**LAPORAN TUGAS KECIL 1**

**IF2211 STRATEGI ALGORITMA**

**WORD SEARCH PUZZLE**



Oleh

Primanda Adyatma Hafiz (13520022)

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

202

# Daftar Isi

[Daftar Isi 1](#_Toc93957690)

[BAB I ALGORITMA BRUTE FORCE 2](#_Toc93957691)

[1.1 Cara Kerja Program 2](#_Toc93957692)

[1.2 Deskripsi Fungsi dan Prosedur 2](#_Toc93957693)

[BAB 2 SOLUSI PROGRAM 4](#_Toc93957694)

[2.1 Kode Program 4](#_Toc93957695)

[BAB 3 HASIL PENGUJIAN 10](#_Toc93957696)

[3.1 Kasus Ukuran Puzzle Small 10](#_Toc93957697)

[3.1.1 Uji 1 10](#_Toc93957698)

[3.1.2 Uji 2 11](#_Toc93957699)

[3.1.3 Uji 3 12](#_Toc93957700)

[3.2 Kasus Ukuran Puzzle Medium 13](#_Toc93957701)

[3.2.1 Uji 1 13](#_Toc93957702)

[3.2.2 Uji 2 14](#_Toc93957703)

[3.2.3 Uji 3 15](#_Toc93957704)

[3.3 Kasus Ukuran Puzzle Large 17](#_Toc93957705)

[3.3.1 Uji 1 17](#_Toc93957706)

[3.3.2 Uji 2 19](#_Toc93957707)

[3.3.3 Uji 3 21](#_Toc93957708)

[LAMPIRAN 23](#_Toc93957709)

[Alamat Drive 23](#_Toc93957710)

[Cek List Tabel 23](#_Toc93957711)

# BAB I ALGORITMA BRUTE FORCE

## 1.1 Cara Kerja Program

1. Melakukan iterasi dari setiap kata yang ingin dicek pada puzzle
2. Untuk setiap kata dilakukan proses iterasi untuk setiap karakter yang terdapat dalam puzzle
3. Untuk setiap karakter dalam puzzle akan dilakukan pengecekan secara 8 arah yaitu arah atas, bawah, kiri, kanan, kiri atas, kanan atas, kiri bawah, dan kanan bawah
4. Sebelum dilakukan pengecekan, untuk setiap arah akan dilakukan validasi terlebih dahulu sehingga panjang kata tidak melebihi ukuran puzzle yang tersedia untuk setiap arah
5. Jika validasi berhasil akan dilakukan pembandingan setiap karakter dari kata dan puzzle yang berada pada urutan yang sama
6. Jika ditemukan sepasang karakter yang berbeda maka pengecekan untuk karakter puzzle tersebut akan dihentikan dan akan dilanjutkan ke karakter selanjutnya (kembali ke nomor 3)
7. Sedangkan jika keseluruhan karakter sama maka seluruh karakter pada puzzle yang baru saja dicek akan diberikan kode warna baru (kode warna awal diinisiasi dengan warna normal) dan pengecekan akan dilanjutkan ke kata selanjutnya (kembali ke nomor 2)
8. Setelah keseluruhan kata selesai dilakukan pengecekan, setiap karakter dalam puzzle akan ditampilkan pada layar sesuai kode warna yang tersimpan

## 1.2 Deskripsi Fungsi dan Prosedur

|  |  |
| --- | --- |
| **Nama Fungsi dan Prosedur** | **Deskripsi** |
| void inputFile() | Prosedur ini berfungsi untuk memasukkan matriks puzzle ke dalam array dua dimensi dan list kata ke dalam array. |
| void getResult() | Prosedur ini berfungsi untuk mengecek apakah keseluruhan kata yang dimasukkan terdapat dalam puzzle. Jika kata terdapat dalam puzzle maka setiap karakter puzzle yang cocok dengan kata akan diberi kode warna. |
| void printResult() | Prosedur ini berfungsi untuk menampilkan keseluruhan kata yang ingin dicari beserta hasil pencariannya dalam puzzle.. |
| int main() | Fungsi ini merupakan program utama yang memanggil ketiga fungsi di atas serta menghitung total waktu eksekusi program. |

# BAB 2 SOLUSI PROGRAM

## 2.1 Kode Program

Berikut adalah implementasi program Word Search Puzzle dalam bahasa C++

#include <bits/stdc++.h>

#include <chrono>

using namespace std;

using namespace std::chrono;

#define NORMAL "\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"

#define fs first

#define sc second

const int numColor=6;

int m=1;

int n=0;

int k=0;

long long cnt=0;

string colors[7]={NORMAL,RED,GREEN,YELLOW,BLUE,MAGENTA,CYAN};

vector<vector<pair<char,int>>>puzzle;

vector<string>word;

void printColor(string color,char text){

    cout<<color<<text;

    cout<<NORMAL;

}

void inputFile(string path)

{

    ifstream in(path);

    string s;

    getline(in,s);

    int strsize=s.size();

    for(int i=0;i<strsize;i++){

        if(s[i]==' ')m++;

    }

    while(s!=""){

        n++;

        getline(in,s);

    }

    while(getline(in,s)){

        k++;

    }

    in.close();

    freopen(path.c\_str(), "r", stdin);

    puzzle.resize(n);

    word.resize(k);

    for(int i=0;i<n;i++){

        puzzle[i].resize(m);

        for(int j=0;j<m;j++){

            cin>>puzzle[i][j].fs;

        }

    }

    for(int i=0;i<k;i++){

        cin>>word[i];

    }

}

void getResult(){

    int curColor=1;

    for(string s:word){

        int len=s.size();

        bool found=false;

        for(int i=0;i<n;i++){

            if(found)break;

            for(int j=0;j<m;j++){

                if(i+1>=len){

                    int it=0;

                    while(it<len-1 && s[it]==puzzle[i-it][j].fs){

                        it++;

                    }

                    if(s[it]==puzzle[i-it][j].fs){

                        found=true;

                        for(it=0;it<len;it++){

                            if(puzzle[i-it][j].sc)continue;

                            puzzle[i-it][j].sc=curColor;

                        }

                        curColor=(curColor%numColor)+1;

                        cnt+=it+2;

                        break;

                    }

                    cnt+=it+2;

                }

                if(n-i>=len){

                    int it=0;

                    while(it<len-1 && s[it]==puzzle[i+it][j].fs){

                        it++;

                    }

                    if(s[it]==puzzle[i+it][j].fs){

                        found=true;

                        for(it=0;it<len;it++){

                            if(puzzle[i+it][j].sc)continue;

                            puzzle[i+it][j].sc=curColor;

                        }

                        curColor=(curColor%numColor)+1;

                        cnt+=it+2;

                        break;

                    }

                    cnt+=it+2;

                }

                if(j+1>=len){

                    int it=0;

                    while(it<len-1 && s[it]==puzzle[i][j-it].fs){

                        it++;

                    }

                    if(s[it]==puzzle[i][j-it].fs){

                        found=true;

                        for(it=0;it<len;it++){

                            if(puzzle[i][j-it].sc)continue;

                            puzzle[i][j-it].sc=curColor;

                        }

                        curColor=(curColor%numColor)+1;

                        cnt+=it+2;

                        break;

                    }

                    cnt+=it+2;

                }

                if(m-j>=len){

                    int it=0;

                    while(it<len-1 && s[it]==puzzle[i][j+it].fs){

                        it++;

                    }

                    if(s[it]==puzzle[i][j+it].fs){

                        found=true;

                        for(it=0;it<len;it++){

                            if(puzzle[i][j+it].sc)continue;

                            puzzle[i][j+it].sc=curColor;

                        }

                        curColor=(curColor%numColor)+1;

                        cnt+=it+2;

                        break;

                    }

                    cnt+=it+2;

                }

                if(i+1>=len && j+1>=len){

                    int it=0;

                    while(it<len-1 && s[it]==puzzle[i-it][j-it].fs){

                        it++;

                    }

                    if(s[it]==puzzle[i-it][j-it].fs){

                        found=true;

                        for(it=0;it<len;it++){

                            if(puzzle[i-it][j-it].sc)continue;

                            puzzle[i-it][j-it].sc=curColor;

                        }

                        curColor=(curColor%numColor)+1;

                        cnt+=it+2;

                        break;

                    }

                    cnt+=it+2;

                }

                if(n-i>=len && m-j>=len){

                    int it=0;

                    while(it<len-1 && s[it]==puzzle[i+it][j+it].fs){

                        it++;

                    }

                    if(s[it]==puzzle[i+it][j+it].fs){

                        found=true;

                        for(it=0;it<len;it++){

                            if(puzzle[i+it][j+it].sc)continue;

                            puzzle[i+it][j+it].sc=curColor;

                        }

                        curColor=(curColor%numColor)+1;

                        cnt+=it+2;

                        break;

                    }

                    cnt+=it+2;

                }

                if(i+1>=len && m-j>=len){

                    int it=0;

                    while(it<len-1 && s[it]==puzzle[i-it][j+it].fs){

                        it++;

                    }

                    if(s[it]==puzzle[i-it][j+it].fs){

                        found=true;

                        for(it=0;it<len;it++){

                            if(puzzle[i-it][j+it].sc)continue;

                            puzzle[i-it][j+it].sc=curColor;

                        }

                        curColor=(curColor%numColor)+1;

                        cnt+=it+2;

                        break;

                    }

                    cnt+=it+2;

                }

                if(j+1>=len && n-i>=len){

                    int it=0;

                    while(it<len-1 && s[it]==puzzle[i+it][j-it].fs){

                        it++;

                    }

                    if(s[it]==puzzle[i+it][j-it].fs){

                        found=true;

                        for(it=0;it<len;it++){

                            if(puzzle[i+it][j-it].sc)continue;

                            puzzle[i+it][j-it].sc=curColor;

                        }

                        curColor=(curColor%numColor)+1;

                        cnt+=it+2;

                        break;

                    }

                    cnt+=it+2;

                }

            }

        }

    }

}

void printResult(){

    cout<<"Hasil pencarian :"<<endl;

    for(int i=0;i<n;i++){

        for(int j=0;j<m;j++){

            printColor(colors[puzzle[i][j].sc],puzzle[i][j].fs);

            cout<<" ";

        }

        cout<<endl;

    }

}

int main()

{

    ios\_base::sync\_with\_stdio(0);

    cout<<"=================="<<endl;

    cout<<"WORD SEARCH PUZZLE"<<endl;

    cout<<"=================="<<endl;

    cout<<"Masukkan nama file : ";

    string file;

    cin>>file;

    inputFile("test/"+file);

    auto start=high\_resolution\_clock::now();

    getResult();

    printResult();

    auto stop=high\_resolution\_clock::now();

    auto duration = duration\_cast<microseconds>(stop - start);

    double durationMicro=duration.count();

    cout<<fixed;

    cout<<setprecision(3);

    if(durationMicro>=1e6){

        cout<<"Waktu eksekusi : "<<durationMicro/1e6<<" second";

    }else if(durationMicro>=1e3){

        cout<<"Waktu eksekusi : "<<durationMicro/1e3<<" milisecond";

    }else{

        cout<<"Waktu eksekusi : "<<durationMicro<<" microsecond";

    }

    cout<<endl;

    cout<<"Total perbandingan : "<<cnt<<" kali"<<endl;

    return 0;

}

# BAB 3 HASIL PENGUJIAN

## 3.1 Kasus Ukuran Puzzle Small

### 3.1.1 Uji 1

File input1.txt

J S O L U T I S A B C D E F H

S U N A R U U A A B C D E F A

N E P T U N E T A B P D E F U

S O N I E I S U A B L K E F M

R C E V T R E R A B U D E F E

A H T R A E S N A B T D E F A

M M E R C U R Y A B O D E F G

J S O L U T I S A B C D E F G

S U N A R U U A A E R I S F G

N E P T U N E T A B C D E F G

S O N I E I S U A B C D E F G

R C E V T R E R A B C D E F G

A H T R A E S N A B C D E F G

M M E R C U R Y A B C D E F G

EARTH

JUPITER

MARS

MERCURY

NEPTUNE

SATURN

URANUS

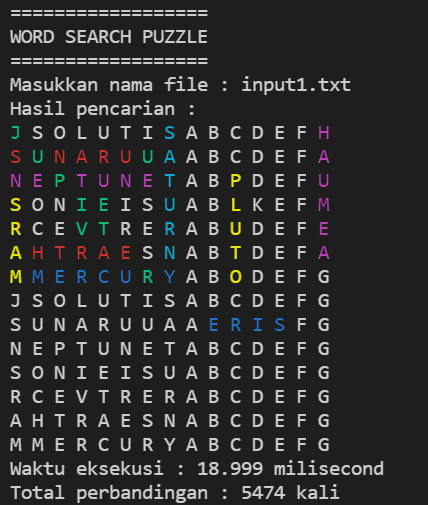
VENUS

PLUTO

ERIS

HAUMEA

Hasil output



### 3.1.2 Uji 2

File input2.txt

U B K D E F K A B C D E F G H

H L H K L M U A B C D E F G H

O P A A S T C A B C D E F G X

V W I R J A I A B C D E F G X

T I K U S A N A B C D E F G X

G N A L E O G A B C D E F G X

A B C D E F G A B K D E F G X

A B C D E F G A B C E E F G X

U B K D E F K A B C D C F G X

H L H K L M U A B C D E O G X

O P A A S T C K B C D E F A X

V W I R J A I U B T A L A L X

T I K U S A N M S E M U T G X

G N A L E O G A B C D E F G X

A B C D E F G Y B C D E F G X

A B C D E F G N B C D E F G X

ULAR

KUCING

TIKUS

ELANG

GAJAH

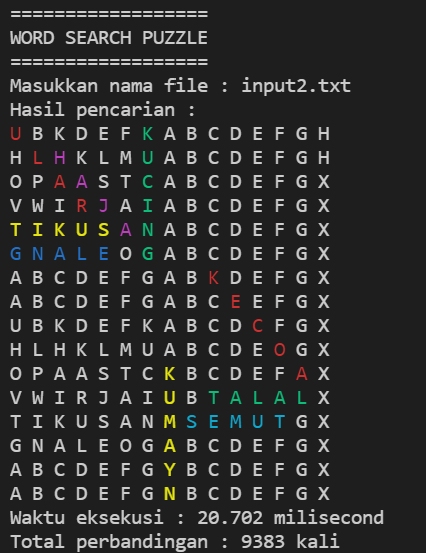
SEMUT

KECOA

LALAT

NYAMUK

Hasil output



### 3.1.3 Uji 3

File input3.txt

A U D D E F K E A B C D E F N

U G H K L M U M A B C D E F A

T P I A A T C P A B C D E S L

A W I T J M I A A B C D E E I

S I K U S A I T A B C D E P B

G N H U J U T L A B C D E U M

G H I M A N E N A B C D E L E

A U D D E F K E A B N D E U S

U G H K L M U M A B A D E H G

T P I A A T C P A B P D E F G

A W I T J M I A A B A D E F G

S I K U S A I T A B L D E F G

G N H U J U T L A B E D E F G

G H I M A N E N A B D D E F G

SATU

DUA

TIGA

EMPAT

LIMA

ENAM

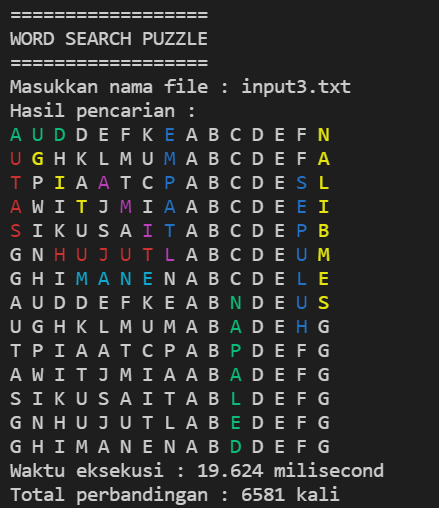
TUJUH

DELAPAN

SEMBILAN

SEPULUH

Hasil output



## 3.2 Kasus Ukuran Puzzle Medium

### 3.2.1 Uji 1

File input4.txt

A U D D E F K E A U D D E D K E A U D D E F K E

U G H K L M U M U G H K L E U M U G H K L M U M

T P I A A T C P T P I A A L C P T P I A A T C P

A W I T J M I A A W I T J A I A A W I T J M I A

S I K U S A I T S I K U S P I T S I K U S A I T

G N H U J U T L G N H U J A T L G N H U J U T L

G H I M A N E N N H I M A N E N G H I M A N E N

A U D D E F K A A U D D E F K E A U D D E F K A

U G H K L M L M U G H K L M U M U G H K L M L M

T P I A A I C P T P I A A T C P T P I A A I C P

A W I T B M I A A W I T J M I A A W I T B M I A

S I K M S E P U L U H U S A I T S I K M S E P U

G N E U J U T L G N H U J U T L G N E U J U T L

G S I M A N E S A L E B E S E N G S I M A N E S

A U D D E F K E A U D D E D K E A U D D E F K E

U G H K L M U M U G H K L E U M U G H K L M U M

T P I A A T C P T P I A A L C P T P I A A T C P

A W I T J M I A A W I T J A I A A W I T J M I A

S I K U S A I T S I K U S P I T S I K U S A I T

G N H U J U T L G N H U J A T L G N H U J U T L

G H I M A N E N N H I M A N E N G H I M A N E N

A U D D E F K A A U D D E F K E A U D D E F K A

SATU

DUA

TIGA

EMPAT

LIMA

ENAM

TUJUH

DELAPAN

SEMBILAN

SEPULUH

SEBELAS

Hasil output



### 3.2.2 Uji 2

File input5.txt

A U D L E F K E A U C D E D P E A U D L E F K E

U G H K O M U M U G O K I E L M U G H K O M U M

E B R A A W C P T P M A A V U G E B R A A W C P

C I I E J M E A A W B L J A S N C I I E J M E A

R N K U E A I R S I I U U P I I R N K U E A I R

O A H U J D T L G N N U J M T M O A H U J D T L

F R I M A N Y N N H A M A N E M F R I M A N Y N

E Y D D E F K A A U T C E F K A E Y D D E F K A

T G H K L M L M B G I I L M U R T G H K L M L M

U P I A A I C R O P O M A T C G U P I A A I C R

R S E A R C H E U W N A J M I O R S E A R C H E

B I K M S E P P N N H N S A I R B I K M S E P P

G N E U J U T P D I H Y J U T P G N E U J U T P

G E L B B U B U A M E D T R O S G E L B B U B U

A U D L E F K E A U C D E D P E A U D L E F K E

U G H K O M U M U G O K I E L M U G H K O M U M

E B R A A W C P T P M A A V U G E B R A A W C P

C I I E J M E A A W B L J A S N C I I E J M E A

R N K U E A I R S I I U U P I I R N K U E A I R

O A H U J D T L G N N U J M T M O A H U J D T L

F R I M A N Y N N H A M A N E M F R I M A N Y N

E Y D D E F K A A U T C E F K A E Y D D E F K A

BRUTEFORCE

GREEDY

BINARY

SEARCH

BUBBLE

SORT

DYNAMIC

PROGRAMMING

MUL

PLUS

MIN

DIV

LOWER

UPPER

BOUND

COMBINATION

Hasil output



### 3.2.3 Uji 3

File input6.txt

O U D L E F F E A U C D S D P E O U D L E F

U G H K O M I O U G O K I E L M U G H K O M

E B R A A W V P U P M A X V U G E B R A A W

C I I E J M E A A R B L S A S N C I I E J M

R N K U T A I R S I I E U P I T R N K U T A

O A H O J D T L G N V U J M T H O A H O J D

F R I W A N Y N N E A M A N E R F R I W A N

E Y D T E F K A N I T C E F K E E Y D T E F

T G E K L M L M B G I I L M U E T G E K L M

U N I A A I C R O H O M A T C G U N I A A I

O S E A R C H E U T N T J M I O O S E A R C

B I K M S E P P N I N E S A I R B I K M S E

G N E U J U T P D I H N J U T P G N E U J U

G E L B B U B U A M E D T R O S G E L B B U

O U D L E F F E A U C D S D P E O U D L E F

U G H K O M I O U G O K I E L M U G H K O M

E B R A A W V P U P M A X V U G E B R A A W

C I I E J M E A A R B L S A S N C I I E J M

R N K U T A I R S I I E U P I T R N K U T A

O A H O J D T L G N V U J M T H O A H O J D

F R I W A N Y N N E A M A N E R F R I W A N

E Y D T E F K A N I T C E F K E E Y D T E F

ONE

TWO

THREE

FOUR

FIVE

SIX

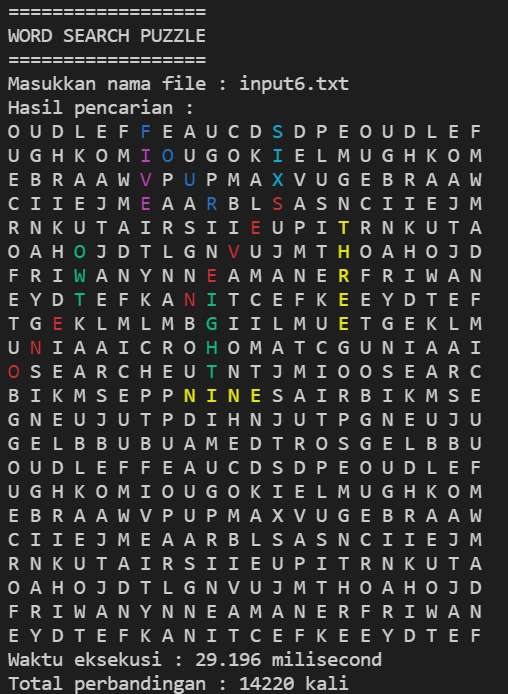
SEVEN

EIGHT

NINE

TEN

Hasil output



## 3.3 Kasus Ukuran Puzzle Large

### 3.3.1 Uji 1

File input7.txt

T N C W R M Y Y L W M M V S R P O A S N U Z I A N M L U J W

E U E B W B W L C K P H C X U V P X U B L D I Y Y U F H S I

J X J F D A R O D U N V S S I I Z N G F N T Z P P Z S T N T

Y O H U O A Q U S B O G X C A T K J I R U O Q U Y V H T B J

Q K D I H L M B C Q G U D G I K E F B E R R F F D P D J R C

R F K M G M Z G R N N G Z E L L U V E K X A L F X Y H T R J

G W B F T T S Q R O N W M P K P H R X N J Y W K W W D J Q Y

V K V B E A L Z B C P S A D D J F D X U B F V P U A U W G I

M D B K Z F K Q K F D S I A U H Y H K M T E X C M K J C V H

A S B X B W I G X I B V H S G D V A L L E O V I P F R M I R

Z P A I S G G X K J S T A P O S Y U K M O T L X C O L C U L

V E G L M S T K E I P T U Z I E U N I Z P P K M G L Y L Q Z

K M A K E Q J B S H U A X Y S P M Q Z W J T J I T Q K K I I

V U Z M V B U Z G E A Q J E F U G W C L P C H I F Z Z Y Y N

D O B L V V E I D P A G M Q B V A U D D V X L L G J F I H C

F H W U Z P T S C M R B I B Q U C V B Q I E E I K P C C L V

G V K U T F A P W V I C D T I P K O H M P E V L V X D F F K

I V R Q O J P N L L D X Z E F T W X U G M J J P U V N E W G

D X X X Q K M X A P J S H U L U P E S L Z E H L F B L V X Z

E Y B V C P E N S O R O L B C I U X O U Y V H V T K K W Y E

L V P F M B W Z Z S I V N O J J L P I N K Z D Q N G K M J J

A H Y L P P T H P O N C X K L D L L L Y C T O A U M W T P B

P H K B S A Z V S V E N H T P P C A U D J H F I N F Q E G D

A L M J X M X V U X K D W Y G N G X B A E V W U U J U G A E

N A N P M F I Q X N K U C E N A M R A E I V F B A X G P D I

K N K D E Z H S F F W E N T G H P A O O S R D H H Z A P B U

I F A I P W V J M K P O R K S V X H Q F O B G K R Z T X C M

V X V L K T F G E S L C I E A W L Z Q H B J V C R E B B M N

D K V U B J U F D H G F T I Y C F D D T B L X L E P W C E Y

D F C P N Y Z Z U G B C I R G K B N G G C Z S T N Q P S U P

SATU

DUA

TIGA

EMPAT

LIMA

ENAM

TUJUH

DELAPAN

SEMBILAN

SEPULUH

SEBELAS

Hasil output



### 3.3.2 Uji 2

File input8.txt

D R I V I W C R V C F R H R B T Z R G W S R G O J J X R J R

T Y Z M B F E C S G U K A D G E V A W J I D M P K E M O Z E

C Z L V U I D T B Z Y R W E S E T R R V B I K U R J T Y Q C

K L F V S D F H D G B K F W G I F E E H W O A R K M K Q Z N

Q S W H Z S K Q U I U H A A E Y W O L Z C I Z Q Y U J F J A

W N E K I H U N L K P F X U H F Y R S G S D E P W U U E K C

J P L G W T C Y E V R J G B Y N H A L I E Y S Z W F I X P P

A C U C Q R C H U P I K U P Y F H B Y U H U L D J L H D A J

E R U C H P V K T A J W X H F A Q B E Q K F S Q S W X G H S

O Q U T K C W T P P M D A L X T E N U N O P Z E Y B K X Y Y

D J R K Z K N Q I E H R Y G T B L F K Q H B M C S K G C G E

K R M V P P W H S U L G U B X T Z V O X B C Q H T W O K L Z

F I L K U D Z O R Y F Z G V M X G P B F Z G I Z S Q G U G A

M Z S X P L G C H G A C Q X O Y B R H Y L E J G E G C P W F

H Y K Z Q K Q B U X Q S E T P O J X G L H Z R T M W O X I V

H W V C W R O Q A H A Q Q S J U I Q Y J W S I R I U S C Y M

O P X R T Y E W F L A T Q I E O L D K U X F I A A B O E B N

D E S C Z Y T U E E U H X B R D P E O H D A C Z C N K N Y F

J S J Y U E W W V Y H V G O E C K V B O G M D V E I W T R Q

U R H N X T M P S E D V K F O Q Q P L L Q L K L Z Z H A L E

N O O Z H T J M O M F S Z R B B C S P G I U E V L P D U D E

G D J J W D U H A V M V U E A V M N U M F E F F B H S R Y I

R V S K A G P W N L R V U I F X K L O T K W F M D I N I E J

I I X Z N O I R O V L V X C R S S D E S B U B L E H B S S B

D J K Z S A R R X W R S Q K H A I H J S P O C P X R E A X K

D W N Z O M J A V P X H N W M R T T D Z L S Z K T D K B Y C

N B L E O Q R R J A J C V Y H K F I W L C U I N A F C F E A

K U D G W N S H W K R I P P J N T K G Z T R O G J G V S L H

P G Q N D M Z O K I K O H O Q N O A I A D M G D D U C C J B

Y N B V L A I L F U D I W J R E V T G R S I W E Z R V T A Q

SIRIUS

BETELGEUSE

CENTAURI

VEGA

ORION

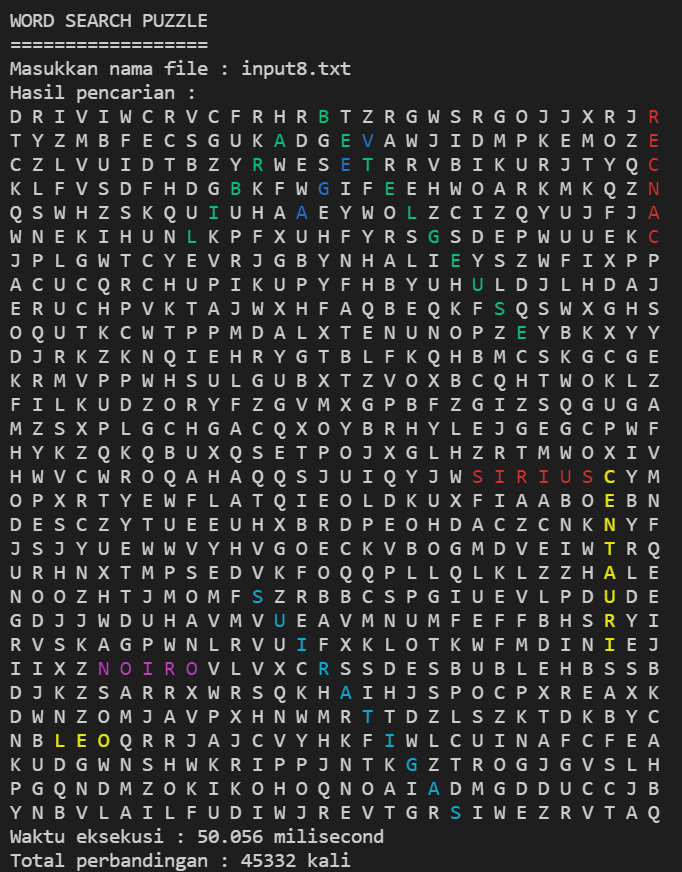
SAGITARIUS

CANCER

LIBRA

LEO

Hasil output



### 3.3.3 Uji 3

File input9.txt

O U D L E F F E A U C D S D P E O U D L E F

K R R P G O P H H O V D W H W H V X R C M V Z Y M U G X O F

B L M W I Z A I B O E W A W Z D G Y X D I B T H E A D B U F

Y V Q C H Z Y T B V Q H Z S W U P J P I H Z U I Z F T U Q N

Q F T O A G O U F A H D X S Z C Z I M W I F I Z W X H I Z F

H N X G E P T P X H E H W R V W H N G T J E P W F D H R H Y

I M N S A E L Z M J G C C L U S R O E V Q P Z J D R U S K Y

S V Q L G W G J K K G X U O N R P A H R Y N Z H U S B C L X

Q A W W Y F C U V S F E D K O I Q L H S M I J N P J S T T Q

Q C M M H Y K R U I R I C O N Y Z I T B U A O Y M Z R M K G

N J V E K E W U N Y L P K K Y E S N B V K H U L X P L L V X

Y D K D F T W J L W G K G Q T U A I U P N H F Y I T Q X C M

Q W Z G C Q K E N P R M R V T R M O R J Z B H I U F U X I Y

F D L D Q X C U B S J M L D U T W C E R K Y Q M L R B W Y J

T Q V E M Z H B T T P X E Q X Y F U I Y W I V N T Y A K W M

S V C A B E P D H D P Y J L F Q K Q R Z E V A P K D Q D X B

H W J O H F P O Y P M T H H N I M Y C K D B X S P V K W P I

R J O F I B I X L F Y O U M E Z T A L K O C O Q Z H B O C R

Z P K K G E D C U U R R D P O X C R Y D B J L Z U F A Q V U

U K O K H G Y Q G J G S Z C H V O A Z F F Y T W F L T E Y S

G V J I U G X H Z C Q H S T U X B E T O W Y O Z K F U D B O

V D J I J G J K J W W N X R H T V U K J M Q X P T E M F Y V

V A Z S Y J O V Z D Q Y F S Z G M K W J R R G L P V C D O F

U V J Q A M V D G L R B P C M W M X U V W Z E O X N L G Z I

N A M H Y J A B X Z U A M H U G G B M U S M Q N R H O V W F

G V F N C H C X O S G W B L M W U L B V I A J C T A M P M Q

U Y I O O G I U Q N G L W L D E I T Z B O Y S B T W N E Q T

T C M P P J Q Z I D F K F O P N U U Y S P W U T X I Q G M T

I Y X H Z E L N M E R A H Q F K Z P X W H Q V S W Z U N E G

A G I S H L U C H B I E V L O E K M Y L I U B A Z Z R E D L

D C Q F J K P L E H K M H N G K W O Y N M Q O O N G G U R F

BIRU

HIJAU

COKLAT

HITAM

PUTIH

KUNING

MERAH

ORANGE

UNGU

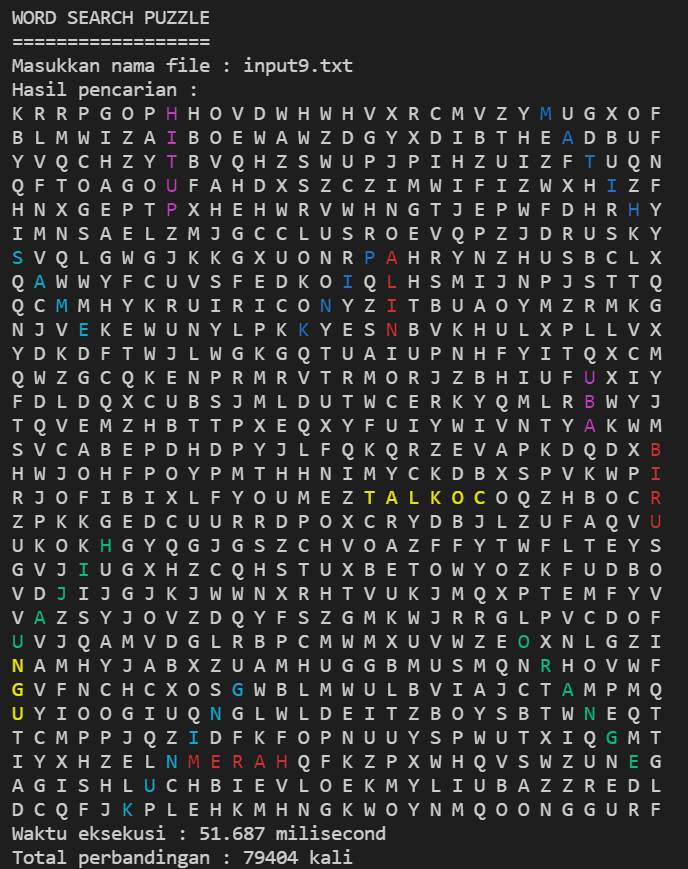
PINK

ABU

EMAS

NILA

Hasil output



# LAMPIRAN

## Alamat Drive

<https://github.com/primahafiz/Tubes-TBFO>

## Cek List Tabel

|  |  |  |
| --- | --- | --- |
| Poin | Ya | Tidak |
| 1. Program berhasil dikompilasi tanpa kesalahan (no syntax error) | ✓ |  |
| 1. Program berhasil running | ✓ |  |
| 1. Program dapat membaca file masukan dan menuliskan luaran. | ✓ |  |
| 1. Program berhasil menemukan semua kata di dalam puzzle. | ✓ |  |