Algoritma Penjualan\_Tiket

{I.S : User menginputkan pilihan nomor menu}

{F.S: Menampilkan program pilihan nomor menu}

Kamus:

Type

Dtiket = ↑Rtiket

Rtiket = Record

< nokursi : integer

nama : string

alamat : string

nohape : string

kodetujuan : string

tujuan : string

kodekelas : integer

kelas : string

harga : integer

next : DTiket >

EndRecord

{*Kamus Global*}

data, awal, akhir : Dtiket

pilihan : integer

{*Prototype*}

Procedure InputData(I/O isi : Dtiket)

{I.S : user memasukkan data yang akan diinputkan}

{F.S: menghasilkan data yang telah diinputkan}

Kamus :

Algoritma :

Output(‘No. Kursi : ‘) Input(isi↑.nokursi)

Output(‘Nama : ‘) Input(isi↑.nama)

Output(‘Alamat : ‘) Input(isi↑.alamat)

Output(‘No. HP) : ‘) Input(isi↑.nohape)

Repeat

Output (‘Kode Tujuan : ‘) Input(isi↑.kodetujuan)

isi↑.kodetujuan ← upcase(isi↑.kodetujuan)

Until((isi↑.kodetujuan=‘CRB’) or (isi↑.kodetujuan=‘BDG’) or (isi↑.kodetujuan=‘JKT’))

If(isi↑.kodetujuan=‘CRB’) then isi↑.tujuan ← ‘Cirebon’

else

If(isi↑.kodetujuan=‘BDG’) then isi↑.tujuan ← ‘Bandung’

else

If(isi↑.kodetujuan=‘JKT’) then isi↑.tujuan ← ‘Jakarta’

EndIf

Endif

Endif

Repeat

Output(‘Kode Kelas : ‘) Input(isi↑.kodekelas)

Until((isi↑.kodekelas>0) and (isi↑.kodekelas<4))

Case (isi↑.kodekelas) of

1 : isi↑.kelas←’Eksekutif’

2 : isi↑.kelas←’Bisnis’

3 : isi↑.kelas←’Ekonomi’

Endcase

If(isi↑.kodetujuan=‘CRB’) then

If(isi↑.kodekelas=1) then isi↑.harga←100000

else

If (isi↑.kodekelas=2) then isi↑.harga←80000

else

If (isi↑.kodekelas=3) then isi↑.harga←45000

EndIf

EndIf

EndIf

else

If(isi↑.kodetujuan=‘BDG’) then

If(isi↑.kodekelas=1) then isi↑.harga←150000

else

If (isi↑.kodekelas=2) then isi↑.harga←100000

else

If (isi↑.kodekelas=3) then isi↑.harga←80000

EndIf

EndIf

EndIf

else

If(isi↑.kodetujuan=‘JKT’) then

If(isi↑.kodekelas=1) then isi↑.harga←200000

else

If (isi↑.kodekelas=2) then isi↑.harga←150000

else

If (isi↑.kodekelas=3) then isi↑.harga←100000

EndIf

EndIf

EndIf

EndIf

EndIf

EndIf

EndProcedure

Procedure LihatData(Output isi:Dtiket)

{I.S : Data yang akan ditampilkan sudah terdefinisi}

{F.S: Menampilkan data}

Kamus :

Algoritma:

isi←awal

While(isi ≠ nil) do

Output(‘No: ‘,isi↑.nokursi)

Output(‘Nama : ‘, isi↑.nama)

Output(‘Alamat : ‘,isi↑.alamat)

Output(‘No. HP : ‘,isi↑.nohape)

Output(‘Tujuan : ‘,isi↑.tujuan)

Output(‘Kelas : ‘,isi↑.kelas)

isi←isi↑.next

EndWhile

EndProcedure

{*Penyisipan*}

Procedure SisipDepan(I/O awal, akhir : Dtiket)

{I.S : Data yag disisipkan (elemen), pointer awal dan pointer penunjuka akhir sudah terdefinisi}

{F.S: Menghasilkan satu simpul yang disisipkan di depan pada single linked list}

Kamus :

baru : Dtiket

Algoritma :

alloc(baru)

InputData(baru)

If (awal=nil) then

baru↑.next←nil

akhir←baru

else

baru↑.next←awal

awal←baru

EndIf

EndProcedure

Procedure SisipBelakang(I/O awal, akhir:Dtiket)

{I.S : Data yag disisipkan (elemen), pointer awal dan pointer penunjuka akhir sudah terdefinisi}

{F.S: Menghasilkan satu simpul yang disisipkan di belakang pada single linked list}

Kamus :

Baru : Dtiket

Algoritma :

alloc(baru)

InputData(baru)

baru↑.next←nil

If(awal=nil) then

awal←baru

else

akhir↑.next←baru

akhir←baru

EndIf

EndProcedure

Procedure SisipTengah(I/O awal, akhir : Dtiket)

{I.S : Data yag disisipkan (elemen), pointer awal dan pointer penunjuka akhir sudah terdefinisi}

{F.S: Menghasilkan satu simpul yang disisipkan di tengah pada single linked list}

Kamus :

Baru, bantu : Dtiket

ketemu : boolean

sisip : integer

Algoritma :

alloc(baru)

InputData(baru)

If(awal=nil) then

baru↑.next←nil

awal←baru

akhir←baru

else

Output(‘Data Akan Dimasukan Setelah Nomor : ‘) Input(sisip)

bantu←awal

ketemu←false

While ((not ketemu) and (bantu ≠ nil)) do

If(sisip=bantu↑.nokursi) then

ketemu←true

else

bantu←bantu↑.next

EndIf

EndWhile

EndIf

If ketemu then

If(bantu=akhir) then

SisipBelakang(awal,akhir)

else

alloc(baru)

InputData(baru)

baru↑.next←bantu↑.next

bantu↑.next←baru

EndIf

else

Output(‘Nomor yang dimasukkan tidak ada’)

EndIf

EndProcedure

{*Penghapusan*}

Procedure HapusDepan(I/O awal, akhir : DTiket) //Hapus Depan

{I.S : data yang akan dihapus, pointer penunjuk awal dan pointer penunjuk akhir sudah terdefinisi}

{F.S: menghapus satu simpul yang berada didepan pada single link list}

Kamus:

phapus : DTiket

Algoritma:

phapus 🡨 awal

If (awal = akhir) then

awal 🡨 nil

akhir 🡨 nil

else

awal 🡨 awal↑.next

EndIf

Dealloc(phapus)

Output(‘Data Telah Dihapus’)

EndProcedure

Procedure HapusBelakang(I/O awal, akhir : DTiket) //Hapus Belakang

{I.S : data yang akan dihapus, pointer penunjuk awal dan pointer penunjuk akhir sudah terdefinisi}

{F.S: menghapus satu simpul yang berada dibelakang pada single link list}

Kamus:

phapus : DTiket

Algoritma:

phapus 🡨 awal

If (awal = akhir) then

awal 🡨 nil

akhir 🡨 nil

else

While (phapus↑.next ≠ akhir) do

phapus 🡨 phapus↑.next

EndWhile

akhir 🡨 phapus

phapus 🡨 phapus↑.next

akhir↑.next 🡨 nil

EndIf

Dealloc(phapus)

Output(‘Data Telah Dihapus’)

EndProcedure

Procedure HapusTengah(I/O awal, akhir : DTiket) //Hapus Tengah

{I.S : data yang akan dihapus, pointer penunjuk awal dan pointer penunjuk akhir sudah terdefinisi}

{F.S: menghapus satu simpul yang berada ditengah pada single link list}

Kamus:

phapus, bantu : DTiket

ketemu : boolean

hapus : integer

Algoritma:

phapus 🡨 awal

If(awal = akhir) then

awal 🡨 nil

akhir 🡨 nil

Dealloc(phapus)

Output(‘Data Telah Dihapus’)

else

LihatData(data)

Output(‘Nomor Yang Akan Dihapus : ‘) Input(hapus)

ketemu 🡨 false

While ((not ketemu) and (phapus ≠ nil)) do

If (hapus = phapus↑.nama) then

ketemu 🡨 true

else

phapus 🡨 phapus↑.next

EndIf

EndWhile

If (ketemu) then

If (phapus = awal) then

HapusDepan(awal, akhir)

else

If (phapus = akhir) then

HapusBelakang(awal, akhir)

else

bantu 🡨 awal

While (bantu↑.next ≠ phapus) do

bantu 🡨 bantu↑.next

EndWhile

bantu↑.next 🡨 phapus↑.next

Dealloc(phapus)

Output(‘Data Telah Dihapus’)

EndIf

EndIf

else

Output(‘Keyword Tidak Sesuai’)

EndIf

EndIf

EndProcedure

{*Pencarian*}

Procedure CariKursi(Input awal:DTiket)

{I.S : Penunjuk pointer awal sudah terdefinisi}

{F.S: Menampilkan nama yang dicari atau tidak}

Kamus:

bantu : DTiket

ketemu : boolean

cari : integer

Algoritma:

Output(‘Nomor Kursi Yang di Cari : ‘) Input(cari)

bantu 🡨 awal

ketemu 🡨 false

While((not ketemu) and (bantu≠nil)) do

If (cari=bantu↑.nokursi) then

ketemu 🡨 true

else

bantu 🡨 bantu↑.next

EndIf

EndWhile

If (ketemu) then

Output(‘bantu↑.nokursi’)

Output(‘bantu↑.nama’)

Output(‘bantu↑.alamat’)

Output(‘bantu↑.nohape’)

Output(‘bantu↑.tujuan’)

Output(‘bantu↑.kelas’)

Output(‘bantu↑.harga’)

else

Output(‘Data Tidak Ditemukan’)

EndIf

EndProcedure

Procedure CariNama(Input awal:DTiket)

{I.S : Penunjuk pointer awal sudah terdefinisi}

{F.S: Menampilkan data yang dicari atau tidak}

Kamus:

bantu : DTiket

cari : string

Algoritma:

Output(‘NamaYang di Cari : ‘) Input(cari)

bantu 🡨 awal

While(bantu≠nil) do

If (cari=bantu↑.nama) then

Output(‘bantu↑.nokursi’)

Output(‘bantu↑.nama’)

Output(‘bantu↑.alamat’)

Output(‘bantu↑.nohape’)

Output(‘bantu↑.tujuan’)

Output(‘bantu↑.kelas’)

Output(‘bantu↑.harga’)

bantu 🡨 bantu↑.next

else

bantu 🡨 bantu↑.next

EndIf

EndWhile

EndProcedure

{*Pengurutan*}

Procedure SortingKursi(input awal,akhir:DTiket)

{I.S : Pointer penunjuk awal dan pointer akhir sudah terdefinisi}

{F.S: Menghasilkan harga yang sudah tersusun secara ascending}

Kamus:

min,i,j : DTiket

temo : string

temp : integer

Algoritma:

LihatData(data)

i 🡨 awal

While(i≠akhir) do

min 🡨 i

j 🡨 i↑.next

While(j≠nil) do

If(j↑.nokursi < min↑.nokursi) then

min 🡨 j

j 🡨 j↑.next

EndIf

EndWhile

temp 🡨 i↑.nokursi

i↑.nokursi 🡨 min↑.nokursi

min↑.nokursi 🡨 temp

temo 🡨 i↑.nama

i↑.nama 🡨 min↑.nama

min↑.nama 🡨 temo

temo 🡨 i↑.alamat

i↑.alamat 🡨 min↑.alamat

min↑.alamat 🡨 temo

temo 🡨 i↑.nohape

i↑.nohape 🡨 min↑.nohape

min↑.nohape 🡨 temo

temo 🡨 i↑.tujuan

i↑.tujuan 🡨 min↑.tujuan

min↑.tujuan 🡨 temo

temo 🡨 i↑.kelas

i↑.kelas 🡨 min↑.kelas

min↑.kelas 🡨 temo

temp 🡨 i↑.harga

i↑.harga 🡨 min↑.harga

min↑.harga 🡨 temp

i 🡨 i↑.next

EndWhile

LihatData(data)

EndProcedure

Procedure SortingHarga(input awal,akhir:DTiket)

{I.S : Pointer penunjuk awal dan pointer akhir sudah terdefinisi}

{F.S: Menghasilkan harga yang sudah tersusun secara ascending}

Kamus:

min,i,j : DTiket

temo : string

temp : integer

Algoritma:

LihatData(data)

i 🡨 awal

While(i≠akhir) do

min 🡨 i

j 🡨 i↑.next

While(j≠nil) do

If(j↑.harga < min↑.harga) then

min 🡨 j

j 🡨 j↑.next

EndIf

EndWhile

temp 🡨 i↑.nokursi

i↑.nokursi 🡨 min↑.nokursi

min↑.nokursi 🡨 temp

temo 🡨 i↑.nama

i↑.nama 🡨 min↑.nama

min↑.nama 🡨 temo

temo 🡨 i↑.alamat

i↑.alamat 🡨 min↑.alamat

min↑.alamat 🡨 temo

temo 🡨 i↑.nohape

i↑.nohape 🡨 min↑.nohape

min↑.nohape 🡨 temo

temo 🡨 i↑.tujuan

i↑.tujuan 🡨 min↑.tujuan

min↑.tujuan 🡨 temo

temo 🡨 i↑.kelas

i↑.kelas 🡨 min↑.kelas

min↑.kelas 🡨 temo

temp 🡨 i↑.harga

i↑.harga 🡨 min↑.harga

min↑.harga 🡨 temp

i 🡨 i↑.next

EndWhile

LihatData(data)

EndProcedure

{*Penghancuran*}

Procedure Penghancuran(Output awal:DTiket)

{I.S : Pointer penunjuk akhir sudah terdefinisi}

{F.S: Menghapus seluruh data yang ada di list}

Kamus:

Phapus : DTiket

Algoritma:

phapus 🡨 awal

While(awal≠nil) do

phapus 🡨 awal

dealloc(phapus)

awal 🡨 awal↑.next

EndWhile

awal 🡨 nil

akhir 🡨 nil

LihatData(data)

Output(‘Semua Data Telah Dihapus’)

EndProcedure

{*Menu*}

Procedure Menu(Output pil: integer)

{I.S : user menginputkan nomor menu}

{F.S: menampilkan menu yang akan dipilih user}

Kamus :

Algoritma :

Output(‘ MENU PENJUALAN TIKET’)

Output(‘ PT KERETA API TORABIKA’);

Output(‘==============================‘)

Output(‘1. Sisip Data Pembeli’)

Output(‘2. Hapus Data Pembeli’)

Output(‘3. Cari Data Pembeli’)

Output(‘4. Urut Data Pembeli’)

Output(‘5. Penghancuran Data Pembeli’)

Output(‘6. Lihat Data Pembeli’)

Output(‘7. Keluar’)

Output(‘==============================‘)

Output(‘Pilihan :’) Input(pil)

EndProcedure

Procedure MenuSisip

{I.S : user menginputkan nomor menu sisip}

{F.S : menampilkan menu yang dipilih user}

Kamus :

Pil : integer

Algoritma :

Repeat

Output(‘ MENU PENYISIPAN DATA’)

Output(‘=========================‘)

Output(‘1. Sisip di Depan’)

Output(‘2. Sisip di Tengah’)

Output(‘3. Sisip di Belakang’)

Output(‘4. Kembali ke Menu Utama’)

Output(‘=========================‘)

Output(‘Pilihan :’) Input(pil)

Case pil of

1: SisipDepan(awal,akhir)

2: SisipTengah(awal,akhir)

3: SisipBelakang(awal,akhir)

Endcase

Until(pil=4)

EndProcedure

Procedure MenuHapus

{I.S : User menginputkan nomor menu hapus}

{F.S: Menampilkan menu yang dipilih user}

Kamus :

pil : integer

Algoritma :

Repeat

Output(‘ MENU PENGHAPUSAN DATA’)

Output(‘=========================‘)

Output(‘1. Hapus di Depan’)

Output(‘2. Hapus di Tengah’)

Output(‘3. Hapus di Belakang’)

Output(‘4. Kembali ke Menu Utama’)

Output(‘=========================‘)

Output(‘Pilihan :’) Input(pil)

Case pil of

1: HapusDepan(awal,akhir)

2: HapusTengah(awal,akhir)

3: HapusBelakang(awal,akhir)

Endcase

Until(pil=4)

Endprocedure

Procedure MenuCari

{I.S : User menginputkan nomor pilihan menu}

{F.S: Menampilkan program pilihan user}

Kamus:

pil : integer

Algoritma:

Repeat

Output (‘ MENU PENCARIAN DATA’)

Output(‘================================‘)

Output(‘1. Cari Berdasarkan Nomor Kursi’)

Output(‘2. Cari Berdasarkan Nama’)

Output(‘3. Kembali ke menu Utama’)

Output(‘================================‘)

Output(‘Pilihan :’) Input(pil)

Case pil of

1: CariKursi(awal)

2: CariNama(awal)

Endcase

Until(pil=3)

EndProcedure

Procedure MenuUrut

{I.S : User menginputkan nomor pilihan menu}

{F.S: Menampilkan program pilihan user}

Kamus:

pil : integer

Algoritma:

Repeat

Output (‘ MENU PENGURUTAN DATA’)

Output(‘================================‘)

Output(‘1. Urut Berdasarkan Nomor Kursi’)

Output(‘2. Urut Berdasarkan Harga’)

Output(‘3. Kembali ke menu Utama’)

Output(‘================================‘)

Output(‘Pilihan :’) Input(pil)

Case pil of

1: SortingKursi(awal,akhir)

2: SortingHarga(awal,akhir)

Endcase

Until(pil=3)

EndProcedure

{Algoritma Utama}

Algoritma :

awal←nil

akhir←nil

Repeat

Menu(pilihan)

Case pilihan of

1 : MenuSisip

2 : MenuHapus

3 : MenuCari

4 : SortingHarga(awal,akhir)

5 : Penghancuran(awal)

6 : LihatData(data)

Endcase

Until(pilihan=7)