POSTTEST 5

Ibnu Yafi Arya Wardana --- 2109106039

A2 Informatika’21

Program Pendataan Hotel

**== Instruksi Program: ==**  
Lanjutkan Program Posttest Sebelumnya Dengan Menambahkan : 

* Fitur Sorting (Pengurutan) Data Array Struct
* Pengurutan bisa dilakukan secara ascending atau descending
* Minimal menggunakan 3 metode sorting
* Dapat memilih ingin sorting berdasarkan attribute yang mana [ ex : berdasarkan nama/nim/ipk ]   
  minimal 2 attribute

**Nilai (+)**

* Kerapian Program
* Error Handling
* Menggunakan semua metode sorting yang ada di modul

**Tips:**

* **Usahakan Kurangi Penggunaan goto. goto membuat kode anda sulit diprediksi dan sulit dibaca**
* **Gunakan getline untuk input string**
* **Letakkan comment di line/baris baru, jangan disebelah kodingan**

# **SOURCE CODE**

|  |
| --- |
| DEV C++ |
| // Header Dasar: cout, cin, endl  #include <iostream>  // Header untuk menggunakan getch() dan getche()  #include <conio.h>  // Header untuk mewarnai console  #include <windows.h>  // Header dipakai untuk mengubah int menjadi str  #include <string.h>  // Untuk meringkas penulisan std::  using namespace std;  /\*------------------------ STRUCT DATA PENGUNJUNG ------------------------\*/  struct dataPengunjung{  int nomorKamar ;  string nomorPengenal ;  string nama, tipeKamar, catatan ;  string tanggalCheckin, tanggalCheckout ;  bool wni ;  };  /\*------------------------ DEKLARASI VAR GLOBAL ------------------------\*/  // Var untuk prasyarat perulangan program  bool berjalan = true ;  // Var untuk kekosongan selection sort  bool state = false;  // Var untuk pergi ke informasi data  int menujuData ;  // Var untuk jawaban pengguna terhadap menu  char responMenu ;  /\*------------------------ PROTOTYPE ------------------------\*/  // Dekorasiprogram  void banner () ;  void clear () ;  void palette (string namaWarna) ;  void setColor (WORD c) ;  void ccout (string warnaAwal, string kalimat) ;  int ccin (string warnaAwal, int var) ;  // Usable  void bubbleSort (dataPengunjung struk[]) ;  bool sortCondition (dataPengunjung struk[]) ;  void selectionSort (dataPengunjung struk[]) ;  int papanKetik (dataPengunjung struk[]) ;  char menuMetode (int rute) ;  int search (dataPengunjung struk[], int nomor) ;  int search (dataPengunjung struk[], string tipeKamar) ;  int search (string nama, dataPengunjung struk[]) ;  // Menu  void checkin (dataPengunjung struk[]) ;  void checkout (dataPengunjung struk[]) ;  void editHotel (dataPengunjung struk[]) ;  void properEditMenu(dataPengunjung struk[], int index) ;  // Penampil atau Pengubah Data  void dataTampil (dataPengunjung struk[]) ;  void dataTampil (dataPengunjung struk[], int index) ;  void dataTampilNama(dataPengunjung struk[], string pengurutan, bool state);  void hapusData (dataPengunjung struk[], int index) ;  char konfCheckout (dataPengunjung struk[], int index) ;  string pilihTipeKamar() ;  void pindahKamar (dataPengunjung struk[], int kamarSatu, int kamarDua) ;  // Default  void namaDefault (dataPengunjung struk[]) ;  void kamarDefault (dataPengunjung struk[]) ;  /\*------------------------ FUNGSI MAIN ------------------------\*/  int main(){  struct dataPengunjung kunjung[50] ;    kamarDefault(kunjung) ;  namaDefault (kunjung) ;  system ("color 70") ;  // Ubah warna keseluruhan data    do{  // Penggunaan Do-while    clear() ;      //\_\_\_\_\_\_\_\_ Menu Utama \_\_\_\_\_\_\_\_//  cout << " Menu Interaktif: \n" << endl  << "\t [I / 1] Check-in" << endl  << "\t [O / 2] Check-out" << endl  << "\t [D / 3] Data Pengunjung" << endl  << "\t [E / 4] Edit Data" << endl  << "\t [M / 5] Olah Kamar" << endl  << "\t [S / 6] Data Nama" << endl  << "\t [X / 7] Hentikan Program" << endl;  responMenu = getch();  // Getch() agar pengguna tak perlu menekan tombol Enter    switch(responMenu){  // Penggunaan switch-case untuk respon pengguna    // Check-In  case '1': case 'I': case 'i':  checkin (kunjung);  break ;  // Check-Out  case '2': case 'O': case 'o':  checkout (kunjung);  break ;  // All Data Viewer  case '3': case 'D': case 'd':  dataTampil (kunjung);  break ;  // Data Editor  case '4': case 'E': case 'e':  clear() ;  menujuData = papanKetik (kunjung) ;  if (menujuData == 99 || menujuData == 50){  break ;}  else{properEditMenu(kunjung, menujuData) ;}  break ;    // Main Data Editor  case '5': case 'M': case 'm':  editHotel(kunjung) ;  break ;  // Used Data Viewer  case '6': case 'S': case 's':  state = sortCondition(kunjung) ;  if (state == true){selectionSort(kunjung);} ;  dataTampilNama(kunjung,"ASC",state) ;  break ;  // Quit  case '7': case 'X': case 'x':  berjalan = false;  break ;}    }  while(berjalan == true);    clear () ;  ccout ("Merah", "\n Program Dihentikan.");  }      /\*------------------------ DECORATION ------------------------\*/  void banner(){  palette("MerahPutih");  cout << " " << endl  << " \_-\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-\_ " << endl  << " | | | | " << endl  << " | | [HOTEL BERNAMA] | | " << endl  << " | | | | " << endl  << " |\_-\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-\_| " << endl  << " " << endl;  palette("");  }  void clear(){  // Hapus layar CMD  system ("CLS");  banner () ;  }  // Set Warna  void setColor(WORD c){  SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), c); // Pengubah warna console windows  }  // Beri Warna  void palette(string namaWarna){  if (namaWarna == "Biru") {setColor(113);}  else if (namaWarna == "Hijau") {setColor(114);}  else if (namaWarna == "Merah") {setColor(116);}  else if (namaWarna == "Kuning") {setColor(118);}  else if (namaWarna == "Abu-abu") {setColor(120);}  else if (namaWarna == "MerahPutih") {setColor(508);}  else {setColor(112);}  }  // Colour Finale  void ccout(string warnaAwal, string kalimat){  palette (warnaAwal) ;  cout << kalimat << endl ;  palette ("") ;  }  int ccin(string warnaAwal, int var){  palette (warnaAwal) ;  cin >> var ;  palette ("") ;  return var ;  }  /\*------------------------ UTENSILS ------------------------\*/  // Cari Nomor Kamar  int search(dataPengunjung struk[], int nomor){  // var untuk menampung index dicari  int indexDitemukan = 0;    // Penggunaan for untuk menuju semua data  for (int i = 0; i <= 50; i++){  if (i == 50) {indexDitemukan = 50;}  else if (struk[i].nomorKamar == nomor) {break;}  else {indexDitemukan++;}  }    return indexDitemukan;  }  // Cari Ketersediaan Kamar  //--Fungsi overloading  int search(dataPengunjung struk[], string tipeKamar){  // var untuk menampung index dicari  int indexDitemukan = 0;    for (int i = 0; i <= 50; i++){  if (i == 50) {indexDitemukan = 50;}  else if (struk[i].nama == "" && struk[i].tipeKamar == tipeKamar) {break;}  else {indexDitemukan++;}  }    return indexDitemukan;  }  // Cari Nama  int search(string nama, dataPengunjung struk[]){  // var untuk menampung index dicari  int indexDitemukan = 0;    for (int i = 0; i <= 50; i++){  if (i == 50) {indexDitemukan = 50;}  else if (struk[i].nama == nama) {break;}  else {indexDitemukan++;}  }    return indexDitemukan;  }  // Tipe Kamar  string pilihTipeKamar(){  // var respon user  char opsiTipeKamar ;    // var pilihan user  string tipeKamar ;    cout << "\n Tipe Kamar: " << endl  << " [0] Standard Room" << endl  << " [1] Superior Room" << endl  << " [2] Deluxe Room" << endl  << " [3] Junior Suite Room" << endl  << " [4] Suite Room" << endl  << " [5] Single Room" << endl  << " [6] Twin Room" << endl  << " [7] Double Room" << endl  << " >> " ;  opsiTipeKamar = getche() ;    switch(opsiTipeKamar){  case '0':  tipeKamar = "Standard Room" ;  break ;  case '1':  tipeKamar = "Superior Room" ;  break ;  case '2':  tipeKamar = "Deluxe Room" ;  break ;  case '3':  tipeKamar = "Junior Suite Room" ;  break ;  case '4':  tipeKamar = "Suite Room" ;  break ;  case '5':  tipeKamar = "Single Room" ;  break ;  case '6':  tipeKamar = "Twin Room" ;  break ;  case '7':  tipeKamar = "Double Room" ;  break ;  default:  tipeKamar = "invalid" ;  break ;  }  return tipeKamar ;  }  //------ Sorting Kamar  // Bubble Sort  void bubbleSort(dataPengunjung struk[]){  for(int i = 0; i < 49; i++){  for(int j = 0; j < 48; j++){  if (struk[j].nomorKamar == 0){  swap(struk[j], struk[j+1]);  }  else if (struk[j].nomorKamar > struk[j+1].nomorKamar  && struk[j+1].nomorKamar != 0){  swap(struk[j], struk[j+1]);  }  }  }}  // Cek Isi Data  bool sortCondition(dataPengunjung struk[]){  int count = 0;      for (int f = 0; f < 50; f++){  if (struk[f].nama == ""){count++;}}    if (count == 50){  clear() ;  ccout("Merah", "\n\t Belum ada data pengunjung.") ;  ccout("Kuning", "\t [Tekan apa saja.]") ;  getch() ;  return false ;}  else {  return true;  }  }  // Selection Sort  void selectionSort(dataPengunjung struk[]){  int key ;    for (int i = 0; i < 48; i++){  key = i ;  for (int j = i+1; j < 49; j++){  if (struk[j].nama.compare(struk[key].nama) < 0){  key = j ;}  swap(struk[key], struk[i]);}  }  }  // Insertion Sort  void insertionSort(dataPengunjung struk[]){  int j;    for (int i = 1; i < 48; i++){  j = i;    while(j - 1 > -1){  if(struk[j].nama.compare(struk[j-1].nama) > 0){  swap(struk[j], struk[j-1]);}  j--;}}  }  // Delete aka Check Out Data  void hapusData(dataPengunjung struk[], int index){  struk[index].nama = "" ;  struk[index].nomorPengenal = "" ;  struk[index].tanggalCheckin = "" ;  struk[index].tanggalCheckout = "" ;  struk[index].catatan = "" ;  }  // Peringkas Menu Yang Ditulis Berulang  char menuMetode(int rute){  clear() ;    if (rute == 1){  // Olah Data    cout << "\n Pilihan Metode Cari: " << endl  << " [1] Dengan nomor kamar" << endl  << " [2] Dengan nama" << endl;  rute = getch() ;}    if (rute == 2){  // Olah Hotel    cout << "\n Pilihan Pengolahan: " << endl  << " [1] Tambahkan Kamar baru" << endl  << " [2] Olah Kamar lama" << endl;  rute = getch() ;}    return rute ;  }  // Untuk Papan Ketikan  int papanKetik(dataPengunjung struk[]){  // var untuk menampung inputan user  int ketik;      ccout ("Abu-abu",  " Ketik '0' untuk kembali ke menu awal") ;  cout << "\t [|| " ;    ketik = ccin("Biru", ketik) ;      cout << "\t ````````" << endl ;    // Kembali ke Menu  if (ketik == 0) {return 99;}  else {  ketik = search(struk, ketik) ;    // Data tidak ada  if (ketik == 50){    ccout("Merah", " Kamar Tersebut Tidak Ada." );  ccout("Kuning", " [Tekan apa saja.]" );  getch() ;  return 50 ;}    else{return ketik ;}}  }  // Untuk Pindah Kamar  void pindahKamar(dataPengunjung struk[], int kamarSatu, int kamarDua){  ccout("Hijau", "\t\t Berhasil Dipindahkan." );  ccout("Kuning", "\t\t [Tekan apa saja.]" );  getch() ;    swap(struk[kamarSatu], struk[kamarDua]) ;  swap(struk[kamarSatu].nomorKamar, struk[kamarDua].nomorKamar);  swap(struk[kamarSatu].tipeKamar, struk[kamarDua].tipeKamar) ;  properEditMenu(struk, kamarDua) ;}  /\*------------------------ CHECKIN ------------------------\*/  void checkin(dataPengunjung struk[]){  // var untuk pilihan kewarganegaraan  char kewarganegaraan ;    // var untuk nomor pengenal  string nomorPengenal ;    // var nama, var tipe kamar dipilih, var catatan tambahan  string nama, tipeKamar, catatan ;    clear() ;    cout << "\n Nama: "; fflush(stdin) ;  getline(cin,nama) ;  // Penggunaan fflush dan getline untuk mengambil inputan dengan spasi      cout << " Nomor Pengenal: " ;  cin >> nomorPengenal ;    tipeKamar = pilihTipeKamar() ;    // var untuk menampung indeks kamar yang tersedia  int slot = search(struk, tipeKamar) ;      if (slot == 50){  ccout("Merah", "\t\t Kamar Tidak Tersedia.");  ccout("Kuning", "\t\t [Tekan apa saja.]" );  getch() ;}    else{  cout << "\n Nomor Kamar: " ;  palette("Biru") ;  cout << struk[slot].nomorKamar << "\n\n" ;  palette("") ;    cout << " Opsi Kewarganegaraan:" << endl  << " [0] WNA" << endl  << " [1] WNI" << endl  << " >> " ;  kewarganegaraan = getche() ;  // getche(), sama dengan getch() namun inputan ditampilkan  struk[slot].wni = (kewarganegaraan == '1')? true : false;  // Penggunaan ternary untuk meringkas percabangan    cout << "\n Catatan: "; fflush(stdin) ;  getline(cin,catatan) ;    struk[slot].nama = nama ;  struk[slot].nomorPengenal = nomorPengenal ;  struk[slot].catatan = catatan ;      cout << "\n Tanggal Check-in: "; fflush(stdin) ;  getline(cin,struk[slot].tanggalCheckin) ;    cout << " Tanggal Check-out: " ; fflush(stdin) ;  getline(cin,struk[slot].tanggalCheckout) ;      ccout("Biru", "\n\t\t Berhasil ditambahkan.") ;  ccout("Kuning", "\t\t [Tekan apa saja.]") ;  getch() ;}  }  /\*------------------------ CHECKOUT ------------------------\*/  void checkout(dataPengunjung struk[]){  // var jalur menu metode, var  char rute = menuMetode(1) ;    // var indeks  int nomor ;    // var nama  string nama ;    switch (rute){  case '1':  // dengan nomor kamar    cout << " Masukkan nomor kamar: " ;  nomor = ccin("Biru", nomor) ;  nomor = search(struk, nomor) ;    if (nomor == 50){  ccout("Merah", " Kamar tersebut tidak ada.") ;  ccout("Kuning", " [Tekan apa saja.]") ;  getch() ;}    else{  konfCheckout(struk, nomor) ;}  break ;    case '2':  // dengan nama    cout << " Masukkan nama: "; ;  palette ("Biru") ;  fflush(stdin) ;  getline (cin,nama) ;  palette("") ;  nomor = search(nama, struk) ;    if (nomor == 50){  ccout("Merah", "\n Nama tidak ditemukan.") ;  ccout("Kuning", " [Tekan apa saja.]") ;  getch() ;}    else{  konfCheckout(struk, nomor) ;}  break ;}  }  char konfCheckout(dataPengunjung struk[], int index){  // var untuk menampung jawaban user  char opsi;    cout << " Lanjutkan Check-out? (y/n)" << endl ;  opsi = getch() ;  opsi = (opsi == 'y' || opsi == 'Y') ? 'y' : 'n' ;    if (opsi == 'y'){hapusData(struk, index) ;  system ("CLS") ;  banner () ;  palette("Hijau") ;  cout << "\t Checkout Berhasil." << endl ;  palette("Kuning") ;  cout << "\t [Tekan apa saja.]" << endl ;  palette("") ;  getch() ;}  return opsi;  }  /\*------------------------ TAMPILKAN DATA ------------------------\*/  // Data Keseluruhan  void dataTampil(dataPengunjung struk[]){    // Array banyaknya kamar pada tiap lantai  int lantai[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0} ;    // var index  int index ;    // var Inputan Menu Advance  char responMenu2 ;    for (int l = 0; l < 10; l++) {  clear() ;    // Vanish lantai 4  if (l == 3) {continue ;}  else if ((l+1) \* 100 > struk[index].nomorKamar) {break ;}  else {cout << "\t\tLANTAI " << l+1 << endl;}      for (int i = 0; i < 20; i++) {  if (struk[index].nomorKamar == 0) {break;}  // Mengatasi penulisan kamar kosong    if (struk[index].nomorKamar > (l+1) \* 100 + 100) {break;}  // Pemisahan kamar per lantai    cout << " [" << struk[index].nomorKamar << "] ";    if (struk[index].nama == "") {palette("Hijau"); cout << "Kosong\n" << endl;}  else {palette("Merah"); cout << "Tidak Kosong\n" << endl;}    index++ ;  lantai[l]++ ;  palette("") ;}    if (l != 0) {cout << " [<] << \t" ;}  cout << " [X] Papan Ketik" ;  if ((l+1) \* 100 < struk[index].nomorKamar) {cout << " \t\t>> [>]" ;}    responMenu = getch();    switch(responMenu){  // Memunculkan papan ketik  case'x': case'X':  cout << "\n\n" << endl ;  menujuData = papanKetik(struk) ;    if (menujuData == 50){  index -= lantai[l] ;  lantai[l] = 0 ;  l -= 1 ;}    else if (menujuData == 99) {  l = 10 ;}    else {  l = 10 ;  properEditMenu(struk, menujuData) ;}    break;      case -32:    responMenu2 = getch();  // Halaman Sebelumnya  if(responMenu2 == 75){  if(l > 0){  if(l == 4){  // Vanish lantai 4  index -= (lantai[4] + lantai[2]) ;  lantai[4] = 0 ;  lantai[2] = 0 ;  l = 1 ;}    else {  index -= (lantai[l] + lantai[l-1]);  lantai[l] = 0 ;  lantai[l-1] = 0 ;  l -= 2 ;}  }    else{  index -= lantai[l] ;  lantai[l] = 0 ;  l -= 1 ;  }  break;}    // Halaman Selanjutnya  else if(responMenu2 == 77){  if ((l+1) \* 100 < struk[index].nomorKamar) {  break ;}}    default:  index -= lantai[l] ;  lantai[l] = 0 ;  l -= 1 ;  break;}  }  }  // Data Spesifik  void dataTampil(dataPengunjung struk[], int index){  clear();    cout << " .Informasi Kamar " << struk[index].nomorKamar << "." << endl  << " Tipe Kamar: \t\t" << struk[index].tipeKamar << "\n" << endl  << " Nama: \t\t\t" << struk[index].nama << endl  << " Kewarganegaraan: \t" ;  string kewarganegaraan = (struk[index].wni == true) ? "WNI" : "WNA" ;    if(struk[index].nama == ""){cout << "" << endl;}  else {cout << kewarganegaraan << endl;}    cout << " Nomor Pengenal: \t" << struk[index].nomorPengenal << endl  << "\n Tanggal Check-in:\t" << struk[index].tanggalCheckin << endl  << " Tanggal Check-out:\t" << struk[index].tanggalCheckout << endl  << "\n Catatan:\n " << struk[index].catatan << endl;  }  // Nama pemilik data berisi  void dataTampilNama(dataPengunjung struk[], string pengurutan, bool state){  char responMenu2;    clear();    if(state == true){    if (pengurutan == "ASC"){  ccout("Hijau", "\n\t\t\t\t ## Menampilkan data A-Z ##");  ccout("Abu-abu", "\t\t\tTekan Tombol Bawah untuk menampilkan data Z-A");  selectionSort(struk);  }    else if (pengurutan == "DESC"){  ccout("Hijau", "\n\t\t\t\t ## Menampilkan data Z-A ##");  ccout("Abu-abu", "\t\t\tTekan Tombol Bawah untuk menampilkan data A-Z");  insertionSort(struk);  }    cout << "=========================" << endl;  for(int i = 0; i < 49; i++){  if (struk[i].nama != "" && struk[i].nomorKamar != 0){  cout << "[" << struk[i].nomorKamar << "] "  << struk[i].nama << endl;  }}  bubbleSort(struk);  cout << "=========================\n\n" << endl  << " [X] Papan Ketik" << endl;      responMenu = getch();  switch(responMenu){  case -32:  responMenu2 = getch();    // REKURSIF  if (responMenu2 == 80 && pengurutan == "ASC"){  dataTampilNama(struk, "DESC", true);}    else if (responMenu2 == 80 && pengurutan == "DESC"){  dataTampilNama(struk, "ASC", true);}    else {  dataTampilNama(struk, pengurutan, true);}  break;      case 'x': case 'X':  menujuData = papanKetik(struk) ;    if (menujuData == 50){  }  else if (menujuData == 99){  break;}    else {  properEditMenu(struk, menujuData) ;}  }  }  }  /\*------------------------ MENU EDIT ------------------------\*/  void properEditMenu(dataPengunjung struk[], int index){  // var untuk pilihan menu user  char opsi ;    // var untuk menampung kamar pindahan  int kamarLain ;    // var untuk pengeditan  char kewarganegaraan ;  string nama ;      if(struk[index].nama == ""){  system ("CLS") ;  banner () ;    ccout("Merah", "\t Data tidak ada.") ;  ccout("Kuning", "\t[Tekan apa saja.]") ;  getch() ;}    else{  do{  dataTampil(struk, index) ;  cout << "\n Ubah: " << endl  << " [1] Check-out" << "\t\t\t"  << " [5] Ubah Kewarganegaraan" << endl  << " [2] Pindah Kamar" << "\t\t"  << " [6] Ubah Waktu Check-in" << endl  << " [3] Ubah Nama" << "\t\t\t"  << " [7] Ubah Waktu Check-out" << endl  << " [4] Ubah Nomor Pengenal" << "\t"  << " [8] Ubah Catatan\n" << endl  << "\t\t[0] Kembali ke menu awal\n" << endl;  opsi = getch() ;  switch(opsi){  case '1':  opsi = konfCheckout(struk, index) ;  if (opsi == 'y'){opsi = '3' ;}  break ;    case '2':  cout << " Pindahkan ke kamar nomor: " ;  cin >> kamarLain ;  kamarLain = search(struk, kamarLain) ;  if (kamarLain == 50){  ccout("Merah", "\t\t Kamar Tidak Tersedia.");  ccout("Kuning", "\t\t [Tekan apa saja.]") ;  getch() ;}    else if (struk[kamarLain].nama != ""){  ccout ("Merah","\t\t Kamar Sedang Ditempati") ;  cout << " Tukar Kamar? (y/n)" << endl ;  opsi = getch() ;  opsi = (opsi == 'y' || opsi == 'Y') ? '0' : 'n';  if (opsi == '0'){  pindahKamar(struk, index, kamarLain) ;  }}    else{  opsi = '0' ;  pindahKamar(struk, index, kamarLain) ;}    break ;    case '3':  cout << "\n Ubah nama menjadi: "; fflush(stdin) ;  getline(cin,nama) ;  if(nama != ""){struk[index].nama = nama ;}  break ;    case '4':  cout << "\n Ubah nomor pengenal menjadi: " ;  cin >> struk[index].nomorPengenal ;  break ;    case '5':  cout << "\n Opsi Kewarganegaraan:" << endl  << " [0] WNA" << endl  << " [1] WNI" << endl  << " >> " ;  kewarganegaraan = getche() ;  struk[index].wni = (kewarganegaraan == '1')? true : false;  break ;    case '6':  cout << "\n Ubah tanggal check-in: "; fflush(stdin) ;  getline(cin,struk[index].tanggalCheckin) ;  break ;    case '7':  cout << " Ubah tanggal check-out: " ; fflush(stdin) ;  getline(cin,struk[index].tanggalCheckout) ;  break ;    case '8':  cout << "\n Catatan: "; fflush(stdin) ;  getline(cin,struk[index].catatan) ;  break;    case '0':  default:  break;}  }  while(opsi != '0');}  }  /\*------------------------ OLAH HOTEL ------------------------\*/  void editHotel(dataPengunjung struk[]){  // var index kosong, nomor kamar dituju, index  int slot, nomorKamarBaru, index;    // var pilihan menu user, konfirmasi hapus user  char opsi, konfHapus ;    // var tipe kamar  string tipeKamar ;    // var konfirmasi user  bool konfirmasi ;    opsi = menuMetode(2);  switch(opsi){  case '1':  clear() ;    cout << " Ketik nomor kamar baru: " ;  cin >> nomorKamarBaru ;  slot = search(struk, nomorKamarBaru) ;  if (slot == 50){  slot = search(struk, 0) ;  tipeKamar = pilihTipeKamar() ;    struk[slot].nomorKamar = nomorKamarBaru ;  struk[slot].tipeKamar = tipeKamar ;    ccout("Hijau", "\t\t Kamar Berhasil ditambahkan.");  ccout("Kuning", "\t\t [Tekan apa saja.]" );  getch() ;}    else{  ccout("Merah", "\t\t Kamar Sudah ada." );  ccout("Kuning", "\t\t [Tekan apa saja.]" );  getch() ;}  break;    case '2':  clear() ;    index = papanKetik(struk) ;    if (index == 50){cout << endl;}  else if (struk[index].nama == ""){  do{  dataTampil(struk, index) ;  cout << " Olah Kamar: " << endl  << " [1] Ubah Jenis Kamar" << endl  << " [2] Hapus Kamar" << endl  << " [3] Kembali" << endl;  opsi = getch() ;    switch(opsi){  case '1':  tipeKamar = pilihTipeKamar();  struk[index].tipeKamar = tipeKamar ;  break;    case '2':  cout << " Hapus Kamar? (y/n)" << endl ;  konfHapus = getch() ;  konfirmasi = (konfHapus == 'y' || konfHapus == 'Y')? true:false;    if (konfirmasi == true){  struk[index].nomorKamar = 0 ;  struk[index].tipeKamar = "" ;  for (int i = index; i < 48; i++){  swap(struk[i], struk[i+1]);} ;  opsi = '3' ;}  else {break;}  break;}    }while(opsi != '3');  }      else{  ccout("Merah", "\t\t Kamar Sedang Ditempati.") ;  ccout("Kuning", "\t\t [Tekan apa saja.]") ;  getch() ;}  break;    }  bubbleSort(struk);  }  /\*------------------------ DEFAULT ------------------------\*/  // Nama default yang diberikan  void namaDefault(dataPengunjung struk[]){  for (int i = 0; i < 50; i++) {struk[i].nama = "";}  }  // Kamar Default yang diberikan  void kamarDefault(dataPengunjung struk[]){  int kamar = 100 ;  int index = 0 ;    for (int i = 1; i < 6; i++){    if (kamar == 400){kamar = 405;}    else{  for (int j = 1; j < 6; j++){  kamar += 1 ;  struk[index].nomorKamar = kamar ;  struk[index].tipeKamar = "Standard Room";  index++ ;}}    kamar += 95 ;}    struk[index].nomorKamar = kamar + 1 ;  index++ ;    struk[index].nomorKamar = 701 ;  index++ ;    for (int i = index; i < 50; i++){struk[i].nomorKamar = 0;}  } |

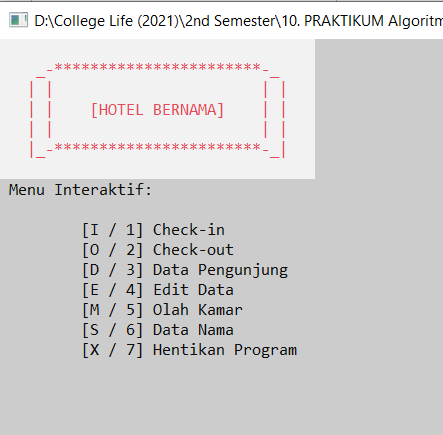
\* Di Dev C++ Lumayan rapi bang, di sini berhamburan :D

Link Github:

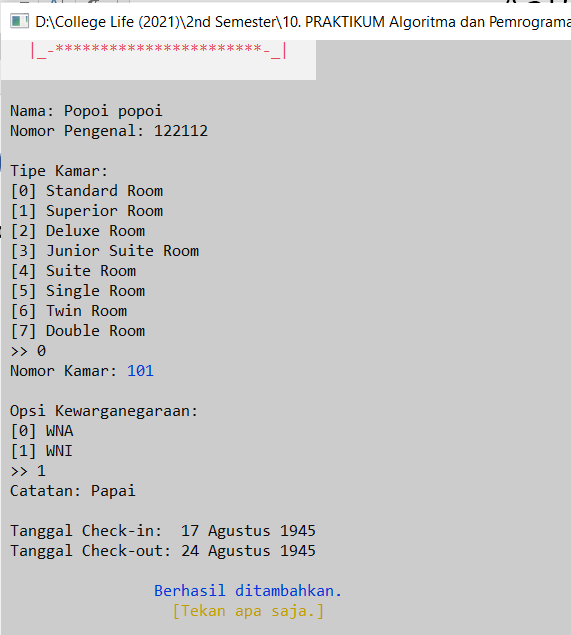
<https://github.com/Viabelous/PRAKTIKUM-ALPRO-SMST-2/blob/31b5456fd9ec566abe921b2e46e8d75fb2455afc/POSTTEST%205.cpp>

# **OUTPUT PROGRAM**

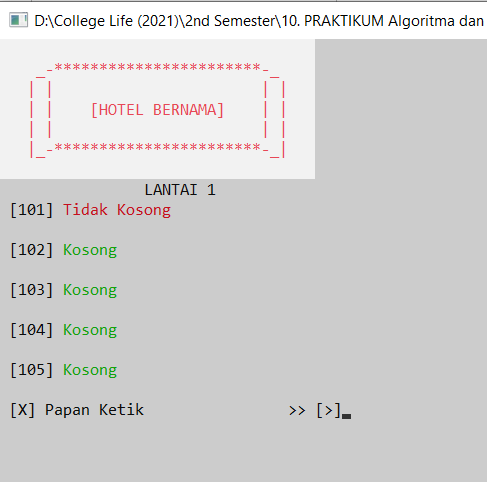
1. Tampilan Awal Program



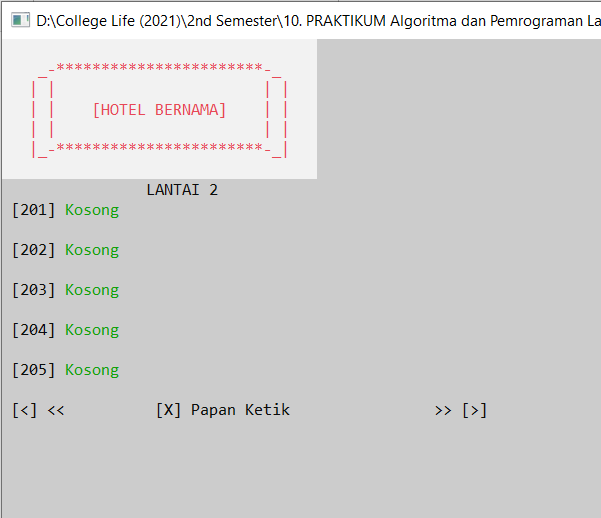
1. Tampilan Check-in



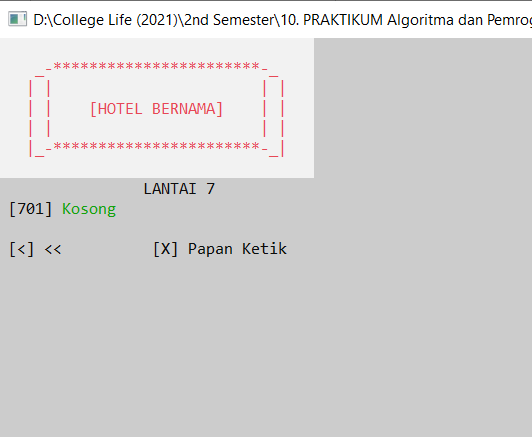
1. Tampilan Data Pengunjung (Lantai 1)



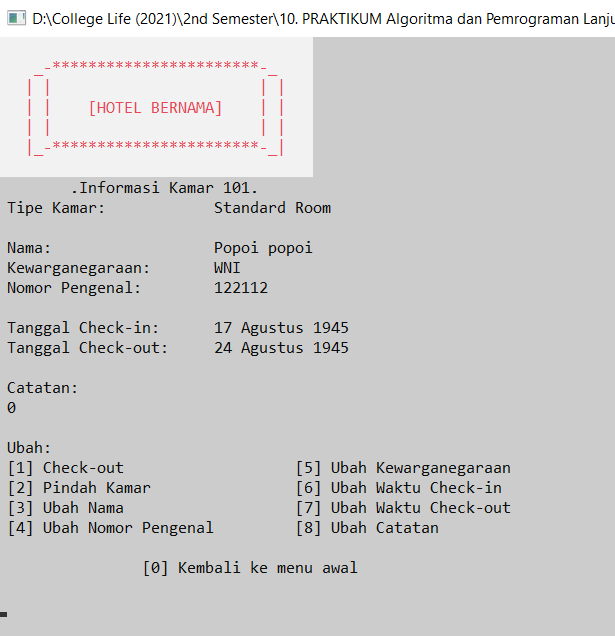
1. Tampilan Data Pengunjung (Lantai 2)



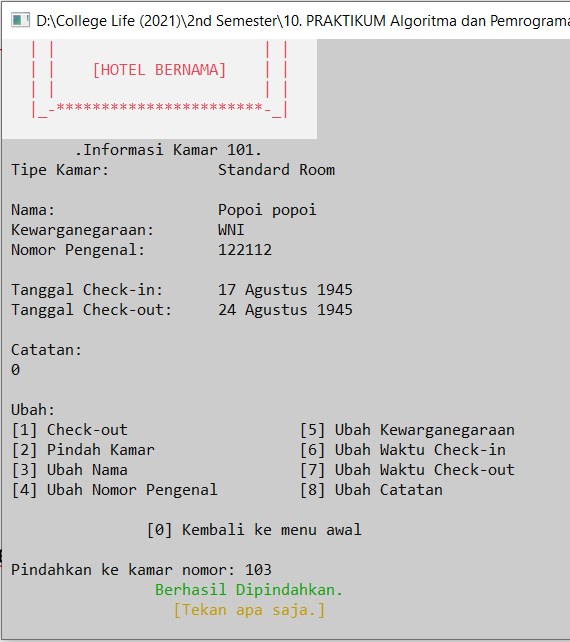
1. Tampilan Data Pengunjung (Lantai Terakhir)



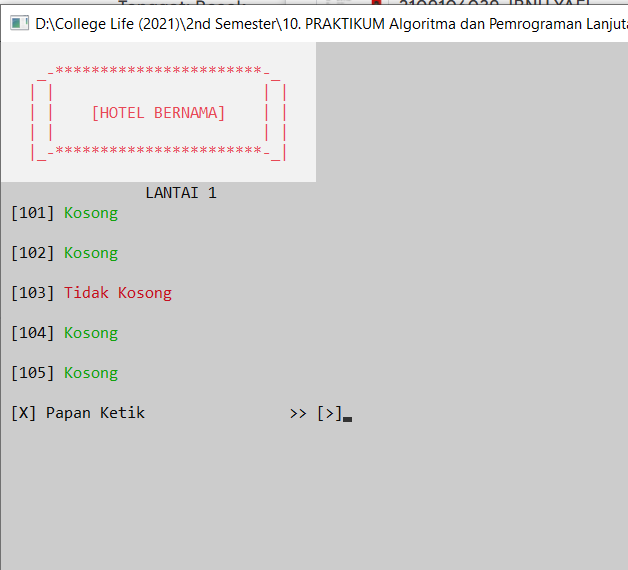
1. Tampilan Informasi Pengunjung



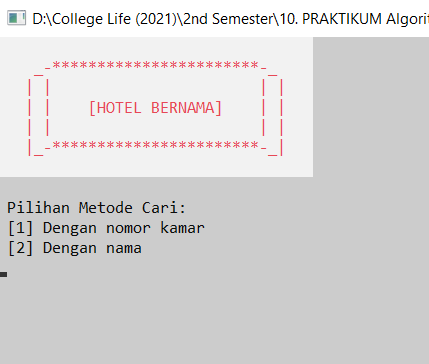
1. Tampilan pilihan pindah kamar



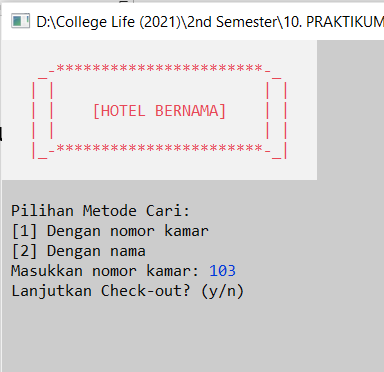
1. Tampilan Data Pengunjung (Setelah pindah kamar)



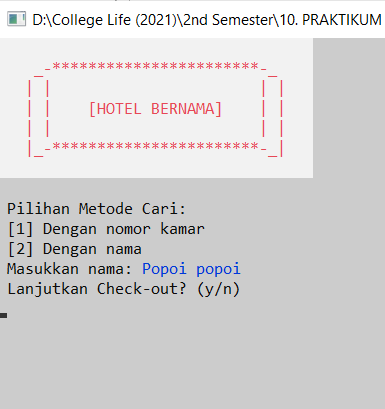
1. Tampilan menu Check-out



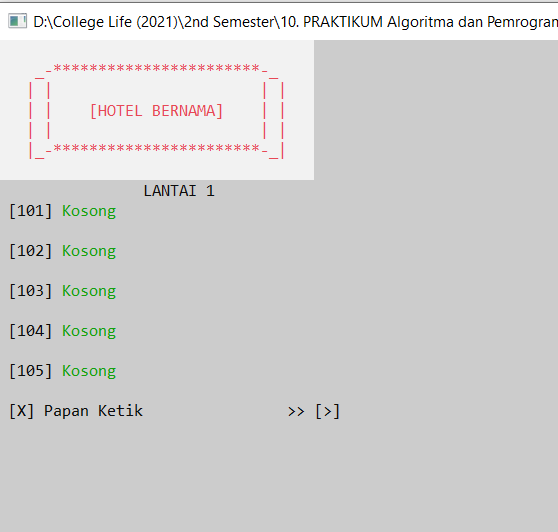
1. Tampilan pilihan check-out dengan nomor kamar



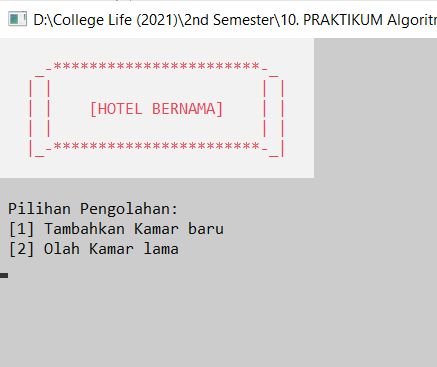
1. Tampilan pilihan check-out dengan nama pengunjung



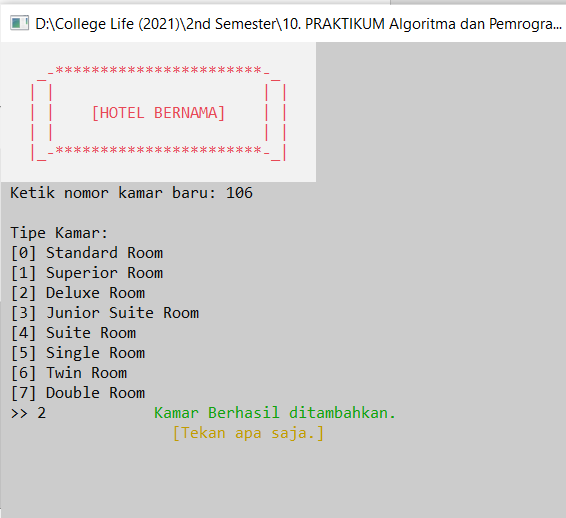
1. Tampilan Data Pengunjung (Setelah Check-out)



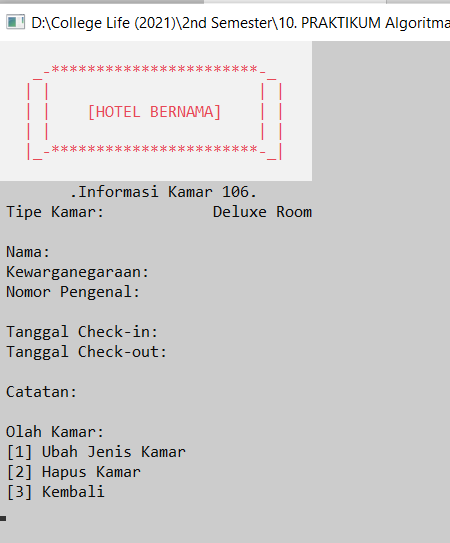
1. Tampilan Menu Olah Kamar



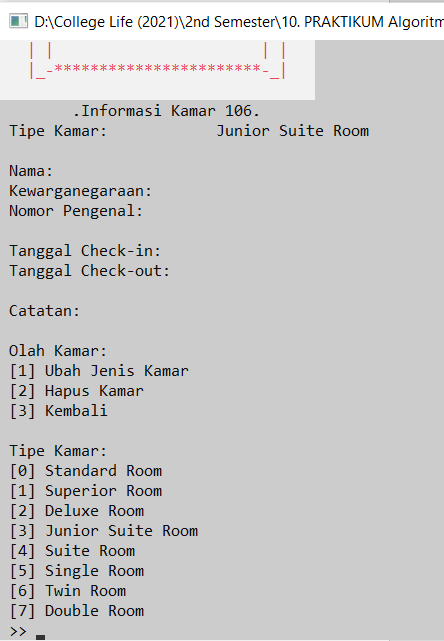
1. Tampilan pilihan tambah kamar baru



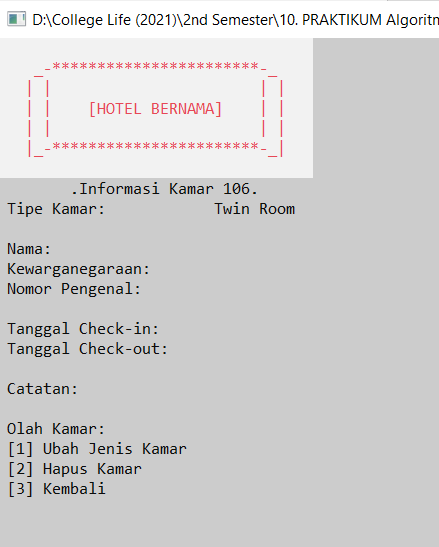
1. Tampilan pilihan olah kamar lama



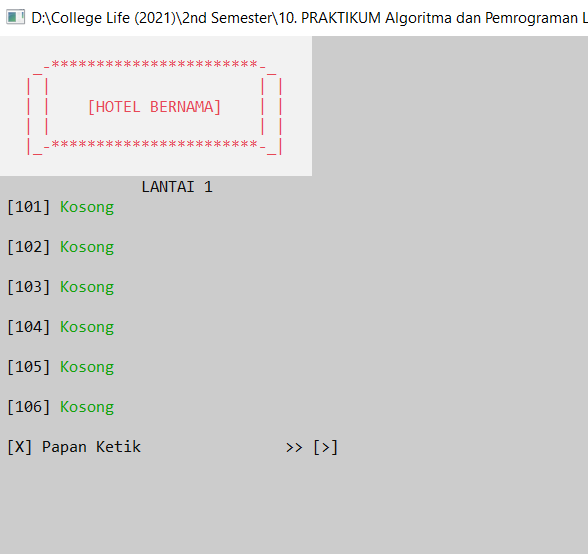
1. Tampilan pilihan ubah jenis kamar



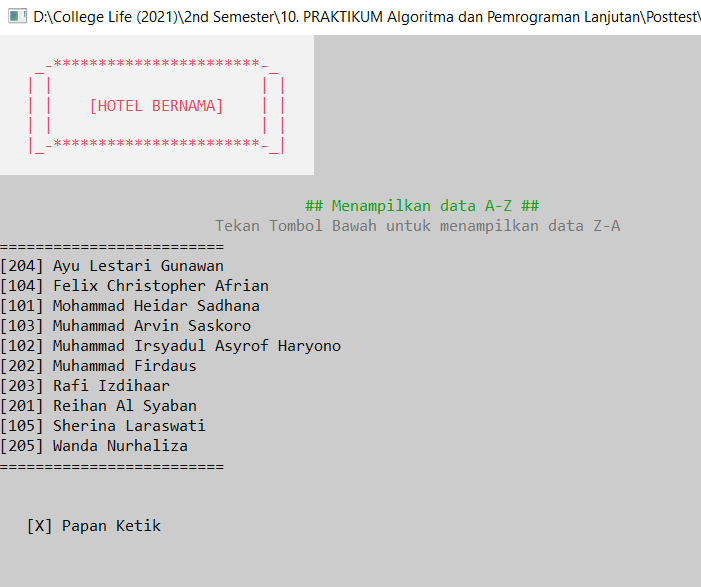
1. Tampilan informasi data setelah ubah jenis kamar



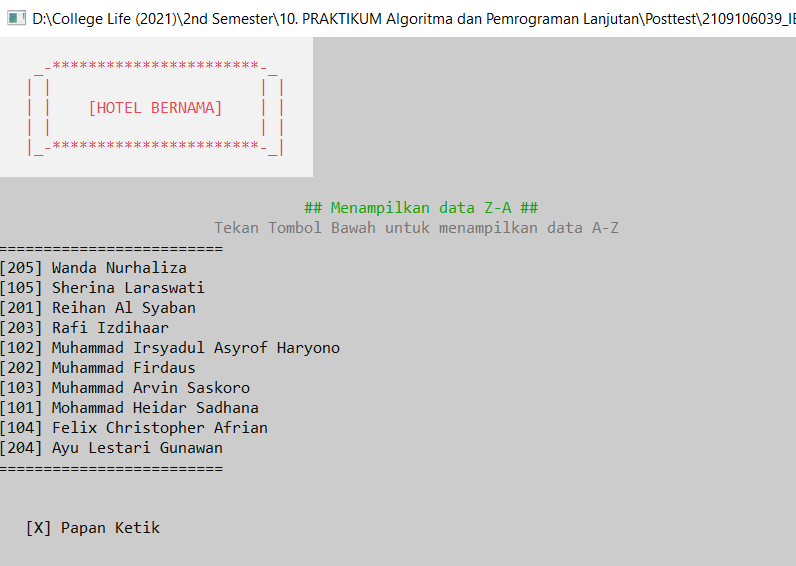
1. Tampilan Data setelah penambahan kamar



1. Tampilan Data Nama A-Z (Setelah Diisi)



1. Tampilan Data Nama Z-A



1. Tampilan program dihentikan

