**LAPORAN**

**TUGAS KECIL 1**

**IF2121 STRATEGI ALGORITMA**

Dosen: Dr. Masayu Leylia Khodra, S.T., M.T.



Oleh:

Maharani Ayu Putri Irawan / 13520019

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2022

**BAB I  
ALGORITMA BRUTE FORCE**

Pada tugas kecil ini, digunakan algoritma *brute force* dengan deskripsi sebagai berikut:

1. Mulai dengan kata pertama yang hendak dicari dalam puzzle.
2. Menelusuri setiap huruf yang terdapat dalam word puzzle untuk mencari huruf yang sama dengan huruf pertama kata yang dicari.
3. Jika ditemukan huruf yang sesuai, lanjut dengan mencocokkan huruf kedua dan seterusnya dengan mencari ke arah Timur, Tenggara, Selatan, Barat Daya, Barat, Barat Laut, Utara, dan terakhir Timur Laut. Apabila huruf kedua ditemukan pada salah satu arah, akan diteruskan dengan mencocokkan huruf ketiga, keempat, dan seterusnya ke arah yang sama. Jika seluruh huruf pada kata tersebut telah ditemukan, pencarian ke arah mata angin setelahnya (yang belum dilalui) maupun huruf dalam word puzzle lainnya tidak perlu dilakukan. Namun, jika kata tersebut tidak ditemukan pada suatu arah, akan diteruskan pencarian ke arah setelahnya. Bila pada seluruh arah mata angin telah dicari namun tidak ditemukan, akan dilanjutkan ke langkah berikutnya.
4. Bila tidak ditemukan huruf yang sesuai, lanjut dengan mencocokkan huruf pertama dengan huruf pada word puzzle selanjutnya.
5. Jika sebuah kata telah ditemukan pada word puzzle tersebut, pencarian dilanjutkan pada langkah kedua terhadap kata selanjutnya.
6. Bila seluruh kata telah ditemukan posisinya dalam word puzzle, pencarian berakhir.
7. Setiap kali satu kata ditemukan posisinya di dalam word puzzle, ditampilkan matriks yang menunjukkan posisi suatu kata di dalam word puzzle.
8. Pada akhir program, ditampilkan waktu eksekusi program dan jumlah perbandingan huruf yang diperlukan.

**BAB II  
KODE SUMBER**

Kode ditulis dalam Bahasa C. Berikut merupakan kode sumber yang terdapat di dalam file main.c. Kode sumber juga dapat diakses melalui Github <https://github.com/rannnayy/Tucil1_13520019> atau Google Drive .

#include <stdio.h>

#include <stdlib.h>

#include <sys/time.h>

#include <dirent.h>

#include <string.h>

#define NEWLINE '\n'

#define NULLCHAR '\0'

#define BLANK ' '

#define MARK\_EOF EOF

#define boolean unsigned char

#define true 1

#define false 0

typedef struct {

char\* contents;

int length;

} Word;

void printAns(int row, int col, char posRowCol[row][col])

{

for (int i = 0; i < row; i++){

for (int j = 0; j < col; j++){

if (j == col-1)

printf("%c", posRowCol[i][j]);

else

printf("%c ", posRowCol[i][j]);

}

printf("\n");

}

}

int incIndexRow(int row, int dir)

{

switch (dir)

{

case 1: // Timur

return row; break;

case 2: // Tenggara

return (row+1); break;

case 3: // Selatan

return (row+1); break;

case 4: // Barat Daya

return (row+1); break;

case 5: // Barat

return row; break;

case 6: // Barat Laut

return (row-1); break;

case 7: // Utara

return (row-1); break;

case 8: // Timur Laut

return (row-1); break;

}

}

int incIndexCol(int col, int dir)

{

switch (dir)

{

case 1: // Timur

return (col+1); break;

case 2: // Tenggara

return (col+1); break;

case 3: // Selatan

return col; break;

case 4: // Barat Daya

return (col-1); break;

case 5: // Barat

return (col-1); break;

case 6: // Barat Laut

return (col-1); break;

case 7: // Utara

return col; break;

case 8: // Timur Laut

return (col+1); break;

};

}

void sequentialMatching(int j, int row, int col, char wordPuzzle[], Word words, boolean \*found, char posRowCol[row][col], int dir, int \*numCompare)

{

int idxRow = j/col; // wordPuzzle index

int idxCol = j%col;

int idxWord = 1; // word content index

while(incIndexCol(idxCol, dir) >= 0 && incIndexCol(idxCol, dir) < col && incIndexRow(idxRow, dir) >= 0 && incIndexRow(idxRow, dir) < row

&& wordPuzzle[(incIndexRow(idxRow, dir)\*col)+incIndexCol(idxCol, dir)] == words.contents[idxWord] && words.length > idxWord && (!(\*found))){

posRowCol[idxRow][idxCol] = words.contents[idxWord-1];

idxRow = incIndexRow(idxRow, dir);

idxCol = incIndexCol(idxCol, dir);

idxWord++;

}

(\*numCompare)++;

if (idxWord == words.length){

\*found = true;

posRowCol[idxRow][idxCol] = words.contents[idxWord-1];

}

else{

for (int a = 0; a < row; a++){

for (int b = 0; b < col; b++){

posRowCol[a][b] = '-';

}

}

}

}

void displayFile()

{

DIR \*dir;

char \*p1, \*p2;

int ret;

struct dirent \*dirs;

dir = opendir("./test");

int cnt = 0;

if(dir){

while((dirs = readdir(dir)) != NULL){

p1 = strtok(dirs->d\_name, ".");

p2 = strtok(NULL, ".");

if(p2 != NULL){

ret = strcmp(p2, "txt");

if(ret == 0){

cnt++;

printf("%d. %s\n", cnt, strcat(strcat(p1, "."), "txt"));

}

}

}

closedir(dir);

}

}

void welcomeMsg()

{

printf(" \_\_ \_\_ \_ \_\_\_\_\_ \_ \_\_\_\_\_ \_ \n");

printf(" \\ \\ / / | | / \_\_\_\_| | | | \_\_ \\ | | \n");

printf(" \\ \\ /\\ / /\_\_ \_ \_\_ \_\_| | | (\_\_\_ \_\_\_ \_\_ \_ \_ \_\_ \_\_\_| |\_\_ | |\_\_) | \_ \_\_\_\_\_\_\_| | \_\_\_ \n");

printf(" \\ \\/ \\/ / \_ \\| '\_\_/ \_` | \\\_\_\_ \\ / \_ \\/ \_` | '\_\_/ \_\_| '\_ \\ | \_\_\_/ | | |\_ /\_ / |/ \_ \\ \n");

printf(" \\ /\\ / (\_) | | | (\_| | \_\_\_\_) | \_\_/ (\_| | | | (\_\_| | | | | | | |\_| |/ / / /| | \_\_/ \n");

printf(" \\/ \\/ \\\_\_\_/|\_| \\\_\_,\_| |\_\_\_\_\_/ \\\_\_\_|\\\_\_,\_|\_| \\\_\_\_|\_| |\_| |\_| \\\_\_,\_/\_\_\_/\_\_\_|\_|\\\_\_\_| \n");

printf("\n");

printf("Maharani Ayu Putri Irawan - 13520019\n");

printf("Tucil1\_13520019\n\n");

}

int main()

{

char filename[50];

welcomeMsg();

printf("File yang tersedia:\n");

displayFile();

printf("\nMasukkan nama file (contoh: ./test/cobainput.txt) : ");

scanf("%[^\n]s", filename);

FILE \*fp = fopen(filename, "r");

if (fp == NULL){

printf("File tidak tersedia!");

}

else{

char\* wordPuzzle;

char currChar;

int charCtr = 0;

int row = 0;

int col = -1;

boolean beforeNL = false;

boolean emptyLine = false;

boolean memory = false; // true if reallocation unsuccessfull

int numWord = 0;

int lenWord = 0;

wordPuzzle = (char\*)malloc(charCtr \* sizeof(char));

if (wordPuzzle != NULL){

// Read wordPuzzle matrix

currChar = getc(fp);

while ((!emptyLine) && (!memory)){

if (currChar != BLANK && currChar != NEWLINE){

if(beforeNL){

beforeNL = false;

}

charCtr++;

wordPuzzle = realloc(wordPuzzle, charCtr \* sizeof(char));

if(wordPuzzle != NULL){

wordPuzzle[charCtr-1] = currChar;

}

else{

printf("Memory dynamic reallocation unsuccessfull!");

memory = true;

}

wordPuzzle[charCtr] = currChar;

}

else if (currChar == NEWLINE && col == -1){

col = charCtr;

row = 1;

beforeNL = true;

}

else if (currChar == NEWLINE){

if (beforeNL){

emptyLine = true;

}

else{

row++;

beforeNL = true;

}

}

currChar = getc(fp);

}

// Read words to be searched within wordPuzzle

if (row > col){

lenWord = row;

}

else{

lenWord = col;

}

memory = false;

Word\* words = malloc(numWord \* sizeof(Word));

if (words != NULL){

while (currChar != MARK\_EOF && (!memory)){

numWord++;

words = realloc(words, numWord \* sizeof(Word));

if (words != NULL){

words[numWord-1].contents = (char\*)malloc(lenWord \* sizeof(char\*));

int ctr = 0;

while(currChar != MARK\_EOF && currChar != NEWLINE && currChar != BLANK && currChar != NULLCHAR){

words[numWord-1].contents[ctr] = currChar;

ctr++;

currChar = getc(fp);

}

words[numWord-1].length = ctr;

currChar = getc(fp);

}

else{

printf("Memory dynamic reallocation unsuccessfull!");

memory = true;

}

}

}

else{

printf("Memory allocation unsuccessfull!");

}

// Brute Force Algorithm for Word Puzzle Problem

// Algorithm :

// 1. Start with a word. Iterate through wordPuzzle to find the first letter of the word.

// 2. If the word's first letter is found, check for all directions for the second letter.

// If second letter is found, continue to search letter per letter on that direction.

// Otherwise, move to another direction. Repeat until found or all directions have searched.

// If not found, continue iteration through wordPuzzle. Repeat step 1.

// 3. If found, continue step 1 to another word. Repeat until all words have been found.

// Beginning of brute force algorithm. Start measuring execution time

struct timeval start, end;

clock\_t startS, endS;

int ms\_elapsed = 0;

double s\_elapsed = 0.0;

int numCompare = 0;

for (int i = 0; i < numWord; i++){

int j = 0;

boolean found = false;

char posRowCol[row][col];

for (int a = 0; a < row; a++){

for (int b = 0; b < col; b++){

posRowCol[a][b] = '-';

}

}

for(int j=0; j < words[i].length; j++){

printf("%c", words[i].contents[j]);

}

printf("\n");

while (j < charCtr && (!found)){

gettimeofday(&start, NULL);

startS = clock();

if(wordPuzzle[j] == words[i].contents[0]){

// First letter found

if (words[i].length == 1){ // One letter retrieved

found = true;

}

else {

// Continue to search the rest to all directions

int dir = 1;

while (!found && dir <= 8){

sequentialMatching(j, row, col, wordPuzzle, words[i], &found, posRowCol, dir, &numCompare);

dir++;

}

}

}

gettimeofday(&end, NULL);

endS = clock();

ms\_elapsed += ((end.tv\_sec - start.tv\_sec) \* 1000000) + (end.tv\_usec - start.tv\_usec);

s\_elapsed += (double)(endS - startS) / (double)(CLOCKS\_PER\_SEC);;

numCompare++;

j++;

}

// Finished searching a word

printAns(row, col, posRowCol);

printf("\n");

}

// Print Result

printf("\n%d microseconds ~ %lf\n", ms\_elapsed, s\_elapsed);

printf("\nNumber of comparations made: %d\n", numCompare);

free(wordPuzzle);

free(words);

}

else{

printf("Memory allocation unsuccessfull!");

}

}

fclose(fp);

return 0;

}

**BAB III  
EKSPERIMEN**

1. Eksperimen 1 (Word puzzle berukuran 14 x 12)

Text

Description automatically generated with low confidenceText

Description automatically generated

Shape, rectangle

Description automatically generatedShape, rectangle

Description automatically generatedShape, rectangle

Description automatically generated

Text

Description automatically generated with medium confidence

1. Eksperimen 2 (Word puzzle berukuran 16 x 14)

Text

Description automatically generated with medium confidenceText

Description automatically generatedShape, rectangle

Description automatically generatedShape

Description automatically generatedShape

Description automatically generatedShape

Description automatically generatedShape

Description automatically generatedShape

Description automatically generatedShape

Description automatically generatedShape

Description automatically generatedShape

Description automatically generatedShape

Description automatically generated A picture containing shape

Description automatically generated

1. Eksperimen 3 (Word puzzle berukuran 16 x 16)

A picture containing text, keyboard

Description automatically generated

AFGHANISTAN

ALBANIA

ALGERIA

AND

ANGOLA

ANTIGUA

ARGENTINA

ARMENIA

AUSTRIA

AZERBAIJAN

BAHRAIN

BANGLADESH

BARBUDA

BELARUS

BELIZE

BENIN

BHUTAN

BOLIVIA

CAMBODIA

CAMEROON

CANADA

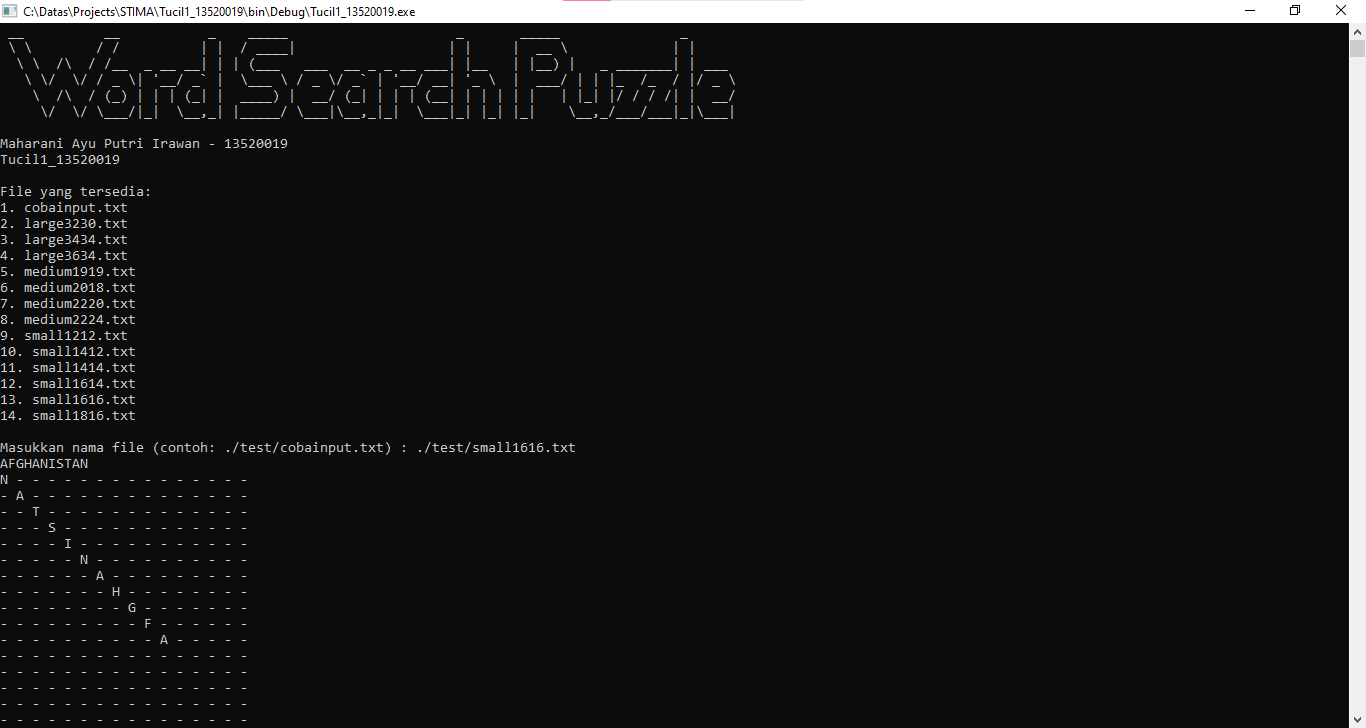
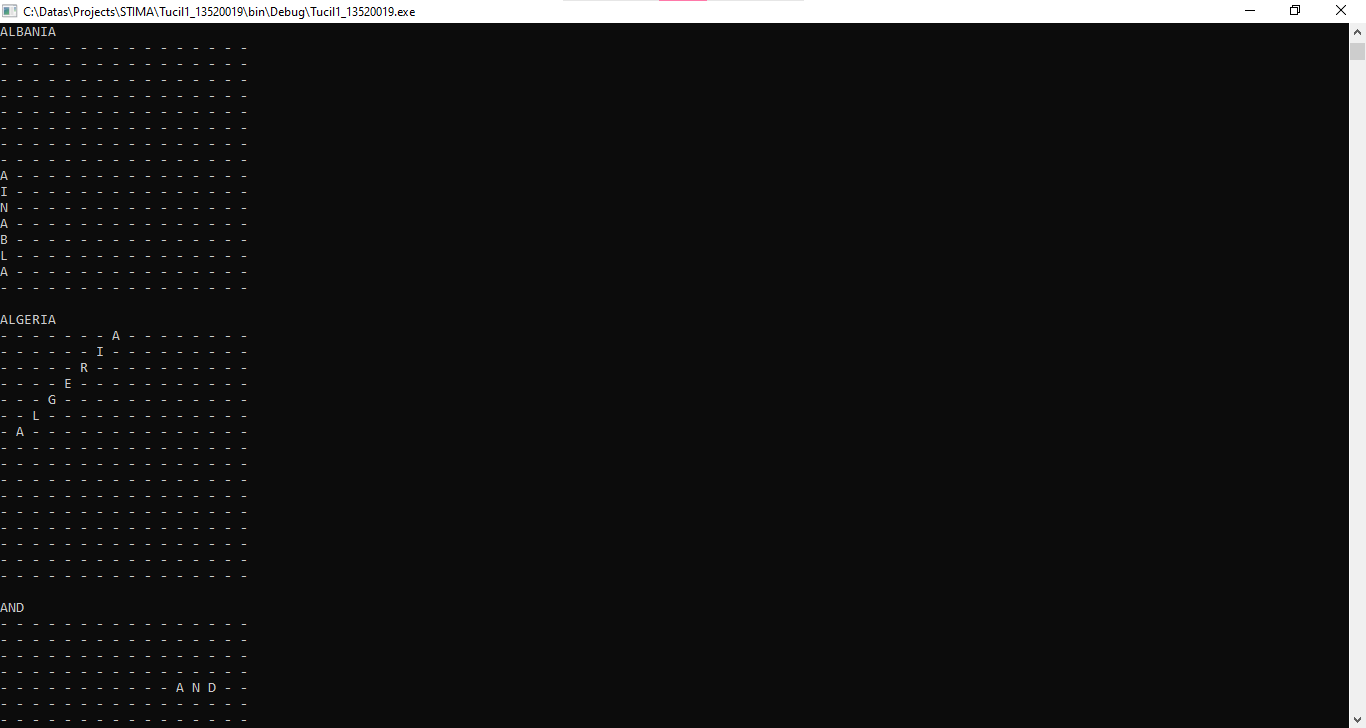
DOMINICA

ECUADOR

EGYPT

ERITREA

ETHIOPIA

  Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Shape

Description automatically generated with low confidence

1. Eksperimen 4 (Word puzzle berukuran 12 x 12)

Text

Description automatically generated with low confidenceText

Description automatically generated

Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Shape, rectangle

Description automatically generated Text

Description automatically generated

1. Eksperimen 5 (Word puzzle berukuran 18 x 16)

Text

Description automatically generated

ALL

AM

ARE

AT

ATE

BE

BLACK

BROWN

BUT

DID

DO

EAT

FOUR

GOOD

HAVE

HE

INTO

LIKE

MUST

NEW

NO

NOW

ON

OUR

OUT

PLEASE

PRETTY

RAN

RIDE

SAW

SAY

SHE

SO

SOON

THAT

THERE

THEY

THIS

TOO

UNDER

WANT

WAS

WELL

WENT

WHAT

WHITE

WHO

WILL

WITH

YES

Text

Description automatically generated

Shape

Description automatically generated Shape, rectangle

Description automatically generated Shape

Description automatically generated Shape

Description automatically generated Shape, rectangle

Description automatically generated Shape

Description automatically generated Shape

Description automatically generated Shape, rectangle

Description automatically generated Shape

Description automatically generated Shape

Description automatically generated Shape, rectangle

Description automatically generated Shape

Description automatically generated Shape

Description automatically generated Shape, rectangle

Description automatically generated Shape

Description automatically generated Shape

Description automatically generated Shape, rectangle

Description automatically generated Shape

Description automatically generated Shape

Description automatically generated Shape, rectangle

Description automatically generated Shape

Description automatically generated Shape

Description automatically generated Shape, rectangle

Description automatically generated Shape

Description automatically generated

Shape

Description automatically generated with low confidence

1. Eksperimen 6 (Word puzzle berukuran 19 x 19)

Shape, arrow

Description automatically generated

AARDVARK

GIRAFFE

MACAW

BABOON

GULL

MAGPIE

BAT

HARTEBEEST

MANDRILL

BLUE

HAWK

ORIOLE

BONOBO

HEDGEHOG

OWL

CANARY

HERON

PENGUIN

CHEETAH

HYRAX

PUFFIN

CHIMPANZEE

IBIS

QUETZAL

CRANE

IMPALA

RHINOCEROS

CROW

JACKAL

SWAN

DUCK

JAY

TERN

EAGLE

KIWI

TOUCAN

EGRET

LARK

TURKEY

FALCON

LEOPARD

WARTHOG

GELADA

LION

GENET

LOON

Text

Description automatically generated

Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Shape

Description automatically generated with medium confidence Text

Description automatically generated

1. Eksperimen 7 (Word puzzle berukuran 20 x 18)

A picture containing shape

Description automatically generated Text

Description automatically generated A picture containing text

Description automatically generated A picture containing text

Description automatically generated Shape

Description automatically generated with low confidence A picture containing text

Description automatically generated Text

Description automatically generated with medium confidence

1. Eksperimen 8 (Word puzzle berukuran 22 x 20)

Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated with low confidence Text

Description automatically generated

1. Eksperimen 9 (Word puzzle berukuran 22 x 24)

A picture containing text, indoor, tiled

Description automatically generated Text, chat or text message

Description automatically generatedText

Description automatically generated

Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated

1. Eksperimen 10 (Word puzzle berukuran 32 x 30)

Background pattern

Description automatically generated Text, chat or text message

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated with medium confidence Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated with medium confidence Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated

1. Eksperimen 11 (Word puzzle berukuran 34 x 34)

Background pattern

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated Text

Description automatically generated Text

Description automatically generated with low confidence Text

Description automatically generated with medium confidence Text

Description automatically generated with medium confidence Text

Description automatically generated with medium confidence Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated with medium confidence Text

Description automatically generated with medium confidence Text

Description automatically generated Text

Description automatically generated with medium confidence Text

Description automatically generated with medium confidence Text

Description automatically generated with medium confidence Text

Description automatically generated with medium confidence Text

Description automatically generated with medium confidence Text

Description automatically generated Text

Description automatically generated with low confidence Text

Description automatically generated with medium confidence Text

Description automatically generated with low confidence A picture containing text

Description automatically generated Text

Description automatically generated with medium confidence A picture containing text

Description automatically generated

Text

Description automatically generated with medium confidence

1. Eksperimen 12 (Word puzzle berukuran 36 x 34)

Background pattern

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated A picture containing text

Description automatically generated Text

Description automatically generated with low confidence Text

Description automatically generated with low confidence A picture containing text

Description automatically generated Text

Description automatically generated with medium confidence A picture containing text

Description automatically generated Text

Description automatically generated with low confidence Text

Description automatically generated with medium confidence Text

Description automatically generated with medium confidence Text

Description automatically generated with medium confidence Text

Description automatically generated with low confidence Text

Description automatically generated Text

Description automatically generated with medium confidence Text

Description automatically generated with medium confidence A picture containing text

Description automatically generated Text

Description automatically generated with low confidence Text

Description automatically generated with medium confidence A picture containing text

Description automatically generated Text

Description automatically generated with medium confidence Text

Description automatically generated with low confidence Text

Description automatically generated with medium confidence Text

Description automatically generated with low confidence Text

Description automatically generated with low confidence Text

Description automatically generated with low confidence A picture containing text

Description automatically generated Text

Description automatically generated with low confidence Text

Description automatically generated with low confidence A picture containing text

Description automatically generated Text

Description automatically generated with low confidence Text

Description automatically generated with low confidence A picture containing text

Description automatically generated Text

Description automatically generated with low confidence A picture containing text

Description automatically generated A picture containing text

Description automatically generated Text

Description automatically generated with low confidence A picture containing text

Description automatically generated Text

Description automatically generated with medium confidence A picture containing text

Description automatically generated Text

Description automatically generated with medium confidence

**BAB IV  
CHECKLIST**

|  |  |  |
| --- | --- | --- |
| Poin | Ya | Tidak |
| 1. Program berhasil dikompilasi tanpa kesalahan (no syntax error) | **✓** |  |
| 1. Program berhasil *running* | **✓** |  |
| 1. Program dapat membaca file masukan dan menuliskan luaran. | **✓** |  |
| 1. Program berhasil menemukan semua kata di dalam puzzle. | **✓** |  |