**PEMROGRAMAN LANJUT**

**LAPORAN RESMI FINAL PROJECT**

**APLIKASI PELABUHAN**

****

**PARALEL : B**

**KELOMPOK 9 :**

Mochammad Faisal Nur Sayyid (18081010104)

Dimas Amrulloh (18081010125)

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”**

**JAWA TIMUR**

**2019**

**DESKRIPSI APLIKASI**

Final project ini memiliki judul “Aplikasi Pelabuhan” merupakan tugas akhir dari mata kuliah “Pemrograman Lanjut” yang mana menggunakan bahasa C sebagai penyelesaiannya. Aplikasi Pelabuhan yang kami buat dan kembangkan ini menerapkan metode CRUD (Create, Read, Update, Delete) dan juga menambahkan metode searching berupa linier search rekursif (sequential search menggunakan rekursif) serta metode sorting berupa bucket sort.

Dalam aplikasi final project ini, terdapat 5 menu utama yaitu: tambah data, update data, cari data, urutkan data,dan keluar aplikasi. Dimana setiap menu memiliki fungsinya masing-masing sehingga aplikasi tersebut dapat melakukan dan mempermudah eksekusi, baik dari input data, read data, delete data, dan update data.

Tujuan dari pembuatan aplikasi ini yaitu untuk memudahkan user dalam melakukan sebuah pengecekan kendaraan yang keluar dan masuk pada pelabuhan tersebut. Aplikasi yang kami buat ini juga sangat mudah dipahami sehingga dalam penggunaannya nanti diharapkan tidak mengalami suatu kendala.

**METODE APLIKASI**

Pada final project ini menitik beratkan pada metode *searching* dan *sorting,* disini metode yang digunakn adalah linear searching rekursif dan bucket sorting. *Linear Search* atau *Sequential Search* adalah pencarian data secara linier (garis lurus), artinya adalah pencarian dilakukan secara teratur (secara sekuensial) dari awal sampai akhir data (atau bisa juga dari akhir ke awal data). Dengan kompleksitas waktu terbaik adalah O(n).

Sedangkan *Bucket Sort* sendiri adalah algoritme berbasis perbandingan yang beroperasi pada elemen dengan mengelompokkannya pada beberapa bucket berbeda lantas bucket tersebut diurutkan secara terpisah. Setiap bucket dapat diurutkan secara terpisah menggunakan algoritme sorting yang berbeda atau dengan menerapkan algoritme bucket sort secara rekursif. Bucket sort amat sangat berguna saat digunakan pada input yang terdistribusi secara seragam dalam rentangan. Dengan kompleksitas waktu terbaik sebesar O(NlogN)

**SOURCECODE**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

void print(int n,int tiket[10],int nopol[10],char jenisK[10][12],int bobot[10],char tujuan[10][12],int tglT[10]);

void bucketSort(int nopol[], int max,int k){

int i, j;

int c=0;

int bucket[max];

int n;

n = sizeof(bucket)/sizeof(bucket[0]);

// printf("P = %d", p);

printf("masuk fungsi");

// Isi bucket dengan bilangan default 0

for (i = k; i <= k; i++){

bucket[i] = 0;

}

printf("masuk fungsi");

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

bucket[nopol[i]] = bucket[nopol[i]] + 1;

}

printf("masuk fungsi");

for (i = 0; i <= k; i++) { // for ini sebagai index tempat bilangan nya

for(j = 0; j < bucket[i]; j++){ // for ini akan ke skip jika bucket kosong

nopol[c] = i;

//printf("\n bucket = %d \n", bucket[i]);

//printf("\n Test 2 = %d \n", array[c]);

//printf("\n C = %d", c);

c++;

}

//printf("\n===\n");

}

}

Gambar 1 - Sourcecode 1

int main() {

int tiket[10];

int bobot[10];

char jenisK[10][12];

char tujuan[10][12];

int tglT[10];

int nopol[10];

int loop1=1,loop2=1,menu,i=1,j,k,icari;

while(loop1==1){

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

printf("======Aplikasi Pelabuhan Kendaraan Masuk Kapal======\n");

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

printf("\t\t Menu\t:\n\n\t1. Tambah Data\n\t2. Update Data\n\t3. Search Data\n\t4. Sort Data\n\t5. Exit\n");

printf("\tMasukkan Kode Menu : ");

fflush(stdin);

scanf("%i",&menu);

switch(menu){

case 1 :{

system("cls");

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

printf("================Tambah Data===================\n");

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

tiket[i]=i;

printf("| Nomor Tiket : %i",tiket[i]);

printf("\n| Masukkan Nomor Kendaraan(Nomor Plat)\t: ");

fflush(stdin); scanf("%i",&nopol[i]);

printf("| Kode Tujuan\t: 1.TRUK, 2. BUS, 3. MOBIL, 4. MOTOR\n");

printf("| Masukkan Jenis Kendaraan\t: ");

fflush(stdin); scanf("%i",&icari);

switch(icari){

Gambar 2 - Sourcecode 2

printf("| Kode Tujuan\t: 1.TRUK, 2. BUS, 3. MOBIL, 4. MOTOR\n");

printf("| Masukkan Jenis Kendaraan\t: ");

fflush(stdin); scanf("%i",&icari);

switch(icari){

case 1:{strcpy(jenisK[i],"TRUK"); printf("| Jenis Kendaraan\t: TRUK"); break;}

case 2:{strcpy(jenisK[i],"BUS"); printf("| Jenis Kendaraan\t: BUS"); break;}

case 3:{strcpy(jenisK[i],"MOBIL"); printf("| Jenis Kendaraan\t: MOBIL"); break;}

case 4:{strcpy(jenisK[i],"MOTOR"); printf("| Jenis Kendaraan\t: MOTOR"); break;}

default:printf("| Kode Jenis Kendaraan Salah!!!"); }

printf("\n| Masukkan Bobot Kendaraan(Ton)\t: ");

fflush(stdin); scanf("%i",&bobot[i]);

printf("| Kode Tujuan\t: 1. KETAPANG, 2. PADANG BAI, 3. LEMBAR, 4. TANJUNG PERAK");

printf("\n| Masukkan Kode Tujuan Pelabuhan\t: ");

fflush(stdin); scanf("%i",&icari);

switch(icari){

case 1:{strcpy(tujuan[i],"KETAPANG"); printf("| Tujuan\t: KETAPANG"); break;}

case 2:{strcpy(tujuan[i],"PADANG BAI"); printf("| Tujuan\t: PADANG BAI"); break;}

case 3:{strcpy(tujuan[i],"LEMBAR"); printf("| Tujuan\t: LEMBAR"); break;}

case 4:{strcpy(tujuan[i],"TANJUNG PERAK");printf("| Tujuan\t: TANJUNG PERAK"); break;}

default:printf("Kode Tujuan Salah!!!"); }

printf("\n| Masukkan Tanggal Tiket(TTBBYYYY)\t: ");

fflush(stdin); scanf("%i",&tglT[i]);

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

i++;getch();

}break;

case 2 :{

system("cls");

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

printf("| Masukkan Nomor Tiket Data Yang Akan Diubah \t: ");

fflush(stdin); scanf("%i",&icari);

printf("=============================================\n\n"); j=0;

j = search(tiket,0,12,icari);

if(j!=0){

Gambar 3 - Sourcecode 3

}break;

case 2 :{

system("cls");

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

printf("| Masukkan Nomor Tiket Data Yang Akan Diubah \t: ");

fflush(stdin); scanf("%i",&icari);

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

j = search(tiket,0,12,icari);

if(j!=0){

printf("| Kode Update Data\t:\n| 1. Ubah Data\n| 2. Delete Data\n");

printf("| Masukkan Update Data \t: ");

fflush(stdin); scanf("%i",&icari);

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

switch(icari){

case 1:{

printf("| Nomor Tiket : %i",tiket[j]);

printf("\n| Masukkan Nomor Kendaraan(Nomor Plat)\t: ");

fflush(stdin); scanf("%i",&nopol[j]);

printf("| Kode Tujuan\t: 1.TRUK, 2. BUS, 3. MOBIL, 4. MOTOR\n");

printf("| Masukkan Jenis Kendaraan\t: ");

fflush(stdin); scanf("%i",&icari);

switch(icari){

case 1:{strcpy(jenisK[j],"TRUK"); printf("| Jenis Kendaraan\t: TRUK"); break;}

case 2:{strcpy(jenisK[j],"BUS"); printf("| Jenis Kendaraan\t: BUS"); break;}

case 3:{strcpy(jenisK[j],"MOBIL"); printf("| Jenis Kendaraan\t: MOBIL"); break;}

case 4:{strcpy(jenisK[j],"MOTOR"); printf("| Jenis Kendaraan\t: MOTOR"); break;}

default:printf("| Kode Jenis Kendaraan Salah!!!"); }

printf("\n| Masukkan Bobot Kendaraan(Ton)\t: ");

fflush(stdin); scanf("%i",&bobot[j]);

printf("| Kode Tujuan\t: 1. KETAPANG, 2. PADANG BAI, 3. LEMBAR, 4. TANJUNG PERAK\n");

printf("| Masukkan Kode Tujuan Pelabuhan\t: ");

fflush(stdin); scanf("%i",&icari);

switch(icari){

case 1:{strcpy(tujuan[j],"KETAPANG"); printf("| Tujuan\t: KETAPANG"); break;}

case 2:{strcpy(tujuan[j],"PADANG BAI"); printf("| Tujuan\t: PADANG BAI"); break;}

case 3:{strcpy(tujuan[j],"LEMBAR"); printf("| Tujuan\t: LEMBAR"); break;}

case 4:{strcpy(tujuan[j],"TANJUNG PERAK");printf("| Tujuan\t: TANJUNG PERAK"); break;}

default:printf("Kode Tujuan Salah!!!"); }

Gambar 4 - Sourcecode 4

4:{strcpy(jenisK[j],"MOTOR"); printf("| Jenis Kendaraan\t: MOTOR"); break;}

default:printf("| Kode Jenis Kendaraan Salah!!!"); }

printf("\n| Masukkan Bobot Kendaraan(Ton)\t: ");

fflush(stdin); scanf("%i",&bobot[j]);

printf("| Kode Tujuan\t: 1. KETAPANG, 2. PADANG BAI, 3. LEMBAR, 4. TANJUNG PERAK\n");

printf("| Masukkan Kode Tujuan Pelabuhan\t: ");

fflush(stdin); scanf("%i",&icari);

switch(icari){

case 1:{strcpy(tujuan[j],"KETAPANG"); printf("| Tujuan\t: KETAPANG"); break;}

case 2:{strcpy(tujuan[j],"PADANG BAI"); printf("| Tujuan\t: PADANG BAI"); break;}

case 3:{strcpy(tujuan[j],"LEMBAR"); printf("| Tujuan\t: LEMBAR"); break;}

case 4:{strcpy(tujuan[j],"TANJUNG PERAK");printf("| Tujuan\t: TANJUNG PERAK"); break;}

default:printf("Kode Tujuan Salah!!!"); }

printf("\n| Masukkan Tanggal Tiket(TTBBYYYY)\t: ");

fflush(stdin); scanf("%i",&tglT[j]); break;}

case 2:{

nopol[j]=0; strcpy(jenisK[j],""); bobot[j]=0; strcpy(tujuan[j],""); tglT[j]=0;

printf("| Data Berhasil Dihapus"); break;}

default:printf("Kode Update Salah!!!"); }

}printf("=============================================\n"); getch();

}break;

case 3 :{

while(loop2==1){

system("cls");

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

printf("| Kode Pencarian\t:\n| 1. Nomor Kendaraan\n| 2. Nomor Tiket\n| 3. Bobot Kendaraan\n");

printf("| Masukkan Kode Pencarian \t: ");

fflush(stdin); scanf("%i",&icari);

printf("=============================================\n\n"); j=0;

switch(icari){

case 1:{

printf("| Masukkan Nomor Kendaraan \t: ");

fflush(stdin); scanf("%i",&icari);

j = search(nopol,0,12,icari);

if(j!=0){print(j,tiket,nopol,jenisK,bobot,tujuan,tglT);} break;}

Gambar 5 - Sourcecode 5

system("cls");

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

printf("| Kode Pencarian\t:\n| 1. Nomor Kendaraan\n| 2. Nomor Tiket\n| 3. Bobot Kendaraan\n");

printf("| Masukkan Kode Pencarian \t: ");

fflush(stdin); scanf("%i",&icari);

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

switch(icari){

case 1:{

printf("| Masukkan Nomor Kendaraan \t: ");

fflush(stdin); scanf("%i",&icari);

j = search(nopol,0,10,icari);

if(j!=0){print(j,tiket,nopol,jenisK,bobot,tujuan,tglT);} break;}

case 2:{

printf("| Masukkan Nomor Tiket \t: ");

fflush(stdin); scanf("%i",&icari);

j = search(tiket,0,10,icari);

if(j!=0){print(j,tiket,nopol,jenisK,bobot,tujuan,tglT);} break;}

case 3:{

printf("| Masukkan Bobot Kendaraan \t: ");

fflush(stdin); scanf("%i",&icari);

j = search(bobot,0,10,icari);

if(j!=0){print(j,tiket,nopol,jenisK,bobot,tujuan,tglT);} break;}

default:printf("Kode Pencarian Salah!!!"); }

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

printf("| Ingin Melakukan Pencarian Lagi(1.YA|2.TIDAK) \t: ");

fflush(stdin); scanf("%i",&loop2);

}

}break;

Gambar 6 - Sourcecode 6

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

printf("| Ingin Melakukan Pencarian Lagi(1.YA|2.TIDAK) \t: ");

fflush(stdin); scanf("%i",&loop2);

}

}break;

case 4 :{

system("cls");

nopol[0]=0;

int max=nopol[0];

for(i = 0; i < sizeof(nopol)/sizeof(nopol[0]); i++ ){

if(nopol[i] > max){

max = nopol[i];

}

}

// printf("Max = %d ", max);

bucketSort(nopol,max);

for( i=0;i<sizeof(nopol)/sizeof(nopol[0]);i++){

printf("%d ",nopol[i]);

}

}break;

case 5 :{

loop1=0;

}break;

default : printf("\n| Kode Menu Salah!!!"); sleep(3);

}

system("cls");

}

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

printf("===============Terima Kasih==================\n");

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

return 0;

}

Gambar 7 - Sourcecode 7

void print(int n,int tiket[10],int nopol[10],char jenisK[10][12],int bobot[10],char tujuan[10][12],int tglT[10]){

printf("| Nomor Tiket\t\t: %i",tiket[n]);

printf("\n| Nomor Kendaraan\t: %i",nopol[n]);

printf("\n| Jenis Kendaraan\t: %s",jenisK[n]);

printf("\n| Bobot Kendaraan\t: %i",bobot[n]);

printf("\n| tujuan Pelabuhan\t: %s",tujuan[n]);

printf("\n| Tanggal Tiket\t\t: %i",tglT[n]);

}

int search(int array[10], int m, int n, int o){

if(m<=n){

if(array[m]==o){

return m;

}else{

m++;

return search(array, m, n, o); }

}else{

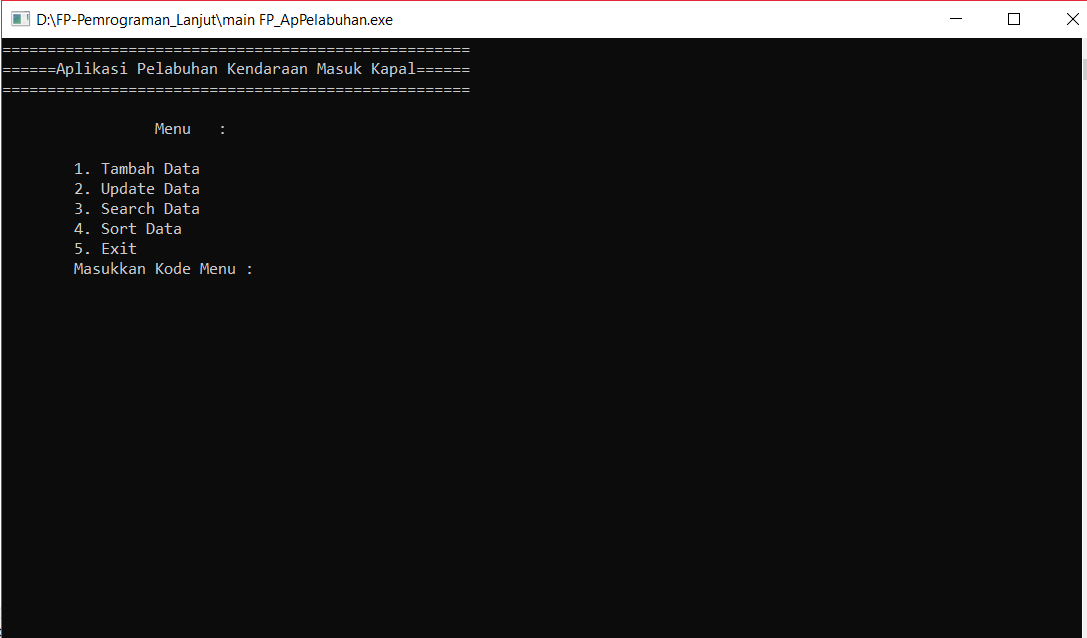
printf("| Data Tidak Ditemukan !!!");

return 0;

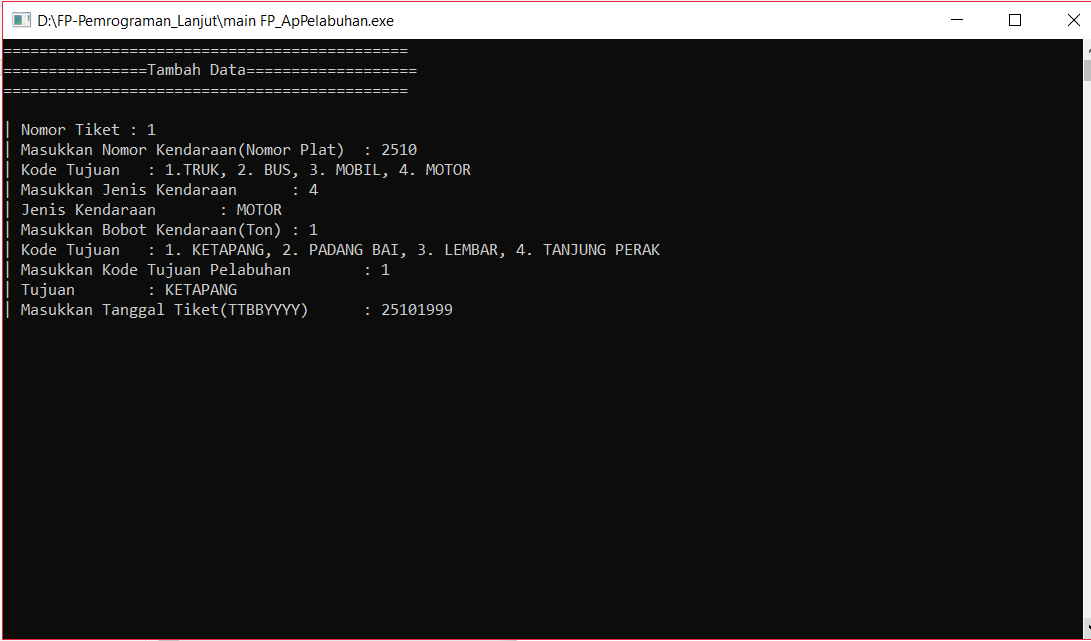
}

Gambar 8 - Sourcecode 8

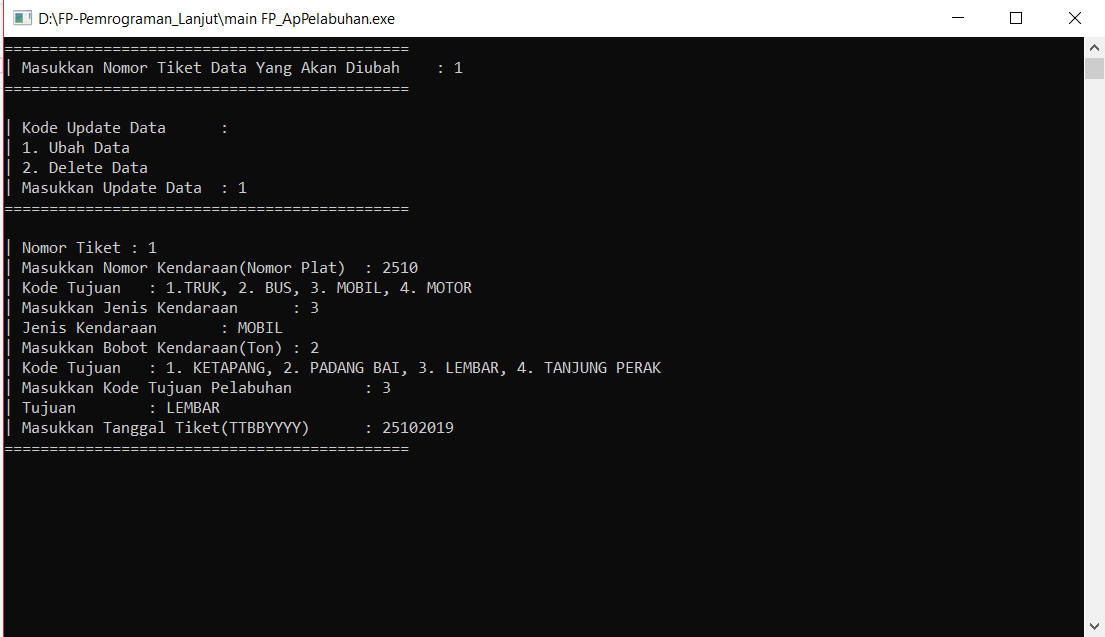
**SCREENSHOOT HASIL APLIKASI**



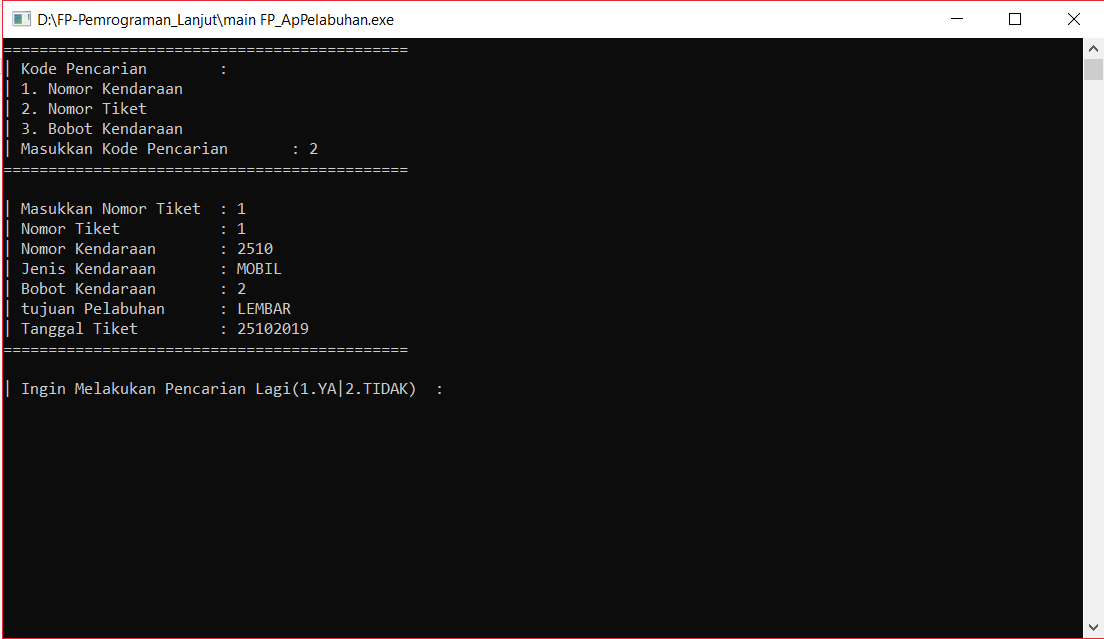
Gambar 9 - Menu utama



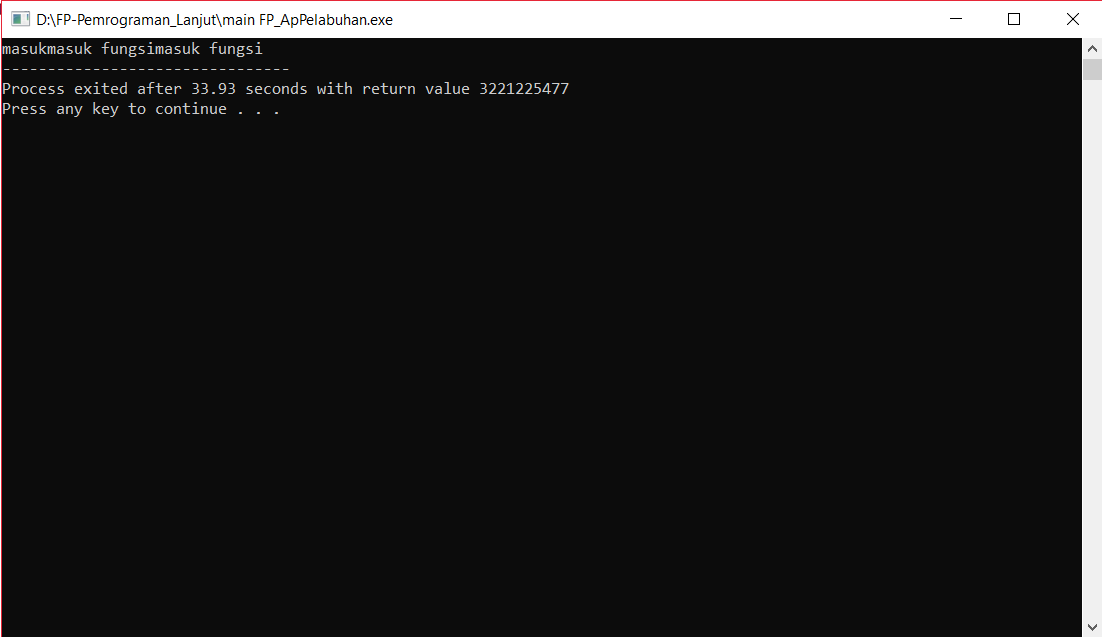
Gambar 10 - menu input 1 menambah data



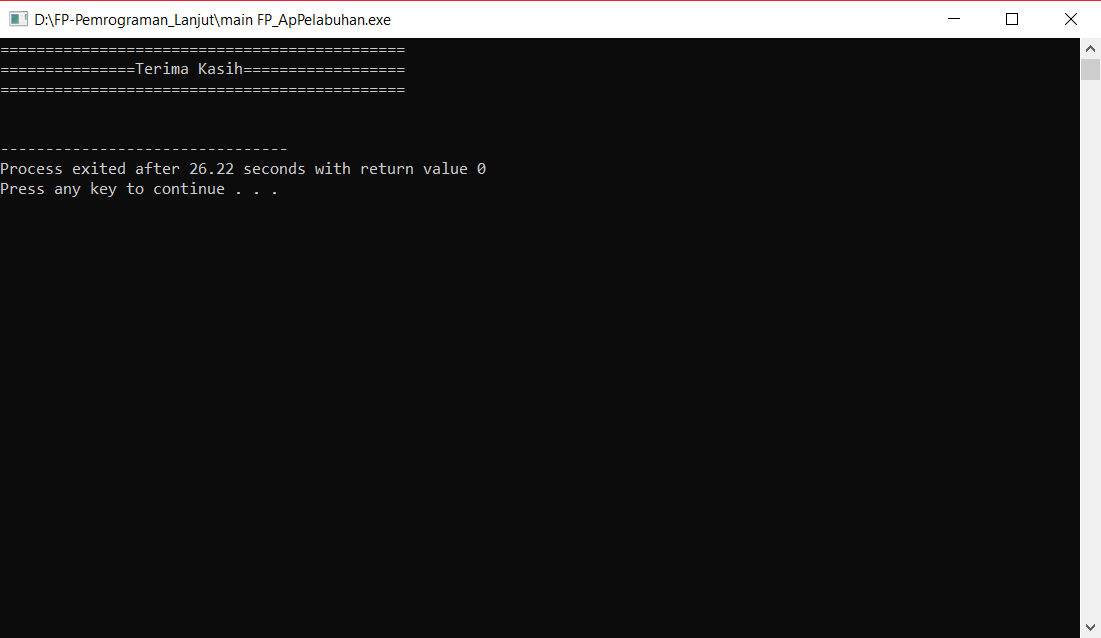
Gambar 11 - Menu input 2 update data



Gambar 12 - Menu input 3 search data



Gambar 13 - Sort data



Gambar 14 - Exit Program