LAPORAN TUGAS KECIL 1
IF2211 STRATEGI ALGORITMA SEMESTER II TAHUN 2021/2022
PENYELESAIAN WORD SEARCH PUZZLE DENGAN ALGORITMA BRUTE FORCE

DANIEL SALIM
13520008
K02

BAB I
ALGORITMA BRUTE FORCE

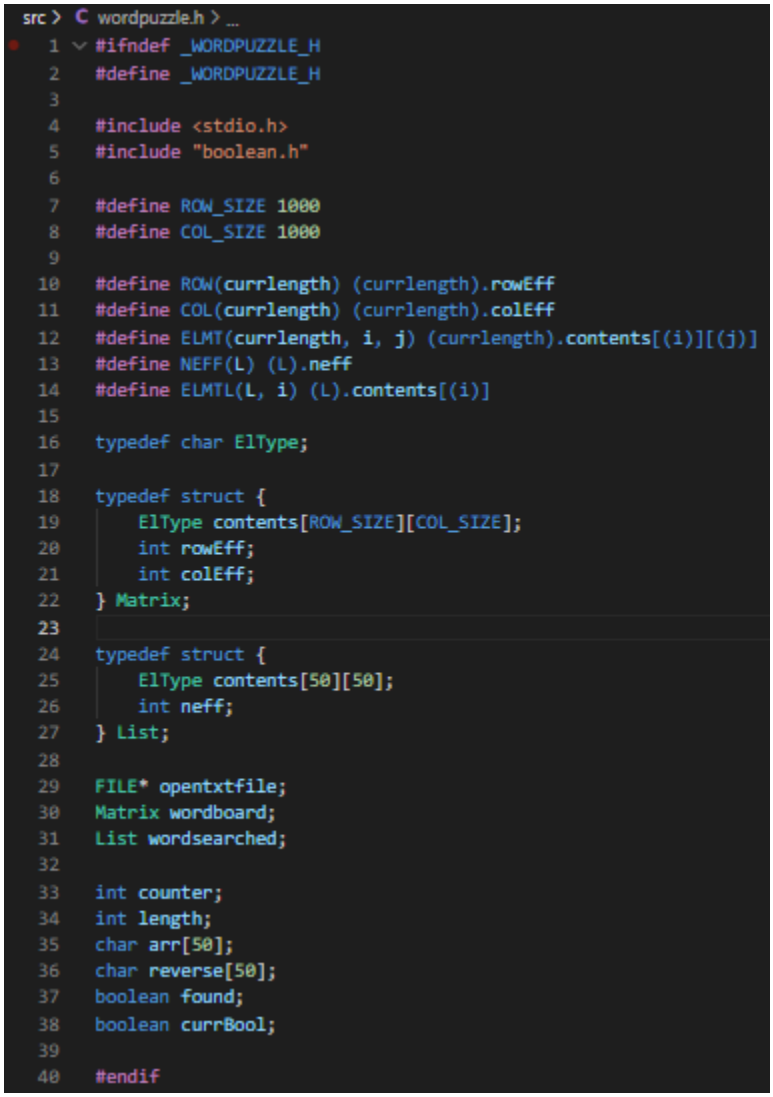
Dalam pengerjaan Tugas Kecil kali ini didasari oleh konsep algoritma brute force, yaitu pada bagian mencari persamaan string atau pattern yang diinginkan. Langkah pertama, word puzzle yang hendak diolah dibuat dalam bentuk matriks untuk mempermudah pengaksesan dan kata-kata yang ingin dicari dalam bentuk list. Selanjutnya, program akan memanggil 8 fungsi boolean yang masing-masing memiliki fungsi untuk mencari kata dalam arah yang bersangkutan (east, southeast, south, dst). Di dalam fungsi ini terdapat while loop yang mengecek setiap kemungkinan adanya kata yang ingin dicari dalam setiap arah di dalam word puzzle tersebut dengan menggunakan patokan yaitu panjang kata yang ingin dicari. Jika loop tersebut telah mengulang sebanyak panjang kata yang ingin dicari, maka kata tersebut ada di dalam word puzzle dan program akan menampilkan kata tersebut di dalam word puzzle dengan elemen-elemennya yang lain bernilai ".", bagian ini menggunakan for loop dengan cara memeriksa apakah indeks yang sedang diduduki sekarang merupakan indeks salah satu kata yang dicari yang terdapat di dalam word puzzle.

Selain proses pencarian, dalam setiap mencari kemungkinan kata yang terdapat di dalam word puzzle, program akan menghitung tiap pergerakannya dalam variabel counter yang nantinya akan di totalkan untuk menyatakan jumlah perbandingan huruf yang dilakukan untuk menemukan kata di dalam puzzle.

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.	✓	

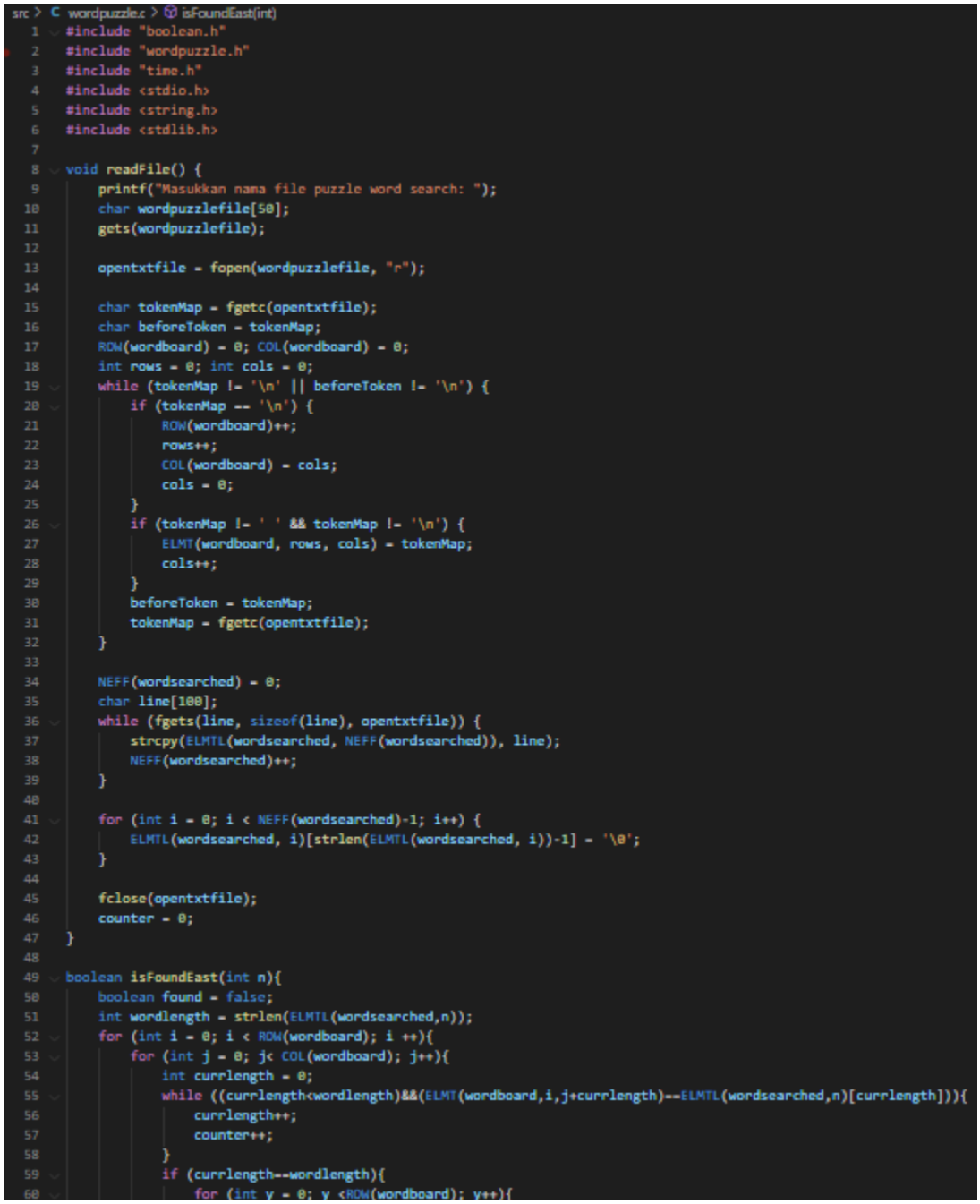
BAB II
SOURCE CODE PROGRAM

Screenshot wordpuzzle.h



```
src > C wordpuzzle.h > ...
1  #ifndef _WORDPUZZLE_H
2  #define _WORDPUZZLE_H
3
4  #include <stdio.h>
5  #include "boolean.h"
6
7  #define ROW_SIZE 1000
8  #define COL_SIZE 1000
9
10 #define ROW(currlength) (currlength).rowEff
11 #define COL(currlength) (currlength).colEff
12 #define ELMT(currlength, i, j) (currlength).contents[(i)][(j)]
13 #define NEFF(L) (L).neff
14 #define ELMTL(L, i) (L).contents[(i)]
15
16 typedef char ElType;
17
18 typedef struct {
19     ElType contents[ROW_SIZE][COL_SIZE];
20     int rowEff;
21     int colEff;
22 } Matrix;
23
24 typedef struct {
25     ElType contents[50][50];
26     int neff;
27 } List;
28
29 FILE* opentxtfile;
30 Matrix wordboard;
31 List wordsearched;
32
33 int counter;
34 int length;
35 char arr[50];
36 char reverse[50];
37 boolean found;
38 boolean currBool;
39
40 #endif
```

Screenshot wordpuzzle.c



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

void readFile() {
    printf("Masukkan nama file puzzle word search: ");
    char wordpuzzlefile[50];
    gets(wordpuzzlefile);

    opentxtfile = fopen(wordpuzzlefile, "r");

    char tokenMap = fgetc(opentxtfile);
    char beforeToken = tokenMap;
    ROW(wordboard) = 0; COL(wordboard) = 0;
    int rows = 0; int cols = 0;
    while (tokenMap != '\n' || beforeToken != '\n') {
        if (tokenMap == '\n') {
            ROW(wordboard)++;
            rows++;
            COL(wordboard) = cols;
            cols = 0;
        }
        if (tokenMap != ' ' && tokenMap != '\n') {
            ELMT(wordboard, rows, cols) = tokenMap;
            cols++;
        }
        beforeToken = tokenMap;
        tokenMap = fgetc(opentxtfile);
    }

    NEFF(wordsearched) = 0;
    char line[100];
    while (fgets(line, sizeof(line), opentxtfile)) {
        strcpy(ELMTL(wordsearched, NEFF(wordsearched)), line);
        NEFF(wordsearched)++;
    }

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

    fclose(opentxtfile);
    counter = 0;
}

boolean isFoundEast(int n){
    boolean found = false;
    int wordlength = strlen(ELMTL(wordsearched,n));
    for (int i = 0; i < ROW(wordboard); i ++){
        for (int j = 0; j< COL(wordboard); j++){
            int currlength = 0;
            while ((currlength<wordlength)&&(ELMT(wordboard,i,j+currlength)==ELMTL(wordsearched,n)[currlength])){
                currlength++;
                counter++;
            }
            if (currlength==wordlength){
                for (int y = 0; y <ROW(wordboard); y++){
```

```
                    for (int z = 0; z<COL(wordboard); z++){
                        if (y == i && z >= j && z <= j + wordlength - 1){
                            printf("%c ",(ELMT(wordboard, y, z)));
                        }
                        else {
                            printf(". ",(ELMT(wordboard, y, z)));
                        }
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
        }
    }
    return false;
}

boolean isFoundSoutheast(int n){
    boolean found = false;
    int wordlength = strlen(ELMTL(wordsearched,n));
    int counterwordlength = wordlength;
    for (int i = 0; i < ROW(wordboard); i ++){
        for (int j = 0; j< COL(wordboard); j++){
            int currlength = 0;
            while ((currlength<wordlength)&&(ELMT(wordboard,i+currlength,j+currlength)==ELMTL(wordsearched,n)[currlength])){
                currlength++;
                counter++;
            }
            if (currlength==wordlength){
                for (int y = 0; y <ROW(wordboard); y++){
                    for (int z = 0; z<COL(wordboard); z++){
                        if (y >= i && y == i + wordlength - counterwordlength && z >= j && z == j + wordlength - counterwordlength && counterwordlength > 0){
                            printf("%c ",(ELMT(wordboard, y, z)));
                            counterwordlength--;
                        }
                        else {
                            printf(". ",(ELMT(wordboard, y, z)));
                        }
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
        }
    }
    return false;
}

boolean isFoundSouth(int n){
    boolean found = false;
    int wordlength = strlen(ELMTL(wordsearched,n));
    for (int i = 0; i < ROW(wordboard); i ++){
        for (int j = 0; j< COL(wordboard); j++){
            int currlength = 0;
            while ((currlength<wordlength)&&(ELMT(wordboard,i+currlength,j)==ELMTL(wordsearched,n)[currlength])){
                currlength++;
                counter++;
            }
```

```
            if (currlength==wordlength){
                for (int y = 0; y <ROW(wordboard); y++){
                    for (int z = 0; z<COL(wordboard); z++){
                        if (y >= i && y <= i + wordlength - 1 && z == j){
                            printf("%c ",(ELMT(wordboard, y, z)));
                        }
                        else {
                            printf(". ",(ELMT(wordboard, y, z)));
                        }
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
        }
    }
    return false;
}

boolean isFoundSouthwest(int n){
    boolean found = false;
    int wordlength = strlen(ELMTL(wordsearched,n));
    int counterwordlength = wordlength;
    for (int i = 0; i < ROW(wordboard); i ++){
        for (int j = 0; j< COL(wordboard); j++){
            int currlength = 0;
            while ((currlength<wordlength)&&(ELMT(wordboard,i+currlength,j-currlength)==ELMTL(wordsearched,n)[currlength])){
                counter++;
                currlength++;
            }
            if (currlength==wordlength){
                for (int y = 0; y <ROW(wordboard); y++){
                    for (int z = 0; z<COL(wordboard); z++){
                        if (y >= i && y == i + wordlength - counterwordlength && z <= j && z == j - wordlength + counterwordlength && counterwordlength > 0){
                            printf("%c ",(ELMT(wordboard, y, z)));
                            counterwordlength--;
                        }
                        else {
                            printf(". ",(ELMT(wordboard, y, z)));
                        }
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
        }
    }
    return false;
}

boolean isFoundWest(int n){
    /* n adalah urutan kata ke berapa yang ingin di cari */
    boolean found = false;
    int wordlength = strlen(ELMTL(wordsearched,n));
    for (int i = 0; i < ROW(wordboard); i ++){
        for (int j = 0; j< COL(wordboard); j++){
            int currlength = 0;
            while ((currlength<wordlength)&&(ELMT(wordboard,i,j-currlength)==ELMTL(wordsearched,n)[currlength])){
```

```
                counter++;
                currlength++;
            }
            if (currlength==wordlength){
                for (int y = 0; y <ROW(wordboard); y++){
                    for (int z = 0; z<COL(wordboard); z++){
                        if (y == i && j - wordlength + 1 <= z && z <= j){
                            printf("%c ",(ELMT(wordboard, y, z)));
                        }
                        else {
                            printf(". ",(ELMT(wordboard, y, z)));
                        }
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
        }
    }
    return false;
}

boolean isFoundNorthwest(int n){
    boolean found = false;
    int wordlength = strlen(ELMTL(wordsearched,n));
    int counterwordlength = wordlength;
    for (int i = 0; i < ROW(wordboard); i ++){
        for (int j = 0; j< COL(wordboard); j++){
            int currlength = 0;
            while ((currlength<wordlength)&&(ELMT(wordboard,i-currlength,j-currlength)==ELMTL(wordsearched,n)[currlength])){
                counter++;
                currlength++;
            }
            if (currlength==wordlength){
                for (int y = 0; y <ROW(wordboard); y++){
                    for (int z = 0; z<COL(wordboard); z++){
                        if (y >= i - wordlength + 1 && y == i + 1 - counterwordlength && z >= j - wordlength + 1 && z == j - counterwordlength + 1 && counterwordlength > 0){
                            printf("%c ",(ELMT(wordboard, y, z)));
                            counterwordlength--;
                        }
                        else {
                            printf(". ",(ELMT(wordboard, y, z)));
                        }
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
        }
    }
    return false;
}

boolean isFoundNorth(int n){
    boolean found = false;
    int wordlength = strlen(ELMTL(wordsearched,n));
    for (int i = 0; i < ROW(wordboard); i ++){
        for (int j = 0; j< COL(wordboard); j++){
```

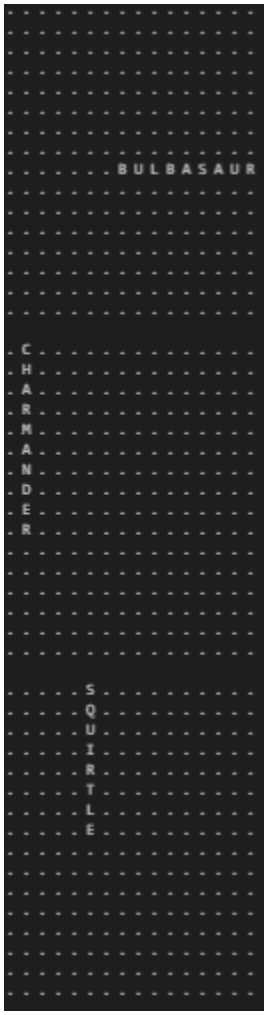
```
            int currlength = 0;
            while ((currlength<wordlength)&&(ELMT(wordboard,i-currlength,j)==ELMTL(wordsearched,n)[currlength])){
                counter++;
                currlength++;
            }
            if (currlength==wordlength){
                for (int y = 0; y <ROW(wordboard); y++){
                    for (int z = 0; z<COL(wordboard); z++){
                        if (i - wordlength + 1 <= y && y <= i && z == j){
                            printf("%c ",(ELMT(wordboard, y, z)));
                        }
                        else {
                            printf(". ",(ELMT(wordboard, y, z)));
                        }
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
        }
    }
    return false;
}

boolean isFoundNortheast(int n){
    boolean found = false;
    int wordlength = strlen(ELMTL(wordsearched,n));
    int counterwordlength = wordlength;
    for (int i = 0; i < ROW(wordboard); i ++){
        for (int j = 0; j< COL(wordboard); j++){
            int currlength = 0;
            while ((currlength<wordlength)&&(ELMT(wordboard,i-currlength,j+currlength)==ELMTL(wordsearched,n)[currlength])){
                counter++; currlength++;
            }
            if (currlength==wordlength){
                for (int y = 0; y <ROW(wordboard); y++){
                    for (int z = 0; z<COL(wordboard); z++){
                        if (y >= i - wordlength + 1 && y == i + 1 - counterwordlength && z <= j + wordlength - 1 && z == j - 1 + counterwordlength && counterwordlength > 0){
                            printf("%c ",(ELMT(wordboard, y, z)));
                            counterwordlength--;
                        }
                        else {
                            printf(". ",(ELMT(wordboard, y, z)));
                        }
                    }
                    printf("\n");
                }
                printf("\n");
                return true;
            }
        }
    }
    return false;
}

int main(){
    readFile();
    printf("\n");
    printf("----- W O R D  P U Z Z L E ----- \n");
```

```c
int main(){
    readFile();
    printf("\n");
    printf("----- W O R D  P U Z Z L E ----- \n");
    for (int i = 0; i <ROW(wordboard); i++){
        for (int j = 0; j<COL(wordboard); j++){
            printf("%c ",(ELMT(wordboard, i, j)));
        }
        printf("\n");
    }
    printf("\n------ K E Y  W O R D -----\n");
    for (int i = 0; i <= NEFF(wordsearched); i++){
        printf("%s\n", ELMTL(wordsearched, i));
    }
    clock_t initiatepuzzle = clock();
    for (int n = 0; n < NEFF(wordsearched); n++){
        isFoundEast(n);
        isFoundSoutheast(n);
        isFoundSouth(n);
        isFoundSouthwest(n);
        isFoundWest(n);
        isFoundNorthwest(n);
        isFoundNorth(n);
        isFoundNortheast(n);
    }
    clock_t terminatepuzzle = clock();
    double duration = ((double) (terminatepuzzle - initiatepuzzle))/(1000);

    printf("Durasi eksekusi program adalah sebesar : %.3f sekon\n", (float)duration);
    printf("Jumlah perbandingan huruf adalah sebanyak : %d kali", counter);
}
```

BAB III
INPUT DAN OUTPUT PROGRAM

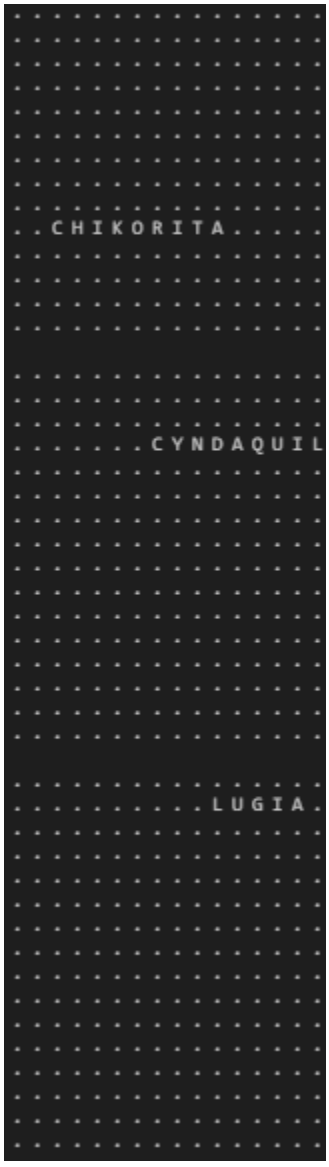
Small (16x16) :

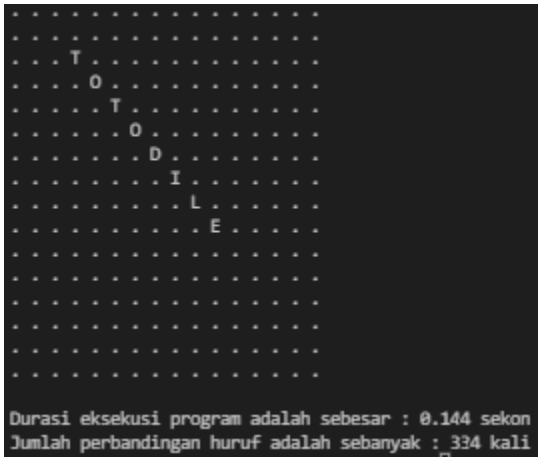
1.



2.

```
----- W O R D  P U Z Z L E -----
U F H D R P V F J N B G H H H R
K Y W O N I W M X C L U G I A K
O R Y T E S S W O B A H V O U D
Q B V L O Y N C Y N D A Q U I L
E Y H C D T W C R G J D V P V O
R Q Z W U I O S G N G E T A F T
V S P J X B M D F W Q G A L N X
B N O X D R R A I Z S C K X S P
O H L M W K P X C L E R A D F Q
W H K H H T P H U A E C L E T D
H P I M G W H M B K M H C T D F
Z K C H I K O R I T A H J N V C
A H A P Q Q Z I S H B S J C Q I
B M Y E J R X G G B T Q Z J Y M
H A N X Z U K U J V J K U I K M
A H A P Q Q Z I S H B S J C Q I

------ K E Y  W O R D -----
CHIKORITA
CYNDAQUIL
LUGIA
TOTODILE
```





3.

```
----- W O R D   P U Z Z L E -----
N W M W D Y Z F Z S H V Y N R B
L G N U W O H A R S Q K B L N T
F S I Q D G L O P L Q P H T L O
W Z Z P W K U X C B U N P Z D R
O M T M C W I F J I O I V Z K C
G V F S W E T P F Y J B Y K G H
T R E C K K O W Y N B T Z P X I
C Q L M V Q K R A Y Q U A Z A C
A T Z H A T E Q Q I L O F K R O
Y V D E R W N N R D N N A J D B
X S P M N I F F N Y F S R Q T X
Y F G F X L D K F J N L X X Y C
A H A P Q Q Z I S H B S J C Q I
B M Y E J R X G G B T Q Z J Y M
H A N X Z U K U J V J K U I K M
A H A P Q Q Z I S H B S J C Q I

------ K E Y   W O R D -----
MUDKIP
RAYQUAZA
TORCHIC
TRECKKO
```

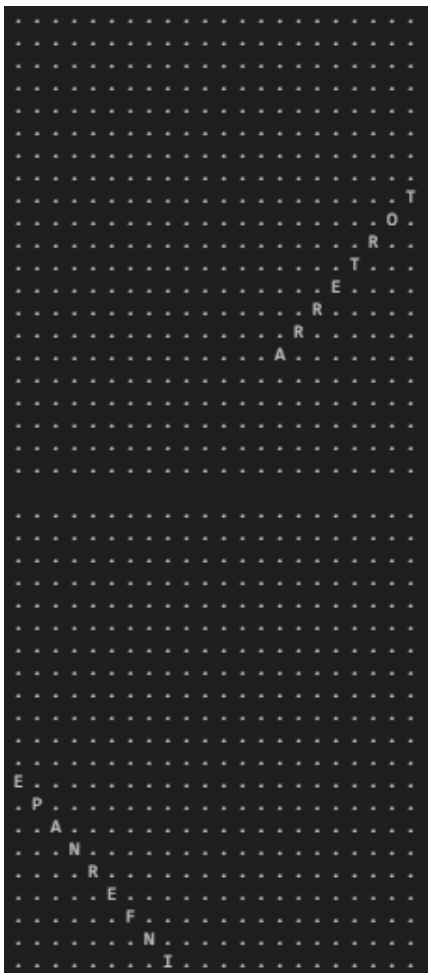
```
. . M . . . . . . . . . . . . .
. . . U . . . . . . . . . . . .
. . . . D . . . . . . . . . . .
. . . . . K . . . . . . . . . .
. . . . . . I . . . . . . . . .
. . . . . . . P . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . R A Y Q U A Z A .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . T
. . . . . . . . . . . . . . . O
. . . . . . . . . . . . . . . R
. . . . . . . . . . . . . . . C
. . . . . . . . . . . . . . . H
. . . . . . . . . . . . . . . I
. . . . . . . . . . . . . . . C
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
```

```
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
T R E C K K O . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .

Durasi eksekusi program adalah sebesar : 0.143 sekon
Jumlah perbandingan huruf adalah sebanyak : 308 kali
```

Medium (21x22) :

1.

```
----- W O R D  P U Z Z L E -----
J W I Z C U W I G V B P A K W W N B M O C F
G F A C C G I W T R U T I O B N H Z W V Q F
A X O G I X S Q I J P D U P G A Q D E W Z V
E E R T N R I E X W V X E W L K A V G C J G
B W E G I T B C W U K U H V Y U D N V E H G
P P I E B B O E X E C L R X L D P A J P B A
V A P T Y G I R A T I N A Y D A L V X I X N
Z G U B S T M Z G R H X V Y K Z X D Q Y L V
G M Y R E A M I C F S Y D W T N Y T W R Z T
Q W O Z Q M I V T W F B J B U G A E A D O D
G G S F X M P U L E S B C M R C P L T R L L
I F Y W C H I O C H A R P X Q X Q R T Z E U
E S R T E M L W L X M V B D X N R E Q S W P
G P K L Q D C B I E N F T Q R A R D Y L Y M
I N A R O C O L W B O Y Q I U R W X R D S I
R D F N O X M N K H C N B Q A I H H J D M W
J W I Z R U W I G V B P A K W W N B M O C F
G F A C C E I W T R U T I O B N H Z W V Q F
B W E G I T F C W U K U H V Y U D N V E H G
G M Y R A M I N C F S Y D W T N Y T W R Z Y
W S R T E M L W I X M V B D X N S B Q S W P

------ K E Y  W O R D -----
TORTERRA
INFERNAPE
EMPOLEON
GIRATINA
```



2.

```
----- W O R D  P U Z Z L E -----
M Q C F E Y V I L U K J C A Y V I N S W Z I
H L A P B K R D Z K F S X E T X N M P J E G
J P F I W D Y M W H V V S J E X R B O I H X
B M Z P X P A Q N W W H C W P K D L Y R J E
T C C J R Q O J R E A N J J I J K W N H V V
T A W X W P H A S O I V E B G F I Y P A B A
O X Q N C C A U C R H H L C E N A T U R N P
W T X B S P M J O E A F R L Q D G F P R M O
A S T B J C G D X N J T I I V S N D Y T E V
H B X R N T O X X H M J T A Q J X L I G W M
S B X E B N J A F A Q M Q S Y N F N G E V J
O U J W N J V R X M S U T Z B Y K Y Z G K X
K I K C O L Z Z G L P E C Q F Q V Z U O X O
E M V E V T V Q X A S Z K M D L G H F I E D
L L Z R Y R S F Z D O T U W F U F Z Z L J Y
Y Q S O Q M P P B A I D J R S H S U T U X X
J W I Z C U W I G V B P A K W W N B M O C F
G F A C C G I W T R U T I O B N H Z W V Q F
B W E G I T B C W U K U H V Y U D N V E H G
G M Y R A M I C F S Y D W T N Y T W R Z Y L
W S R T E M L W R X M V B D X N S B Q S W P

------ K E Y  W O R D -----
SNIVY
TEPIG
OSHAWOTT
KYUREM
```

```
. . . . . Y V I N S . . .
```

```
. . . . . T . . . . . .
. . . . . E . . . . . .
. . . . . P . . . . . .
. . . . . I . . . . . .
. . . . . G . . . . . .
```

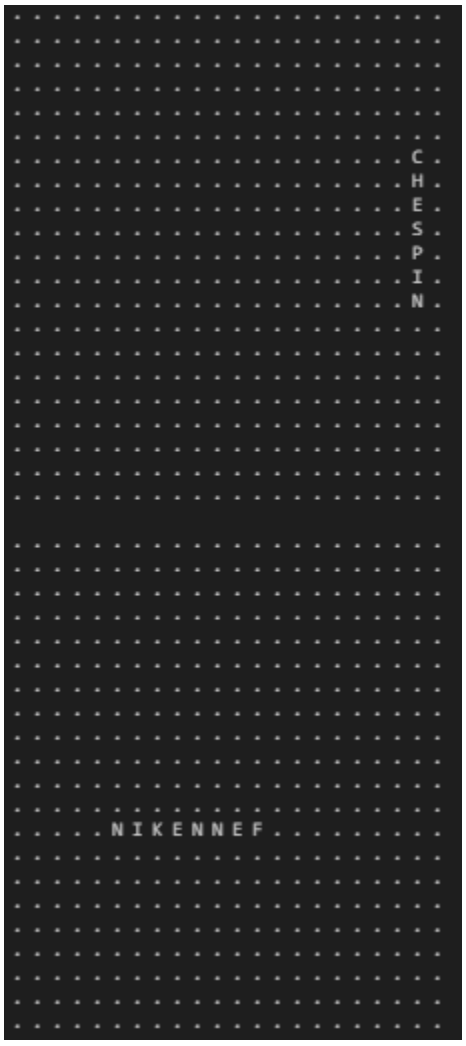
```
T . . . . . .
T . . . . . .
O . . . . . .
W . . . . . .
A . . . . . .
H . . . . . .
S . . . . . .
O . . . . . .
```

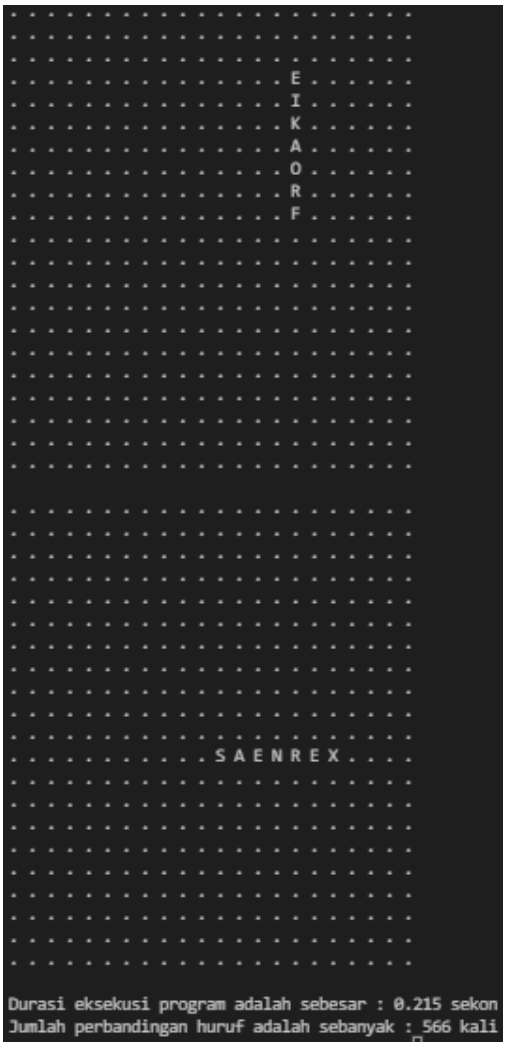
```
. . . . . K . . . .
. . . . . Y . . . .
. . . . . U . . .
. . . . . R . .
. . . . . E .
. . . . . M
```

```
Durasi eksekusi program adalah sebesar : 0.239 sekon
Jumlah perbandingan huruf adalah sebanyak : 511 kali
```

3.

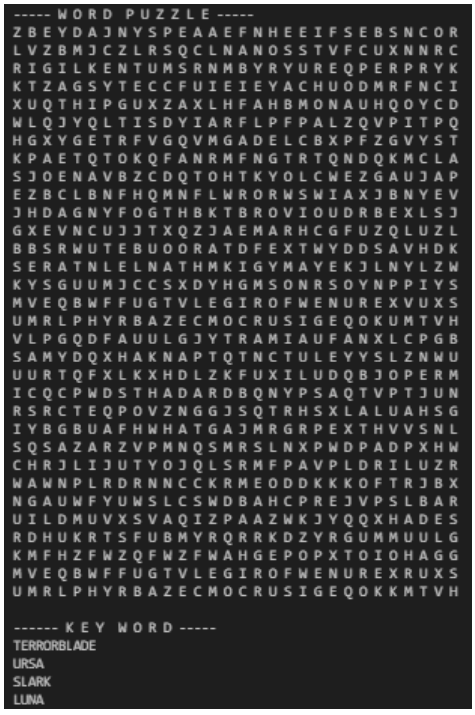
```
----- W O R D  P U Z Z L E -----
S Z Y A K R B F L Q W G Z T I E Z D C N P U
N Y T N G W K B B G B M N M S F V H V H F J
I V C G W B C B O A O K Q B S J E R U G S O
S I T E B F K J W H M Z H R T E V S J Z R P
E Y C I X G I P Q D Y Q F B J I U V U P K I
S A I W W Z C J Y N C N J D T K J I C R J B
C K D C A O Y L O V C Z S D N A I X Y O C G
B Q R O U D M T X A B D J C M O S R X H H I
W E R S K F S P C J M S U G A R W O L R E P
U U L J F V L R T W R Z W F K F D I N A S J
K V Z W I U A N X S Y Q S Q K T I A P P P X
D Q H F C B Y Z R W Z S A E N R E X R J I K
L N H B Z N I K E N N E F S S Q B A S L N Q
J X N A A V Y E X O Q P C K R U X J A H V I
Q O I Y M C N P A Y D O B R V A S I B X B N
G U I A Z Q S R X P C V J D Z Z P Q X G D F
J W I Z C U W I G V B P A K W W N B M O C F
G F A C C G I W T R U T I O B N H Z W V Q F
B W E G I T B C W U K U H V Y U D N V E H G
G M Y R A M I C F S Y D W T N Y T W R Z Y L
W S R T E M L W R X M V B D X N S B Q S W P

------ K E Y  W O R D -----
CHESPIN
FENNEKIN
FROAKIE
XERNEAS
```

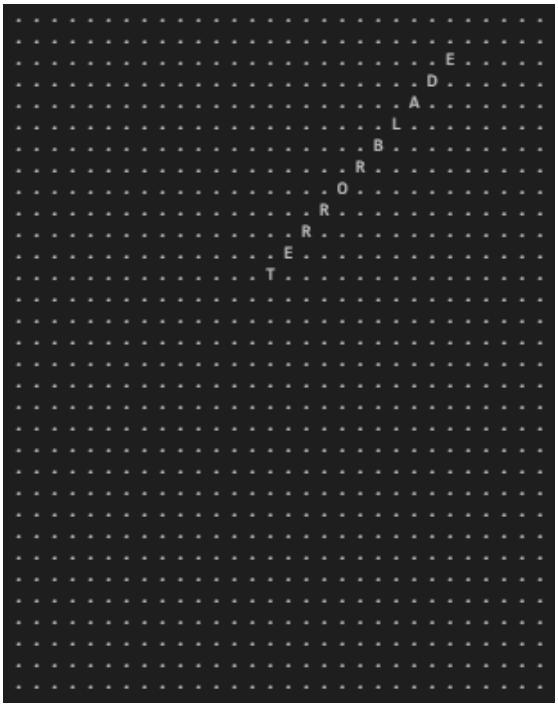


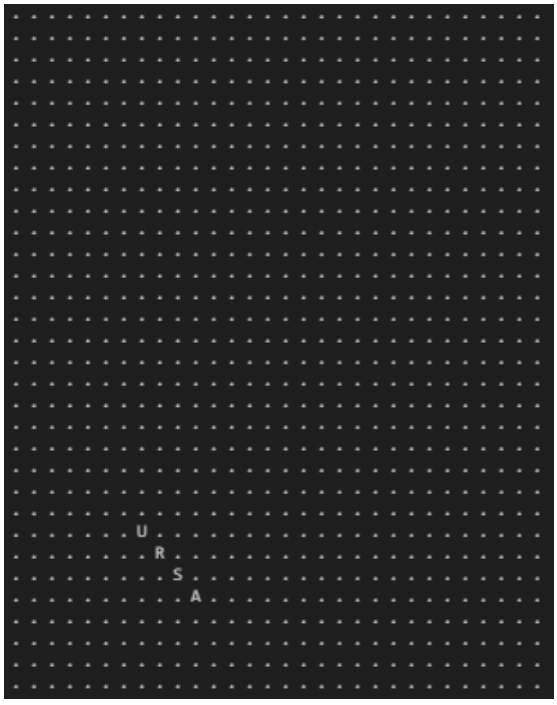


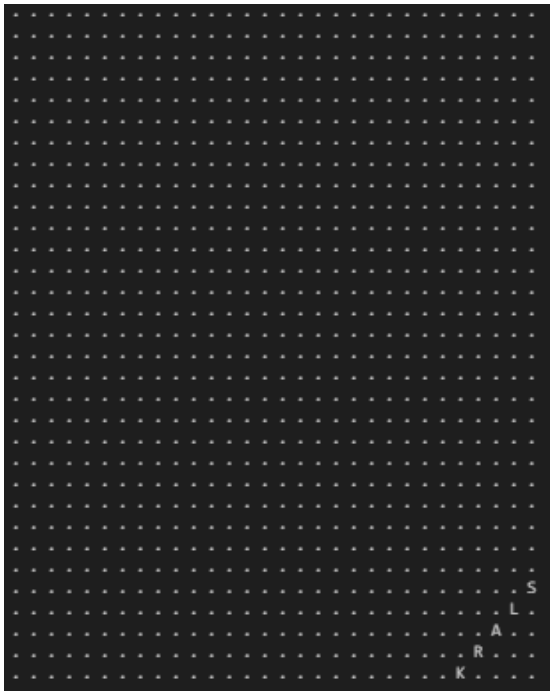
Large (32x30) :

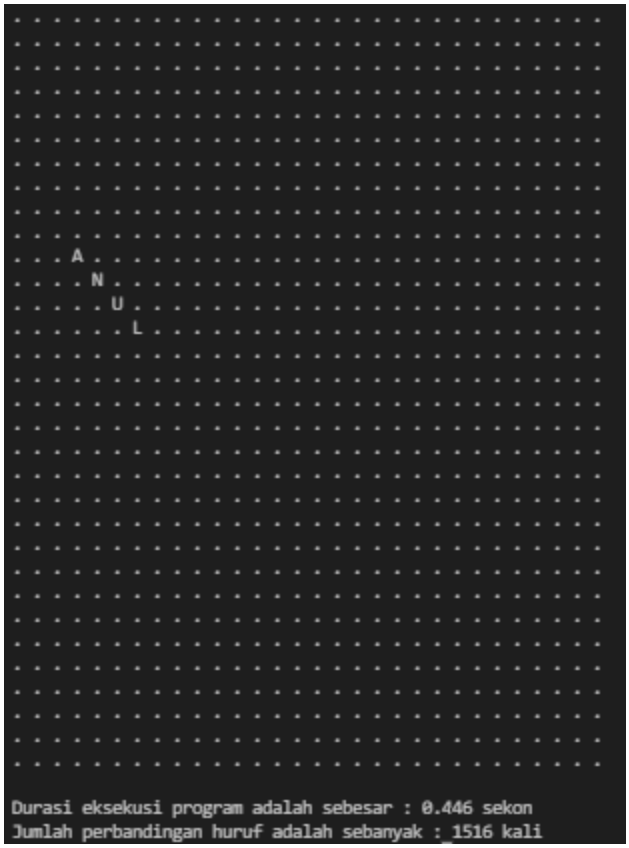
1.



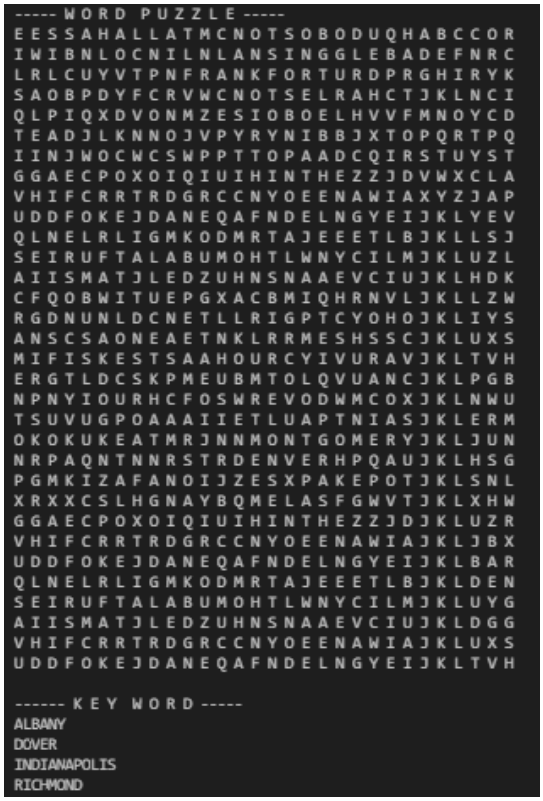


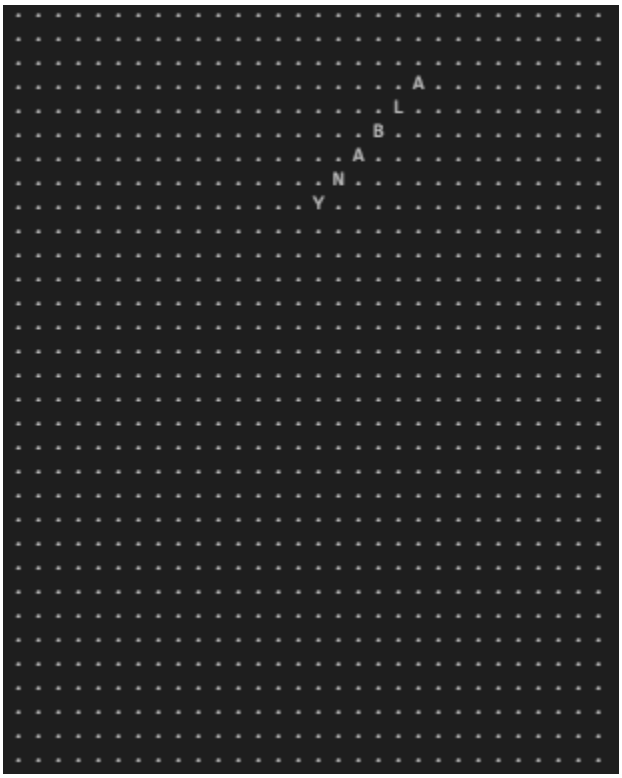


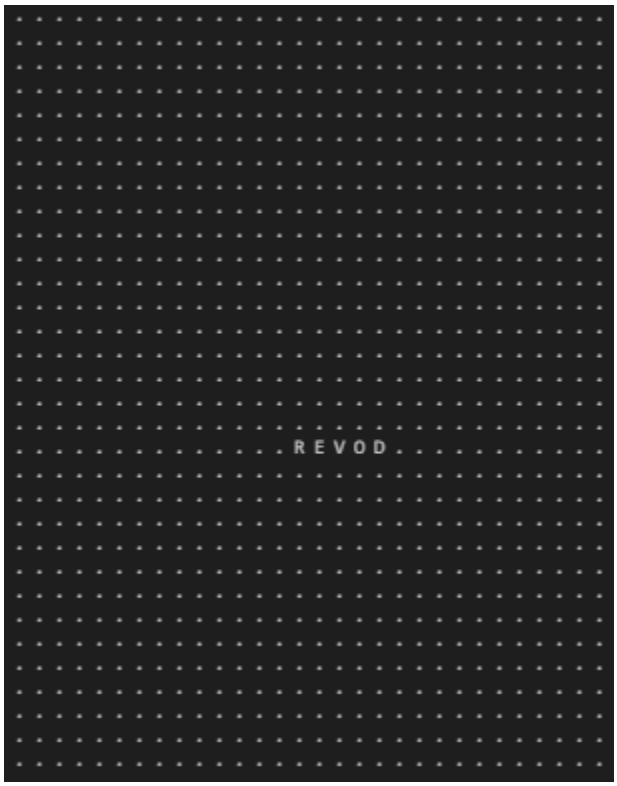


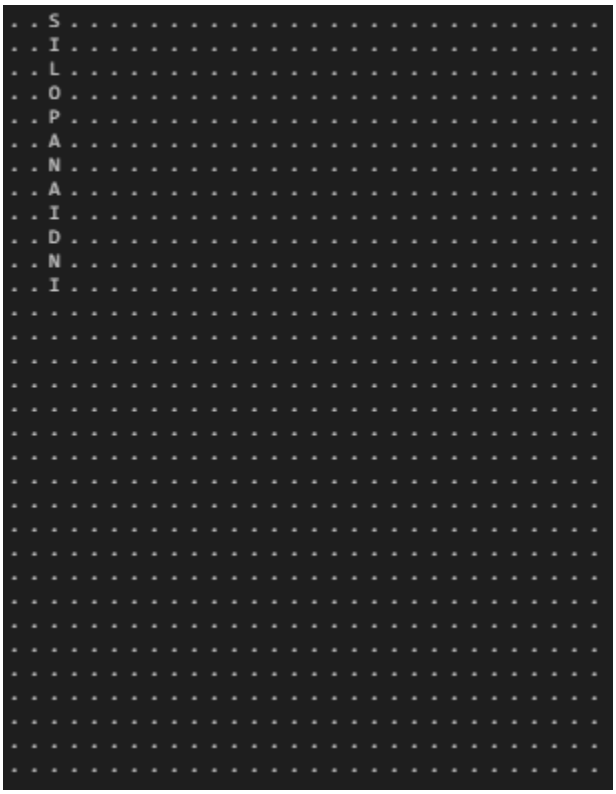

A
N
U
L
Durasi eksekusi program adalah sebesar : 0.446 sekon
Jumlah perbandingan huruf adalah sebanyak : 1516 kali

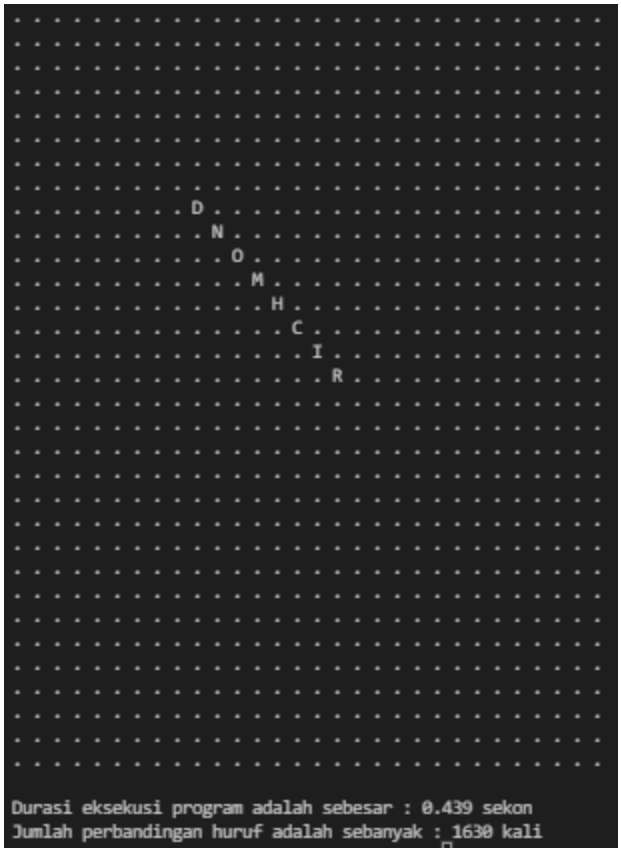
2.


----- WORD PUZZLE -----
EESSAHALLATMCNOTSOBODUQHABCCOR
IWIBNLOCNILNLANSINGGLEBADEFNRC
LRLCUYVTPNFRANKFORTURDPRGHIRYK
SAOBPDYFCRVWCNOTSELRAHCTJKLNCI
QLPIQXDVONMZESIOBOELHVVFMNOYCD
TEADJLKNNOJVPYRYNIBBJXTOPQRTPQ
IINJWOCWCSWPPTTOPAADCQIRSTUYST
GGAECPOXOIQIUIHINTHEZZJDVWXCLA
VHIFCRRTRDGRCCNYOEENAWIAXYZJAP
UDDFOKEJDANEQAFNDELNGYEIJKLYEV
QLNELRLIGMKODMRTAJEEETLBJKLLSJ
SEIRUFTALABUMOHTLWNYCILMJKLUZL
AIISMATJLEDZUHNSNAAEVCIUJKLHDK
CFQOBWITUEPGXACBMIQHRNVLJKLLZW
RGDNUNLDCNETLLRIGPTCYOHOJKLIYS
ANSCSAONEAETNKLRRMESHSSCJKLUXS
MIFISKESTSAAHOURCYIVURAVJKLTVH
ERGTLDCSKPMEUBMTOLQVUANCJKLPGB
NPNYIOURHCFOSWREVODWMCOXJKLNWU
TSUVUGPOAAAIIETLUAPTNIASJKLERM
OKOKUKEATMRJNNMONTGOMERYJKLJUN
NRPAQNTNNRSTRDENVERHPQAUJKLHSG
PGMKIZAFANOIJZESXPAKEPOTJKLSNL
XRXXCSLHGNAYBQMELASFGWVTJKLXHW
GGAECPOXOIQIUIHINTHEZZJDJKLUZR
VHIFCRRTRDGRCCNYOEENAWIAJKLJBX
UDDFOKEJDANEQAFNDELNGYEIJKLBAR
QLNELRLIGMKODMRTAJEEETLBJKLDEN
SEIRUFTALABUMOHTLWNYCILMJKLUYG
AIISMATJLEDZUHNSNAAEVCIUJKLDGG
VHIFCRRTRDGRCCNYOEENAWIAJKLUXS
UDDFOKEJDANEQAFNDELNGYEIJKLTVH
------ KEY WORD -----
ALBANY
DOVER
INDIANAPOLIS
RICHMOND


A
L
B
A
N
Y

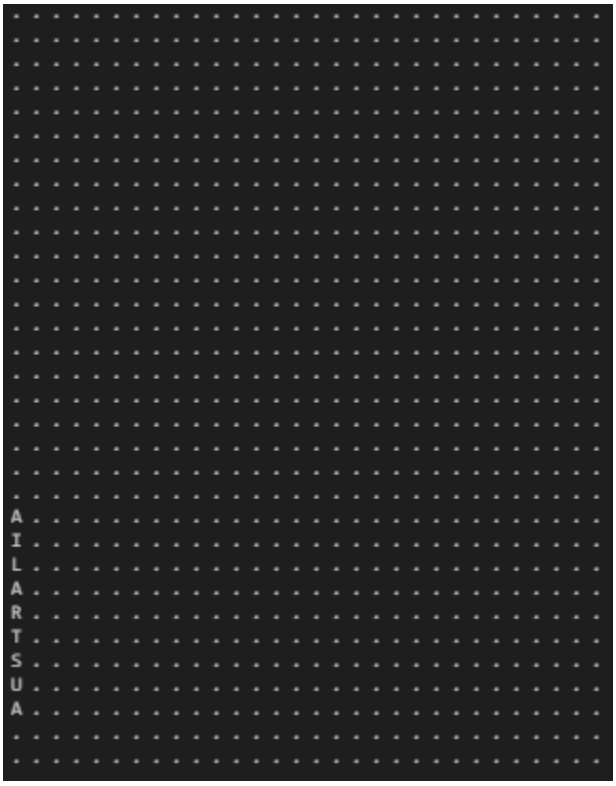

REVOD

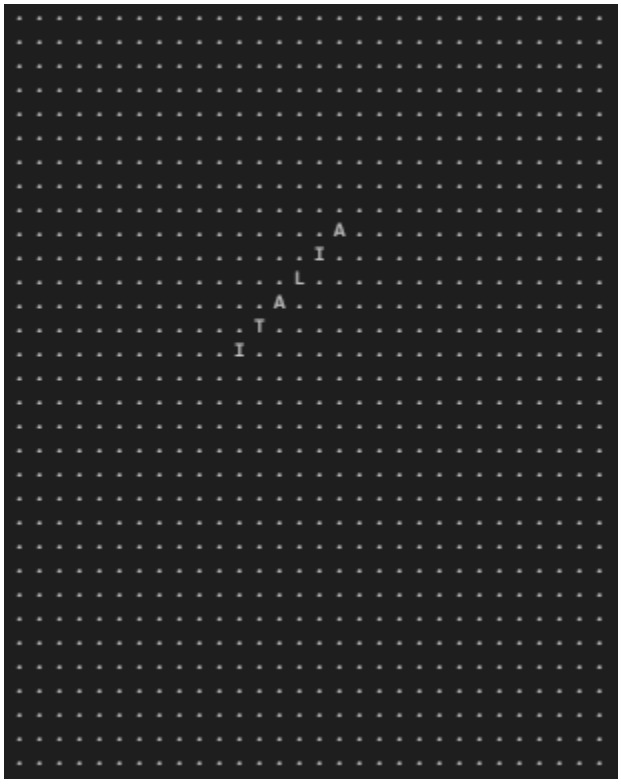

S
I
L
O
P
A
N
A
I
D
N
I

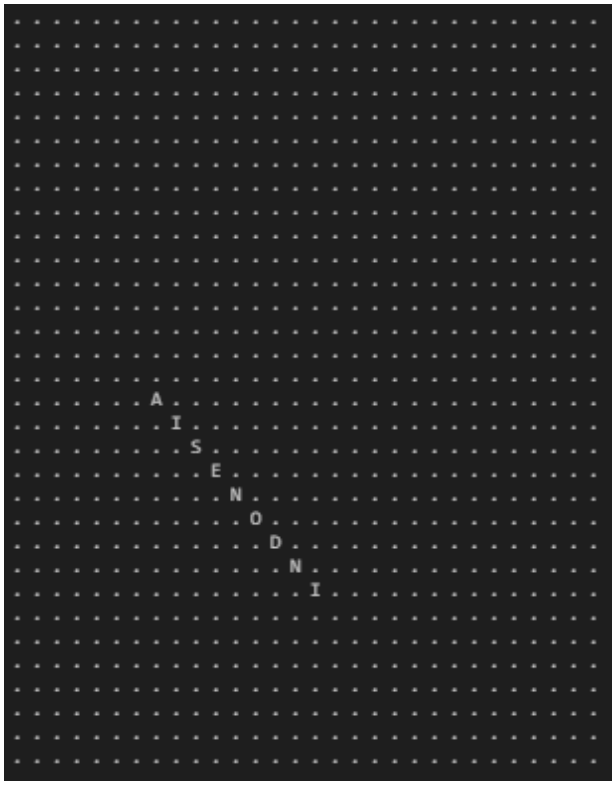

D
N
O
M
H
C
I
R
Durasi eksekusi program adalah sebesar : 0.439 sekon
Jumlah perbandingan huruf adalah sebanyak : 1630 kali

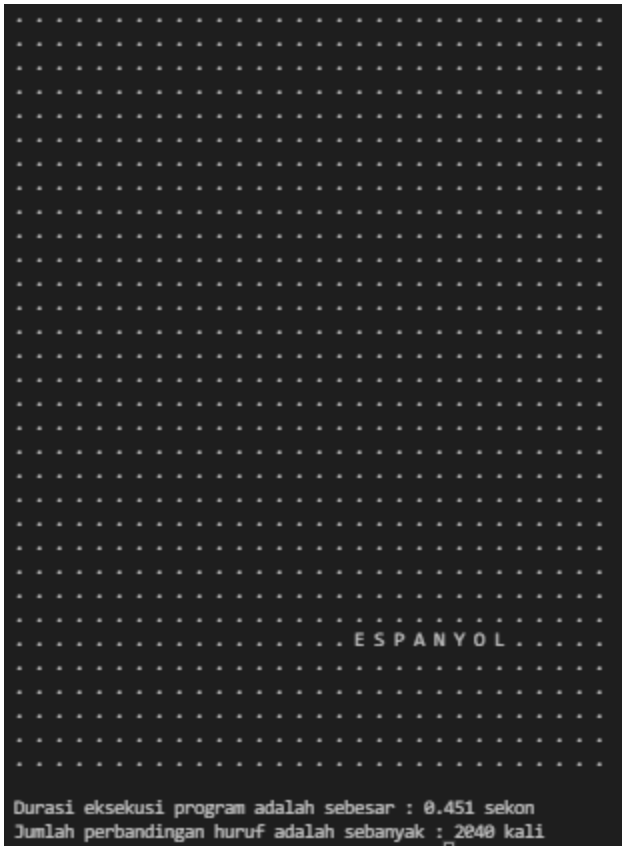
3.

```
----- W O R D  P U Z Z L E -----
E E S S A H A L L A T M C N O T S O B O D U Q H A B C C O R
I W I B N L O C N I L N L A N S I N G G L E B A D E F N R C
L R L C U Y V T P N F R A N K F O R T U R D P R G H I R Y K
S A O B P D Y F C R V W C N O T S E L R A H C T J K L N C I
Q L P I Q X D V O N M Z E S I O B O E L H V V F M N O Y C D
T E A D J L K N N O J V P Y R Y N I B B J X T O P Q R T P Q
I I N J W O C W C S W P P T T O P A A D C Q I R S T U Y S T
G G A E C P O X O I Q I U I H I N T H E Z Z J D V W X C L A
V H I F C R R T R D G R C C N Y O E E N A W I A X Y Z J A P
U D D F O K E J D A N E Q A F N A E L N G Y E I J K L Y E V
Q L N E L R L I G M K O D M R I A J E E E T L B J K L L S J
S E I R U F T A L A B U M O L T L W N Y C I L M J K L U Z L
A I I S M A T J L E D Z U A N S N A A E V C I U J K L H D K
C F Q O B W I T U E P G T A C B M I Q H R N V L J K L L Z W
R G D N U N L D C N E I L L R I G P T C Y O H O J K L I Y S
A N S C S A O N E A E T N K L R R M E S H S S C J K L U X S
M I F I S K E A T S A A H O U R C Y I V U R A V J K L T V H
E R G T L D C S I P M E U B M T O L Q V U A N C J K L P G B
N P N Y I O U R H S F O S W R E V O D W M C O X J K L N W U
T S U V U G P O A A E I I E T L U A P T N I A S J K L E R M
O K O K U K E A T M R N N N M O N T G O M E R Y J K L J U N
A R P A Q N T N N R S T O D E N V E R H P Q A U J K L H S G
I G M K I Z A F A N O I J D E S X P A K E P O T J K L S N L
L R X X C S L H G N A Y B Q N E L A S F G W V T J K L X H W
A G A E C P O X O I Q I U I H I N T H E Z Z J D J K L U Z R
R H I F C R R T R D G R C C N Y O E E N A W I A J K L J B X
T D D F O K E J D A N E Q A F N D E S P A N Y O L K L B A R
S L N E L R L I G M K O D M R T A J E E E T L B J K L D E N
U E I R U F T A L A B U M O H T L W N Y C I L M J K L U Y G
A I I S M A T J L E D Z U H N S N A A E V C I U J K L D G G
V H I F C R R T R D G R C C N Y O E E N A W I A J K L U X S
U D D F O K E J D A N E Q A F N D E L N G Y E I J K L T V H

------ K E Y  W O R D -----
AUSTRALIA
ITALIA
INDONESIA
ESPANYOL
```








ESPANYOL
Durasi eksekusi program adalah sebesar : 0.451 sekon
Jumlah perbandingan huruf adalah sebanyak : 2040 kali

BAB IV
LINK

https://github.com/danielsalim/wordpuzzle