

**Laporan Tugas Kecil 1**  
**IF 2211 Strategi Algoritma**  
**Penyelesaian Word Search Puzzle dengan Algoritma**  
**Brute Force**



Disusun Oleh:  
13520163 Frederik Imanuel Louis

Sekolah Tinggi Elektro dan Informatika  
Institut Teknologi Bandung  
2022

# Daftar Isi

Daftar Isi .....	2
BAB 1 Algoritma Program .....	3
1.1 Input Puzzle.....	3
1.2 Pencarian Solusi (Brute Force) .....	3
1.3 Output Solusi.....	3
BAB II Source Code Program .....	4
BAB III Pengujian Program.....	7
3.1 Puzzle “mudah” .....	7
3.2 Puzzle “sedang” .....	8
3.3 Puzzle “sulit” .....	9
Lampiran .....	11

# **BAB 1**

## **Algoritma Program**

### **1.1 Input Puzzle**

Setelah program dijalankan, user akan diminta memasukkan nama file puzzle yang sudah disimpan dalam folder test, dengan bentuk text file. Jika file tidak ditemukan, program akan menampilkan error message dan berhenti. Ekstensi “.txt” tidak perlu dituliskan. Kemudian, puzzle akan dimuat dalam array of vector of char “arr”, dengan ukuran puzzle disimpan pada variabel “n” dan “m”. Kata yang ingin dicari kemudian disimpan pada vector of string “s”, dan banyak kata disimpan dalam variabel “k”.

### **1.2 Pencarian Solusi (Brute Force)**

Pada tiap entri character puzzle, program akan mencari, untuk tiap kata dan untuk tiap arah (8 arah), apakah kata yang dicari dapat dimulai dari entri tersebut dengan arah yang pilih. Iterasi dilakukan dengan urutan (dari luar ke dalam) baris puzzle, kolom puzzle, kata yang diuji, dan arah pengecekan puzzle. Jika solusi ditemukan, solusi akan disimpan dalam rupa koordinat awal posisi kata pada puzzle (x,y), beserta arah pengecekan puzzle dan panjang kata tersebut. Informasi tersebut disimpan dalam vector of info “ans”, dimana info adalah struct yang memuat keempat data tersebut.

### **1.3 Output Solusi**

Program akan mengecek apakah vector “ans” kosong. Jika vector kosong, maka program akan menampilkan bahwa tidak ada solusi yang ditemukan. Jika vector memiliki isi, maka untuk tiap jawaban yang disimpan ans, huruf yang merupakan bagian dari jawaban pada puzzle akan diberi warna dan garis bawah (underline). Informasi warna tersebut akan disimpan pada matriks (array dua dimensi) “col”. Kemudian, puzzle dioutput dengan memperhatikan warna tiap huruf. Untuk tidak memperumit output, warna jawaban hanya ada maksimum 6 macam.

## BAB II

### Source Code Program

```
#include<bits/stdc++.h>
#include<unistd.h>
#include<chrono>
#define pb push_back
#define underline "\033[4m"
#define DEFAULT "\x1B[0m"
#define RED "\x1B[31m"
#define GREEN "\x1B[32m"
#define YELLOW "\x1B[33m"
#define BLUE "\x1B[34m"
#define MAGENTA "\x1B[35m"
#define CYAN "\x1B[36m"
using namespace std;
using namespace std::chrono;

// 8 search directions
int dx[8]={0,0,1,1,1,-1,-1,-1};
int dy[8]={1,-1,0,1,-1,0,1,-1};

// Puzzle
vector<char> arr[100];
int col[100][100]; //determines color for output

// Words
vector<string> s;

// Answers
struct info{
    int x;
    int y;
    int dir; //direction
    int l; //length
};

vector<info> ans;

// Sizes
int n,m=0; //array size
int k=0; //words size

// Functions
bool inside(int x,int y){
    return (x>=0&&y>=0)&&(x<n&&y<m);
}

// Main
int main(){
    // time and comparison counter
    int cnt=0;
    // input
```

```

string filename;
cout<<"Masukkan nama file: ";
cin>>filename;
filename="test/"+filename+".txt";
ifstream in;
in.open(filename);
if(!in){
    cout<<"File tidak ditemukan!"<<endl;
    return 0;
}
int t=0; // puzzle idx
string temp;
while(getline(in,temp)){
    if(temp.size()==0)continue;
    if(temp[1]!=' '||temp[1]!='\n'){
        for(int i=0;i<temp.size();i+=2)arr[t].pb(temp[i]);
        t++;
    }else{
        s.pb(temp);
    }
}
n=t;
m=arr[0].size();
k=s.size();

// search
auto start = high_resolution_clock::now();
for(int i=0;i<n;i++){ //iterate over coordinates
    for(int j=0;j<m;j++){
        for(auto word:s){ //iterate over words
            for(int d=0;d<8;d++){ //iterate over directions
                int idx=0;
                int x=i;
                int y=j;

                while(inside(x,y)&&idx<word.size()&&word[idx]==arr[x][y]){
                    cnt++;
                    x+=dx[d];
                    y+=dy[d];
                    idx++;
                    while(word[idx]==' ')idx++;
                }
                if(idx==word.size()){ //ans found
                    info temp={i,j,d,idx};
                    ans.pb(temp);
                }
            }
        }
    }
}

// output answer
if(ans.size()==0){
    cout<<"Tidak ada solusi yang ditemukan!"<<endl;
    return 0;
}

```

```

    }
    int curcol=1;
    for(auto i:ans){
        while(i.l--){ //set color
            col[i.x][i.y]=curcol;
            i.x+=dx[i.dir];
            i.y+=dy[i.dir];
        }
        curcol++;
        if(curcol>6)curcol-=6;
    }

    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){ //print everything
            switch(col[i][j]){
                case 1:
                    printf("%s%s%c%s",underline,RED,arr[i][j],DEFAULT);
                    break;
                case 2:
                    printf("%s%s%c%s",underline,GREEN,arr[i][j],DEFAULT);
                    break;
                case 3:
                    printf("%s%s%c%s",underline,YELLOW,arr[i][j],DEFAULT);
                    break;
                case 4:
                    printf("%s%s%c%s",underline,BLUE,arr[i][j],DEFAULT);
                    break;
                case 5:
                    printf("%s%s%c%s",underline,MAGENTA,arr[i][j],DEFAULT);
                    break;
                case 6:
                    printf("%s%s%c%s",underline,CYAN,arr[i][j],DEFAULT);
                    break;
                default:
                    printf("%c ",arr[i][j]);
            }
        }
        printf("\n");
    }

    auto stop = high_resolution_clock::now();
    auto duration = duration_cast<microseconds>(stop - start);
    cout<<endl;
    cout<<"Waktu eksekusi program: "<< duration.count()<<" mikrosekond<<endl;
    cout<<"Banyak perbandingan yang dilakukan: "<<cnt<<" perbandingan"<<endl;
    cout<<"Banyak kata yang ditemukan: "<<ans.size()<<"/"<<k<<endl;

    return 0;
}

```

## BAB III

# Pengujian Program

### 3.1 Puzzle “mudah”

```
C:\Users\Erik\Github\STIMA-TC1>main
Masukkan nama file: easy_genshin
Z Q U Y A S V G R J
J R V R O F T P U D
L U C F S L V A S R
H P R Q Q K O V T T
C S M E F V M I Q I
S A R N B H U S W A
I E L Q X M W X J B
E M A Y A K A Z E A
W A U Z G T Z Y H A
I G Z A J R A F W L

Waktu eksekusi program: 21081 mikrosekond
Banyak perbandingan yang dilakukan: 495 perbandingan
Banyak kata yang ditemukan: 7/7
```

Gambar 1. Output pengujian “easy\_genshin”

```
C:\Users\Erik\Github\STIMA-TC1>main
Masukkan nama file: easy_composers
X B M O Z A R T R S
L L A M I A E P P A
A U Y C A Z V I B T
N S H H H M S N Z I
D J G R J T V I E E
Y J R E I F M P L V
A Q A O U X V O H W
H X V P U A P H I U
F T E P F K Z C N T
W Y L K E Y M B I V

Waktu eksekusi program: 27157 mikrosekond
Banyak perbandingan yang dilakukan: 257 perbandingan
Banyak kata yang ditemukan: 7/7
```

Gambar 2. Output pengujian “easy\_composers”

Pada puzzle “mudah”, diberikan masukan puzzle berukuran 10 x 10, dengan 7 kata yang ingin dicari. Pada kedua kasus, semua kata berhasil ditemukan.

### 3.2 Puzzle “sedang”

```
C:\Users\Erik\Github\STIMA-TC1>main
Masukkan nama file: medium_youzitsu
C H A B A S H I R A D C A C C
G U Y K K F N E U U Y R G O I
A B O R Y A M A U C H I A I M
Y A A Y G A P O M A T A R I H
A N O A Z E D H O R I K A T A
N A S A K U R A S E N C S T A
O M O G S A D I H S U K Z A O
K A T S U R A G I P E O P F C
O R S A K A Y A N A G I A F I
U D E L S E T E P F G V M E J
J F V E G D H S V N E Y I J N
I A N I H A S A A C Y F U Q A
E D S T S K A R U I Z A W A K
S H A V Z O Q U M G N Y F K D
S H D S P L P U F F K W O F S

Waktu eksekusi program: 82499 mikrosekon
Banyak perbandingan yang dilakukan: 2075 perbandingan
Banyak kata yang ditemukan: 15/15
```

Gambar 3. Output pengujian “medium\_youzitsu”

```
C:\Users\Erik\Github\STIMA-TC1>main
Masukkan nama file: medium_ducks
G Q Z G V Z K M S U V B W A D
P D C C Z J T K N D E S L W N
O O M G D U J C G U R S D U T
C H O U J R D U S D C Z V F E
H C U R S H A D Q O D O A I X
A F A V T K P L T I C L R S U
R R I P S T J E L Z C U T Q A
D E M D E M R H I A D N N V N
E C R J A T B S T D M W E C B
P K O S E U E E Y N C O R B S
F L K W Z X D A A C T O R F B
X E V M I I Q W L H Q D O W N
D D O I I G A D W A L L T E K
N E J D J B V G D A T U R M Z
V V C Y U J B A S P W Z I S X

Waktu eksekusi program: 64035 mikrosekon
Banyak perbandingan yang dilakukan: 1115 perbandingan
Banyak kata yang ditemukan: 15/15
```

Gambar 4. Output pengujian “medium\_ducks”



```

C:\Users\Erik\Github\STIMA-TC1>main
Masukkan nama file: medium_currencies
G T Z N R U B L E K K P P K K
N U U O G R O S D L D Q O Q G
P Z A C Q D Y X Y D B D K O X
O X E G I Z R R D I A P M F
Q V E L I R A E O N E Y H I G
D J P B S N U L A Y B T E I H
U Y U D M G L R O S E P N N D
Q B R N L A P F K S N V D N F
M Q J M R B O H Z W D F A W C
C A L Y S N U W Z I A R O R O
N Y D K R O N E A N X R J W C
A L G S R C D F T L U H T O B
R S G E A I S G P E A C W N J
F M G N N N A Q U A U F I H O
N Z T E N X N L L Z Y A Y P Z

Waktu eksekusi program: 69156 mikrosekon
Banyak perbandingan yang dilakukan: 1388 perbandingan
Banyak kata yang ditemukan: 16/15

```

Gambar 5. Output pengujian “medium\_currencies”

Pada puzzle “sedang”, diberikan masukan puzzle berukuran 15 x 15, dengan 15 kata yang ingin dicari. Pada semua kasus, semua kata berhasil ditemukan. Pada puzzle “medium\_currencies”, ditemukan 2 buah kata “YEN”, maka banyak kata yang ditemukan melebihi banyak kata yang diberikan (16/15).

### 3.3 Puzzle “sulit”

```

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

Waktu eksekusi program: 84679 mikrosekon
Banyak perbandingan yang dilakukan: 2446 perbandingan
Banyak kata yang ditemukan: 15/15

```

Gambar 6. Output pengujian “hard\_war”

```

C:\Users\Erik\Github\STIMA-TC1>main
Masukkan nama file: hard_anime
J N E A N A H O N A Q B R S W F L S Q W B N
A B N K I C B D P J T I H I T W X E C T Y H
R Z M O N O G A T A R I K V V F H Y O U K A
Q I K L R O O O G N O D U F S U A O S R W I
Q T B G M A U N N J L J S A N T E U B R A K
A W L M B D G S U E A U K H R G G Z L U M Y
R P D S Q Y L A K T C B I E G N G I G N T U
Q V E R L O R D M A Z Z E N E C Q I Q F G U
I A J L T X X C T I C F C F D F Q S E E J M
M E I A N E I T E O M R A C T O D U F M F V
T X A Y X C Z P H D K A H F E Y Q N B Y R O
M B Z S Z C I D G B H A W B Y Z Z W Y M Y I
L U U X P P F Y O A R I W F A S T M X G P M
Z V N M R O D T I L R P C H E Y X H Q J Y I
S B M I K A W T O N C C D T R B I B I G Z C
G T H P N R S I U K W K A T V Z V M D S Q X
P W D L E L I J B N H F M X T S A P I A G L
K E H V Z E Z E U A B Z R O R E G A I R U I
B W S Q F C L A N N A D G V Y J L X A C O E
G Z K F X T X H A R U S N E I T J K Y X D H

Waktu eksekusi program: 99100 mikrosekond
Banyak perbandingan yang dilakukan: 2551 perbandingan
Banyak kata yang ditemukan: 15/15

```

Gambar 6. Output pengujian “hard\_anime”

```

C:\Users\Erik\Github\STIMA-TC1>main
Masukkan nama file: hard_bedroom
K S J K X I K P P A L A R M C L O C K I B F
A O R N H E A J M A W L U W A A Y G C K W Y
Y N P E T X S J A W Y N S Z Q V F B I D N O
N H R P P J A Y L Z R X N U T W N D X M D Z
C Y I O E P C V P Y S J Y M Z F S I V F K I
M O A U B Z I N W O G G N I S S E R D V D E
D J M C M E N L O C N P R P T M G O F R A S
X H R F A K B R S H I X E P X J P N E P Y O
K L T M O U B Y O L O N M Q K R A S Q P A L
N V Y K X R Y Y L R A S O C L H S B J G M C
T A R B U Q T O J F R O T S E E O Q E S Y R
Y S Y H D S W E C E Z I E R R P N I I D M Z
P N Z O U S W B R W I N M F J V O F O D A R
D I K T V S V Q L E H L Y A S G X Q L H A O
Q A W E E T E L E V I S I O N Z G L D E P R
Q T U T I V D U O M S W I N D O W S D R H T
R R R E R V K E K K L K S Q J T K O U S Z S
W U J T E D D Y B E A R E E S Y A G P Y T R
N C N L G V U P T N S A D Y S L Y Q Z Q Z P
V S R H Z A L Q I H G I L O N Y N V A H I P

Waktu eksekusi program: 78937 mikrosekond
Banyak perbandingan yang dilakukan: 2550 perbandingan
Banyak kata yang ditemukan: 16/15

```

Gambar 7. Output pengujian “hard\_bedroom”

Pada puzzle “sulit”, diberikan masukan puzzle berukuran 22 x 20, dengan 15 kata yang ingin dicari. Pada semua kasus, semua kata berhasil ditemukan. Pada puzzle “hard\_bedroom”, ditemukan 2 buah kata “BED”, maka banyak kata yang ditemukan melebihi banyak kata yang diberikan (16/15).

# Lampiran

Tabel 1. Cek List Status Program

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan (no syntax error)	✓	
2. Program berhasil <i>running</i>	✓	
3. Program dapat membaca file masukan dan menuliskan keluaran	✓	
4. Program berhasil menemukan semua kata dalam puzzle	✓	

Repositori program: <https://github.com/dxt99/STIMA-TC1>

Link Google Drive (sama dengan isi repo github):

[https://drive.google.com/drive/folders/1mS7LxI66utwUvM3fxRKw\\_P6u4Q0KZUSL?usp=sharing](https://drive.google.com/drive/folders/1mS7LxI66utwUvM3fxRKw_P6u4Q0KZUSL?usp=sharing)