

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(curlength) (curlength).rowEff
11   #define COL(curlength) (curlength).colEff
12   #define ELMT(curlength, i, j) (curlength).contents[(i)][(j)]
13   #define NEFF(L) (L).neff
14   #define ELMTL(L, i) (L).contents[(i)]
15
16   typedef char EType;
17
18   typedef struct {
19       EType contents[ROW_SIZE][COL_SIZE];
20       int rowEff;
21       int colEff;
22   } Matrix;
23
24   typedef struct {
25       EType 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

```
src > C wordpuzzle.c > isFoundEast(int)
1  #include "boolean.h"
2  #include "wordpuzzle.h"
3  #include "time.h"
4  #include <stdio.h>
5  #include <string.h>
6  #include <stdlib.h>
7
8  void readFile() {
9      printf("Masukkan nama file puzzle word search: ");
10     char wordpuzzlefile[50];
11     gets(wordpuzzlefile);
12
13     opentxtfile = fopen(wordpuzzlefile, "r");
14
15     char tokenMap = fgetc(opentxtfile);
16     char beforeToken = tokenMap;
17     ROW(wordboard) = 0; COL(wordboard) = 0;
18     int rows = 0; int cols = 0;
19     while (tokenMap != '\n' || beforeToken != '\n') {
20         if (tokenMap == '\n') {
21             ROW(wordboard)++;
22             rows++;
23             COL(wordboard) = cols;
24             cols = 0;
25         }
26         if (tokenMap != ' ' && tokenMap != '\n') {
27             ELMT(wordboard, rows, cols) = tokenMap;
28             cols++;
29         }
30         beforeToken = tokenMap;
31         tokenMap = fgetc(opentxtfile);
32     }
33
34     NEFF(wordsearched) = 0;
35     char line[100];
36     while (fgets(line, sizeof(line), opentxtfile)) {
37         strcpy(ELMTL(wordsearched, NEFF(wordsearched)), line);
38         NEFF(wordsearched)++;
39     }
40
41     for (int i = 0; i < NEFF(wordsearched)-1; i++) {
42         ELMTL(wordsearched, i)[strlen(ELMTL(wordsearched, i))-1] = '\0';
43     }
44
45     fclose(opentxtfile);
46     counter = 0;
47 }
48
49 boolean isFoundEast(int n){
50     boolean found = false;
51     int wordlength = strlen(ELMTL(wordsearched,n));
52     for (int i = 0; i < ROW(wordboard); i++){
53         for (int j = 0; j < COL(wordboard); j++){
54             int currlength = 0;
55             while ((currlength < wordlength) && (ELMT(wordboard,i,j+currlength) == ELMTL(wordsearched,n)[currlength])){
56                 currlength++;
57                 counter++;
58             }
59             if (currlength == wordlength){
60                 for (int y = 0; y < ROW(wordboard); y++){
```

```

61         for (int z = 0; z < COL(wordboard); z++){
62             if (y == i && z >= j && z <= j + wordlength - 1){
63                 printf("%c ",(ELMT(wordboard, y, z)));
64             }
65             else {
66                 printf(" ",(ELMT(wordboard, y, z)));
67             }
68         }
69         printf("\n");
70     }
71     printf("\n");
72     return true;
73 }
74 }
75 }
76 return false;
77 }
78
79 boolean isFoundSoutheast(int n){
80     boolean found = false;
81     int wordlength = strlen(ELMTL(wordsearched,n));
82     int counterwordlength = wordlength;
83     for (int i = 0; i < ROW(wordboard); i++){
84         for (int j = 0; j < COL(wordboard); j++){
85             int currlength = 0;
86             while ((currlength < wordlength) && (ELMT(wordboard,i+currlength,j+currlength) == ELMTL(wordsearched,n)[currlength])){
87                 currlength++;
88                 counter++;
89             }
90             if (currlength == wordlength){
91                 for (int y = 0; y < ROW(wordboard); y++){
92                     for (int z = 0; z < COL(wordboard); z++){
93                         if (y >= i && y <= i + wordlength - counterwordlength && z >= j && z <= j + wordlength - counterwordlength && counterwordlength > 0){
94                             printf("%c ",(ELMT(wordboard, y, z)));
95                             counterwordlength--;
96                         }
97                         else {
98                             printf(" ",(ELMT(wordboard, y, z)));
99                         }
100                     }
101                     printf("\n");
102                 }
103                 printf("\n");
104                 return true;
105             }
106         }
107     }
108     return false;
109 }
110
111 boolean isFoundSouth(int n){
112     boolean found = false;
113     int wordlength = strlen(ELMTL(wordsearched,n));
114     for (int i = 0; i < ROW(wordboard); i++){
115         for (int j = 0; j < COL(wordboard); j++){
116             int currlength = 0;
117             while ((currlength < wordlength) && (ELMT(wordboard,i+currlength,j) == ELMTL(wordsearched,n)[currlength])){
118                 currlength++;
119                 counter++;
120             }

```

```

121         if (currlength--wordlength){
122             for (int y = 0; y < ROW(wordboard); y++){
123                 for (int z = 0; z < COL(wordboard); z++){
124                     if (y >= i && y <= i + wordlength - 1 && z == j){
125                         printf("%c ",(ELMT(wordboard, y, z)));
126                     }
127                     else {
128                         printf(" ",(ELMT(wordboard, y, z)));
129                     }
130                 }
131                 printf("\n");
132             }
133             printf("\n");
134             return true;
135         }
136     }
137 }
138 return false;
139 }
140
141 boolean isFoundSouthwest(int n){
142     boolean found = false;
143     int wordlength = strlen(ELMTL(wordsearched,n));
144     int counterwordlength = wordlength;
145     for (int i = 0; i < ROW(wordboard); i++){
146         for (int j = 0; j < COL(wordboard); j++){
147             int currlength = 0;
148             while ((currlength < wordlength) && (ELMT(wordboard,i+currlength,j-currlength) == ELMTL(wordsearched,n)[currlength])){
149                 counter++;
150                 currlength++;
151             }
152             if (currlength == wordlength){
153                 for (int y = 0; y < ROW(wordboard); y++){
154                     for (int z = 0; z < COL(wordboard); z++){
155                         if (y >= i && y == i + wordlength - counterwordlength && z <= j && z == j - wordlength + counterwordlength && counterwordlength > 0){
156                             printf("%c ",(ELMT(wordboard, y, z)));
157                             counterwordlength--;
158                         }
159                         else {
160                             printf(" ",(ELMT(wordboard, y, z)));
161                         }
162                     }
163                     printf("\n");
164                 }
165                 printf("\n");
166                 return true;
167             }
168         }
169     }
170     return false;
171 }
172
173 boolean isFoundWest(int n){
174     /* n adalah urutan kata ke berapa yang ingin di cari */
175     boolean found = false;
176     int wordlength = strlen(ELMTL(wordsearched,n));
177     for (int i = 0; i < ROW(wordboard); i++){
178         for (int j = 0; j < COL(wordboard); j++){
179             int currlength = 0;
180             while ((currlength < wordlength) && (ELMT(wordboard,i,j-currlength) == ELMTL(wordsearched,n)[currlength])){

```

```

181         counter++;
182         currlength++;
183     }
184     if (currlength==wordlength){
185         for (int y = 0; y < ROW(wordboard); y++){
186             for (int z = 0; z<COL(wordboard); z++){
187                 if (y == i && j - wordlength + 1 <= z && z <= j){
188                     printf("%c ",ELMT(wordboard, y, z));
189                 }
190                 else {
191                     printf(" ",(ELMT(wordboard, y, z)));
192                 }
193             }
194             printf("\n");
195         }
196         printf("\n");
197         return true;
198     }
199 }
200 }
201 return false;
202 }
203
204 boolean isFoundNorthwest(int n){
205     boolean found = false;
206     int wordlength = strlen(ELMTL(wordsearched,n));
207     int counterwordlength = wordlength;
208     for (int i = 0; i < ROW(wordboard); i++){
209         for (int j = 0; j< COL(wordboard); j++){
210             int currlength = 0;
211             while ((currlength<wordlength)&&(ELMT(wordboard,i-currlength,j-currlength)==ELMTL(wordsearched,n)[currlength])){
212                 counter++;
213                 currlength++;
214             }
215             if (currlength==wordlength){
216                 for (int y = 0; y <ROW(wordboard); y++){
217                     for (int z = 0; z<COL(wordboard); z++){
218                         if (y >= i - wordlength + 1 && y == i + 1 - counterwordlength && z >= j - wordlength + 1 && z == j - counterwordlength + 1 && counterwordlength > 0){
219                             printf("%c ",(ELMT(wordboard, y, z)));
220                             counterwordlength--;
221                         }
222                         else {
223                             printf(" ",(ELMT(wordboard, y, z)));
224                         }
225                     }
226                     printf("\n");
227                 }
228                 printf("\n");
229                 return true;
230             }
231         }
232     }
233     return false;
234 }
235
236 boolean isFoundNorth(int n){
237     boolean found = false;
238     int wordlength = strlen(ELMTL(wordsearched,n));
239     for (int i = 0; i < ROW(wordboard); i++){
240         for (int j = 0; j< COL(wordboard); j++){

```

```

241     int currlength = 0;
242     while ((currlength < wordlength) && (ELMT(wordboard, i - currlength, j) == ELMTL(wordsearched, n)[currlength])){
243         counter++;
244         currlength++;
245     }
246     if (currlength == wordlength){
247         for (int y = 0; y < ROW(wordboard); y++){
248             for (int z = 0; z < COL(wordboard); z++){
249                 if (i - wordlength + 1 <= y && y <= i && z == j){
250                     printf("%c ", (ELMT(wordboard, y, z)));
251                 }
252                 else {
253                     printf(" ", (ELMT(wordboard, y, z)));
254                 }
255             }
256             printf("\n");
257         }
258         printf("\n");
259         return true;
260     }
261 }
262 }
263 return false;
264 }
265
266 boolean isFoundNortheast(int n){
267     boolean found = false;
268     int wordlength = strlen(ELMTL(wordsearched, n));
269     int counterwordlength = wordlength;
270     for (int i = 0; i < ROW(wordboard); i++){
271         for (int j = 0; j < COL(wordboard); j++){
272             int currlength = 0;
273             while ((currlength < wordlength) && (ELMT(wordboard, i - currlength, j + currlength) == ELMTL(wordsearched, n)[currlength])){
274                 counter++;
275                 currlength++;
276             }
277             if (currlength == wordlength){
278                 for (int y = 0; y < ROW(wordboard); y++){
279                     for (int z = 0; z < COL(wordboard); z++){
280                         if (y >= i - wordlength + 1 && y == i + 1 - counterwordlength && z <= j + wordlength - 1 && z == j - 1 + counterwordlength && counterwordlength > 0){
281                             printf("%c ", (ELMT(wordboard, y, z)));
282                             counterwordlength--;
283                         }
284                         else {
285                             printf(" ", (ELMT(wordboard, y, z)));
286                         }
287                     }
288                     printf("\n");
289                 }
290                 printf("\n");
291                 return true;
292             }
293         }
294     }
295     return false;
296 }
297
298 int main(){
299     readFile();
300     printf("\n");
301     printf("----- W O R D   P U Z Z L E ----- \n");

```



```

297 int main(){
298     readFile();
299     printf("\n");
300     printf("----- WORD PUZZLE ----- \n");
301     for (int i = 0; i < ROW(wordboard); i++){
302         for (int j = 0; j < COL(wordboard); j++){
303             printf("%c ", (ELMT(wordboard, i, j)));
304         }
305         printf("\n");
306     }
307     printf("\n----- KEY WORD ----- \n");
308     for (int i = 0; i <= NEFF(wordsearched); i++){
309         printf("%s\n", ELMTL(wordsearched, i));
310     }
311     clock_t initiatepuzzle = clock();
312     for (int n = 0; n < NEFF(wordsearched); n++){
313         isFoundEast(n);
314         isFoundSoutheast(n);
315         isFoundSouth(n);
316         isFoundSouthwest(n);
317         isFoundWest(n);
318         isFoundNorthwest(n);
319         isFoundNorth(n);
320         isFoundNortheast(n);
321     }
322     clock_t terminatepuzzle = clock();
323     double duration = ((double) (terminatepuzzle - initiatepuzzle))/(1000);
324
325     printf("Durasi eksekusi program adalah sebesar : %.3f sekon\n", (float)duration);
326     printf("Jumlah perbandingan huruf adalah sebanyak : %d kali", counter);
327 }

```

BAB III

INPUT DAN OUTPUT PROGRAM

Small (16x16) :

1.

```

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

----- KEY WORD -----
BULBSALIR
CHARMANDER
SQUIRTLE
MENTHO

```

BULBASAU

CHARMANDER

SQUIRTLE

[illegible]

2.

```

----- WORD PUZZLE -----
UFHDRNPVFNJBGHHHR
KAYONWIWMXCLUGIAK
ORYTESSWNOBAHVQID
QBVLQYNCYNDAAQUIL
EYHCDDTWCGRGJDVPVO
RQZWMYUOISGNGETAFT
VSPJXBMDFWGQGALNX
BNOXDRRAIZSCXXSP
OHLMWKPKXCLEARDFQ
WKHHTTHPUAECLETD
HPIMGWHMBKMHCTD
ZKCHKQZITAHJNVCA
AHAPQKQZISHBSJCQI
BMYEJRXXGGBTQZJYM
HANXZUKUJVKIUKM
AHAPQKQZISHBSJCQI

----- KEY WORD -----
CHIXORITA
CYNDAQUIL
LUGIA
TOTODILE

```

CHIKORITA

CYNDAQUIL

LUGIA

```
. . . . .  
. . . . .  
. . T . . . . .  
. O . . . . .  
. T . . . . .  
. O . . . . .  
. D . . . . .  
. I L . . . . .  
. L . . . . .  
. E . . . . .  
. . . . .  
. . . . .  
. . . . .  
. . . . .  
. . . . .
```

Durasi eksekusi program adalah sebesar : 0.144 sekon
Jumlah perbandingan huruf adalah sebanyak : 334 kali

3.

```

----- WORD PUZZLE -----
NWMWDYZFZFSHVYNNRB
LGNUWOHARSQKBLNT
FSIQQDGLLOLPHTLTO
WZZPWKUXCBUNPZDR
OMTMCWIFFYJOIVZKC
GVFSWETPFYJBYKGH
TRECKKOWYNBTZPXI
CQLMVKQRAQYUAZAC
ATZHATEQQILOFKRO
YVDERWNNRDNNAJDB
XSPMNIFFNYFSRQTX
YFGFXLDKFJNLXXYC
AHAPQQZISHSBSJCQI
BMYEJRXXGGBTQZJYM
HANXZUKUJVJKUIM
AHAPQQZISHSBSJCQI

----- KEY WORD -----
MUDKIP
RAYQUAZA
TORCHIC
TRECCKO

```

M
 U
 D
 K
 I
 P
 RAYQUAZA
 TORCHIC

```
TRECKKO.
```

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

1.

```

----- WORD PUZZLE -----
JWIZCUIWIGVBPAKWNNBMOCF
GFACCGIWRTRUTOBNHZWVQF
AXOGIXSQIJPDPUGAQDEWZV
EERTNRIEXWVXEKLKAVGCEJ
BWEGITFCWUKUHVVYUDNVEHG
PPIEBBOEXECLRLXLDPAJPBA
VAPTYGIRATINAYDALVXIXN
ZGUBSTMZGRHXYVKZXDQYLV
GMYREAMINCFSYDWTNYTWRZT
QWOZQMIVTWFBJBUGAEADOD
GGSFXMPULESBMCRCLPTRL
IFYWCHIOCHARPQXQORTZSE
ESRTMELWLXMYBDXNREQSWP
GPKLQDCBIENFTQARDYLYM
INAROCLEWBOYQIURWXRDSI
RDFNOXMKNKCNBQAIHHJDMW
JWIZRUIWIGVBPAKWNNBMOCF
GFACCEIWRTRUTOBNHZWVQF
BWEGITFCWUKUHVVYUDNVEHG
GMYRAMINCFSYDWTNYTWRZY
WSRTEMLWIXMYBDXNSBQSWP

----- KEY WORD -----
TORTERRA
INFERRAPE
EMPOLEON
GIRATINA

```

TOGETHER WE CAN SAVE THE WORLD

```

E
M
P
O
L
E
O
N
.

GIRATINA.

Durasi eksekusi program adalah sebesar : 0.232 sekon
Jumlah perbandingan huruf adalah sebanyak : 767 kali

```

2.

```

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

S N I V Y
T E P I G
O S H A W O T T
K Y U R E M

```

YVINS.

TEPIG.

```

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.

```

----- WORD PUZZLE -----
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 A 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 K C B O A O K Q B S J E R U G S O
S I T E B F C 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 X
K V Z W I U A N X S Y S Q S 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 F E 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 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

----- KEY WORD -----

CHESPIN
FENNEKIN
FROAKIE
XERNEAS

```

CHESPIN
NIKENEF

[illegible]

Large (32x30) :
1.

----- WORD PUZZLE -----

ZBEYDAJNYSPEAAEFNHEEIFSEBSNCOR
LVZBMJCZLRSQCLNANOSSTVFCUXNNRC
RIGILKENTUMSRNMBYRYUREQPERPRYK
KTZAGSYTECCFUIEIEYACHUODMRFNCI
XUQTHIPGUXZAXLHFAHBMONAHOYCD
WLQJYQLTISDYIARFLPFPALZQVPITPQ
HGXYGETRFVGVQVMGADELBCBXPFGVYST
KPAETQTOKQFANRMFNGTRTQNDQKMCLA
SJOENAVBZCDQTOHTKYOLCWEZGAUJAP
EZBCLBNFHQMNFLWRORWSWIAJBNYEV
JHDAGNYFOGTHBKTBRVIOUDRBEXLSJ
GXEVCUJJTXQZJAEMARHCGFUZQLUZL
BBSRWUTEBUOORATDFEXTMYDDSAVHDK
SERATNLELNATHMKIGYHAYEKJLNYLZW
KYSGUUMJCCSXDYHGMSONRSONPPIYS
MVEQBWFFUGTVLEGIROFWENUREXVUXS
UMRLPHYRBAZECMOCRUSIGEQQKUMTVH
VLPQQDFAUULGJYTRAMIAUFANXLCPCGB
SAMYDQXHAKNAPTQTNCTULEYYSLZNWU
UURTQFXLKKXHDZKFUXILUDQBJOPERM
ICQCPWDSTHADARDBQNYPSAQTVPTJUN
RSRCTEQPOVZNGGJSQTRHSXLALUAHSG
IYBGBUAFHWHATGAJMRGRPEXTHVVSNL
SQSAZARZVPMNQSMRSLNXPWDPADPXHW
CHRJLIJUTYQJQLSRMFPAVPLDRILUZR
WAWNPLRDRNCCCKRMEODDKKKOFTRJBX
NGAUWFYUWSLCSWDBAHCPREJVPSSLBAR
UJLDMUVXSVAQIZPAAZWKJYQXHADES
RDHUKRTSFUBMYRQRRKDZYRGUMMUULG
KMFHZFMZQFMZFWAHGEPDPTOIOHAGG
MVEQBWFFUGTVLEGIROFWENUREXRUXS
UMRLPHYRBAZECMOCRUSIGEQQKUMTVH

----- KEY WORD -----
TERRORBLADE
URSA
SLARK
LUNA

E
D
A
L
B
R
O
R
E
T

U
R
S
A

S
L
A
R
K

A
N
U
L

Durasi eksekusi program adalah sebesar : 0.446 sekon
Jumlah perbandingan huruf adalah sebanyak : 1516 kali

2.

```

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

----- KEY WORD -----
ALBANY
DOVER
INDIANAPOLIS
RICHMOND

```

A 20x20 dot grid with points A, B, L, N, and Y marked. Point A is at (15, 10), B is at (12, 12), L is at (13, 13), N is at (11, 14), and Y is at (10, 15).

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

```

----- WORD PUZZLE -----
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 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 M 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 I L K N N O J V P Y R Y N I B B J X T O P Q R T P O
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 D 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 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 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 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 A 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 R
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 X
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

----- KEY WORD -----
AUSTRALIA
ITALIA
INDONESIA
ESPANYOL

```

AIRLARTSUSA

A 20x20 grid of dots. The word "LITANI" is spelled out in larger dots, centered in the grid. The letters are arranged as follows: L (row 10, col 10), I (row 10, col 12), T (row 10, col 14), A (row 10, col 16), N (row 10, col 18), I (row 10, col 20). The word is written in a simple, blocky font.

A 20x20 grid of small dots. The word "ANSION" is spelled out by larger dots. The letters are arranged as follows: 'A' at row 10, column 5; 'N' at row 10, column 7; 'S' at row 11, column 6; 'I' at row 11, column 8; 'O' at row 12, column 7; 'N' at row 12, column 9; 'E' at row 13, column 8; 'D' at row 13, column 10; 'I' at row 14, column 9.

E S P A N Y O L

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

BAB IV
LINK

<https://github.com/danielsalim/wordpuzzle>