

Anggota kelompok : Syaifudin Amin  
Judul: penyewaan CD

```
#include <iostream>
#include <string>
using namespace std;

//----- D A T A   C D -----
typedef struct {
    string kode_cd;
    string judul;
    int harga_sewa;
    int status;
}InfoKaset;

typedef struct tNodeKaset *addressKaset;
typedef struct tNodeKaset
{
    InfoKaset InfoKst;
    addressKaset NextKst;
}NodeKaset;

typedef NodeKaset *ListKaset;

//----- D A T A   P E L A N G G A N -----
typedef struct {
    string no_ktp;
    string nama;
    string alamat;
    string no_telp;
}InfoPelanggan;

typedef struct tNodePelanggan *addressPelanggan;
typedef struct tNodePelanggan
{
    InfoPelanggan InfoPel;
    addressPelanggan NextPel;
}NodePelanggan;

typedef NodePelanggan *ListPelanggan;

//----- D A T A   T R A N S A K S I -----
typedef struct {
    string kode_transaksi;
    string kode_cd;
    string no_ktp;
    string tgl_pinjam;
    string tgl_kembali;
    int total_biaya;
}InfoTransaksiSewa;

typedef struct tNodeTransaksiSewa *addressTransaksiSewa;
typedef struct tNodeTransaksiSewa
{
    InfoTransaksiSewa InfoTrans;
    addressTransaksiSewa NextTrans;
}NodeTransaksiSewa;

typedef NodeTransaksiSewa *ListTransaksiSewa;

void CreateEmpty(ListKaset &L);
void CreateEmpty(ListPelanggan &L);
void CreateEmpty(ListTransaksiSewa &L);
```

Anggota kelompok : Syaifudin Amin  
Judul: penyewaan CD

```
bool IsEmpty(ListKaset L);
bool IsEmpty(ListPelanggan L);
bool IsEmpty(ListTransaksiSewa L);
bool IsOneElmt(ListKaset L);
bool IsOneElmt(ListPelanggan L);
bool IsOneElmt(ListTransaksiSewa L);
addressKaset Alokasi(InfoKaset X);
addressPelanggan Alokasi(InfoPelanggan X);
addressTransaksiSewa Alokasi(InfoTransaksiSewa X);
void InsertFirst(ListKaset &L, addressKaset P);
void InsertFirst(ListPelanggan &L, addressPelanggan P);
void InsertFirst(ListTransaksiSewa &L, addressTransaksiSewa P);
void PrintInfo(ListKaset L);
void PrintInfo(ListPelanggan L);
void PrintInfo(ListTransaksiSewa L);
bool IsFoundJudul(ListKaset L, string _judul);
bool IsFoundNamaPelanggan(ListPelanggan L, string _namaPelanggan);
bool IsFoundKodeTrans(ListTransaksiSewa L, string _kdTrans);

void main()
{
    ListKaset collectionCD;
    ListPelanggan collectionP;
    ListTransaksiSewa collectionT;
    CreateEmpty(collectionCD);
    CreateEmpty(collectionP);
    CreateEmpty(collectionT);
    InfoKaset kaset_baru;
    InfoPelanggan pelanggan_baru;
    InfoTransaksiSewa transaksi_baru;
    //-----Menu utama -----
    -----
    int pilih;
menu:
    cout << "Menu Utama\n";
    cout << "1. Masukkan Data Kaset\n";
    cout << "2. Masukkan Data Pelanggan\n";
    cout << "3. Masukkan Data Transaksi\n";
    cout << "4. Tampil Data Kaset\n";
    cout << "5. Tampil Data Pelanggan\n";
    cout << "6. Tampil Data Transaksi\n";
    cout << "7. Cari Data\n";
    cout << "Pilih Menu: ";
    cin >> pilih;

    switch (pilih)
    {
        case 1:
        {
            cout << "Kode Kaset: ";
            cin >> kaset_baru.kode_cd;
            cout << "Judul Kaset: ";
            cin >> kaset_baru.judul;
            cout << "Jumlah Barang: ";
            cin >> kaset_baru.status;
            cout << "Harga Sewa: ";
            cin >> kaset_baru.harga_sewa;
            addressKaset node_kaset_baru = Alokasi(kaset_baru);
            InsertFirst(collectionCD, node_kaset_baru);
            system("cls");
            goto menu;
        }
    }
}
```

Anggota kelompok : Syaifudin Amin

Judul: penyewaan CD

```
break;
case 2:
{
    cout << "no_ktp: ";
    cin >> pelanggan_baru.no_ktp;
    cout << "nama: ";
    cin >> pelanggan_baru.nama;
    cout << "alamat: ";
    cin >> pelanggan_baru.alamat;
    cout << "no_telp: ";
    cin >> pelanggan_baru.no_telp;
    addressPelanggan node_pelanggan = Alokasi(pelanggan_baru);
    InsertFirst(collectionP, node_pelanggan);
    system("cls");
    goto menu;
}
break;
case 3:
{
    cout << "kode transaksi: ";
    cin >> transaksi_baru.kode_transaksi;
    cout << "kode CD: ";
    cin >> transaksi_baru.kode_cd;
    cout << "no.ktp pelanggan: ";
    cin >> transaksi_baru.no_ktp;
    cout << "tanggal peminjaman: ";
    cin >> transaksi_baru.tgl_pinjam;
    cout << "tanggal kembali: ";
    cin >> transaksi_baru.tgl_kembali;
    cout << "biaya: ";
    cin >> transaksi_baru.total_biaya;
    addressTransaksiSewa node_transaksi = Alokasi(transaksi_baru);
    InsertFirst(collectionT, node_transaksi);
    system("cls");
    goto menu;
}
break;
case 4:
{
    if (collectionCD == 0)
    {
        cout << "Data Kaset Kosong";
    }
    else if (collectionCD != 0)
    {
        PrintInfo(collectionCD);
        cout << endl;
    }
    else;
}
break;

case 5:
{
    if (collectionP == 0)
    {
        cout << "Data Pelanggan Kosong";
    }
    else if (collectionP != 0)
    {
        PrintInfo(collectionP);
        cout << endl;
    }
}
```

Anggota kelompok : Syaifudin Amin  
Judul: penyewaan CD

```
        else;

    }
    break;

    case 6:
    {
        if (collectionT == 0)
        {
            cout << "Data Transaksi Kosong\n";
        }
        else if (collectionT != 0)
        {
            PrintInfo(collectionT);
            cout << endl;
        }
        else;
    }
    break;
    case 7:
    {
        string cari;
        cout << "masukkan judul/nama pelanggan/kode transaksi:";
        cin >> cari;
        cout << "cd = ";
        cout << IsFoundJudul(collectionCD, cari);
        cout << "\npelanggan = ";
        cout << IsFoundNamaPelanggan(collectionP, cari);
        cout << "\nkode transaksi = ";
        cout << IsFoundKodeTrans(collectionT, cari);
        cout << endl;
    }
    break;
    default:
        cout << "masukan salah";
    }
    system("pause");
}

void CreateEmpty(ListKaset &L)
{
    L = NULL;
}
void CreateEmpty(ListPelanggan &L)
{
    L = NULL;
}
void CreateEmpty(ListTransaksiSewa &L)
{
    L = NULL;
}

bool IsEmpty(ListKaset L)
{
    return(L == NULL);
}
bool IsEmpty(ListPelanggan L)
{

```

Anggota kelompok : Syaifudin Amin  
Judul: penyewaan CD

```
        return(L == NULL);
    }
bool IsEmpty(ListTransaksiSewa L)
{
    return(L == NULL);
}

bool IsOneElmt(ListKaset L)
{
    return(IsEmpty(L->NextKst));
}
bool IsOneElmt(ListPelanggan L)
{
    return(IsEmpty(L->NextPel));
}
bool IsOneElmt(ListTransaksiSewa L)
{
    return(IsEmpty(L->NextTrans));
}

addressKaset Alokasi(InfoKaset X)
{
    addressKaset P = new(NodeKaset);
    if (P == NULL)        //alokasi gagal
        return NULL;
    else                    //alokasi berhasil
    {
        P->InfoKst = X;
        P->NextKst = NULL;
        return P;
    }
}
addressPelanggan Alokasi(InfoPelanggan X)
{
    addressPelanggan P = new(NodePelanggan);
    if (P == NULL)        //alokasi gagal
        return NULL;
    else                    //alokasi berhasil
    {
        P->InfoPel = X;
        P->NextPel = NULL;
        return P;
    }
}
addressTransaksiSewa Alokasi(InfoTransaksiSewa X)
{
    addressTransaksiSewa P = new(NodeTransaksiSewa);
    if (P == NULL)        //alokasi gagal
        return NULL;
    else                    //alokasi berhasil
    {
        P->InfoTrans = X;
        P->NextTrans = NULL;
        return P;
    }
}

void InsertFirst(ListKaset &L, addressKaset P)
{

```

Anggota kelompok : Syaifudin Amin  
Judul: penyewaan CD

```
        P->NextKst = L;
        L = P;
    }
void InsertFirst(ListPelanggan &L, addressPelanggan P)
{
    P->NextPel = L;
    L = P;
}
void InsertFirst(ListTransaksiSewa &L, addressTransaksiSewa P)
{
    P->NextTrans = L;
    L = P;
}

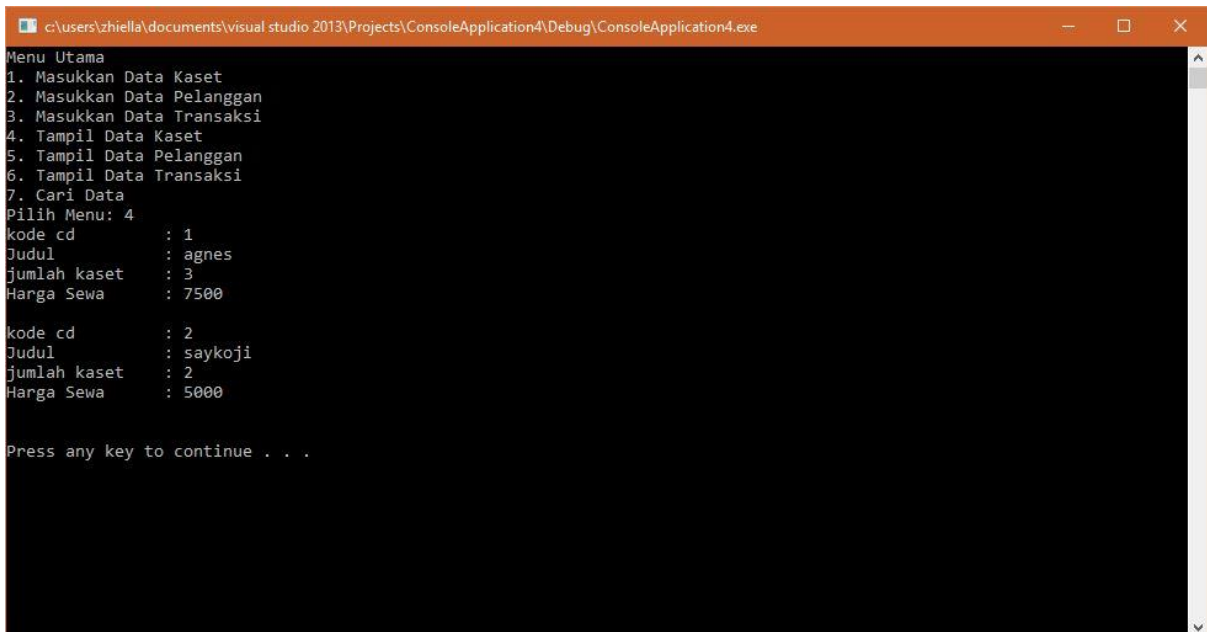
void PrintInfo(ListKaset L)
{
    if (IsEmpty(L))                //basis
    { //no action
    }
    else                            //rekurens
    {
        cout <<"kode cd\t\t: " << L->InfoKst.kode_cd << " \n";
        cout <<"Judul\t\t: " << L->InfoKst.judul << "\n";
        cout <<"jumlah kaset\t: " << L->InfoKst.status << "\n";
        cout <<"Harga Sewa\t: " << L->InfoKst.harga_sewa << "\n";
        cout << endl;
        PrintInfo(L->NextKst);
    }
}

void PrintInfo(ListPelanggan L)
{
    if (IsEmpty(L))                //basis
    { //no action
    }
    else                            //rekurens
    {
        cout << "No. ktp\t: " << L->InfoPel.no_ktp << " \n";
        cout << "Nama\t: " << L->InfoPel.nama << "\n";
        cout << "Alamat\t: " << L->InfoPel.alamat << "\n";
        cout << "No.telip\t: " << L->InfoPel.no_telp << "\n";
        cout << endl;
        PrintInfo(L->NextPel);
    }
}

void PrintInfo(ListTransaksiSewa L)
{
    if (IsEmpty(L))                //basis
    { //no action
    }
    else                            //rekurens
    {
        cout << "Kode Transaks\t\t: " << L->InfoTrans.kode_transaksi << " \n";
        cout << "Kode CD\t\t\t: " << L->InfoTrans.kode_cd << "\n";
        cout << "no.Ktp\t\t\t: " << L->InfoTrans.no_ktp << "\n";
        cout << "tanggal Pinjam\t\t: " << L->InfoTrans.tgl_pinjam << "\n";
        cout << "Tanggal kembali Sewa\t: " << L->InfoTrans.tgl_kembali << "\n";
        cout << "Total Biaya\t\t: " << L->InfoTrans.total_biaya << "\n";
        PrintInfo(L->NextTrans );
    }
}
}
```

Anggota kelompok : Syaifudin Amin  
Judul: penyewaan CD

```
bool IsFoundJudul(ListKaset L, string _judul)
{
    if (IsEmpty(L)) //basis 0
        return false;
    else //rekurens
        return (L->InfoKst.judul == _judul || IsFoundJudul(L->NextKst, _judul));
}
bool IsFoundNamaPelanggan(ListPelanggan L, string _namaPelanggan)
{
    if (IsEmpty(L)) //basis 0
        return false;
    else //rekurens
        return (L->InfoPel.nama == _namaPelanggan || IsFoundNamaPelanggan(L->NextPel, _namaPelanggan));
}
bool IsFoundKodeTrans(ListTransaksiSewa L, string _kdTrans)
{
    if (IsEmpty(L)) //basis 0
        return false;
    else //rekurens
        return (L->InfoTrans.kode_transaksi == _kdTrans || IsFoundKodeTrans(L->NextTrans, _kdTrans));
}
```



```
c:\users\zhiella\documents\visual studio 2013\Projects\ConsoleApplication4\Debug\ConsoleApplication4.exe
Menu Utama
1. Masukkan Data Kaset
2. Masukkan Data Pelanggan
3. Masukkan Data Transaksi
4. Tampil Data Kaset
5. Tampil Data Pelanggan
6. Tampil Data Transaksi
7. Cari Data
Pilih Menu: 4
kode cd      : 1
Judul       : agnes
jumlah kaset : 3
Harga Sewa  : 7500

kode cd      : 2
Judul       : saykoji
jumlah kaset : 2
Harga Sewa  : 5000

Press any key to continue . . .
```