# LAPORAN TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA

## Penyelesaian Word Search Puzzle dengan Algoritma Brute Force



## Disusun oleh:

Monica Adelia 13520096

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2022

#### 1. PENJELASAN ALGORITMA BRUTE FORCE

Algoritma *Brute Force* adalah algoritma yang memecahkan masalah dengan pendekatan yang lempang atau langsung (*straightforward*). Algoritma *Brute Force* memecahkan masalah dengan sangat sederhana dan dengan cara yang jelas. Contohnya adalah menghitung nilai factorial dari suatu bilangan bulat tak-negatif (n). Algoritma *brute force* akan mengalikan n buah bilangan, dimulai dari 1 hingga n. Jumlah operasi perkalian yang terjadi pada proses ini adalah n kali sehingga kompleksitas waktu algoritmanya adalah O(n).

Salah satu proses yang dapat diselesaikan dengan Algoritma *Brute Force* adalah pencocokan string (*string matching/pattern matching*). Langkah-langkahnya adalah dengan mensejajarkan *pattern* dengan awal teks. Lalu, ditelusuri mulai dari huruf awal *pattern* sampai akhir. Huruf tersebut dibandingkan dengan karakter di dalam teks sampai semua karakter *pattern* yang dibandingkan cocok (pencarian berhasil) atau ditemukan ketidakcocokan karakter (pencarian belum berhasil). Jika *pattern* tidak menemukan kecocokan dengan teks, *pattern* digeser satu karakter ke kanan dan proses diulangi kembali.

Pencarian jawaban dari sebuah *Word Search Puzzle* dapat diselesaikan dengan Algoritma *Brute Force* pencocokan string. Kata yang harus dicari pada *puzzle* merupakan *pattern* dan *puzzle* merupakan teks yang menjadi pembanding. Kumpulan huruf *puzzle* disimpan sebagai sebuah matriks huruf. Lalu, diiterasi menggunakan proses pencocokan string untuk setiap huruf pada matriks huruf *puzzle*. Proses pencocokan string ini dilakukan dalam delapan arah, yaitu vertikal ke atas, vertikal ke bawah, horizontal ke kanan, horizontal ke kiri, diagonal ke kanan atas, diagonal ke kanan bawah, diagonal ke kiri atas, dan diagonal ke kiri bawah.

#### 2. SOURCE CODE PROGRAM

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <ctype.h>
#include <sys/time.h>
#define ROWS 200
#define COLS 200
#define WORDBANK 200
FILE* filestream;
char puzzle[ROWS][COLS];
int puzzleAnswers[ROWS][COLS];
char words[WORDBANK][ROWS];
char fileName[25];
char buf;
int sumKata, lengthKata;
int row = 0;
int col = 0;
int prevCol = 0;
int i, j, k, l, m, n, o;
int compared = 0;
void proccessFile(){
    char dir[30] = {'.', '.', '/', 't', 'e', 's', 't', '/'};
    printf("Input File Name: ");
    gets(fileName);
    strcat(dir, fileName);
    filestream = fopen(dir, "r");
    if (filestream == NULL) {
        printf("File open error \n");
    }else{
        while(!feof(filestream)) {
            buf = fgetc(filestream);
            if (buf == '\n') {
                if(col == 0){
                    break;
                }else{
                    row++;
                    prevCol = col;
                    col = 0;
                }
            }
            else if(isalnum(buf)) {
                puzzle[col][row] = toupper(buf);
                puzzleAnswers[col][row] = 9;
                col++;
            }
        }
```

```
sumKata = 0;
        lengthKata = 0;
        while(!feof(filestream)) {
            buf = fgetc(filestream);
            if(buf == '\n'){
                words[sumKata][lengthKata] = '.';
                sumKata = sumKata + 1;
                lengthKata = 0;
            }else if(isalnum(buf)){
                words[sumKata][lengthKata] = toupper(buf);
                lengthKata++;
        words[sumKata][lengthKata] = '.';
        sumKata ++;
       printf("\n---P U Z Z L E A N S W E R---\n'");
}
int findWord(int rowKata, int baris, int kolom, int a, int b) {
   int found = 0;
    int colorCode;
    m = 0;
    n = baris; o = kolom;
    while((words[rowKata][m] != '.') && found == 0){
        if(words[rowKata][m] == puzzle[o][n]){
            compared=compared+1;
            n = n + a;
            0 = 0 + b;
            m = m + 1;
        }else{
            compared=compared+1;
            found = 1;
            break;
        }
    }
    m = 0;
    n = baris; o = kolom;
    if(found == 0){
        while((words[rowKata][m] != '.')){
            colorCode = rowKata % 6;
            puzzleAnswers[o][n] = colorCode;
            n = n + a;
            o = o + b;
            m = m + 1;
        }
    return found;
void printf COLOR(const char *s, int code) {
    if(code == 0){
        printf("\033[0m\033[1;31m%s\033[0m", s);
```

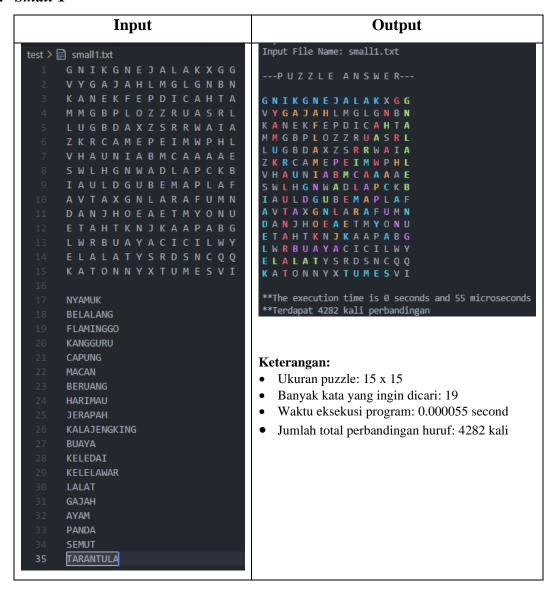
```
}else if(code == 1){
        printf("\033[0m\033[1;32m%s\033[0m", s);
    else if(code == 2){
        printf("\033[0m\033[1;33m%s\033[0m", s);
    }else if(code == 3) {
       printf("\033[0m\033[1;34m%s\033[0m", s);
    }else if(code == 4){
        printf("\033[0m\033[1;35m%s\033[0m", s);
    }else if(code == 5) {
       printf("\033[0m\033[1;36m%s\033[0m", s);
}
void printResult(){
   int colorCode;
    m = 0;
    n = 0;
    for (m=0; m < row; m++) {
        for (n=0; n < prevCol; n++) {
            if(puzzleAnswers[n][m] != 9) {
                if(n == 0 \&\& m != 0) {
                    printf("\n");
                    buf = puzzle[n][m];
                    colorCode = puzzleAnswers[n][m] % 6;
                    printf_COLOR(&buf, colorCode); printf(" ");
                }else{
                    buf = puzzle[n][m];
                    colorCode = puzzleAnswers[n][m] % 6;
                    printf COLOR(&buf, colorCode); printf(" ");
            }else{
                if(n == 0){
                    printf("\n");
                    printf("%c", puzzle[n][m]); printf(" ");
                }else{
                    printf("%c", puzzle[n][m]); printf(" ");
                }
            }
       }
    }
}
int main() {
    int found;
    struct timeval start, end;
    proccessFile();
    gettimeofday(&start, NULL);
    for(i=0; i < sumKata; i++) {</pre>
       found = 0;
        k = 0;
        while (k < row \&\& found == 0) {
```

```
j = 0;
        while(j < prevCol && found == 0){
            if(words[i][0] == puzzle[j][k]){
                if(findWord(i,k,j,0,1) == 0){
                    found = 1;
                }else if(findWord(i,k,j,0,-1) == 0){
                    found = 1;
                }else if(findWord(i,k,j,-1,0) == 0){
                    found = 1;
                }else if(findWord(i,k,j,1,0) == 0){
                    found = 1;
                }else if(findWord(i,k,j,-1,1) == 0){
                    found = 1;
                }else if(findWord(i,k,j,-1,-1) == 0){
                    found = 1;
                }else if(findWord(i,k,j,1,1) == 0){
                    found = 1;
                }else if(findWord(i,k,j,1,-1) == 0){
                    found = 1;
                }
            }else{
               compared++;
            }
            j++;
        k++;
   }
gettimeofday(&end, NULL);
long seconds = (end.tv_sec - start.tv_sec);
long micros = ((seconds * 1000000) + end.tv_usec) - (start.tv_usec);
printResult();
printf("\n\n**The execution time is %d seconds and %d microseconds\n", seconds, micros);
printf("**Terdapat %d kali perbandingan", compared);
return 0;
```

#### 3. SCREENSHOT INPUT DAN OUTPUT

#### 3.1. Data uji yang berukuran small

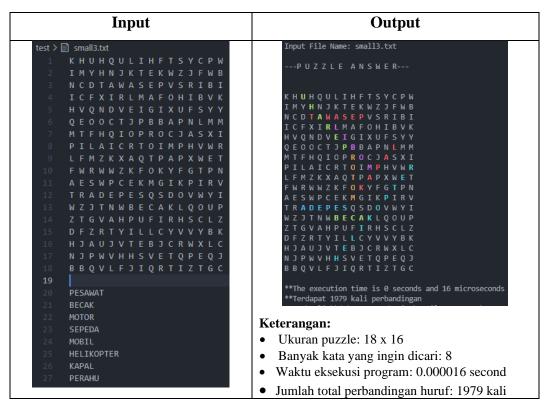
#### 3.1.1. Small 1



#### 3.1.2. Small 2

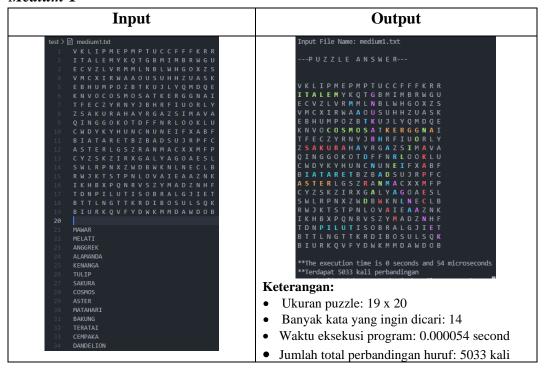
Input	Output
test >	Input File Name: small2.txtPUZZLE ANSWER  EBXGBONWORBQFIAJV PGGQUTRPEAEESLTEY NFKHTICTGSYDSFQRE QEBRZVXPFODEEKYRG EHECBWDIOOATLUDMN HYORUBVNBLUYXLPKA QEWRGLHKUSZLJGOAR ELAUZAMUNVKCBPGWO LPVEICNNRZILWSMIC ARSNVKESZJUIIXJDH
14 T V A C H N W U W U D K N J X V U 15 16 WHITE 17 PURPLE 18 RED 19 PINK 20 GREEN 21 YELLOW 22 ORANGE 23 BLACK 24 BLUE 25 BROWN	YUNGUHTIZEZOMXILJ PPUUQEIWNLQXQZHFA STUHMMHLQZKAXKKYO TVACHNWUWUDKNJXVU  **The execution time is 0 seconds and 20 microseconds **Terdapat 1771 kali perbandingan  Keterangan: Ukuran puzzle: 14 x 17 Banyak kata yang ingin dicari: 10 Waktu eksekusi program: 0.000020 second Jumlah total perbandingan huruf: 1771 kali

#### 3.1.3. *Small* 3

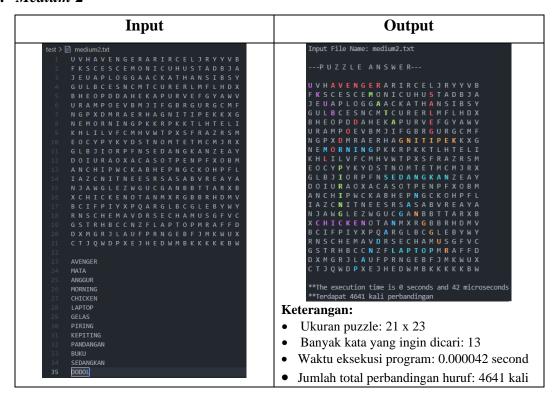


#### 3.2. Data uji yang berukuran medium

#### 3.2.1. *Medium* 1



#### 3.2.2. *Medium* 2

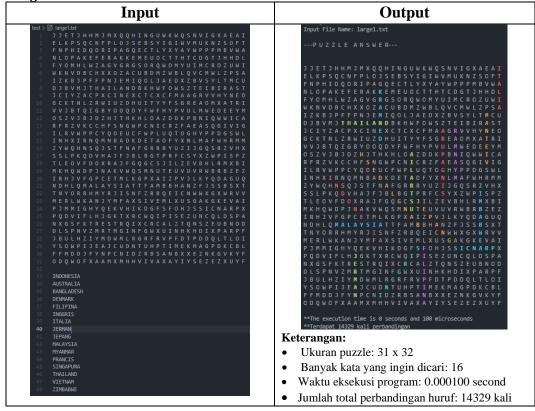


## **3.2.3.** *Medium* **3**

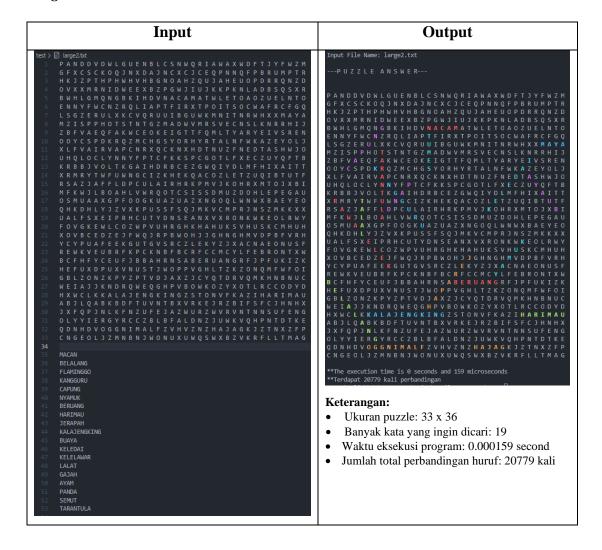
Input	put Output	
test > □ medium3.txt  1	Input File Name: medium3.txt PUZZLE ANSWER  HNAWCEDICYSVKNNUBDJVZD YUKVRNBDPUOZNPFXNPBVPA DKEQTTAMNYPZKHBMQNTIZO YULJWWMAMRSOROMHQVSKAL ISAZGKAGGNAMGGMBXALTOH EZPXJYJQFDNGBCGYNCUPSR WEAGVDMUQAMILEDOXRRBQC CRSZNNODIEWPISOMZONGYN IPSZWWERDJJPGZHISONSIY XYJMAUUCIMNZUZUNEIPNKS QTCSEDHCJRJPZIZRBRFWTA BZBKWSRMUEELDHSNYJLNSE WPEZPICIAXGBRJIVPBAYAO FQWPBSCVZQTLOLPYUTDDQS FJHNJQPWNXASERGVULNQLR OCUYQAXBXVKBVWTBNRBGEI THNIDBMXPIUWIJNSSYHTPK BLNVQYFBXRPTVAAELSDBAA RUGOWACJULLIRBMEGHIYLY RLKNMPXMYIAGSAVCHWDHCA AUUFLADAJOVNBEMIIPENJ BERRVYXFRUGGNAXKJEAPXQ LGERSTHKXJKHUBPTQUEGUV ROJVXHOQYAZCEKIUWNQOAO  **The execution time is 0 seconds and 90 microseconds  **Terdapat 9110 kali perbandingan  Keterangan:  Ukuran puzzle: 24 x 22  Banyak kata yang ingin dicari: 18  Waktu eksekusi program: 0.000090 second  Jumlah total perbandingan huruf: 9110 kali	

## 3.3. Data uji yang berukuran *large*

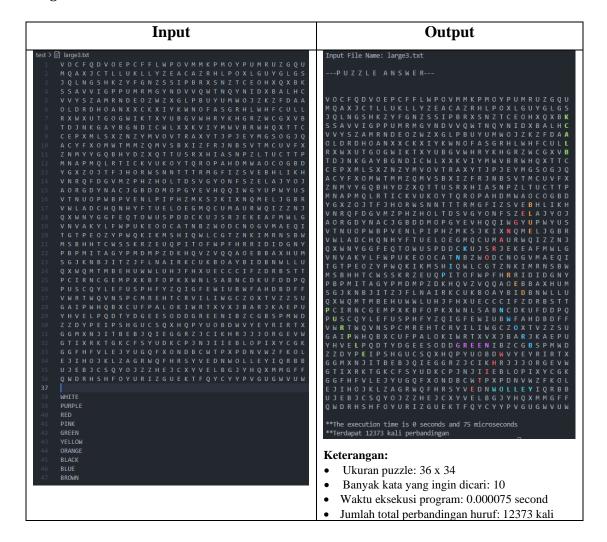
## 3.3.1. *Large* 1



#### 3.3.2. *Large* 2



## 3.3.3. *Large* 3



## 3.3.4. *Large* 4

Input	
	Meth
	SERBIA SE

## **Output** -PUZZLE ANSWER---U Q T H K P M K V J K A L G I E S J E S C E U B J E C G I W F O F M J W E R R U R Y K D V M I N F P H S U P Q F E M P K A I D N A L N I F G G E N Q C J G S O H K M A R P A G U V X C C M I E P U L LDJETRVWJGVD**B**QUQ**E**ASYLQVFYINDTVB**S**STIEHPJQA**K**VNPIWEFX SEJXTILVEBHTIRPZTTAPDQWEM**N**JHWREQ**E**TW**L**NWXMN**O**KCMQJJLL Q K J P O W Y B E Z M Y **A** X **U** U E S D E M T U N X <mark>G</mark> E N W Y Q C X <mark>D</mark> S W **A** J P E L L B G L T C G U T K Y H U Y P S Z I S P A J **E** N A U W L D G I D P U **G** N V O A B P O N **A** O S **R I** S M **O** K D Y B N M J Z R I M G Q Y A C H A J M L N U J O U A W D D B C Z R N D K H Z X F E U L H A T N C M Q G N Y N Q K Y **S** Z I X W M **Z Y** K C X N M L R N N L **S** S C G G N K Y I C E C **N O** P **U** R Z Q T F R E A A N Z A F I E S O U P V X O J T Z T L K U Y U Z R B Y O K A A G L A A E A M W D F P R T X P L U M H C I M L C Q N E I V U Y U C T Z O Y R B E L F I B Z U B L F G E E M T M O U Q S Y S J B A Y H A G L M A L L B M Y U N G A T S L B D R N P P I Q P A R D B E E B Z D L U Y I C A N Y S S R R I W W L V J G I D Q T E U S B Z N S U O D T J V U Y D Y N V Y Q G J K H V N S T A O C B I G C Q N N A V Z H G U C D C O M N C R Z E R A T A Q B O O S X P E G R R T U R K I P O K C X A O W A F O X A G S K C L E V B U V F V I N P K J S N Y A G N B P W E G R X K B G Z U D C K E L L B P I B S Q W R V G J A I L X U P X E M D R D U T B F V U Q U C N A M A Y U A X I H T A S H K C U Z H S T G X E C V L A N V N H N D I W E B K R D Q K B B Y S T **D** C Y E P Z **O** E O T X **R** K T I Z N X T E V G U R R S **S S** T F J P V A V O N M M Q F Y M U J Z G K U K P D T I C Z B Q X F N V H L T H D C G Q C A A C G D M J A I O K L K H H N Y Y R M A N T E I V S I R U M V Z P O D B V F Y **A** W L G W **E** H W L F Z K C R O K L W H M S A S T S D W E M C Y N T K P L T Y H P E Q R B U S H D G X D G B B K Q C L Y S T A K I F Z W K H P L J O G G A G U P W N T Y W K J X R X C J F D Q D X N T A Q Y Y F A S E N Q X N A O A N Y H K F Y T F G Q F I F Z J A U S X V O N X R D A I O I X Y B E Z L I O Q U M T Q F M A L F U G P V G O M L Y B F N W P Q N A D B R Q M V Y S F E A N G A I G L K H U D D F O K Î E N L **K** Q I **R** H G Z P **I** C X K **D** Q J E Î **N** S M M W D D I L Y T E **G** A S A U W N Q L F V P R J D J N N X M W S C G A J Y P D D E O Y K Z K I M C U A O U W B A J D X N P J P T L Q I M W I F H M B X V N U O J P O Z G O N O E E <mark>E</mark> Q S I L N E L K Z O G A **K J E R M A N H J** W G P Z O U I K P A R U R J F Y J **A** A D B E V W **D** I **I** K D X C **G** Q Q Y B **K** E I H B Z **M** K E U J S X G Q M U E A L M G N W J V J K V U L Q R K R O T H I C W R E Y F S L I A R F F O P Z O F G B A R U K U O A I R A N Y A S V Q V P E O Q D O A M K C O C M J T S V K S B A M P A D I E N G B G A I E I H S K E N H T P J X Q F T L N N B V P A K I S T A N T S R D F L R S K A K R G N Y M B E K C I S I S Q T N H Y I N T F M E I F F Q Q R A T R A Z W K T A X H I C D A V A X D A V T M M R V X Z C \*\*The execution time is 0 seconds and 626 microseconds \*\*Terdapat 114261 kali perbandingan

#### Keterangan:

- Ukuran puzzle: 50 x 50
- Banyak kata yang ingin dicari: 55
- Waktu eksekusi program: 0.000626 second
- Jumlah total perbandingan huruf: 114261 kali

## 4. ALAMAT DRIVE KODE PROGRAM

https://github.com/adeliaaaa/Tucil1-STIMA.git

## 5. CEKLIST

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