

**LAPORAN TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA
PENYELESAIAN WORD SEARCH PUZZLE MENGGUNAKAN
ALGORITMA BRUTE FORCE**



Oleh :

Ilham Prasetyo Wibowo 13520013

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

2022

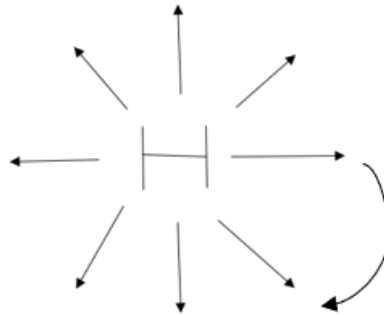
A. DESKRIPSI ALGORITMA

1. PEMROSESAN FILE

Sebelum diselesaikan menggunakan algoritma brute force, masukan dari file diproses terlebih dahulu. Yang pertama, program akan mencari dimensi matriks. Kemudian menghitung jumlah kata yang akan dicari. Menggunakan dimensi matriks yang sudah dicari, program mengalokasikan sebuah matriks 2 dimensi untuk menyimpan data masukan. Sembari membaca kata satu per satu, program mencari kata tersebut sekaligus dari dalam matriks dan mencetak hasilnya ke layar.

2. ALGORITMA PENYELESAIAN

Dengan setiap kata yang dibaca dari file masukan, program akan melakukan proses pencarian di dalam matriks. Seperti proses pencarian matriks biasanya, dimulai dari kiri atas, sampai kanan bawah. Setiap pencarian akan dilakukan pencocokan kata mulai dari arah kanan, kemudian berputar searah jarum jam. Ilustrasi pencariannya adalah sebagai berikut.



Pemutaran berpindah arah jika terdapat huruf yang tidak sama dengan target, atau pencarian keluar batas dari matriks. Jika pada suatu arah pencarian tidak berhenti sampai akhir huruf sesuai target, maka pencarian berhenti dan menampilkan hasil ke layar.

3. MENAMPILKAN HASIL

Setiap huruf yang ditemukan dari matriks, akan disimpan posisinya dalam sebuah list. Kemudian akan ditampilkan huruf huruf tersebut ke layar.

B. SOURCE CODE

```

int get_rows_columns(char* filename, int *rows, int *cols,int *word_ctr){
    FILE *file;
    file = fopen (filename, "r");
    char buff;
    char buffore;
    //mencari rows and columns dari matrix
    while (1)
    {
        fscanf(file, "%c", &buff);
        if (buff == 10 && buffore == 10) {
            break;
        }
        else if (buff == 10) {
            *rows += 1;
        }
        else if (buff != ' ' && *rows == 0) {
            *cols += 1;
        }
        buffore = buff;
    }

    //mencari jumlah kata yang akan dicari
    while(1) {
        if (fscanf(file, "%c", &buff) != EOF) {
            if (buff == 10) {
                *word_ctr += 1;
            }
        }
        else {
            *word_ctr += 1;
            break;
        }
    }
    fclose (file);
}

```

Algoritma untuk mencari dimensi matriks dan banyak kata

```
int is_point_in(int a, int b, int** mat, int length){  
    int exist = 0;  
    for (int i = 0; i < length; i++) {  
        if (mat[i][0] == a && mat[i][1] == b) {  
            exist = 1;  
            break;  
        }  
    }  
  
    if (exist) {  
        return 1;  
    }  
    else {  
        return 0;  
    }  
}
```

Algoritma untuk mengecek apakah sebuah titik terdapat dalam sebuah list titik

```

int print_word(char** grid, int dir,int row,int col, int word_length, int R, int C){
    int x[8] = {0, 1, 1, 1, 0,-1,-1,-1}; //geser baris
    int y[8] = {1, 1, 0,-1,-1,-1, 0, 1}; //pergeseran kolom
    int i,j;

    //inisialisasi matriks untuk menyimpan posisi word pada grid
    int** position=malloc(word_length*sizeof(int*));
    for(i=0;i<word_length;++i) {
        position[i]=malloc(2*sizeof(int));
    }

    //mengisi matriks dengan posisi word
    int rptr = row;
    int cptr = col;
    for (i = 0;i < word_length;i++) {
        position[i][0] = rptr;
        position[i][1] = cptr;
        rptr += x[dir];
        cptr += y[dir];
    }

    //mencetak kata sesuai posisinya pada grid
    for (i = 0; i < R;i++) {
        for (j = 0; j < C; j++) {
            if (is_point_in(i,j,position,word_length)) {
                printf("%c ", grid[i][j]);
            }
            else {
                printf("- ");
            }
        }
        printf("\n");
    }

    free(position);
}

```

Algoritma untuk menampilkan kata yang ditemukan ke layar

```

void find_word(char** grid, int row, int col, int *ctr, char* word, int R, int C)
{
    int k;
    int rowdir;
    int coldir;
    int x[8] = {0, 1, 1, 1, 0, -1, -1, -1}; //geser baris
    int y[8] = {1, 1, 0, -1, -1, -1, 0, 1}; //pergeseran kolom
    if (grid[row][col] != word[0]) {
        *ctr += 1;
        return;
    }
    int len = strlen(word);
    for (int dir = 0; dir < 8; dir++) {
        rowdir = row + x[dir];
        coldir = col + y[dir];
        for (k = 1; k < len; k++) {
            *ctr += 1;
            if (rowdir >= R || rowdir < 0 || coldir >= C || coldir < 0) { //keluar batas
                break;
            }
            else if (grid[rowdir][coldir] != word[k]) { //tidak sama
                break;
            }
            else {
                rowdir += x[dir];
                coldir += y[dir];
            }
        }
        //kata sudah ditemukan
        if (k == len) {
            print_word(grid, dir, row, col, len, R, C);
            return;
        }
    }
    return;
}

```

Algoritma pencarian kata dalam matriks

```

int search_word_grid(char** grid, char* word, int R, int C)
{
    int temp_ctr = 0;
    for (int row = 0; row < R; row++) {
        for (int col = 0; col < C; col++) {
            find_word(grid, row, col, &temp_ctr, word, R, C);
        }
    }
    return temp_ctr;
}

```

Algoritma iterasi dalam matriks untuk setiap kata yang dibaca

```

int main(void)
{
    int rows = 0;
    int cols = 0;
    int word_ctr = 0;
    clock_t start_time, end_time;
    double elapsed;
    int counter = 0;

    char filenm[100];
    printf("Masukkan nama file yang ingin digunakan (contoh : ../test/test.txt) : ");
    scanf("%s", &filenm);

    //cari banyak kolom, baris, dan banyak kata yang akan dicari
    get_rows_columns(filenm,&rows,&cols,&word_ctr);

    //alokasi matriks untuk word grid
    char** mat=malloc(rows*sizeof(char*));
    for(int i=0;i<rows;++i) {
        mat[i]=malloc(cols*sizeof(char));
    }

    //baca matriks dan masukkan hasil kedalam mat
    char* word[20];
    FILE *f;
    f = fopen (filenm, "r");

    //mengubah matriks dalam file menjadi variable matriks
    for(int i = 0; i < rows; ++i)
    {
        for(int j = 0; j < cols; ++j)
        {
            fscanf(f, "%c ", &mat[i][j]);
        }
        printf("\n");
        printf("SEARCHING ALGORITHM STARTING ... ");
        printf("\n");
        start_time = clock();

        //memulai algoritma pencarian
        for (int i = 0; i < word_ctr; i++) {
            fscanf(f,"%s", &word);
            printf("\n%s\n\n", word);
            counter += search_word_grid(mat, word, rows, cols);
        }
        end_time = clock();
        elapsed = (((double)(end_time - start_time))/ CLOCKS_PER_SEC);
        fclose (f);

        printf("\n");
        printf("SEARCH COMPLETED,\nTIME TAKEN : %f seconds,\n%d TOTAL COMPARISON", elapsed,counter);
        printf("\n");
        // for(int i = 0; i < rows; ++i)
        // {
        //     for(int j = 0; j < cols; ++j) {
        //         printf("%c ", mat[i][j]);
        //     }
        //     printf("\n");
        // }

        return 0;
    }
}

```

Main program

C. HASIL PENGUJIAN

1. Kasus Kecil

[illegible]

CANCEL

VELVET

RIVER

- - - - -
- - - - -
- - - - -
- - - - - R E V I R

GOVERNOR

R
O
N
R
E
V
O
G

DIPLOMAT

FIELD

SENTENCE

SENTENCE

```
SEARCH COMPLETED,  
TIME TAKEN : 0.437000 seconds,  
2603 TOTAL COMPARISON  
PS C:\MyData\Sedang Kuliah\Semester 4\Strategi Algoritma\Tugas-Kecil-1-STIMA\src>
```

X
I
S
P
F
C
N
G
Y
C
R
3
K
U
A
F
Q
I
Z
P
N
O
R
O
P
U
L
T
E
G
A
J
A
E
A
N
T
E
Q
S
Q
N
O
T
R
M
E
B
K
N
T
G
L
E
V
E
L
J
U
T
E
G
K
U
E
R
F
W
I
Y
X
P
Q
A
M
O
Z
I
O
M
A
A
Z
J
D
S
J
S
G
L
B
R
C
P
L
Y
S
Z
H
B
B
E
Q
D
C
C
D
E
T
L
E
P
H
O
N
E
O
N
Z
E
R
K
W
I
Y
X
Y
Q
V
I
C
Z
I
Z
O
V
G
X
A
C
A
F
R
O
D
K
Z
P
E
H
E
E
G
T
M
E
E
E
U
S
A
Y
M
V
S
J
F
J
U
H
V
Z
I
I
F
L
R
Q
G
A
C
H
C
W
B
V
F
B
E
H
L
C
C
S
R
C
Y
N
N
K
T
C
K
V
A
Q
T
A
P
D
I
C
Q
Z
M
N
V
B
T
S
H
R
Q
G
S
T
P
A
V
M
T
C
G
N
P
Y
K
G
I

SHOWER
COUNTRY
TELEPHONE
CONFIDENCE
DEFEAT
CONTEMPT
GALLERY
SAMPLE
LARGE

PS C:\MyData\Sedang Kuliah\Semester 4\Strategi Algoritma\Tugas-Kecil-1-STIMA\src>

DETECTOR
RECORDING
LIFESTYLE
PAUSE
WAGE
BUTTOCKS
HEAL
GLUE
ACCESS
MUSHROOM

R
 O
 T
 C
 E
 T
 E
 D

RECORDING

L
 I
 F
 E
 S
 T
 Y
 L
 E

P
A
U
S
E

W
A
G
E

SKCOTTUB-----

L
A
E
H

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525



MUSHROOM

```
SEARCH COMPLETED,  
TIME TAKEN : 0.527000 seconds,  
2842 TOTAL COMPARISON  
PS C:\MyData\Sedang Kuliah\Sem
```

Hasil Pengujian	
-----------------	--

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

ABASH
ASSERTING
ASTHMA
CELSIUS
DECOMPRESS
DESPISING
DRAUGHT
GLANCE
LAUGH
LEES
LEFTY
MIGHT
NEWSFLASH
OSCAR
PASSPORT
SLIME
SMUG
SOURCELESS
THESAURUS
WARPED

SEARCHING ALGORITHM STARTING ...

ABASH

ASSERTING

A S S E R T I N G

ASTHMA

A
M
H
T
S
A

CELSIUS

DECOMPRESS

DESPISING

G N I S I P S E D

DRAUGHT

GLANCE

LAUGH

LEES

LEFTY

MIGHT

THESAURUS

WARPED

```
SEARCH COMPLETED,  
TIME TAKEN : 1.724000 seconds,  
11892 TOTAL COMPARISON  
PS C:\MyData\Sedang Kuliah\Sem
```

NEWSFLASH

OSCAR

PASSPORT

SLIME

SMUG

SOURCELESS

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

MIRACLE
THUMB
HEAD
WASTE
EMBOX
DELIVERY
CONTAIN
STRAW
DECREASE
REVOKE

SEARCHING ALGORITHM STARTING ...

MIRACLE

ELCARIM

THUMB

HEAD

DAEH

WASTE

EMBOX

DELIVERY

CONTAIN

STRAW

DECREASE

REVOKE

SEARCH COMPLETED,
TIME TAKEN : 0.855000 seconds,
5217 TOTAL COMPARISON

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

APPLIED
TRAIT
RICE
DECORATIVE
SNIFF
PUMP
HARASS
BORROW
CEMETERY
INSURE
HURT
RELIABLE
HESITATE

SEARCHING ALGORITHM STARTING ...

APPLIED

TRAIT

RICE

- R -
- I -
- C -
- E -

DECORATIVE

- - - - - D -
- - - - - E -
- - - - - C -
- - - - - O -
- - - - - R -
- - - - - A -
- - - - - T -
- - - - - I -
- - - - - V -
- - - - - E -
- - - - - S -
- - - - - S -
- - - - - A -
- - - - - R -
- - - - - A -
- - - - - H -

SNIFF

- F F I N S -

PUMP

- P -
- M -
- U -
- P -

HARASS

BORROW

CEMETERY

- C E M E T E R Y

3. Kasus Besar
Kasus yang diuji :
a.

a.

OFJFYUWVSRFOCJSWQLISOEZWCQZQW
FEPFDBMUGALLONLXJBVQIUFCUNDCPZ
PATRWXGHIRREHDZFFJZDWN CZWVMXRIM
OTORIQOMSEIODEILPPAEGBJDALVKMF
XUOCINZJBWOXIAZUSAVNCHAFMVHCRG
VRQXISTKFFJTSHPMHQAPHHHVIOSIYXT
XEBAOUZQAWLQTDUIRXVOLVBSDFQDA
QJXJRSYCDTHAUCUKCGQPVSWRWYRAT
USQULHTVZMGLGTJKKWIVRTWOC LPBFM
SNHLPTKNCULVOCXTFYVRARFVSBEZVR
FAJPUERPEOZMCKLIDBTCWYCRPSPZXM
OEQR LADBUWYZXLDDDJLATCTAEUHQT
YQHTUBLTLILZDCTIYEEIUSXRFIITTNW
AWJCRWHYQWTZLKS IJQFLXHLUWNGKIG
FAMFOXZDAEXCLJQQNC DMSKQBENQHRJ
YWARVGGVLSZFASFBIUMKZJGPRCSATT
AZPBIEBNOIAXTDNMWOBTFYDTZZOODA
JLWIDIDENVTGSTCEPSUSBSNQHSHPOE
ZZXNFTCAOREVRESBOY MZHD CSBWFXTW
VSCBEPUUAJZIGMLH WLPUNTNP CDXYIH
ZLZIHJR VIRDLIERAUXHPQNK EOIOJPG
PBMZFIAIISXSMGLEHIMFKBF OFWHPD
C PNACTRHKKFJLSDJLSD FUMZPCOF OJEJ
NCKEIUJODCAYMGVQCRRWVOECXHORSU
RHZEIDY CQNKLD SHXSQMBVOTRKKBDYP
HWT OXUEQKCITHOZTTDEEZ XEOBKUDDA
NYXNHK OIMHACZVTEMUEAVMHMNC LFBRR
OXEMJHNMLEVDAJIDMBHVXGLENKEEON
ZDMKJITBOSTHXFQEHKWINREAVMJVZR
BIDZQDZUIPEGJECY MVEVKSHINEFFNL

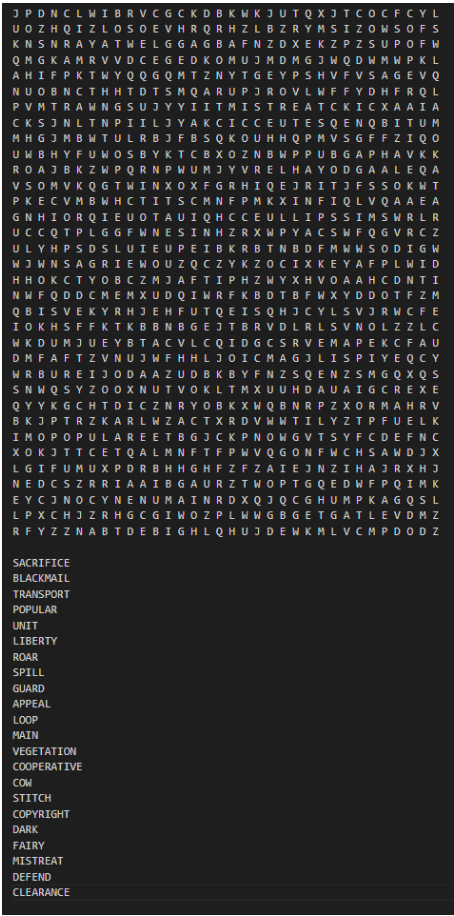
SIGHT
CHOP
OBSTACLE
APPLIED
HOOK
HELMET
SOAR
RAT
PRINT
BIN
FAVOURITE
OBSERVER
OFFEND
STALL
AVENUE
GALLON
FREIGHT
FEATURE
VARIETY
SUSPECT

b.

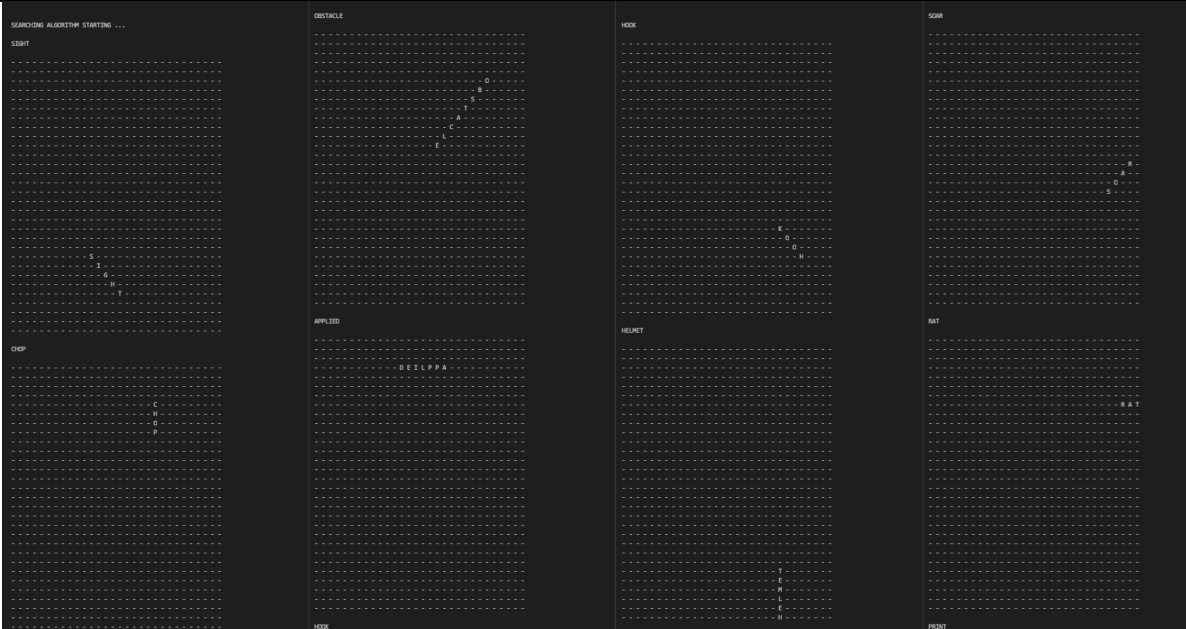
RS PKRMSTHGVRPDNOAIZWIVPEXPZWHSSQY
GKLETHVFRTEBITMYQHIOXNPGOJSKJTW R
MXAECEYRFEPSPVASGALVNHMBODJLJTUME
XTCHKLCKQWTEHRAEZKCR O EZYVJIPBJIZ
NRECSFYPDUJEJJPDPNPJTC LBVRNSKQJUD
PORTERRVRBMHYBAXREORVRNRSES IPTJN
GYGUJSXBUIHCHREUMGDYUGHFZTISERYPL
PQIRIEADDSJMIEBSRVBSVNDZFLFMR LKW
ASUPBNBXXJXVDIXAKFFCWOPT RCLXWPNX
NTSZCNXZOHMYCQPMSSYJBRAKXYVNOAGPC
EJWELZLQDMOYUHNWKM TAYODTCJASBUJT
YMRUAAAKBEMHGZJMPMPURD VHLBBAMNVGG
EYHVDWUPTAALTUJUZHKVYLKBJBPGEAEPKP
AOQNABHYQPDSHSRBDINFVUPETUGCYVKJ
EZATHODAEFOOOWMEKRVP SZHTDQJETGWR
PCOANBZZFYPRD WQGOEUGOLAIDRBYGFDN
SAPKDMEETOTCOUEDJAPVFRBOPITUEOMTS
IJDZCZGNNGIMANEBARZTZXXVIMLFMAHJL
YROTSPSPPEOPDZWRIPCVP EIGLSAMVTHK
RUUMRZECXMMRZRZBMC PHQJHHDLBSVGH
PPWDSYQPT SJUYSA BRFTVOKURCPPQRWQW
YRAHIRPPSCAPXN JNM TTFD TWSQEMHEZUE
ETPPRV IURGAGITKRCHCNGHRKKZTUSPEY
DVQYUVFBYBSRW FETKCFMHZ ZNV SOKTBOV
ZEKZFBQXXMRFTRPMDY P YTLRLUZXLPCXS
ELGB OUMZCBBWUTPUMEDBVJCB RZAGPOSX
QBG YTMWJVJTBFFAHFPEATPMLYBPNMPXF
YAKOWKEBZGYNKSTNORTHBJJPYUQKYEUC
ITTFVFQZEJ IUVNOLF XGEVTICLLKUCMUM
GALDANZBTANIISD KKG IATKLVA AJLBDOU
LMWUMIRXHUORCHLWACGWJZXPMTLSAQHQ
LWZGBPKCKDVENDRLPCITUMBKFGJCCOIIY

RUN
PLACE
PHOTOGRAPH
STORY
ATTRACT
CHEEK
CHAIN
PORTER
ZONE
ADOPTION
TOTAL
DIALOGUE
MURDER
TABLE
SCANDAL
NORTH
BEAT
EGO
DISTURBANCE
PRIMARY

c.



a



<div>PRINT</div> <div>P R I T</div>	<div>FAVORITE</div> <div>F A V O R I T E</div>	<div>OFFEND</div> <div>D E F O</div>	<div>VARIETY</div> <div>V A R I E T Y</div>
<div>BDN</div> <div>B I N</div>	<div>OBSERVER</div> <div>R E V R E S B O</div>	<div>STALL</div> <div>L A T S</div>	<div>SUSPECT</div> <div>T C E P S U S</div>

SEARCH COMPLETED,
TIME TAKEN : 3.350000 seconds,
23321 TOTAL COMPARISON

b

<div>RUN</div> <div>R U N</div>	<div></div> <div>P H O T O G R A P H</div>	<div>STORY</div> <div>Y R O T S</div>	<div></div> <div>C E H C</div>
<div>PLACE</div> <div>P L A C E</div>	<div>STORY</div> <div>Y R O T S</div>	<div>ATTRACT</div> <div>T C A R T A</div>	<div>CHAIN</div> <div>N T</div>

[illegible][illegible]

SACRIFICE

E
C
I
F
I
R
C
A
S

BLACKHILL

L
I
A
H
K
C
A
L
B

TRANSPORT

T
O
R
S
N
A
R
T

POPULAR

UNIT

U
N
I
T

LIBERTY

L
I
B
E
R
T
Y

ROAR

O
A
R

SPILL

L L I P S

SPILL

L L I P S

GUARD

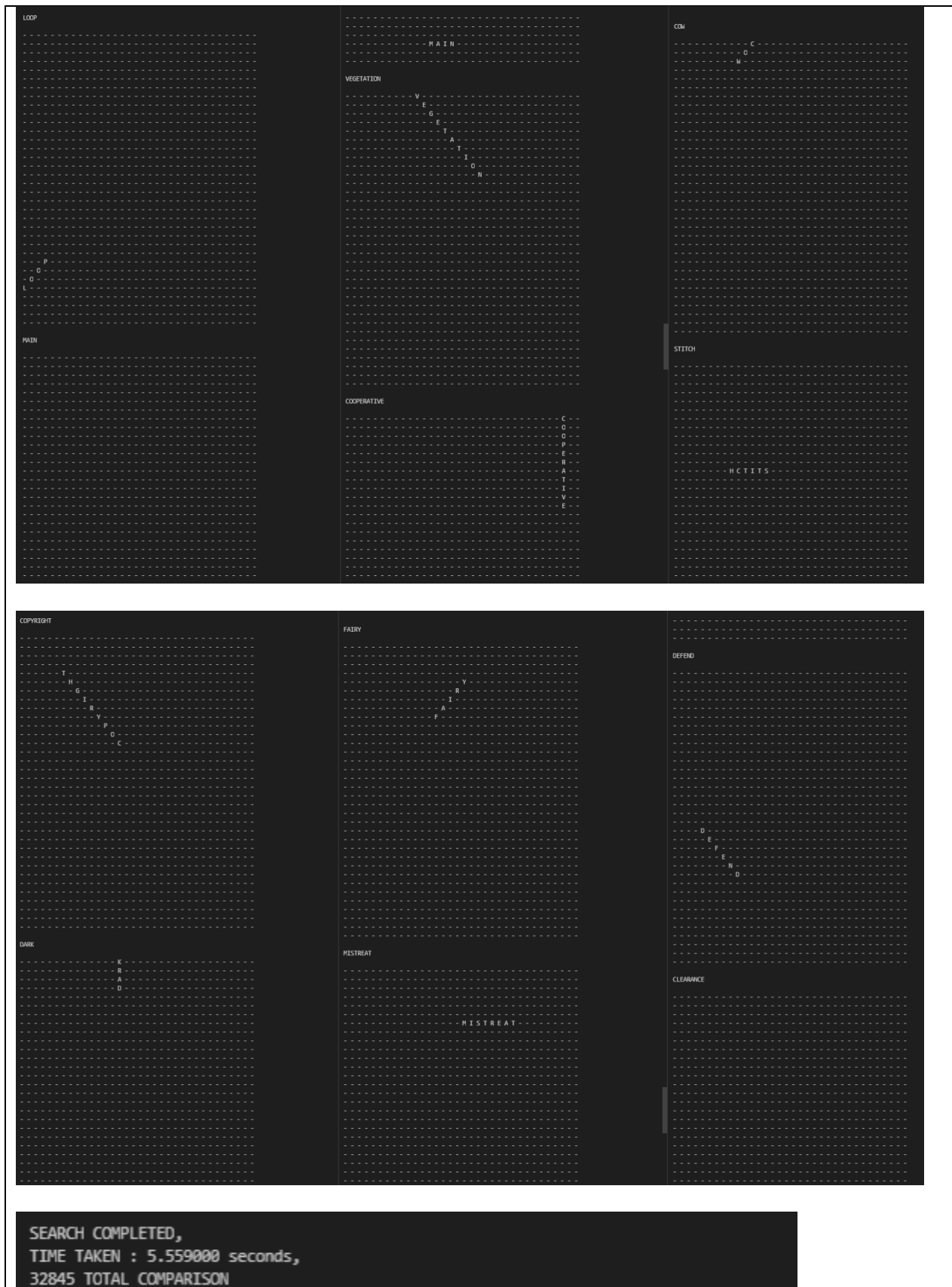
D
R
A
U
G

GUARD

D
R
A
U
G

APPEAL

L
A
E
P
A



D. LINK MENUJU DRIVE

<https://github.com/ilhamwibowo/Tugas-Kecil-1-STIMA>