**ONLINE RETAILER**

LAPORAN PROYEK AKHIR

MATA KULIAH COMP6360 – ALGORITHM AND PROGRAMMING

KELAS LC20



Oleh:

2301949091 ARIA RIFQI

2301930476 BILLY MOSES

2301943491 JULIO RIVALDO

2301940861 KEVIN REYNALDI

Semester Ganjil (2019)

MALANG

**LEMBAR PERSETUJUAN PROYEK AKHIR**

**ONLINE RETAILER**

MATA KULIAH COMP6360 – ALGORITHM AND PROGRAMMING

KELAS LC20

Semester Ganjil (2019)

Laporan akhir ini adalah benar karya kami :

**(ARIA RIFQI) (BILLY MOSES)**

**2301949091 2301930476**

**(JULIO RIVALDO) (KEVIN REYNALDI)**

**2301943491 2301940861**

**Malang,.………………………**

**(Elizabeth Paskahlia Gunawan, S.Kom., M.Cs.)**

**D5707**

BAB I

LATAR BELAKANG

Pada masa sekarang ini penggunaan *online shop* untuk mempermudah proses jual-beli barang semakin sering dan bahkan selalu digunakan oleh khalayak ramai. Para penjual tidak perlu lagi repot-repot mencari tempat untuk menjajakan produk mereka karena hanya dengan gambar para penjual dapat dengan mudah menampilkan produk yang ingin mereka jual. Pembeli juga tidak perlu direpotkan untuk datang langsung ke toko membeli barang yang mereka butuhkan. Cukup duduk manis di rumah, dan dengan sekali klik, uang sudah ditransfer dan barang siap dikirim. Kemudahan ini tentu membuat semakin banyak orang tertarik untuk berjualan dan membeli melalui *online shop*. Belum lagi ditambah dengan perilaku masyarakat Indonesia yang cenderung konsumtif, ini tentu menjadi ladang emas bagi para penjual barang di *online shop*.

Namun, ini semua tidak akan pernah terjadi bila tidak ada orang-orang inspiratif yang membuat *online shop* menjadi hal yang meledak dan sering digunakan. Contoh saja Jack Ma pendiri “*Alibaba*”, William Tanuwijaya pendiri “*Tokopedia*” atau Achmad Zaky pendiri “*Bukalapak*”. Semua yang telah disebutkan tadi adalah sederetan *online shop* terkenal dan pendiri dibaliknya. Mereka membuka lapangan pekerjaan bagi banyak orang yang ingin berjualan dengan modal yang tidak banyak.

Berdasarkan pemaparan di atas, kelompok kami berencana merancang sebuah program sederhana bertema *online shop*. Kami menggunakan bahasa pemrograman C untuk merancang program ini. Program ini nantinya dapat digunakan oleh pembeli untuk mengecek ketersediaan barang, membeli barang, membayar, dan mengecek nomor resi. Selain itu program ini juga dapat diakses oleh pemilik online shop untuk melakukan penambahan barang, mengonfirmasi pembayaran, memasukkan nomor resi, dan menyimpan barang yang telah ditambahkan. Kelompok kami ingin membuat program ini karena kami ingin tahu bagaimana bila *online shop* dibuat secara sederhana. Membuat program seperti ini tentu akan mengasah logika dan pemahaman kami tentang bahasa C. Selain itu bila ada kemungkinan kami berharap dengan program dasar ini kami menjadi terlatih dan dapat mengembangkan aplikasi yang lebih besar dan dapat digunakan untuk semua orang.

BAB II

KAJIAN PUSTAKA

Algoritma ditemukan oleh seorang ilmuwan timur tengah bernama Abu Ja’far Muhammad Ibnu Musa Al-Khwarizmi. Ditinjau dari bahasa, algoritma berasal dari kata *algorism* yang artinya perhitungan dalam angka Arab, namun para ahli sejarah kesulitan menemukan asal-usul istilah ini, sampai akhirnya mereka menemukan bahwa *algorithm*/algoritma berasal dari nama penemunya, Al-Khwarizmi.

Algoritma adalah serangkaian instruksi atau langkah-langkah sistematis dan logis yang dirancang untuk menyelesaikan suatu masalah. Algoritma dapat berupa proses sederhana seperti mengalikan dua buah angka, atau operasi kompleks seperti menjalankan video yang telah dikompres. Karena algoritma adalah sebuah konsep, maka setiap orang dapat memiliki algoritma yang berbeda-beda terhadap masalah yang sama.

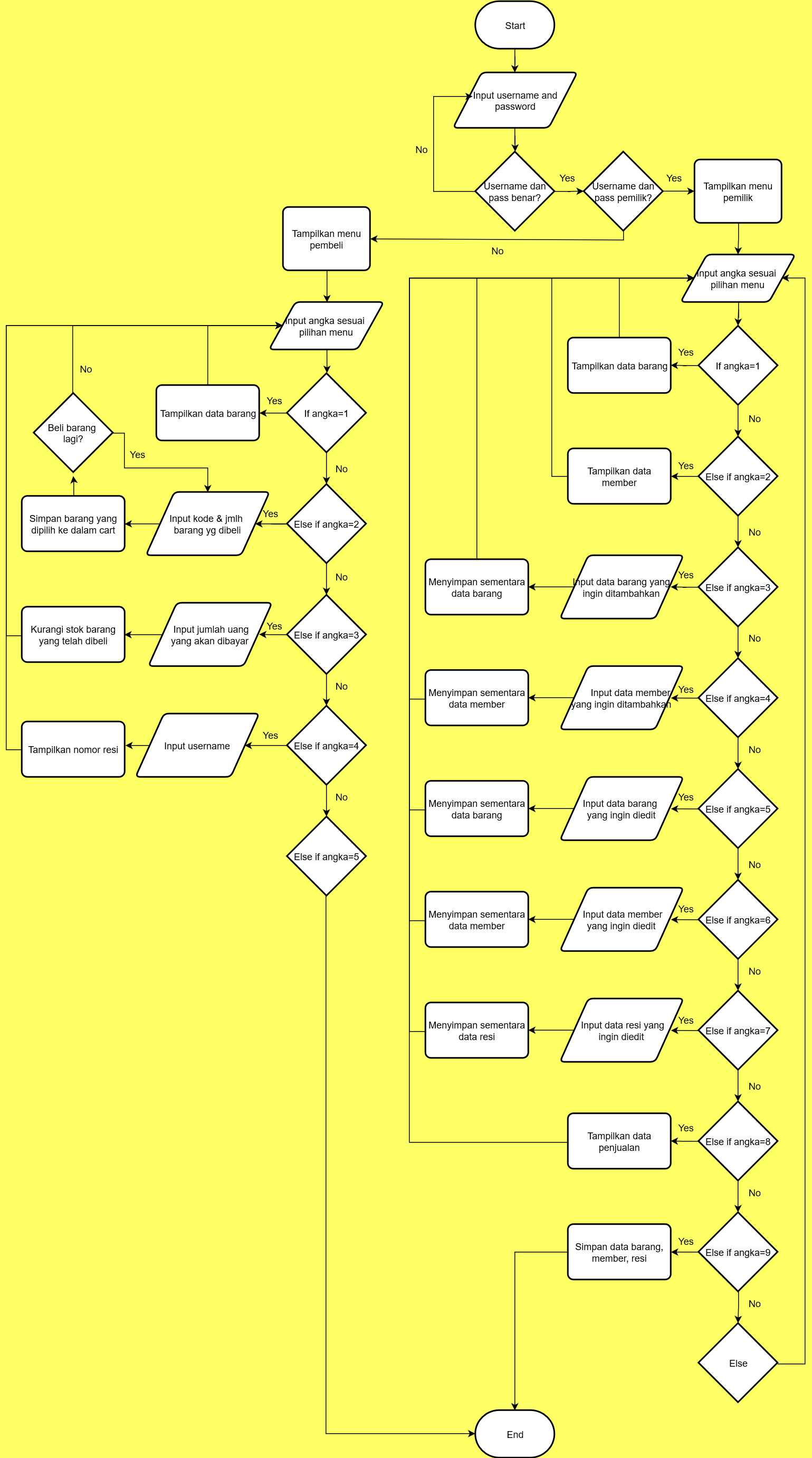
Penerapan algoritma haruslah efisien. Efisien di sini artinya cepat, tepat, dan sederhana. Contohnya software komputer, di mana software harus dibuat dengan algoritma yang baik agar tidak banyak bug dan tidak memakan banyak memori.

Pemrograman adalah sebuah proses membuat program menggunakan algoritma dan bahasa pemrograman. Penggunaan algoritma dalam pemrograman sangatlah penting. Dalam pemrograman, hal yang penting untuk dipahami adalah logika dalam berpikir bagaimana cara untuk memecahkan masalah pemrograman yang dibuat. Itulah mengapa algoritma sebagai dasar langkah-langkah dan logika dalam membuat program sangatlah penting. Selain itu pemrograman dapat berarti cara untuk memerintahakan komputer untuk melakukan berbagai tugas.

*Online shop* adalah adalah sebuah situs web di mana pelanggan dapat memesan barang. *Online shop* dapat mewakili toko kecil, toko eceran besar, toko *e-commerce* atau seseorang yang menjual barang melalui situs pihak ketika seperti *eBay*. Untuk menjalankan *online shop*, dibutuhkan katalog produk, keranjang belanja dan beberapa barang lainnya. Selain itu dibutuhkan juga *host web server* untuk menciptakan, menjalankan, dan mengurus *online shop*. *Online shop* membutuhkan sebuah nama domain agar keberadaannya dalam internet dianggap ada. Pemilik bisnis mendaftarkan nama domain dan menyambungkannya dengan *online shop*.

BAB III

GAMBARAN UMUM PROGRAM



Dalam proyek akhir ini kelompok kami membuat sebuah program sederhana yang bertemakan *online shop*. Dalam program sederhana ini pembeli dapat melakukan *login* dan kemudian mengecek stok barang yang ada di dalam katalog. Setelah melakukan pengecekan, pembeli dapat memasukkan barang yang ingin dibeli ke dalam keranjang belanja kemudian membayar sesuai jumlah yang tertera. Pembeli akan mendapatkan nomor resi pengiriman yang dapat dicek di laman cek resi. Setelah itu pembeli dapat keluar dari program atau membuka kembali katalog barang untuk berbelanja lagi.

Selain *login* pembeli, yang dapat masuk ke dalam program ini adalah pemilik dari *online shop* ini. Bila *login* menggunakan *username* dan *password* pemilik, maka laman dan pilihan yang ditampilkan akan berbeda dari isi laman pembeli di mana laman pemilik ini lebih rumit dalam pengerjaannya. Pemilik dapat memeriksa data barang yang masih ada dan data anggota. Dalam laman untuk mengecek data barang, pemilik dapat menyortir barang berdasarkan harga dan jumlah stok barang dari yang terkecil ke yang terbesar. Bila ada barang baru yang ingin dijual, pemilik dapat menambah data barang dengan memasukkan kode dan nama barang serta stok barang yang tersedia. Untuk data anggota, pemilik akan menambahkan bila ada yang mendaftar secara langsung dengan menghubungi pemilik. Selain menambahkan data, pemilik dapat mengedit data barang, data anggota, dan data resi. Edit data barang digunakan bila barang yang sudah tecantum di daftar katalog barang mendapatkan restok, sehingga tidak perlu dilakukan penambahan data barang karena yang perlu diubah hanyalah jumlah stok barang yang ada. Kemudian ada edit data member, di mana pemilik dapat mengedit data member bila terjadi kesalahan penulisan nama atau alamat atau tanggal lahir. Selain itu ada pula pilihan edit resi untuk mengedit nomor resi bila terjadi kesalahan angka dalam nomor resi. Bila pembeli melakukan pembelian, maka pemilik dapat mengecek data penjualan pada menu selanjutnya. Detil barang dan tanggal berapa barang tersebut terjual akan ditampilkan dalam halaman ini. Pilihan terakhir yang ada di dalam menu pemilik adalah menu simpan. Semua data yang telah diubah dalam menu-menu sebelumnya akan tersimpan setelah pemilik memilih opsi ini.

Program ini menggunakan *struct* untuk semua pembacaan dan penulisan ke dalam *file.* Dalam program ini, untuk bagian *login* menggunakan fungsi “*string compare”*. Fungsi ini akan melakukan perbandingan apakah karakter yang dimasukkan ke dalam *username* dan *password* sama dengan apa yang program ini inginkan. Untuk mengakses menu pemilik, maka pengguna harus memasukkan kata “*owner*” untuk *username* dan *password*nya. Program akan melakukan perbandingan apakah benar yang diinput oleh pengguna adalah kata “*owner*” atau bukan. Bila berbeda, maka program akan kembali ke halaman awal *login*. Sedangkan untuk pembeli, program akan melakukan perbandingan *string* dengan membaca *file* yang berisikan *username* dan *password* pembeli yang telah menjadi anggota. *Username* dan *password* yang diinput harus sama dengan apa yang telah tersimpan di dalam *file* yang ditunjuk dalam fungsi *login*. Untuk memanggil *file* tersebut, maka di dalam *string compare* dimasukkan susunan *struct* yang sesuai dengan *file* anggota.

Setelah *login* sebagai pemilik, akan tampil menu yang hanya dapat diakses oleh pemilik tersebut. Opsi pertama adalah untuk melihat data barang. Jika memilih opsi ini, maka akan tampil daftar barang yang ada di dalam *file* barang. Opsi kedua adalah melihat data anggota. Jika dipilih, opsi ini akan menampilkan daftar anggota yang telah terdaftar. Sama dengan opsi sebelumnya, jika memilih opsi ini maka program akan memanggil fungsi *file* yang berisikan data anggota dan menampilkannya di layar. Opsi ketiga adalah menambah data barang. Setelah mengisi kode barang, nama barang, stok barang, dan harga barang, maka data tersebut akan disimpan secara sementara. Kemudian pada opsi keempat, pemilik dapat menambahkan data anggota. Semua data anggota yang telah dimasukkan akan sementara disimpan. Opsi kelima dapat digunakan untuk mengedit data barang yang telah ada di dalam katalog (*file* barang), namun bukan menambahkan data barang baru yang belum ada di dalam katalog. Data yang telah diedit akan sementara disimpan. Opsi keenam memiliki fungsi yang kurang lebih sama dengan opsi kelima, namun kali ini yang diedit adalah data anggota, dan dalam opsi ketujuh pemilik dapat melakukan edit terhadap nama pembeli dan memiliki proses yang sama dengan opsi kelima dan keenam untuk penyimpanannya. Opsi kedelapan digunakan untuk melihat data penjualan dengan memanggil *file* penjualan. Pemilik dapat melihat barang apa yang telah terjual dan tanggal berapa barang tersebut terjual. Kemudian pada opsi kesembilan, pemilik dapat menggunakannya untuk menyimpan data-data yang telah diedit dan dimasukkan dari opsi ketiga sampai ketujuh secara permanen. Opsi terakhir ini akan memanggil semua fungsi *file* yang memiliki perintah *write*, agar semua data yang telah dimasukkan tertulis di dalam *file*. Setelah memilih opsi terakhir ini dan menyimpan semua data, program akan berakhir.

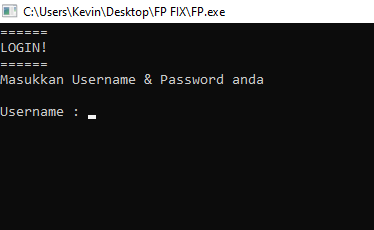
Jika *login* sebagai pembeli, maka akan muncul 5 opsi. Opsi pertama adalah katalog barang, di mana jika opsi ini dipilih program akan memanggil fungsi *file* yang berisikan data barang dan menampilkannya di layar. Kemudian pembeli dapat menyortir barang berdasarkan harga dan jumlah stok barang dari yang terkecil ke yang terbesar. Setelah pembeli menyortir barang dan menekan *“Enter”*, program akan mengembalikannya ke laman utama pembeli. Pada opsi kedua, pembeli dapat melihat lagi daftar barang yang ada dan memilih barang yang ingin dibeli. Pembeli harus memasukkan kode barang dan jumlah barang yang ingin dibeli. Barang dan jumlah barang yang akan dibeli kemudian disimpan secara sementara dan dimasukkan ke dalam keranjang belanja. Untuk melakukan pembayaran maka pembeli harus melakukan *checkout* dengan memilih opsi ketiga yaitu “bayar”. Data yang disimpan sementara di dalam keranjang belanja akan dimunculkan lagi di sini dengan total tagihan yang harus dibayarkan oleh pembeli. Pembeli kemudian memasukkan jumlah uang yang harus dibayarkan dan proses *checkout* berhasil. Data dari opsi kedua tidak disimpan secara permanen seperti pada menu pemilik. Setelah membeli, pembeli dapat melakukan cek resi dengan memilih opsi keempat dan memasukkan *username*. Nomor resi ini dihasilkan dengan menggunakan fungsi “*random*” agar angka yang dihasilkan acak dan menggunakan fungsi “*srand”* agar setiap kali program mengacak angka, angka yang dihasilkan berbeda dari angka acak yang telah dihasilkan sebelumnya. Dan untuk mengakhiri belanja maka pembeli dapat memilih opsi kelima, di mana opsi ini akan membawa pembeli keluar dari program.

Semua program yang membutuhkan opsi di atas diproses dengan menggunakan *“switch case”*. Untuk membaca *file* dan menampilkannya di layar tanpa mengubah isinya digunakan fungsi *file* dengan perintah “r” atau “*read”*. Semua data yang akan ditambahkan maupun diedit menggunakan *file* dengan perintah “w” atau *“write”*, di mana data yang telah diketik dan diinput akan masuk dan tertambah di dalam *file* sesuai nama *file* yang digunakan. Kemudian untuk mempermudah pembacaan *source code*, digunakan *“function”* agar mudah dalam melakukan perbaikan bila terjadi kesalahan.

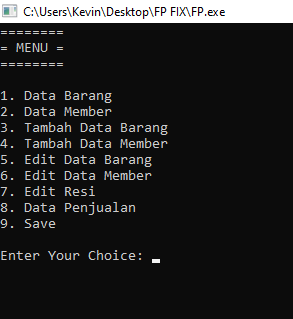
BAB IV

HASIL

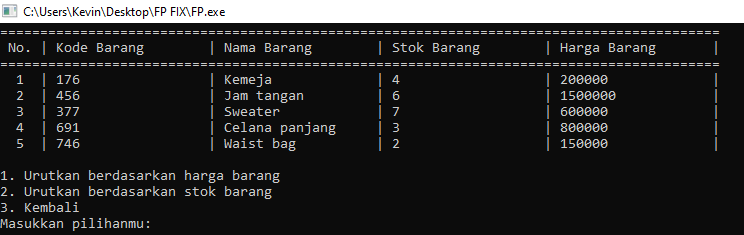
1. **Program display**



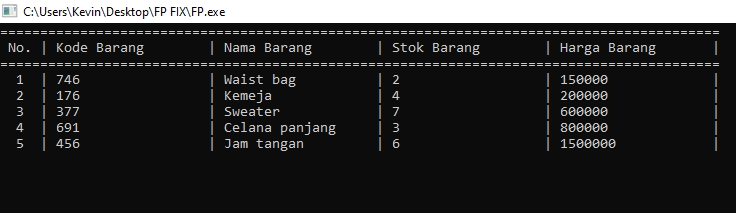
Halaman *login*



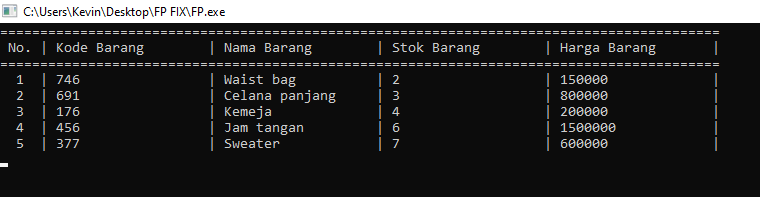
Menu utama pemilik



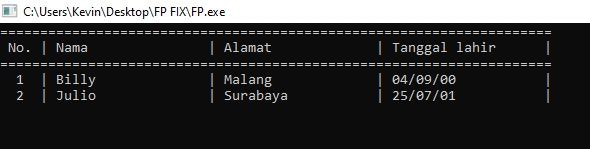
Menu 1



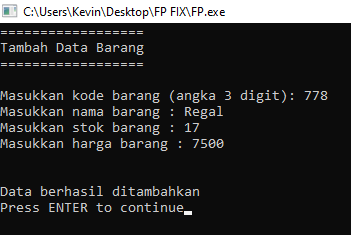
Menu 1 dengan *sorting* berdasarkan harga



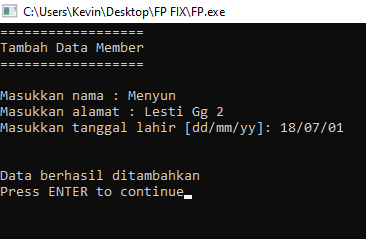
Menu 1 dengan *sorting* berdasarkan jumlah stok barang



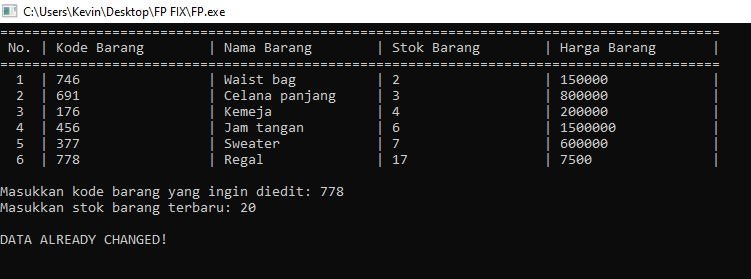
Menu 2



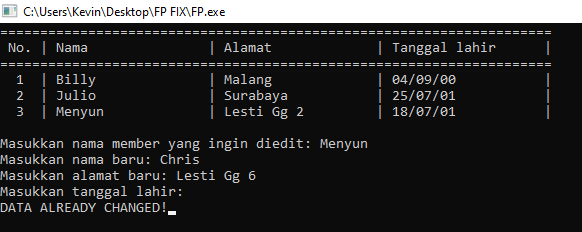
Menu 3



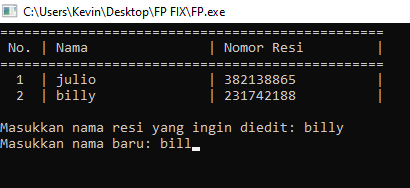
Menu 4



Menu 5



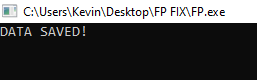
Menu 6



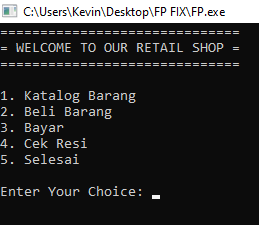
Menu 7



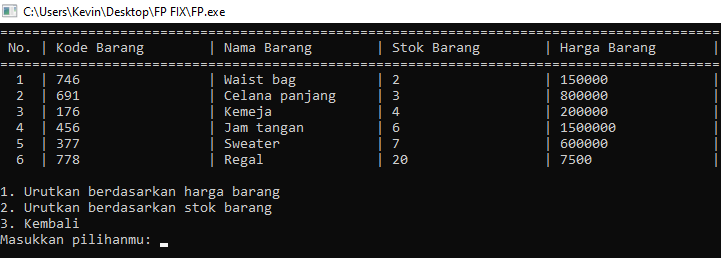
Menu 8



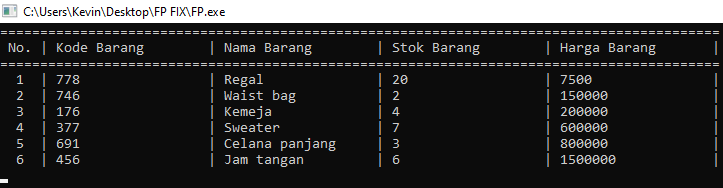
Hasil setelah menu 9 dipilih



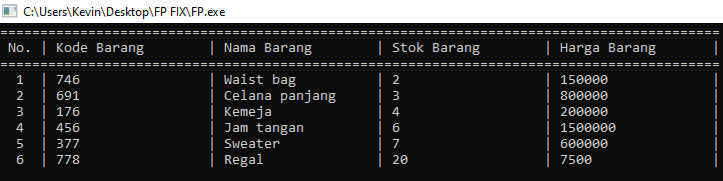
Menu utama pembeli



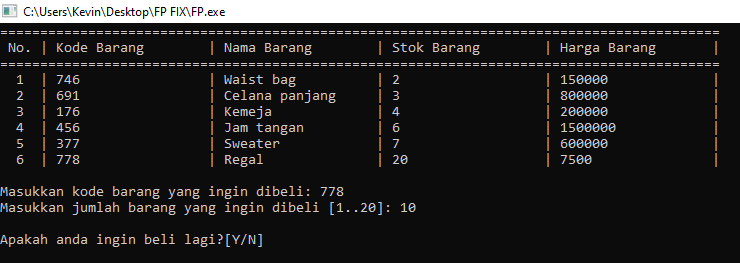
Menu 1



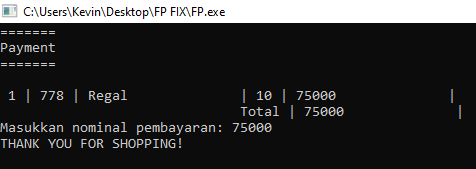
Menu 1 dengan *sorting* berdasarkan harga



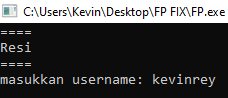
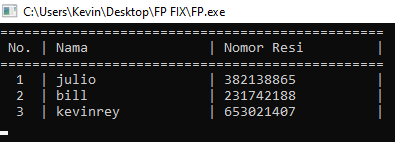
Menu 1 dengan *sorting* berdasarkan jumlah stok barang



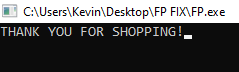
Menu 2



Menu 3



Menu 4



Hasil setelah menu 5 dipilih

1. **Source code**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<conio.h>

#include<time.h>

int ind\_brg=0;

int ind\_mmbr=0;

int ind\_resi=0;

int ind\_pjln=0;

int ind\_cart=0;

int index=0;

int index\_beli=0;

int stok\_beli;

struct pt {

char us[50],ps[20];

}LOGIN[50];

struct brg

{

char kode\_brg[50];

char nama\_brg[50];

int stok\_brg;

int harga\_brg;

}BRG[500],cart[500];

struct mmbr

{

char nama[50];

char alamat[50];

char tgllhr[50];

}MMBR[500];

struct trx

{

char no\_resi[50];

char nama\_pembeli[50];

char brg\_dibeli[50];

char tgl\_beli[50];

}TRX[500];

FILE \*brg;

FILE \*mmbr;

FILE \*resi;

FILE \*pjln;

FILE \*pf;

void file(){

pf=fopen("daftar.txt","r");

while(fscanf(pf,"%s %s\n",&LOGIN[index].us,&LOGIN[index].ps)!=EOF){

index++;

}

fclose(pf);

}

//BACA FILE"

//baca file barang

void load\_brg()

{

brg=fopen("brg.txt","r");

while(fscanf(brg,"%[^\'.'].%[^\'.'].%d.%d\n",BRG[ind\_brg].kode\_brg,BRG[ind\_brg].nama\_brg,&BRG[ind\_brg].stok\_brg,&BRG[ind\_brg].harga\_brg)!=EOF)

{

ind\_brg++;

}

fclose(brg);

}

//baca file member

void load\_mmbr()

{

mmbr=fopen("mmbr.txt","r");

while(fscanf(mmbr,"%[^\'.'].%[^\'.'].%[^\n]\n",MMBR[ind\_mmbr].nama,MMBR[ind\_mmbr].alamat,MMBR[ind\_mmbr].tgllhr)!=EOF)

{

ind\_mmbr++;

}

fclose(mmbr);

}

//baca file resi

void load\_resi()

{

resi=fopen("resi.txt","r");

while(fscanf(resi,"%[^\'.'].%[^\n]\n",TRX[ind\_resi].nama\_pembeli,TRX[ind\_resi].no\_resi)!=EOF)

{

ind\_resi++;

}

fclose(resi);

}

//baca file penjualan

void load\_pjln()

{

pjln=fopen("pjln.txt","r");

while(fscanf(pjln,"%[^\'.'].%[^\n]\n",TRX[ind\_pjln].brg\_dibeli,TRX[ind\_pjln].tgl\_beli)!=EOF)

{

ind\_pjln++;

}

fclose(pjln);

}

//FUNGSI" SEARCH

//search kode

int cari\_kode(char \*key)

{

for(int i=0; i<ind\_brg; i++)

{

if(strcmp(key,BRG[i].kode\_brg)==0)

return i;

}

return -1;

}

//search nama member

int cari\_nama(char \*key)

{

for(int i=0; i<ind\_mmbr; i++)

{

if(strcmp(key,MMBR[i].nama)==0)

return i;

}

return -1;

}

//search nama resi

int cari\_resi(char \*key)

{

for(int i=0; i<ind\_resi; i++)

{

if(strcmp(key,TRX[i].nama\_pembeli)==0)

return i;

}

return -1;

}

//FUNGSI" SORT

//sort harga barang

void sorthrgbrg()

{

for (int i = 0; i < ind\_brg; i++)

{

for (int j = ind\_brg-1; j >= 1 ; j--)

{

if (BRG[j-1].harga\_brg > BRG[j].harga\_brg)

{

struct brg temp;

temp = BRG[j];

BRG[j]= BRG[j-1];

BRG[j-1] = temp;

}

}

}

}

//sort stok barang

void sortstokbrg()

{

for (int i = 0; i < ind\_brg; i++)

{

for (int j = ind\_brg-1; j >= 1 ; j--)

{

if (BRG[j-1].stok\_brg > BRG[j].stok\_brg)

{

struct brg temp;

temp = BRG[j];

BRG[j]= BRG[j-1];

BRG[j-1] = temp;

}

}

}

}

//FUNCTION MENU OWNER

//Display data barang(owner)/menu1

void databrg1()

{

system("cls");

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

printf(" No. | Kode Barang | Nama Barang | Stok Barang | Harga Barang |\n");

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

for(int i=0;i<ind\_brg;i++)

{

printf(" %2d | %-18s | %-18s | %-18d | %-18d |\n",i+1, BRG[i].kode\_brg, BRG[i].nama\_brg, BRG[i].stok\_brg, BRG[i].harga\_brg);

}

fflush(stdin);

}

//Display data member(owner)/menu2

void datammbr1()

{

system("cls");

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

printf(" No. | Nama | Alamat | Tanggal lahir |\n");

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

for(int i=0;i<ind\_mmbr;i++)

{

printf(" %2d | %-18s | %-18s | %-18s |\n",i+1, MMBR[i].nama, MMBR[i].alamat, MMBR[i].tgllhr);

}

fflush(stdin);

}

//tambah data barang/menu3

void tmbbrg1()

{

system("cls");

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

printf("Tambah Data Barang\n");

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

printf("Masukkan kode barang (angka 3 digit): ");

fflush(stdin);

scanf("%s",BRG[ind\_brg].kode\_brg);

printf("Masukkan nama barang : ");

fflush(stdin);

scanf("%[^\n]",BRG[ind\_brg].nama\_brg);

printf("Masukkan stok barang : ");

fflush(stdin);

scanf("%d",&BRG[ind\_brg].stok\_brg);

printf("Masukkan harga barang : ");

fflush(stdin);

scanf("%d",&BRG[ind\_brg].harga\_brg);

printf("\n\nData berhasil ditambahkan\nPress ENTER to continue");

fflush(stdin);

ind\_brg++;

}

//tambah data member/menu4

void tmbmmbr1()

{

system("cls");

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

printf("Tambah Data Member\n");

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

printf("Masukkan nama : ");

fflush(stdin);

scanf("%[^\n]",MMBR[ind\_mmbr].nama);

printf("Masukkan alamat : ");

fflush(stdin);

scanf("%[^\n]",MMBR[ind\_mmbr].alamat);

printf("Masukkan tanggal lahir [dd/mm/yy]: ");

fflush(stdin);

scanf("%s",&MMBR[ind\_mmbr].tgllhr);

printf("\n\nData berhasil ditambahkan\nPress ENTER to continue");

fflush(stdin);

ind\_mmbr++;

}

//edit data barang/menu5

void editbrg1()

{

char kode\_edit[4];

int index\_edit;

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

printf("Edit Data Barang\n");

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

databrg1();

printf("\nMasukkan kode barang yang ingin diedit: ");

scanf("%s",kode\_edit);

index\_edit = cari\_kode(kode\_edit);

if(index\_edit == -1)

{

printf("Barang tidak ditemukan!");

}

else

{

printf("Masukkan stok barang terbaru: ");

scanf("%d",&BRG[index\_edit].stok\_brg);

printf("\nDATA ALREADY CHANGED!");

}

fflush(stdin);

}

//edit data member/menu6

void editmmbr1()

{

char data\_edit[4];

int index\_edit;

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

printf("Edit Data Member\n");

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

datammbr1();

printf("\nMasukkan nama member yang ingin diedit: ");

scanf("%s",data\_edit);

index\_edit = cari\_nama(data\_edit);

if(index\_edit == -1)

{

printf("Member tidak ditemukan!");

}

else

{

printf("Masukkan nama baru: ");

scanf("%s",MMBR[index\_edit].nama);

printf("Masukkan alamat baru: ");

scanf("%s",MMBR[index\_edit].alamat);

printf("Masukkan tanggal lahir: ");

scanf("%s",MMBR[index\_edit].tgllhr);

printf("\nDATA ALREADY CHANGED!");

}

fflush(stdin);

}

//display resi/menu4

void dataresi1()

{

system("cls");

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

printf(" No. | Nama | Nomor Resi |\n");

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

for(int i=0;i<ind\_resi;i++)

{

printf(" %2d | %-18s | %-18s |\n",i+1, TRX[i].nama\_pembeli, TRX[i].no\_resi);

}

fflush(stdin);

}

//Edit data resi//menu7

void editresi1()

{

char data\_edit[4];

int index\_edit;

printf("%s", TRX[ind\_resi].nama\_pembeli);

printf("\nMasukkan nama resi yang ingin diedit: ");

scanf("%s",data\_edit);

index\_edit = cari\_resi(data\_edit);

if(index\_edit == -1)

{

printf("Nama tidak ditemukan!");

}

else

{

printf("Masukkan nama baru: ");

scanf("%s",TRX[index\_edit].nama\_pembeli);

printf("\nDATA ALREADY CHANGED!");

}

fflush(stdin);

}

//display data penjualan/menu8

void datapjln1()

{

system("cls");

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

printf(" No. | Nama Barang | Tanggal Beli |\n");

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

for(int i=0;i<ind\_pjln;i++)

{

printf(" %2d | %-18s | %-18s |\n",i+1, TRX[i].brg\_dibeli, TRX[i].tgl\_beli);

}

fflush(stdin);

}

//save data/menu9/menu5

//cetak ke file brg

void savebrg1()

{

brg=fopen("brg.txt","w");

for(int i=0;i<ind\_brg;i++)

{

fprintf(brg,"%s.%s.%d.%d\n",BRG[i].kode\_brg,BRG[i].nama\_brg,BRG[i].stok\_brg,BRG[i].harga\_brg);

}

fclose(brg);

}

//cetak ke file member

void savemmbr1()

{

mmbr=fopen("mmbr.txt","w");

for(int i=0;i<ind\_mmbr;i++)

{

fprintf(mmbr,"%s.%s.%s\n",MMBR[i].nama,MMBR[i].alamat,MMBR[i].tgllhr);

}

fclose(mmbr);

}

//cetak ke file resi

void saveresi1()

{

resi=fopen("resi.txt","w");

for(int i=0;i<ind\_resi;i++)

{

fprintf(resi,"%s.%s\n",TRX[i].nama\_pembeli,TRX[i].no\_resi);

}

fclose(resi);

}

//cetak ke file penjualan

void savepjln1()

{

pjln=fopen("pjln.txt","a");

for(int i=0;i<ind\_cart;i++)

{

fprintf(pjln,"%s.%s\n",TRX[i].brg\_dibeli,TRX[i].tgl\_beli);

}

fclose(pjln);

}

//FUNCTION MENU BUYER

//beli barang/menu2

void beli2()

{

char lanjut;

do

{

system("cls");

char kode\_beli[4];

printf("===\n");

printf("BUY\n");

printf("===\n");

databrg1();

printf("\nMasukkan kode barang yang ingin dibeli: ");

scanf("%s",kode\_beli);

index\_beli = cari\_kode(kode\_beli);

if(index\_beli == -1)

{

printf("Barang tidak ditemukan!");

}

else

{

do

{

printf("Masukkan jumlah barang yang ingin dibeli [1..%d]: ",BRG[index\_beli].stok\_brg);

fflush(stdin);

scanf("%d",&stok\_beli);

}while(stok\_beli<1 || stok\_beli>BRG[index\_beli].stok\_brg);

BRG[index\_beli].stok\_brg=BRG[index\_beli].stok\_brg-stok\_beli;

cart[ind\_cart] = BRG[index\_beli];

cart[ind\_cart].stok\_brg = stok\_beli;

cart[ind\_cart].harga\_brg = stok\_beli \* BRG[index\_beli].harga\_brg;

ind\_cart++;

}

do

{

printf("\nApakah anda ingin beli lagi?[Y/N] ");

fflush(stdin);

scanf("%c",&lanjut);

}while(lanjut != 'Y' && lanjut != 'y' && lanjut != 'n' && lanjut != 'N');

}while(lanjut == 'Y' || lanjut == 'y');

fflush(stdin);

}

//bayar barang/menu3

void bayar2()

{

int total\_harga = 0,pembayaran,kembalian,jadi;

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

printf("Payment\n");

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

for(int i=0; i<ind\_cart; i++)

{

printf(" %d | %3s | %-18s | %d | %-18d |\n", i+1, cart[i].kode\_brg, cart[i].nama\_brg, cart[i].stok\_brg, cart[i].harga\_brg);

}

for(int i=0; i<ind\_cart; i++)

{

total\_harga = total\_harga + cart[i].harga\_brg;

}

for(int i=0; i<30; i++)

{

printf(" ");

}

printf("Total | %-18d |", total\_harga);

printf("\nMasukkan nominal pembayaran: ");

scanf("%d",&pembayaran);

kembalian=pembayaran-total\_harga;

if(kembalian<0)

{

printf("\nMaaf uang anda kurang!\n");

printf("Apakah anda jadi beli [Y/N]: ");

fflush(stdin);

scanf("%c",&jadi);

if(jadi == 'Y'||jadi == 'y')

{

printf("\n");

bayar2();

}

else

{

printf("\nPastikan uangmu cukup ya lain kali..");

BRG[index\_beli].stok\_brg=BRG[index\_beli].stok\_brg+stok\_beli;

ind\_cart=0;

}

}

else if(kembalian>0)

{

printf("\nUang yang anda berikan lebih\n");

printf("Kembalian anda: %d", kembalian);

ind\_cart = 0;

}

else

{

printf("THANK YOU FOR SHOPPING!");

ind\_cart = 0;

}

}

//MAIN MENU

//Menu OWNER

void menu1()

{

int pilih1;

do

{

system("cls");

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

printf("= MENU =\n");

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

printf("1. Data Barang\n");

printf("2. Data Member\n");

printf("3. Tambah Data Barang\n");

printf("4. Tambah Data Member\n");

printf("5. Edit Data Barang\n");

printf("6. Edit Data Member\n");

printf("7. Edit Resi\n");

printf("8. Data Penjualan\n");

printf("9. Save\n");

printf("\nEnter Your Choice: ");

scanf("%d", &pilih1);

switch(pilih1)

{

case 1:

{

system("cls");

int choice;

databrg1();

printf("\n1. Urutkan berdasarkan harga barang\n2. Urutkan berdasarkan stok barang\n3. Kembali");

printf("\nMasukkan pilihanmu: ");

fflush(stdin);

scanf("%d", &choice);

if(choice == 1)

{

sorthrgbrg();

system("cls");

databrg1();

getch();

}

else if(choice == 2)

{

sortstokbrg();

system("cls");

databrg1();

getch();

}

break;

}

case 2:

{

system("cls");

datammbr1();

getch();

break;

}

case 3:

{

system("cls");

tmbbrg1();

getch();

break;

}

case 4:

{

system("cls");

tmbmmbr1();

getch();

break;

}

case 5:

{

system("cls");

editbrg1();

getch();

break;

}

case 6:

{

system("cls");

editmmbr1();

getch();

break;

}

case 7:

{

system("cls");

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

printf("Edit Resi\n");

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

dataresi1();

editresi1();

break;

}

case 8:

{

system("cls");

datapjln1();

getch();

break;

}

case 9:

{

system("cls");

savebrg1();

savemmbr1();

saveresi1();

savepjln1();

printf("DATA SAVED!\n");

getch();

break;

}

default:

{

break;

}

}

}while(pilih1 != 9);

}

//MENU BUYER

void menu2()

{

int pilih2;

do

{

system("cls");

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

printf("= WELCOME TO OUR RETAIL SHOP =\n");

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

printf("1. Katalog Barang\n");

printf("2. Beli Barang\n");

printf("3. Bayar\n");

printf("4. Cek Resi\n");

printf("5. Selesai\n");

printf("\nEnter Your Choice: ");

scanf("%d", &pilih2);

switch(pilih2)

{

case 1:

{

system("cls");

int choice;

databrg1();

printf("\n1. Urutkan berdasarkan harga barang\n2. Urutkan berdasarkan stok barang\n3. Kembali");

printf("\nMasukkan pilihanmu: ");

fflush(stdin);

scanf("%d", &choice);

if(choice == 1)

{

sorthrgbrg();

system("cls");

databrg1();

getch();

}

else if(choice == 2)

{

sortstokbrg();

system("cls");

databrg1();

getch();

}

break;

}

case 2:

{

system("cls");

beli2();

break;

}

case 3:

{

system("cls");

bayar2();

getch ();

break;

}

case 4:

{

system("cls");

srand (time(0));

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

printf("Resi\n");

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

printf("masukkan username: ");

fflush(stdin);

scanf("%s",TRX[ind\_resi].nama\_pembeli);

for(int i=0;i<9;i++)

{

TRX[ind\_resi].no\_resi[i]=rand()%9+48;

}

printf("\n\nNomor resi telah ditambahkan!");

ind\_resi++;

dataresi1();

getch ();

break;

}

case 5:

{

system("cls");

savebrg1();

saveresi1();

printf("THANK YOU FOR SHOPPING!");

getch();

break;

}

default:

{

break;

}

}

}while(pilih2!=5);

}

//LOGIN

void login()

{

int k,l=1;

char user[50],pass[20];

file();

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

printf("LOGIN!\n");

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

printf("Masukkan Username & Password anda\n\n");

printf("Username : ");

scanf("%s",user);

printf("Password : ");

scanf("%s",pass);

if(strcmp(user,"owner")==0 || strcmp(pass,"owner")==0){

menu1();

}

if(strcmp(user,"owner")!=0 || strcmp(pass,"owner")!=0){

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

if(strcmp(user,LOGIN[k].us)==0 && strcmp(pass,LOGIN[k].ps)==0){

l=k;

menu2();

}

}

}

if(strcmp(user,"owner")!=0 || strcmp(pass,"owner")!=0){

if(strcmp(user,LOGIN[l].us)!=0 || strcmp(pass,LOGIN[l].ps)!=0){

system("cls");

login();

}

}

}

int main()

{

load\_brg();

load\_mmbr();

load\_resi();

load\_pjln();

login();

return 0;

}

LAIN-LAIN

Pembagian tugas :

**IDE and I/O instruction Syntax I/O and Variable** : Kevin, Julio, Aria, Billy

**Arithmetic Operation** : Kevin

**Selection** : Julio, Aria

**Repetition** : Aria

**Array 1 D** : Kevin

**Function** : Aria

**Recursive** : Julio

**File** : Kevin, Billy

**Sorting** : Billy

**Searching** : Billy, Julio

DAFTAR PUSTAKA

“Pengertian Algoritma Pemrograman”.TeknoJurnal.com.28 April 2016.4 Januari 2020.<https://teknojurnal.com/pengertian-algoritma-pemrograman/>

“Algoritma dan Pemrograman Itu Apa Sih?”.teknosains.com.13 Maret 2018.4 Januari 2020. <https://teknosains.com/sains-teknologi/algoritma-dan-pemrograman-itu-apa-sih>

“Algorithm in Programming”.Programiz.com.4 Januari 2020.<https://www.programiz.com/article/algorithm-programming>

“Algorithm”.techterms.com.4 Januari 2020.<https://techterms.com/definition/algorithm>

“What is Programming? [Beginner's Guide]”.hackr.io.4 Januari 2020.<https://hackr.io/blog/what-is-programming>

“How Does an Online Store Work?”.11 Februari 2019.4 Januari 2020.<https://bizfluent.com/how-does-4671808-online-store-work.html>

LEMBAR PENILAIAN

**ONLINE RETAILER**

MATA KULIAH COMP6360 – ALGORTIHM AND PROGRAMMING

KELAS LC20

Semester Ganjil (2019)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DAFTAR MAHASISWA** | **NILAI** | | | | **BOBOT** | | | | **KREDIT** | | | | **TOTAL**  **KREDIT** |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| 2301949091 ARIA RIFQI |  |  |  |  | 20% | 20% | 24% | 36% |  |  |  |  |  |
| 2301930476 BILLY MOSES |  |  |  |  | 20% | 20% | 24% | 36% |  |  |  |  |  |
| 2301943491  JULIO RIVALDO |  |  |  |  | 20% | 20% | 24% | 36% |  |  |  |  |  |
| 2301940861 KEVIN REYNALDI |  |  |  |  | 20% | 20% | 24% | 36% |  |  |  |  |  |
| **TOTAL** | | | | | | | | | | | | |  |

**KETERANGAN :**

* **Skala Penilaian : 0 sd. 100**
* **Komponen**

1 : Laporan

2 : Produk

3 : Pengetahuan

4 : Solusi

**Malang,…………………**

**(Elizabeth Paskahlia Gunawan, S.Kom., M.Cs.)**

**D5707**