Program Utama tampilan awal

#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <time.h>

Pada library yang dipakai di program ini ada 4.

#include <stdio.h>

Sebagai standar input output program yang merupakan standar library wajib dalam bahasa C

#include <stdlib.h>

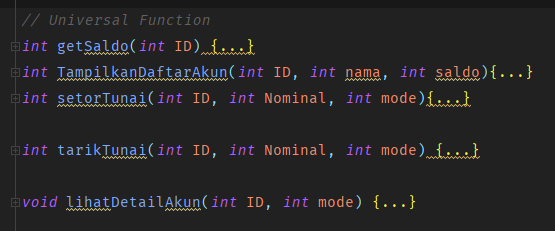
Sebagai bagian untuk kebutuhan library tambahan

#include <string.h>

Sebagai manipulasi string untuk memodifikasi beberapa input string

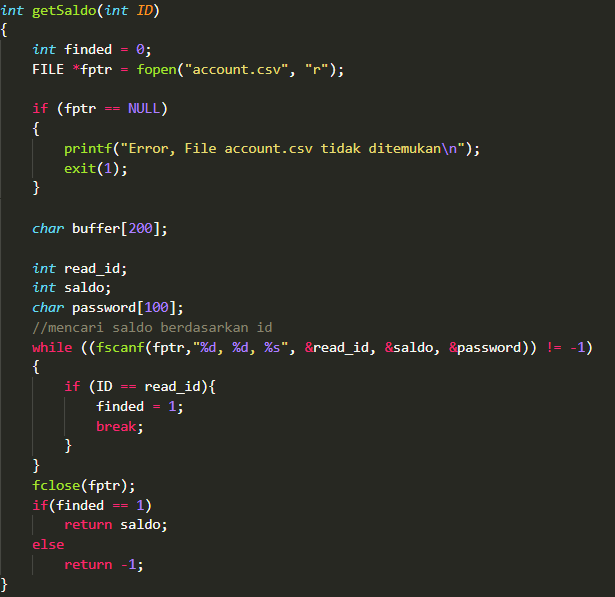
#include <time.h>

Sebagai pembaca waktu ketika transaksi terjadi



Berikut ini adalah universal function, diberikan nama seperti itu karena fungsi ini bisa dipakai untuk kedua mode, yaitu mode user dan mode admin. Yang akan dijelaskan lebih detail pada penjelasan berikutnya

Fungsi getSaldo()



Fungsi getSaldo merupakan fungsi universal yang sesuai dengan namanya untuk membaca saldo dari user bersangkutan pada database file csv. Fungsi getSaldo ini mendapatkan sebuah parameter berupa ID dari user. File account.csv akan dibuka kemudian setiap akun yang ada akan dibaca. Jika ID dari akun yang dicari ditemukan maka fungsi ini akan mengembalikan saldo dari ID tersebut, sedangkan jika tidak ada ID yang cocok dengann ID yang dicari maka akan dikembalikan nilai -1 yang berarti bahwa akun tidak ditemukan.

Fungsi TampilkanDaftarAkun()

*int* TampilkanDaftarAkun(*int* ID, *int* nama, *int* saldo){  
 FILE \* fpointer, \*fp;  
 fpointer = fopen("account.csv", "r");  
  
 *char* dumpCsv[100];  
 *char* dumpTxt[100];  
 *char* fileNameTXT[15];  
 *char* IdLog[15];  
 *char* saldoLog[100];  
 *char* \*token;  
 *int* cek = 0;  
 *while* (fgets(dumpCsv, *sizeof*(dumpCsv), fpointer)){  
 cek++;  
 }  
 *if*(cek == 1){  
 printf("Tidak ada akun yang terdaftar\n");  
 *return* 1;  
 }  
 fclose(fpointer);  
 fpointer = fopen("account.csv", "r");  
 fgets(dumpCsv, *sizeof*(dumpCsv), fpointer);  
 *while* (fgets(dumpCsv, *sizeof*(dumpCsv), fpointer)){  
 token = strtok(dumpCsv, ",");  
 strcpy(IdLog, token);  
 token = strtok(**NULL**, ",");  
 strcpy(saldoLog, token);  
  
 *if* (ID == 1){  
 printf(" ID: %s\n", IdLog);  
 }  
  
 strcpy(fileNameTXT,IdLog);  
 strcat(fileNameTXT, ".txt");  
 fp = fopen(fileNameTXT, "r");  
 fscanf(fp, "%[^\n]", dumpTxt);  
 *if* (nama == 1){  
 printf(" %s\n", strtok(dumpTxt, " "));  
 }  
  
 *if* (saldo == 1){  
 printf(" Saldo: %s\n", saldoLog);  
 }  
 printf("\n");  
 }  
 fclose(fp);  
 fclose(fpointer);  
 printf("-----------------------------\n");  
 *return* 0;  
}

Pada fungsi tampilkan daftar akun ini bertujuan agar kita bisa melihat daftar-daftar akun yang ada dalam database, dalam menampilkan datanya ada tiga, yaitu ID, nama, dan saldo. Tidak semua informasi ditampilkan ke layar, kita bisa menampilkan namanya saja, dan lain-lain. Konsepnnya yaitu memakai switch menggunakan parameter. Jika parameter ID bernilai satu, maka ID akan ditampilkan, jika nama bernilai 1, maka nama akan ditampilkan, dan jika saldo bernilai 1, maka saldo akan ditampilkan. Apabila kita ingin menampilkan semua datanya kita hanya perlu membuat parameter tersebut bernilai 1 semua.

Fungsi setorTunai()

*int* setorTunai(*int* ID, *int* Nominal, *int* mode){ *// PAKAI INI UNTUK SETOR TUNAI  
 //mode : 0 = setor tunai, 1 = transfer masuk  
 int* nominalDump;  
 *char* fileTXT[15];  
  
 itoa(ID, fileTXT, 10);  
 strcat(fileTXT, ".txt");  
  
 FILE \*fpointer, \*fp;  
 fpointer = fopen("account.csv", "r");  
 fp = fopen("temp.csv", "w");  
  
 *char* dump[100], dump2[100], charNominal[100];  
 *char* \*token;  
 fgets(dump, *sizeof*(dump), fpointer);  
 fprintf(fp, dump);  
 *while* (fgets(dump, *sizeof*(dump), fpointer)){  
 strcpy(dump2, dump);  
 token = strtok(dump, ",");  
 *if* (atoi(token) == ID){  
 fprintf(fp, token);  
 fprintf(fp, ",");  
 token = strtok(**NULL**, ",");  
 nominalDump = atoi(token);  
 nominalDump = nominalDump + Nominal;  
 itoa(nominalDump, charNominal, 10);  
 fprintf(fp, charNominal);  
 fprintf(fp, ",");  
 token = strtok(**NULL**, ",");  
 fprintf(fp, token);  
 }*else*{  
 fprintf(fp, dump2);  
 }  
 }  
 fclose(fp);  
 fclose(fpointer);  
  
 fpointer = fopen("temp.csv", "r");  
 fp = fopen("account.csv", "w");  
  
 *while* (fgets(dump, *sizeof*(dump), fpointer)){  
 fprintf(fp, dump);  
 }  
  
 fclose(fp);  
 fclose(fpointer);  
 remove("temp.csv");  
  
 *char* tgl[20], waktu[20];  
  
 time\_t t = time(**NULL**);  
 *struct* tm tm = \*localtime(&t);  
 *char* Bulan[10],Hari[10],Jam[10],Menit[10],Detik[10];  
 itoa(tm.tm\_year + 1900, tgl,10); strcat(tgl, "/");  
 itoa(tm.tm\_mon + 1, Bulan, 10);  
 itoa(tm.tm\_mday, Hari, 10);  
 strcat(tgl,Bulan); strcat(tgl, "/");  
 strcat(tgl,Hari);  
  
 itoa(tm.tm\_hour, waktu, 10); strcat(waktu, ":");  
 itoa(tm.tm\_min, Menit, 10);  
 itoa(tm.tm\_sec, Detik, 10);  
 strcat(waktu,Menit); strcat(waktu, ":");  
 strcat(waktu,Detik);  
  
  
 fpointer = fopen(fileTXT, "a");  
 *if*(mode == 0)  
 fprintf(fpointer, "[+] SETOR TUNAI\n");  
 *else if*(mode == 1)  
 fprintf(fpointer, "[+] TRANSFER MASUK\n");  
  
 fprintf(fpointer, " TGL: %s\n", tgl);  
  
 fprintf(fpointer, " PUKUL: %s\n", waktu);  
 fprintf(fpointer, " NOMINAL: %d\n\n", Nominal);  
  
 fclose(fpointer);  
}

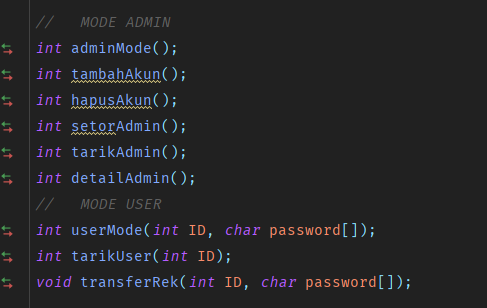
Fungsi setor tunai bertujuan untuk mengirim uang ke dalam akun. Konsepnya dalam fungsi ini membutuhkan tiga parameter untuk dijalankan yaitu ID, nominal dan mode. Untuk mode ini digunakan untuk identitas fungsi digunakan oleh mode apa, mode 0 adalah untuk setor tunai sedangkan mode 1 adalah setor tunai untuk transfer.

Fungsi ini sama seperti fungsi Tarik tunai. Hanya saja yang membedakan, fungsi Tarik tunai untuk mengurangi nominal yang ada dalam akun tersebut.

Fungsi lihatDetailAkun()

*void* lihatDetailAkun(*int* ID, *int* mode)  
{  
*// system("cls");  
 char* buff[100], txtFile[15];  
 itoa(ID, txtFile, 10);  
 strcat(txtFile, ".txt");  
 *//membuka akun file txt* FILE \*fptr = fopen(txtFile, "r");  
  
 *if* (fptr == **NULL**)  
 {  
 printf("Error, file %d.txt tidak ditemukan\n", ID);  
 exit(1);  
 }  
  
 printf(" PROGRAM KAS KELAS\n"  
 " LIHAT DETAIL AKUN\n");  
 *if* (mode == 0)  
 printf(" --ADMIN--\n");  
 *else* printf(" --USER--\n");  
 printf("INFORMASI AKUN :\n");  
  
 printf(" ID : %d\n", ID);  
 *for* (*int* i = 0; i < 4; i++)  
 {  
 fgets(buff, *sizeof*(buff), fptr);  
 printf("%s", buff);  
 }  
 *//mencari saldo id bersangkutan di account.csv* printf("Saldo : %d\n", getSaldo(ID));  
 *//mencetak keseluruhan isi file akun txt  
 while* (fgets(buff, *sizeof*(buff), fptr))  
 printf("%s", buff);  
  
 printf("\n----------------------------\n");  
 system("pause");  
 fclose(fptr);  
}

Fungsi lihat detail akun ini bertujuan untuk membaca file txt akun dan menampilkannya ke layar. Karna fungsi ini hanya untuk menampilkan saja, maka tidak ada mode write pada operasi filenya. Fungsi ini akan mencetak semua isi data dalam file txt akun tersebut.



Disini adalah beberapa fungsi yang dipakai untuk kedua mode, jadi program akan berpindah ke dalam fungsi-fungsi tersebut ketika memilih menu yang sesuai dengan fungsinya.

Fungsi main

*int* main(){  
 *int* pilih, ID;  
 *char* password[100];  
 *char* \*token;  
  
 printf(" PROGRAM KAS KELAS \n");  
 printf("1. Admin\n");  
 printf("2. User\n");  
 printf("3. Keluar Program\n");  
 *while* (1){  
 printf("Pilih Menu:");  
 scanf("%d", &pilih);  
  
 *if* (pilih == 1){ *// 1. Admin  
 while* (1){  
 printf("Masukan ID:");  
 scanf("%d", &ID);  
 fflush(**stdin**);  
 printf("Masukan password:");  
 fgets(password, *sizeof*(password), **stdin**);  
  
 FILE \*fpointer;  
 fpointer = fopen("account.csv", "r");  
  
 *char* akun[100];  
 *while* (fgets(akun, *sizeof*(akun), fpointer)){  
 token = strtok(akun, ",");  
 *while* (token != **NULL**){  
 *if* ((ID == atoi(token)) && (ID / 100000 == 10)){  
 token = strtok(**NULL**, ",");  
 token = strtok(**NULL**, ",");  
 *if*(strcmp(token, password) == 0){  
 fclose(fpointer);  
 adminMode();  
 *return* 0;  
 }  
 }  
 token = strtok(**NULL**, ",");  
 }  
 }  
 *if* (token == **NULL**){  
 printf("ID atau Password salah\n");  
 }  
 }  
 }  
 *else if* (pilih == 2){ *// 2. User  
 while* (1){  
 printf("Masukan ID:");  
 scanf("%d", &ID);  
 fflush(**stdin**);  
 printf("Masukan password:");  
 fgets(password, *sizeof*(password), **stdin**);  
  
 FILE \*fpointer;  
 fpointer = fopen("account.csv", "r");  
  
 *char* akun[100];  
 *while* (fgets(akun, *sizeof*(akun), fpointer)){  
 token = strtok(akun, ",");  
 *while* (token != **NULL**){  
 *if* ((ID == atoi(token)) && (ID / 100000 == 20)){  
 token = strtok(**NULL**, ",");  
 token = strtok(**NULL**, ",");  
 *if*(strcmp(token, password) == 0){  
 fclose(fpointer);  
 userMode(ID, password);  
 *return* 0;  
 }  
 }  
 token = strtok(**NULL**, ",");  
 }  
 }  
 *if* (token == **NULL**){  
 printf("ID atau Password salah\n");  
 }  
 }  
 }  
 *else if* (pilih == 3){  
 exit(0);  
 } *else*{  
 printf("Input tidak ada dalam menu\n");  
 }  
 }  
 *return* 0;  
}

Pada fungsi main itu sendiri ketika program dijalankan pertama kali, ia akan meminta user untuk login, tetapi sebelum itu program akan menyuruh pengguna untuk memilih mode apa yang akan dijalankan, apakah itu user mode atau admin mode, perbedaannya adalah bahwa user mode memiliki keterbatasan fitur, sedangkan untuk admin mode memiliki banyak fitur. Jika pengguna memilih untuk masuk mode admin, maka program akan menyuruh pengguna untuk memasukan ID admin dan password admin, secara default ID dan password admin adalah, ID:1012345 dan pass: admin. Apabila salah aka nada peringatan bahwa ID atau password salah, begitupun dengan login user mode. Untuk melakukan pengecekan berdasarkan ID dahulu, untuk admin dan user dibedakan melalui dua angka didepanya, admin yaitu 10, user yaitu 20. Program akan membaca file csv secara berurutan dan mengambilan kolom pertama untuk ID dan kolom ketiga sebagai password akun tersebut. Apabila benar, program akan masuk ke fungsi admin/user sesuai dengan pilihan mode.

Admin Mode

*int* adminMode(){  
 *int* pilih;  
 *while* (1){  
 printf(" PROGRAM KAS KELAS\n"  
 " --ADMIN--\n"  
 "1. TAMBAH AKUN\n"  
 "2. HAPUS AKUN\n"  
 "3. SETOR TUNAI\n"  
 "4. TARIK TUNAI\n"  
 "5. LIHAT DETAIL AKUN\n"  
 "6. exit\n");  
 printf("pilih:");  
 scanf("%d", &pilih);  
  
 *switch* (pilih) {  
 *case* 1:  
 tambahAkun();  
 *break*;  
 *case* 2:  
 hapusAkun();  
 *break*;  
 *case* 3:  
 setorAdmin();  
 *break*;  
 *case* 4:  
 tarikAdmin();  
 *break*;  
 *case* 5:  
 detailAdmin();  
 *break*;  
 *case* 6:  
 exit(0);  
 *default*:  
 printf("maaf tidak ada dalam menu\n");  
 }  
 }  
 *return* 0;  
}

Pada code di sini, program admin mode akan menampilkannya sebagai menu admin dan pengguna bisa memilih beberapa menu yang bisa dioperasikan sebagai admin. Seperti yang dikatakan di awal bahwa setiap menu akan masuk ke dalam fungsi-fungsi yang sudah dibuat sesuai kategorinya, di sini fungsi-fungsi yang dipakai sebagai admin mode adalah tambahAkun(), setorAdmin(), tarikAdmin(), detailAdmin(), dan exit(). Apabila user salah menginput menu, maka program akan meminta ulang inputan yang benar.

Fungsi tambahAkun()

*int* tambahAkun(){  
 *int* umur, pilKelamin, ID;  
 *char* nama[50], tanggal[12], jenisKelamin[12], password[100];  
 printf(" PROGRAM KAS KELAS\n"  
 " --ADMIN--\n"  
 " TAMBAH AKUN\n");  
 printf("NAMA:");  
 fflush(**stdin**);  
 scanf("%s", nama);  
 fflush(**stdin**);  
 printf("UMUR:");  
 scanf("%d", &umur);  
 fflush(**stdin**);  
 printf("TANGGAL LAHIR (dd/mm/yy):");  
 fgets(tanggal, *sizeof*(tanggal), **stdin**);  
 fflush(**stdin**);  
 printf("JENIS KELAMIN:\n");  
 printf("1. Laki-laki | 2. Perempuan\n");  
 printf("pilih:");  
 scanf("%d", &pilKelamin);  
 *if*(pilKelamin == 1){  
 strcpy(jenisKelamin, "Laki-laki");  
 }  
 *else if* (pilKelamin = 2){  
 strcpy(jenisKelamin, "Perempuan");  
 }  
 printf("------------------------------------\n");  
 srand(time(0));  
 ID = (10\*200000)+(rand()%100000);  
 printf("ID: %d (ingat ID ini baik-baik)\n", ID);  
 printf("Masukan password:");  
 fflush(**stdin**);  
 fgets(password, *sizeof*(password), **stdin**);  
 *int* passRepeat[100];  
 *while* (1){  
 printf("Masukan kembali password:");  
 fgets(passRepeat, *sizeof*(passRepeat), **stdin**);  
 *if* (strcmp(passRepeat, password)==0){  
 printf("Pembuatan akun berhasil\n");  
 *break*;  
 }*else*{  
 printf("Password tidak sesuai dengan sebelumnya, masukan kembali\n");  
 }  
 }  
  
 *//Write to database BEGIN  
 char* txtFile[15];  
 itoa(ID, txtFile, 10);  
 strcat(txtFile, ".txt");  
  
 FILE \*fpointer;  
 fpointer = fopen(txtFile, "w");  
 fprintf(fpointer, " NAMA:%s\n"  
 " UMUR:%d\n"  
 " TANGGAL LAHIR:%s"  
 " KELAMIN:%s\n"  
 "------History Transaksi------\n",nama, umur, tanggal, jenisKelamin);  
 fclose(fpointer);  
  
 fpointer = fopen("account.csv", "a");  
 fprintf(fpointer, "%d,0,%s", ID, password);  
 fclose(fpointer);  
 *//Write to database END* system("pause");  
 *return* 0;  
}

Fungsi tambahAkun() adalah fungsi yang hanya dimiliki oleh mode admin. Di sini program akan menyimpan data-data ke dalam variabel yang independent, yaitu ada variabel

*int* umur, pilKelamin, ID;  
*char* nama[50], tanggal[12], jenisKelamin[12], password[100];

untuk nama, tanggal, jenisKelamin, dan umur akan ditulis ke dalam file .txt dan ID dan password akan dimasukan ke dalam file csv

Program akan meminta user pertama kali untuk menginputkan Nama, kemudian dilanjutkan dengan umur, tanggal lahir, jenis kelamin (untuk jenis kelamin hanya diberikan pilihan 1.Laki-laki. 2.Perempuan), dan terakhir adalah password yang diinputkan dua kali sebagai konfirmasi. ID tidak ditentukan oleh admin, tetapi akan digenerate oleh sistem di mana algoritma generatenya adalah di sini menggunakan random yang di modulus dengan 100000 untuk angka belakang dan ditambahkan dengan 2000000 sebagai angka depan 20 untuk ID user.

Disini program ketika user selesai menginputkan password, maka semua data yang sudah disimpan pada variabel akan dituliskan menggunakan operasi file yang pertama adalah menuliskan ke dalam file .txt dan nama filenya adalah ID user. Dan terakhir program akan menuliskan ID dan password ke dalam file csv, untuk saldo akan dituliskan 0 pada kolom kedua sebagai nominal default.

Fungsi hapusAkun()

*int* hapusAkun(){  
 *int* pilID;  
 printf(" PROGRAM KAS KELAS\n"  
 " --ADMIN--\n"  
 " HAPUS AKUN\n"  
 "AKUN YANG TERDAFTAR\n");  
 *if* (TampilkanDaftarAkun(1,1,0))  
 {  
 *return* 0;  
 }*else*{  
 printf("PILIH ID:");  
 scanf("%d", &pilID);  
 }  
  
 FILE \*fpointer, \*fp;  
 fpointer = fopen("account.csv", "r");  
 fp = fopen("temp.csv", "w");  
  
 *char* dump[50];  
 *char* dump2[50];  
 *char* \*token;  
 fgets(dump, *sizeof*(dump), fpointer);  
 fprintf(fp,dump);  
 token = strtok(dump, ",");  
 *while* (fgets(dump, *sizeof*(dump), fpointer)){  
 strcpy(dump2, dump);  
 token = strtok(dump, ",");  
  
 *if* (atoi(token) != pilID){  
 fprintf(fp, dump2);  
 }  
  
 }  
 fclose(fp);  
 fclose(fpointer);  
  
 fpointer = fopen("temp.csv", "r");  
 fp = fopen("account.csv", "w");  
  
 *while* (fgets(dump, *sizeof*(dump), fpointer)){  
 fprintf(fp, dump);  
 }  
  
 fclose(fp);  
 fclose(fpointer);  
  
 remove("temp.csv");  
 *char* fileTarget[15];  
 itoa(pilID, fileTarget, 10);  
 strcat(fileTarget, ".txt");  
 remove(fileTarget);  
  
 printf("HAPUS AKUN BERHASIL\n");  
 system("pause");  
}

Fungsi hapusAkun() di sini juga adalah fungsi yang khusus dimiliki oleh mode admin.

Pada code ini program akan menampilkan semua daftar akun yang ada atau terdaftar, yang ditampilkan oleh fungsi univerisal yaitu fungsi

TampilkanDaftarAkun(1,1,0)

Yang akan ditampilkan di sini adalah semua akun user, admin tidak termasuk

Namun jika tidak ada akun user yang terdaftar atau kosong, maka program tidak akan bisa melakukan hapus akun, akan langsung Kembali (return) ke menu admin mode.

Jika ada program akan peminta pengguna untuk memasukan ID akun. Pencarian akun di sini menggunaka algoritma secara linear, artinya akan dicari satu persatu dari yang pertama sampai akhir akun yang terdaftar. Jika ditemukan maka program akan langsung menghapusnya. Untuk operasi filenya logikanya kita membuat sebuah file csv Salinan yang diberi nama dump.csv untuk menyalin semua data csv yang original. Kemudian file csv yang original akan ditulis ulang berdasarkan data di dump.csv kecuali akun yang dihapus.

Fungsi setorAdmin()

*int* setorAdmin(){  
 *int* plhID, nominal;  
 printf(" PROGRAM KAS KELAS\n"  
 " --ADMIN--\n"  
 " SETOR TUNAI\n"  
 "AKUN YANG TERDAFTAR\n");  
 *if* (TampilkanDaftarAkun(1,1,1))  
 {  
 *return* 0;  
 }*else*{  
 printf("PILIH ID:");  
 scanf("%d", &plhID);  
 }  
  
 FILE \*fpointer;  
 fpointer = fopen("account.csv", "r");  
  
 *char* dump[100];  
 *char* \*token;  
 *while* (fgets(dump, *sizeof*(dump), fpointer)){  
 token = strtok(dump, ",");  
 *if* (atoi(token) == plhID){  
 printf("Masukan nominal:");  
 scanf("%d", &nominal);  
 setorTunai(plhID, nominal, 0);  
 printf("Setor tunai berhasil dilakukan\n");  
 system("pause");  
 fclose(fpointer);  
 *return* 0;  
 }  
 }  
  
 fclose(fpointer);  
 printf("Akun tidak ditemukan!!\n");  
 system("pause");  
}

Pada code ini, pertama sama dengan fungsi hapus akun, yaitu program akan menampilkan daftar akun yang ada, jika tidak ada akun user yang ada maka setor tunai tidak akan bisa dilakukan dan akan langsung return.

Jika ada, program akan meminta pengguana untuk input ID akun yang akan disetorkan. Pada pencarian akan dilakukan seperti hapus akun, yaitu pencarian secara linear. Jika ditemukan, maka program akan mengambil kolom kedua pada akun tersebut yang merupakan nominal, di awal program melakukan read untuk mengecek ID yang dicek di fungsi if sampai pada bari akhir file csv. Ketika sudah diapatkan, program akan menyuruh pengguan untuk input nominal berapa yang akan ditambahkan. Sesudah itu fungsi setor tunai akan dipanggil di mana fungsi ini adalah fungsi universal, data yang diminta sebagai parameter fungsi itu adalah ID dan nominal.

Fungsi tarikAdmin()

*int* tarikAdmin()  
{  
 *int* plhID, nominal;  
 printf(" PROGRAM KAS KELAS\n"  
 " --ADMIN--\n"  
 " TARIK TUNAI\n"  
 "AKUN YANG TERDAFTAR\n");  
 *if* (TampilkanDaftarAkun(1,1,1))  
 {  
 *return* 0;  
 }*else*{  
 printf("PILIH ID:");  
 scanf("%d", &plhID);  
 }  
  
 FILE \*fpointer;  
 fpointer = fopen("account.csv", "r");  
  
 *char* dump[100];  
 *char* \*token;  
 *while* (fgets(dump, *sizeof*(dump), fpointer))  
 {  
 token = strtok(dump, ",");  
 *if* (atoi(token) == plhID)  
 {  
 printf("Masukan nominal:");  
 scanf("%d", &nominal);  
 *if* (tarikTunai(plhID, nominal, 0))  
 {  
 printf("Tarik tunai berhasil dilakukan\n");  
 }  
 *else* {  
 printf("Tarik tunai gagal karena saldo tidak mencukupi\n");  
 }  
 system("pause");  
 fclose(fpointer);  
 *return* 0;  
 }  
 }  
  
 fclose(fpointer);  
 printf("Akun tidak ditemukan!!\n");  
 system("pause");  
}

Pada code, program akan menampilkan daftar akun user yang akan dilakukan Tarik tunai, apabila tidak ada maka Tarik tunai tidak bisa dilakukan.

Algoritmanya hampir mirip dengan fungsi setor tunai hanya yang membedakannya adalah pada fungsi Tarik tunai pada I kondisional if, di sana program akan memanggil fungsi universal tarikTunai(); jika fungsi itu berhasil dilakukan makan akan mereturn hasil 1 (true) dan akan printf bahwa Tarik tunai berhasil dilakukan, tetapi apabila nominalnya tidak mencukupi, fungsi Tarik tunai akan meretun 0 (false) dan diberitahukan bahwa transaksi gagal dilakukan.

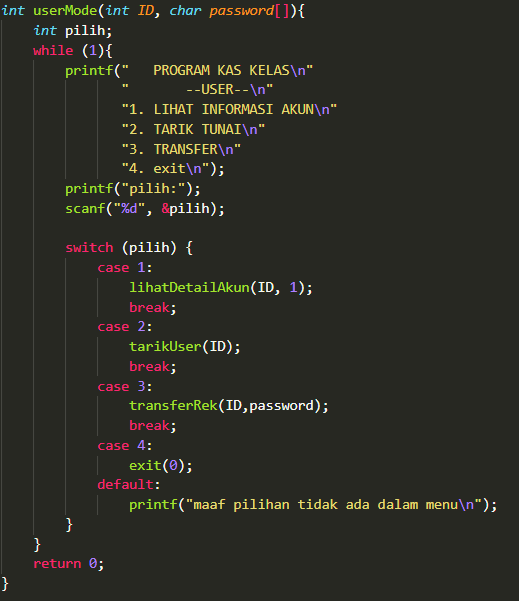
Fungsi detailAdmin()

*int* detailAdmin(){  
 *int* plhID, nominal;  
 printf(" PROGRAM KAS KELAS\n"  
 " --ADMIN--\n"  
 " TARIK TUNAI\n"  
 "AKUN YANG TERDAFTAR\n");  
 *if* (TampilkanDaftarAkun(1,1,0))  
 {  
 *return* 0;  
 }*else*{  
 printf("PILIH ID:");  
 scanf("%d", &plhID);  
 }  
 lihatDetailAkun(plhID, 0);  
}

Di sini program akan pertama kali akan menampilkan daftar akun yang akan dilihat detail data-data akun user tersebut, apabila tidak ada akun user sama sekali, maka program akan return Kembali ke menu admin mode.

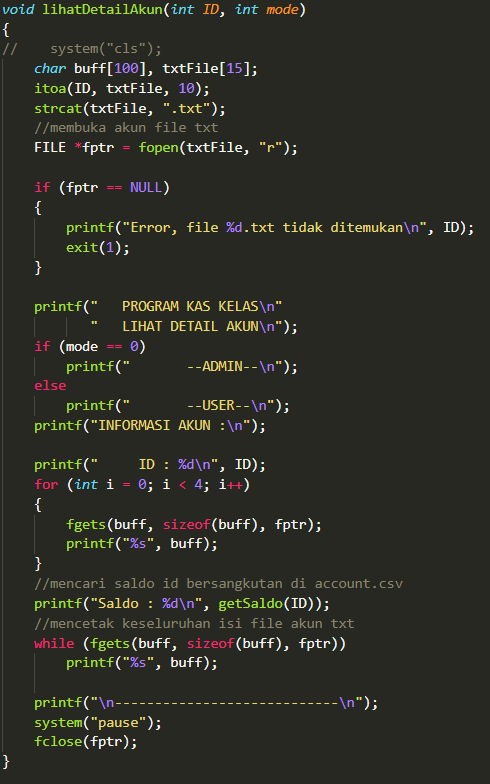
Namun jika ada, program meminta pengguna untuk menginputkan ID akun apa yang ingin dilihat. Lihat detail akun akan dilakukan oleh fungsi universal yaitu fungsi lihatDetailAkun(); dan ada dua parameter untuk dikirim sebagai data yang akan dipakai dalam fungsi tersebut, yaitu ID dan mode yang merupakan Boolean/integer sebagai switch dimana 0 untuk admin mode, 1 untuk user mode.

User Mode



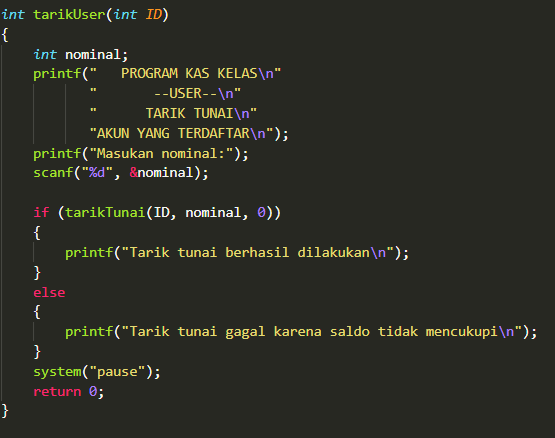
Pada fungsi user mode, user akan disajikan daftar menu yang dapat diakses sebagai seorang user. User dapat memilih dari menu 1 hingga menu 4, jika menu yang dimasukkan tidak tersedia maka user akan mendapat pesan bahwa menu yang dimasukkan tidak tersedia. Masing-masing menu didukung oleh sebuah fungsi yang akan menjalankan tugas tertentu sesuai dengan menu yang dipilih. Fungsi tersebut yaitu fungsi lihatDetailAkun(), tarikUser(), transferRek(), dan exit() untuk keluar.

Fungsi lihatDetailAkun()



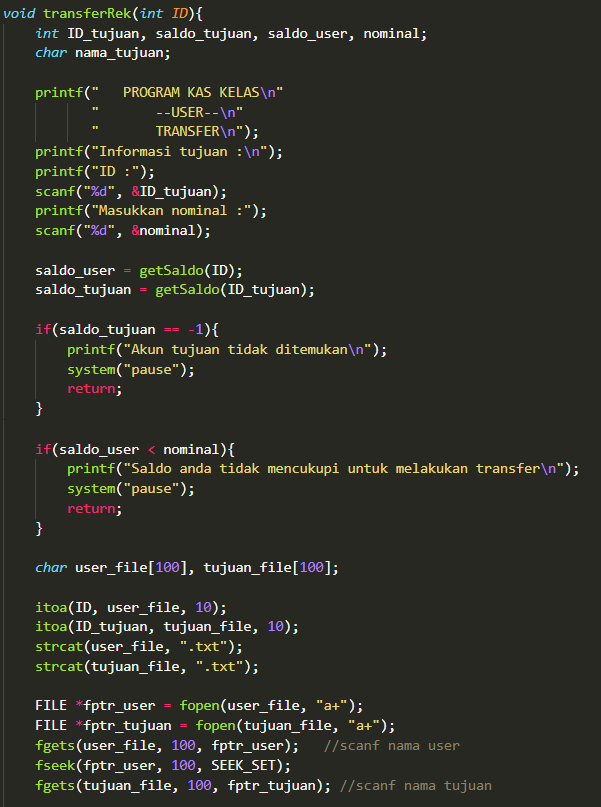
Fungsi lihatDetailAkun memiliki parameter yang diterima melalui call by value yaitu ID dari user dan sebuah mode. Mode disini hanya digunakan untuk mengatur tampilan karena fungsi ini merupakan fungsi yang akan digunakan pada user mode dan admin mode. File dari user bersangkutan akan dibuka. Jika file tidak ada maka akan ditampilkan pesan bahwa file tidak ditemukan. Jika file berhasil dibuka maka seluruh isi dari file tersebut yang berisi detail identitas user dan histori transaksi yang terjadi akan ditampilkan secara keseluruhan.

Fungsi tarikUser()



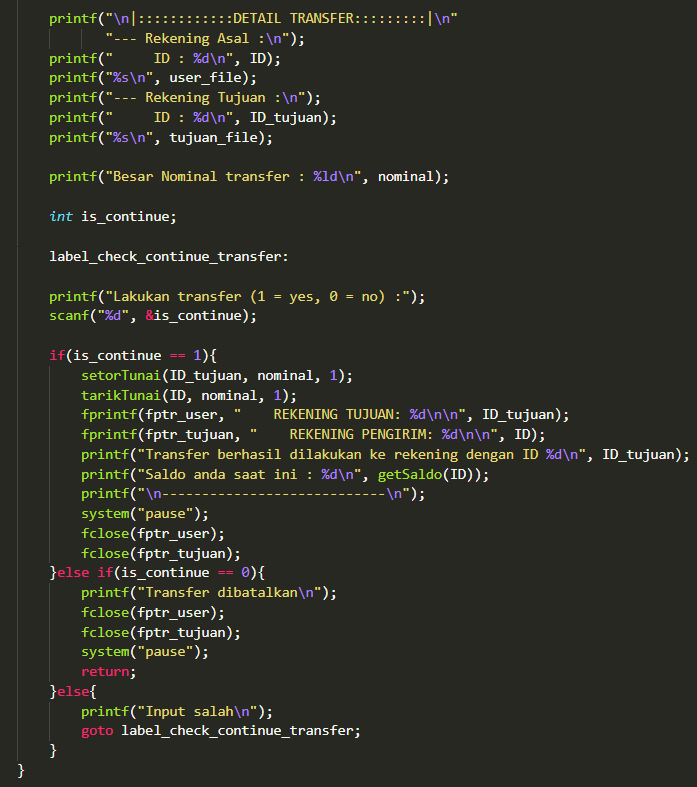
Fungsi tarikUser dideklarasikan sebagai fungsi yang menerima parameter dengan call by value. Parameter yang diterima berupa ID dari user. Pada fungsi ini terlebih dahulu user akan diminta untuk memasukkan besarnya nominal penarikan yang akan dilakukan. Fungsi universal tarikTunai akan dipanggil dengan call by value antara lain ID, nominal, dan sebuah mode yaitu 0 sebagai tanda bahwa akan dilakukan tarik tunai. Mode ini diperlukan karena fungsi universal tarikTunai juga digunakan untuk melakukan transfer.

Fungsi trasferRek()

­­ 

Menu transfer merupakan fasilitas khusus yang diberikan untuk user yang memungkinkan user melakukan transfer ke akun lain yang terdaftar. Parameter yang diterima oleh fungsi ini yaitu ID dari user. User akan diminta untuk memasukkan informasi tujuan transfer dimana berupa ID dari akun tujuan. Selanjutnya user juga akan diminta untuk memasukkan nominal besarnya transfer yang ingin dilakukan. Saldo user dan saldo id tujuan akan dibaca terlebih dahulu menggunakan fungsi universal getSaldo().

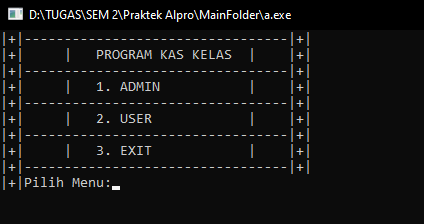
Jika fungsi getSaldo mengembalikan nilai -1 berarti akun tujuan tidak ditemukan dan user akan nmendapat pesan bahwa akun tidak ditemukan. Jika saldo user kurang dari nominal transfer maka pesan saldo tidak mencukupi akan ditampilkan.



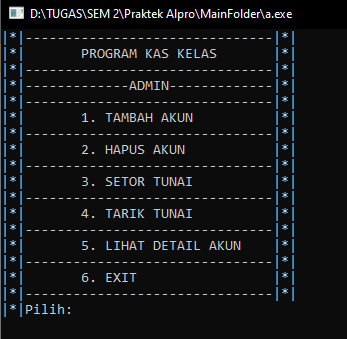
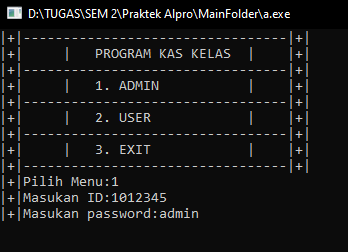
Untuk keperluan tampilan, identitas nama dari user dan ID tujuan akan ditampilkan. Nama ini diambil dari file txt detail akun bersangkutan. Setelah detail transfer ditampilkan ke user, user kemudian diberikan pilihan apakah akan melanjutkan untuk melakukan transfer. Jika user memilih untuk melanjutkan transfer maka fungsi universal tarikTunai akan dilakukan untuk akun user dan fungsi setorTunai akan dilakukan untuk akun tujuan transfer. Jika user memilih untuk tidak melanjutkan transfer maka transfer yang sebelumnya dipersiapkan tidak dilanjutkan.

OUTPUT PROGRAM

Tampilan awal program

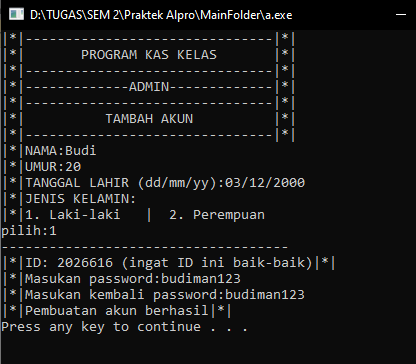


MODE ADMIN

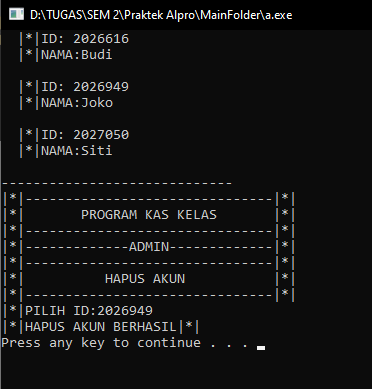


\

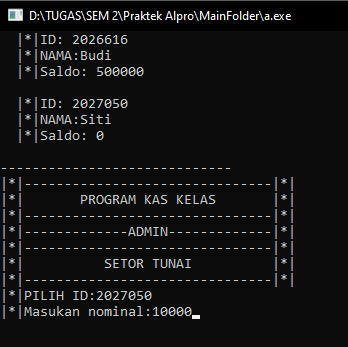
1. Tambah Akun (Mode Admin)

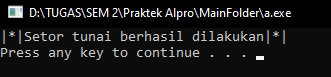


1. Hapus Akun (Mode Admin)

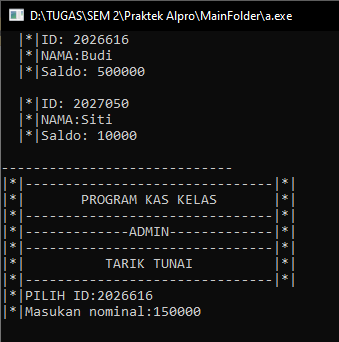


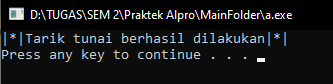
1. Setor tunai (Mode Admin)



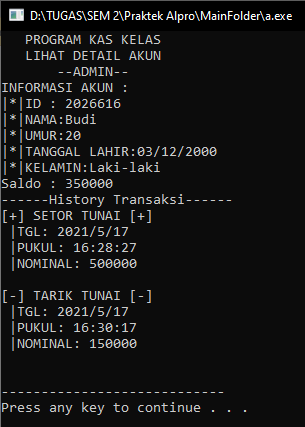


1. Tarik tunai (Mode Admin)

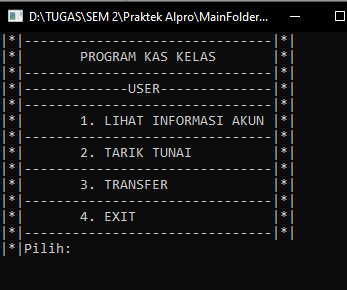
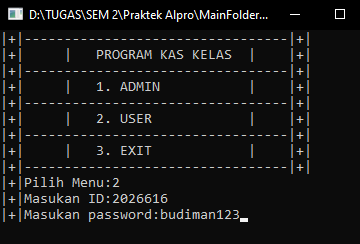




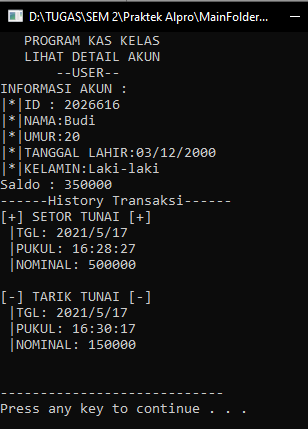
1. Lihat detail akun (Mode Admin)



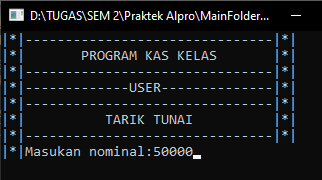
MODE USER

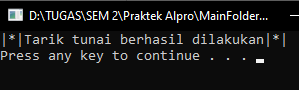


1. Lihat informasi akun (Mode User)



1. Tarik tunai (mode user)





1. Transfer tunai (Mode User)

