Algoritma Penjualan\_Tiket\_KA

{I.S : user menginputkan data}

{F.S: menghasilkan data yang diinputkan}

Kamus :

Type

Tiket = Record

< nokursi : integer,

nama : string,

alamat : string,

nohape : string,

kodetujuan : string,

tujuan : string,

kodekelas : integer,

kelas : string,

harga : integer >

EndRecord

RTiket = Tiket

Pointer = ↑Simpul

Simpul = Record

< prev : Pointer,

info : RTiket,

next : Pointer >

EndRecord

data,awal,akhir : Pointer

elemen : RTiket

pilihan : integer

total : integer

Procedure Isi (Output elemen : RTiket)

{I.S : user menginputkan data}

{F.S: menghasilkan data yang telah diinputkan}

Kamus :

Algoritma :

Output ('Nomor Kursi : ') Input (elemen.nokursi)

Output ('Nama : ') Input (elemen.nama)

Output ('Alamat : ') Input (elemen.alamat)

Output ('No. HP : ') Input (elemen.nohape)

Repeat

Output ('Kode Tujuan : ') Input (elemen.kodetujuan)

elemen.kodetujuan 🡨 upcase(elemen.kodetujuan)

Until (elemen.kodetujuan = 'BDG') or (elemen.kodetujuan = 'CRB') or (elemen.kodetujuan = 'JKT')

If (elemen.kodetujuan = 'CRB')

Then

elemen.tujuan 🡨 'Cirebon'

Else

If (elemen.kodetujuan = 'BDG')

Then

elemen.tujuan 🡨 'Bandung'

Else

If (elemen.kodetujuan = 'JKT')

Then

elemen.tujuan 🡨 'Jakarta'

EndIf

EndIf

EndIf

Repeat

Output ('Kode Kelas : ') Input (elemen.kodekelas)

Until (elemen.kodekelas > 0) and (elemen.kodekelas < 4)

Case (elemen.kodekelas) of

1: elemen.kelas 🡨 'Eksekutif'

2: elemen.kelas 🡨 'Bisnis'

3: elemen.kelas 🡨 'Ekonomi'

EndCase

If (elemen.kodetujuan = 'CRB')

Then

If (elemen.kodekelas = 1)

Then

elemen.harga 🡨 100000

Else

If (elemen.kodekelas = 2)

Then

elemen.harga 🡨 80000

Else

If (elemen.kodekelas = 3)

Then

elemen.harga 🡨 45000

EndIf

EndIf

EndIf

Else

If (elemen.kodetujuan = 'BDG')

Then

If (elemen.kodekelas = 1)

Then

elemen.harga 🡨 150000

Else

If (elemen.kodekelas = 2)

Then

elemen.harga 🡨 100000

Else

If (elemen.kodekelas = 3)

Then

elemen.harga 🡨 80000

EndIf

EndIf

EndIf

Else

If (elemen.kodetujuan = 'JKT')

Then

If (elemen.kodekelas = 1)

Then

elemen.harga 🡨 200000

Else

If (elemen.kodekelas = 2)

Then

elemen.harga 🡨 150000

Else

If (elemen.kodekelas = 3)

Then

elemen.harga 🡨 100000

EndIf

EndIf

EndIf

EndIf

EndIf

EndIf

EndProcedure

Procedure Tampil (Input data : Pointer)

{I.S : list sudah terdefinisi}

{F.S: menampilkan list}

Kamus :

Algoritma :

data 🡨 awal

While (data ≠ nil) Do

Output ('Nomor Kursi : ',data↑.info.nokursi)

Output ('Nama : ',data↑.info.nama)

Output ('Alamat : ',data↑.info.alamat)

Output ('No. HP : ',data↑.info.nohape)

Output ('Tujuan : ',data↑.info.tujuan)

Output ('Kelas : ',data↑.info.kelas)

Output ('Harga : ',data↑.info.harga)

data 🡨 data↑.next

EndWhile

EndProcedure

*{Penyisipan}*

Procedure SisipDepan (Input elemen : RTiket, I/O awal,akhir : Pointer)

{I.S : data yang akan disisipkan (elemen), pointer penunjuk awal dan pointer penunjuk akhir sudah terdefinisi}

{F.S: menghasilkan satu simpul yang disisipkan didepan pada double linked list}

Kamus :

baru : Pointer

Algoritma :

Alloc (baru)

baru↑.info 🡨 elemen

baru↑.prev 🡨 nil

If (awal = nil)

Then

baru↑.next 🡨 nil

akhir 🡨 baru

Else

baru↑.next 🡨 awal

awal↑.prev 🡨 baru

EndIf

awal 🡨 baru

Tampil (data)

EndProcedure

Procedure SisipTengah (Input elemen : RTiket, I/O awal,akhir : Pointer)

{I.S : data yang akan disisipkan (elemen), pointer penunjuk awal dan pointer penunjuk akhir sudah terdefinisi }

{F.S: menghasilkan satu simpul yang disisipkan ditengah pada double linked list}

Kamus :

baru,bantu : Pointer

ketemu : boolean

datasisip : integer

Algoritma :

Alloc (baru)

baru↑.info 🡨 elemen

If (awal = nil)

Then

baru↑.prev 🡨 nil

baru↑.next 🡨 nil

awal 🡨 baru

akhir 🡨 baru

Else

Tampil (data)

Output ('Data akan disisipkan sebelum Nomor : ') Input (datasisip)

bantu 🡨 awal

ketemu 🡨 false

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

If (datasisip = bantu↑.info.nokursi)

Then

ketemu 🡨 true

Else

bantu 🡨 bantu↑.next

EndIf

EndWhile

EndIf

If (ketemu)

Then

If (bantu = awal)

Then

SisipDepan (elemen,awal,akhir)

Else

baru↑.next 🡨 bantu

baru↑.prev 🡨 bantu↑.prev

bantu↑.prev↑.next 🡨 baru

bantu↑.prev 🡨 baru

Tampil (data)

EndIf

Else

Output ('Data dengan Nomor ',datasisip,' tidak ditemukan')

EndIf

EndProcedure

Procedure SisipBelakang (Input elemen : RTiket, I/O awal,akhir : Pointer)

{I.S : data yang akan disisipkan (elemen), pointer penunjuk awal dan pointer penunjuk akhir sudah terdefinisi }

{F.S: menghasilkan satu simpul yan disisipkan ditengah pada double linked list}

Kamus :

baru : Pointer

Algoritma :

Alloc (baru)

baru↑.info 🡨 elemen

baru↑.next 🡨 nil

If (akhir = nil)

Then

baru↑.prev 🡨 nil

awal 🡨 baru

Else

baru↑.prev 🡨 akhir

akhir↑.next 🡨 baru

EndIf

akhir 🡨 baru

Tampil (data)

EndProcedure

*{Penghapusan}*

Procedure HapusDepan (Output elemen : RTiket, I/O awal,akhir : Pointer)

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

{F.S: menghapus satu simpul yang berada didepan pada double linked list}

Kamus :

phapus : Pointer

Algoritma :

phapus 🡨 awal

elemen 🡨 phapus↑.info

If (awal = akhir)

Then

awal 🡨 nil

akhir 🡨 nil

Else

awal 🡨 awal↑.next

awal↑.prev 🡨 nil

EndIf

Dealloc (phapus)

Output ('Data telah dihapus')

EndProcedure

Procedure HapusBelakang (Output elemen : RTiket, I/O awal,akhir : Pointer)

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

{F.S: menghapus satu simpul yang berada dibelakang pada double linked list}

Kamus :

phapus : Pointer

Algoritma :

phapus 🡨 akhir

elemen 🡨 phapus↑.info

If (awal = akhir)

Then

awal 🡨 nil

akhir 🡨 nil

Else

akhir 🡨 akhir↑.prev

akhir↑.next 🡨 nil

EndIf

Dealloc (phapus)

Output ('Data Telah Dihapus')

EndProcedure

Procedure HapusTengah (Output elemen : RTiket, I/O awal,akhir : Pointer)

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

{F.S: menghapus satu simpul yang berada ditengah pada double linked list}

Kamus :

phapus : Pointer

ketemu : boolean

datahapus : integer

Algoritma :

If (awal = akhir)

Then

phapus 🡨 awal

elemen 🡨 phapus↑.info

awal 🡨 nil

akhir 🡨 nil

Dealloc (phapus)

Output ('Data Telah Dihapus')

Else

Tampil (data)

Output ('Masukkan Nomor yang akan dihapus : ') Input (datahapus)

phapus 🡨 awal

ketemu 🡨 false

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

If (datahapus = phapus↑.info.nokursi)

Then

ketemu 🡨 true

Else

phapus 🡨 phapus↑.next

EndIf

EndWhile

If (ketemu)

Then

elemen 🡨 phapus↑.info

If (phapus = awal)

Then

HapusDepan (elemen,awal,akhir)

Else

If (phapus = akhir)

Then

HapusBelakang (elemen,awal,akhir)

Else

phapus↑.prev↑.next 🡨 phapus↑.next

phapus↑.next↑.prev 🡨 phapus↑.prev

Dealloc (phapus)

Output ('Data Telah Dihapus')

EndIf

EndIf

Else

Output('Data dengan Nomor ',datahapus,' tidak ditemukan')

EndIf

EndIf

EndProcedure

*{Pencarian}*

Procedure CariKursi (Input awal : Pointer)

{I.S : penunjuk pointer awal sudah terdefinisi}

{F.S: menampilkan nomor kursi yang dicari atau tidak}

Kamus :

bantu : Pointer

ketemu : boolean

datacari : integer

Algoritma :

Output ('Nomor Kursi yang Dicari : ') Input (datacari)

bantu 🡨 awal

ketemu 🡨 false

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

If (datacari = bantu↑.info.nokursi)

Then

ketemu 🡨 true

Else

bantu 🡨 bantu↑.next

EndIf

EndWhile

If (ketemu)

Then

Output ('Nomor Kursi : ',bantu↑.info.nokursi)

Output ('Nama : ',bantu↑.info.nama)

Output ('Alamat : ',bantu↑.info.alamat)

Output ('No. HP : ',bantu↑.info.nohape)

Output ('Kode Tujuan : ',bantu↑.info.tujuan)

Output ('Kode Kelas : ',bantu↑.info.kelas)

Output ('Harga : ',bantu↑.info.harga)

Else

Output ('Data dengan Nomor ',datacari, ' Tidak Ditemukan')

EndIf

EndProcedure

Procedure CariTujuan (Input awal : Pointer)

{I.S : penunjuk pointer awal sudah terdefinisi}

{F.S: menampilkan rute tujuan yang dicari atau tidak}

Kamus :

bantu : Pointer

datacari : string

Algoritma :

Output ('Tujuan yang Dicari : ') Input(datacari)

bantu 🡨 awal

Repeat

If (datacari = bantu↑.info.tujuan)

Then

Output ('Nomor Kursi : ',bantu↑.info.nokursi)

Output ('Nama : ',bantu↑.info.nama)

Output ('Alamat : ',bantu↑.info.alamat)

Output ('No. HP : ',bantu↑.info.nohape)

Output ('Kode Tujuan : ',bantu↑.info.tujuan)

Output ('Kode Kelas : ',bantu↑.info.kelas)

Output ('Harga : ',bantu↑.info.harga)

bantu 🡨 bantu↑.next

Else

bantu 🡨 bantu↑.next

EndIf

Until (bantu = nil)

EndProcedure

*{Total Harga}*

Procedure TotalHarga (Output total : integer)

{I.S : harga sudah terdefinisi}

{F.S: menampilkan total harga}

Kamus :

Algoritma :

data 🡨 awal

total 🡨 0

While (data ≠ nil) Do

total 🡨 total+data↑.info.harga

data 🡨 data↑.next

EndWhile

Output ('Total Harga = ',total)

EndProcedure

*{Pengurutan}*

Procedure UrutKursi (Input awal,akhir : Pointer)

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

{F.S: menghasilkan nomor kursi yang sudah terurut secara ascending}

Kamus :

min,i,j : Pointer

temo : string

temp : integer

Algoritma :

i 🡨 awal

While (i ≠ akhir) Do

min 🡨 i

j 🡨 i↑.next

While (j ≠ nil) Do

If (j↑.info.nokursi < min↑.info.nokursi)

Then

min 🡨 j

j 🡨 j↑.next

EndIf

EndWhile

temp 🡨 i↑.info.nokursi

i↑.info.nokursi 🡨 min↑.info.nokursi

min↑.info.nokursi 🡨 temp

temo 🡨 i↑.info.nama

i↑.info.nama 🡨 min↑.info.nama

min↑.info.nama 🡨 temo

temo 🡨 i↑.info.alamat

i↑.info.alamat 🡨 min↑.info.alamat

min↑.info.alamat 🡨 temo

temo 🡨 i↑.info.nohape

i↑.info.nohape 🡨 min↑.info.nohape

min↑.info.nohape 🡨 temo

temo 🡨 i↑.info.tujuan

i↑.info.tujuan 🡨 min↑.info.tujuan

min↑.info.tujuan 🡨 temo

temo 🡨 i↑.info.kelas

i↑.info.kelas 🡨 min↑.info.kelas

min↑.info.kelas 🡨 temo

temp 🡨 i↑.info.harga

i↑.info.harga 🡨 min↑.info.harga

min↑.info.harga 🡨 temp

i 🡨 i↑.next

EndWhile

EndProcedure

*{Tampil Data}*

Procedure TampilData

{I.S : list sudah terdefinisi}

{F.S: menampilkan list}

Kamus :

Algoritma :

If (awal ≠ nil) and (akhir ≠ nil)

Then

UrutKursi (awal,akhir)

Tampil (data)

TotalHarga (total)

Else

Output ('List Kosong')

EndIf

EndProcedure

*{Penghancuran}*

Procedure Penghancuran (Output awal : Pointer)

{I.S : penunjuk pointer awal sudah terdefinisi}

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

Kamus :

phapus : Pointer

Algoritma :

phapus 🡨 awal

While (awal ≠ nil) Do

phapus 🡨 awal

Dealloc (phapus)

awal 🡨 awal↑.next

EndWhile

awal 🡨 nil

akhir 🡨 nil

Output ('Semua Data Telah Dihapus')

Output ('Terima Kasih telah menggunakan program ini')

EndProcedure

Procedure Keluar

{I.S : list sudah terdefinisi}

{F.S: menghapus semua data dalam list}

Kamus :

Algoritma :

Penghancuran (awal)

EndProcedure

*{Menu}*

Procedure Menu (Output pil : integer)

{I.S : pilihan user (pil) sudah terdefinisi}

{F.S: menampilkan menu pilihan}

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. Tampil Data Pembeli')

Output ('5. Keluar')

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

Output (' Pilihan : ') Input (pil)

EndProcedure

Procedure MenuSisip

{I.S : pilihan (pil) user sudah terdefinisi}

{F.S: menampilkan menu pilihan}

Kamus :

pil : integer

Algoritma :

Repeat

Output ('MENU SISIP DATA')

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

Output ('1. Sisip Depan')

Output ('2. Sisip Tengah')

Output ('3. Sisip Belakang')

Output ('4. Kembali ke Menu Utama')

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

Output (' Pilihan : ') Input (pil)

Case pil of

1: Isi (elemen)

SisipDepan (elemen,awal,akhir)

2: Isi (elemen)

SisipTengah (elemen,awal,akhir)

3: Isi (elemen)

SisipBelakang (elemen,awal,akhir)

EndCase

Until (pil = 4)

EndProcedure

Procedure MenuHapus

{I.S : pilihan (pil) sudah terdefinisi}

{F.S: menampilkan menu pilihan}

Kamus :

pil : integer

Algoritma :

Repeat

Output ('MENU HAPUS DATA')

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

Output ('1. Hapus Depan')

Output ('2. Hapus Tengah')

Output ('3. Hapus Belakang')

Output ('4. Kembali ke Menu Utama')

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

Output (' Pilihan : ') Input (pil)

Case pil of

1: HapusDepan (elemen,awal,akhir)

2: HapusTengah (elemen,awal,akhir)

3: HapusBelakang (elemen,awal,akhir)

EndCase

Until (pil = 4)

EndProcedure

Procedure MenuCari

{I.S : pilihan (pil) user sudah terdefinisi}

{F.S: menampilkan menu pilihan}

Kamus :

pil : integer

Algoritma :

Repeat

Output ('MENU CARI DATA')

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

Output ('1. Cari Nomor Kursi')

Output ('2. Cari Rute Tujuan')

Output ('3. Kembali ke Menu Utama')

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

Output (' Pilihan : ') Input (pil)

Case pil of

1: CariKursi (awal)

2: CariTujuan (awal)

EndCase

Until (pil = 3)

EndProcedure

*{Algoritma Utama}*

Algoritma :

*{Penciptaan List}*

awal 🡨 nil

akhir 🡨 nil

*{Menu Utama}*

Repeat

Menu (pilihan)

Case pilihan of

1: MenuSisip

2: MenuHapus

3: MenuCari

4: TampilData

5: Keluar

EndCase

Until (pilihan = 5)