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

Semester II Tahun 2021/2022

DISUSUN OLEH

Kevin Roni **13520114**

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG

2022

BAB I
ALGORITMA BRUTE FORCE

Deskripsi

Word search puzzle adalah sebuah permainan kata yang mengharuskan pemain untuk menemukan kata yang tersembunyi dalam sebuah papan permainan yang berisikan huruf acak. Pada umumnya *word search puzzle* memiliki papan permainan yang berukuran persegi panjang (termasuk persegi). Dalam penyelesaiannya, *word search puzzle* merupakan sebuah permainan yang dapat dikomputasi untuk penyelesaiannya. Salah satu cara paling umum dalam menyelesaikan permainan ini adalah dengan menggunakan algoritma *brute force*.

Algoritma Brute Force

Pada dasarnya, algoritma *brute force* merupakan algoritma yang *straightforward*, dengan kata lain, penggunaannya dilakukan dengan Teknik yang sangat sederhana dan mudah dipahami. Berikut adalah Langkah-langkah algoritma *brute force* yang saya gunakan.

1. Karena ada 8 arah yang memungkin kata kunci ditemukan pada *puzzle*, maka pencocokan kata kunci akan dilakukan untuk 8 arah.
2. Mula-mula kata kunci yang dicari akan disejajarkan dengan awal matriks (pojok kiri atas) yang berisikan kata acak *puzzle*
3. Misalkan untuk arah kanan mendatar, akan dilakukan pembandingan karakter yang bersesuaian dimulai dari kiri ke kanan hingga:
    - Kata kunci memiliki kecocokan (berhasil)
    - Ada ketidakcocokan
4. Penelusuran akan dilakukan dari kiri ke kanan, jika sudah mencapai akhir matriks (pojok kanan bawah), maka pencocokan akan bergeser 1 langkah ke bawah dan diulang dari kiri ke kanan
5. Jika tidak ditemukan kecocokan hingga akhir matriks, maka program akan mengubah arah sesuai jarum jam (misalkan setelah kanan mendatar akan dilakukan pencocokan ke arah serong kanan bawah) dan mengulang kembali langkah 2.

*notes: untuk arah selain kanan mendatar, pembandingan karakter dilakukan dengan mengikuti arah pencocokan tersebut, jika serong kanan bawah makan pembandingan karakter dilakukan secara serong kanan bawah.

BAB II
Source Code Program

Pada pembuatan algoritma, saya membagi program menjadi 3 bagian, yaitu file configuration, algoritma pencarian, main

1. File configuration

Bagian ini berisikan program untuk membaca input teks dan mengassign teks tersebut menjadi map permainan dan kata kunci yang akan dicari. Secara garis besar, program akan membaca map terlebih dahulu. Ketika mendeteksi adanya line kosong maka program akan berhenti membaca map dan mulai membaca keyword.

```c
#include "boolean.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>

#define ROW_SIZE 50
#define COL_SIZE 50

typedef char ElType;
typedef struct {
    ElType contents[ROW_SIZE][COL_SIZE];
    int rowEff;
    int colEff;
} Matrix;

#define ROW(M) (M).rowEff
#define COL(M) (M).colEff
#define ELMT(M, i, j) (M).contents[(i)][(j)]

typedef struct {
    ElType contents[50][50];
    int neff;
} List;

#define NEFF(L) (L).neff
#define ELMTL(L, i) (L).contents[(i)]

Matrix map;
Matrix result;
List words;
char fileName[50];
```

```c
int count = 0;

void file_config(){
    printf("Input nama file: ");
    scanf("%s", &fileName);

    FILE* fp = fopen(fileName, "r");

    char c = fgetc(fp);
    char currc = c;
    ROW(map) = 0;
    COL(map) = 0;
    int brs = 0;
    int kol = 0;
    while (c!='\n' || currc!='\n'){
        if ((c!=' ') && (c!='\n')){
            ELMT(map, brs, kol) = c;
            kol++;
        }
        if (c=='\n'){
            ROW(map)++;
            brs++;
            COL(map) = kol;
            kol = 0;
        }
        currc = c;
        c = fgetc(fp);
    }

    ROW(result) = ROW(map);
    COL(result) = COL(map);

    NEFF(words) = 0;
    char line[50];
    while (fgets(line, sizeof(line), fp)){
        strcpy(ELMTL(words, NEFF(words)), line);
        NEFF(words)++;
    }

    for (int i = 0; i < NEFF(words)-1; i++){
        ELMTL(words, i)[strlen(ELMTL(words, i))-1] = '\0';
    }
    fclose(fp);
}
```

2. Algoritma pencarian

Bagian ini berisikan program algoritma *bruteforce* untuk mencocokan kata kunci dan memiliki 8 subprogram, yaitu pencarian sesuai dengan 8 arah kemungkinan kata kunci ditemukan

```
boolean search_east(int n){
    int currLen = strlen(ELMTL(words,n));
    for (int i = 0; i < ROW(map); i ++){
        for (int j = 0; j< COL(map); j++){
            int m = 0;
            count++;
            while ((m<currLen)&&(ELMT(map,i,j+m)==ELMTL(words,n)[m])){
                m++;
                count++;
            }
            if (m==currLen){
                for (int a = 0; a <ROW(result); a++){
                    for (int b = 0; b<COL(result); b++){
                        ELMT(result, a, b) = '-';
                    }
                }
                while (m != 0){
                    ELMT(result, i, j)=ELMT(map, i, j);
                    j++;
                    m--;
                }
                for (int i = 0; i <ROW(map); i++){
                    for (int j = 0; j<COL(map); j++){
                        printf("%c ",(ELMT(result, i, j)));
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
        }
    }
    return false;
}

boolean search_southeast(int n){
    int currLen = strlen(ELMTL(words,n));
```

```
    for (int i = 0; i < ROW(map); i ++){
        for (int j = 0; j< COL(map); j++){
            int m = 0;
            count++;
            while ((m<currLen)&&(ELMT(map,i+m,j+m)==ELMTL(words,n)[m])){
                m++;
                count++;
            }
            if (m==currLen){
                for (int a = 0; a <ROW(result); a++){
                    for (int b = 0; b<COL(result); b++){
                        ELMT(result, a, b) = '-';
                    }
                }
                while (m != 0){
                    ELMT(result, i, j)=ELMT(map, i, j);
                    i++;
                    j++;
                    m--;
                }
                for (int i = 0; i <ROW(map); i++){
                    for (int j = 0; j<COL(map); j++){
                        printf("%c ",(ELMT(result, i, j)));
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
        }
    }
    return false;
}

boolean search_south(int n){
    int currLen = strlen(ELMTL(words,n));
    for (int i = 0; i < ROW(map); i ++){
        for (int j = 0; j< COL(map); j++){
            int m = 0;
            count++;
            while ((m<currLen)&&(ELMT(map,i+m,j)==ELMTL(words,n)[m])){
                m++;
                count++;
            }
            if (m==currLen){
```

```
                for (int a = 0; a <ROW(result); a++){
                    for (int b = 0; b<COL(result); b++){
                        ELMT(result, a, b) = '-';
                    }
                }
                while (m != 0){
                    ELMT(result, i, j)=ELMT(map, i, j);
                    i++;
                    m--;
                }
                for (int i = 0; i <ROW(map); i++){
                    for (int j = 0; j<COL(map); j++){
                        printf("%c ",(ELMT(result, i, j)));
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
        }
    }
    return false;
}

boolean search_southwest(int n){
    int currLen = strlen(ELMTL(words,n));
    for (int i = 0; i < ROW(map); i ++){
        for (int j = 0; j< COL(map); j++){
            int m = 0;
            count++;
            while ((m<currLen)&&(ELMT(map,i+m,j-m)==ELMTL(words,n)[m])){
                count++;
                m++;
            }
            if (m==currLen){
                for (int a = 0; a <ROW(result); a++){
                    for (int b = 0; b<COL(result); b++){
                        ELMT(result, a, b) = '-';
                    }
                }
                while (m != 0){
                    ELMT(result, i, j)=ELMT(map, i, j);
                    i++;
                    j--;
                    m--;
```

```
                }
                for (int i = 0; i <ROW(map); i++){
                    for (int j = 0; j<COL(map); j++){
                        printf("%c ",(ELMT(result, i, j)));
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
        }
    }
    return false;
}

boolean search_west(int n){
    int currLen = strlen(ELMTL(words,n));
    for (int i = 0; i < ROW(map); i ++){
        for (int j = 0; j< COL(map); j++){
            int m = 0;
            count++;
            while ((m<currLen)&&(ELMT(map,i,j-m)==ELMTL(words,n)[m])){
                count++;
                m++;
            }
            if (m==currLen){
                for (int a = 0; a <ROW(result); a++){
                    for (int b = 0; b<COL(result); b++){
                        ELMT(result, a, b) = '-';
                    }
                }
                while (m != 0){
                    ELMT(result, i, j)=ELMT(map, i, j);
                    j--;
                    m--;
                }
                for (int i = 0; i <ROW(map); i++){
                    for (int j = 0; j<COL(map); j++){
                        printf("%c ",(ELMT(result, i, j)));
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
```

```
        }
    }
    return false;
}

boolean search_northwest(int n){
    int currLen = strlen(ELMTL(words,n));
    for (int i = 0; i < ROW(map); i ++){
        for (int j = 0; j< COL(map); j++){
            int m = 0;
            count++;
            while ((m<currLen)&&(ELMT(map,i-m,j-m)==ELMTL(words,n)[m])){
                count++;
                m++;
            }
            if (m==currLen){
                for (int a = 0; a <ROW(result); a++){
                    for (int b = 0; b<COL(result); b++){
                        ELMT(result, a, b) = '-';
                    }
                }
                while (m != 0){
                    ELMT(result, i, j)=ELMT(map, i, j);
                    i--;
                    j--;
                    m--;
                }
                for (int i = 0; i <ROW(map); i++){
                    for (int j = 0; j<COL(map); j++){
                        printf("%c ",(ELMT(result, i, j)));
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
        }
    }
    return false;
}

boolean search_north(int n){
    int currLen = strlen(ELMTL(words,n));
    for (int i = 0; i < ROW(map); i ++){
        for (int j = 0; j< COL(map); j++){
```

```
            int m = 0;
            count++;
            while ((m<currLen)&&(ELMT(map,i-m,j)==ELMTL(words,n)[m])){
                count++;
                m++;
            }
            if (m==currLen){
                for (int a = 0; a <ROW(result); a++){
                    for (int b = 0; b<COL(result); b++){
                        ELMT(result, a, b) = '-';
                    }
                }
                while (m != 0){
                    ELMT(result, i, j)=ELMT(map, i, j);
                    i--;
                    m--;
                }
                for (int i = 0; i <ROW(map); i++){
                    for (int j = 0; j<COL(map); j++){
                        printf("%c ",(ELMT(result, i, j)));
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
        }
    }
    return false;
}

boolean search_northeast(int n){
    int currLen = strlen(ELMTL(words,n));
    for (int i = 0; i < ROW(map); i ++){
        for (int j = 0; j< COL(map); j++){
            int m = 0;
            count++;
            while ((m<currLen)&&(ELMT(map,i-m,j+m)==ELMTL(words,n)[m])){
                count++;
                m++;
            }
            if (m==currLen){
                for (int a = 0; a <ROW(result); a++){
                    for (int b = 0; b<COL(result); b++){
                        ELMT(result, a, b) = '-';
```

```
                    }
                }
                while (m != 0){
                    ELMT(result, i, j)=ELMT(map, i, j);
                    i--;
                    j++;
                    m--;
                }
                for (int i = 0; i <ROW(map); i++){
                    for (int j = 0; j<COL(map); j++){
                        printf("%c ",(ELMT(result, i, j)));
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
        }
    }
    return false;
}
```

3. main

File ini berisikan algoritma program utama yang memanggil fungsi pembacaan file, algoritma *brute force* pencocokan serta output hasil, waktu, serta jumlah perbandingan ke layer. Perlu diperhatikan, waktu yang digunakan adalah asumsi program pada device yang penulis gunakan menggunakan satuan ms. Waktu yang diberikan adalah waktu running program secara keseluruhan. Jumlah perbandingan kata yang penulis gunakan adalah operasi perbandingan yang dilakukan program hingga semua kata selesai diproses.

```
int main(){
    file_config();
    printf("\n");
    printf("berikut adalah puzzle yang akan diselesaikan: \n");
    for (int i = 0; i <ROW(map); i++){
        for (int j = 0; j<COL(map); j++){
            printf("%c ",(ELMT(map, i, j)));
        }
        printf("\n");
    }
    printf("\n");
    printf("\nberikut adalah kata yang akan dicari: \n");
    for (int i = 0; i <= NEFF(words); i++){
        printf("%s\n", ELMTL(words, i));
```

```
    }
    clock_t start_time = clock();
    for (int n = 0; n < NEFF(words); n++){
        search_east(n);
        search_north(n);
        search_northeast(n);
        search_northwest(n);
        search_south(n);
        search_southeast(n);
        search_southwest(n);
        search_west(n);
    }
    clock_t end_time = clock();
    printf("Jumlah perbandingan kata yang dicari: %d", count);
    printf("\n");
    double total = end_time-start_time;
    printf("Waktu running algoritma: %f detik", total/1000);
}
```

4. **boolean.h**

File ini berisikan struktur data baru, yaitu boolean

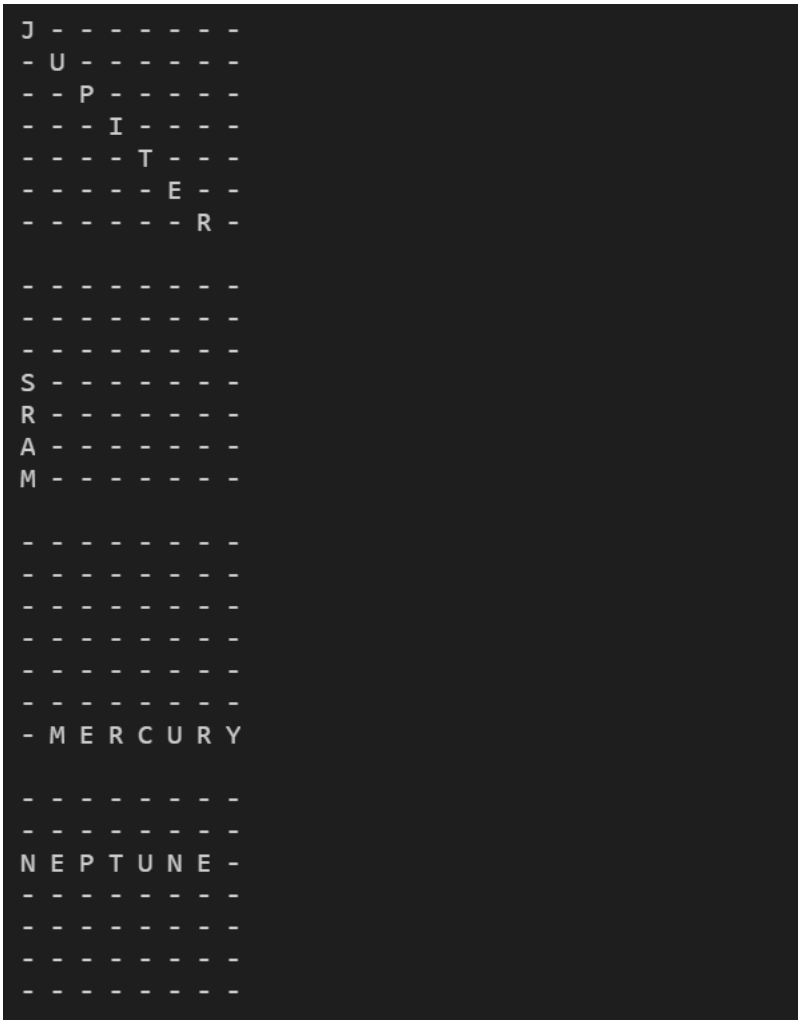
```
/* Definisi type boolean */

#ifndef _BOOLEAN_h
#define _BOOLEAN_h

#define boolean unsigned char
#define true 1
#define false 0

#endif
```

BAB III
Input dan Output

Pada input dan output ini, program saya sengaja dibatasi untuk jumlah kata yang dicari tidak terlalu banyak karena jumlah kata yang akan dicari hanya diassign maksimal 50 dan juga karena keterbatasan pada terminal (dan menghemat space pada laporan), pada bagian pengujian large akan terpotong beberapa bagian (tidak semuanya ditampilkan).

Contoh 1(default):

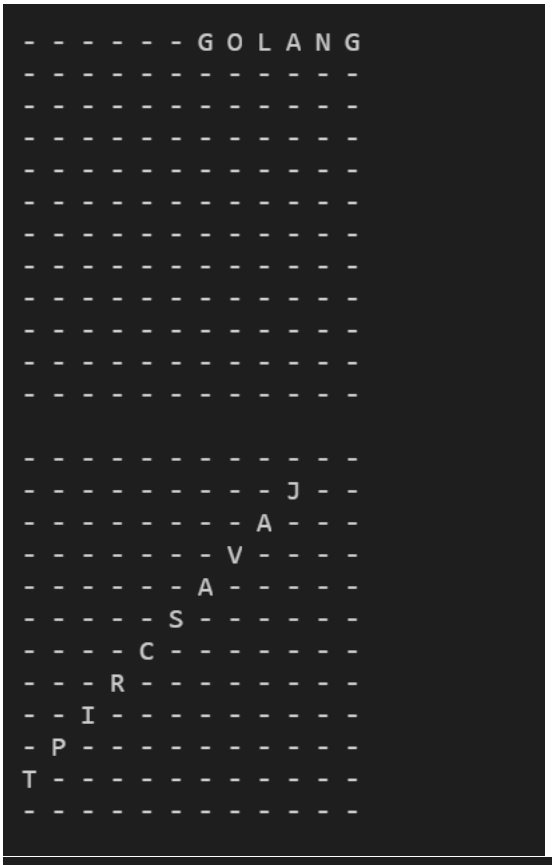
```
Input nama file: test.txt

berikut adalah puzzle yang akan diselesaikan:
J S O L U T I S
S U N A R U U A
N E P T U N E T
S O N I E I S U
R C E V T R E R
A H T R A E S N
M M E R C U R Y


berikut adalah kata yang akan dicari:
EARTH
JUPITER
MARS
MERCURY
NEPTUNE
SATURN
URANUS
VENUS

- - - - - - - -
- - - - - - - -
- - - - - - - -
- - - - - - - -
- - - - - - - -
- H T R A E - -
- - - - - - - -
```

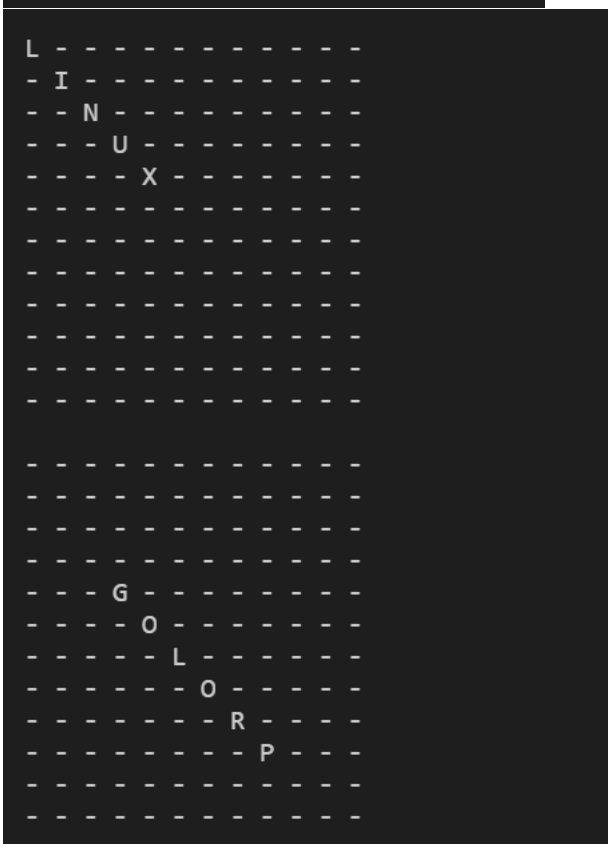

J - - - - - - -
- U - - - - - -
- - P - - - - -
- - - I - - - -
- - - - T - - -
- - - - - E - -
- - - - - - R -
- - - - - - - -
- - - - - - - -
- - - - - - - -
S - - - - - - -
R - - - - - - -
A - - - - - - -
M - - - - - - -
- - - - - - - -
- - - - - - - -
- - - - - - - -
- - - - - - - -
- - - - - - - -
- - - - - - - -
- M E R C U R Y
- - - - - - - -
- - - - - - - -
N E P T U N E -
- - - - - - - -
- - - - - - - -
- - - - - - - -
- - - - - - - -

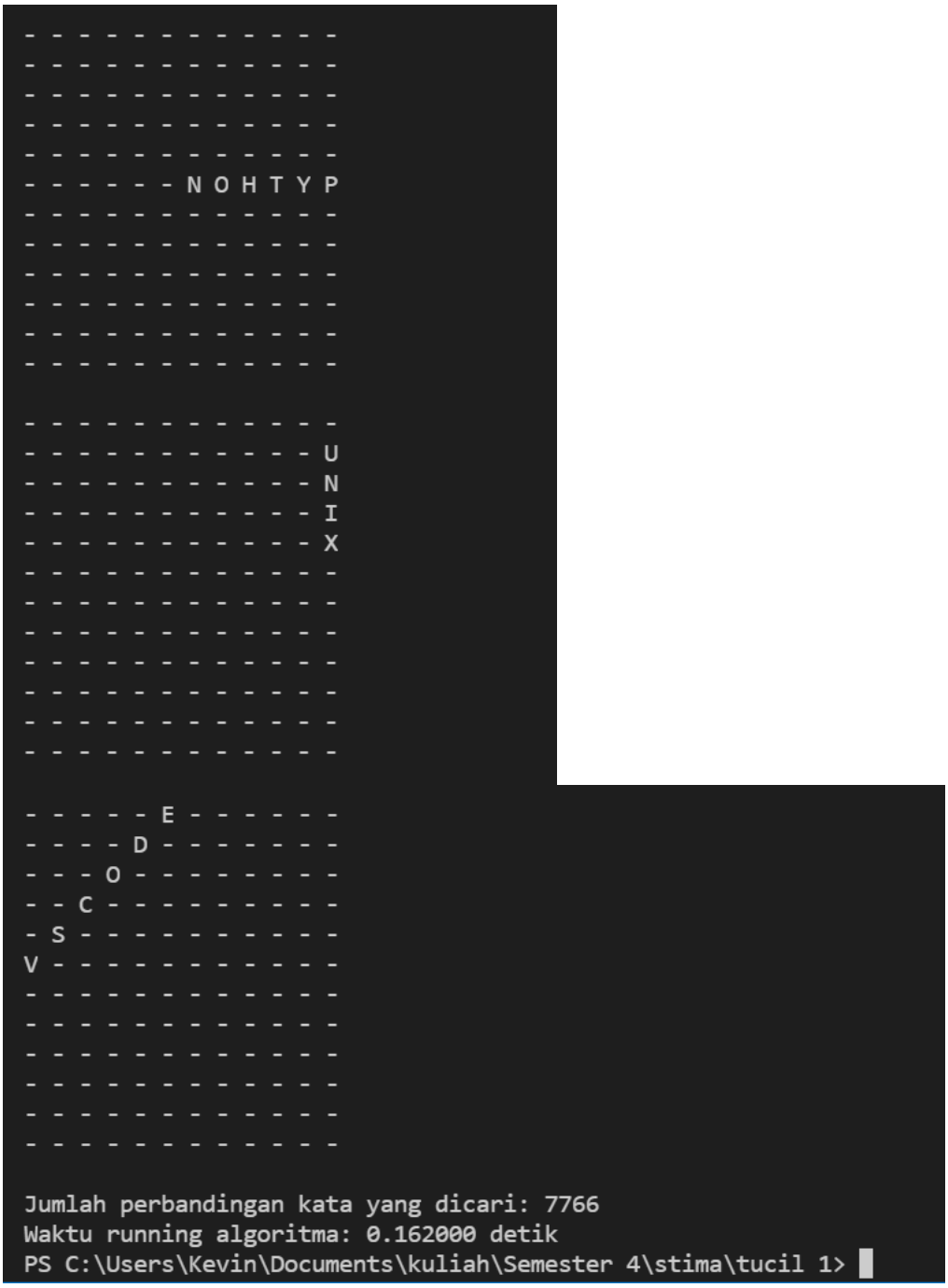
```
- - - - - - - - S
- - - - - - - - A
- - - - - - - - T
- - - - - - - - U
- - - - - - - - R
- - - - - - - - N
- - - - - - - - -

- - - - - - - - -
S U N A R U - - -
- - - - - - - - -
- - - - - - - - -
- - - - - - - - -
- - - - - - - - -
- - - - - - - - -

- - - - - - - - S
- - - - - - - U -
- - - - - N - - -
- - - - E - - - -
- - - V - - - - -
- - - - - - - - -
- - - - - - - - -

Jumlah perbandingan kata yang dicari: 3643
Waktu running algoritma: 0.036000 detik
PS C:\Users\Kevin\Documents\kuliah\Semester 4\stima\tucil 1>
```

Contoh 2(small):

```
PS C:\Users\Kevin\Documents\kuliah\Semester 4\sti
 main.c -o main } ; if ($?) { .\main }
Input nama file: test3.txt

berikut adalah puzzle yang akan diselesaikan:
L U F C W E G O L A N G
I I K O D O B V K J Y U
W L N O Q P T N A M B N
H N C U R E C V Y X Z I
X S P G X L A M I O G X
V J B Z O S N O H T Y P
V R M E C L H P Q U A T
K U E R Y E O P O Q L W
P Y I C C I K R D K M Z
B P Z S G E D I P U S Z
T H B I I S Y U E J E H
P M O H P G T A G H G K


berikut adalah kata yang akan dicari:
GOLANG
JAVASCRIPT
LINUX
PROLOG
PYTHON
UNIX
VSCODE
```

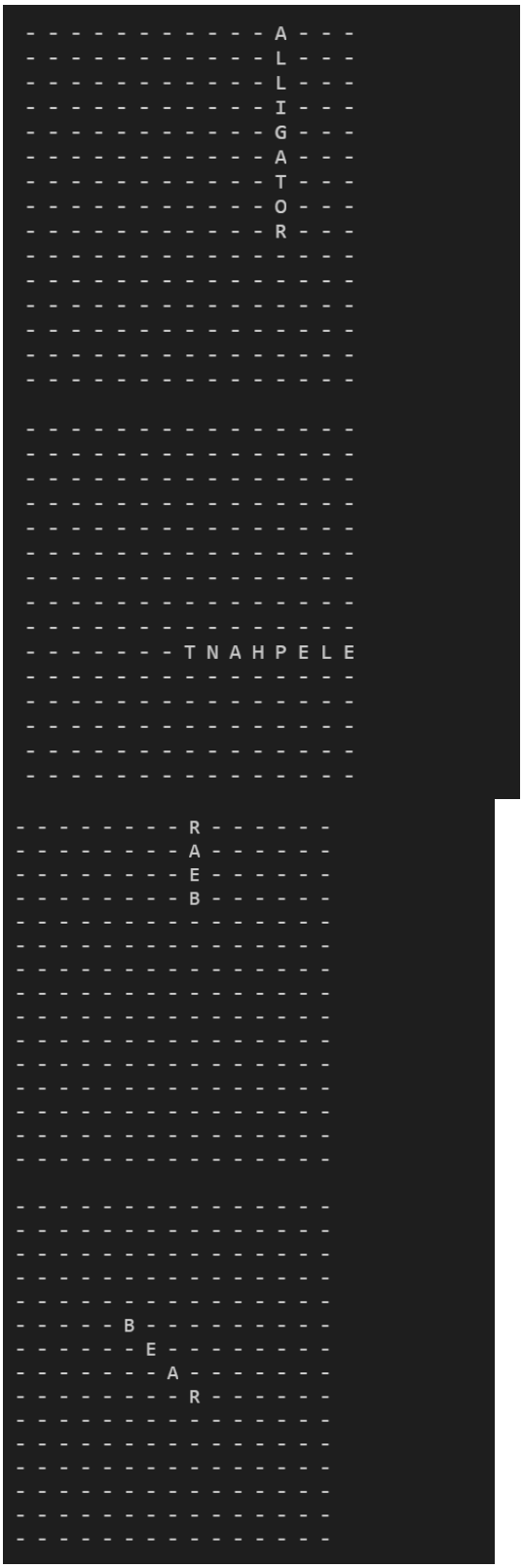

- - - - - - G O L A N G
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - J - -
- - - - - - - - A - - -
- - - - - - - V - - - -
- - - - - - A - - - - -
- - - - - S - - - - - -
- - - - C - - - - - - -
- - - R - - - - - - - -
- - I - - - - - - - - -
- P - - - - - - - - - -
T - - - - - - - - - - -
- - - - - - - - - - - -

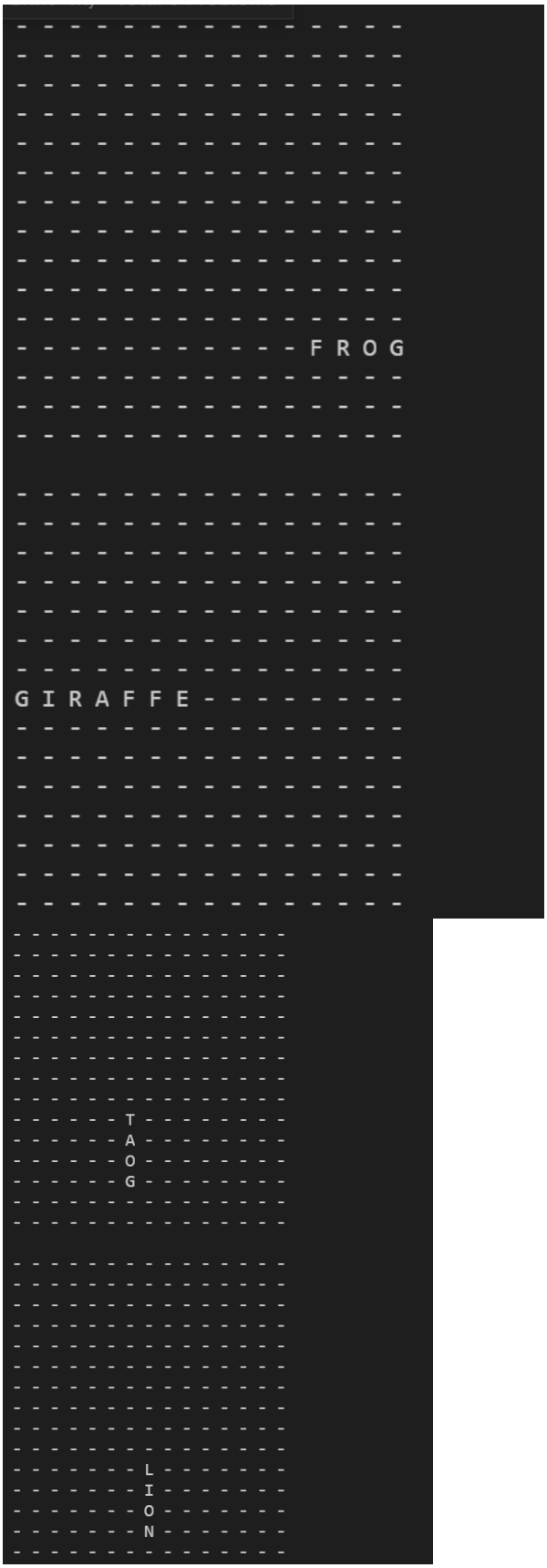

L - - - - - - - - - - -
- I - - - - - - - - - -
- - N - - - - - - - - -
- - - U - - - - - - - -
- - - - X - - - - - - -
- - - - - - - - - - - -
- - - - - - - - - - - -
- - - - - - - - - - - -
- - - - - - - - - - - -
- - - - - - - - - - - -
- - - - - - - - - - - -
- - - - - - - - - - - -
- - - G - - - - - - - -
- - - - O - - - - - - -
- - - - - L - - - - - -
- - - - - - O - - - - -
- - - - - - - R - - - -
- - - - - - - - P - - -
- - - - - - - - - - - -
- - - - - - - - - - - -

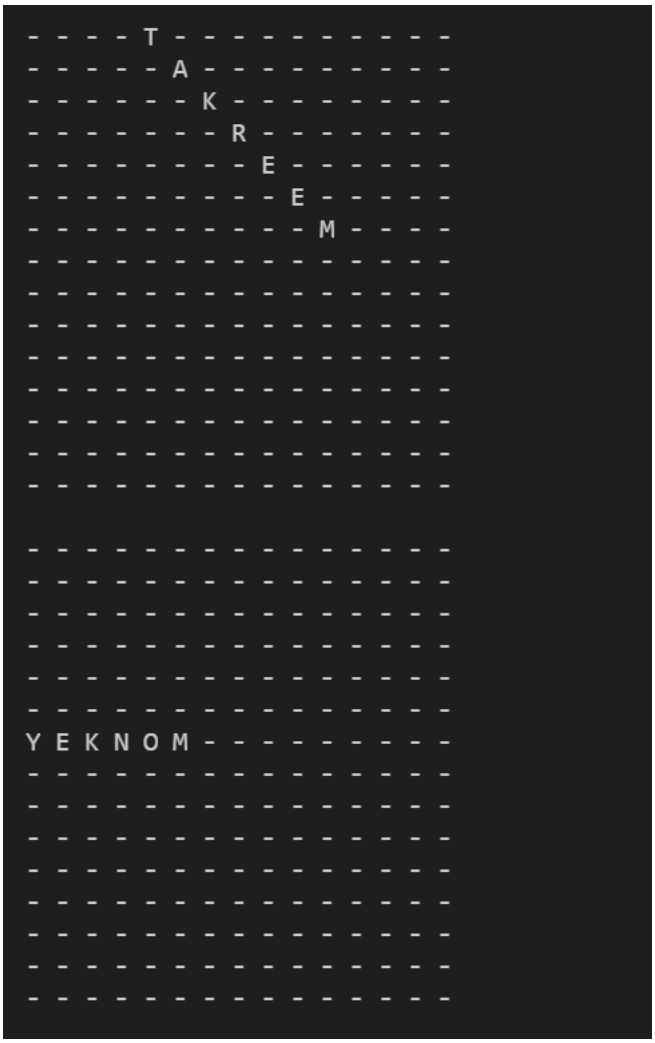


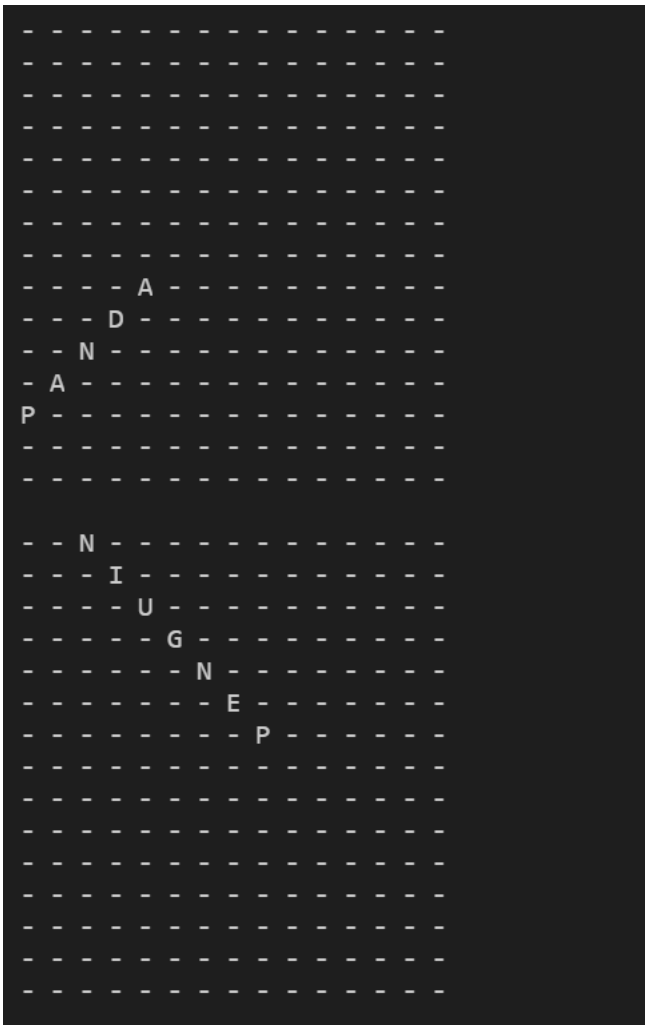
Contoh 3(small):

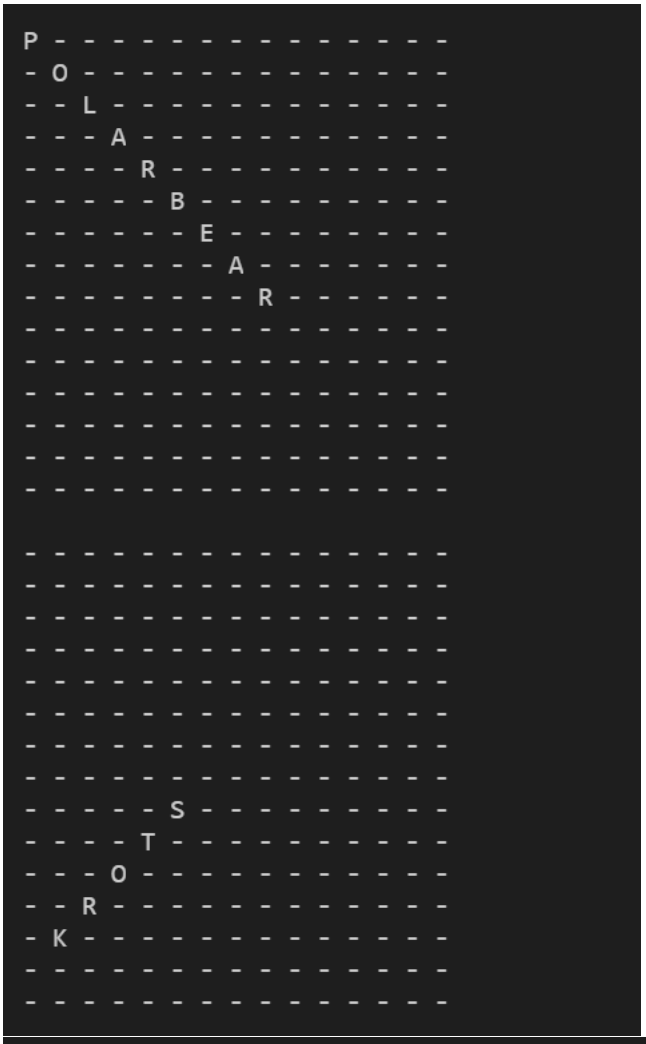
```
Input nama file: test2.txt

berikut adalah puzzle yang akan diselesaikan:
P V N I T T V D R Y E A W Q K
G O N I I A D I A X K L Q Q U
W X L G U V K P E L U L S R O
I V E A Q G E R B K B I M B O
A R D P R F N Q E P R G F X X
E Z O H J B N E H E R A H W Y
Y E K N O M E Y P R M T B M Z
G I R A F F E A H X M O W N S
D I T I A S G X R Q I R B V D
O A J D T Y T T N A H P E L E
K A N O C D A L O L U T G R Q
W A R I A H O I M A M F R O G
P K G O O J G O I H Q Y F S N
O E T P U Z R N F G V Q R A J
W N R Q X Y X C I W A S D E Z


berikut adalah kata yang akan dicari:
ALLIGATOR
ELEPHANT
BEAR
FROG
GIRAFFE
GOAT
LION
MEERKAT
MONKEY
PANDA
PENGUIN
POLARBEAR
STORK
TIGER
TOAD
```

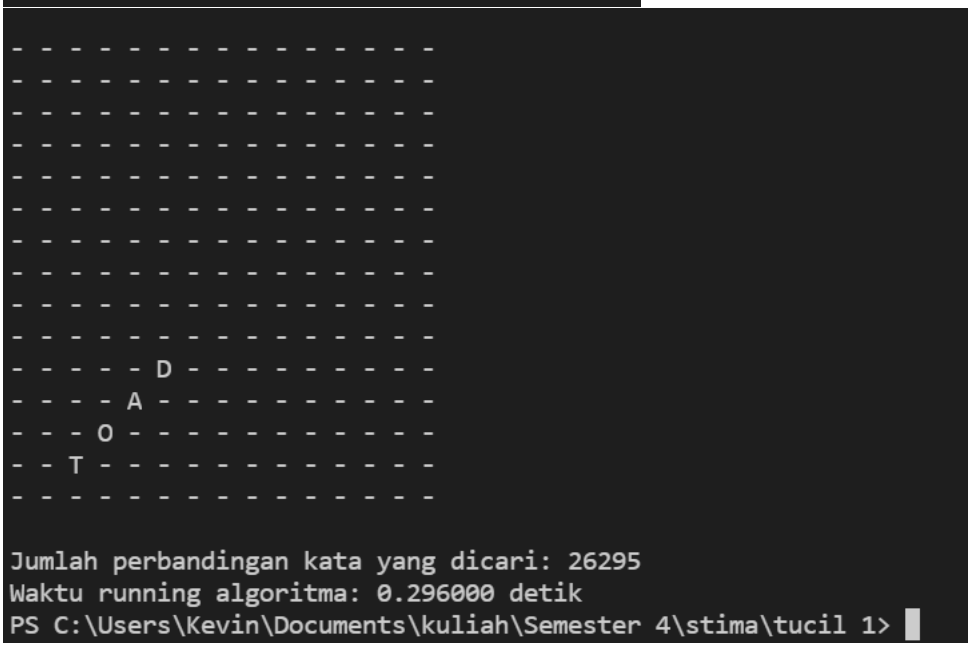

ALLIGATOR
THANKFUL
RABBIT
BEAR


FROG
GIRAFFE
TAG
LION


T
A
K
R
E
E
M
Y E K N O M


A
D
N
A
P
N
I
U
G
N
E
P

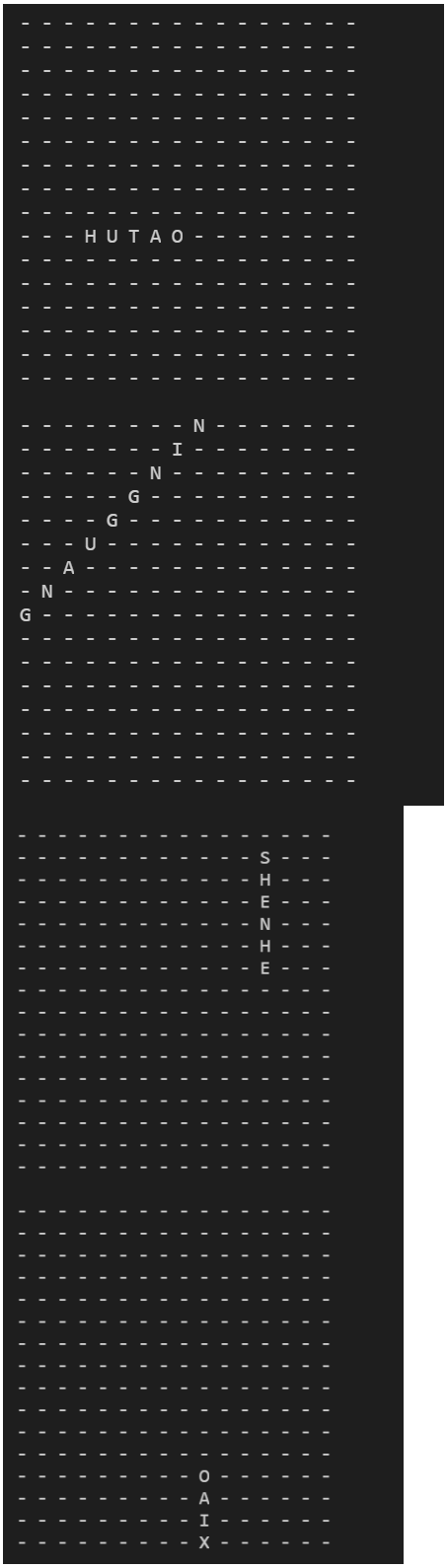




Contoh 4(small):

```
Input nama file: test4.txt

berikut adalah puzzle yang akan diselesaikan:
H X H I K U I N N X F Y C M M W
T H W I N L G I H R F W S R H L
N V R F G T N H J X D P H T W I
E U A N K G H B W I K V E W E P
Z A O S G J N B U B H H N Y Y G
K H M U V V L Y G Y T Z H C J W
Z S A B S S E H B E N S E J W Y
J N F Z E W Z G G O Z A I J J E
G U P B Z R O D P E O X G H K X
G M E H U T A O K B S G T O Z Z
T L J A R G I X L C B G K G M B
V A A D V C E G G W V I W S O O
Z N A S H A D R R O H J B P D Z
Y M S N C D V E I A O C M G E P
Z E U M J L B K O I A F A V M M
B C P W C D U W A X N H E W L B


berikut adalah kata yang akan dicari:
AMBER
GANYU
HUTAO
NINGGUANG
SHENHE
XIAO
ZHONGLI
```

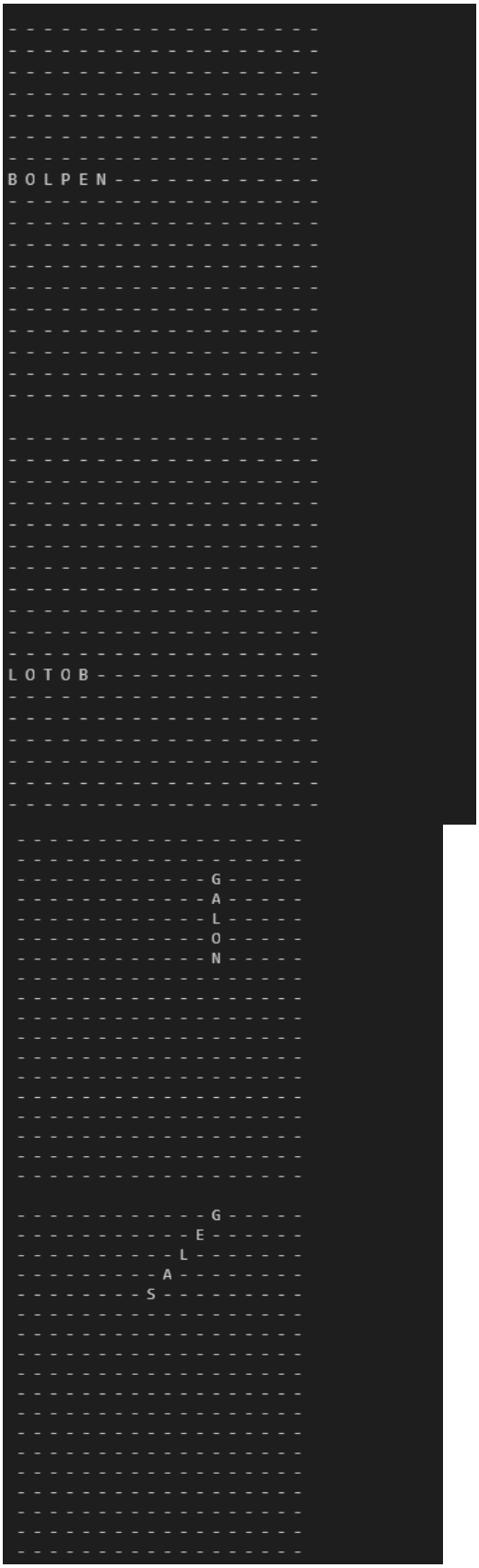
```
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- A - - - - - - - - - - - - - -
- - M - - - - - - - - - - - - -
- - - B - - - - - - - - - - - -
- - - - E - - - - - - - - - - -
- - - - - R - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -

- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - U - - - - - - -
- - - - - - - - - Y - - - - - -
- - - - - - - - - - N - - - - -
- - - - - - - - - - - A - - - -
- - - - - - - - - - - - G - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
```

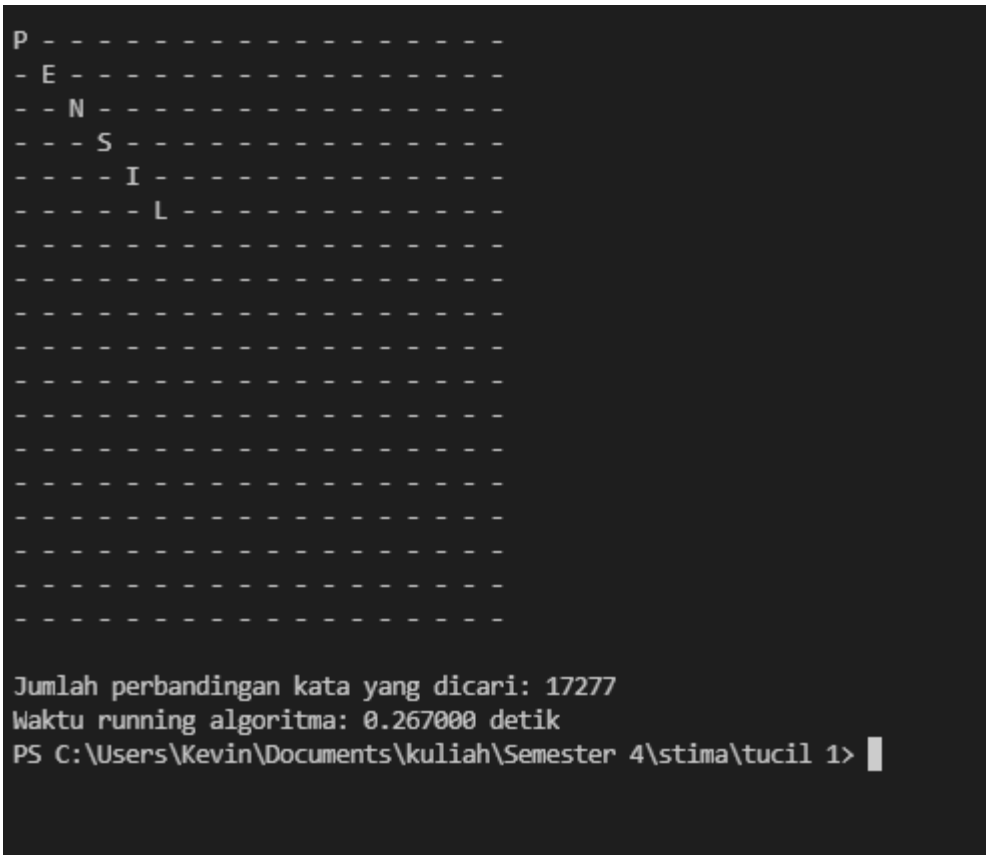

HUTAO
N
I
N
G
G
U
A
N
G
S
H
E
N
H
E
O
A
I
X

```
- - - - - - I - - - - - - - - -
- - - - - L - - - - - - - - - -
- - - - G - - - - - - - - - - -
- - - N - - - - - - - - - - - -
- - O - - - - - - - - - - - - -
- H - - - - - - - - - - - - - -
Z - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -

Jumlah perbandingan kata yang dicari: 14027
Waktu running algoritma: 0.249000 detik
PS C:\Users\Kevin\Documents\kuliah\Semester 4\stima\tucil 1>
```

Contoh 5(medium):

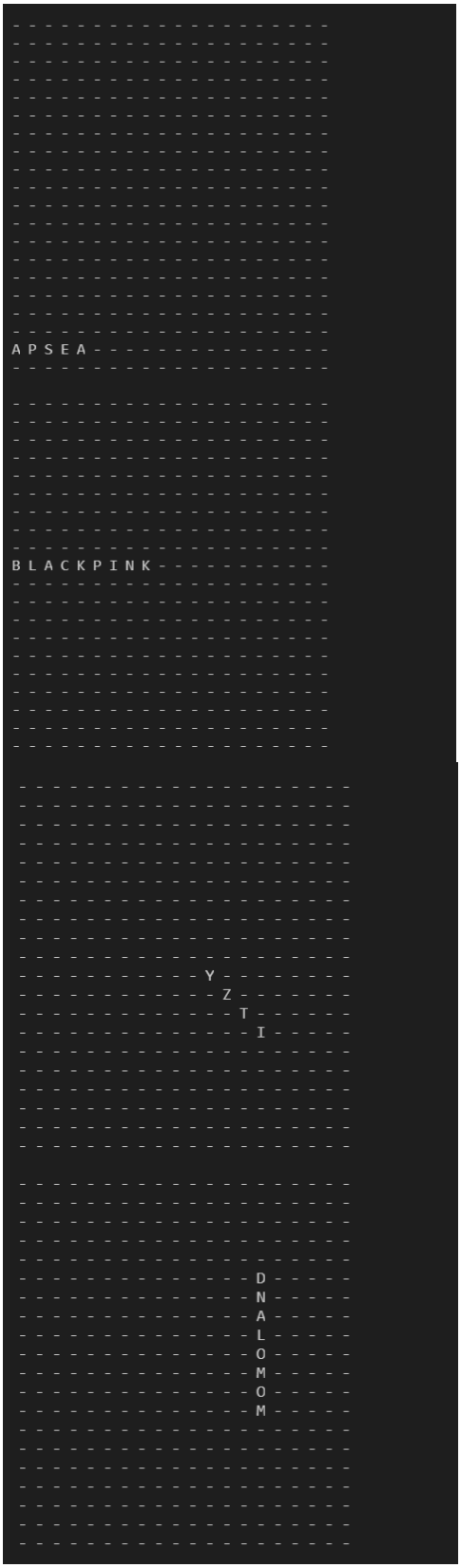
```
Input nama file: test5.txt

berikut adalah puzzle yang akan diselesaikan:
P X I G V J F D M B J O G S T T F C
W E T H C U Q S N W K E A U X E E B
C B N Q Q S P V B L L P G J L F P C
B E E S N N D T R A I T A N L I Y R
J Q R Z I F C S S K W Q L E N C F C
Q I I S S L R Q U X H J O J H R N F
E R F W L V A E O Y L S N T N B V J
B O L P E N D W S Z Y A J U I U E R
H J N Z O R V G M F G O W G I K V C
S O N A Y F S H M K B O O V V A O G
Q U C N N J J K O S F W A V Y H V G
L O T O B G D R E S E H N K V Y Z J
B L Z N Y I B W A D P B A L N L T X
E D E N K T I U Y V S V O J F E Q Q
T I I N O R I S L P F P D W N E I T
P F O H C E X G J L J R P H U F Q P
V C W X Q U J E I U Z W K L H P D Y
D C G M D R J Y A E I C S E H J C V


berikut adalah kata yang akan dicari:
BOLPEN
BOTOL
GALON
GELAS
KARPET
KIPAS
PENSIL
```


BOLPEN
LOTOB
GALON
GELASS

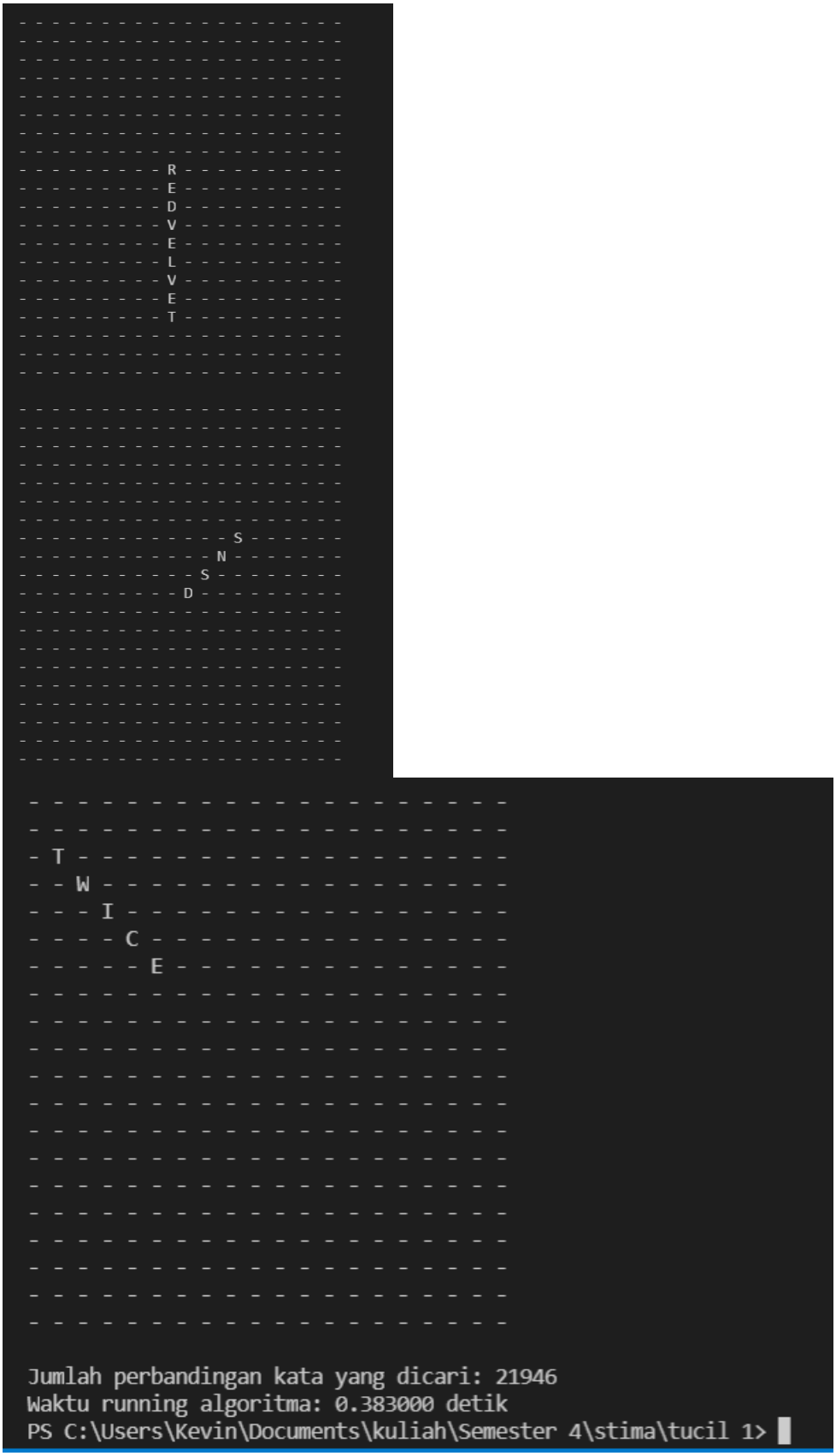


Contoh 6(medium):

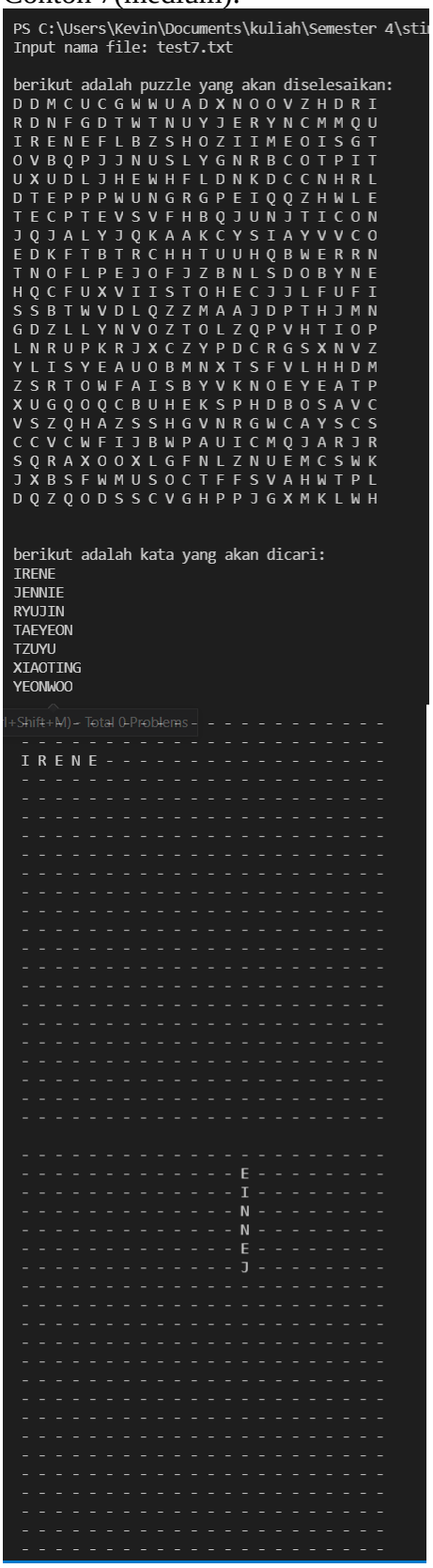
```
Input nama file: test6.txt

berikut adalah puzzle yang akan diselesaikan:
U A M R G S L R U S A J V Q E X E B D J
J L M F M Y A C K Z M E N M Q S R N L V
R T O J O G D T A H H K X I A Y X R Z Q
Q G W D O F E K H H O A T A Z A A X U C
G Q J I A E D K L I V Z G H Q C P R W Z
S C M H C S W F N M N O J L D M F U W N
M I K I F E F W R G R N T A N V I A U M
L V B V H E W O D C U C Z S A F J Z X Q
A B J E B D H W A R I R N Z L M G N Z W
B L A C K P I N K E X S Y C O S S P L H
C K A I L V R J R D D Y L M M G W P W C
G X W P N J L B B V S U Z Z O G K W E U
U S H P N T R O C E B F J T M G L X R F
X L N V W F I N J L W K N V I A Z Z P X
O H N U X C C U I V X T F Z B C C I Q Q
J M Z V J M D F L E V G O G S M C F Z O
T Y J P N Y Z C K T H X X G E D A V Z X
A I X D R J C Q B Y R Q W M P H G X J B
A P S E A C Q R R X F P N W Q S D L V U
Y K P X U D X O Q D C U Y S N R A X E Z


berikut adalah kata yang akan dicari:
AESPA
BLACKPINK
ITZY
MOMOLAND
REDVELVET
SNSD
TWICE
```

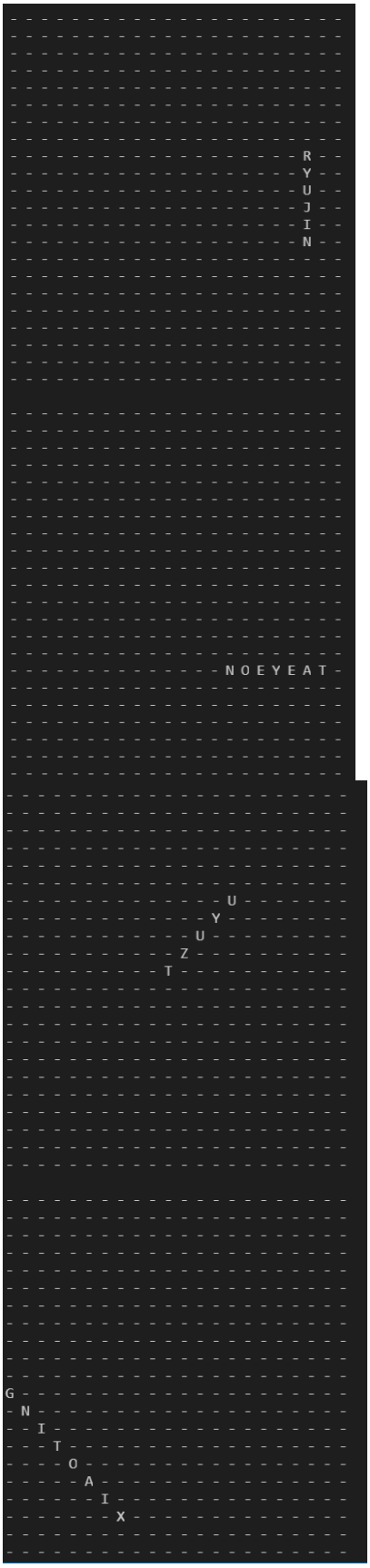

APSEA
BLACKPINK

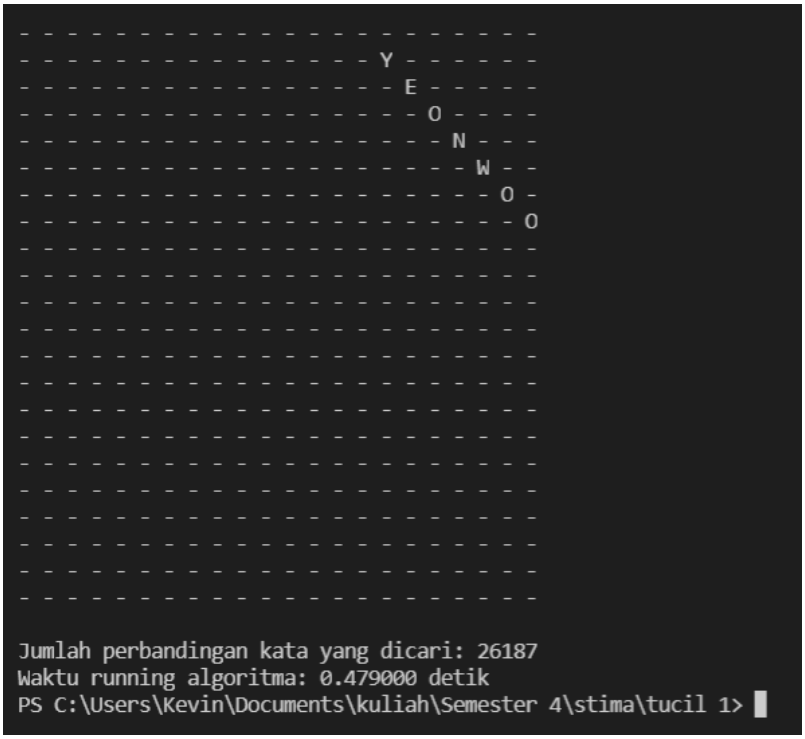
YZTI
DNALOMOM


R
E
D
V
E
L
V
E
T
S
N
S
D
T
W
I
C
E
Jumlah perbandingan kata yang dicari: 21946
Waktu running algoritma: 0.383000 detik
PS C:\Users\Kevin\Documents\kuliah\Semester 4\stima\tucil 1>

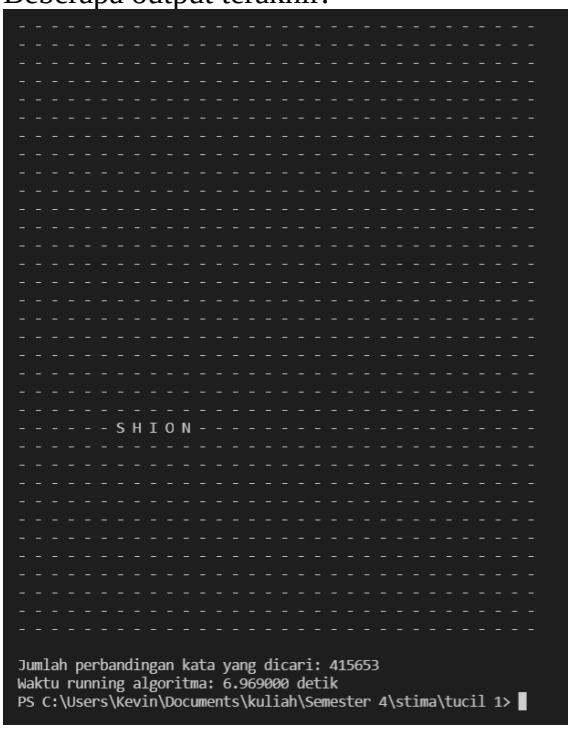
Contoh 7(medium):

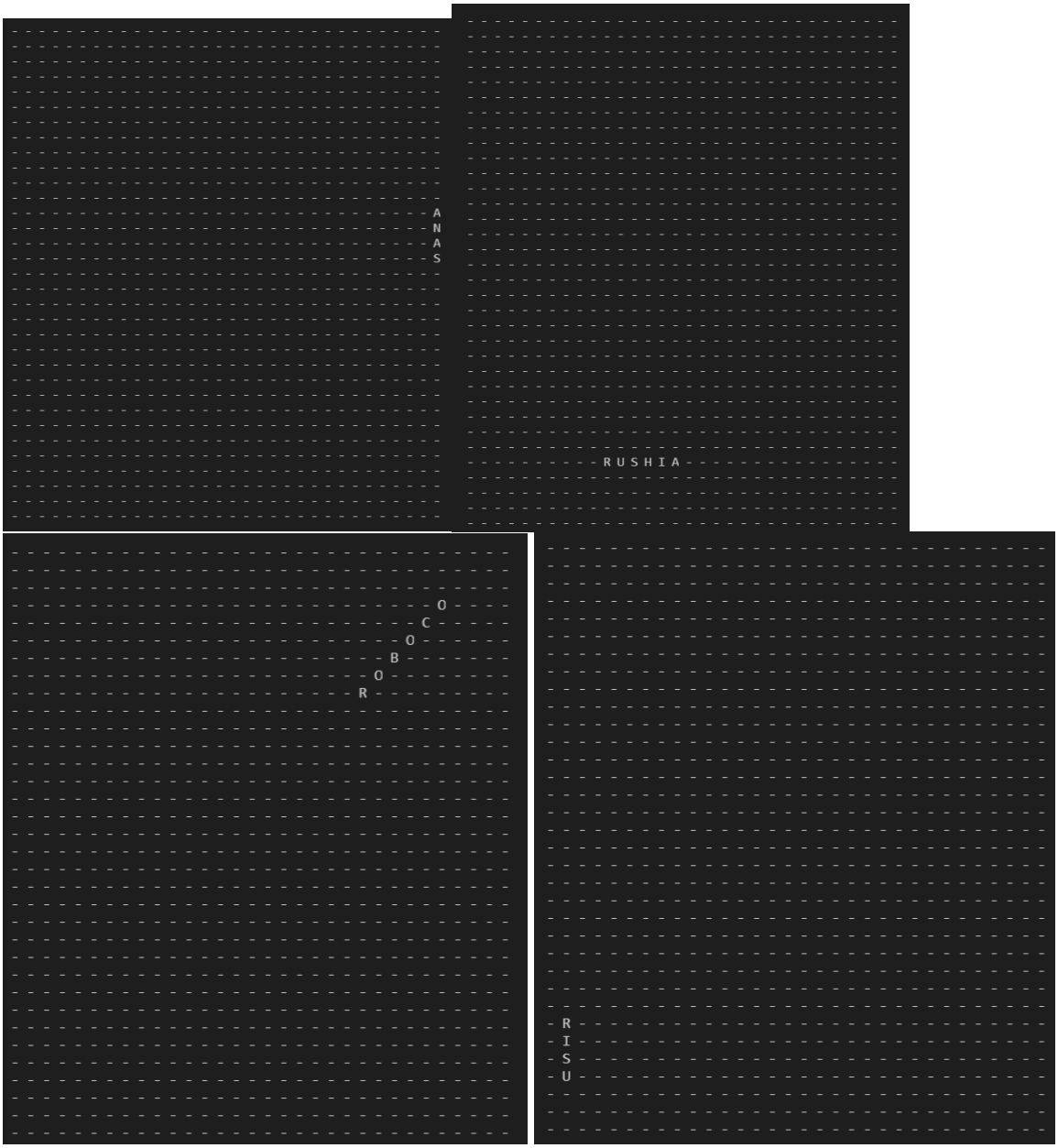


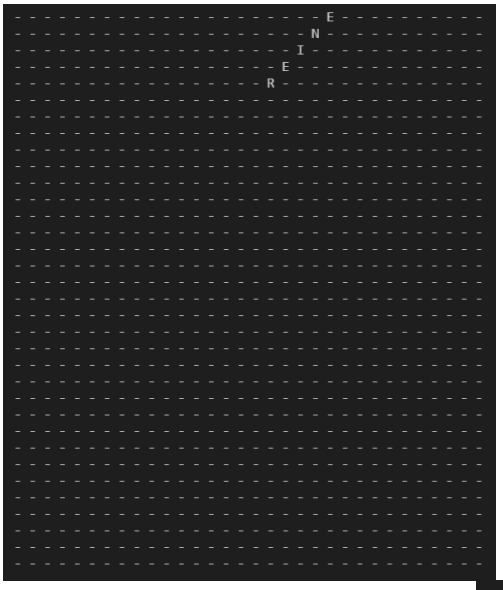

R
Y
U
J
I
N
N O E Y E A T
U
Y
U
Z
T
G
N
I
T
O
A
I
X

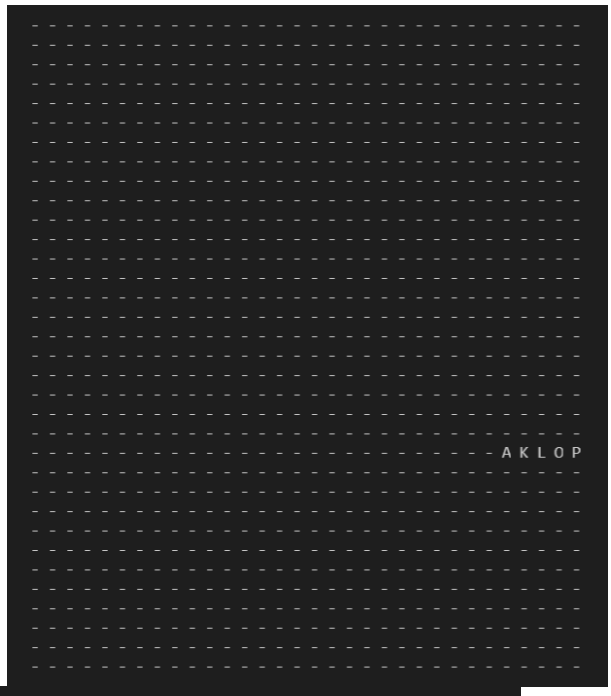


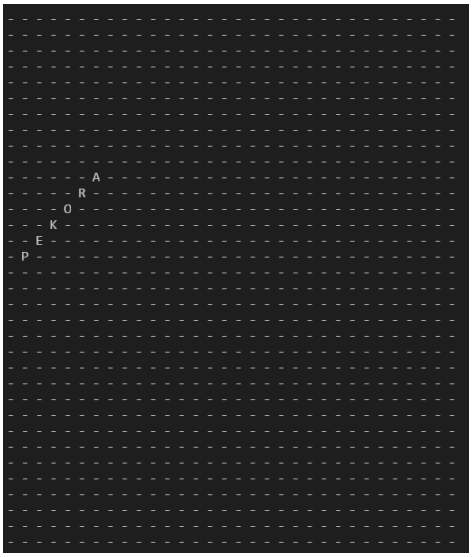
Contoh 8(large): (cetakan ke layer dibuat secara reverse(dari akhir output hingga input)
Beberapa output terakhir:

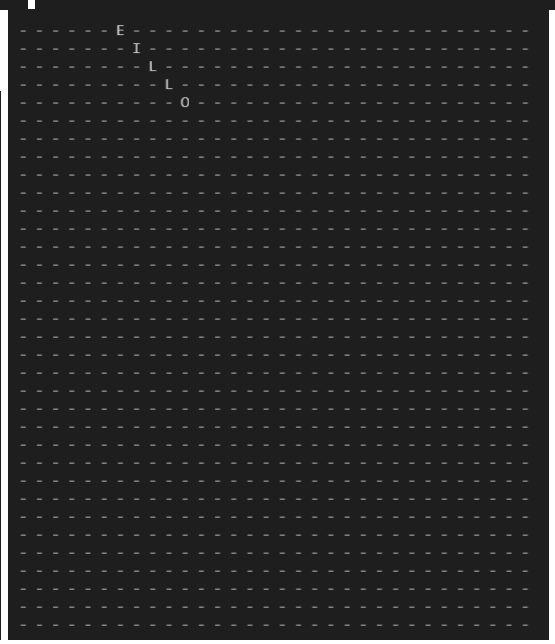



ANAS
RUSHIA
OCOBOR
RISU


E
N
I
E
R


A K L O P

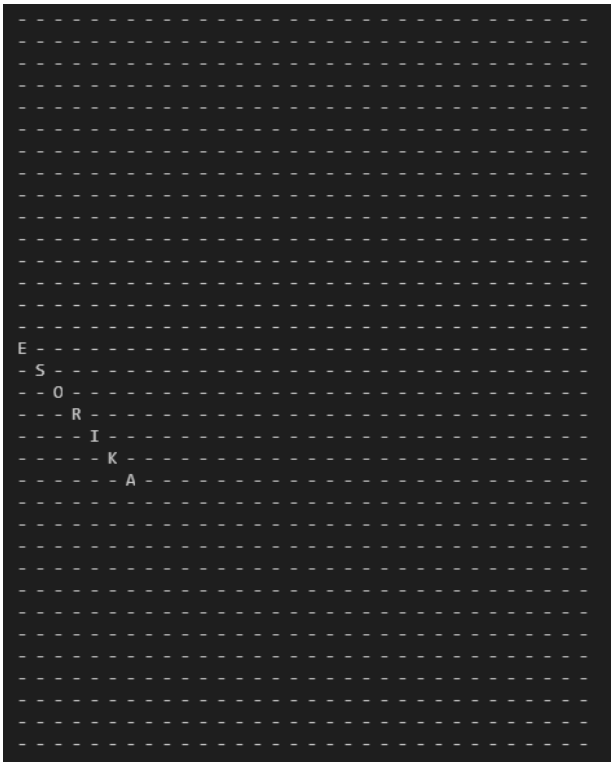

A
R
O
K
E
P

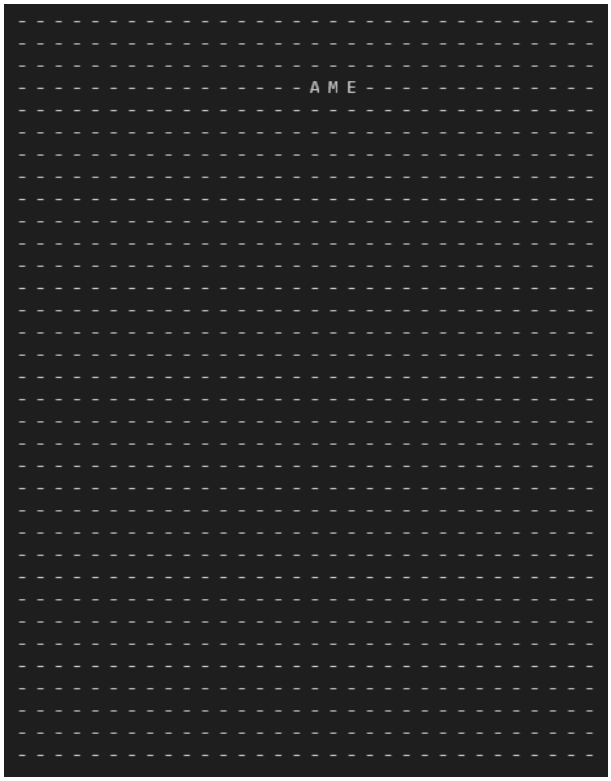

E
I
L
L
O

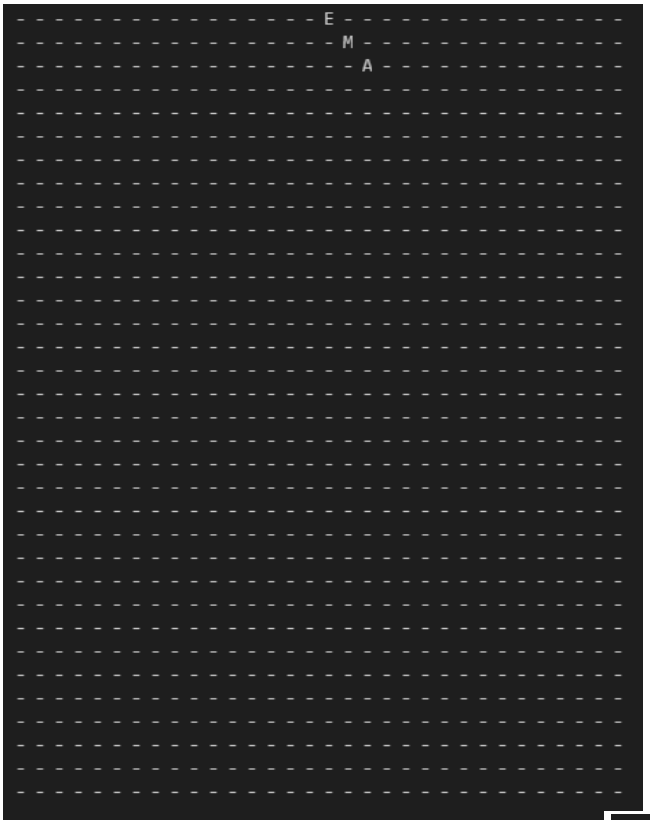
Input beserta beberapa output awal:

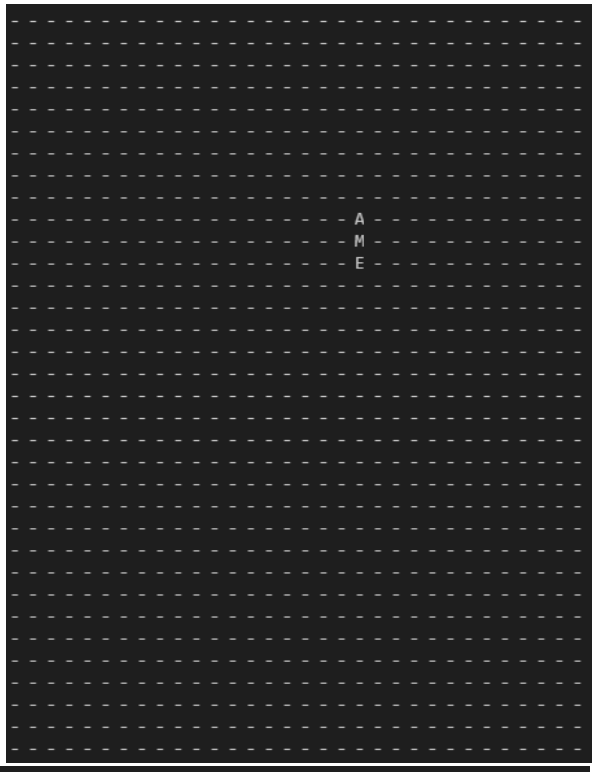
```
berikut adalah puzzle yang akan diselesaikan:
C V L U X G E L L B Z Z R D S W E N D T I E M Z G Z Q X I U Y X
P E Q G Y N L I O A Y V X K R I H M Z W N T N H V B M R A Y I E
X M I S T A C Q L C A L L I M J V Z A I Z K D U Z A O S T S D Z
C D H S T U K A G L O I R U S T A M E T S L S G N Y B O O P M E
S E J Y F I A O K C O A L P Y U L R D I A H K U O H C Y C S Y N
U B G I F I K Z M I Y U L K B F L S Y G D W L K L O R W N C Z Q
B K T U H X D U K K E X B F E I A P Y N S P W O B R Z E E S Y G
J Y G O E K A S T I F C H M M V U F A R I Q X O N H B S E D T F
W C F E B B E X L O D K F F S M U E A L I B R O I C W W S Y X P
W V K I X U F J J Q P A A E W B J N U A A X E A L N J M Y Y S G
D N S E M Y A K Y B D E C M U T U O O M M L N X F B H A Q M T U
L O Y M F R V Y Z H E A T K A N X R R E A M K J H N M Z I R S X
A X K U O C K V A I M H I Z R R Q O F P H K N O A M Y V P E Y A
G S Q K R W R L S M D O C P O S I K H Z C M C K N G U H U F X N
T Z E A R I X E N D E R J C J B K N B X A G H T E Z D M L N X A
E P I W Z S R O A Z R I P N X N N B E H A V N I N A S X E Z E S
Y S Q O W E T Z K I L U A F X Y M O G U H Q F C E N I D V I Q H
O S O T C K R O N I I L A R J A G J Y X L F P W Z P V E W R Y F
Y R C R O J M Z M S G J G J O N O Z K O S Q O G F U K I V P S L
Y D A H I M E A R A I K L I W S K W R Z A H K M S I S H K U P A
R N K Y V K L J X C P K C T H M E V Z H A U N E F P A J B A Z R
A T A N A K A L U I O X V Y W Q V A H A R O I Y L U B A W G U E
L L B E B Y S H I O N S D C Z E D O J R J L N Q C A R A K L O P
M Q Z D W L A V A Q Z F P Q I O S U I S E I U J H U M M T M L O
G T L V M U P B G A K Z Z X X F F R T U Y M Z S O W Y Y O R P T
Q A T I Q E U R B K J N L E G S I H I O T V G X C U R O D D A G
S Z P A Q I U C X H F F C O C O C N G U L Y I U O K N B V Q M S
M R N C L G X X F B O A M U R N B H K T G N G U R A C O N M D H
B I E X Q I Z Z P G W R J F R I I X L P A G O N W A K Y U K V A
B S A Z J D A V V O R U S H I A B R Q O B I L H C B O A D Y C Y
A U L M B K E A K K H J X H O O K S R G E H P O O F M I E F D N
E W B K E N X C C I T K U P T N O I M P D V X M F D U E A F I A
L L D J C X C K Y M E V N A G J U B G N X V L H R Z I F O I D T
Z I M A Z J E T J Q C I N S H I A K A P C I F O D K I C S T M Z
```

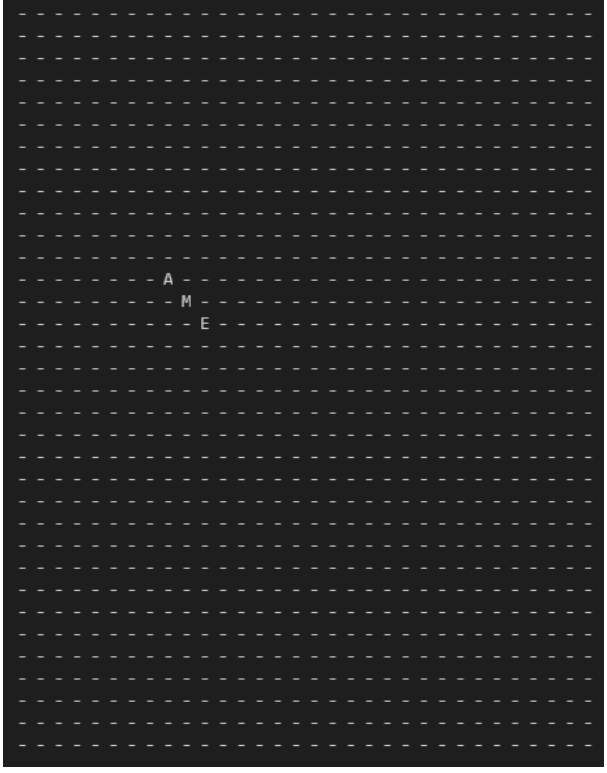
```
berikut adalah kata yang akan dicari:
AKIROSE
AME
ANYA
AQUA
AYAME
AZKI
BAELS
BOTAN
CALLI
CERES
CHLOE
CHOCO
COCO
FLARE
FUBUKI
GURA
HAACHAMA
INA
IOFI
IROHA
IRYS
KANATA
KIARA
KORONE
KOYORI
KRONII
LAMY
LAPLUS
LUI
LUNA
MARINE
MATSURI
MEL
MIKO
MIO
MOONA
MUMEI
NENE
NOEL
OKAYU
OLLIE
PEKORA
POLKA
REINE
RISU
ROBOCO
RUSHIA
SANA
SHION
```

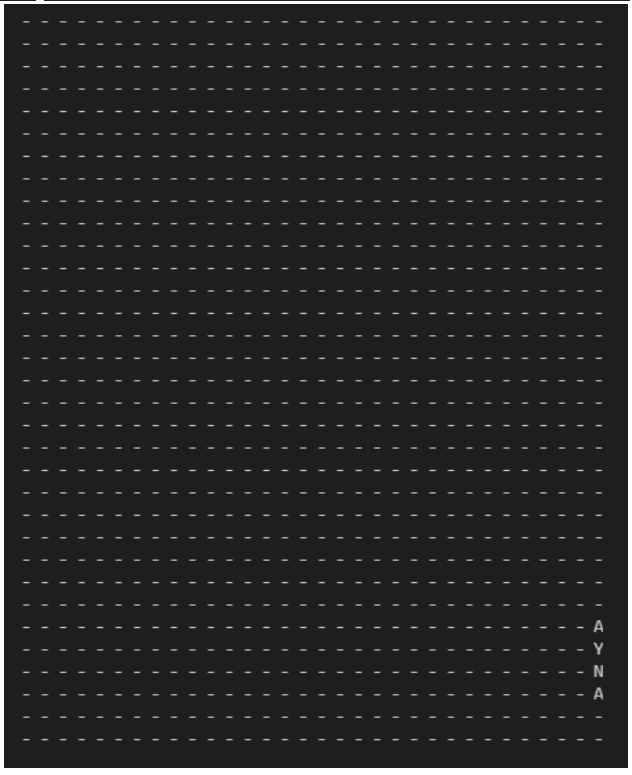

E
S
O
R
I
K
A

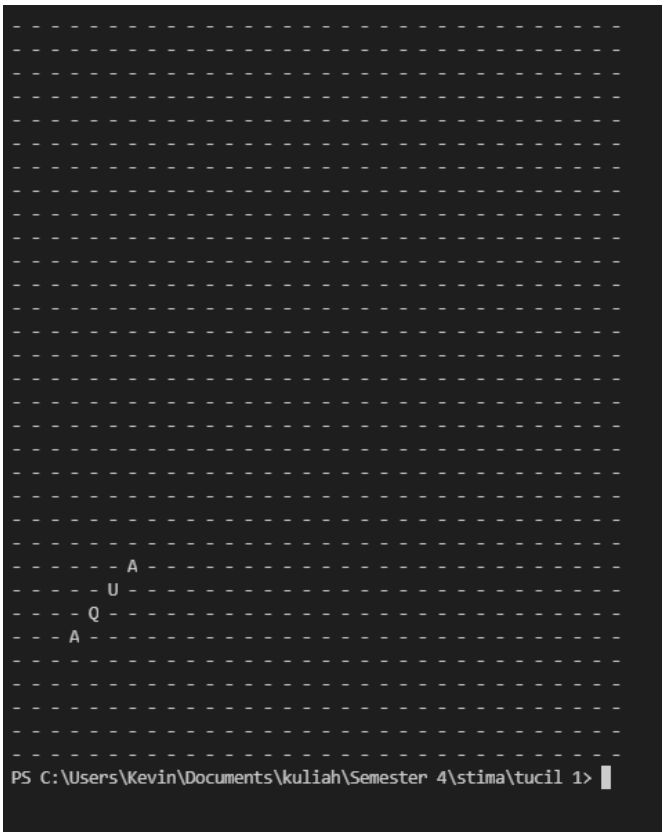

AME


E
M
A


A
M
E


A
M
E


A
Y
N
A



(program sengaja dihentikan di tengah jalan agar dapat melihat interaksi input dan beberapa output awal)

BAB IV
LINK TERKAIT

SOURCE CODE & TEST CASE

https://github.com/jakartasipirok/Word-Search-Solver.git

REFERENSI

https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Tugas-Kecil-1-(2022).pdf
https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2016-2017/Makalah2017/Makalah-IF2211-2017-077.pdf

BAB V
LAMPIRAN

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	✓	