# KATA PENGANTAR

Segala puji dan syukur yang sedalam-dalamnya dipanjatkan ke khadirat Allah SWT yang telah memberikan karunia dan hidayah-Nya sehingga penulis dapat menyelesaikan penelitian ini.

Penulis menyadari sepenuhnya bahwa dalam penulisan penelitian ini masih terdapat kekurangan, karena itu penulis terbuka untuk menerima saran dan kritik yang sifatnya membangun dari pembaca agar dapat digunakan sebagai bahan kajian untuk pembuatan karya ilmiah yang lebih baik. Walaupun demikian, penulis telah berusaha semaksimal mungkin untuk dapat menyelesaikan penelitian ini dengan harapan dapat memberikan manfaat khususnya dalam wawasan ilmu pengetahuan. Tidak lupa penulis pun mengucapkan banyak terima kasih atas segala.

Penulis berharap tulisan ini dapat bermanfaat bagi semua pihak dan semoga hasil penelitian ini dapat dilanjutkan sehingga didapatkan hasil yang bisa lebih baik dari yang telah penulis susun.

Bandung, Januari 2017

Penulis

# DAFTAR ISI

[KATA PENGANTAR i](#_Toc472499149)

[DAFTAR ISI ii](#_Toc472499150)

[BAB 1 PENDAHULUAN 3](#_Toc472499151)

[1.1. Latar Belakang Masalah 3](#_Toc472499152)

[1.2. Maksud dan Tujuan 3](#_Toc472499153)

[1.2.1. Maksud 3](#_Toc472499154)

[1.2.2. Tujuan 3](#_Toc472499155)

[BAB 2 ANALISIS DAN PEMBAHASAN 4](#_Toc472499156)

[2.1. Tentang Program 4](#_Toc472499157)

[2.2. Algoritma 6](#_Toc472499158)

[2.3. Program 28](#_Toc472499159)

[2.4. Tampilan Layar 40](#_Toc472499160)

[1.2.3. User Login (admin:admin) 40](#_Toc472499161)

[1.2.4. Percobaan salah (hanya 3x) 41](#_Toc472499162)

[1.2.5. Menu Utama 41](#_Toc472499163)

[1.2.6. Validasi Menu Utama 41](#_Toc472499164)

[1.2.7. Menu ke-1 (isi data tamu) 42](#_Toc472499165)

[1.2.8. Menu ke-2 (Cari data) 42](#_Toc472499166)

[1.2.9. Menu ke-2 (cari kode) 42](#_Toc472499167)

[1.2.10. Menu ke-2 (cari KTP) 42](#_Toc472499168)

[1.2.11. Menu ke-2 (cari Nama) 43](#_Toc472499169)

[1.2.12. Menu ke-2 (cari harga) 43](#_Toc472499170)

[1.2.13. Menu cari harga (lebih besar dari) 43](#_Toc472499171)

[1.2.14. Menu cari harga (kurang dari) 43](#_Toc472499172)

[1.2.15. Validasi Menu Harga 43](#_Toc472499173)

[1.2.16. Menu Tampil (telah terurut secara ascending) 43](#_Toc472499174)

[1.2.17. Terurut (Kode) 44](#_Toc472499175)

[1.2.18. Terurut (KTP) 44](#_Toc472499176)

[1.2.19. Terurut (Nama case-sensitive, kapital didahulukan) 44](#_Toc472499177)

[1.2.20. Terurut (Harga) 44](#_Toc472499178)

[1.2.21. Simpan ke file (Menu utama ke-4) 44](#_Toc472499179)

[1.2.22. Hapus file (Menu utama ke-5) 45](#_Toc472499180)

[BAB 3 Kontribusi 47](#_Toc472499181)

[3.1. Kontribusi 47](#_Toc472499182)

# PENDAHULUAN

## Latar Belakang Masalah

Makalah ini membahas tentang hasil laporan aplikasi **Reservasi Hotel**. Struktur makalahnya yaitu: Pendahuluan, Algoritma Pemrograman, Daftar Program, Tampilan Layar Program dan kontribusi pengerjaan program (sub-rutin, fitur)

Setelah program ini berhasil terkompilasi dan telah melalui tahap debugging, resepsionis dapat mengoperasikan data untuk memeriksa kebenaran data tamu tersebut atau menambahkan data baru. Program ini dibuat untuk memudahkan resepsionis dalam pencatatan riwayat tamu yang berkunjung tanpa harus dilakukan secara manual.

## Maksud dan Tujuan

### Maksud

Maksud dari penelitian ini untuk menciptakan program berbasis desktop replika dari **Reservasi Hotel** dengan bahasa Pascal.

### Tujuan

Tujuan dari penelitian ini antara lain :

1. Untuk menyelesaikan salah satu tugas besar mata kuliah Algoritma dan Pemograman sekaligus sebagai nilai tambahan untuk UAS.
2. Dapat bermanfaat bagi penulis dan pembaca yang membutuhkan referensi ataupun sarana menambah wawasan pemrograman Pascal.
3. Melatih *problem-solving* penulis untuk kasus yang diangkat yaitu **Aplikasi Reservasi Hotel.**
4. Memahami cara kerja penerapan bahasa pemrograman di dunia nyata.

# ANALISIS DAN PEMBAHASAN

1. Tentang Program

Program yang kami buat yaitu dapat mengolah data riwayat tamu atau pengunjung di lokasi penilitian **Ibis Hotel**.

Demikian batasan-batasan yang dibutuhkan: *struktur (runtunan, percabangan & perulangan), tipe data bentukan (Array of Record, Tamu), tipe data File (penyimpanan data di luar program, agar data mudah diproses ulang), perhitungan Check-In dan Check-Out (tipe TDateTime), perhitungan pajak + total harga kamar (diperlukan tipe data double dan longint, untuk menerima data lebih banyak), prosedur dan fungsi (untuk setiap proses: isiData, tampilData, urutData, etc), serta menggunakan algoritma pengurutan Bubble dan Selection, algoritma pencarian Binary dan Sequential.*

Program ini terdiri dari 19 prosedur dan 3 fungsi:

1. bukaData  
   Prosedur ***membuka / membuat file*** baru dengan mode (1: Membaca, 2: Menulis ulang / timpa data sebelumnya)
2. loadData  
   Prosedur untuk ***mengembalikan data dari file*** ke variabel (memori).
3. simpanData  
   Prosedur untuk ***menyimpan data dari variabel*** (memori) ke file.
4. resetData  
   Prosedur untuk ***menghapus*** semua data
5. hargaTipe  
   Fungsi untuk menentukan ***harga*** tipe kamar
6. jenisTipe  
   Fungsi untuk menentukan ***jenis*** tipe kamar
7. kodeTipe  
   Fungsi untuk menentukan ***kode*** tipe kamar
8. isiData  
   Prosedur untuk ***mengisi*** data tamu
9. tampilData  
   Prosedur untuk ***menampilkan*** data tamu
10. tukar  
    Prosedur untuk ***menukar*** data tamu (dengan by reference)
11. bubbleSort  
    Prosedur untuk ***menyortir*** data tamu dengan metode ***Bubble Sort***
12. selectionSort  
    Prosedur untuk ***menyortir*** data tamu dengan metode ***Selection Sort***
13. cariKode  
    Prosedur untuk ***mencari*** data Kode tamu dengan metode ***Binary Search (karena unik)***
14. cariNoKtp  
    Prosedur untuk ***mencari*** data KTP tamu dengan metode ***Binary Search (karena unik)***
15. cariNama  
    Prosedur untuk ***mencari*** data Nama tamu dengan metode ***Sequential Search (karena tidak unik)***
16. cariHarga  
    Prosedur untuk ***mencari*** data Harga yang dibayarkan tamu dengan metode ***Sequential Search (karena tidak unik)***
17. menuPilihan  
    Prosedur ***menu utama*** (Isi data, cari data, tampil data terurut, simpan atau hapus ke file)
18. menuCari  
    Prosedur ***submenu cari*** (kode, ktp, nama atau harga)
19. menuCariHarga  
    Prosedur ***submenu harga dari menu cari*** (lebih besar atau lebih kecil)
20. menuTampil  
    Prosedur ***submenu tampil***  (urut sesuai kode, ktp, nama atau harga)
21. pilihan  
    Prosedur yang ***menggabungkan*** semua prosedur di atas
22. login  
    Prosedur ***login*** untuk masuk sebagai admin
23. Algoritma

{File purwarupa}

Procedure bukaData(I/O database: hotel; Input nama: String; Input mode: Integer) // mode 1: membaca 2: menulis

{I.S: database belum terdefinisi}

{F.S: menentukan nilai database dan mode proses file}

Kamus: {Tidak ada}

Algoritma:

Assign(database, nama)

{ memulai pemeriksaan file }

{$I-} Reset(database)

{$I+} If (IOResult <> 0) Then

Output('File belum dibuat!')

Rewrite(database)

EndIf { selesai pemeriksaan file }

DependOn (mode)

1: Reset(database) // untuk membaca

2: Rewrite(database) // untuk menulis ulang

EndDependOn

EndProcedure // bukaData

Procedure loadData(I/O database: hotel I/O n: Integer I/O reservasi: larik\_tamu)

{I.S.: database telah terdefinisi, n dan reservasi belum terdefinisi}

{F.S.: menentukan nilai n dan reservasi}

Kamus:

temp: Tamu

Procedure bukaData(I/O database: hotel; Input nama: String; Input mode: Integer)

Algoritma:

bukaData(database, BUKUTAMU, 1) // buka file

n <- 0

Output('Loading data sebelumnya...')

Seek(database, 0) // tentukan posisi file ke indeks awal

While Not Eof(database) Do

Inc(n)

Input(database, temp)

reservasi(n) <- temp

If (temp.kode <> '') Then

// delay(700) Output('sukses')

EndIf

EndWhile

Close(database)

EndProcedure //End loadData

Procedure simpanData(I/O database: hotel I/O n: Integer I/O reservasi: larik\_tamu)

{I.S: semua variable telah terdefinisi}

{F.S: simpan data reservasi ke dalam file database}

Kamus:

i : Integer

temp: Tamu

Procedure bukaData(I/O database: hotel; Input nama: String; Input mode: Integer)

Algoritma:

bukaData(database, BUKUTAMU, 2) // siap2 menulis

For i <- 1 To n Do

temp <- reservasi[i]

Output(database, temp)

EndFor

Close(database)

EndProcedure // End simpanData

Procedure resetData(I/O database: hotel I/O reservasi: larik\_tamu)

{I.S: semua variable telah terdefinisi}

{F.S: menghapus semua data}

Kamus:

riwayat: hotel

temp : Tamu

Procedure bukaData(I/O database: hotel; Input nama: String; Input mode: Integer)

Algoritma:

bukaData(database, BUKUTAMU, 2) // menimpa yang sebelumnya

Close(database)

EndProcedure // End resetData

{ End File purwarupa }

Function hargaTipe(Input tipe: Integer): longint

{I.S.: tipe telah terdefinisi}

{F.S.: mengembalikan harga tipe kamar}

Kamus: {Tidak ada}

Algoritma:

{ harga relatif perhari akan selalu berubah }

DependOn (tipe)

1 : hargaTipe <- 510509

2 : hargaTipe <- 510509

3 : hargaTipe <- 598209

4 : hargaTipe <- 598209

EndDependOn

End //endfunction

Function jenisTipe(Input tipe: Integer): String

{I.S.: tipe telah terdefinisi}

{F.S.: mengembalikan rincian tipe kamar}

Kamus: {Tidak ada}

Algoritma:

DependOn (tipe)

1: jenisTipe <- 'Standard Twin'

2: jenisTipe <- 'Standard Queen'

3: jenisTipe <- 'Standard Twin + Breakfast'

4: jenisTipe <- 'Standard Queen + Breakfast'

EndDependOn

EndFunction // jenisTipe

Function kodeTipe(Input tipe: Integer): String

{I.S.: tipe telah terdefinisi}

{F.S.: mengembalikan kode tipe kamar}

Kamus: {Tidak ada}

Algoritma:

DependOn (tipe)

1 : kodeTipe <- 'STW' // standard twin

2 : kodeTipe <- 'STQ' // standard queen

3 : kodeTipe <- 'SWB' // standard twin + breakfast

4 : kodeTipe <- 'SQB' // standard queen + breakfast

EndDependOn

EndFunction // kodeTipe

Procedure isiData(I/O reservasi: larik\_tamu I/O n: Integer)

{I.S.: reservasi dan n belum terdefinisi}

{F.S.: menentukan nilai reservasi dan n}

Kamus:

lagi : Char

i, tipe: Integer

kode : String

dd, mm, yy,

hh, nn, ss, ms: Word // menyimpan tanggal check-In dan check-out untuk di-Format

Algoritma:

i <- 0

Repeat

Inc(n)

Output('Memasukkan Data Pemesanan Kamar ke-', n)

Output('======================================')

Output('NO KTP : ') Input(reservasi(n).ktp)

Output('Nama Lengkap : ') Input(reservasi(n).nama)

Output('Alamat : ') Input(reservasi(n).alamat)

Output('No Telpon : ') Input(reservasi(n).telp)

Output('====================================')

Output('| Harga Tipe Kamar |')

Output('====================================')

Output('| 1: Twin Rp. 510.509 |')

Output('| 2: Queen Rp. 510.509 |')

Output('| 3: Twin + Breakfast Rp. 598.209 |')

Output('| 4: Queen + Breakfast Rp. 598.209 |')

Output('====================================')

Output('Tipe Kamar : ') Input(tipe)

While (tipe < 1) Or (tipe > 4) Do

Output('Tipe Kamar Hanya ada dari 1-4, Ulangi Tekan (Enter)')

Input(tipe)

EndWhile

reservasi(n).check\_in <- Now // simpan waktu sekarang

decodeDate(reservasi(n).check\_in, yy, mm, dd)

// decodeTime(reservasi(n).check\_in, hh, nn, ss)

Output('Waktu Check-In : ', Format('%d/%d/%d', [dd, mm, yy]))

Output('Durasi Menginap : ') Input(reservasi(n).lama)

Output('Waktu Check-out : ', Format('%d/%d/%d', [dd + reservasi(n).lama, mm, yy]))

reservasi(n).tipe\_kamar <- jenisTipe(tipe)

reservasi(n).pajak <- (21/100) \* (hargaTipe(tipe) \* reservasi(n).lama) { Pajaknya 21% }

// Output('[debug] hasil pajak: ', reservasi(n).pajak:0:2)

reservasi(n).total\_bayar <- (hargaTipe(tipe) \* reservasi(n).lama) + reservasi(n).pajak

Str(n, kode)

// Output('[debug] kodenya: Str(', n, ', ', kode, ')')

reservasi(n).kode <- Concat(kodeTipe(tipe),'-',kode)

// Output('[debug] hasil Concat: ', reservasi(n).kode)

Output('Kode Pemesanan : ', reservasi(n).kode)

Output('Tipe Kamarnya adalah: ', reservasi(n).tipe\_kamar)

Output('Pajak (21%) : Rp. ', reservasi(n).pajak:0:2)

Output('Total Biaya : Rp. ', reservasi(n).total\_bayar:0:2)

Output('Mau Memasukkan Data Lagi [Y/T]: ')

Repeat

lagi <- upCase(readkey)

Until lagi In ['Y', 'T']

Inc(i) // menambah berpa kali?

Output('data bertambah menjadi ',i)

Until (lagi = 'T')

EndProcedure // isiData

Procedure tampilData(Input reservasi: larik\_tamu; Input n: Integer)

{I.S.: Data tamu(1:n) sudah terdefinisi}

{F.S.: Menampilkan daftar tamu}

Kamus:

i: Integer

Algoritma:

//membuat header tabel

Output(' DAFTAR TAMU ')

Output('----------------------------------------------------------------------------------------------------------------------------------------')

Output('| Kode | KTP | Nama | Tipe Kamar | Lama | Total |')

Output('----------------------------------------------------------------------------------------------------------------------------------------')

//menampilkan data mahasiswa (1:n)

For i <- 1 To n Do

Output('| | | | | | Rp. |')

//menampilkan nomor

Output(reservasi[i].kode)

//menampilkan ktp

Output(reservasi[i].ktp)

//menampilkan Nama

Output(reservasi[i].nama)

//menampilkan tipe kamar

Output(reservasi[i].tipe\_kamar)

//menampilkan lama inap

Output(reservasi[i].lama)

//menampilkan total bayar

Output('Rp. ', reservasi[i].total\_bayar:0:2)

EndFor

//garis penutup tabel

Output('----------------------------------------------------------------------------------------------------------------------------------------')

EndProcedure // tampilData

Procedure tukar(I/O a, b: Tamu)

{I.S.: a dan b telah terdefinisi}

{F.S.: menghasilkan nilai a yang telah bertukar}

Kamus:

temp: Tamu

Algoritma:

temp <- a

a <- b

b <- temp

EndProcedure // tukar

{IF3 - 10116095, 10116101, 10116111, 10116122, 10116135}

Procedure bubbleSort(I/O reservasi: larik\_tamu; Input n: Integer; Input mode: Integer)

{I.S.: reservasi, n dan mode telah terdefinisi}

{F.S.: mengurutkan data reservasi secara ascending}

Kamus:

Procedure tukar(I/O a, b: Tamu)

Algoritma:

{ Bubble sort }

//

// Output('Urutkan data secara (asc / desc): ') Input(metode)

{ Mode }

{ 1. Kode }

{ 2. KTP }

For i <- 1 To (n-1) Do

For j <- n Downto (i+1) Do

DependOn (mode)

1: Begin {kode}

If (reservasi[j].kode < reservasi[j-1].kode) Then

tukar(reservasi[j], reservasi[j - 1])

EndIf

EndCase

2: Begin {KTP}

If (reservasi[j].ktp < reservasi[j-1].ktp) Then

tukar(reservasi[j], reservasi[j - 1])

EndIf

EndCase

EndDependOn

EndFor // langkah

EndFor // tahap

EndProcedure // bubbleSort

Procedure selectionSort(I/O reservasi: larik\_tamu n: Integer mode: Integer)

{I.S.: reservasi, n dan mode telah terdefinisi}

{F.S.: mengurutkan data reservasi secara ascending}

Kamus:

Procedure tukar(I/O a, b: Tamu)

Algoritma:

{ Minimum Selection sort }

//

// Output('Urutkan data secara (asc / desc): ') Input(metode)

{ Mode }

{ 1. Nama }

{ 2. Harga }

For i <- 1 To n Do

min <- i

For j <- (i + 1) To n Do

DependOn (mode)

1: Begin

If (reservasi[j].nama < reservasi[min].nama) Then

min <- j

EndIf

EndCase // 1

2: Begin

If (reservasi[j].total\_bayar < reservasi[min].total\_bayar) Then

min <- j

EndIf

EndCase // 2

EndDependOn

EndFor // langkah

tukar(reservasi[min], reservasi[i])

EndFor // tahap

EndProcedure // selectionSort

Procedure cariKode(reservasi: larik\_tamu n: Integer)

{I.S.: reservasi dan n telah terdefinisi}

{F.S.: menampilkan data kode pemesanan}

Kamus:

awal, tengah, akhir: Integer

kode\_cari: String

ketemu: boolean

temp : larik\_tamu

Procedure tampilData(Input reservasi: larik\_tamu; Input n: Integer)

Algoritma:

//memasukan Kode yang dicari

Output('kode Yang Dicari: ') Input(kode\_cari)

//Proses Pencarian Kode Pemesanan (Binary Search)

ketemu <- False

awal <- 1

akhir <- n

While (Not ketemu) And (awal <= akhir) Do

tengah <- (awal + akhir) Div 2 // cari indeks tengah

If (kode\_cari > reservasi(tengah).kode) Then

awal <- tengah + 1

// Output('[debug] kode\_cari > : ', reservasi(tengah).kode)

EndIf ElseIf (kode\_cari < reservasi(tengah).kode) Then

akhir <- tengah - 1

// Output('[debug] kode\_cari < : ', reservasi(tengah).kode)

EndIf Else

temp(1) <- reservasi(tengah)

// Output('[debug] kode\_cari found : ', reservasi(tengah).kode)

ketemu <- True

EndIf

EndWhile

//menampilkan data apakah Kode Pemesanan yang dicari ditemukan atau tidak

If (ketemu) Then

tampilData(temp, 1)

EndIf Else

Output('Kode ',kode\_cari,' tidak ditemukan!')

EndIf

EndProcedure // cariKode

Procedure cariNoKtp(reservasi: larik\_tamu n: Integer)

{I.S.: reservasi dan n telah terdefinisi}

{F.S.: menampilkan data ktp tamu}

Kamus:

awal, tengah, akhir: Integer

ktp\_cari : String

ketemu : boolean

temp : larik\_tamu

Procedure tampilData(Input reservasi: larik\_tamu; Input n: Integer)

Algoritma:

//memasukan KTP yang dicari

Output('KTP Yang Dicari: ') Input(ktp\_cari)

//Proses Pencarian KTP (Binary Search)

ketemu <- False

awal <- 1

akhir <- n

While (Not ketemu) And (awal <= akhir) Do

tengah <- (awal + akhir) Div 2 // cari indeks tengah

If (ktp\_cari > reservasi(tengah).ktp) Then

awal <- tengah + 1

// Output('[debug] ktp\_cari > : ', reservasi(tengah).ktp)

EndIf ElseIf (ktp\_cari < reservasi(tengah).ktp) Then

akhir <- tengah - 1

// Output('[debug] ktp\_cari < : ', reservasi(tengah).ktp)

EndIf Else

temp(1) <- reservasi(tengah)

// Output('[debug] ktp\_cari found : ', reservasi(tengah).ktp)

ketemu <- True

EndIf

EndWhile

//menampilkan data apakah KTP yang dicari ditemukan atau tidak

If (ketemu) Then

tampilData(temp, 1)

EndIf Else

Output('KTP ', ktp\_cari,' tidak ditemukan!')

EndIf

EndProcedure // cariNoKtp

Procedure cariNama(reservasi: larik\_tamu n: Integer)

{I.S.: reservasi dan n telah terdefinisi}

{F.S.: menampilkan data nama tamu}

Kamus:

cari\_nama: String

i, k: Integer

tempTamu: larik\_tamu

Procedure tampilData(Input reservasi: larik\_tamu; Input n: Integer)

Algoritma:

Output('Masukan nama yang akan dicari: ') Input(cari\_nama)

i <- 0

k <- 0

// Sequential search (karena tidak unik, perlu mencari semua data)

While (i < n) Do

Inc(i)

If (pos(lowerCase(cari\_nama), lowerCase(reservasi[i].nama)) > 0) Then

Inc(k)

tempTamu[k] <- reservasi[i] // simpan ke variabel sementara

// Output('[debug] nama : ', cari\_nama, ' = ', reservasi[i].nama)

EndIf

EndWhile

// Output('[debug] nilai k : ', k)

If (k > 0) Then

tampilData(tempTamu, k)

EndIf Else

Output('Nama yang mengandung ', cari\_nama,' tidak ditemukan!')

EndIf

EndProcedure // cariNama

Procedure cariHarga(reservasi: larik\_tamu n: Integer mode: Integer)

{I.S.: reservasi, n dan mode telah terdefinisi}

{F.S.: menampilkan data harga}

Kamus:

cari\_harga: longint

i, k: Integer

tempTamu: larik\_tamu

Procedure tampilData(Input reservasi: larik\_tamu; Input n: Integer)

Algoritma:

Output('Masukan kisaran harga yang akan dicari: ') Input(cari\_harga)

i <- 0

k <- 0

// Sequential search (karena tidak unik, perlu mencari semua data)

While (i < n) Do

Inc(i)

{ Mode }

{ 1. Lebih dari cari\_harga }

{ 2. Kurang dari dari cari\_harga }

DependOn (mode)

1: Begin

If (cari\_harga < reservasi[i].total\_bayar) Then

Inc(k)

tempTamu[k] <- reservasi[i] // simpan ke variabel sementara

// Output('[debug] harga : Rp. ', reservasi[i].total\_bayar:0:2, ' > Rp.', cari\_harga)

EndIf

EndCase // 1

2: Begin

If (cari\_harga > reservasi[i].total\_bayar) Then

Inc(k)

tempTamu[k] <- reservasi[i] // simpan ke variabel sementara

// Output('[debug] harga : Rp. ', reservasi[i].total\_bayar:0:2, ' < Rp.', cari\_harga)

EndIf

EndCase // 2

EndDependOn

EndWhile

// Output('[debug] nilai k : ', k)

If (k > 0) Then

tampilData(tempTamu, k)

EndIf Else

Output('Nominal harga dengan kisaran Rp. ', cari\_harga,' tidak ditemukan!')

EndIf

EndProcedure // cariHarga

Procedure menuPilihan(I/O menu: Integer)

{I.S.: menu belum terdefinisi}

{F.S.: menghasilkan nilai menu dan menampilkan menu utama}

Kamus: {Tidak ada}

Algoritma:

Output('-----------------------------------------------------------------------')

Output(' Selamat datang di aplikasi reservasi Ibis Hotel ')

Output('Jl. Gatot Subroto No 289, Pusat Kota Bandung, Bandung, Indonesia, 40273')

Output('-----------------------------------------------------------------------')

Output('===================================')

Output(' Menu Pilihan ')

Output('===================================')

Output('1. Isi data reservasi')

Output('2. Menu cari data reservasi')

Output('3. Menu tampil data yang terurut')

Output('4. Simpan data ke file')

Output('5. Hapus data dari file')

Output('0. Keluar')

Output('Pilihan Anda [0-5]: ') Input(menu)

{validasi menu pilihan}

If (menu < 0) Or (menu > 5) Then

Output('Salah Memilih Menu, Ulangi! (Tekan Enter)')

menuPilihan(menu)

EndIf

// Output

// Output('terima kasih telah menggunakan aplikasi kami..')

// Input

EndProcedure // menuPilihan

Procedure menuCari(I/O menu: Integer)

{I.S.: menu belum terdefinisi}

{F.S.: menghasilkan nilai menu dan menampilkan menu cari}

Kamus: {Tidak ada}

Algoritma:

Output(' Menu Pilihan Cari ')

Output('=======================================')

Output('1. Cari Data Berdasarkan Kode Pemesanan')

Output('2. Cari Data Berdasarkan No KTP')

Output('3. Cari Data Berdasarkan Nama')

Output('4. Cari Data Berdasarkan Harga')

Output('0. Keluar')

Output('Pilihan Anda [0-4]: ') Input(menu)

{validasi menu pilihan}

If (menu < 0) Or (menu > 4) Then

Output('Salah Memilih Menu, Ulangi! (Tekan Enter)')

menuCari(menu)

EndIf

// Output

// Output('terima kasih telah menggunakan aplikasi kami..')

// Input

EndProcedure // menuCari

Procedure menuCariHarga(I/O menu: Integer)

{I.S.: menu belum terdefinisi}

{F.S.: menghasilkan nilai menu dan menampilkan menu cari harga}

Kamus: {Tidak ada}

Algoritma:

Output(' Menu Pilihan Cari Harga ')

Output('==============================')

Output('1. Cari Harga Lebih besar dari')

Output('2. Cari harga lebih kecil dari')

Output('0. Keluar')

Output('Pilihan Anda [0-2]: ') Input(menu)

{validasi menu pilihan}

If (menu < 0) Or (menu > 2) Then

Output('Salah Memilih Menu, Ulangi! (Tekan Enter)')

menuCariHarga(menu)

EndIf

// Output

// Output('terima kasih telah menggunakan aplikasi kami..')

// Input

EndProcedure // menuCariHarga

Procedure menuTampil(I/O menu: Integer)

{I.S.: menu belum terdefinisi}

{F.S.: menghasilkan nilai menu dan menampilkan data yang terurut}

Kamus: {Tidak ada}

Algoritma:

Output('Lihat Data Pemesanan Reservasi yang sudah terurut (Asc)')

Output('=======================================================')

Output('1. Urut Berdasarkan Kode Pemesanan')

Output('2. Urut Berdasarkan KTP Pelanggan')

Output('3. Urut Berdasarkan Nama Pelanggan')

Output('4. Urut Berdasarkan Harga')

Output('0. Keluar')

Output('Pilihan Anda [0-4]: ') Input(menu)

{validasi menu pilihan}

If (menu < 0) Or (menu > 4) Then

Output('Salah Memilih Menu, Ulangi! (Tekan Enter)')

menuTampil(menu)

EndIf

// Output

// Output('terima kasih telah menggunakan aplikasi kami..')

// Input

EndProcedure // menuTampil

Procedure pilihan(I/O menu: Integer)

{I.S.: menu belum terdefinisi}

{F.S.: menghasilkan keseluruhan menu utama dan sub-menu}

Kamus:

m: Integer

Procedure loadData(I/O database: hotel I/O n: Integer I/O reservasi: larik\_tamu)

Procedure menuPilihan(I/O menu: Integer)

Procedure isiData(I/O reservasi: larik\_tamu I/O n: Integer)

Procedure simpanData(I/O database: hotel I/O n: Integer I/O reservasi: larik\_tamu)

Procedure resetData(I/O database: hotel I/O reservasi: larik\_tamu)

Procedure bubbleSort(I/O reservasi: larik\_tamu; Input n: Integer; Input mode: Integer)

Procedure selectionSort(I/O reservasi: larik\_tamu n: Integer mode: Integer)

Procedure cariKode(reservasi: larik\_tamu n: Integer)

Procedure cariNoKtp(reservasi: larik\_tamu n: Integer)

Procedure cariNama(reservasi: larik\_tamu n: Integer)

Procedure menuCariHarga(I/O menu: Integer)

Procedure cariHarga(reservasi: larik\_tamu n: Integer)

Procedure menuTampil(I/O menu: Integer)

Procedure tampilData(Input reservasi: larik\_tamu; Input n: Integer)

Algoritma:

Repeat

loadData(database, n, reservasi)

menuPilihan(menu)

// menu <- 1

DependOn (menu)

1: Begin {Input Data Pemesanan reservasi}

isiData(reservasi, n)

simpanData(database, n, reservasi)

End // End 1

2: Begin {Cari Data Pemesanan Reservasi}

Repeat

menuCari(m)

DependOn (m)

1: Begin {Cari Data Berdasarkan Kode Pemesanan}

bubbleSort(reservasi, n, 1)

cariKode(reservasi, n)

EndCase // 1

2: Begin {Cari Data Berdasarkan No ktp}

bubbleSort(reservasi, n, 2)

cariNoKtp(reservasi, n)

EndCase // 2

3: Begin {Cari Data Berdasarkan Nama}

cariNama(reservasi, n)

EndCase // 3

4: Begin {Cari ascending Harga}

Repeat

menuCariHarga(m)

DependOn (m)

1: Begin {Cari Harga Lebih besar dari}

cariHarga(reservasi, n, 1)

EndCase // 1

2: Begin {Cari harga lebih kecil dari}

cariHarga(reservasi, n, 2)

EndCase // 2

EndDependOn

Until (m = 0)

EndCase // 4

EndDependOn // menuCari

Until (m = 0)

EndCase // cari data

3: Begin {Lihat Data Pemesanan reservasi yang sudah terurut}

Repeat

menuTampil(m)

DependOn (m)

1: Begin {Urut ascending Kode Pemesanan}

bubbleSort(reservasi, n, 1)

tampilData(reservasi, n)

EndCase // 1

2: Begin {Urut ascending KTP}

bubbleSort(reservasi, n, 2)

tampilData(reservasi, n)

EndCase // 2

3: Begin {Urut ascending Nama}

selectionSort(reservasi, n, 1)

tampilData(reservasi, n)

EndCase // 3

4: Begin {Urut ascending Harga}

selectionSort(reservasi, n, 2)

tampilData(reservasi, n)

EndCase // 4

EndDependOn

Until (m = 0)

EndCase // 3

4: Begin {Simpan Data}

simpanData(database, n, reservasi)

Output('Data telah berhasil disimpan!')

EndCase // 4

5: Begin {Reset Data}

resetData(database, reservasi)

Output('Data telah berhasil dihapus!')

EndCase // 5

EndDependOn

Until (menu = 0)

EndProcedure // pilihan

Procedure login(user, pass: String)

{I.S.: pass dan user belum terdefinisi}

{F.S.: memvalidasi user dan pass, ketika berhasil masuk ke menuPilihan}

Kamus:

i, x: Integer

u, p: String

Procedure pilihan(I/O menu: Integer)

Algoritma:

x <- 3

Repeat

Output('==================================')

Output('Masukan Username dan Password Anda')

Output('----------------------------------')

Output('Username: ')Input(u)

Output('Password: ')Input(p)

If (user = u) And (pass = p) Then

pilihan(menu)

Break

EndIf Else

Output('Username dan Password Salah, Silahkan Ulangi Lagi!')

EndIf

Dec(x) { x <- x - 1 }

If (x = 0) Then

Output('Maaf Anda Tidak Bisa Login, Karna Lebih dari 3 Kali Salah')

EndIf

Until (x = 0)

EndProcedure // login

Reservasi\_hotel

{I.S.: user memasukan data pemesan dan Tamu}

{F.S.: simpan lalu menampilkan list data pemesan dan Tamu}

Const

USER = 'admin'

PASS = 'admin'

MAKS = 100

BUKUTAMU = 'data.dat'

RIWAYAT = 'riwayat.dat'

Type

{ Tipe data bentukan }

Tamu = Record

kode, ktp, nama, alamat, telp, tipe\_kamar: String

lama: Integer

pajak, total\_bayar: double // agar menerima data lebih banyak

check\_in, check\_out: TDateTime

End

larik\_tamu = array[1..MAKS] Tamu

hotel = File Tamu

{Kamus Global}

Kamus:

reservasi : larik\_tamu

database : hotel

menu, n : Integer

i, j, min : Integer // untuk pengurutan

Procedure login(user, pass: String)

{Algoritma Utama}

Algoritma:

login(USER, PASS)

1. Program

program reservasi\_hotel;

{I.S.: user memasukan data pemesan dan Tamu}

{F.S.: simpan lalu menampilkan list data pemesan dan Tamu}

uses crt, sysutils; // window dan date unit

const

USER = 'admin';

PASS = 'admin';

MAKS = 100;

BUKUTAMU = 'data.dat';

RIWAYAT = 'riwayat.dat';

type

{ Tipe data bentukan }

Tamu = record

kode, ktp, nama, alamat, telp, tipe\_kamar: string;

lama: integer;

pajak, total\_bayar: double; // agar menerima data lebih banyak

check\_in, check\_out: TDateTime;

end;

larik\_tamu = array[1..MAKS] of Tamu;

hotel = File of Tamu;

{Kamus Global}

var

reservasi : larik\_tamu;

database : hotel;

menu, n : integer;

i, j, min : integer; // untuk pengurutan

{File purwarupa}

procedure bukaData(var database: hotel; nama: string; mode: integer); // mode 1: membaca; 2: menulis

{I.S: database belum terdefinisi}

{F.S: menentukan nilai database dan mode proses file}

begin

assign(database, nama);

{ memulai pemeriksaan file }

{$I-} reset(database);

{$I+} if (IOResult <> 0) then

begin

writeln('File belum dibuat!');

rewrite(database); readkey;

end; { selesai pemeriksaan file }

case (mode) of

1: reset(database); // untuk membaca

2: rewrite(database); // untuk menulis ulang

end;

end;//endprocedure

procedure loadData(var database: hotel; var n: integer; var reservasi: larik\_tamu);

{I.S.: database telah terdefinisi, n dan reservasi belum terdefinisi}

{F.S.: menentukan nilai n dan reservasi}

var

temp: Tamu;

begin

bukaData(database, BUKUTAMU, 1); // buka file

n := 0;

writeln('Loading data sebelumnya...');

seek(database, 0); // tentukan posisi file ke indeks awal

while not eof(database) do begin

inc(n);

read(database, temp);

reservasi[n] := temp;

if (temp.kode <> '') then

begin

// delay(700); writeln('sukses');

end;

end;

close(database);

end; //end loadData

procedure simpanData(var database: hotel; var n: integer; var reservasi: larik\_tamu);

{I.S: semua variable telah terdefinisi}

{F.S: simpan data reservasi ke dalam file database}

var

i : integer;

temp: Tamu;

begin

bukaData(database, BUKUTAMU, 2); // siap2 menulis

for i := 1 to n do

begin

temp := reservasi[i];

write(database, temp);

end;

close(database);

end; // end simpanData

procedure resetData(var database: hotel; var reservasi: larik\_tamu);

{I.S: semua variable telah terdefinisi}

{F.S: menghapus semua data}

var

riwayat: hotel;

temp : Tamu;

begin

bukaData(database, BUKUTAMU, 2); // menimpa yang sebelumnya

close(database);

end; // end resetData

{ End File purwarupa }

function hargaTipe(tipe: integer): longint;

{I.S.: tipe telah terdefinisi}

{F.S.: mengembalikan harga tipe kamar}

begin

{ harga relatif perhari akan selalu berubah }

case (tipe) of

1 : hargaTipe := 510509;

2 : hargaTipe := 510509;

3 : hargaTipe := 598209;

4 : hargaTipe := 598209;

end; //endchase

end; //endfunction

function jenisTipe(tipe: integer): string;

{I.S.: tipe telah terdefinisi}

{F.S.: mengembalikan rincian tipe kamar}

begin

case (tipe) of

1: jenisTipe := 'Standard Twin';

2: jenisTipe := 'Standard Queen';

3: jenisTipe := 'Standard Twin + Breakfast';

4: jenisTipe := 'Standard Queen + Breakfast';

end; //endchase

end; //endfunction

function kodeTipe(tipe:integer): string;

{I.S.: tipe telah terdefinisi}

{F.S.: mengembalikan kode tipe kamar}

begin

case (tipe) of

1 : kodeTipe := 'STW' ; // standard twin

2 : kodeTipe := 'STQ' ; // standard queen

3 : kodeTipe := 'SWB' ; // standard twin + breakfast

4 : kodeTipe := 'SQB' ; // standard queen + breakfast

end; //endchase

end; //endfunction

procedure isiData(var reservasi: larik\_tamu; var n: integer);

{I.S.: reservasi dan n belum terdefinisi}

{F.S.: menentukan nilai reservasi dan n}

var

lagi : char;

i, tipe: integer;

kode : string;

dd, mm, yy,

hh, nn, ss, ms: word; // menyimpan tanggal check-in dan check-out untuk di-format

begin

i := 0;

repeat

clrscr;

inc(n);

writeln('Memasukkan Data Pemesanan Kamar ke-',n);

writeln('======================================');

write('NO KTP : '); readln(reservasi[n].ktp);

write('Nama Lengkap : '); readln(reservasi[n].nama);

write('Alamat : '); readln(reservasi[n].alamat);

write('No Telpon : '); readln(reservasi[n].telp);

writeln;

writeln('===================================');

writeln('| Harga Tipe Kamar |');

writeln('===================================');

writeln('|1: Twin Rp. 510.509|');

writeln('|2: Queen Rp. 510.509|');

writeln('|3: Twin + Breakfast Rp. 598.209|');

writeln('|4: Queen + Breakfast Rp. 598.209|');

writeln('===================================');

write('Tipe Kamar : '); readln(tipe);

while (tipe < 1) or (tipe > 4) do

begin

gotoxy(25,18); textcolor(yellow);

write('Tipe Kamar Hanya ada dari 1-4, Ulangi Tekan (Enter)'); readln;

gotoxy(25,18); clreol;

gotoxy(18,17); clreol; textcolor(white); readln(tipe);

end;

reservasi[n].check\_in := Now; // simpan waktu sekarang

decodeDate(reservasi[n].check\_in, yy, mm, dd);

// decodeTime(reservasi[n].check\_in, hh, nn, ss);

writeln('Waktu Check-in : ', format('%d/%d/%d', [dd, mm, yy]));

write('Durasi Menginap : '); readln(reservasi[n].lama);

writeln('Waktu Check-out : ', format('%d/%d/%d', [dd + reservasi[n].lama, mm, yy]));

reservasi[n].tipe\_kamar := jenisTipe(tipe);

reservasi[n].pajak := (21/100) \* (hargaTipe(tipe) \* reservasi[n].lama); { Pajaknya 21% }

// writeln('[debug] hasil pajak: ', reservasi[n].pajak:0:2); readkey;

reservasi[n].total\_bayar := (hargaTipe(tipe) \* reservasi[n].lama) + reservasi[n].pajak;

str(n, kode);

// writeln('[debug] kodenya: str(', n, ', ', kode, ')');

reservasi[n].kode := concat(kodeTipe(tipe),'-',kode);

// writeln('[debug] hasil concat: ', reservasi[n].kode);

writeln('Kode Pemesanan : ', reservasi[n].kode);

writeln('Tipe Kamarnya adalah: ', reservasi[n].tipe\_kamar);

writeln('Pajak (21%) : Rp. ', reservasi[n].pajak:0:2);

writeln('Total Biaya : Rp. ', reservasi[n].total\_bayar:0:2);

writeln;

write('Mau Memasukkan Data Lagi [Y/T]: ');

repeat

lagi := upCase(readkey);

until lagi in ['Y', 'T'];

inc(i); // menambah berpa kali?

writeln;

writeln('data bertambah menjadi ',i);

readkey;

until (lagi = 'T');

end;

procedure tampilData(reservasi: larik\_tamu; n: integer);

{I.S.: Data tamu(1:n) sudah terdefinisi}

{F.S.: Menampilkan daftar tamu}

var

i: integer;

begin

//membuat header tabel

clrscr;

writeln(' DAFTAR TAMU ');

writeln('----------------------------------------------------------------------------------------------------------------------------------------');

writeln('| Kode | KTP | Nama | Tipe Kamar | Lama | Total |');

writeln('----------------------------------------------------------------------------------------------------------------------------------------');

//menampilkan data mahasiswa (1:n)

for i := 1 to n do

begin

gotoxy(1, i + 4);

write('| | | | | | Rp. |');

//menampilkan nomor

gotoxy(3, i + 4); write(reservasi[i].kode);

//menampilkan ktp

gotoxy(14, i + 4); write(reservasi[i].ktp);

//menampilkan Nama

gotoxy(37, i + 4); write(reservasi[i].nama);

//menampilkan tipe kamar

gotoxy(73, i + 4); write(reservasi[i].tipe\_kamar);

//menampilkan lama inap

gotoxy(102, i + 4); write(reservasi[i].lama);

//menampilkan total bayar

gotoxy(111, i + 4); write('Rp. ', reservasi[i].total\_bayar:0:2);

end; //endfor

//garis penutup tabel

gotoxy(1, i + 5); write('----------------------------------------------------------------------------------------------------------------------------------------');

readln;

end;

procedure tukar(var a, b: Tamu);

{I.S.: a dan b telah terdefinisi}

{F.S.: menghasilkan nilai a yang telah bertukar}

var

temp: Tamu;

begin

temp := a;

a := b;

b := temp;

end; // EndProcedure tukar

{IF3 - 10116095, 10116101, 10116111, 10116122, 10116135}

procedure bubbleSort(var reservasi: larik\_tamu; n: integer; mode: integer);

{I.S.: reservasi, n dan mode telah terdefinisi}

{F.S.: mengurutkan data reservasi secara ascending}

begin

{ Bubble sort }

// clrscr;

// write('Urutkan data secara (asc / desc): '); readln(metode);

{ Mode }

{ 1. Kode }

{ 2. KTP }

for i := 1 to (n-1) do

begin

for j := n downto (i+1) do

begin

case (mode) of

1: begin {kode}

if(reservasi[j].kode < reservasi[j-1].kode) then

begin

tukar(reservasi[j], reservasi[j - 1]);

end; // EndIf

end; // EndCase 1

2: begin {KTP}

if(reservasi[j].ktp < reservasi[j-1].ktp) then

begin

tukar(reservasi[j], reservasi[j - 1]);

end; // EndIf

end; // EndCase 3

end; // EndCase mode

end; // EndFor langkah

end; // EndFor tahap

end; // EndProcedure bubbleSort

procedure selectionSort(var reservasi: larik\_tamu; n: integer; mode: integer);

{I.S.: reservasi, n dan mode telah terdefinisi}

{F.S.: mengurutkan data reservasi secara ascending}

begin

{ Minimum Selection sort }

// clrscr;

// write('Urutkan data secara (asc / desc): '); readln(metode);

{ Mode }

{ 1. Nama }

{ 2. Harga }

for i := 1 to n do

begin

min := i;

for j := (i + 1) to n do

begin

case (mode) of

1: begin

if (reservasi[j].nama < reservasi[min].nama) then

begin

min := j;

end;

end; // EndCase 1

2: begin

if (reservasi[j].total\_bayar < reservasi[min].total\_bayar) then

begin

min := j;

end;

end; // EndCase 2

end; // EndCase mode

end; // EndFor langkah

tukar(reservasi[min], reservasi[i]);

end; // EndFor tahap

end; // EndProcedure bubbleSort

procedure cariKode(reservasi: larik\_tamu; n: integer);

{I.S.: reservasi dan n telah terdefinisi}

{F.S.: menampilkan data kode pemesanan}

var

awal, tengah, akhir: integer;

kode\_cari: string;

ketemu: boolean;

temp : larik\_tamu;

begin

//memasukan Kode yang dicari

clrscr;

write('kode Yang Dicari: '); readln(kode\_cari);

//Proses Pencarian Kode Pemesanan (Binary Search)

clrscr;

ketemu := false;

awal := 1; akhir := n;

while (not ketemu) and (awal <= akhir) do

begin

tengah := (awal + akhir) div 2; // cari indeks tengah

if (kode\_cari > reservasi[tengah].kode) then

begin

awal := tengah + 1;

// writeln('[debug] kode\_cari > : ', reservasi[tengah].kode); readkey;

end else if (kode\_cari < reservasi[tengah].kode) then

begin

akhir := tengah - 1;

// writeln('[debug] kode\_cari < : ', reservasi[tengah].kode); readkey;

end else

begin

temp[1] := reservasi[tengah];

// writeln('[debug] kode\_cari found : ', reservasi[tengah].kode); readkey;

ketemu := true;

end;

end;

//menampilkan data apakah Kode Pemesanan yang dicari ditemukan atau tidak

if (ketemu) then

begin

tampilData(temp, 1);

end else

begin

writeln('Kode ',kode\_cari,' tidak ditemukan!');

end;

readln;

end; // EndProcedure cariKode

procedure cariNoKtp(reservasi: larik\_tamu; n: integer);

{I.S.: reservasi dan n telah terdefinisi}

{F.S.: menampilkan data ktp tamu}

var

awal, tengah, akhir: integer;

ktp\_cari : string;

ketemu : boolean;

temp : larik\_tamu;

begin

//memasukan KTP yang dicari

clrscr;

write('KTP Yang Dicari: '); readln(ktp\_cari);

//Proses Pencarian KTP (Binary Search)

clrscr;

ketemu := false;

awal := 1; akhir := n;

while (not ketemu) and (awal <= akhir) do

begin

tengah := (awal + akhir) div 2; // cari indeks tengah

if (ktp\_cari > reservasi[tengah].ktp) then

begin

awal := tengah + 1;

// writeln('[debug] ktp\_cari > : ', reservasi[tengah].ktp); readkey;

end else if (ktp\_cari < reservasi[tengah].ktp) then

begin

akhir := tengah - 1;

// writeln('[debug] ktp\_cari < : ', reservasi[tengah].ktp); readkey;

end else

begin

temp[1] := reservasi[tengah];

// writeln('[debug] ktp\_cari found : ', reservasi[tengah].ktp); readkey;

ketemu := true;

end;

end;

//menampilkan data apakah KTP yang dicari ditemukan atau tidak

if (ketemu) then

begin

tampilData(temp, 1);

end else

begin

writeln('KTP ', ktp\_cari,' tidak ditemukan!');

end;

readln;

end; // EndProcedure cariNoKtp

procedure cariNama(reservasi: larik\_tamu; n: integer);

{I.S.: reservasi dan n telah terdefinisi}

{F.S.: menampilkan data nama tamu}

var

cari\_nama: string;

i, k: integer;

tempTamu: larik\_tamu;

begin

clrscr;

write('Masukan nama yang akan dicari: '); readln(cari\_nama);

i := 0;

k := 0;

// Sequential search (karena tidak unik, perlu mencari semua data)

while (i < n) do

begin

inc(i);

if (pos(lowerCase(cari\_nama), lowerCase(reservasi[i].nama)) > 0) then

begin

inc(k);

tempTamu[k] := reservasi[i]; // simpan ke variabel sementara

writeln('[debug] nama : ', cari\_nama, ' = ', reservasi[i].nama); readkey;

end;

end;

writeln('[debug] nilai k : ', k); readkey;

if (k > 0) then

begin

tampilData(tempTamu, k);

end else

begin

writeln('Nama yang mengandung ', cari\_nama,' tidak ditemukan!');

end;

readln;

end;

procedure cariHarga(reservasi: larik\_tamu; n: integer; mode: integer);

{I.S.: reservasi, n dan mode telah terdefinisi}

{F.S.: menampilkan data harga}

var

cari\_harga: longint;

i, k: integer;

tempTamu: larik\_tamu;

begin

clrscr;

write('Masukan kisaran harga yang akan dicari: '); readln(cari\_harga);

i := 0;

k := 0;

// Sequential search (karena tidak unik, perlu mencari semua data)

while (i < n) do

begin

inc(i);

{ Mode }

{ 1. Lebih dari cari\_harga }

{ 2. Kurang dari dari cari\_harga }

case (mode) of

1: begin

if (cari\_harga < reservasi[i].total\_bayar) then

begin

inc(k);

tempTamu[k] := reservasi[i]; // simpan ke variabel sementara

writeln('[debug] harga : Rp. ', reservasi[i].total\_bayar:0:2, ' > Rp.', cari\_harga); readkey;

end;

end; // EndCase 1

2: begin

if (cari\_harga > reservasi[i].total\_bayar) then

begin

inc(k);

tempTamu[k] := reservasi[i]; // simpan ke variabel sementara

writeln('[debug] harga : Rp. ', reservasi[i].total\_bayar:0:2, ' < Rp.', cari\_harga); readkey;

end;

end; // EndCase 2

end; // EndCase mode

end;

writeln('[debug] nilai k : ', k); readkey;

if (k > 0) then

begin

tampilData(tempTamu, k);

end else

begin

writeln('Nominal harga dengan kisaran Rp. ', cari\_harga,' tidak ditemukan!');

end;

readln;

end;

procedure menuPilihan(var menu: integer);

{I.S.: menu belum terdefinisi}

{F.S.: menghasilkan nilai menu dan menampilkan menu utama}

begin

clrscr;

textColor(15);

writeln('-----------------------------------------------------------------------');

writeln(' Selamat datang di aplikasi reservasi Ibis Hotel ');

writeln('Jl. Gatot Subroto No 289, Pusat Kota Bandung, Bandung, Indonesia, 40273');

writeln('-----------------------------------------------------------------------');

writeln;

writeln('===================================');

writeln(' Menu Pilihan ');

writeln('===================================');

writeln('1. Isi data reservasi');

writeln('2. Menu cari data reservasi');

writeln('3. Menu tampil data yang terurut');

writeln('4. Simpan data ke file');

writeln('5. Hapus data dari file');

writeln('0. Keluar');

write('Pilihan Anda [0-5]: '); readln(menu);

{validasi menu pilihan}

if (menu < 0) or (menu > 5) then

begin

gotoxy(1, 16); textcolor(yellow);

write('Salah Memilih Menu, Ulangi! (Tekan Enter)'); readln;

menuPilihan(menu);

end;

// writeln;

// writeln('terima kasih telah menggunakan aplikasi kami..');

// readln;

end;

procedure menuCari(var menu: integer);

{I.S.: menu belum terdefinisi}

{F.S.: menghasilkan nilai menu dan menampilkan menu cari}

begin

clrscr;

textcolor(15);

writeln(' Menu Pilihan Cari ');

writeln('=======================================');

writeln('1. Cari Data Berdasarkan Kode Pemesanan');

writeln('2. Cari Data Berdasarkan No KTP');

writeln('3. Cari Data Berdasarkan Nama');

writeln('4. Cari Data Berdasarkan Harga');

writeln('0. Keluar');

write('Pilihan Anda [0-4]: ');readln(menu);

{validasi menu pilihan}

if (menu < 0) or (menu > 4) then

begin

gotoxy(1, 9); textcolor(yellow);

write('Salah Memilih Menu, Ulangi! (Tekan Enter)'); readln;

menuCari(menu);

end;

// writeln;

// writeln('terima kasih telah menggunakan aplikasi kami..');

// readln;

end;

procedure menuCariHarga(var menu: integer);

{I.S.: menu belum terdefinisi}

{F.S.: menghasilkan nilai menu dan menampilkan menu cari harga}

begin

clrscr;

textcolor(15);

writeln(' Menu Pilihan Cari Harga ');

writeln('==============================');

writeln('1. Cari Harga Lebih besar dari');

writeln('2. Cari harga lebih kecil dari');

writeln('0. Keluar');

write('Pilihan Anda [0-2]: ');readln(menu);

{validasi menu pilihan}

if (menu < 0) or (menu > 2) then

begin

gotoxy(1, 7); textcolor(yellow);

write('Salah Memilih Menu, Ulangi! (Tekan Enter)'); readln;

menuCariHarga(menu);

end;

// writeln;

// writeln('terima kasih telah menggunakan aplikasi kami..');

// readln;

end;

procedure menuTampil(var menu: integer);

{I.S.: menu belum terdefinisi}

{F.S.: menghasilkan nilai menu dan menampilkan data yang terurut}

begin

clrscr;

textcolor(15);

writeln('Lihat Data Pemesanan Reservasi yang sudah terurut (Asc)');

writeln('=======================================================');

writeln('1. Urut Berdasarkan Kode Pemesanan');

writeln('2. Urut Berdasarkan KTP Pelanggan');

writeln('3. Urut Berdasarkan Nama Pelanggan');

writeln('4. Urut Berdasarkan Harga');

writeln('0. Keluar');

write('Pilihan Anda [0-4]: '); readln(menu);

{validasi menu pilihan}

if (menu < 0) or (menu > 4) then

begin

gotoxy(1, 9);textcolor(yellow);

write('Salah Memilih Menu, Ulangi! (Tekan Enter)'); readln;

menuTampil(menu);

end;

// writeln;

// writeln('terima kasih telah menggunakan aplikasi kami..');

// readln;

end;

procedure pilihan(var menu: integer);

{I.S.: menu belum terdefinisi}

{F.S.: menghasilkan keseluruhan menu utama dan sub-menu}

var

m: integer;

begin

repeat

clrscr;

loadData(database, n, reservasi);

menuPilihan(menu);

// menu := 1;

case (menu) of

1: begin {Input Data Pemesanan reservasi}

clrscr;

isiData(reservasi, n);

simpanData(database, n, reservasi);

readln;

end; // end 1

2: begin {Cari Data Pemesanan Reservasi}

repeat

clrscr;

menuCari(m);

case (m) of

1: begin {Cari Data Berdasarkan Kode Pemesanan}

clrscr;

bubbleSort(reservasi, n, 1);

cariKode(reservasi, n);

end; // end 2 kode

2: begin {Cari Data Berdasarkan No ktp}

clrscr;

bubbleSort(reservasi, n, 2);

cariNoKtp(reservasi, n);

end; // end 2 no ktp

3: begin {Cari Data Berdasarkan Nama}

clrscr;

cariNama(reservasi, n);

end; // end 3 nama

4: begin {Cari ascending Harga}

repeat

clrscr;

menuCariHarga(m);

case (m) of

1: begin {Cari Harga Lebih besar dari}

clrscr;

cariHarga(reservasi, n, 1);

end;

2: begin {Cari harga lebih kecil dari}

clrscr;

cariHarga(reservasi, n, 2);

end;

end; //endcase

until (m = 0);

end; // end 4 harga

end; //endcase

until (m = 0);

end; // end 2

3: begin {Lihat Data Pemesanan reservasi yang sudah terurut}

repeat

clrscr;

menuTampil(m);

case (m) of

1: begin {Urut ascending Kode Pemesanan}

clrscr;

bubbleSort(reservasi, n, 1);

tampilData(reservasi, n);

end; // EndCase 1

2: begin {Urut ascending KTP}

clrscr;

bubbleSort(reservasi, n, 2);

tampilData(reservasi, n);

end; // EndCase 2

3: begin {Urut ascending Nama}

clrscr;

selectionSort(reservasi, n, 1);

tampilData(reservasi, n);

end; // EndCase 3

4: begin {Urut ascending Harga}

clrscr;

selectionSort(reservasi, n, 2);

tampilData(reservasi, n);

end; // EndCase 4

end; //endcase

until (m = 0);

end; // end 3

4: begin {Simpan Data}

clrscr;

simpanData(database, n, reservasi);

write('Data telah berhasil disimpan!'); readkey;

end; // end 4 simpan data

5: begin {Reset Data}

clrscr;

resetData(database, reservasi);

write('Data telah berhasil dihapus!'); readkey;

end; // end 5 reset data

end; //endcase utama

until (menu = 0);

end; // endprocedure

procedure login(user, pass: string);

{I.S.: pass dan user belum terdefinisi}

{F.S.: memvalidasi user dan pass, ketika berhasil masuk ke menuPilihan}

var

i, x: integer;

u, p: string;

begin

clrscr;

x := 3;

repeat

writeln('==================================');

writeln('Masukan Username dan Password Anda');

writeln('----------------------------------');

write('Username: ');readln(u);

write('Password: ');readln(p);

if (user = u) and (pass = p) then

begin

pilihan(menu);

break;

end else

begin

clrscr;

writeln('Username dan Password Salah, Silahkan Ulangi Lagi!');

readln;

end;

clrscr;

dec(x);

if (x = 0) then

begin

writeln('Maaf Anda Tidak Bisa Login, Karna Lebih dari 3 Kali Salah');

end;

until (x = 0);

end;//endprocedure

{program utama}

begin

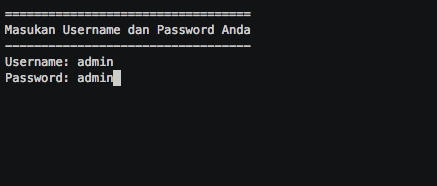
login(USER, PASS);

readln;

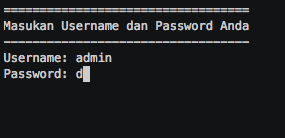
end.

1. Tampilan Layar

### User Login (admin:admin)



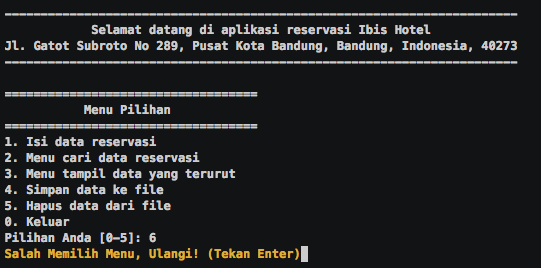
### Percobaan salah (hanya 3x)

screenshot/01.%20isi_data/login/02a.%20salah.pngscreenshot/01.%20isi_data/login/03.%20kesempatan_habis.png

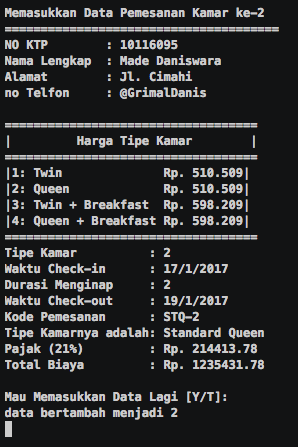
### Menu Utama



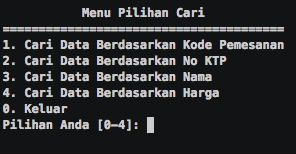
### Validasi Menu Utama



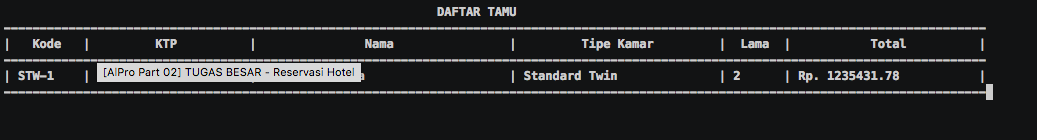
### Menu ke-1 (isi data tamu)



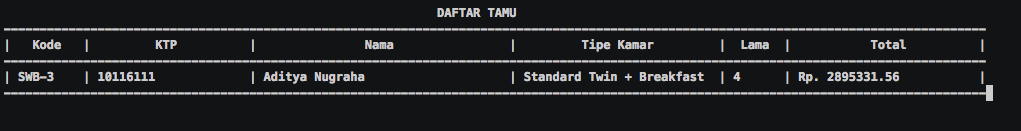
### Menu ke-2 (Cari data)



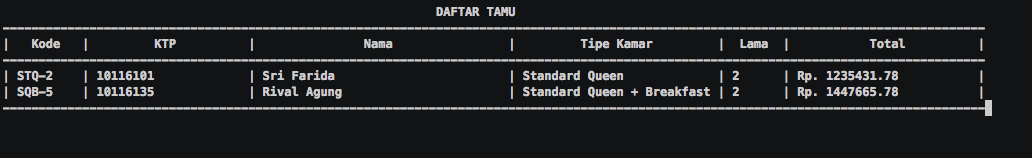
### Menu ke-2 (cari kode)

screenshot/02.%20menu_cari/02.%20kode_pesan.png

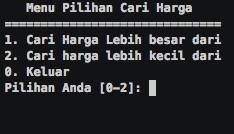
### Menu ke-2 (cari KTP)

screenshot/02.%20menu_cari/03.%20ktp.png

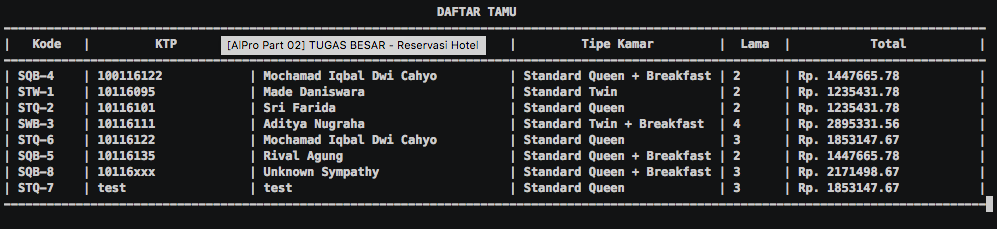
### Menu ke-2 (cari Nama)

screenshot/02.%20menu_cari/04.%20nama.png

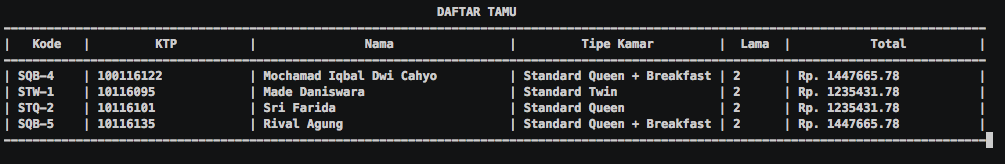
### Menu ke-2 (cari harga)



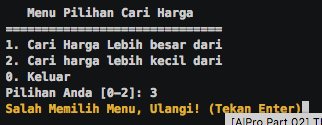
### Menu cari harga (lebih besar dari)

screenshot/02.%20menu_cari/harga/02.%20greater%20than.png

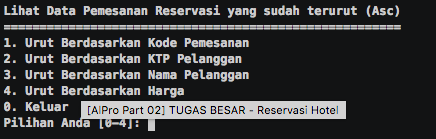
### Menu cari harga (kurang dari)

screenshot/02.%20menu_cari/harga/03.%20less%20than.png

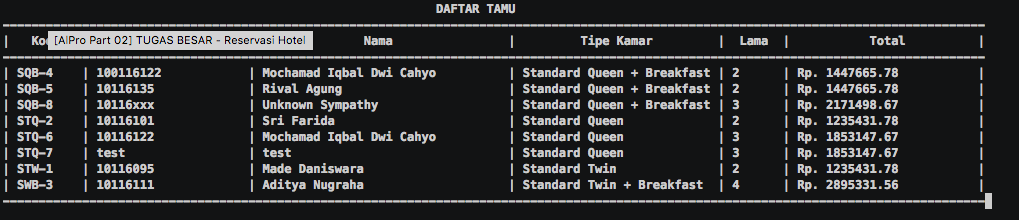
### Validasi Menu Harga



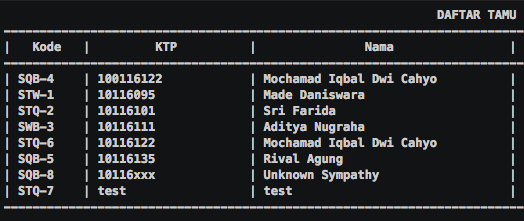
### Menu Tampil (telah terurut secara ascending)



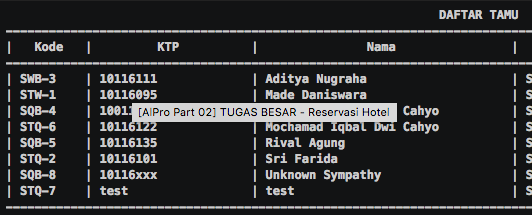
### Terurut (Kode)



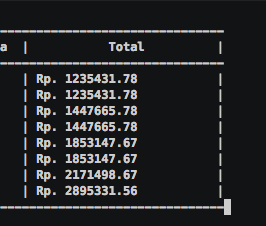
### Terurut (KTP)



### Terurut (Nama case-sensitive, kapital didahulukan)



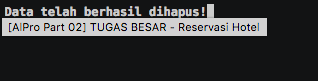
### Terurut (Harga)



### Simpan ke file (Menu utama ke-4)

screenshot/04.%20simpan/01.%20.png

### Hapus file (Menu utama ke-5)



**DAFTAR PUSTAKA**

|  |  |
| --- | --- |
| [1] | F. D. Documentation. [Online]. Available: http://www.freepascal.org/docs-html/rtl/sysutils/datetimeroutines.html. |
| [2] | F. P. &. F. O. Documentation. [Online]. Available: http://www.math.uni-leipzig.de/pool/tuts/FreePascal/ref/node11.html. |
| [3] | F. F. Documentation. [Online]. Available: http://www.freepascal.org/docs-html/rtl/system/filefunctions.html. |
| [4] | F. T.-c. e. code. [Online]. Available: http://www.freepascal.org/docs-html/3.0.0/user/userse63.html. |
| [5] | F. (. o. b. Untyped files. [Online]. Available: http://computer-programming-forum.com/29-pascal/e1d19cb8b13d0e8c.htm. |
| [6] | F. u. f. h. Blockread & Blockwrite. [Online]. Available: http://www.math.uni-leipzig.de/pool/tuts/FreePascal/ref/node17.html. |

# Kontribusi

1. Kontribusi

**- 10116095 - Made Daniswara**: pembuatan operasi check-in dan check-out, fungsi jenisTipe, kodeTipe, hargaTipe (procedure concat dan str, untuk penyatuan nomor ke string)

**- 10116101 – Sri Farida**: prosedur isiData tamu, menuCari (Kode, KTP, Nama & Harga)

**- 10116111 – Aditya Nugraha**: prosedur menuPilihan, pilihan, login beserta validasinya

**- 10116122 – Mochamad Iqbal Dwi Cahyo**: prosedur pemrosesan File (bukaData, loadData, simpanData, resetData). Serta pencarian Binary dan Selection (memunculkan data yang sama)

**- 10116135 – Rival Agung**: prosedur tampilData, pengurutan Bubble dan Minimum Selection