

Laporan Tugas Kecil 1 IF2211 Strategi Algoritma  
Penyelesaian Word Search Puzzle Menggunakan Algoritma  
Brute Force



DIBUAT OLEH  
William Manuel Kurniawan

PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
Semester II Tahun 2021/2022

## Daftar Isi

Bab 1 .....	2
Bab 2 .....	3
Bab 3 .....	5
Bab 4 .....	25
Tabel Cek List .....	41

## **Bab 1 Pendahuluan**

Word search puzzle merupakan sebuah permainan teka-teki dimana pemain diminta untuk mencari kata yang terdapat dalam sebuah tabel berisi huruf-huruf sembarang. Kata yang dicari dalam tabel dapat memiliki berbagai macam arah tergantung dari peraturan permainannya dan dalam kasus ini, kata dapat dicari secara horizontal ke kiri, horizontal ke kanan, vertikal ke atas, vertikal ke bawah, dan semua arah diagonal. Dari peraturan tersebut, pemain harus memeriksa 8 arah berbeda yang dapat menyembunyikan kata yang ingin ditemukan dalam tabel tersebut. Untuk membantu menyelesaikan permainan tersebut, dapat dibuat program yang pada kasus ini menggunakan algoritma brute force.

Algoritma brute force merupakan pendekatan langsung dan sederhana untuk memecahkan suatu masalah. Menggunakan algoritma brute force dalam penyelesaian masalah akan menghasilkan sebuah solusi yang jelas dan mudah dimengerti tetapi bukan merupakan algoritma yang paling baik digunakan dalam komputer. Contoh aplikasi dari algoritma brute force adalah ketika mencocokkan kata. Dapat dibuat algoritma yang akan memeriksa setiap huruf dalam kata sampai kata yang dicocokkan sudah sama dengan yang diminta. Pencocokkan kata tersebut akan digunakan untuk menyelesaikan word search puzzle.

## Bab 2 Teori Dasar Cara Kerja Program

Untuk menyelesaikan word search puzzle, program akan mencocokkan kata dalam sebuah tabel dengan kata yang diminta berdasarkan arah yang diminta. Jika misalnya program diminta untuk menyelesaikan sebuah word search puzzle sebagai berikut :

U	N	I	U	G	N	E	P
L	T	O	A	D	A	F	T
A	I	V	P	T	D	R	A
T	H	O	R	I	N	O	K
A	A	A	N	G	A	G	R
P	E	O	X	E	P	K	E
B	U	Q	G	R	O	O	E
G	I	R	A	F	F	E	M

Gambar 2.1 Contoh word search puzzle yang ingin diselesaikan

lalu program diminta untuk mencari kata “GIRAFFE”, maka program akan mencoba mencari kata tersebut dengan arah horizontal dari kiri ke kanan terlebih dahulu. Berikut adalah contoh dari pencarian tersebut.

U	N	I	U	G	N	E	P
G	I	R	A	F	F	E	
L	T	O	A	D	A	F	T
A	I	V	P	T	D	R	A
T	H	O	R	I	N	O	K
A	A	A	N	G	A	G	R
P	E	O	X	E	P	K	E
B	U	Q	G	R	O	O	E
G	I	R	A	F	F	E	M

Gambar 2.2 Tahap awal penyelesaian word search puzzle dengan arah horizontal kiri ke kanan

Program memeriksa huruf dalam kata dan huruf yang ada dalam tabel dan menentukan bahwa huruf tidak cocok. Karena tidak cocok, program akan membandingkan kata dari awal dengan huruf selanjutnya di tabel.

U	N	I	U	G	N	E	P
	G	I	R	A	F	F	E
L	T	O	A	D	A	F	T
A	I	V	P	T	D	R	A
T	H	O	R	I	N	O	K
A	A	A	N	G	A	G	R
P	E	O	X	E	P	K	E
B	U	Q	G	R	O	O	E
G	I	R	A	F	F	E	M

Gambar 2.3 Tahap penyelesaian word search puzzle dengan arah horizontal kiri ke kanan

Setelah ditentukan bahwa ada huruf yang tidak cocok dengan kata yang dibandingkan, program akan memeriksa baris ke-2 dari tabel tersebut (huruf selanjutnya di baris pertama tidak akan diperiksa karena panjang baris sudah lebih pendek daripada kata sehingga tidak mungkin sesuai dengan kata yang dicari). Hasil akhir dari proses tersebut ketika program sudah menemukan kata yang diinginkan dalam tabel diperlihatkan sebagai berikut.

U	N	I	U	G	N	E	P
L	T	O	A	D	A	F	T
A	I	V	P	T	D	R	A
T	H	O	R	I	N	O	K
A	A	A	N	G	A	G	R
P	E	O	X	E	P	K	E
B	U	Q	G	R	O	O	E
G	I	R	A	F	F	E	M
G	I	R	A	F	F	E	

Gambar 2.4 Tahap akhir penyelesaian word search puzzle ketika kata sudah ditemukan

Dari gambar diatas, ditunjukkan bahwa program sudah membandingkan semua huruf dalam kata dan semua huruf terbukti cocok dengan kata yang dicari. Jika program harus mencari kata “BEAR”, program harus memeriksa semua kata horizontal dari kiri ke kanan terlebih dahulu dan memastikan bahwa tidak ada kata tersebut di arah horizontal, lalu kembali mencari kata “BEAR” dengan arah dengan arah yang berbeda sampai program masuk ke tahap untuk memeriksa kata secara diagonal dari kiri bawah ke kanan atas.

## Bab 3 Source Program

Source program untuk menyelesaikan word search puzzle dibuat menggunakan program C++ dan bagian-bagian penting dari program akan dibahas. Program memiliki 2 versi dengan 1 menggunakan teknik heuristik dan yang lain tidak menggunakan teknik heuristik. Program bekerja dengan menerima input nama file yang berisi tabel dari word search puzzle dan kata yang ingin dicari lalu mengeluarkan posisi kata dalam tabel, jumlah kata yang ditemukan, waktu yang diperlukan, jumlah langkah yang diperlukan, dan perbandingan antara jumlah langkah yang diperlukan dan jumlah huruf dari semua kata. Bagian dari program adalah sebagai berikut.

```
int main(int argc, char* argv){  
    using std::chrono::high_resolution_clock;  
    using std::chrono::duration_cast;  
    using std::chrono::duration;  
    using std::chrono::milliseconds;  
    using std::chrono::nanoseconds;  
    string namafile;  
    namafile = argv[1];  
    ifstream file_matriks(namafile);  
    ifstream file_matriks2(namafile);  
    ...  
}
```

Pertama, program akan menerima nama file yang disimpan dalam string namafile, lalu file akan dibuka dan disimpan ke 2 variabel berbeda yaitu file\_matriks dan file\_matriks2. Bagian berisi std::chrono digunakan untuk menyimpan waktu yang dibutuhkan untuk membandingkan huruf dalam kata dengan huruf dalam tabel.

```
getline(file_matriks,baristemp);
temp = baristemp.length();
col = (baristemp.length()+1)/2;
if (temp != 0){
    row++;
}
while(temp != 0){
    getline(file_matriks,baristemp);
    temp = baristemp.length();
    if (temp != 0){
        row++;
    }
}

getline(file_matriks,baristemp);
temp = baristemp.length();
if (temp != 0){
    wordsearch++;
}
while(temp!=0){
    getline(file_matriks,baristemp);
    temp = baristemp.length();
    if (temp != 0){
        wordsearch++;
    }
}
```

Variabel `file_matriks` digunakan untuk menghitung baris dan kolom pada tabel serta jumlah kata yang dicari.

```
char **data1;
data1 = new char*[row];
for (int temp2 = 0; temp2<row; temp2++){
    data1[temp2] = new char[col];
}

string *data2;
data2 = new string[wordsearch];

int **data3;
data3 = new int*[row];
for (int temp2_2 = 0; temp2_2<row; temp2_2++){
    data3[temp2_2] = new int[col];
}

for(int temp3_2 = 0; temp3_2<row; temp3_2++){
    for(int temp4_2 = 0; temp4_2<col; temp4_2++){
        data3[temp3_2][temp4_2] = 15;
    }
}
```

Dari data baris dan kolom dalam tabel word search puzzle dan jumlah kata yang diminta, dibuat 2 matriks dan 1 array. 2 matriks tersebut yaitu data1 yang menyimpan tabel word search puzzle dan data3 yang menyimpan warna untuk isi dari data1 dengan data3 diisi nilai untuk warna putih. Array data2 menyimpan kata yang ingin dicari oleh program. Setelah disimpan, baris dan kolom tabel, jumlah kata dan jumlah huruf dari semua kata serta isi dari data1 dan data2 akan ditampilkan ulang ke terminal untuk membantu memastikan isi sudah benar.

Tahap selanjutnya adalah mencari kata dalam tabel. Kata akan dicari berdasarkan urutan di dalam file input dan program harus memeriksa 8 arah berbeda. Berikut adalah program untuk memeriksa kata untuk arah horizontal dari kiri ke kanan.



```
for(int data_id = 0; data_id<wordsearch; data_id++){
    simpan = data2[data_id];
    foundfinal = 0;

    // mencari kata horizontal dari kiri ke kanan
    if(foundfinal == 0){
        for(int puzzle_id_y = 0; puzzle_id_y<row; puzzle_id_y++){
            for(int puzzle_id_x = 0; puzzle_id_x<col-simpan.length()+1; puzzle_id_x++){
                found = 1;
                word_id = 0;
                while(word_id<simpan.length() && found == 1){
                    totalstep++;
                    if(simpan[word_id]!=data1[puzzle_id_y][puzzle_id_x+word_id]){
                        found = 0;
                    }
                    word_id++;
                }
                if(found==1){
                    for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
                        data3[puzzle_id_y][puzzle_id_x+word_id_2] = k;
                    }
                    foundfinal = 1;
                    foundcount++;
                    k+=8;
                    if (k > 230){
                        k = 64;
                    }
                }
            }
        }
    }
}
```

Kata yang dicari dari data2 akan disimpan dalam variabel “simpan” sedangkan variabel “foundfinal” digunakan untuk memeriksa jika kata sudah ditemukan dalam sebuah arah. Variabel “puzzle\_id\_y” dan “puzzle\_id\_x” digunakan untuk mengakses data pada tabel huruf word search puzzle dan variabel “word\_id” digunakan untuk mengakses huruf dalam kata yang dicari. Variabel “found” menandakan jika huruf yang diperiksa program masih sesuai dengan kata atau tidak dengan nilai awal 1 menunjukkan bahwa huruf yang dibandingkan masih sama. Jika ada huruf yang berbeda ketika dibandingkan dengan kata, maka nilai “found” akan berubah menjadi 0 dan program akan berhenti memeriksa bagian tabel tersebut dan memeriksa ke bagian selanjutnya. Untuk urutan pemeriksaan, program akan mulai memeriksa dari bagian kiri atas tabel dan bergerak ke kanan sampai panjang bagian tabel yang diperiksa sudah menjadi lebih pendek daripada kata yang dicari sehingga program pindah ke baris yang baru. Untuk memeriksa arah lain, barisan kode untuk memeriksa arah horizontal di atas akan digunakan kembali dengan sedikit perubahan. Perbedaan program yang menggunakan teknik heuristik di atas dengan yang tidak menggunakan teknik tersebut terlihat dengan variabel “foundfinal” yang tidak ditemukan dan variabel “found” tidak akan menghentikan perbandingan huruf dalam tabel dan kata yang dicari. Efek samping dari perubahan tersebut adalah program yang tidak menggunakan teknik heuristik bekerja lebih pelan. Berikut adalah hasil akhir kode yang tidak menggunakan teknik heuristic (SolvePuzzle.cpp).

```
#include <iostream>
#include <fstream>
#include <string>
#include <chrono>
#include <windows.h>
using namespace std;

int main(int argc, char* argv[]){
    using std::chrono::steady_clock;
    using std::chrono::duration_cast;
    using std::chrono::duration;
    using std::chrono::milliseconds;
    using std::chrono::nanoseconds;

    string namafile;
    namafile = argv[1];
    ifstream file_matriks(namafile); // membaca file untuk mencari baris, kolom, jumlah
kata
    ifstream file_matriks2(namafile); // membaca file untuk dimasukkan ke program
    string baris1;
    string baristemp;
    string baristemp2;
```

```
int col;
int temp;
int wordsearch = 0;
int row = 0;
int jumlahhuruf = 0;
int isihuruf;

getline(file_matriks, baristemp);
temp = baristemp.length();
col = (baristemp.length()+1)/2;
if (temp != 0){
    row++;
}

while(temp != 0){
    getline(file_matriks, baristemp);
    temp = baristemp.length();
    if (temp != 0){
        row++;
    }
}

cout << "jumlah kolom : " << col << "\n";
cout << "jumlah baris : " << row << "\n";

getline(file_matriks, baristemp);
temp = baristemp.length();
if (temp != 0){
    wordsearch++;
}
while(temp!=0){
    getline(file_matriks, baristemp);
    temp = baristemp.length();
    if (temp != 0){
        wordsearch++;
    }
}

cout << "jumlah kata : " << wordsearch << "\n";

char **data1;
data1 = new char*[row];
for (int temp2 = 0; temp2<row; temp2++){
    data1[temp2] = new char[col];
}

string *data2;
data2 = new string[wordsearch];
```

```
int **data3;
data3 = new int*[row];
for (int temp2_2 = 0; temp2_2<row; temp2_2++){
    data3[temp2_2] = new int[col];
}

for(int temp3_2 = 0; temp3_2<row; temp3_2++){
    for(int temp4_2 = 0; temp4_2<col; temp4_2++){
        data3[temp3_2][temp4_2] = 15;
    }
}

int tempcol;

for(int temp3 = 0; temp3<row; temp3++){
    getline(file_matriks2,baristemp2);
    tempcol = 0;
    for(int temp4 = 0; temp4<baristemp2.length(); temp4++){
        if (isspace(baristemp2[temp4]) == false){
            data1[temp3][tempcol] = baristemp2[temp4];
            tempcol++;
        }
    }
}

getline(file_matriks2,baristemp2);

for(int temp5 = 0; temp5<wordsearch; temp5++){
    getline(file_matriks2,baristemp2);
    data2[temp5] = baristemp2;
    jumlahhuruf += baristemp2.length();
}

cout << "jumlah huruf : " << jumlahhuruf << "\n";

for (int x = 0; x < row; x++){
    for (int y = 0; y<col; y++){
        cout<<data1[x][y]<<" ";
    }
    cout<<"\n";
}

for (int z=0; z<wordsearch; z++){
    cout<<data2[z]<<"\n";
}

cout << "\n";
```

```
string simpan;
int found; // Digunakan untuk membandingkan huruf
int foundfinal; // Digunakan untuk mengecek jika kata sudah ditemukan
int totalstep = 0;
int k = 35;
int foundcount = 0;
int word_id; // Digunakan untuk menyimpan huruf berapa yang diperiksa

auto t1 = steady_clock::now();
cout << "Hasil Pencarian : ";
for(int data_id = 0; data_id < wordsearch; data_id++){
    simpan = data2[data_id];

    // mencari kata horizontal dari kiri ke kanan

    for(int puzzle_id_y = 0; puzzle_id_y < row; puzzle_id_y++){
        for(int puzzle_id_x = 0; puzzle_id_x < col-simpan.length()+1; puzzle_id_x++){
            found = 1;
            word_id = 0;
            while(word_id < simpan.length()){
                totalstep++;
                if(simpan[word_id] != data1[puzzle_id_y][puzzle_id_x+word_id]){
                    found = 0;
                }
                word_id++;
            }
            if(found == 1){
                for(int word_id_2 = 0; word_id_2 < simpan.length(); word_id_2++){
                    data3[puzzle_id_y][puzzle_id_x+word_id_2] = k;
                }
                foundcount++;
                k+=8;
                if (k > 230){
                    k = 64;
                }
            }
        }
    }

    // mencari kata horizontal dari kanan ke kiri
    for(int puzzle_id_y = 0; puzzle_id_y < row; puzzle_id_y++){
        for(int puzzle_id_x = col-1; puzzle_id_x >= simpan.length()-1; puzzle_id_x--){
            found = 1;
            word_id = 0;
            while(word_id < simpan.length()){
                totalstep++;
                if(simpan[word_id] != data1[puzzle_id_y][puzzle_id_x-word_id]){
                    found = 0;
                }
            }
        }
    }
}
```

```
        }
        word_id++;
    }
    if(found==1){
        for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
            data3[puzzle_id_y][puzzle_id_x-word_id_2] = k;
        }
        foundcount++;
        k+=8;
        if (k > 230){
            k = 64;
        }
    }
}

// mencari kata vertikal dari atas ke bawah
for(int puzzle_id_y = 0; puzzle_id_y<row-simpan.length()+1; puzzle_id_y++){
    for(int puzzle_id_x = 0; puzzle_id_x<col; puzzle_id_x++){
        found = 1;
        word_id = 0;
        while(word_id<simpan.length()){
            totalstep++;
            if(simpan[word_id]!=data1[puzzle_id_y+word_id][puzzle_id_x]){
                found = 0;
            }
            word_id++;
        }
        if(found==1){
            for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
                data3[puzzle_id_y+word_id_2][puzzle_id_x] = k;
            }
            foundcount++;
            k+=8;
            if (k > 230){
                k = 64;
            }
        }
    }
}

// mencari kata vertikal dari bawah ke atas
for(int puzzle_id_y = row-1; puzzle_id_y>=simpan.length()-1; puzzle_id_y--){
    for(int puzzle_id_x = 0; puzzle_id_x<col; puzzle_id_x++){
        found = 1;
        word_id = 0;
        while(word_id<simpan.length()){
            totalstep++;
```

```
        if(simpan[word_id]!=data1[puzzle_id_y-word_id][puzzle_id_x]){
            found = 0;
        }
        word_id++;
    }
    if(found==1){
        for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
            data3[puzzle_id_y-word_id_2][puzzle_id_x] = k;
        }
        foundcount++;
        k+=8;
        if (k > 230){
            k = 64;
        }
    }
}

// mencari kata diagonal dari kiri atas ke kanan bawah
for(int puzzle_id_y = 0; puzzle_id_y<row-simpan.length()+1; puzzle_id_y++){
    for(int puzzle_id_x = 0; puzzle_id_x<col-simpan.length()+1; puzzle_id_x++){
        found = 1;
        word_id = 0;
        while(word_id<simpan.length()){
            totalstep++;
            if(simpan[word_id]!=data1[puzzle_id_y+word_id][puzzle_id_x+word_id]){
                found = 0;
            }
            word_id++;
        }
        if(found==1){
            for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
                data3[puzzle_id_y+word_id_2][puzzle_id_x+word_id_2] = k;
            }
            foundcount++;
            k+=8;
            if (k > 230){
                k = 64;
            }
        }
    }
}

// mencari kata diagonal dari kanan atas ke kiri bawah
for(int puzzle_id_y = 0; puzzle_id_y<row-simpan.length()+1; puzzle_id_y++){
    for(int puzzle_id_x = col-1; puzzle_id_x>=simpan.length()-1; puzzle_id_x--){
        found = 1;
        word_id = 0;
```

```
        while(word_id<simpan.length()){
            totalstep++;
            if(simpan[word_id]!=data1[puzzle_id_y+word_id][puzzle_id_x-word_id]){
                found = 0;
            }
            word_id++;
        }
        if(found==1){
            for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
                data3[puzzle_id_y+word_id_2][puzzle_id_x-word_id_2] = k;
            }
            foundcount++;
            k+=8;
            if (k > 230){
                k = 64;
            }
        }
    }
}

// mencari kata diagonal dari kiri bawah ke kanan atas
for(int puzzle_id_y = row-1; puzzle_id_y>=simpan.length()-1; puzzle_id_y--){
    for(int puzzle_id_x = 0; puzzle_id_x<col-simpan.length()+1; puzzle_id_x++){
        found = 1;
        word_id = 0;
        while(word_id<simpan.length()){
            totalstep++;
            if(simpan[word_id]!=data1[puzzle_id_y-word_id][puzzle_id_x+word_id]){
                found = 0;
            }
            word_id++;
        }
        if(found==1){
            for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
                data3[puzzle_id_y-word_id_2][puzzle_id_x+word_id_2] = k;
            }
            foundcount++;
            k+=8;
            if (k > 230){
                k = 64;
            }
        }
    }
}

// mencari kata diagonal dari kanan bawah ke kiri atas
for(int puzzle_id_y = row-1; puzzle_id_y>=simpan.length()-1; puzzle_id_y--){
    for(int puzzle_id_x = col-1; puzzle_id_x>=simpan.length()-1; puzzle_id_x--){
```



```
        found = 1;
        word_id = 0;
        while(word_id<simpan.length()){
            totalstep++;
            if(simpan[word_id]!=data1[puzzle_id_y-word_id][puzzle_id_x-word_id]){
                found = 0;
            }
            word_id++;
        }
        if(found==1){
            for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
                data3[puzzle_id_y-word_id_2][puzzle_id_x-word_id_2] = k;
            }
            foundcount++;
            k+=8;
            if (k > 230){
                k = 64;
            }
        }
    }
}

auto t2 = steady_clock::now();

cout << "\n";
for(int temp_final_1 = 0; temp_final_1 < row; temp_final_1++){
    for(int temp_final_2 = 0; temp_final_2 < col; temp_final_2++){
        printf("%c[38;5;%dm%c ", 0x1B, data3[temp_final_1][temp_final_2],
data1[temp_final_1][temp_final_2]);
    }
    cout << "\n";
}

auto time = duration_cast<nanoseconds>(t2-t1);
cout << "\e[37mjumlah kata yang ditemukan : " << foundcount << "\n";
printf("%c[%dmwaktu yang dibutuhkan : %d ns\n", 0x1B, 37, time.count());
cout << "langkah yang dibutuhkan : " << totalstep << "\n";
cout << "perbandingan langkah dengan jumlah huruf : " << totalstep / jumlahhuruf <<
"\n";
}
```

Berikut adalah hasil akhir kode yang menggunakan teknik heuristik (SolvePuzzle\_Heuristic.cpp).

```
#include <iostream>
#include <fstream>
#include <string>
#include <chrono>
#include <windows.h>
using namespace std;

int main(int argc, char* argv[]){
    using std::chrono::steady_clock;
    using std::chrono::duration_cast;
    using std::chrono::duration;
    using std::chrono::milliseconds;
    using std::chrono::nanoseconds;

    string namafile;
    namafile = argv[1];
    ifstream file_matriks(namafile); // membaca file untuk mencari baris, kolom, jumlah
kata
    ifstream file_matriks2(namafile); // membaca file untuk dimasukkan ke program
    string baris1;
    string baristemp;
    string baristemp2;
    int col;
    int temp;
    int wordsearch = 0;
    int row = 0;
    int jumlahhuruf = 0;
    int isihuruf;
    getline(file_matriks, baristemp);
    temp = baristemp.length();
    col = (baristemp.length()+1)/2;
    if (temp != 0){
        row++;
    }

    while(temp != 0){
        getline(file_matriks, baristemp);
        temp = baristemp.length();
        if (temp != 0){
            row++;
        }
    }

    cout << "jumlah kolom : " << col << "\n";
    cout << "jumlah baris : " << row << "\n";
```

```
getline(file_matriks,baristemp);
temp = baristemp.length();
if (temp != 0){
    wordsearch++;
}
while(temp!=0){
    getline(file_matriks,baristemp);
    temp = baristemp.length();
    if (temp != 0){
        wordsearch++;
    }
}

cout << "jumlah kata : " << wordsearch << "\n";

char **data1;
data1 = new char*[row];
for (int temp2 = 0; temp2<row; temp2++){
    data1[temp2] = new char[col];
}

string *data2;
data2 = new string[wordsearch];

int **data3;
data3 = new int*[row];
for (int temp2_2 = 0; temp2_2<row; temp2_2++){
    data3[temp2_2] = new int[col];
}

for(int temp3_2 = 0; temp3_2<row; temp3_2++){
    for(int temp4_2 = 0; temp4_2<col; temp4_2++){
        data3[temp3_2][temp4_2] = 15;
    }
}

int tempcol;

for(int temp3 = 0; temp3<row; temp3++){
    getline(file_matriks2,baristemp2);
    tempcol = 0;
    for(int temp4 = 0; temp4<baristemp2.length(); temp4++){
        if (isspace(baristemp2[temp4]) == false){
            data1[temp3][tempcol] = baristemp2[temp4];
            tempcol++;
        }
    }
}
```

```
getline(file_matriks2, baristemp2);

for(int temp5 = 0; temp5 < wordsearch; temp5++){
    getline(file_matriks2, baristemp2);
    data2[temp5] = baristemp2;
    jumlahhuruf += baristemp2.length();
}

cout << "jumlah huruf : " << jumlahhuruf << "\n";

for (int x = 0; x < row; x++){
    for (int y = 0; y < col; y++){
        cout << data1[x][y] << " ";
    }
    cout << "\n";
}

for (int z = 0; z < wordsearch; z++){
    cout << data2[z] << "\n";
}

cout << "\n";

string simpan;
int found; // Digunakan untuk membandingkan huruf
int foundfinal; // Digunakan untuk mengecek jika kata sudah ditemukan
int totalstep = 0;
int k = 35;
int foundcount = 0;
int word_id; // Digunakan untuk menyimpan huruf berapa yang diperiksa

auto t1 = steady_clock::now();
cout << "Hasil Pencarian : ";
for(int data_id = 0; data_id < wordsearch; data_id++){
    simpan = data2[data_id];
    foundfinal = 0;

    // mencari kata horizontal dari kiri ke kanan
    if(foundfinal == 0){
        for(int puzzle_id_y = 0; puzzle_id_y < row; puzzle_id_y++){
            for(int puzzle_id_x = 0; puzzle_id_x < col - simpan.length() + 1; puzzle_id_x++){
                found = 1;
                word_id = 0;
                while(word_id < simpan.length() && found == 1){
                    totalstep++;
                    if(simpan[word_id] != data1[puzzle_id_y][puzzle_id_x + word_id]){
                        found = 0;
                    }
                }
            }
        }
    }
}
```

```
        }
        word_id++;
    }
    if(found==1){
        for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
            data3[puzzle_id_y][puzzle_id_x+word_id_2] = k;
        }
        foundfinal = 1;
        foundcount++;
        k+=8;
        if (k > 230){
            k = 64;
        }
    }
}

}

}

// mencari kata horizontal dari kanan ke kiri
if(foundfinal == 0){
    for(int puzzle_id_y = 0; puzzle_id_y<row; puzzle_id_y++){
        for(int puzzle_id_x = col-1; puzzle_id_x>=simpan.length()-1; puzzle_id_x--
    ){

        found = 1;
        word_id = 0;
        while(word_id<simpan.length() && found == 1){
            totalstep++;
            if(simpan[word_id]!=data1[puzzle_id_y][puzzle_id_x-word_id]){
                found = 0;
            }
            word_id++;
        }
        if(found==1){
            for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
                data3[puzzle_id_y][puzzle_id_x-word_id_2] = k;
            }
            foundfinal = 1;
            foundcount++;
            k+=8;
            if (k > 230){
                k = 64;
            }
        }
    }
}

}

}

// mencari kata vertikal dari atas ke bawah
```

```
if(foundfinal == 0){
    for(int puzzle_id_y = 0; puzzle_id_y<row-simpan.length()+1; puzzle_id_y++){
        for(int puzzle_id_x = 0; puzzle_id_x<col; puzzle_id_x++){
            found = 1;
            word_id = 0;
            while(word_id<simpan.length() && found == 1){
                totalstep++;
                if(simpan[word_id]!=data1[puzzle_id_y+word_id][puzzle_id_x]){
                    found = 0;
                }
                word_id++;
            }
            if(found==1){
                for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
                    data3[puzzle_id_y+word_id_2][puzzle_id_x] = k;
                }
                foundfinal = 1;
                foundcount++;
                k+=8;
                if (k > 230){
                    k = 64;
                }
            }
        }
    }
}

// mencari kata vertikal dari bawah ke atas
if(foundfinal == 0){
    for(int puzzle_id_y = row-1; puzzle_id_y>=simpan.length()-1; puzzle_id_y--){
        for(int puzzle_id_x = 0; puzzle_id_x<col; puzzle_id_x++){
            found = 1;
            word_id = 0;
            while(word_id<simpan.length() && found == 1){
                totalstep++;
                if(simpan[word_id]!=data1[puzzle_id_y-word_id][puzzle_id_x]){
                    found = 0;
                }
                word_id++;
            }
            if(found==1){
                for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
                    data3[puzzle_id_y-word_id_2][puzzle_id_x] = k;
                }
                foundfinal = 1;
                foundcount++;
                k+=8;
                if (k > 230){
```

```
        k = 64;
    }
}
}
}

// mencari kata diagonal dari kiri atas ke kanan bawah
if(foundfinal == 0){
    for(int puzzle_id_y = 0; puzzle_id_y<row-simpan.length()+1; puzzle_id_y++){
        for(int puzzle_id_x = 0; puzzle_id_x<col-simpan.length()+1; puzzle_id_x++){
            found = 1;
            word_id = 0;
            while(word_id<simpan.length() && found == 1){
                totalstep++;
                if(simpan[word_id]!=data1[puzzle_id_y+word_id][puzzle_id_x+word_id]
){
                    found = 0;
                }
                word_id++;
            }
            if(found==1){
                for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
                    data3[puzzle_id_y+word_id_2][puzzle_id_x+word_id_2] = k;
                }
                foundfinal = 1;
                foundcount++;
                k+=8;
                if (k > 230){
                    k = 64;
                }
            }
        }
    }
}

// mencari kata diagonal dari kanan atas ke kiri bawah
if(foundfinal == 0){
    for(int puzzle_id_y = 0; puzzle_id_y<row-simpan.length()+1; puzzle_id_y++){
        for(int puzzle_id_x = col-1; puzzle_id_x>=simpan.length()-1; puzzle_id_x--
){
            found = 1;
            word_id = 0;
            while(word_id<simpan.length() && found == 1){
                totalstep++;
                if(simpan[word_id]!=data1[puzzle_id_y+word_id][puzzle_id_x-
word_id]){
                    found = 0;
                }
            }
        }
    }
}
```

```
        }
        word_id++;
    }
    if(found==1){
        for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
            data3[puzzle_id_y+word_id_2][puzzle_id_x-word_id_2] = k;
        }
        foundfinal = 1;
        foundcount++;
        k+=8;
        if (k > 230){
            k = 64;
        }
    }
}
}
}

// mencari kata diagonal dari kiri bawah ke kanan atas
if(foundfinal == 0){
    for(int puzzle_id_y = row-1; puzzle_id_y>=simpan.length()-1; puzzle_id_y--){
        for(int puzzle_id_x = 0; puzzle_id_x<col-simpan.length()+1; puzzle_id_x++){
            found = 1;
            word_id = 0;
            while(word_id<simpan.length() && found == 1){
                totalstep++;
                if(simpan[word_id]!=data1[puzzle_id_y-
word_id][puzzle_id_x+word_id]){
                    found = 0;
                }
                word_id++;
            }
            if(found==1){
                for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
                    data3[puzzle_id_y-word_id_2][puzzle_id_x+word_id_2] = k;
                }
                foundfinal = 1;
                foundcount++;
                k+=8;
                if (k > 230){
                    k = 64;
                }
            }
        }
    }
}

// mencari kata diagonal dari kanan bawah ke kiri atas
```



```
        if(foundfinal == 0){
            for(int puzzle_id_y = row-1; puzzle_id_y>=simpan.length()-1; puzzle_id_y--){
                for(int puzzle_id_x = col-1; puzzle_id_x>=simpan.length()-1; puzzle_id_x--
            ){
                found = 1;
                word_id = 0;
                while(word_id<simpan.length() && found == 1){
                    totalstep++;
                    if(simpan[word_id]!=data1[puzzle_id_y-word_id][puzzle_id_x-
word_id]){
                        found = 0;
                    }
                    word_id++;
                }
                if(found==1){
                    for(int word_id_2=0; word_id_2<simpan.length(); word_id_2++){
                        data3[puzzle_id_y-word_id_2][puzzle_id_x-word_id_2] = k;
                    }
                    foundfinal = 1;
                    foundcount++;
                    k+=8;
                    if (k > 230){
                        k = 64;
                    }
                }
            }
        }
    }
}

auto t2 = steady_clock::now();

cout << "\n";
for(int temp_final_1 = 0; temp_final_1 < row; temp_final_1++){
    for(int temp_final_2 = 0; temp_final_2 < col; temp_final_2++){
        printf("%c[38;5;%dm%c ", 0x1B, data3[temp_final_1][temp_final_2],
data1[temp_final_1][temp_final_2]);
    }
    cout << "\n";
}

auto time = duration_cast<nanoseconds>(t2-t1);
cout << "\e[37mjumlah kata yang ditemukan : " << foundcount << "\n";
printf("%c[%dmwaktu yang dibutuhkan : %d ns\n", 0x1B, 37, time.count());
cout << "langkah yang dibutuhkan : " << totalstep << "\n";
cout << "perbandingan langkah dengan jumlah huruf : " << totalstep / jumlahhuruf <<
"\n";
}
```

## Bab 4 Hasil Pengujian

```
Hasil Pencarian :
N L Q U G P Y M P Z N T U H H
I I J M O C A O T I F N X I T
N R J H X R L N U B E A R L L
X O Q X S A T G D I J H P N H
N O I L R A N Y J A A P W Y F
H N S B K E K C E L M E X R Z
U B E R P G G Z L K K L P V E
F A E Y G I O I G O N E V K P
R E E V Q R G T T I M O Q F A
M P M F E A I T T E P D M R C
B E F Z T F P A R R T P U O S
E P I O A F K R O T S O J G V
J O R C R E E U C L U D A F K
C F J Z Y G P S U V M W H D U
S W X G O A T G T C M M Q W V
jumlah kata yang ditemukan : 16
waktu yang dibutuhkan : 1033200 ns
langkah yang dibutuhkan : 84200
perbandingan langkah dengan jumlah huruf : 956
```

Gambar 4.1 Hasil pengujian dari file small\_puzzle1.txt tanpa menggunakan teknik heuristik

```
Hasil Pencarian :
M U U K Y W G O S E E B K S
P A S C Q D G Q Z D B B C Q
O N I H R N U Q E E U Z U H
L T O X I I H P U P C N D H
D I D M D E I Q L I O O U T
B Z A L Y L Y A D T C Q B P
J L J N I H T C T N K A B Q
F C F M S Y S Q F E R M L M
R H T M P C G V Z C O N X E
D L V U I S M O W H A L E Z
L D S T D V D R R B C X J Y
I M B W W K B G N F H Q P I
F T G O W B A R C Z H N C I
H T G T R M S Y I Y J F X H
Q I B B E D E P I T N E C Z
P S D Q V B E R T A M M X N
jumlah kata yang ditemukan : 15
waktu yang dibutuhkan : 851200 ns
langkah yang dibutuhkan : 70936
perbandingan langkah dengan jumlah huruf : 921
```

Gambar 4.2 Hasil pengujian dari file small\_puzzle2.txt tanpa menggunakan teknik heuristik

```
Hasil Pencarian :  
B D E A L O K I N B G C O E  
S A I H S X U S A Z H O F C  
O I L J U W W P M K M P F T  
A H H L L J R Y E L T N E B  
A A G I A T O Y O T J W Z Q  
T T K V E M I Z W X J A K U  
C S Q J W C B R I O I S J J  
A U U A Q Q S O F E E M A B  
L L X W H G M B R P Z N N V  
L J M J A O N F N G Z H W G  
I B T K C H N F U E H W N I  
D M B S E Z E D D Q K I X Z  
A N U H U R Q E A N L F N C  
C H A W R H C O P U O A V I  
Z V A A R R G E W J A Z L K  
J K R M E Z T E S L A U N M  
U I O M A W S M F R A U U E  
A G H M I Y B W X R U P Y W  
jumlah kata yang ditemukan : 13  
waktu yang dibutuhkan : 1456900 ns  
langkah yang dibutuhkan : 86208  
perbandingan langkah dengan jumlah huruf : 1002
```

Gambar 4.3 Hasil pengujian dari file small\_puzzle3.txt tanpa menggunakan teknik heuristik

```
Hasil Pencarian :
P C F F O M B P Q T A B L W N W M R K J N H U
N O C I L I S I P R P R N F E S K D E N M Y X
X G B U T I T A N I U M E L M U I N I M U L A
B B O G K B N P D Z W G G A N Q K K U H N B Z
P D R V P P B C N M U Z Y K R X R I M R I T G
C C G T I K A X H M A S X L X G D E U E R Y A
Q W L X R B R R R D I O N M N O S J N B O C
C O I V B F I M N T I K H T A B N A I B U S
G P I O F P C A Q R B N A C Q X S A M R A H I
P C N A H L L G N L N Z S C D N U B I O A R I
N I T R O G E N A K E N I R O U L F U L T E T
Q C Y T B C Z E J N K O M S B M H E Q H F J J
F E M E E V M S M G V U C U X V U X L C V W P
T R U Q R G L I Q A I N F R M E H I K K I Z Q
K D I F Y O E U E C N E G O R D Y H H D Q M P
V S L O L M O M L U K S C P G L I V I T G N F
W H E H L U B A P O T A S S I U M N I I I N G
G X H B I I C F V A L B Z O I Y B E M R S L Y
S F Q O U D J Z Z T G C S H C L L O W M H S E
D F M R M O S U W P Y W C P P F H N N Q S S T
O B L O L S O S U L F U R Q O J P H S X Z X J
W Z U N Z X E O V F V U O F G Q J T P G A V P
jumlah kata yang ditemukan : 22
waktu yang dibutuhkan : 3571400 ns
langkah yang dibutuhkan : 390840
perbandingan langkah dengan jumlah huruf : 2473
```

Gambar 4.4 Hasil pengujian dari file medium\_puzzle1.txt tanpa menggunakan teknik heuristik

```
Hasil Pencarian :
B D P B H V T K D W C W H A V W H D E B X A F N
L K E O V T W T E I O V E G A Z J X T E M Q F
Y R E Y A X T M R E T T I B L W A B C D R U X K
E Z V T O C X Z C I P N K V S E F R I I G S C S
L L E V Z N E O H C X O I W G I E R T J R E Y N
B E D T H V N W E X V I Y O I Z P V E H I M O J
A D N Z H S X A A M V T L P P J Z D M O E E B Y
R F R U S T R A T E D C U Z P P O J E J V N V X
E Q A T O E U O E C Y A F H K A A O N O E T O O
S T H W B W H C D X E F E C Y D H S T Y D S H O
I N O I S S A P M O C S G O V N Z N I X G N Y J
M K W N G C X I Q H U I N N H O J Z U D M G T C
H X G V E R M G K R E T E T H A P P I N E S S P
R E S I G N E D Z F S A V E H O P E L E S S L Q
D F S F N O C C R S L S R N Z Q M J I U M H L E
Q O J T S R C R E L I E F T O T Q E N S A O X Z
J M U N M Y C J Y Q O J K M B V Y G S I V Z S Y
H D E L B U O R T Q Y E A E F R B O U E E I Y L
L W A E G L O O M Y G S A N X Z O J L W Z Z U E
G O C M M S B W B E U S C T M E Z K T O C O D N
S N S D E T A I R U F N I N W Y J O E E P K L O
N M C T Z D E T A T I R R I C S M H D N P S I L
jumlah kata yang ditemukan : 32
waktu yang dibutuhkan : 5268300 ns
langkah yang dibutuhkan : 590688
perbandingan langkah dengan jumlah huruf : 2420
```

Gambar 4.5 Hasil pengujian dari file medium\_puzzle2.txt tanpa menggunakan teknik heuristik

```
Hasil Pencarian :
G J H B H C O N C O R D I X C T I R A D I R B F
T A Z T S R Q O P T V W N H F A U R I U W O G V
Y D J C D X N W C F Q K D N B L R A F D S R S W
Z C O L U M B U S G Q V I D E B A S T T R H R F
S P R I N G F I E L D T A O I Q S P O K S E J C
B Y X C P S L S I L S Q N M E G P N R N D P E G
A O E N X I J H Z U L E A I Z C M F L E C U X H
T L L A G F I N A F E B P B U D E A X X V I S M
O Y L T K L Q X O S K C O R E L T T I L U O T L
N M I S E Z Y N S S K P L F W E T B N Q G M D Y
R P V U F E O A I J K F I K A H N Y E E O L T N
O I H G U O H L T Y C C S H F O U U O M V O X E
U A S U W A O A E Y K A A E Y I U K H E O X U A
G Y A A L P G Q N R C X Z J D M I M P V E E D L
E Z N L A G J M D E R E I L E P T N O M Z F R S
Y U A N P K W M S E L Z W N O T S E L R A H C G
A T N A K C U Y T I C E K A L T L A S E B P X J
S A N T A F E X H H K M H A H Z A V T D T Y B N
jumlah kata yang ditemukan : 22
waktu yang dibutuhkan : 3000200 ns
langkah yang dibutuhkan : 320240
perbandingan langkah dengan jumlah huruf : 1712
```

Gambar 4.6 Hasil pengujian dari file medium\_puzzle3.txt tanpa menggunakan teknik heuristik

```

Hasil Pencarian :
F P H B X N C A L S E U O S M I A C E A E U N Y M Q O A C N G S E X G O
I Z L C G N U A R E A E C A L L Y H P O G R A S T E L I A C E A E P Q T
A K V A P J K D Q X B I W M J S X U M B S N V A D B E R J Z E N G D R L
G O E R R T I H V U C T D O U M H B Z Y I A X M U N M O B C L S K R N G
F E A M T C B Z U W I E U A A N A R T H R I A C E A E X A G R Q Y N F C
D Y E O V Q E G C R F F A A H L J G Z E E B O N W Q X H S S M T K H G X
W S C E H I L L G R Y L O N K J G K M J A B C H A S T Y W O J B N A Z U
N V A A J W H U F R T W H L J A A A T E M D Z V A N G U P U C A H F F P
H S G U E V U X R I E G I W I F N V N M E A E C A N O N N A N E E V U O
G T A B J A R D N A A V R L W A J I Z S F D Y C O C U U O T L K W J A N
L L R Q F N E G E K A H H K Z K C Y A X G R A V A R E C A C E A E W L V
R N A M O U I C T H S L F A L Z Z E C C A P A M B O R E L L A C E A E J
P C P M M A A A A K M P P I E G X R A D E C A L I S M A T A C E A E S W
T Y S A C N X C S N Z L A L U X T N G E A A O E A E C A I R A H C A C N
X O A E Y O H U T O O B F P V T T U F V A O E R B M C K T J U Y B E X I
I L A C U L C V E J K T W K I C P O X Z X E Z X A R B I O O I E L C P O
L E O Y G U G Z R V U Z E H P A A F X M P A C A B C C N I Y U D G A G A
Q P H E F F O N O L G Z A G I O C N X I Q E R R B P E D G D B F Y I C D
A F A R K H A G P A V P B U O P P E E Y C C A A N S I A X Q J C A O C O
Z M O G U X S D E U O S F B H N E B A X P A W E C W B Q E M C M T L K X
E Q J R F K L L I D V O E J G D O R E E E P C V C E V H E X F P Z H I A
E A R M N Y X X A T B A X H C J A P C V A R W E X A A Q C Z H V R P W C
N A A F E I K N C B E I Q D W P S G A O E A R H A M D E I O X H I A V E
T H E A O A T G E C J K N A U X Q G H U C C F M H E P I Q W Z R F E B A
P V W C F H K J A K W P F I O E D A T L A O T V F J E F L F P Y G T C E
G E L C A T G I E H T F N C I Q O K N C O T O Q F E Y J G L P I D O A N
K G S C K I D H U D J M C E V E S K A A Z A H H K I O E L B Y Z A E J S
N L E A K R D E H I F I Q S U J D F R D I H K W S S S O I I O R C U Y Q
A A H W A Z N I Z W O M H L L D O L A M A C V O I E I K M S V A A S A D
E Y B C M C K O N I W H G C W T S P M U G A I Y R E B Q H X I X T M F E
E E A E U U P B B I F X U F V V A W A N Y C Z O Z W W Z S L I A V Q A Z
I N S O U U G N O X T U L L Q C R E Y P A E P T E B M Z A H D E M J V E
A P G T Z I R I D U N C Y K K U T A U V J J X N T W O R J C J W K V A H
Y X Q B A J S H B S S J A W I S B Z F R U B D Y E C A E Q R M A S H T A
jumlah kata yang ditemukan : 31
waktu yang dibutuhkan : 17383900 ns
langkah yang dibutuhkan : 2047760
perbandingan langkah dengan jumlah huruf : 5656

```

Gambar 4.7 Hasil pengujian dari file big\_puzzle1.txt tanpa menggunakan teknik heuristik

```

Hasil Pencarian :
P C A L O P H Y L L A C E A E U Y I C H L O R A N T H A C E A E Q M
R Y L G K R K L W R C E L A S T R A C E A E K H W V X J S X B Z S P
N X M D X C O M P O S I T A E E L C A R Y O C A R A C E A E P J L J
A R E T E C D I K F E H L Q E K A S Q F I G O C U T A T P C P O M Q
H H H R A A V B N K Q A F N A L X E A C Z P Q T H Y X T U Y E W W O
L C L P E P O Z D Z Z U E R E H G I C X I E A E C A C I H C L O C Q
S O U M C R B J Y T U F Z C C F I W P A K S I E T B U R A L U A A O
F M Z P A I L U C Y V L H Z A C A A E M L R T L J C U D M A B T L E
V B Z B I F E C A V S E U C R N U C V A L L D A W U R Q Z N C R A E
E R X C N O A X B B I F T O E K I C A F E D E Y C S P B D T L E M A
Z E X E N L E C O B S S V N C C X T U R Q C T N H E Z W R H C M D E
V T A N A I C R M E G S F V Y S O M Y R Y M A E A L A G F A X S O C
B A O T M A A M B A A U C O L D Q D W C B O V N U C X E H C W Z L A
L C I R E C I W A E E B O L A W H H G J U I P I I R K T F E E X E L
U E Y O L E S S C C A V L V C E Z K O X H T T H A R N P L A U V A U
R A O P R A I A E A E O U U Y A E I A R Q L H A Y A A B F E N T E N
N E H L A E T P A R C S M L M E A B A W S E Q M C L A U L W B T C A
O O C A C N R M E A A C E A W C E L N Y X D Y Y K E L R S Q E O A P
R D U C Z A U Y F P L L L C P A C K L H N E L E E M A A X A N V N M
P U N A T Z C H J P L E L E P E A A R U S A A A M W K E C N C P I A
V Y O C S U U Z F A I T I A T C C J G K C E E E P E N V A E D J L C
C Y N E E Q Q Q N C R H A E J O I S I J Q C C C C W C R L J A P E B
O K I A K P B F B Y Y R C F W D R J S M A A R A V A A Z R R D E M I
R L A E T N H E B N C A E F H O A M P P Y T L I V C B R M C P Q M K
I Q C K R V E A L O H C A M R M C G R J M A K R E V L A H J C A O Z
A Y E D G O O A L M W E E R O Y D A T Z Y M H A H J V E N U Y D C Z
R Z A G G X O Q X O Z A C V U C C Y P A L E E L C V M B H N L G X O
I H E W I G V P F R T E L T Y O E C C E G N K O Y P X M G K A D X G
A X R H A H Q B F I I A D P N Z Q X L C C Y P E R A C E A E X C H S
C H I F R K H N E A K L C Y F U T P L M H P A C I C C A X A Q R T K
E A E C A T C A C C O G R E C G E V S M S M U L R T P V O C S L W Z
A S X B R M F X G E X O E S A S C M M H J A Q A J T Z R Z A G N W P
E K Y J S N G L W A C K T R X E Z S J V M C I C L H X S D T N D N F
S F A P E G G G B E D E I D E A E C A L U S S A R C Y N W Q L X R F
jumlah kata yang ditemukan : 42
waktu yang dibutuhkan : 22373700 ns
langkah yang dibutuhkan : 2632256
perbandingan langkah dengan jumlah huruf : 5052

```

Gambar 4.8 Hasil pengujian dari file big\_puzzle2.txt tanpa menggunakan teknik heuristik



```

Hasil Pencarian :
J R P Y P W K N H I B R G D W V Z V T M P Z I B H A I S K R N O
P E C I M L B O Z B P T Q D M W I Q O O Q W H Z K O T A K S V F
B H L A Z M K R J Y D E S Y K G C D M S B X V H F R U U Y K A E
J T R F P M U E B S P W I M J S P Q X X L Q B O R A N F T Q N J
D I Z Y N U N R P A L E U S P P I H A C F K F V N W I J L L J X
T K J O A X A C X M A T K A O J E V U L Q J U L Y N P W R C A P
T C F M Z N M O E B M A F T K N M F P T N O L O I Q M D P V P K
A T C Z H I Y H W A B C B S E K I H E L B Z Z X Y S X J B F S W
O R G X C M T G I I G I L A R S O S G M R W B V Z K N F S E J G
S I L N N E T F A W T L N N H P P V A A I D C T Q Q B L H F U S
D U R S F H V T G Z O P G A S F R X B C D K F O G H Z V W S Q A
P M E R H C U E Z K N U I C I P E C B E G E Z Z F L U K V D D X
Y P P D H N L S Z B M D O U U V S O I G E U N J K V X Z Y V K B
F H L N S W O B P B S T W F Q X I Y R W H Y Y M X P X L H G L E
M P D I X V Y V I Q C V K G C O D K C R A Z Y M L R P G O A O W
G D H W X Q Y L E H L X J W G G E O P W X H R X D Z V U P N I K
E H M A B T X J S V M S L D T T N S O L I T A I R E C Z I F H O
S N M M F I Y X H C T I R I B H T Y W T A R J P W V T V Q I A E
H J N A S I S U E L E R F W G O G S I S W F K T B R V D U H A I
A O Y I W I N L H U N K J H K M E R C P Z B O D V P Y K E R T Q
M T G R F F C J Q P Q R U I Y L G D U L M Z O N A B X T T E R S
H A E I J R A I E M N A M S E D A P S I Z M H M D Y O R K K G B
W R T O X C Z J R G L O R T S U L C X B I Z S Q O L D Y F Y Y S
V A Y O I E U M B Q T G I S M J D U U N Z G W Q E G F F C D P Q
O C H R B M A G A J H V E G E L H C O N I P X B P H K J S O K S
T C D W L R G Z P Y D I S L B S D V N K G P Z Y K C H M J J N Y
C A X G N Y O U R E F Z B K C A J K C A L B D S F Q M M W V N R
Y B C C H J U R S W O D W U M Y D L O G X E C N E R E F E R P I
D O P V R Q O F H I Z B P R G C W Z O K I N S V I F I F G S A G
C X M O C X A Q E Q X Y D V A K O H O R E I G H T S M G X V F T
R F U A L X J M C Y S R M C W A S R A J Z Y O S W K F B I I Q A
jumlah kata yang ditemukan : 44
waktu yang dibutuhkan : 10033500 ns
langkah yang dibutuhkan : 1187704
perbandingan langkah dengan jumlah huruf : 5793

```

Gambar 4.9 Hasil pengujian dari file big\_puzzle3.txt tanpa menggunakan teknik heuristik

```

Hasil Pencarian :
N L Q U G P Y M P Z N T U H H
I I J M O C A O T I F N X I T
N R J H X R L N U B E A R L L
X O Q X S A T G D I J H P N H
N O I L R A N Y J A A P W Y F
H N S B K E K C E L M E X R Z
U B E R P G G Z L K K L P V E
F A E Y G I O I G O N E V K P
R E E V Q R G T T I M O Q F A
M P M F E A I T T E P D M R C
B E F Z T F P A R R T P U O S
E P I O A F K R O T S O J G V
J O R C R E E U C L U D A F K
C F J Z Y G P S U V M W H D U
S W X G O A T G T C M M Q W V
jumlah kata yang ditemukan : 15
waktu yang dibutuhkan : 235600 ns
langkah yang dibutuhkan : 9439
perbandingan langkah dengan jumlah huruf : 107

```

Gambar 4.10 Hasil pengujian dari file small\_puzzle1.txt menggunakan teknik heuristik

```

Hasil Pencarian :
M U U K Y W G O S E E B K S
P A S C Q D G Q Z D B B C Q
O N I H R N U Q E E U Z U H
L T O X I I H P U P C N D H
D I D M D E I Q L I O O U T
B Z A L Y L Y A D T C Q B P
J L J N I H T C T N K A B Q
F C F M S Y S Q F E R M L M
R H T M P C G V Z C O N X E
D L V U I S M O W H A L E Z
L D S T D V D R R B C X J Y
I M B W W K B G N F H Q P I
F T G O W B A R C Z H N C I
H T G T R M S Y I Y J F X H
Q I B B E D E P I T N E C Z
P S D Q V B E R T A M M X N
jumlah kata yang ditemukan : 13
waktu yang dibutuhkan : 232400 ns
langkah yang dibutuhkan : 8986
perbandingan langkah dengan jumlah huruf : 116

```

Gambar 4.11 Hasil pengujian dari file small\_puzzle2.txt menggunakan teknik heuristik

```
Hasil Pencarian :  
B D E A L O K I N B G C O E  
S A I H S X U S A Z H O F C  
O I L J U W W P M K M P F T  
A H H L L J R Y E L T N E B  
A A G I A T O Y O T J W Z Q  
T T K V E M I Z W X J A K U  
C S Q J W C B R I O I S J J  
A U U A Q Q S O F E E M A B  
L L X W H G M B R P Z N N V  
L J M J A O N F N G Z H W G  
I B T K C H N F U E H W N I  
D M B S E Z E D D Q K I X Z  
A N U H U R Q E A N L F N C  
C H A W R H C O P U O A V I  
Z V A A R R G E W J A Z L K  
J K R M E Z T E S L A U N M  
U I O M A W S M F R A U U E  
A G H M I Y B W X R U P Y W  
jumlah kata yang ditemukan : 13  
waktu yang dibutuhkan : 300200 ns  
langkah yang dibutuhkan : 9021  
perbandingan langkah dengan jumlah huruf : 104
```

Gambar 4.12 Hasil pengujian dari file small\_puzzle3.txt menggunakan teknik heuristik

```
Hasil Pencarian :
P C F F O M B P Q T A B L W N W M R K J N H U
N O C I L I S I P R P R N F E S K D E N M Y X
X G B U T I T A N I U M E L M U I N I M U L A
B B O G K B N P D Z W G G A N Q K K U H N B Z
P D R V P P B C N M U Z Y K R X R I M R I T G
C C G T I K A X H M A S X L X G D E U E R Y A
Q W L X R R B R R R D I O N M N O S J N B O C
C O I V B F I M N T I K H T A B N A I B U S
G P I O F P C A Q R B N A C Q X S A M R A H I
P C N A H L L G N L N Z S C D N U B I O A R I
N I T R O G E N A K E N I R O U L F U L T E T
Q C Y T B C Z E J N K O M S B M H E Q H F J J
F E M E E V M S M G V U C U X V U X L C V W P
T R U Q R G L I Q A I N F R M E H I K K I Z Q
K D I F Y O E U E C N E G O R D Y H H D Q M P
V S L O L M O M L U K S C P G L I V I T G N F
W H E H L U B A P O T A S S I U M N I I I N G
G X H B I I C F V A L B Z O I Y B E M R S L Y
S F Q O U D J Z Z T G C S H C L L O W M H S E
D F M R M O S U W P Y W C P P F H N N Q S S T
O B L O L S O S U L F U R Q O J P H S X Z X J
W Z U N Z X E O V F V U O F G Q J T P G A V P
jumlah kata yang ditemukan : 22
waktu yang dibutuhkan : 396200 ns
langkah yang dibutuhkan : 28808
perbandingan langkah dengan jumlah huruf : 182
```

Gambar 4.13 Hasil pengujian dari file medium\_puzzle1.txt menggunakan teknik heuristik

```
Hasil Pencarian :
G J H B H C O N C O R D I X C T I R A D I R B F
T A Z T S R Q O P T V W N H F A U R I U W O G V
Y D J C D X N W C F Q K D N B L R A F D S R S W
Z C O L U M B U S G Q V I D E B A S T T R H R F
S P R I N G F I E L D T A O I Q S P O K S E J C
B Y X C P S L S I L S Q N M E G P N R N D P E G
A O E N X I J H Z U L E A I Z C M F L E C U X H
T L L A G F I N A F E B P B U D E A X X V I S M
O Y L T K L Q X O S K C O R E L T T I L U O T L
N M I S E Z Y N S S K P L F W E T B N Q G M D Y
R P V U F E O A I J K F I K A H N Y E E O L T N
O I H G U O H L T Y C C S H F O U U O M V O X E
U A S U W A O A E Y K A A E Y I U K H E O X U A
G Y A A L P G Q N R C X Z J D M I M P V E E D L
E Z N L A G J M D E R E I L E P T N O M Z F R S
Y U A N P K W M S E L Z W N O T S E L R A H C G
A T N A K C U Y T I C E K A L T L A S E B P X J
S A N T A F E X H H K M H A H Z A V T D T Y B N
jumlah kata yang ditemukan : 22
waktu yang dibutuhkan : 401600 ns
langkah yang dibutuhkan : 24997
perbandingan langkah dengan jumlah huruf : 133
```

Gambar 4.14 Hasil pengujian dari file medium\_puzzle2.txt menggunakan teknik heuristik

```
Hasil Pencarian :  
B D P B H V T K D W C W H A V W H D E B X A F N  
L K E O V T W T E I O V E G A Z J X T E M Q F  
Y R E Y A X T M R E T T I B L W A B C D R U X K  
E Z V T O C X Z C I P N K V S E F R I I G S C S  
L L E V Z N E O H C X O I W G I E R T J R E Y N  
B E D T H V N W E X V I Y O I Z P V E H I M O J  
A D N Z H S X A A M V T L P P J Z D M O E E B Y  
R F R U S T R A T E D C U Z P P O J E J V N V X  
E Q A T O E U O E C Y A F H K A A O N O E T O O  
S T H W B W H C D X E F E C Y D H S T Y D S H O  
I N O I S S A P M O C S G O V N Z N I X G N Y J  
M K W N G C X I Q H U I N N H O J Z U D M G T C  
H X G V E R M G K R E T E T H A P P I N E S S P  
R E S I G N E D Z F S A V E H O P E L E S S L Q  
D F S F N O C C R S L S R N Z Q M J I U M H L E  
Q O J T S R C R E L I E F T O T Q E N S A O X Z  
J M U N M Y C J Y Q O J K M B V Y G S I V Z S Y  
H D E L B U O R T Q Y E A E F R B O U E E I Y L  
L W A E G L O O M Y G S A N X Z O J L W Z Z U E  
G O C M M S B W B E U S C T M E Z K T O C O D N  
S N S D E T A I R U F N I N W Y J O E E P K L O  
N M C T Z D E T A T I R R I C S M H D N P S I L  
jumlah kata yang ditemukan : 32  
waktu yang dibutuhkan : 616100 ns  
langkah yang dibutuhkan : 44343  
perbandingan langkah dengan jumlah huruf : 181
```

Gambar 4.15 Hasil pengujian dari file medium\_puzzle3.txt menggunakan teknik heuristik

```

Hasil Pencarian :
F P H B X N C A L S E U O S M I A C E A E U N Y M Q O A C N G S E X G O
I Z L C G N U A R E A E C A L L Y H P O G R A S T E L I A C E A E P Q T
A K V A P J K D Q X B I W M J S X U M B S N V A D B E R J Z E N G D R L
G O E R R T I H V U C T D O U M H B Z Y I A X M U N M O B C L S K R N G
F E A M T C B Z U W I E U A A N A R T H R I A C E A E X A G R Q Y N F C
D Y E O V Q E G C R F F A A H L J G Z E E B O N W Q X H S S M T K H G X
W S C E H I L L G R Y L O N K J G K M J A B C H A S T Y W O J B N A Z U
N V A A J W H U F R T W H L J A A A T E M D Z V A N G U P U C A H F F P
H S G U E V U X R I E G I W I F N V N M E A E C A N O N N A N E E V U O
G T A B J A R D N A A V R L W A J I Z S F D Y C O C U U O T L K W J A N
L L R Q F N E G E K A H H K Z K C Y A X G R A V A R E C A C E A E W L V
R N A M O U I C T H S L F A L Z Z E C C A P A M B O R E L L A C E A E J
P C P M M A A A A K M P P I E G X R A D E C A L I S M A T A C E A E S W
T Y S A C N X C S N Z L A L U X T N G E A A O E A E C A I R A H C A C N
X O A E Y O H U T O O B F P V T T U F V A O E R B M C K T J U Y B E X I
I L A C U L C V E J K T W K I C P O X Z X E Z X A R B I O O I E L C P O
L E O Y G U G Z R V U Z E H P A A F X M P A C A B C C N I Y U D G A G A
Q P H E F F O N O L G Z A G I O C N X I Q E R R B P E D G D B F Y I C D
A F A R K H A G P A V P B U O P P E E Y C C A A N S I A X Q J C A O C O
Z M O G U X S D E U O S F B H N E B A X P A W E C W B Q E M C M T L K X
E Q J R F K L L I D V O E J G D O R E E E P C V C E V H E X F P Z H I A
E A R M N Y X X A T B A X H C J A P C V A R W E X A A Q C Z H V R P W C
N A A F E I K N C B E I Q D W P S G A O E A R H A M D E I O X H I A V E
T H E A O A T G E C J K N A U X Q G H U C C F M H E P I Q W Z R F E B A
P V W C F H K J A K W P F I O E D A T L A O T V F J E F L F P Y G T C E
G E L C A T G I E H T F N C I Q O K N C O T O Q F E Y J G L P I D O A N
K G S C K I D H U D J M C E V E S K A A Z A H H K I O E L B Y Z A E J S
N L E A K R D E H I F I Q S U J D F R D I H K W S S S O I I O R C U Y Q
A A H W A Z N I Z W O M H L L D O L A M A C V O I E I K M S V A A S A D
E Y B C M C K O N I W H G C W T S P M U G A I Y R E B Q H X I X T M F E
E E A E U U P B B I F X U F V V A W A N Y C Z O Z W W Z S L I A V Q A Z
I N S O U U G N O X T U L L Q C R E Y P A E P T E B M Z A H D E M J V E
A P G T Z I R I D U N C Y K K U T A U V J J X N T W O R J C J W K V A H
Y X Q B A J S H B S S J A W I S B Z F R U B D Y E C A E Q R M A S H T A
jumlah kata yang ditemukan : 31
waktu yang dibutuhkan : 1205100 ns
langkah yang dibutuhkan : 116609
perbandingan langkah dengan jumlah huruf : 322

```

Gambar 4.16 Hasil pengujian dari file big\_puzzle1.txt menggunakan teknik heuristik



```

Hasil Pencarian :
P C A L O P H Y L L A C E A E U Y I C H L O R A N T H A C E A E Q M
R Y L G K R K L W R C E L A S T R A C E A E K H W V X J S X B Z S P
N X M D X C O M P O S I T A E E L C A R Y O C A R A C E A E P J L J
A R E T E C D I K F E H L Q E K A S Q F I G O C U T A T P C P O M Q
H H H R A A V B N K Q A F N A L X E A C Z P Q T H Y X T U Y E W W O
L C L P E P O Z D Z Z U E R E H G I C X I E A E C A C I H C L O C Q
S O U M C R B J Y T U F Z C C F I W P A K S I E T B U R A L U A A O
F M Z P A I L U C Y V L H Z A C A A E M L R T L J C U D M A B T L E
V B Z B I F E C A V S E U C R N U C V A L L D A W U R Q Z N C R A E
E R X C N O A X B B I F T O E K I C A F E D E Y C S P B D T L E M A
Z E X E N L E C O B S S V N C C X T U R Q C T N H E Z W R H C M D E
V T A N A I C R M E G S F V Y S O M Y R Y M A E A L A G F A X S O C
B A O T M A A M B A A U C O L D Q D W C B O V N U C X E H C W Z L A
L C I R E C I W A E E B O L A W H H G J U I P I I R K T F E E X E L
U E Y O L E S S C C A V L V C E Z K O X H T T H A R N P L A U V A U
R A O P R A I A E A E O U Y A E I A R Q L H A Y A A B F E N T E N
N E H L A E T P A R C S M L M E A B A W S E Q M C L A U L W B T C A
O O C A C N R M E A A C E A W C E L N Y X D Y Y K E L R S Q E O A P
R D U C Z A U Y F P L L L C P A C K L H N E L E E M A A X A N V N M
P U N A T Z C H J P L E L E P E A A R U S A A A M W K E C N C P I A
V Y O C S U U Z F A I T I A T C C J G K C E E E P E N V A E D J L C
C Y N E E Q Q Q N C R H A E J O I S I J Q C C C W C R L J A P E B
O K I A K P B F B Y Y R C F W D R J S M A A R A V A A Z R R D E M I
R L A E T N H E B N C A E F H O A M P P Y T L I V C B R M C P Q M K
I Q C K R V E A L O H C A M R M C G R J M A K R E V L A H J C A O Z
A Y E D G O O A L M W E E R O Y D A T Z Y M H A H J V E N U Y D C Z
R Z A G G X O Q X O Z A C V U C C Y P A L E E L C V M B H N L G X O
I H E W I G V P F R T E L T Y O E C C E G N K O Y P X M G K A D X G
A X R H A H Q B F I I A D P N Z Q X L C C Y P E R A C E A E X C H S
C H I F R K H N E A K L C Y F U T P L M H P A C I C C A X A Q R T K
E A E C A T C A C C O G R E C G E V S M S M U L R T P V O C S L W Z
A S X B R M F X G E X O E S A S C M M H J A Q A J T Z R Z A G N W P
E K Y J S N G L W A C K T R X E Z S J V M C I C L H X S D T N D N F
S F A P E G G G B E D E I D E A E C A L U S S A R C Y N W Q L X R F
jumlah kata yang ditemukan : 42
waktu yang dibutuhkan : 1312000 ns
langkah yang dibutuhkan : 129845
perbandingan langkah dengan jumlah huruf : 249

```

Gambar 4.17 Hasil pengujian dari file big\_puzzle2.txt menggunakan teknik heuristik



```

Hasil Pencarian :
J R P Y P W K N H I B R G D W V Z V T M P Z I B H A I S K R N O
P E C I M L B O Z B P T Q D M W I Q O O Q W H Z K O T A K S V F
B H L A Z M K R J Y D E S Y K G C D M S B X V H F R U U Y K A E
J T R F P M U E B S P W I M J S P Q X X L Q B O R A N F T Q N J
D I Z Y N U N R P A L E U S P P I H A C F K F V N W I J L L J X
T K J O A X A C X M A T K A O J E V U L Q J U L Y N P W R C A P
T C F M Z N M O E B M A F T K N M F P T N O L O I Q M D P V P K
A T C Z H I Y H W A B C B S E K I H E L B Z Z X Y S X J B F S W
O R G X C M T G I I G I L A R S O S G M R W B V Z K N F S E J G
S I L N N E T F A W T L N N H P P V A A I D C T Q Q B L H F U S
D U R S F H V T G Z O P G A S F R X B C D K F O G H Z V W S Q A
P M E R H C U E Z K N U I C I P E C B E G E Z Z F L U K V D D X
Y P P D H N L S Z B M D O U U V S O I G E U N J K V X Z Y V K B
F H L N S W O B P B S T W F Q X I Y R W H Y Y M X P X L H G L E
M P D I X V Y V I Q C V K G C O D K C R A Z Y M L R P G O A O W
G D H W X Q Y L E H L X J W G G E O P W X H R X D Z V U P N I K
E H M A B T X J S V M S L D T T N S O L I T A I R E C Z I F H O
S N M M F I Y X H C T I R I B H T Y W T A R J P W V T V Q I A E
H J N A S I S U E L E R F W G O G S I S W F K T B R V D U H A I
A O Y I W I N L H U N K J H K M E R C P Z B O D V P Y K E R T Q
M T G R F F C J Q P Q R U I Y L G D U L M Z O N A B X T T E R S
H A E I J R A I E M N A M S E D A P S I Z M H M D Y O R K K G B
W R T O X C Z J R G L O R T S U L C X B I Z S Q O L D Y F Y Y S
V A Y O I E U M B Q T G I S M J D U U N Z G W Q E G F F C D P Q
O C H R B M A G A J H V E G E L H C O N I P X B P H K J S O K S
T C D W L R G Z P Y D I S L B S D V N K G P Z Y K C H M J J N Y
C A X G N Y O U R E F Z B K C A J K C A L B D S F Q M M W V N R
Y B C C H J U R S W O D W U M Y D L O G X E C N E R E F E R P I
D O P V R Q O F H I Z B P R G C W Z O K I N S V I F I F G S A G
C X M O C X A Q E Q X Y D V A K O H O R E I G H T S M G X V F T
R F U A L X J M C Y S R M C W A S R A J Z Y O S W K F B I I Q A
jumlah kata yang ditemukan : 31
waktu yang dibutuhkan : 946800 ns
langkah yang dibutuhkan : 88105
perbandingan langkah dengan jumlah huruf : 429

```

Gambar 4.18 Hasil pengujian dari file big\_puzzle3.txt menggunakan teknik heuristik

## Tabel Cek List

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.	√	