**PROGRAM   
PERPUSTAKAAN SEDERHANA**



Anggota Kelompok :  
Agung Prabowo (1810631170082)  
Muhamad Iqbal Faddillah (1810631170049)

1. Tugas masing-masing anggota

* Agung prabowo

Membuat Program utama

* Muhamad Iqbal Faddillah

Merapihkan program

Membuat Flowchart

1. Deskripsi fasilitas aplikasi

Program ini di buat untul keperluan pendataan pada sebuah perpustakaan dimana aplikasi ini memiliki akses untuk dua level yaitu untuk Admin dan Guest.

Fasilitas yang tersedia untuk admin adalah kami memberikan fungsi untuk ; Lihat Detail Buku, Peminjaman Buku, Pengembalian Buku. Tambah Data Buku, Hapus Data Buku, dan Cetak Daftar Buku

Sedangkan pada Guest hanya memiliki fungsi Lihat Detail Buku, dan Cetak Daftar Buku.

1. *Source code project*

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_LEN 100 //panjang maksimal string

//komponen kartu

struct kartu

{

char tanggal\_pinjam[MAX\_LEN];

char tanggal\_balik[MAX\_LEN];

int status;

};

typedef struct kartu Kartu; // typedef membentuk suatu variabel baru dengan kumpulan berbagai variabel dari type yang berbeda-beda

//komponen buku

struct buku

{

Kartu card[MAX\_LEN]; //array of kartu

char judul\_buku[MAX\_LEN];

char pengarang[MAX\_LEN];

char penerbit[MAX\_LEN];

int tahun;

int stock;

int id\_buku;

};

typedef struct buku Buku; // typedef membentuk suatu variabel baru dengan kumpulan berbagai variabel dari type yang berbeda-beda

//komponen linked-list

struct node

{

Buku x;

struct node \*next;

};

typedef struct node Node;

void printHeader();

void defaultStock();

void adminSession();

void guestSession();

Node \*head;

int main()

{

int pw;

char user;

printHeader(); //header

defaultStock(); //mengisi linked-list dengan stock awal

//select session

printf("\t\tLOGIN\n");

printf("\t\tUsername :");scanf("%c", &user);

printf("\t\tPassword :");scanf("%d", &pw);

system("cls");

if(pw==1) adminSession();

else if(pw==2) guestSession();

else printf("Error, pilihan tidak tersedia, silakan cek kembali\n");

return 0;

}

/////////////////////////print function/////////////////////////////////

void printHeader()

{

system("Color F3");

printf("------------------------------------------------------\n");

printf("\t\t Proyek UAS \n");

printf(" Agung Prabowo & M. Iqbal Faddillah\n");

printf("\t\tTEKNIK INFORMATIKA\n");

printf("------------------------------------------------------\n");

printf("------------------------------------------------------\n");

printf(" \n");

}

void setcolor(unsigned short color);

void printAdminMenu()

{

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

printf("\t\t MENU \n");

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

printf("\t\t1.Lihat Detail Buku\n");

printf("\t\t2.Pengembalian Buku\n");

printf("\t\t3.Peminjaman Buku\n");

printf("\t\t4.Tambah Data Buku\n");

printf("\t\t5.Hapus Data Buku\n");

printf("\t\t6.Cetak Daftar Buku\n");

printf("\t\t7.Keluar\n");

printf("\t\tPilihan: ");

}

void printGuestMenu()

{

printf("\n\n++++++++++MENU+++++++++++\n");

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

printf("1.Lihat Detail Buku\n");

printf("2.Cetak Daftar Buku\n");

printf("3.Keluar\n");

printf("Pilihan: ");

}

/////////////////////////fungsi linked-list/////////////////////////////

void defaultStock()

{

Node \*reference; // linked list dengan pointer reference

int i;

//buku 1

head = (Node \*) malloc(sizeof(Node)); //alokasi memory untuk head

strcpy(head->x.judul\_buku, "Electrical Book"); // menyalin head dgn variabel judul buku

strcpy(head->x.pengarang, "Evitabasa");

strcpy(head->x.penerbit, "Universindo");

head->x.tahun = 2016; //mengisi struct head dgn akses variabel tahun pd record x dgn data 2015

head->x.stock = 10;

head->x.id\_buku = 1;

for(i=0; i<head->x.stock; i++)

head->x.card[i].status = 1; // banyak buku sesuai inputan stok

//buku 2

reference = head; //reference dipointing ke head

reference->next = (Node \*) malloc(sizeof(Node));//tambah node baru di akhir list

reference = reference->next; //reference dipointing ke node baru

reference->next = 0;

strcpy(reference->x.judul\_buku, "Mechanic Book");

strcpy(reference->x.pengarang, "Gilang L");

strcpy(reference->x.penerbit, "Universindo");

reference->x.tahun = 1996;

reference->x.stock = 10;

reference->x.id\_buku = 2;

for(i=0; i<reference->x.stock; i++)

reference->x.card[i].status = 1;

}

Node \*searchBefore(Node \*current) // untuk proses hapus

{

Node \*before = head;

while(before->next != current && before != NULL)

{

before = before->next;

}

return before;

}

Node \*searchAfter(Node \*current) // untuk proses hapus

{

Node \*after = head;

if(after==current) return after->next;

else

{

while(after->next != current && after != NULL)

{

after = after->next;

}

if(after!=NULL) after = current->next;

return after;

}

}

Node \*searchBuku(int id\_buku)

{

Node \*reference = head;

while(reference != NULL && reference->x.id\_buku != id\_buku)

reference = reference->next;

return reference;

}

Node \*sortingBuku(int id\_buku)

{

Node \*reference, \*before, \*insert;

insert = (Node \*) malloc(sizeof(Node));

insert->next = NULL;

if(head!=NULL)

{

if(id\_buku < head->x.id\_buku) //insert in first node (head)

{

insert->next = head;

head = insert;

}

else

{

reference = head;

before = head;

reference->next = insert; //insert in last node

}

}

else head = insert;

return insert;

}

/////////////////////////fungsi pada menu///////////////////////////////

int getStockBuku(Node \*current)

{

int i=0;

int current\_stock=0;

while(current->x.card[i].status!=0)

{

if(current->x.card[i].status==1) current\_stock++;

i++;

}

return current\_stock;

}

void printBuku(Node \*reference)

{

printf("%d\t", reference->x.id\_buku);

printf("%s\t", reference->x.judul\_buku);

printf("%s\t", reference->x.pengarang);

printf("%s\t", reference->x.penerbit);

printf("%d\t\t",reference->x.tahun);

printf("%d\n", getStockBuku(reference));

}

void printDetail(Node \*detail)

{

int i=0;

printf("\n\nID\tStatus\t\tJudul Buku\t\tTanggal Pinjam\tTanggal Kembali\n");

while(detail->x.card[i].status!=0) // buku masih utuh sejumlah stok

{

if(detail->x.card[i].status!=-1)

{

printf("%d\t", i);

if(detail->x.card[i].status == 2) printf("Dipinjam\t");

else if(detail->x.card[i].status == 1) printf("Available\t");

printf("%s\_%d\t", detail->x.judul\_buku, i);

printf("%s\t", detail->x.card[i].tanggal\_pinjam);

printf("%s\n", detail->x.card[i].tanggal\_balik);

i++;

}

}

}

void cetakListBuku() //void untuk mencetak daftar buku

{

Node \*reference;

reference = head; //reference sama dengan head node selajutnya

printf("\n\nID Buku\tJudul Buku\tNama Pengarang\tPenerbit\tTahun\t\tStock\n");

if (reference != 0)

{

while (reference->next != 0) // Ketika reference dngn pointer next tidak sama dengan 0

{

printBuku(reference);

reference = reference->next;

}

printBuku(reference); //ke void print buku dgn passing variabel reference

}

}

void detailBuku()

{

Node \*detail;

int id\_buku;

printf("Masukkan ID Buku (Ketik 0 untuk melihat ID Buku) : ");

scanf("%d", &id\_buku);

while(id\_buku==0)

{

cetakListBuku();

printf("\nMasukkan ID Buku (Ketik 0 untuk melihat ID Buku) : ");

scanf("%d", &id\_buku);

}

detail = searchBuku(id\_buku); //menampilkan list buku

if(detail!=NULL) // sudah mecapai akhir node, id tidak ditemukan

printDetail(detail);

else printf("\nID Buku tidak ditemukan, silakan cek kembali!");

}

void pinjamBuku()

{

Node \*current;

char tanggal\_pinjam[MAX\_LEN];

char tanggal\_balik[MAX\_LEN];

int id\_buku, current\_stock, i=0;

printf("Masukkan ID Buku yang akan dipinjam (Ketik 0 untuk melihat ID Buku) : ");

scanf("%d", &id\_buku);

while(id\_buku==0)

{

cetakListBuku();

printf("\nMasukkan ID Buku yang akan dipinjam (Ketik 0 untuk melihat ID Buku) : ");

scanf("%d", &id\_buku);

}

current = searchBuku(id\_buku);

current\_stock = getStockBuku(current);

if(current!=NULL && current\_stock!=0)

{

while(current->x.card[i].status!=0)

{

if(current->x.card[i].status==1)

{

getchar(); // menunggu input karakter

printf("Tanggal Pinjam (format: dd-mm-yy) : ");

fgets(tanggal\_pinjam,100,stdin); // fungsi seperti scanf

tanggal\_pinjam[strlen(tanggal\_pinjam) - 1] = '\0'; // \0 untuk konstan karakter printf("Tanggal Kembali (format: dd-mm-yy) : ");

fgets(tanggal\_balik,100,stdin);

tanggal\_balik[strlen(tanggal\_balik) - 1] = '\0';

strcpy(current->x.card[i].tanggal\_pinjam, tanggal\_pinjam);

strcpy(current->x.card[i].tanggal\_balik, tanggal\_pinjam);

current->x.card[i].status=2;

break;

}

i++;

}

}

else if(current\_stock==0) printf("\nMaaf, saat ini stock sedang kosong");

else printf("\nID Buku tidak ditemukan, silakan cek kembali!");

}

void kembaliBuku()

{

Node \*current;

int id\_buku, id\_peminjam;

printf("Masukkan ID Buku yang akan dikembalikan(Ketik 0 untuk melihat ID Buku) : ");

scanf("%d", &id\_buku);

while(id\_buku==0)

{

cetakListBuku();

printf("\nMasukkan ID Buku yang akan dikembalikan (Ketik 0 untuk melihat ID Buku) : ");

scanf("%d", &id\_buku);

}

current = searchBuku(id\_buku);

if(current!=NULL)

{

printf("Masukkan ID Detail (Ketik -1 untuk melihat Detail Buku) : ");

scanf("%d", &id\_peminjam);

while(id\_peminjam==-1)// -1 untuk menampilkan detail buku

{

printDetail(current); // meampilkan detail buku

printf("\nMasukkan ID Detail (Ketik -1 untuk melihat Detail Buku) : ");

scanf("%d", &id\_peminjam);

}

if(current->x.card[id\_peminjam].status == 2)

{

current->x.card[id\_peminjam].status = 1;

strcpy(current->x.card[id\_peminjam].tanggal\_balik, "");

strcpy(current->x.card[id\_peminjam].tanggal\_pinjam, "");

}

else printf("\nID Detail salah, silakan cek kembali!");

}

else printf("\nID Buku tidak ditemukan, silakan cek kembali!");

}

void hapusBuku()

{

Node \*hapus, \*before, \*after;

int id\_buku;

after = searchAfter(head);

head = after;

printf("\nData berhasil dihapus!");

}

void tambahBuku()

{

Node \*insert;

char judul\_buku[MAX\_LEN];

char pengarang[MAX\_LEN];

char penerbit[MAX\_LEN];

int tahun;

int stock;

int id\_buku;

int i;

printf("Masukkan Data Baru:\n");

printf("ID Buku : ");

scanf("%d", &id\_buku);

getchar();

printf("Judul Buku : ");

fgets(judul\_buku,100,stdin);

judul\_buku[strlen(judul\_buku) - 1] = '\0';

printf("Pengarang : ");

fgets(pengarang,100,stdin);

pengarang[strlen(pengarang) - 1] = '\0';

printf("Penerbit : ");

fgets(penerbit,100,stdin);

penerbit[strlen(penerbit) - 1] = '\0';

printf("Tahun : ");

scanf("%d", &tahun);

printf("Stock : ");

scanf("%d", &stock);

// insert->next = (Node \*) malloc(sizeof(Node));

insert = sortingBuku(id\_buku); //sorting buku

insert->x.id\_buku = id\_buku;

insert->x.tahun = tahun;

insert->x.stock = stock;

strcpy(insert->x.judul\_buku, judul\_buku);

strcpy(insert->x.pengarang, pengarang);

strcpy(insert->x.penerbit, penerbit);

for(i=0; i<stock; i++)

insert->x.card[i].status = 1; //looping slot buku sebanyak stok, status =1 berarti available

printf("\nData telah ditambahkan!");

}

/////////////////////////admin session//////////////////////////////////

void adminSession()

{

int menu;

printf("\n\nSelamat Datang, Administrator!\n");

while(1)

{

printAdminMenu();

scanf("%d", &menu);

system("cls");

if(menu==1) detailBuku();

else if(menu==2) kembaliBuku();

else if(menu==3) pinjamBuku();

else if(menu==4) tambahBuku();

else if(menu==5) hapusBuku();

else if(menu==6) cetakListBuku();

else if(menu==7) break;

else printf("\n\nError, menu tidak tersedia, silakan dicek kembali\n\n");

}

}

/////////////////////////guest session//////////////////////////////////

void guestSession()

{

int menu;

printf("\n\nSelamat Datang, Guest!\n");

while(1)

{

printGuestMenu();

scanf("%d", &menu);

system("cls");

if(menu==1) detailBuku();

else if(menu==2) cetakListBuku();

else if(menu==3) break;

else printf("\n\nError, menu tidak tersedia, silakan dicek kembali\n\n");

}

}

**Screenshot Running Program**











 

 