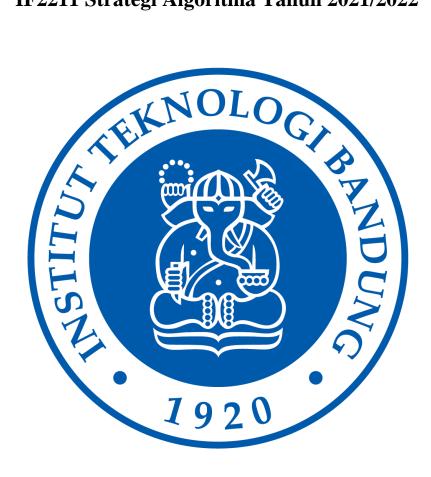
Laporan Tugas Kecil 1 IF2211 Strategi Algoritma Tahun 2021/2022



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 barus 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 susuan 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,	Membaca file txt serta menyimpan hasil
vector <vector<char>> *board, vector<string></string></vector<char>	pembacaan matriks puzzle board ke vector of
*wordlist)	vector of char board dan menyimpan daftar kata-
	kata yang akan dicari ke vector of string wordlist
void printOutput(string word,	Menampilkan <i>output</i> susunan huruf pada <i>board</i>
vector <vector<char>>> board, int row, int col, int</vector<char>	yang bersesuaian dengan kata word
dir)	
bool search(string word, vector <vector<char>></vector<char>	Mencari susunan huruf yang bersesuaian dengan
board, int row, int col, int *dir, int *count)	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;</pre>
        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++){</pre>
         for (j=0;j<boardcols;j++){</pre>
             board[i][j]='-';
         }
    int len=word.size();
    for (k=0;k<word.size();k++){</pre>
         board[row][col]=word[k];
         row+=ydir[dir];
         col+=xdir[dir];
    for (i=0;i<boardrows;i++){</pre>
         for (j=0;j<boardcols;j++){</pre>
             cout << board[i][j] << " ";
         cout << "\n";</pre>
    cout << "\n";</pre>
```

```
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++){</pre>
                if (x<0 \mid | x>=boardcols \mid | y<0 \mid | 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: ";</pre>
    cin >> file name;
    cout << "\nResults:\n";</pre>
    readFile(file_name, &board, &wordlist);
    boardrows=board.size();
    boardcols=board[0].size();
  for (word=0;word<wordlist.size();word++){</pre>
      isAvail=false;
      cout << wordlist[word] << "\n";</pre>
      for (row=0;row<boardrows;row++){</pre>
          for (col=0;col<boardcols;col++){</pre>
              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);
```

```
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;

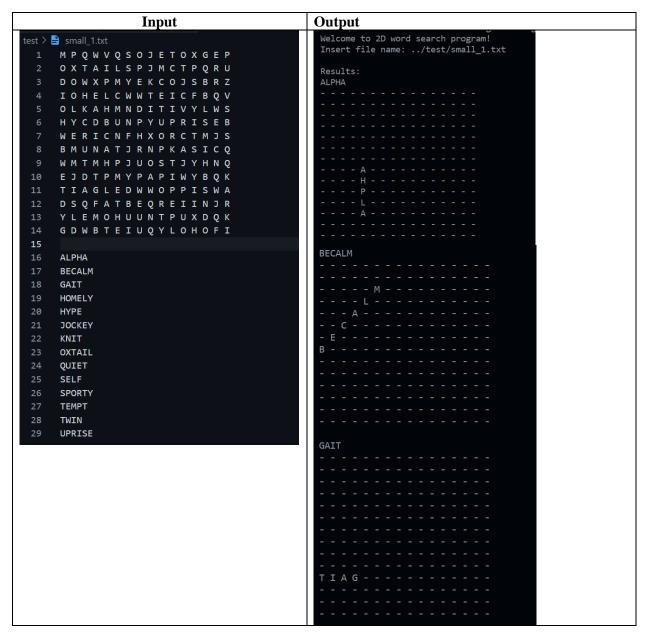
}

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";
}</pre>
```

C. Screenshot Hasil Eksekusi Program

1. Small_1.txt

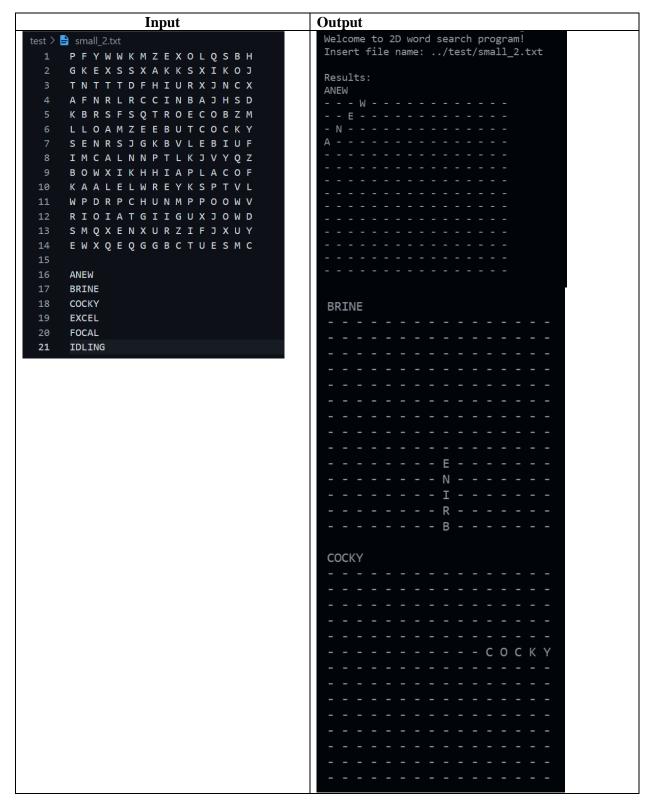


HOMELY	
Y L E M O H	
HYPE	
H	
Y	
P	
E	
JOCKEY	
Y E K C O J	
KNIT	
NILI	
T	
I	
N-	
K	

OXTAIL	
O X T A I L	
OUTET	
QUIET	
T E I U Q	
, <u> </u>	
SELF	
E-	
S	
SPORTY	
S	
P	
0	
R	
T	
Y	

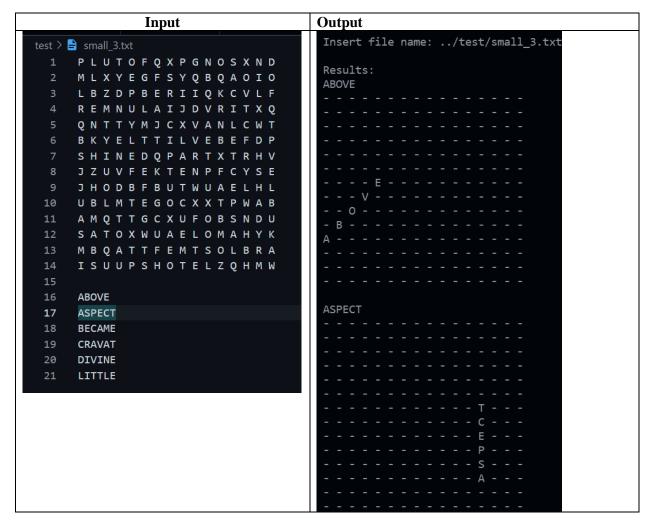
TEMPT
T
P
M
E
Т
TUTN
TWIN
T
W
II
N
UPRISE
U P R I S E -
Number of letter comparison: 2543 times
Word search program runtime: 7.589 ms

2. Small_2.txt

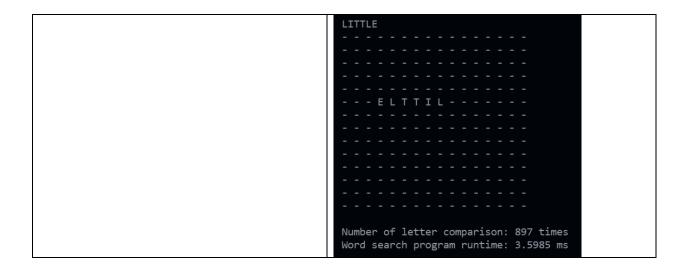


EXCEL
[
- E
C
E
FOCAL
L A C O F
IDLING
G
N
I
L
- I
Number of letter companison: 1100 times
Number of letter comparison: 1109 times Word search program runtime: 4.359 ms
HOLD DAY OF PLOST AND PARTEENED 11999 IIIO

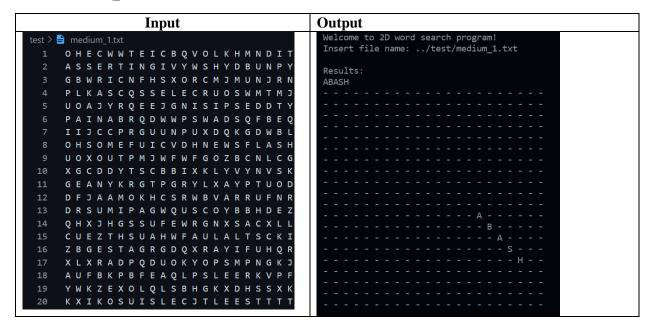
3. Small_3.txt



BECAME	
B	
E	
C	
M	
E	
CRAVAT	
C	
R	
A	
V	
A	
T	
DIVINE	
D	
I - 	
N	



4. Medium_1.txt



21		ASSERTING	
22	ABASH	A C C C D T T N C	
23	ASSERTING	A S S E R T I N G	
24	ASTHMA		
	CELSIUS		
25			
26	DECOMPRESS		
27	DESPISING		
28	DRAUGHT		
29	GLANCE		
30	LAUGH		
31	LEES		
32	LEFTY		
33	MIGHT		
34	NEWSFLASH		
		ASTHMA	
35	OSCAR		
35 36	OSCAR PASSPORT		
36	PASSPORT		
36 37	PASSPORT SLIME		
36	PASSPORT SLIME SMUG		
36 37	PASSPORT SLIME		
36 37 38	PASSPORT SLIME SMUG		
36 37 38 39 40	PASSPORT SLIME SMUG SOURCELESS THESAURUS		
36 37 38 39	PASSPORT SLIME SMUG SOURCELESS	A	
36 37 38 39 40	PASSPORT SLIME SMUG SOURCELESS THESAURUS	A	
36 37 38 39 40	PASSPORT SLIME SMUG SOURCELESS THESAURUS	- A	
36 37 38 39 40	PASSPORT SLIME SMUG SOURCELESS THESAURUS	- A	
36 37 38 39 40	PASSPORT SLIME SMUG SOURCELESS THESAURUS	- A	
36 37 38 39 40	PASSPORT SLIME SMUG SOURCELESS THESAURUS	A	

DESPISING	
GNISIPSED	
CELSIUS	
SUISLEC	
DECOMPRESS	
S	
s	
E	
P	
M	
0	
- E	
D	

DRAUGHT	
G	
A	
R	
GLANCE	
G	
- A	
N	
C	
E	
	i
LAUGH	
G	
A	

LEES
LEES
LEFTY
Y
T
F
E-
L
MIGHT
M
I
G
H

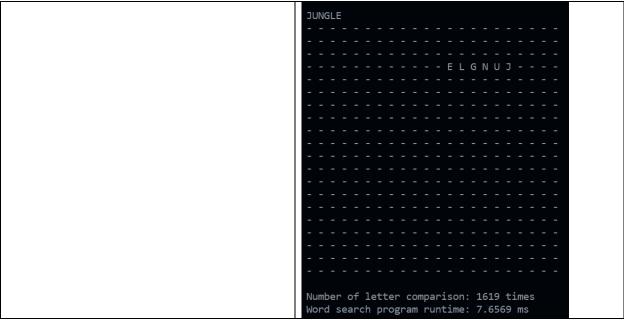
NEWSFLASH
NEWSFLASH
OSCAR
R
C
S
- 0
PASSPORT
<u>T</u>
R
P
S
A
P

SLIME	
L	
I	
SMUG	
G	
U	
M	
SOURCELESS	
500KCELE55	

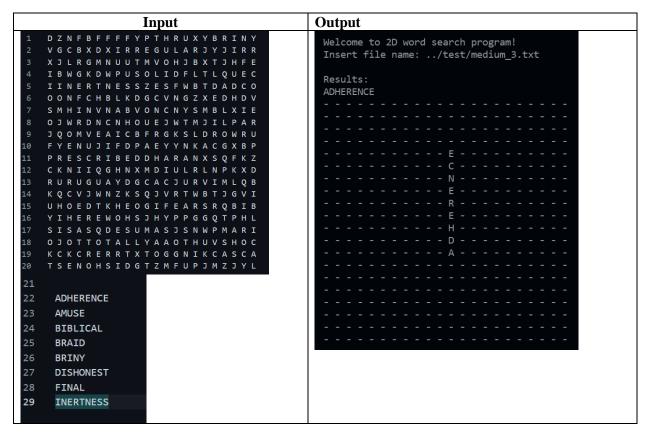
THESAURUS	
S	
UU	
A	
E	
н	
T	
WARPED	
WARFED	
P	
E	
D	
Number of letter comparison: 7218 times	
Word search program runtime: 81.3271 ms	
, , , , , , , , , , , , , , , , , , ,	

5. Medium_2.txt

CLAWS
C
L
FELINE
E
N
L
E
HUNTER
N
H



6. medium_3.txt



AMUSE
E S U M A
BRAID
DRAID
D -
I-
A -
R -
B -
BRINY
BRINY

DISHONEST	
T S E N O H S I D	
FINAL	
L	
A	
N	
I	
INERTNESS _	
INERTNESS	
INERTNESS	
INERTNESS	
INERTNESS	
- I N E R T N E S S	
- I N E R T N E S S	
- I N E R T N E S S	
- I N E R T N E S S	
- I N E R T N E S S	
- I N E R T N E S S	
- I N E R T N E S S	
- I N E R T N E S S	
- I N E R T N E S S	
- I N E R T N E S S	
- I N E R T N E S S	
- I N E R T N E S S	
INERTNESS	
- I N E R T N E S S	

7. large_1.txt

Input	Output
•	FRIGID
	D D
	R
	GERMANY
	C C D M A N V

HEARING
NN
•
I
R
A
EE
H
HOOT
0
0
н

TDTOM
IDIOM
D
I
TNITEGRATION
INTEGRATION
N
N O
N
N O
N
N
N O
N O
N O
N O
N O O I I I I I I I I I I I I I I I I I
N O O O O O O O O O O O O O O O O O O O
N O O O O O O O O O O O O O O O O O O O
N O O O O O O O O O O O O O O O O O O O
N O O O O O O O O O O O O O O O O O O O
N O O O O O O O O O O O O O O O O O O O
N O O I I I I I I I I I I I I I I I I I
N O I I I I I I I I I I I I I I I I I I
N O O I I I I I I I I I I I I I I I I I
N O I I I I I I I I I I I I I I I I I I
N O O I I I I I I I I I I I I I I I I I
N O O I I I I I I I I I I I I I I I I I
N O O O O O O O O O O O O O O O O O O O
N O O I I T A A A R R G G E E T T N N T I T T N N T I T T T T T T T T
N
N O O I I T A A A R R G G E E T T N N T I T T N N T I T T T T T T T T
N
N O O I I I A A A A A A A A A A A A A A A

LINEAR	
I	
E	
A	
MANLINESS	Ī
PANLINESS	ĺ
	ĺ
ss	
S	ĺ
N	
I	
N	
A	
	L
MARTINGALE	
MARTINGALE-	

AU IDTUDTAG	
NURTURING	
N	
U	
R	
U	
RR	
II	
N	
PASTURELAND	
D	
NN	
AA	
<u>E</u>	
R	
T	
A	
P	
PHOTOGRAPHY	
A	
R	
0 0 H	
P -	

	PRACTICE	
	TRACTICE	
	PRACTICE	
	DOTALTADI E	
	PRINTABLE	
	P R R	
	INN	
	T	
	77	
	B	
	E	
	REBATE	
	R	
	E	
	• • • • • • • • • • • • • • • • • • • •	
1		
l I		

REPRISE
REPRISE
Ln 21 Cal 17
SHEPHERD
D R -
E-
P
SKYSCRAPER
S K Y S C R A P E R

SMART
T R A M S
ANALY REPER
SMOULDERED
S
M
0
U
D
EE
RR
E

SNAKED
NN
<u>E</u>
SNEER
R E E N S

SUNS	
S	
U	
S	
CUTNE	
SWINE	
I	
S	
USURP	
P	
R	
S	
U	

/HEREON	
. N	
0	
E	
R	
H	
YULE	
EE	
Number of letter companison: 38199 times	
Number of letter comparison: 38190 times Word search program runtime: 185.128 ms	
word Search program runtime: 185.128 ms	

8. large_2.txt

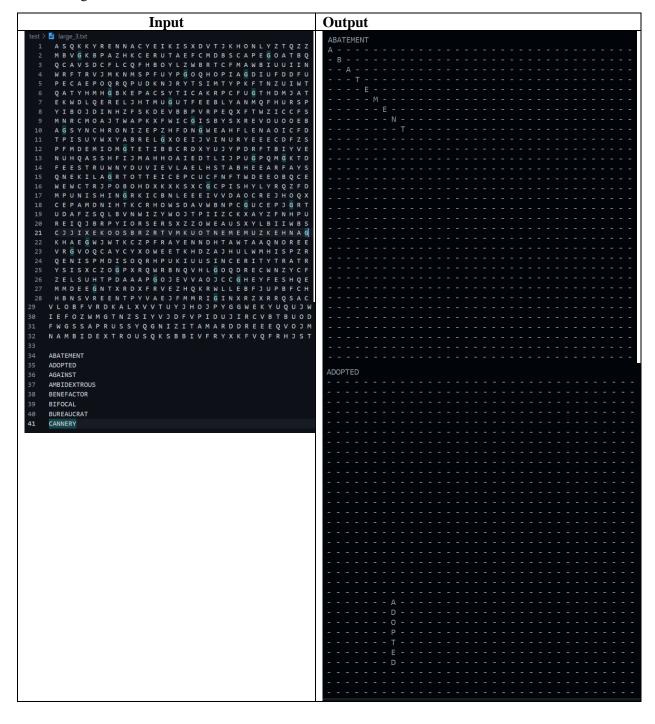


DECOMPOSE	
## ## ## ## ## ## ## ## ## ## ## ## ##	DECOMPOSE
EFFORT O O O O O O O O O O O O O O O O O O O	
EFFORT D O O O O O O O O O O O O	
EFFORT D O O O O O O O O O O O O	
EFORT	
EFFORT E 5 S 9 P 9 D 0 D 0 D 0 D 0 F 6 F 7	
EFFORT E 5 S 9 P 9 D 0 D 0 D 0 D 0 F 6 F 7	
S 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
О О О О О О О О О О О О О О О О О О О	
EFFORT	
# # # # # # # # # # # # # # # # # # #	
EFORT	
EFFORT R R R C E B D D D D D D D D D D D D	
EFFORT F O	
EFFORT R O F	E
EFFORT R	D
EFFORT EFFORT F F E E E	
EFFORT EFFORT F F E E E	
EFFORT R R F O	
EFFORT R R O F F E	
EFFORT R R O F F E	
EFFORT R O F E E	
EFFORT T	
EFFORT	
EFFORT T R O O F F E E	
EFFORT T R O O F F E E	
EFFORT T	
EFFORT T	
EFFORT T	
EFFORT T R O O F E	
EFFORT T	
T	
T T R O - F F F F F F F F F F F F F F F F F F	
T - R - O F E E	
T - R - O F E E	
T - R - O F E E	
T - R - O F E E	
T - R - O F E E	
T - R - O F E E	
T - R - O F E E	
T - R - O F E E	
T - R - O F F E E	
- T R	
- T R	
- T R	
- T R	
- T R	
- T R	
- T R	
T	
F	IX.

FORNICATED	
F	
0	
R	
N	
A	
E	
DD	
TCONTC	
ICONIC	
T	
T	
T	
T	
T	
T	

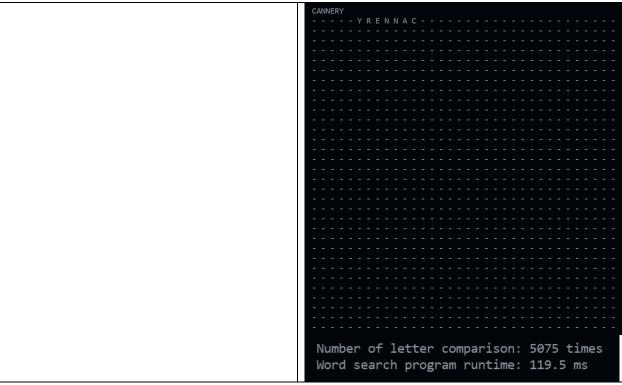
LUDICROUSLY	
Y	
R	
uu	
MANHOLE	
MANHOLE	
Number of letter comparisons	6636 times
Number of letter comparison: (Word search program runtime:	oodo cines
Word search program runtime:	31.4988 ms

9. large_3.txt



AGAINST
T
N
I
A
G G
A
AMBIDEXTROUS

<u> </u>	
	BENEFACTOR
	E
	N
	B
	BUREAUCRAT
	BUREAUCRAT
	BUREAUCRAT
	T
	- T - A -
	- T
	- T
	T A - C C C C C C C C C C C C C C C -
	T A - R - C - C - U - U - U
	T
	T A - R - C - C - U - U - U
	T - A C
	T
	T T
	T
	T T
	T
	T
	T A
	T A C
	T A
	T A C
	A
	A
	A
	A
	A
	A
	A
	A
	A
	T A
	A
	T A
	T A
	T A



^{*}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 <u>tautan berikut</u> mulai Selasa, 26 Januari 2022 pukul 12.30

E. Checklist

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan (no syntax error)	$\sqrt{}$	
2. Program berhasil running	$\sqrt{}$	
3. Program dapat membaca file masukan dan menuliskan luaran.	$\sqrt{}$	
4. Program berhasil menemukan semua kata di dalam puzzle.	V	