruf: 1026

**VULPECULA**

Perbandingan huruf: 305

VALLAR

Perbandingan huruf: 566

	<div>UNRULINESS</div> <div> <div>S</div> <div>S</div> <div>E</div> <div>N</div> <div>I</div> <div>L</div> <div>U</div> <div>R</div> <div>N</div> <div>U</div> </div> <div>Perbandingan huruf: 796</div>	<div>BASINERVED</div> <div> <div>DEVRENISAB</div> </div> <div>Perbandingan huruf: 1035</div>
--	---	--

-----

Total durasi: 0.003003 detik

Total perbandingan huruf keseluruhan: 10913

large3.txt (36x34 16 kata)

<div>Masukkan nama file: large3.txt</div> <div> <div> <div>1 EUJYVAACSAHKBSHEZUUSYNALIDFXYIRAE</div> <div>2 FLGMNFJCVGEWMLWGLKJUUVVGJWDKETKEPSK</div> <div>3 VTRDABAZPZQWHORTNBMVWHFIOXRLSSUCBD</div> <div>4 JVRIBTBQDHKMQYQMRVASRELSGCFRVNQVDK</div> <div>5 UCKARMEXIUGXWPTXGZPIHEXPQQFDBRXVQ</div> <div>6 DXUJBQITQYDAKSBNFHRPNDETCRDADHLLVB</div> <div>7 ZDEZJEUCCORWPLRMOOQEUHEMZXLDDJIFURKU</div> <div>8 YNOCKENLVHYVFJUNDYSKCDDDSQKSEXDI XGR</div> <div>9 NHSSPEWRGESARBJSAWEZQYZNCWMXCZYEQJ</div> <div>10 EXKCI GRPXXUKITYUBILRERSYUXPJTOCSWJ</div> <div>11 GTGNRPUEBQDEVTGBXDKJLITOTTSJMMHBVTX</div> <div>12 PMHMHVEBQCEWRRRESTILWJWCDMPXMYNPIAM</div> <div>13 YEPDOWNLRGGXMXAIZKBRPICGTWKBQAQNRRLG</div> <div>14 GMAXJZIBKDVVDVQKBDGKDIYDZSAAWCIZRRJR</div> <div>15 OJMTSDVSOUQJXYDIOTOVERMEEKLYGXHIIG</div> <div>16 LFXELGPBRWLHMMVWADVSLFSMVXJCXHZQGKZ</div> <div>17 ORTDJMTGRGLPOFWVRIZJKNAXJOXSHXHHVP</div> <div>18 TBQHIWNBQNIJGGQISARZATSJXAYJKEMTOU</div> <div>19 SMYIHAABFNBBVRZEEQYNQAUMTAIIOKGMSQX</div> <div>20 IMJXEOMLYHOIRMSVLUTEEMYIDXHVCHXKN</div> <div>21 RQZVKWYACXRZRPGGALMVVLOIGQYBXHIOL</div> <div>22 AKBKCLUDHRULMIWFDUCQZCZBSHIGOPPHM</div> <div>23 URULPUAUHQEBDMYJCVSFYINJAZCABEUUUU</div> <div>24 XPPPUFOYTTAPSGLSSTTOHSTDUQDQMHYLN</div> <div>25 MAYPHRHBNHKLXPXAKUANUYEAXZRXEOLJGE</div> <div>26 GWJRYUJVMUNKKAUMDSBKWHSRMYJAQVBOFV</div> <div>27 SUHERODIONINECELELZOHZLQJVRHGHCGVSX</div> <div>28 YOHCVEVNOJMYCTCFCKGXXJTLCDIPACYVVB</div> <div>29 NFVADJPCKNHXLGQLPWFFNNFBXIXFOFAIUVVB</div> <div>30 GMUYVTEHTPGDYJBEXCPLXCOTPRRDNXTBEW</div> <div>31 DYZAUZGUOEFCEBQUJAHCSLSKMEATKYINWAT</div> <div>32 GTBENTRMBUAQMHHGEAPJAAKNXYDESODZERC</div> <div>33 OKSRUQQMJYRSICOMUQBNDDDGQYETQEEYBZ</div> <div>34 IZCRSNF8GWJDDJDFSKAQQVFQOMFDDDNLFVIQ</div> <div>35 HUEBNLPELQUYUQKQCOHHRXSVJFSZRVRGTQ</div> <div>36 MAIPJQHRQUPWVEYOJDAVAKSOVQHGHVVUFS</div> <div>37</div> <div>38 ARISTOLOGY</div> <div>39 SHOTTS</div> <div>40 XEROPHYLLUM</div> <div>41 RIGHTSHIP</div> <div>42 CLAPPERCLAW</div> <div>43 HERODIONINE</div> <div>44 AUSCULTATIVE</div> <div>45 PLEDGED</div> <div>46 INFORMATION</div> <div>47 NONSUBSIDIARIES</div> <div>48 PRECAVE</div> <div>49 LIBOCEDRUS</div> <div>50 CHARGABLE</div> <div>51 OVERMEEKLY</div> <div>52 FASTIGIA</div> <div>53 UNDENIABLE</div> </div> <div>Perbandingan huruf: 1324</div> </div>	<div>ARISTOLOGY</div> <div> <div>Y</div> <div>G</div> <div>O</div> <div>L</div> <div>O</div> <div>T</div> <div>S</div> <div>I</div> <div>R</div> <div>A</div> <div>STTOHS</div> </div> <div>Perbandingan huruf: 872</div>	<div>SHOTTS</div> <div> <div>STTOHS</div> </div> <div>Perbandingan huruf: 1034</div>
<div> <div>38 ARISTOLOGY</div> <div>39 SHOTTS</div> <div>40 XEROPHYLLUM</div> <div>41 RIGHTSHIP</div> <div>42 CLAPPERCLAW</div> <div>43 HERODIONINE</div> <div>44 AUSCULTATIVE</div> <div>45 PLEDGED</div> <div>46 INFORMATION</div> <div>47 NONSUBSIDIARIES</div> <div>48 PRECAVE</div> <div>49 LIBOCEDRUS</div> <div>50 CHARGABLE</div> <div>51 OVERMEEKLY</div> <div>52 FASTIGIA</div> <div>53 UNDENIABLE</div> </div> <div>Perbandingan huruf: 1324</div>	<div>XEROPHYLLUM</div> <div> <div>U</div> <div>L</div> <div>V</div> <div>H</div> <div>P</div> <div>O</div> <div>R</div> <div>E</div> <div>X</div> </div> <div>Perbandingan huruf: 1324</div>	<div>RIGHTSHIP</div> <div> <div>R</div> <div>I</div> <div>G</div> <div>H</div> <div>T</div> <div>S</div> <div>H</div> <div>I</div> <div>P</div> </div> <div>Perbandingan huruf: 619</div>

CLAPPERCLAW

W  
A  
L  
C  
R  
E  
P  
P  
A  
L  
C

Perbandingan huruf: 1079

HERODIONINE

HERODIONINE

Perbandingan huruf: 1064

AUSCULTATIVE

E  
V  
I  
T  
A  
T  
L  
U  
C  
S  
U  
A

Perbandingan huruf: 1066

PLEGGED

D  
E  
G  
D  
E  
L  
P

Perbandingan huruf: 655

INFORMATION

N  
O  
I  
T  
A  
M  
R  
I  
F  
N  
I

Perbandingan huruf: 1290

NONSUBSIDIARIES

N  
O  
N  
S  
U  
B  
S  
I  
D  
I  
A  
R  
I  
E  
S

Perbandingan huruf: 216

PRECAVAE

P  
R  
E  
C  
A  
V  
A  
E

Perbandingan huruf: 973

LIBOCEDRUS

S  
U  
R  
D  
E  
C  
O  
B  
I  
L

Perbandingan huruf: 520

CHARGABLE

E  
L  
B  
A  
G  
R  
A  
H  
C

Perbandingan huruf: 1127

OVERMEEKLY

OVERMEEKLY

Perbandingan huruf: 566

FASTIGIA

F  
A  
S  
T  
I  
G  
I  
A

Perbandingan huruf: 584

UNDENTABLE

E  
L  
B  
A  
I  
N  
E  
D  
N  
U

Perbandingan huruf: 410

-----  
Total durasi: 0.003997 detik  
Total perbandingan huruf keseluruhan: 13399



## D. Repository Program (Github)

<https://github.com/Putriliza/Tucil1Stima-WordSearchPuzzle>

## E. Rangkuman Keberjalanan Program

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	✓	-