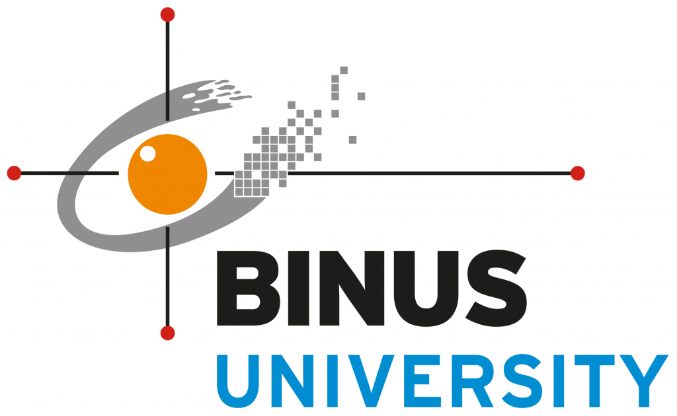
Attendance and Fine Program for Youth in Penestanan Village

AOL TASK CASE STUDY REPORT

MATA KULIAH COMP6360004 – ALGORITHM AND PROGRAMMING

CLASS < LA20>



By:

<2602139362> - <I Made Ananda Mahatresna>

Odd Semester 2022\2023

HAPLESS

Table of Contents

[CHAPTER 1 3](#_Toc125561728)

[CHAPTER 2 5](#_Toc125561729)

[CHAPTER 3 10](#_Toc125561730)

[CHAPTER 4 29](#_Toc125561731)

[BAB 5 36](#_Toc125561732)

# BAB 1

**Introduction**

For the project I made, I intended for my village youth management to be absent at every event or meeting. Therefore, in this program, users must enter a password first to secure youth data from irresponsible parties. Youth or women who do not come or are not seen at a meeting will be fined based on the reason for not attending the activity. The data that will be entered in the program are name, date of recording, amount of fine and name of the recorder/data maker. Here are photos of wantilan that are often used as places for deliberations and village events:

A picture containing outdoor, building

Description automatically generated A picture containing building, outdoor, ground, parked

Description automatically generated

**1. Project Idea**

For the project idea that I used in this final project, it was to create an attendance system with fines within the scope of the youth organization in the village of Penestanan Kaja Bali (the village where I live).

**2. Loop Control Structure**

For loops in my program is to use "for". Some of the functions that use loops include read, write, delete, update, search and sort functions. For the variables I use in the loop there are variables I and j.

**3. Selection**

The use of selection that I use is IF ELSE and SWITCH CASE, for the use of IF ELSE I use for example such as for the creation of login programs, input passwords and output programs. I use SWITCH CASE in menu selection including Login Menu, Main Menu, Search Menu and Sort Menu. For the variables I use in IF ELSE are int pw for password, int deletekey for output of the delete function, int updateiKey for output of the update function, int sNameKey for output of the searchByName function, int sDateKey for output of the searchByDate function and int sNameKey for output of the searchByBy function. The variable used for the SWITCH CASE is the input int.

**4. Data Type**

I use several data types in my program including, main integer, void print, print void, read read, write, void delete, update void, void searchByName, void searchByDate, void searchByBy, void sortByName, void sortByDate, voidByBy, void sort, void search, struct Absent and struct Table.

**5. Storage Media**

The storage media I use in my program are "absen.txt" and "table.txt".

## 

## BAB 2

Proposed program design in the form of a flowchart and its explanation

1. Function Main

Diagram

Description automatically generated



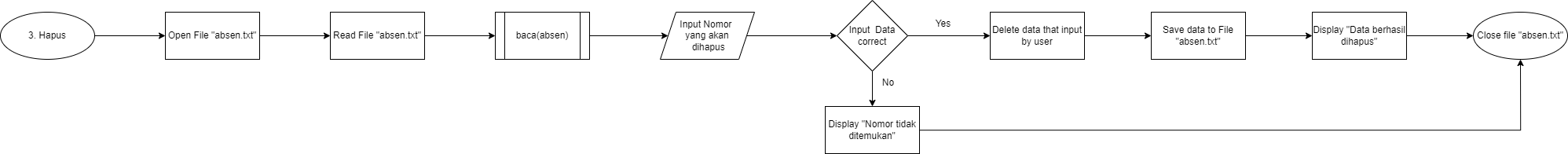
2. Function baca



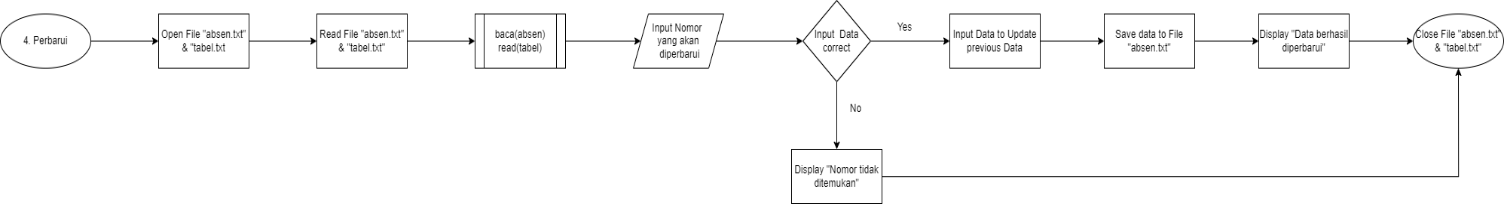
3. Function tulis



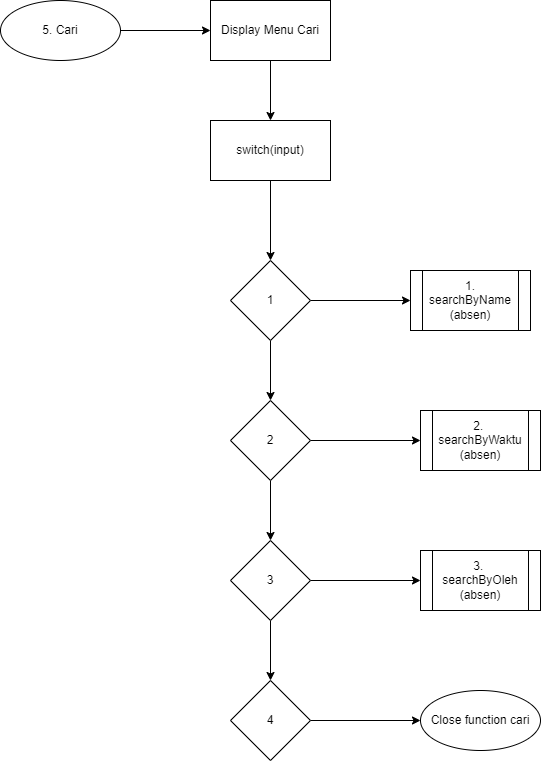
4. Function hapus



5. Update function



6. function cari



7. Function searchByName

Graphical user interface

Description automatically generated

8. Function searchByWaktu

Graphical user interface

Description automatically generated with medium confidence

9. Function searchByOleh

Graphical user interface

Description automatically generated with medium confidence

10. Function urut

A picture containing Teams

Description automatically generated

11. Function sortByName



12. Function sortByTime



13. Function sortByBy



### BAB 3

#include <stdio.h>

#include <string.h>

#include <stdio.h>

#include <time.h>

struct Absent to read and write data from the

struct Absen{

int no;

char name[50];

char date[50];

long long int denda;

char by[50];

};

struct table to display fines

struct Tabel{

slow char[50];

char sleep[50];

In Sir's Lamp[50];

Sir Ezin [50];

Char Noket[50];

};

functions to create/tidy up tables

void print()

{

for (int i=0; i<105; i++){

printf("=");

}

printf("\n");

}

functions to create/tidy up tables

void print()

{

for (int i=0; i<113; i++){

printf("=");

}

printf("\n");

}

functions to call or read the contents of a absen.txt file

will display data from absen.txt if called

void read(struct Absent Absent[100]){//call the Absent struct with the name Absent

system("cls");

int count = 0;

FILE \*fp;

opens the read void function to read the contents of the absen.txt file

fp = fopen("absen.txt", "r");

while(fscanf(fp,"%d;%[^\';']; %[^\';']; %lld;%[^\n]\n", &absen[count].no, absen[count].nama, absen[count].tanggal, &absen[count].denda, absen[count].oleh) != EOF)

{

count++;

}//while menu above to create a loop to read the contents of the absen.txt file

time\_t t = time(NULL);

struct tm tm = \*localtime(&t);

printf("%02d-%02d-%d\n\n", tm.tm\_mday, tm.tm\_mon + 1, tm.tm\_year + 1900); Lines 50-52 i.e. to display the time (date-month-year)

print();

printf("|No#\t|\tNama\t\t|\tTanggal\t\t|\tDenda\t\t|\tOleh\t\t|\n");

print();

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

printf("|%-4d\t|\t%-12s\t|\t%-12s\t|\t%-12lld\t|\t%-10s\t|\n", absen[i].no, absen[i].nama, absen[i].tanggal, absen[i].denda, absen[i].oleh);

}//displays the contents of the absen.txt file

print();

printf("\n");

fclose(fp);//close the file or close the read void function

}

functions to call or read the contents of a tabel.txt file

will display data from tabel.txt if called

void read(struct Table Table[100]){//call the struct Table with the table name

system("cls");

struct Absen absen[100];

int count = 0;

FILE \*fp;

opens the read void function to read the contents of the tabel.txt file

fp = fopen("tabel.txt", "r");

while(fscanf(fp,"%[^\';']; %[^\';']; %[^\';']; %[^\';']; %[^\n]\n", tabel[count].lambat, tabel[count].tidur, tabel[count].sakit, tabel[count].ijin, tabel[count].noket) != EOF)

{

count++;

}//while menu above to create a loop to read the contents of the tabel.txt file until the EOF or file ends

time\_t t = time(NULL);

struct tm tm = \*localtime(&t);

printf("%02d-%02d-%d\n\n", tm.tm\_mday, tm.tm\_mon + 1, tm.tm\_year + 1900); lines 74-76 i.e. to display the time (date-month-year)

read(absent);

printf("Table Fine\n");

print();

printf("|Late\t|\tSleeping\t|\tSick\t\t|\tPermission\t\t|\tWithout Explanation|\n");

print();

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

printf("|%-10s\t|\t%-12s\t|\t%-12s\t|\t%-12s\t|\t%-10s\t|\n", tabel[i].lambat, tabel[i].tidur, tabel[i].sakit, tabel[i].ijin, tabel[i].noket);

}//displays the contents of the tabel.txt file

print();

printf("\n");

fclose(fp);//close the file or close the void read function

}

functions for writing or inputting data into absen.txt files

void write(struct Absent Absent[100]){//call the Absent struct with the name Absent

system("cls");

struct Table table[100];

time\_t t = time(NULL);

struct tm tm = \*localtime(&t);//function to display time

int count = 0;

FILE \*fp;

fp = fopen("absen.txt","r");//open/read absen.txt file

while(fscanf(fp,"%d;%[^\';']; %[^\';']; %lld;%[^\n]\n", &absen[count].no, absen[count].nama, absen[count].tanggal, &absen[count].denda, absen[count].oleh) != EOF){

count++;

}//while menu above to create a loop to read the contents of the absen.txt file to the end of the file

fclose(fp); display

read(table);

fp = fopen("absen.txt","a");

For lines 106-118 the user can enter data to be stored in the absen.txt file

printf("Add Data\n");

printf("Enter Number: ");

scanf("%d", &absen->no);

getchar();

printf("Insert Name : ");

scanf("%[^\n]", absen->nama);

getchar();

printf("Enter Fine : ");

scanf("%lld", &absen->denda);

getchar();

printf("Data entered by : ");

scanf("%[^\n]", absent->by);

getchar();

(count == 0) ? fprintf(fp,"%d;%s;%02d-%02d-%d;%lld;%s", absen->no, absen->nama, tm.tm\_mday, tm.tm\_mon + 1, tm.tm\_year + 1900, absen->denda, absen->oleh) :

fprintf(fp,"\n%d;%s;%02d-%02d-%d;%lld;%s", absen->no, absen->nama, tm.tm\_mday, tm.tm\_mon + 1, tm.tm\_year + 1900, absen->denda, absen->oleh);

printf("Data Added!\n");//on lines 83 and 84 the data is already stored in the

system("pause");

system("cls");

fclose(fp);

}

function to delete row data by entering the row number to be deleted

void delete(struct Absent Absent[100]){//call the Absent struct with the name Absent

system("cls");

FILE \*fp;

int count = 0;

fp = fopen("absen.txt","r");//open/read absen.txt file

while(fscanf(fp,"%d;%[^\';']; %[^\';']; %lld;%[^\n]\n", &absen[count].no, absen[count].nama, absen[count].tanggal, &absen[count].denda, absen[count].oleh) != EOF){

count++;

}//while menu above to create a loop to read the contents of the absen.txt file to the end of the file

fclose(fp);

int delete;

read(absent);//display data in the file to make it easier to select the selection in the delete function

printf("Enter the Number you want to delete : ");

scanf("%d", &delete); getchar();//input and scan the input results

int deletekey = -1;

fp = fopen("absen.txt", "w");//write or make changes to the file

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

if(absen[i].no==delete){

deletekey=i;

continue;//If the user input matches the name data in the absen.txt file, then the number and 1 line will be skipped (deleted) from the file

}

else{

fprintf(fp,"%d;%s;%s;%ld;%s\n", absen[i].no, absen[i].nama, absen[i].tanggal, absen[i].denda, absen[i].oleh);

}

}

if(deletekey != -1){

printf("Successfully deleted\n");//If the user input matches or matches the data in the file, this output will be output

system("pause");

system("cls");

}

else{

printf("Number not found\n");//if the user input does not match/match the data feed, this output will come out

system("pause");

system("cls");

}

fclose(fp);

}

functions to update data

void update(struct Absent absent[100]){//call the absent struct with the name absent

system("cls");

struct Table table[100];

FILE \*fp;

int count = 0;

fp = fopen("absen.txt","r");

while(fscanf(fp,"%d;%[^\';']; %[^\';']; %lld;%[^\n]\n", &absen[count].no, absen[count].nama, absen[count].tanggal, &absen[count].denda, absen[count].oleh) != EOF){

count++;

}//while menu above to create a loop to read the contents of the absen.txt file to the end of the file

fclose(fp);

read(table);//calls the read function to make it easier for the user to select a selection

fp = fopen("absen.txt", "w");

time\_t t = time(NULL);

struct tm tm = \*localtime(&t);

int noUpdate;

int updateKey = -1;

printf("Enter the number to be updated: ");

scanf("%d", &noUpdate); getchar();//input the number variable to be updated

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

if (absent[i].no == noUpdate){

updateKey = i;

printf("Insert New Number : ");

scanf("%d", &absent[updateKey].no);

getchar();

printf("Insert new name :");

scanf("%[^\n]", absent[updateKey].name);

getchar();

printf("Enter a new Fine :");

scanf("%lld", &absent[updateKey].fine);

getchar();

printf("Insert a new Creator :");

scanf("%[^\n]", absent[updateKey].by);

getchar();//enter the data to be updated

fprintf(fp,"%d;%s;%02d-%02d-%d;%lld;%s\n", absen[i].no, absen[i].nama, tm.tm\_mday, tm.tm\_mon + 1, tm.tm\_year + 1900, absen[i].denda, absen[i].oleh);

}

else{

fprintf(fp,"%d;%s;%02d-%02d-%d;%lld;%s\n", absen[i].no, absen[i].nama, tm.tm\_mday, tm.tm\_mon + 1, tm.tm\_year + 1900, absen[i].denda, absen[i].oleh);

}//update data on the file

}

if(updateKey == -1){

printf("Number not found!\n");//if the input variable does not match/match the contents of the file then this output will be output

system("pause");

system("cls");

}

else{

printf("Data Successfully Updated\n");//if the input variables match/match the contents of the file, this output will be output

system("pause");

system("cls");

}

fclose(fp);

}

functions to search for and call names

void searchByName(struct Absen absen[100]){

system("cls");

FILE \*fp;

int count=0;

fp = fopen("absen.txt","r");

while(fscanf(fp,"%d;%[^\';']; %[^\';']; %lld;%[^\n]\n", &absen[count].no, absen[count].nama, absen[count].tanggal, &absen[count].denda, absen[count].oleh) != EOF){

count++;

}

char searchnama[15];

int searchNama;

int sNamaKey = -1;

printf("Enter the name you want to search for : ");

scanf("%s", searchname);//manginput the name to be searched

getchar();

print();

printf("|No#\t|\tNama\t\t|\tTanggal\t\t|\tDenda\t\t|\tOleh\t\t|\n");

print();

int i;

for (i=0; i<count; i++){

if(strcmpi(absen[i].nama,searchnama)==0){

sNamaKey = i;

printf("|%-4d\t|\t%-12s\t|\t%-12s\t|\t%-12lld\t|\t%-10s\t|\n", absen[i].no, absen[i].nama, absen[i].tanggal, absen[i].denda, absen[i].oleh);

}//calls the name and all the data along with the input name

}

print();

if(sNamaKey == -1){

printf("Name not found!\n");//if the input name variable does not match/match the contents of the file, this output will come out

}

printf("\n");

fclose(fp);

}

void searchByWaktu(struct Absen absen[100]){

system("cls");

FILE \*fp;

int count=0;

fp = fopen("absen.txt","r");

while(fscanf(fp,"%d;%[^\';']; %[^\';']; %lld;%[^\n]\n", &absen[count].no, absen[count].nama, absen[count].tanggal, &absen[count].denda, absen[count].oleh) != EOF){

count++;

}

char searchdate[15];

int searchDate;

int sDateKey = -1;

printf("Enter the date you want to search : ");

scanf("%[^\n]", searchdate);//manginput the date to be searched

getchar();

print();

printf("|No#\t|\tNama\t\t|\tTanggal\t\t|\tDenda\t\t|\tOleh\t\t|\n");

print();

int i;

for (i=0; i<count; i++){

if(strcmpi(absent[i].date,searchdate)==0){

sDateKey = i;

printf("|%-4d\t|\t%-12s\t|\t%-12s\t|\t%-12lld\t|\t%-10s\t|\n", absen[i].no, absen[i].nama, absen[i].tanggal, absen[i].denda, absen[i].oleh);

}//calls the time and all the data along with the input time

}

print();

if(sDateKey == -1){

printf("Date not found!\n");//if the input time variable does not match/match the contents of the feed file this output will come out

}

printf("\n");

fclose(fp);

}

function to search for and call the

void searchByOleh(struct Absen absen[100]){

system("cls");

FILE \*fp;

int count=0;

fp = fopen("absen.txt","r");

while(fscanf(fp,"%d;%[^\';']; %[^\';']; %lld;%[^\n]\n", &absen[count].no, absen[count].nama, absen[count].tanggal, &absen[count].denda, absen[count].oleh) != EOF){

count++;

}

char searchnama[15];

int searchNama;

int sNamaKey = -1;

printf("Enter the name you want to search for : ");

scanf("%s", searchname);//input the creator you want to search for

getchar();

print();

printf("|No#\t|\tNama\t\t|\tTanggal\t\t|\tDenda\t\t|\tOleh\t\t|\n");

print();

int i;

for (i=0; i<count; i++){

if(strcmpi(absent[i].by,searchname)==0){

sNamaKey = i;

printf("|%-4d\t|\t%-12s\t|\t%-12s\t|\t%-12lld\t|\t%-10s\t|\n", absen[i].no, absen[i].nama, absen[i].tanggal, absen[i].denda, absen[i].oleh);

}//calls the name and all the data along with the input name

}

print();

if(sNamaKey == -1){

printf("Maker not found!\n");//if the input maker variable does not match/match the contents of the feed file will output this output

}

printf("\n");

fclose(fp);

}

functions for sorting names

void sortByName(struct Absen absen[100]) {

system("cls");

struct Absen temp;

FILE \*fp;

fp=fopen("absen.txt", "r");

int count = 0;

while(fscanf(fp,"%d;%[^\';']; %[^\';']; %lld;%[^\n]\n", &absen[count].no, absen[count].nama, absen[count].tanggal, &absen[count].denda, absen[count].oleh) != EOF){

count++;

}

Four Temptures[100];

int tempInt;

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

for(int j=i+1; j<count; j++){

IF(STRCMP(absen[i].name, absen[j].name)>0){

temp=absen[i];

absen[i]=absen[j];

absent[j]=temp;//function of lines 260-265 swaps the name variable every 2 lines in the file, sorting the earlier alphabet to be changed to the top and the last one to be changed to the bottom

}

}

}

print();

printf("|No#\t|\tNama\t\t|\tTanggal\t\t|\tDenda\t\t|\tOleh\t\t|\n");

print();

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

printf("|%-4d\t|\t%-12s\t|\t%-12s\t|\t%-12lld\t|\t%-10s\t|\n", absen[i].no, absen[i].nama, absen[i].tanggal, absen[i].denda, absen[i].oleh);

}

print();//displays the order of the data according to the order of names starting from the lowest/first

printf("\n");

}

Sort data according to time (date)

void sortByDate(struct Absen absen[100]) {

system("cls");

struct Absen temp;

FILE \*fp;

fp=fopen("absen.txt", "r");

int count = 0;

while(fscanf(fp,"%d;%[^\';']; %[^\';']; %lld;%[^\n]\n", &absen[count].no, absen[count].nama, absen[count].tanggal, &absen[count].denda, absen[count].oleh) != EOF){

count++;

}

Four Temptures[100];

int tempInt;

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

for(int j=i+1; j<count; j++){

if(strcmp(absent[i].date, absent[j].date)<0){

temp=absen[i];

absen[i]=absen[j];

absent[j]=temp;//from lines 290-295 sorts the data on the file according to the time the data was created

}

}

}

print();

printf("|No#\t|\tNama\t\t|\tTanggal\t\t|\tDenda\t\t|\tOleh\t\t|\n");

print();

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

printf("|%-4d\t|\t%-12s\t|\t%-12s\t|\t%-12lld\t|\t%-10s\t|\n", absen[i].no, absen[i].nama, absen[i].tanggal, absen[i].denda, absen[i].oleh);

}//displays data according to the time it was created

print();

printf("\n");

}

functions for sorting names

void sortByOleh(struct Absen absen[100]) {

system("cls");

struct Absen temp;

FILE \*fp;

fp=fopen("absen.txt", "r");

int count = 0;

while(fscanf(fp,"%d;%[^\';']; %[^\';']; %lld;%[^\n]\n", &absen[count].no, absen[count].nama, absen[count].tanggal, &absen[count].denda, absen[count].oleh) != EOF){

count++;

}

Four Temptures[100];

int tempInt;

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

for(int j=i+1; j<count; j++){

if(strcmp(absent[i].by, absent[j].by)>0){

temp=absen[i];

absen[i]=absen[j];

absent[j]=temp;//function of lines 320-325 swaps the name variable every 2 lines in the file, sorting the earlier alphabet to be changed to the top and the last one to be changed to the bottom.

}

}

}

print();

printf("|No#\t|\tNama\t\t|\tTanggal\t\t|\tDenda\t\t|\tOleh\t\t|\n");

print();

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

printf("|%-4d\t|\t%-12s\t|\t%-12s\t|\t%-12lld\t|\t%-10s\t|\n", absen[i].no, absen[i].nama, absen[i].tanggal, absen[i].denda, absen[i].oleh);

}

print();//displays the order of the data according to the order of the creator/by starting from the lowest/initial

printf("\n");

}

function to display the search menu

For search, I use linear search where if the input from the user is the same as the data from the absen.txt file, it will call the data in the program

void search(struct Absen absen[100])

{

system("cls");

FILE \*fp;

fp=fopen("absen.txt", "r");

int input;

do{

printf("Data Search\n");//in the search menu there are 3 options to search by name and creator and menu 3 returns the user to the main menu

printf("1. Search for Name\n");

printf("2. Find Time/Date created\n");

printf("3. Search for Creator Name\n");

printf("4. Exit\n");

printf("Select Menu (1/2/3/4) : ");

scanf("%d", &input); getchar();//user enter the menu to be selected

switch(input)

{

case 1: //number 1 displays/calls the name lookup function

searchByName(absen);

system("pause");

system("cls");

break;

case 2:

searchByTime(absent);//number 2 displays/calls the search function for time/date

system("pause");

system("cls");

break;

case 3://number 3 displays/calls the function of finding creators/by

searchByBy(absent);

system("pause");

system("cls");

break;

}

}while(input >=1 && input <4);//while here serves to return the user to the main menu if input a variable is more than 3

fclose(fp);

}

function to display the sorting menu

For sorting I use bubble sort by operating indexes to I and J

void sort(struct Absen absen[100])

{

system("cls");

FILE \*fp;

fp=fopen("absen.txt", "r");

int input;

do{

printf("Data Sorting\n");//In the sort menu there are 3 options to sort by name and time (date) and menu 3 returns the user to the main menu

printf("1. Sort Name\n");

printf("2. Sort Date\n");

printf("3. Sort Makers\n");

printf("4. Exit\n");

printf("Select Menu (1/2/3) : ");

scanf("%d", &input); getchar();//user enter the menu to be selected

switch(input)

{

case 1://number 1 displays/calls the name sorting function

Shortbinem(Absen);

system("pause");

system("cls");

break;

case 2://number 2 displays/calls the sorting function by time(date)

blackByDate(absen);

system("pause");

system("cls");

break;

case 3:

sortByBy(absent);//number 3 displays/calls the function of sorting by creator/by

system("pause");

system("cls");

break;

}

}while(input >=1 && input <4);//while here serves to return the user to the main menu if input a variable is more than 3

fclose(fp);

}

Main Menu

int main(){

struct Absen absen[100];

time\_t t = time(NULL);

struct tm tm = \*localtime(&t);//function to display time(date-month-year)

int input;

int inputuser;

char pw[8];

char PW1[6];

do{

system("cls");

printf("%02d-%02d-%d\n\n", tm.tm\_mday, tm.tm\_mon + 1, tm.tm\_year + 1900);

printf("Banjar Penestanan Kaja Youth Attendance Data\n");//The first menu shown in the program is the Login menu

printf("Menu Login\n");

printf("1. Login\n");

printf("2. Exit\n");

printf("Select (1/2): ");

scanf("%d", &inputuser);//user enter the menu to be selected

switch(inputuser)

{

case 1://menu 1 The user will log in to the program and must enter the password to go to the main menu

system("cls");

printf("(password=login)\n");

printf("Enter Password : ");//user here enters the correct password

flush(stdin);

char ch;

int i = 0;

while((ch = (char) \_getch()) != '\r'){

pw[i] = ch;

printf("\*");//lines 394-398 to censor user input with an asterisk

i++;

}printf("\n");

if (strcmpi(pw,"login")==0)

{

printf ("Login Successfully\n");//if the user enters the password correctly, the following output will be output and moved to the main menu

}

else

{

printf("Wrong Password!\n");//if the user enters the wrong password, the output will be as follows

printf("Input Password one more time : ");//user will be given one more chance to enter password

flush(stdin);

char ch;

int i = 0;

while((ch = (char) \_getch()) != '\r'){

pw1[i] = ch;

printf("\*");

i++;

}printf("\n");

if (strcmpi(pw1,"login")==0)

{

printf ("Login Successfully\n");

}

else{

printf("Wrong Password!!! \n");

return 0;//If the user's input is wrong again, the user will exit the menu and the program ends

}

}

system("pause");

do{

system("cls");

printf("%02d-%02d-%d\n\n", tm.tm\_mday, tm.tm\_mon + 1, tm.tm\_year + 1900);

printf("Main Menu\n");//on the main menu the user will be given 7 options

printf("1. Baca\n");

printf("2. Tulis\n");

printf("3. Hapus\n");

printf("4. Update\n");

printf("5. Cari\n");

printf("6. Urut\n");

printf("7. Exit\n");

printf("Select Menu (1/2/3/4/5/6/7): ");

scanf("%d", &input);//user input the selected option

switch(input){ //uses a switch case for the user so that the user can choose between existing switch cases

case 1://number 1 will call the read function

read(absent);

system("pause");

system("cls");

break;

case 2://nomnor 2 will call the write function

write (absent);

break;

case 3://number 3 will call the delete function

delete(absent);

break;

case 4://number 4 will call the update function

update(absent);

break;

case 5://number 5 will display the search/search function

search(absen);

system("pause");

system("cls");

break;

case 6://number 6 akakn displays the sort/sorting function

sort(absent);

system("pause");

system("cls");

break;

default:

break;

}

}while(input >=1 && input <7);//while here to return the user to the login menu if the user inputs more than 6 in the main menu

}

}while(inputuser>0 && inputuser<2);//while This serves to terminate the program if the user inputs more than 1 in the login menu.

return 0;

}

#### BAB 4

Program results display

Login Menu (Home View)

Text

Description automatically generated

Input Password

Shape

Description automatically generated with medium confidence

Main Menu

A picture containing text

Description automatically generated

Option no. 1 Read\read Data

Text

Description automatically generated

Option no.2 Write/Write Data (there are read and table functions)

A picture containing text

Description automatically generated

Option 3. Delete Data (there is a read function)

Text

Description automatically generated

Option no.4 Update/update Data (there is a read and write function)

A picture containing text

Description automatically generated

Option no.5 will display the Data Search Menu

A picture containing text

Description automatically generated

1. Search by Name

Text

Description automatically generated

2. Search by Date

Text

Description automatically generated

3. Search by Creator/By Name

Text

Description automatically generated

Option no.6 will display the Data Sorting Menu

Shape

Description automatically generated with medium confidence

1. Sorting by Name

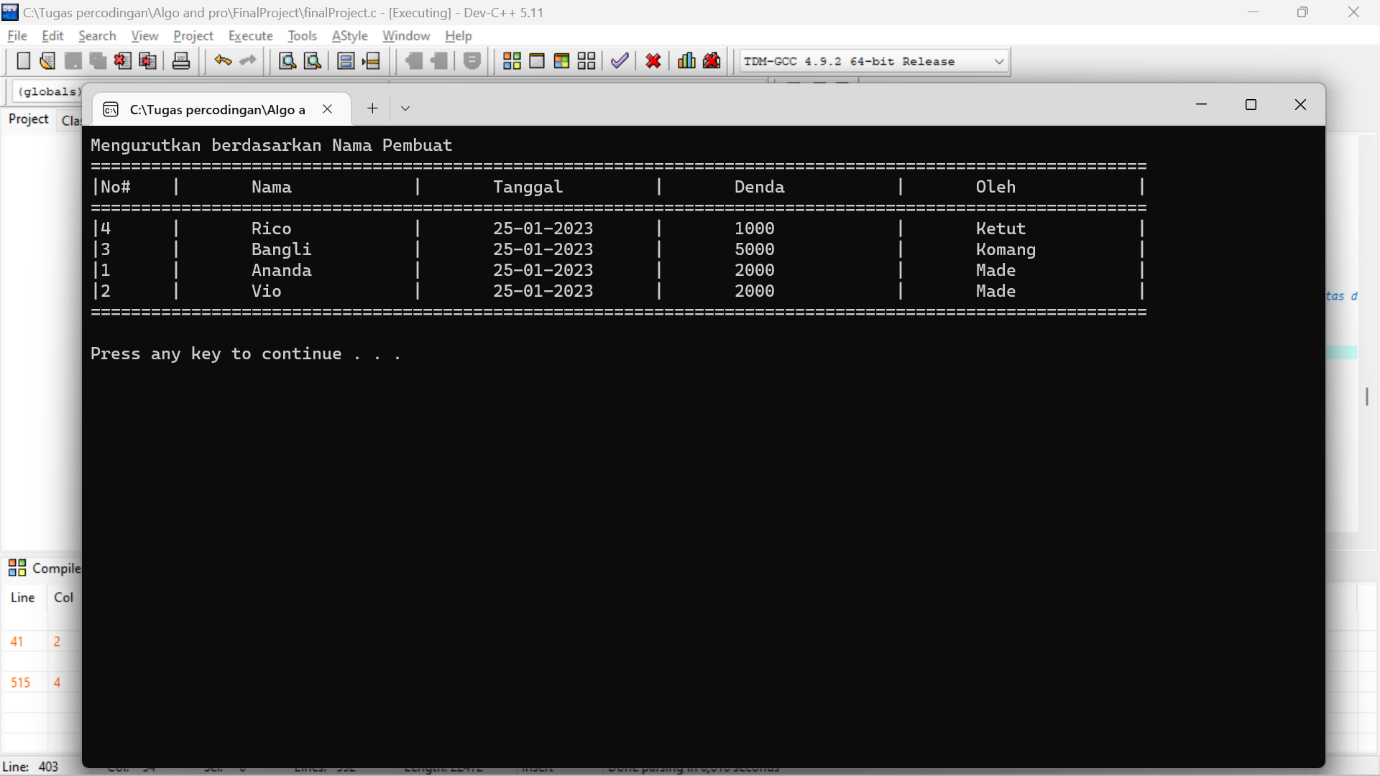
A screenshot of a computer

Description automatically generated

2. Sort by Latest Date

A screenshot of a computer

Description automatically generated

3. Sort by Creator Name/by

##### CHAPTER 5

The coding program entitled "Attendance and Fine Program for Youth in Penestanan Village" is a program made to help the village management to record its members in attendance and attendance at meetings held or organized by youth organizations in Penestanan Village. The program is made based on analysis and organizational needs to record members quickly, accurately and practically. The program can record several data such as the name of the subject or member, the date of data collection, the amount of the fine and the name of the data collector/creator responsible for the data. This data has not been perfected, in the future this data can be developed again to make it easier for organization administrators to operate and use this program.