Laporan Tugas Kecil IF2211
Strategi Algoritma

Word Search Puzzle dengan Brute Force

oleh
Aloysius Gilang Pramudya (13520147)

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

2022

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.	√	

A. Algoritma Brute Force

Langkah-langkah algoritma *Brute Force* yang digunakan pada program ini adalah sebagai berikut :

1. Untuk setiap kata (*pattern*) yang akan dicari, carilah huruf pertama dari kata yang akan dicari pada puzzle.
2. Jika huruf pertama ditemukan dalam puzzle, cek huruf lain di sekitar huruf tersebut ke empat arah yaitu : kiri ke kanan, atas ke bawah, kiri atas ke kanan bawah, dan kiri bawah ke kanan atas.
3. Untuk setiap arah pemeriksaan, jika sisa baris atau kolom puzzle tidak mencukupi untuk membentuk kata yang ingin dicari maka tidak dilakukan pengecekan lebih lanjut.
4. Untuk setiap arah pemeriksaan, jika sisa baris atau kolom puzzle cukup untuk membentuk kata yang ingin dicari, maka lakukan pengecekan tiap huruf lebih lanjut. Jika terdapat huruf yang tidak sesuai maka tidak dilanjutkan pengecekan lebih lanjut.
5. Lakukan langkah-langkah sebelumnya dengan kata (*pattern*) dibalik. Contoh: EARTH, cari juga untuk HTREA.
6. Jika kata yang ingin dicari berhasil dibentuk, maka tampilkan *output.*

B. Source Code Program

Source Code untuk program utama adalah sebagai berikut :

```
#include <stdio.h>
#include "charmachine.h"
#include "wordmachine.h"
#include "boolean.h"
#include "listpos.h"
#include "listOfPoint.h"
#include <string.h>
#include <stdlib.h>
#include <time.h>


char* reversedString(char *string)
{
   int length = strlen(string);
```

```
    char* reversed = (char*)malloc((length+1) * sizeof(char));
    for(int i=0;i<length;i++)
    {
      reversed[(length-1)-i]=string[i];
    }
    reversed[length] = '\0';
    return reversed;
}

void printOutput(ListPoint L, int row, int col,char words[row][col]){

    int i; int j;
    for (i = 0; i < row; i++){
        for (j = 0; j<col; j++){
            if (inList(L, i,j)){
                printf(" %c ", words[i][j]);
            } else {
                printf(" - ");
            }
        } printf("\n");
    }
    printf("\n \n");
}

int searchKiriKanan (int row,int col,char words[row][col], char* pattern, Point P){

    ListPoint l;
    int comparison = 0;
    CreateListPoint(&l);
    int a = P.x; int b = P.y; int c = 0;
    if (b + strlen(pattern) -1 < col ){
        boolean equal = true;
        while (b < (P.y + strlen(pattern)) && equal){
            if (words[a][b] == pattern[c]){
                insertLastPoint(&l,makePoint(a,b));
                b++; c++;
            } else {
                equal = false;
            }
```

```
            comparison++;
        }
        if (equal){
            printOutput(l,row,col,words);
        }
    }
    return comparison;
}

int searchAtasBawah (int row, int col,char words[row][col], char* pattern, Point P){

    ListPoint l;
    int comparison = 0;
    CreateListPoint(&l);
    int a = P.x; int b = P.y; int c = 0;
    if (a + strlen(pattern) -1 < row ){
        boolean equal = true;
        while (a < (P.x + strlen(pattern)) && equal){
            if (words[a][b] == pattern[c]){
                insertLastPoint(&l,makePoint(a,b));
                a++; c++;
            } else {
                equal = false;
            }
            comparison++;
        }
        if (equal){
            printOutput(l,row,col,words);
        }
    }
    return comparison;
}

int searchKiriAtasKananBawah (int row, int col,char words[row][col], char* pattern, Point P){

    ListPoint l;
    int comparison = 0;
    CreateListPoint(&l);
    int a = P.x; int b = P.y; int c = 0;
```

```
    if (a + strlen(pattern) -1 < row && b + strlen(pattern) -1 < col){
        boolean equal = true;
        while (a < (P.x + strlen(pattern))&& b < (P.y + strlen(pattern)) && equal){
            if (words[a][b] == pattern[c]){
                insertLastPoint(&l,makePoint(a,b));
                a++; c++; b++;
            } else {
                equal = false;
            }
            comparison++;
        }
        if (equal){
            printOutput(l,row,col,words);
        }
    }
    return comparison;
}

int searchKiriBawahKananAtas (int row, int col,char words[row][col], char* pattern, Point P){

    ListPoint l;
    int comparison = 0;
    CreateListPoint(&l);
    int a = P.x; int b = P.y; int c = 0; int d = a - strlen(pattern) + 1;
    if ((d >= 0) && b + strlen(pattern) -1 <= col){
        boolean equal = true;
        while (a >= (P.x - strlen(pattern) + 1)&& b < (P.y + strlen(pattern)) && equal){
            if (words[a][b] == pattern[c]){
                insertLastPoint(&l,makePoint(a,b));
                a--; c++; b++;
            } else {
                equal = false;
            }
            comparison++;
        }
        if (equal){
            printOutput(l,row,col,words);
        }
    }
```

```
    return comparison;
}

// ini di loop sebanyak pattern
int searchSameFirstChar (int row, int col, char words[row][col], char* pattern){
    Point P;
    int i = 0; int j = 0;
    boolean match = false;
    int comparison = 0;
    for (i = 0; i < row; i++){
        for(j = 0; j < col;j++){
            comparison++;
            if (words[i][j] == pattern[0]) {
                P.x = i;
                P.y = j;
                match = true;
                // return P;
                comparison += searchKiriKanan(row,col,words,pattern,P);
                comparison += searchAtasBawah(row,col,words,pattern,P);
                comparison += searchKiriAtasKananBawah(row,col,words,pattern,P);
                comparison += searchKiriBawahKananAtas(row,col,words,pattern,P);
            }
        }
    }
    return comparison;
}

int main (){



    char title[100] ={' '};
    printf("Word Search Puzzle by Aloysius Gilang\n");
    printf("Masukan nama file (small1, small2, small3, medium1, medium2, medium3, large1,large2, large3) +
.txt\n");
    // scanf("%s",title);
    char str1[100] = "../test/", str2[100];
    scanf("%s",str2);
    char filename[100];
    int v = 0, w = 0;
```

```
// Insert the first string in the new string
while (str1[v] != '\0') {
    filename[w] = str1[v];
    v++;
    w++;
}

// Insert the second string in the new string
v = 0;
while (str2[v] != '\0') {
    filename[w] = str2[v];
    v++;
    w++;
}
filename[w] = '\0';
int row = 0;int col = 0;int countChar = 0;int countPattern = 0;
startWord(filename);
if (!fileFound) {
    printf("file tidak ditemukan");
    return 0;
}
while (currentChar != NEWLINE && eot != 1 && currentChar != MARK) {
    countChar++;
    advWord();
    if (currentChar == NEWLINE) {
        skipNewline();
        row++;
    }
}
col = countChar /row;
printf("Rows: %d\n",row);printf("Cols: %d\n",col);
char words[row][col];
startWord(filename);
int i = 0; int j = 0;
while (currentChar != NEWLINE){
    advWord();
    words[i][j] = KataToChar(currentWord);
    if (currentChar == NEWLINE) {
```

```
            skipNewline();
            j = 0;
            i++;
        } else {
            j++;
        }
    }
    //printing words
    for (i=0; i<row; i++)
    {
        for(j=0; j<col; j++)
        {
            printf("%c ", words[i][j]);
        }
        printf("\n");
    }
    skipNewline();
    advWord();
    ListPos pattern;
    ListPoint arrPoint;
    CreateListPoint(&arrPoint);
    CreateListPos(&pattern);
    while (currentChar != MARK){
        countPattern++;
        insertLast(&pattern, KataToString(currentWord));
        insertLast(&pattern, reversedString(KataToString(currentWord)));
        if (currentChar == NEWLINE) {
            skipNewline();
        }
        advWord();
    }
    printf("patterns:%d\n", countPattern);
    printf("\n");

    // element checking
    clock_t start = clock();
    int sumComparison = 0;
    for (i = 0; i < length(pattern);i++){
        sumComparison += searchSameFirstChar(row,col,words,pattern.contents[i]);
```

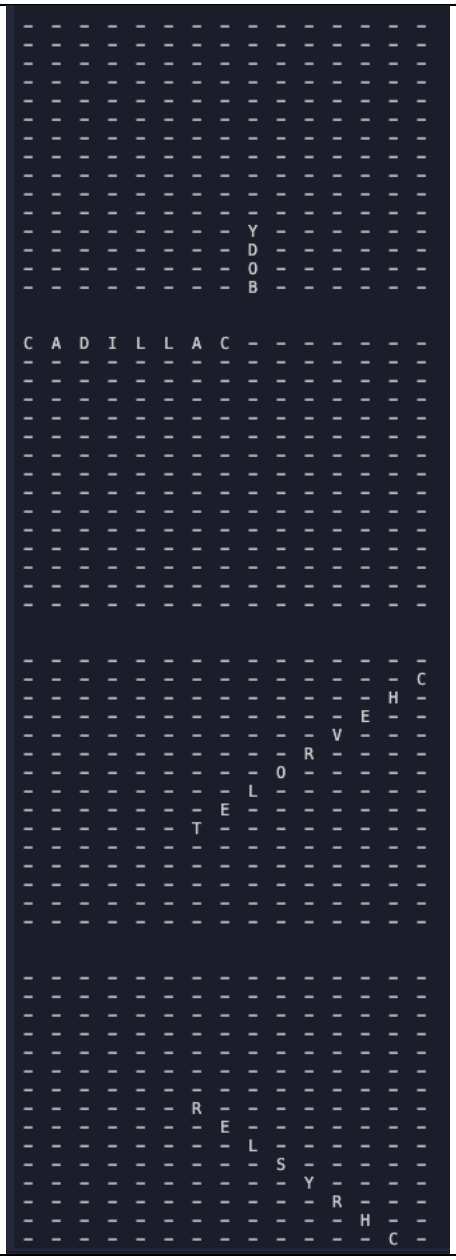
```
    }
    clock_t end = clock();
    float time =  (float)(end - start) / CLOCKS_PER_SEC;
    // printf("total comparison : %d ",comparison);
    printf("Time elapsed : %f\n", time);
    printf("Total comparison : %d\n", sumComparison);
    printf("\n");


    return 0;


}
```

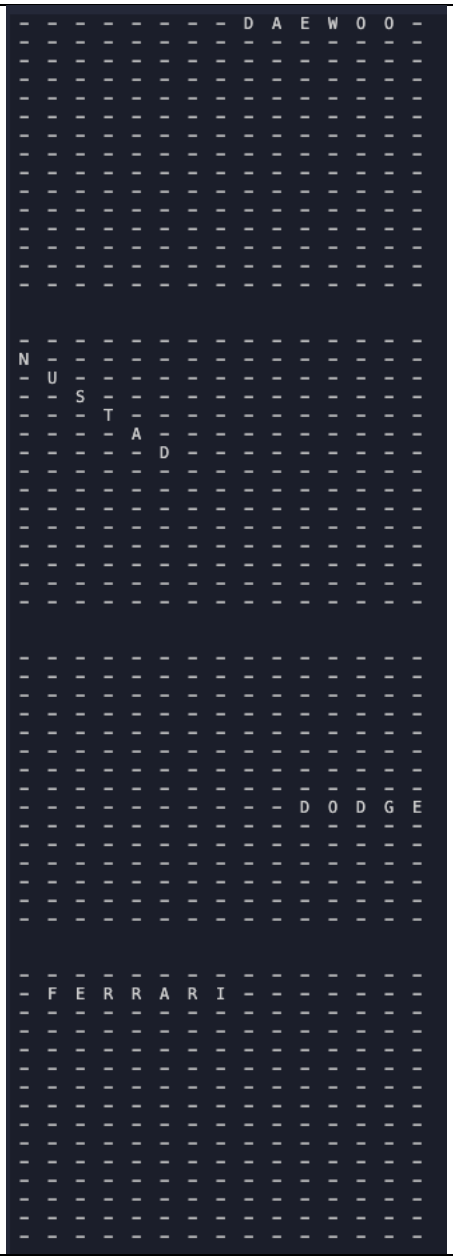
C. Test Case Program

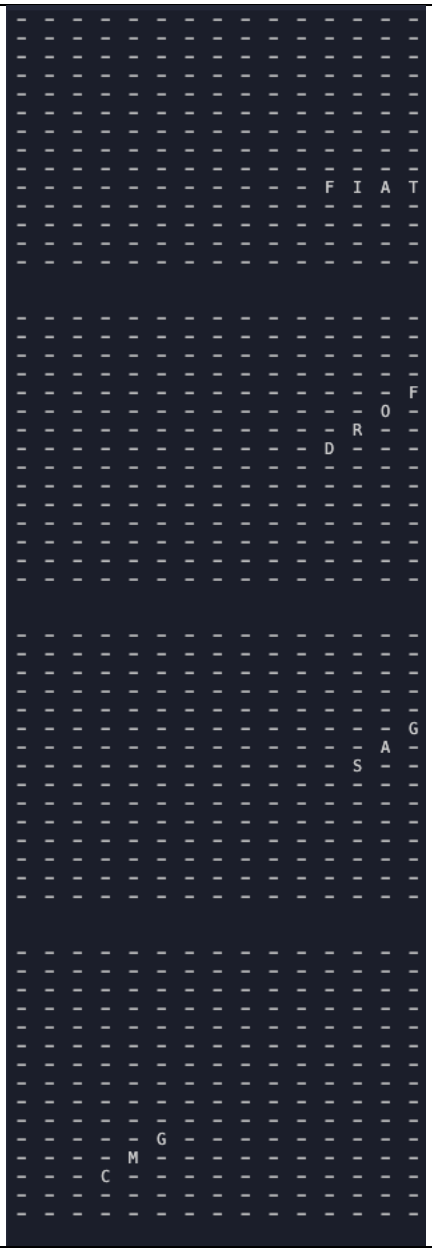
Small Puzzle 1 Input :	Output:

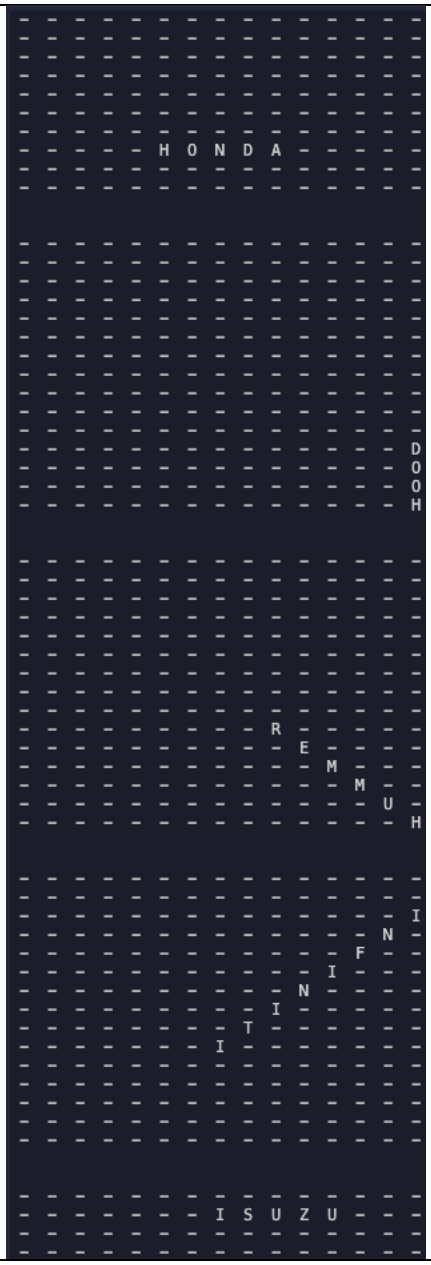
```
C A D I L L A C D A E W O O M
N F E R R A R I S U Z U T E C
N U K L I N C O L N V O R H I
M A S C S U B A R U Y C E N S
O A S T I V O L V O E V F A F
L P S S A U D I T D R I T O G
D O W E I D B A E O N U R A P
S R P M R N R S L I R D S E E
M S O W B A A E T N D O D G E
O C N H A C T I L R A U G A J
B H T I U G L I N S E F I A T
I E I R M E O C Y I Y M L E D
L R A C X H O N D A M R M S O
E I C U P L Y M O U T H H U O
K T S M I T S U B I S H I C H
      You, 36 minutes ago • finished buil
ACURA
AUDI
BMW
BODY
CADILLAC
CHEVROLET
CHRYSLER
DAEWOO
DATSUN
DODGE
FERRARI
FIAT
FORD
GAS
GMC
HONDA
HOOD
HUMMER
INFINITI
ISUZU
JAGUAR
JEEP
KIA
LEXUS
LINCOLN
MASERATI
MINI
MITSUBISHI
NISSAN
OLDSMOBILE
PLYMOUTH
PONTIAC
PROSCHE
SATURN
SUBARU
TIRE
TOYOTA
VOLVO
```

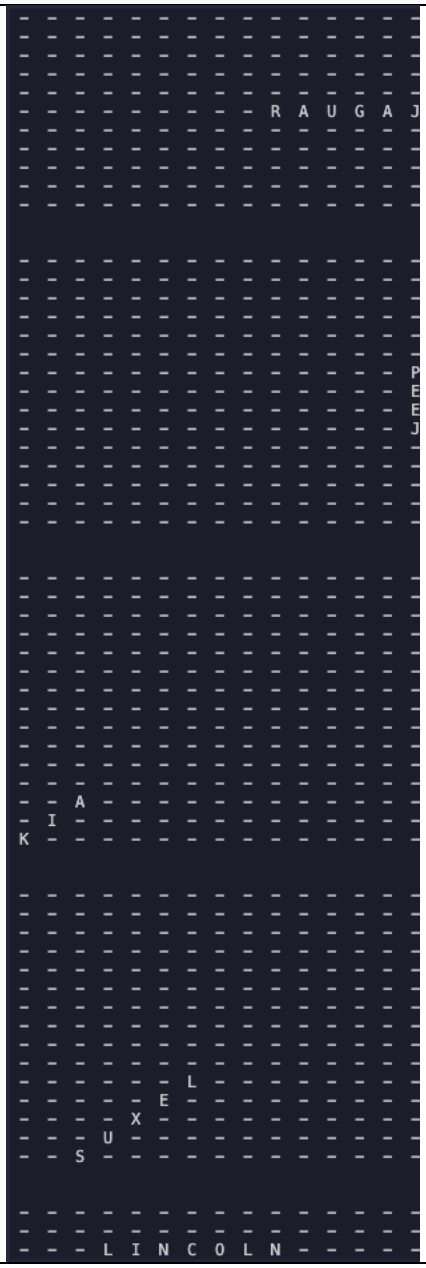
```
Word Search Puzzle by Aloysius Gilang
Masukan nama file (small1, small2, small3, med
small1.txt
Rows: 15
Cols: 15
C A D I L L A C D A E W O O M
N F E R R A R I S U Z U T E C
N U K L I N C O L N V O R H I
M A S C S U B A R U Y C E N S
O A S T I V O L V O E V F A F
L P S S A U D I T D R I T O G
D O W E I D B A E O N U R A P
S R P M R N R S L I R D S E E
M S O W B A A E T N D O D G E
O C N H A C T I L R A U G A J
B H T I U G L I N S E F I A T
I E I R M E O C Y I Y M L E D
L R A C X H O N D A M R M S O
E I C U P L Y M O U T H H U O
K T S M I T S U B I S H I C H
patterns:39

- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - A - - - - - - - -
- - - - - C - - - - - - - - - -
- - - - U - - - - - - - - - - -
- - - R - - - - - - - - - - - -
- - A - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -


- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - A U D I - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -


- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - W - - - - - - - - - - - - -
- - - M - - - - - - - - - - - -
- - - - B - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
```

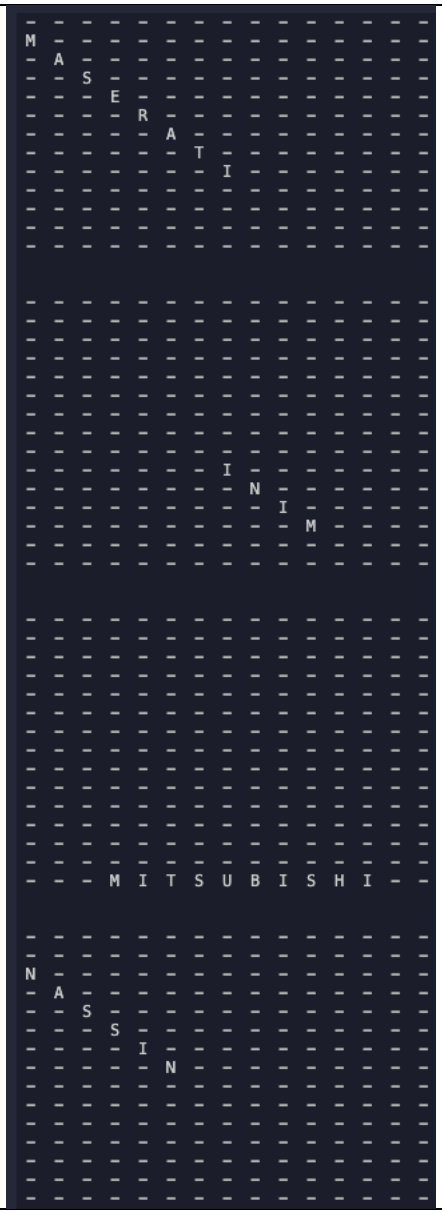

YDOB
CADILLAC
CHEVROLET
CHEVROLET

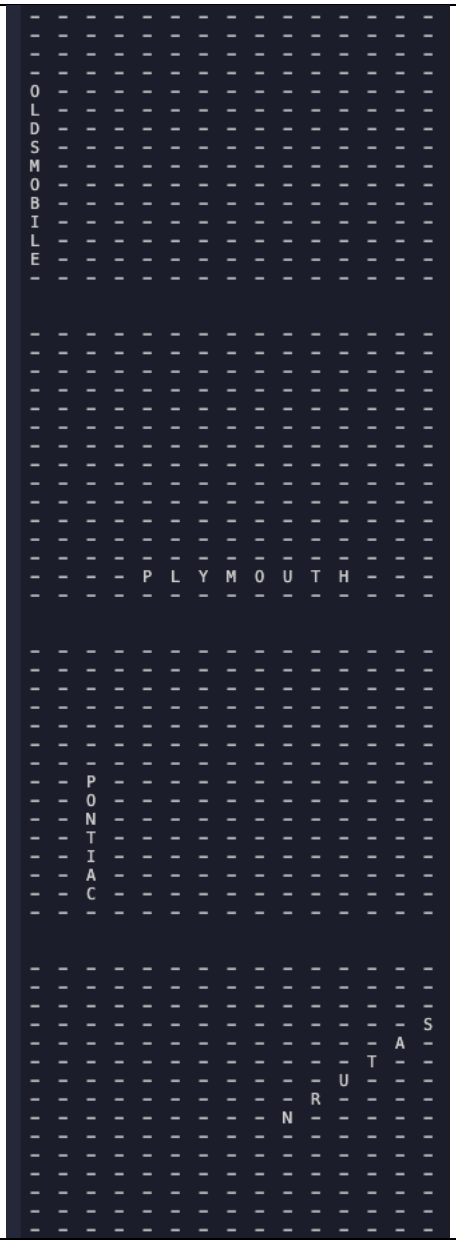

DAEWOO
NUSTAD
DODGE
FERRARI

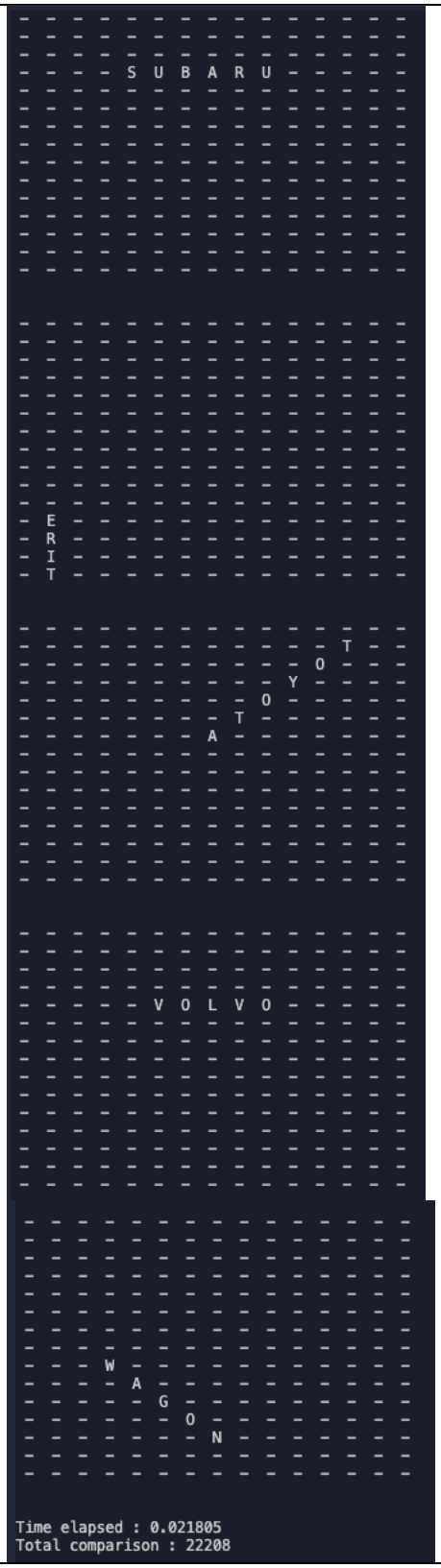

F I A T
F O R D
G A S
G M C


HONDA
DOOH
REMMUH
INFINITI
ISUZU

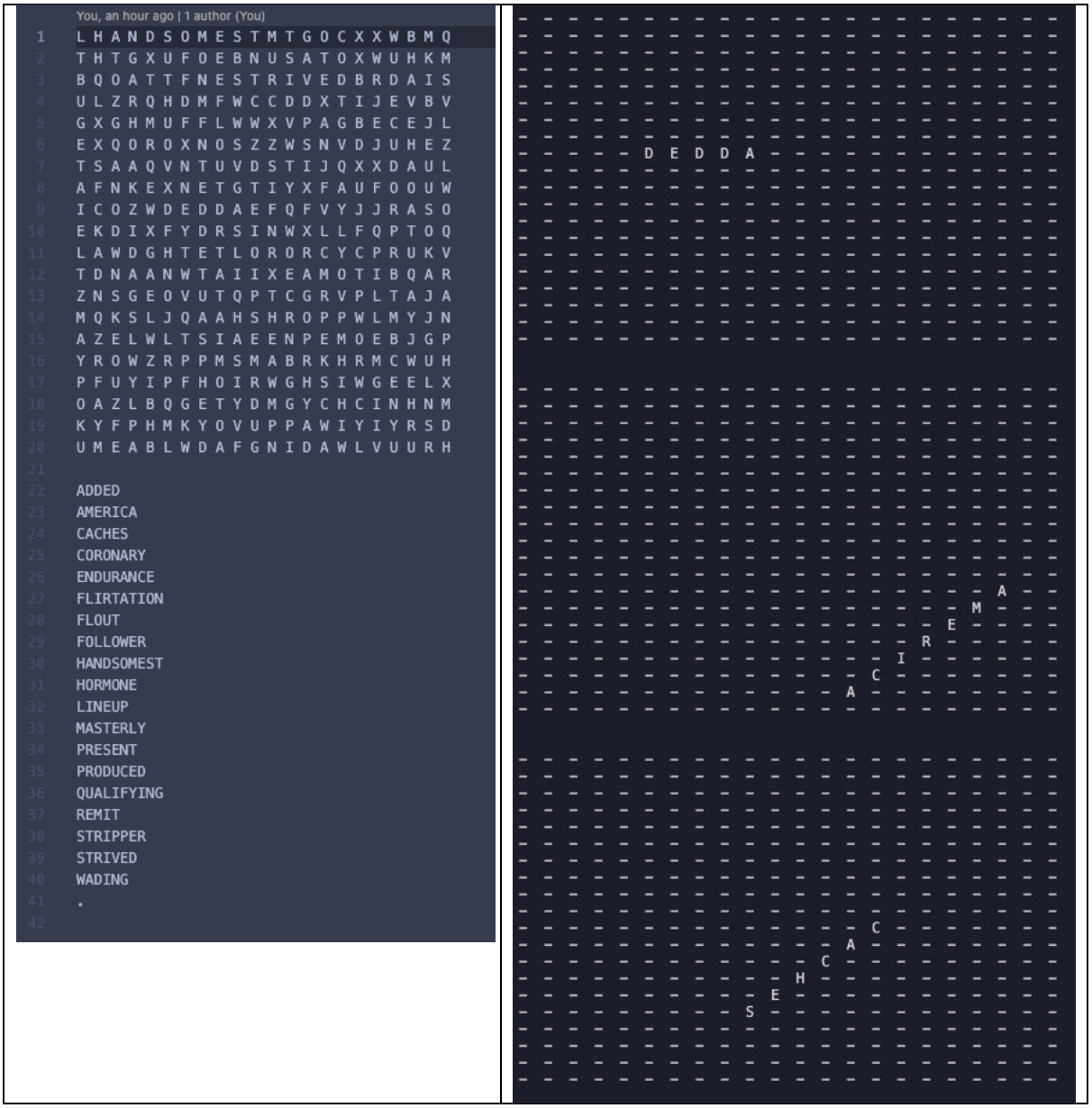

R A U G A J
P
E
E
J
A
I
K
L
E
X
U
S
L I N C O L N

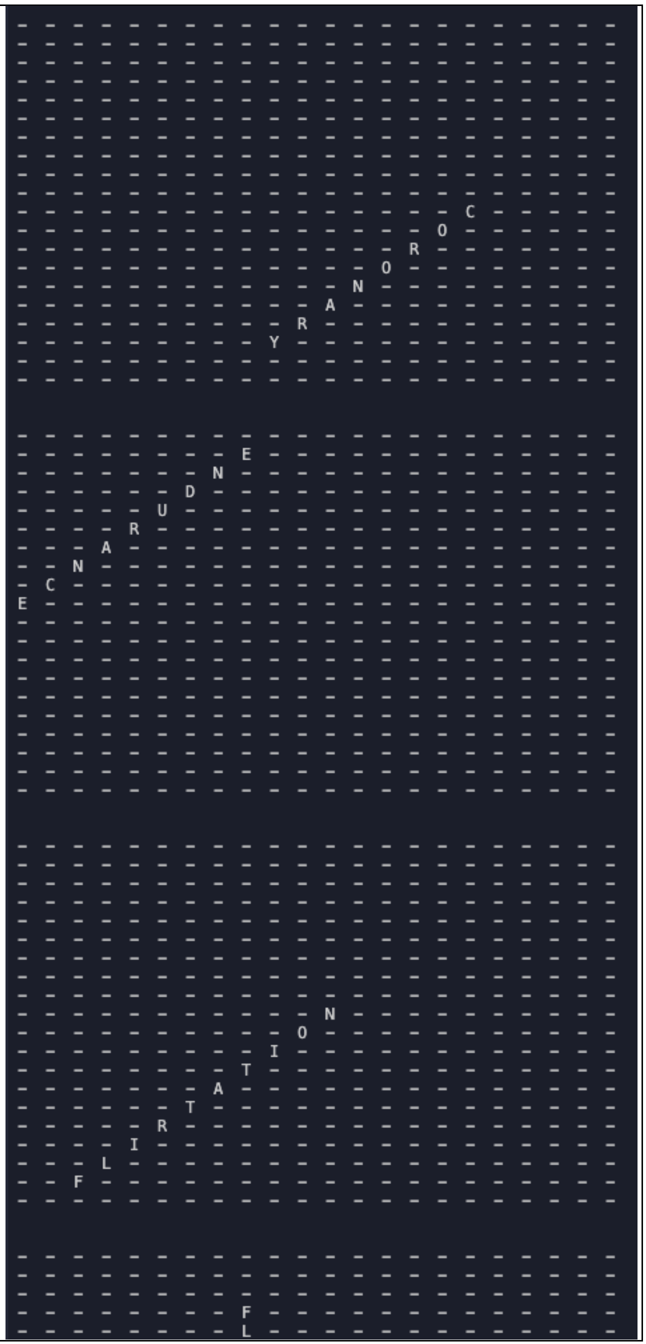

MASERATI
INIM
MITSUBISHI
NASSIN

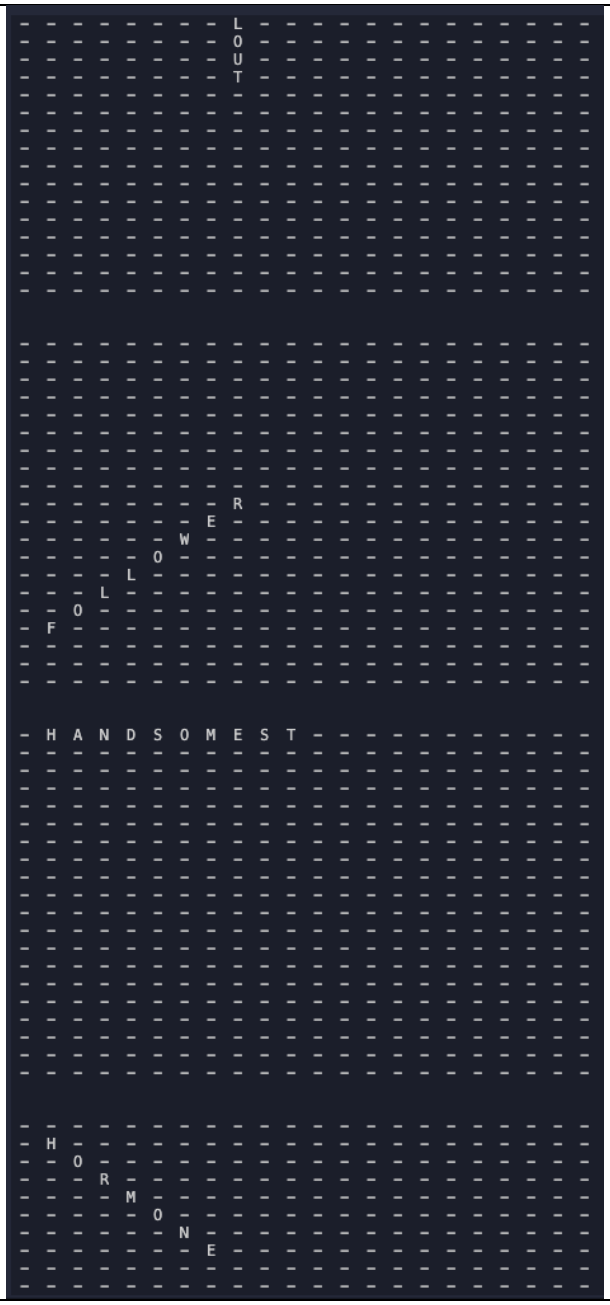

OLDSMOBILE
PLYMOUTH
PONTIAC
NURSTAS

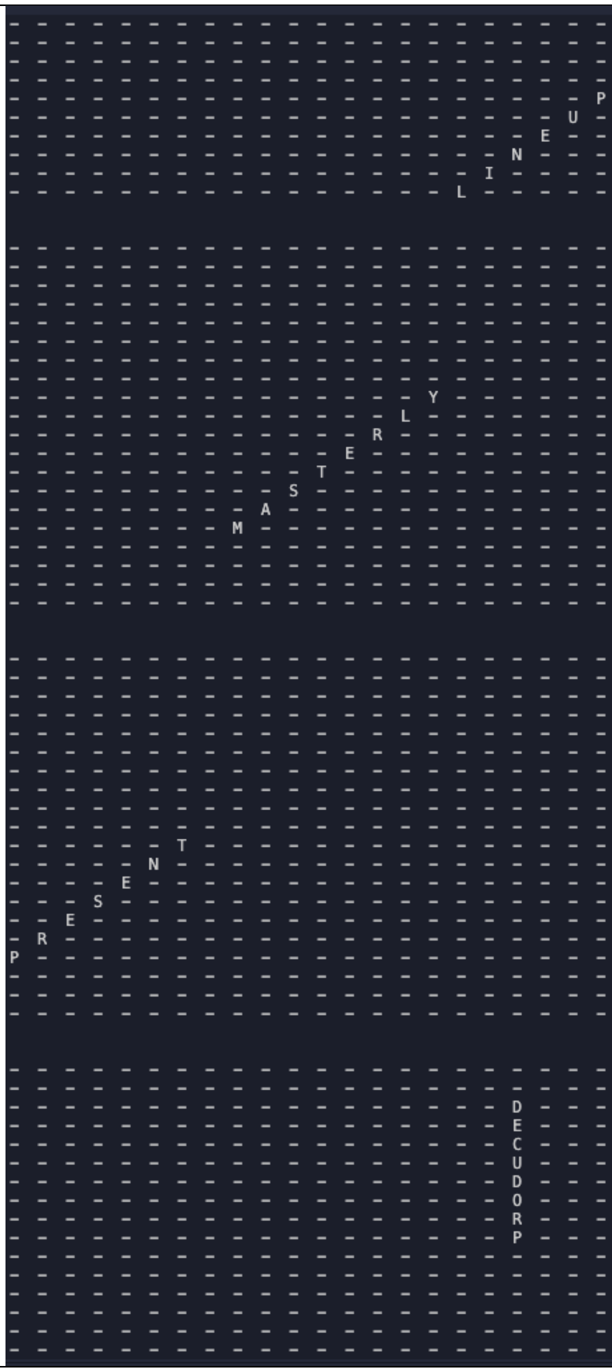


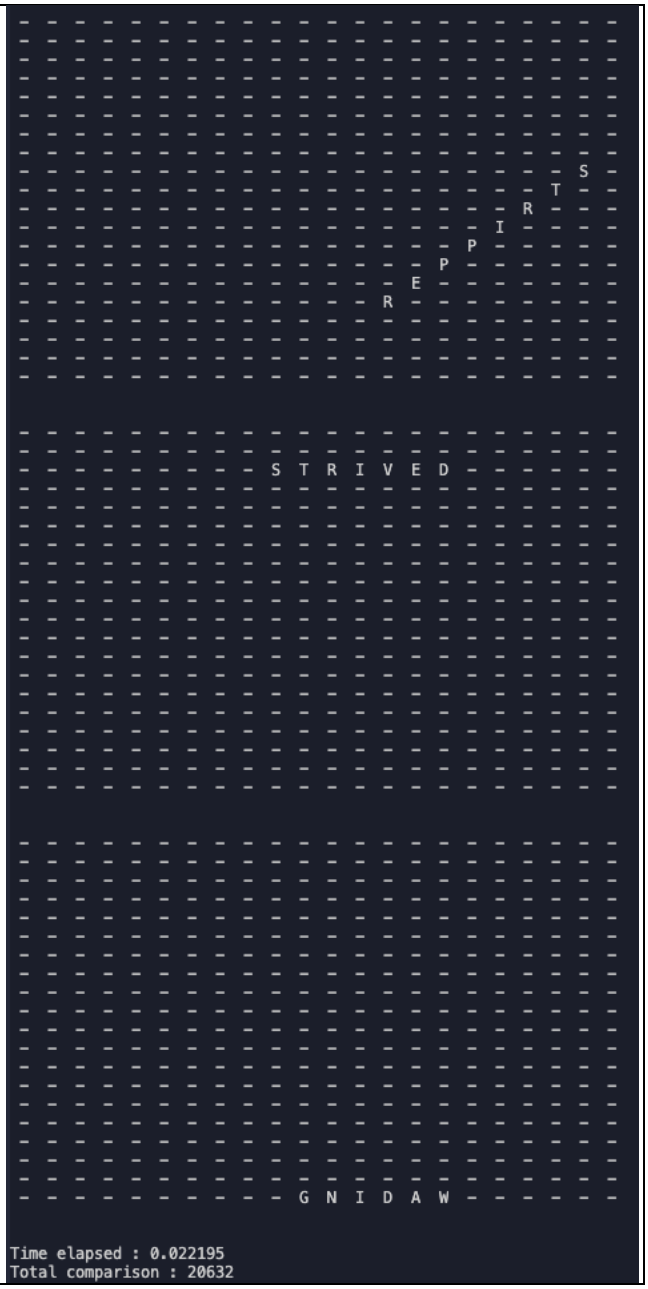
Medium Puzzle 1 Input :	Output :


You, an hour ago | 1 author (You)
L H A N D S O M E S T M T G O C X X W B M Q
T H T G X U F O E B N U S A T O X W U H K M
B Q O A T T F N E S T R I V E D B R D A I S
U L Z R Q H D M F W C C D D X T I J E V B V
G X G H M U F F L W W X V P A G B E C E J L
E X Q O R O X N O S Z Z W S N V D J U H E Z
T S A A Q V N T U V D S T I J Q X X D A U L
A F N K E X N E T G T I Y X F A U F O O U W
I C O Z W D E D D A E F Q F V Y J J R A S O
E K D I X F Y D R S I N W X L L F Q P T O Q
L A W D G H T E T L O R O R C Y C P R U K V
T D N A A N W T A I I X E A M O T I B Q A R
Z N S G E O V U T Q P T C G R V P L T A J A
M Q K S L J Q A A H S H R O P P W L M Y J N
A Z E L W L T S I A E E N P E M O E B J G P
Y R O W Z R P P M S M A B R K H R M C W U H
P F U Y I P F H O I R W G H S I W G E E L X
O A Z L B Q G E T Y D M G Y C H C I N H N M
K Y F P H M K Y O V U P P A W I Y I Y R S D
U M E A B L W D A F G N I D A W L V U U R H
ADDED
AMERICA
CACHES
CORONARY
ENDURANCE
FLIRTATION
FLOUT
FOLLOWER
HANDSOMEST
HORMONE
LINEUP
MASTERLY
PRESENT
PRODUCED
QUALIFYING
REMIT
STRIPPER
STRIVED
WADING
.




LOUT
R
E
W
O
L
L
O
F
HANDSOMEST
H
O
R
M
O
N
E


P
U
E
N
I
L
Y
L
R
E
T
S
A
M
T
N
E
S
E
R
P
D
E
C
U
D
O
R
P



Test Case untuk Small2, Small3, Medium2, Medium3, Large1, Large2, Large3 tidak dicantumkan agar jumlah halaman laporan tidak terlalu banyak.

D. Alamat Repository Github
https://github.com/Aloysiusgilang/Tucil-1-Stima.git