

Laporan Tugas Kecil 1
IF2211 Strategi Algoritma Tahun 2021/2022



Oleh:

Addin Nabilal Huda

13520045

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

2022

A. Algoritma Brute Force

Untuk menyelesaikan persoalan Word Search Puzzle, dapat digunakan algoritma *brute-force*. Berikut merupakan dekripsi langkah-langkah penyelesaian persoalan menggunakan algoritma *brute-force* yang digunakan pada program ini:

1. Program akan membaca *puzzle board* berupa file txt sebagai input dari pengguna. File berisi matriks (antar huruf dipisahkan oleh spasi), diikuti satu baris kosong, dan daftar kata-kata yang akan dicari. Jika file tidak ditemukan, program akan menampilkan pesan error dan program dihentikan.
2. Program akan menyimpan hasil pembacaan file ke dalam vector of vector of char untuk *puzzle board* dan vector of string untuk daftar kata-kata.
3. Program akan melakukan iterasi sebanyak jumlah kata pada daftar kata-kata, pada setiap iterasi yang dimulai dari *puzzle board* (row,col)=(0,0) dilakukan pencarian kata pada board puzzle dengan langkah sebagai berikut:
 - 1) Iterasi setiap huruf dalam *puzzle board*. Pada tiap huruf, diperiksa kemungkinan 8 arah susunan kata yang membentuk kata yang dicari. Pengecekan dengan berpindah dari satu huruf ke huruf di board puzzle dan dicocokkan dengan huruf-huruf pada kata. Pencarian ini memanfaatkan *indexing* pada vector.
 - 2) Jika *indexing* pada pencarian pada suatu arah membuat pencarian *out of bound* dan jika ditemukan huruf yang tidak cocok, iterasi dihentikan dan pencarian dilanjutkan untuk arah selanjutnya.
 - 3) Jika susunan huruf pada suatu arah cocok dengan kata yang dicari hingga huruf terakhir, maka kata dinyatakan ketemu dan iterasi dihentikan
 - 4) Pada setiap iterasi, dihitung jumlah perbandingan huruf yang dilakukan dan waktu eksekusi program
4. Jika pada row,col tidak ditemukan susunan huruf yang cocok, ulangi langkah pencarian pada poin 3 hingga semua huruf diperiksa/hingga kata sudah ketemu
5. Setelah kata ditemukan, program akan menampilkan output ke layar dengan langkah sebagai berikut:
 - 1) Program akan melakukan iterasi pada seluruh huruf pada puzzle board, setiap karakter pada puzzle board diinisiasi dengan “-“
 - 2) Dimulai dari indeks row,col yang sudah dicatat sebelumnya, program akan menginisiasi karakter-karakter yang membentuk susunan kata yang dicari dengan huruf yang bersesuaian
 - 3) Program menampilkan setiap karakter pada puzzle board ke layar
6. Jika semua huruf sudah diperiksa tetapi tidak ada susunan kata yang sesuai, program akan menampilkan pesan error ke pengguna
7. Program menampilkan total dihitung jumlah perbandingan huruf yang dilakukan dan total waktu eksekusi program

B. Source Code

Berikut merupakan beberapa fungsi dan prosedur yang digunakan pada program ini:

Nama Fungsi/Prosedur	Kegunaan
void readFile(string file_name, vector<vector<char>> *board, vector<string> *wordlist)	Membaca file txt serta menyimpan hasil pembacaan matriks <i>puzzle board</i> ke <i>vector of vector of char</i> board dan menyimpan daftar kata-kata yang akan dicari ke <i>vector of string</i> wordlist
void printOutput(string word, vector<vector<char>> board, int row, int col, int dir)	Menampilkan <i>output</i> susunan huruf pada <i>board</i> yang bersesuaian dengan kata word
bool search(string word, vector<vector<char>> board, int row, int col, int *dir, int *count)	Mencari susunan huruf yang bersesuaian dengan kata yang dicari, dimulai dari huruf pada indeks ke (row,col). Fungsi ini mengembalikan true jika kata berhasil dicari dan sebaliknya. Program mencatat arah susunan huruf yang membentuk kata yang bersesuaian pada variabel dir.

Berikut merupakan *screenshot source code* yang digunakan pada program ini.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <ctime>
#include <ratio>
#include <chrono>

using namespace std;
using namespace std::chrono;

// deklarasi dan inisiasi variabel global
int xdir[]={1,1,0,-1,-1,-1,0,1};
int ydir[]={0,-1,-1,-1,0,1,1,1};
int boardrows, boardcols;
```

```

void readFile(string file_name, vector<vector<char>> *board, vector<string> *wordlist ){
    ifstream file_in(file_name.c_str());
    if (!file_in){
        cerr <<"Cannot open the File: " <<file_name<<endl;
        exit(1);
    }
    string line;
    while (getline(file_in, line)) {
        vector<char> vecline;
        for (char &c: line) {
            if (c!=' '){
                vecline.push_back(c);
            }
        }
        if (vecline.size()>0){
            (*board).push_back(vecline);
        }
        else break;
    }

    string wordlistline;
    while (getline(file_in, wordlistline)) {
        (*wordlist).push_back(wordlistline);
    }
    file_in.close();
}

```

```

void printOutput(string word, vector<vector<char>> board, int row, int col, int dir){
    int i,j,k;
    for (i=0;i<boardrows;i++){
        for (j=0;j<boardcols;j++){
            board[i][j]='-';
        }
    }
    int len=word.size();
    for (k=0;k<word.size();k++){
        board[row][col]=word[k];
        row+=ydir[dir];
        col+=xdir[dir];
    }
    for (i=0;i<boardrows;i++){
        for (j=0;j<boardcols;j++){
            cout << board[i][j] << " ";
        }
        cout << "\n";
    }
    cout << "\n";
}

```

```

bool search(string word, vector<vector<char>> board, int row, int col, int *dir, int *count){
    int len = word.size();
    int i,y,x,d;
    *count +=1;
    if (board[row][col]==word[0]){
        for (d=0;d<8;d++){
            y=row+ydir[d];
            x=col+xdir[d];
            for (i=1;i<len;i++){
                if (x<0 || x>=boardcols || y<0 || y>=boardrows) break;
                *count += 1;
                if (board[y][x]!=word[i]) break;
                y+=ydir[d];
                x+=xdir[d];
            }
            if (i==len) {
                *dir=d;
                return true;
            }
        }
        *dir=0;
        return false;
    }
    *dir=0;
    return false;
}

```

```

int main(){
    string file_name;
    vector<vector<char>> board;
    vector<string> wordlist;
    int word, row,col;
    int dir,count, totalcount=0;
    double totaldiff=0;
    bool isAvail;

    cout << "Welcome to 2D word search program!\nInsert file name: ";
    cin >> file_name;
    cout << "\nResults:\n";
    readFile(file_name, &board, &wordlist);
    boardrows=board.size();
    boardcols=board[0].size();

```

```

    for (word=0;word<wordlist.size();word++){
        isAvail=false;
        cout << wordlist[word] << "\n";
        for (row=0;row<boardrows;row++){
            for (col=0;col<boardcols;col++){
                count=0;
                high_resolution_clock::time_point start = high_resolution_clock::now();
                isAvail=search(wordlist[word], board,row,col, &dir, &count);
                high_resolution_clock::time_point end = high_resolution_clock::now();
                duration<double> diff=duration_cast<duration<double>>(end-start);
                totalcount+=count;
                totaldiff+=diff.count();
                if (isAvail){
                    printOutput(wordlist[word], board, row, col, dir);
                    break;
                }
            }
            if (isAvail) break;
        }
        if (!isAvail) {
            cout << "Word not found!\n\n";
        }
    }
    cout << "Number of letter comparison: " << totalcount << " times\n";
    cout << "Word search program runtime: " << totaldiff*1000 << " ms \n\n";
}

```

C. Screenshot Hasil Eksekusi Program

1. Small_1.txt

Input	Output
<pre>test > small_1.txt 1 M P Q W V Q S O J E T O X G E P 2 O X T A I L S P J M C T P Q R U 3 D O W X P M Y E K C O J S B R Z 4 I O H E L C W W T E I C F B Q V 5 O L K A H M N D I T I V Y L W S 6 H Y C D B U N P Y U P R I S E B 7 W E R I C N F H X O R C T M J S 8 B M U N A T J R N P K A S I C Q 9 W M T M H P J U O S T J Y H N Q 10 E J D T P M Y P A P I W Y B Q K 11 T I A G L E D W W O P P I S W A 12 D S Q F A T B E Q R E I I N J R 13 Y L E M O H U U N T P U X D Q K 14 G D W B T E I U Q Y L O H O F I 15 16 ALPHA 17 BECALM 18 GAIT 19 HOMELY 20 HYPE 21 JOCKEY 22 KNIT 23 OXTAIL 24 QUIET 25 SELF 26 SPORTY 27 TEMPT 28 TWIN 29 UPRISE</pre>	<pre>Welcome to 2D word search program! Insert file name: ../test/small_1.txt Results: ALPHA - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - A - - - - - - - - H - - - - - - - - P - - - - - - - - L - - - - - - - - A - - - - - - - - - - - - - - - BECALM - - - - - - - - - - - - - - M - - - - - - - - L - - - - - - - - A - - - - - - - C - - - - - - E - - - - - B - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - GAIT - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - T I A G - - - - - - - - - - - - - - - - - - - - - - - - -</pre>

HOMELY

Y L E M O H

HYPE

H

Y

P

E

JOCKEY

Y E K C O J

KNIT

T

I

N

K

OXTAIL

O X T A I L

QUIET

Q U I E T

SELF

SPORTY

S
P
O
R
T
Y

2. Small_2.txt

[illegible]

```
Number of letter comparison: 1109 times
Word search program runtime: 4.359 ms
```

3. Small_3.txt

[illegible]

BECAME

CRAVAT

DIVINE

	LITTLE - - - - - - - - - - - - - - - - - - - - - - - - - - - E L T T I L - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - Number of letter comparison: 897 times Word search program runtime: 3.5985 ms
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4. Medium_1.txt

[illegible]

21

22 ABASH

23 ASSERTING

24 ASTHMA

25 CELSIUS

26 DECOMPRESS

27 DESPISING

28 DRAUGHT

29 GLANCE

30 LAUGH

31 LEES

32 LEFTY

33 MIGHT

34 NEWSFLASH

35 OSCAR

36 PASSPORT

37 SLIME

38 SMUG

39 SOURCELESS

40 THESAURUS

41 **WARPED**

ASSERTING

ASSERTING

ASTHMA

DESPISING

- G N I S I P S E D

CELSIUS

DECOMPRESS

S

P

M

O

C

E

LEES

LEFTY

MIGHT

LEES

Y

T

F

E

L

M

I

G

H

T

-	-	-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	O	-	-	-	-	-	-	-	-
-	-	-	-	P	-	-	-	-	-	-	-	-
-	-	-	-	S	-	-	-	-	-	-	-	-
-	-	-	-	S	-	-	-	-	-	-	-	-
-	-	-	-	A	-	-	-	-	-	-	-	-
-	-	-	-	P	-	-	-	-	-	-	-	-

THESAURUS

WARPED

```
Number of letter comparison: 7218 times
Word search program runtime: 81.3271 ms
```

5. Medium_2.txt

[illegible]


```
JUNGLE  
- - - - -  
- - - - -  
- - - - -  
- - E L G N U J - - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
  
Number of letter comparison: 1619 times  
Word search program runtime: 7.6569 ms
```

	Input	Output
1	DZNFBBBBFFYPTRHUXYBRINY	Welcome to 2D word search program!
2	VGCBXD XIRREGULARJYJIRR	Insert file name: ../test/medium_3.txt
3	XJLRGMNUUTMVOHJBXTJHFE	
4	IBWKGDWPU SOLIDFLT LQUEC	Results:
5	IINERTNESSZESFWBT DADCO	ADHERENCE
6	OONFCHBLKDGCVNGZXEDHDV	- - - - -
7	SMHINVNABVONCNYSMBLXIE	- - - - -
8	OJWRDN CNHOUEJWTMJILPAR	- - - - -
9	JQOMVEAICBFRGKSLDROWRU	- - - - -
10	FYENUJIFDP AEYYNKACGXBP	- - - - -
11	PREScribedDHARANXS QFKZ	- - - - - E - - - - -
12	CKNIIQGHNXM DIULRLNP KXD	- - - - - C - - - - -
13	RURUGUAYDGCAC JURVIMLBQB	- - - - - N - - - - -
14	KQC VJWNZKSQJVRTWB TJGV I	- - - - - E - - - - -
15	UHOEDTKHEOGIFEARSRQB IB	- - - - - R - - - - -
16	YIHEREWOHSJJHP PG GQT PHL	- - - - - E - - - - -
17	SISASQDESUMASJSNWPMARI	- - - - - H - - - - -
18	OJO TTTOTALLYAAO THUVSHOC	- - - - - D - - - - -
19	KCKCRERRT XT OG GN IKCASCA	- - - - - A - - - - -
20	TSENOHS IDGT ZMFUPJM Z JYL	- - - - -
21		- - - - -
22	ADHERENCE	- - - - -
23	AMUSE	- - - - -
24	BIBLICAL	- - - - -
25	BRAID	- - - - -
26	BRINY	- - - - -
27	DISHONEST	- - - - -
28	FINAL	- - - - -
29	<u>INERTNESS</u>	- - - - -

FINAL

```
Number of letter comparison: 2587 times
Word search program runtime: 23.3059 ms
```

7. large_1.txt

[illegible]

HEARING

G
N
I
R
A
E
H

HOOT

T
O
H

	IDIOM
	MODI
	INTEGRATION
	NOITARI

LINEAR

LINEAR

MANLINESS

SENL
ILN
AM

MARTINGALE

MARTINGALE

NURTURING

N
U
R
T
U
R
I
N
G

PASTURELAND

D
N
A
L
E
R
U
T
S
A
P

PHOTOGRAPHY

Y
H
P
A
R
G
O
T
O
H
P

PRACTICE

PRACTICE

PRINTABLE

P

R

I

N

T

A

B

L

E

REBATE

R

E

B

A

T

E

REPRISE

REPRISE

In 21 Col 17

SHEPHERD

D
R
E
H
P
E
H
S

SKYSCRAPER

SKYSCRAPER

SMART

T R A M S

SMOULDERED

S
M
O
U
L
D
E
R
E
D

	<div>SNAKED</div> <div>S N A K E D</div>	
	<div>SNEER</div> <div>REENS</div>	

SUNS

S
N
U
S

SWINE

E
N
I
W
S

USURP

P
R
U
S
U

HEREON

N
O
E
R
E
H
W

YULE

Y
U
L
E

Number of letter comparison: 38190 times
Word search program runtime: 185.128 ms

8. large_2.txt

[illegible]

DECOMPOSE

EFFORT

FORNICATED

F
O
R
N
I
C
A
T
E
D

ICONIC

I
C
O
N
I
C

LUDICROUSLY

Y
L
S
U
O
R
C
I
D
U
L

MANHOLE

MANHOLE

Number of letter comparison: 6636 times
Word search program runtime: 31.4988 ms

9. large_3.txt

[illegible]

AGAINST

A 20x20 grid of dots. The letters are placed at the following coordinates (row, column):

- T: (10, 12)
- S: (10, 13)
- N: (10, 14)
- I: (10, 15)
- A: (10, 16)
- G: (10, 17)
- A: (10, 18)

AMBIDEXTROUS

- AMBIDEXTROUS -



*catatan: pada testcase large_1.txt tidak seluruh output dimasukkan ke laporan karena terdapat batas karakter yang dapat ditampilkan di terminal

D. Alamat Github

Source code program dapat diakses pada tautan berikut [tautan berikut](#) mulai Selasa, 26 Januari 2022 pukul 12.30

E. Checklist

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.	√	