

Tugas Struktur Data ke – 4

Double Linked List

mengenai Pemesanan Tiket Kereta Api

Oleh:

1. Aziz (10110306)
2. Deden Masmunandar (10112206)
3. Milas Vernando Loby (10111253)
4. Mohammad Abdul Iman Syah (10111143)
5. Rosalina Fazriah (10111313)

Kelas : IF-6



Program Studi Teknik Informatika

Fakultas Teknik dan Ilmu Komputer

Universitas Komputer Indonesia

2013

BAB I

PENDAHULUAN

I.1 Latar Belakang Masalah

Dewasa ini perkembangan teknologi semakin pesat diiringi perkembangan sistem informasi yang berbasis teknologi. Perkembangan sistem informasi tersebut perlu didukung banyak faktor yang diharapkan dapat memberikan kesuksesan dari sistem informasi itu sendiri yang tercermin melalui kepuasan pemakai sistem informasi. Suatu sistem informasi akan sukses apabila didukung oleh beberapa faktor pendukungnya, diantaranya partisipasi pemakai.

Kepercayaan terhadap teknologi sistem informasi baru dalam mengevaluasi kinerja individu diperlukan oleh manajemen untuk memastikan bahwa sistem baru yang berbasis komputer dapat digunakan untuk mengendalikan kinerja bawahan. Keberhasilan sistem informasi suatu perusahaan tergantung bagaimana sistem itu dijalankan, kemudahan sistem itu bagi para pemakainya, dan pemanfaatan teknologi yang digunakan.

Tiket Kereta Api di beberapa daerah dalam mengelola pemesanantempat duduk masih menggunakan cara manual yaitu mencatat data pendaftaran di buku besar setelah itu menulisnya dikarcis yang telah disediakan , penggunaan sistem seperti itu dianggap kurang efektif dan efisien untuk saat ini.

Oleh karena itu, diperlukan adanya suatu aplikasi yang dapat membantu pemesanan tiket Kereta Apidalam pengolahan data. Berdasarkan latar belakang tersebut, maka penulis mengangkat topik ini untuk makalah dengan judul :

“PROGRAM PENGOLAHAN PEMESANAN TIKET KERETA API”

I.2 Identifikasi Masalah

Berdasarkan latar belakang yang dikemukakan di atas, maka diperoleh identifikasi sebagai berikut :

1. Pada proses pemesanan calon penumpang mengalami kesulitan karena banyaknya jumlah kursi yang sudah dipesan sehingga sering terjadi kesalahan data calon penumpang.
2. Pada proses pemesanan tiket, calon penumpang mengalami kesulitan ketika akan mencari data jurusan dan kursi mana yang telah penuh atau masih kosong.
3. Pada proses pembelian tiket, penumpang mengalami kesulitan dalam melakukan perhitungan biaya apabila data pemesanan banyak karena dilakukan secara manual.
4. Pada saat pembuatan tiket jurusan dan laporan pemesanan, penumpang mengalami kesulitan untuk mendapatkan data yang secara sistematis dan akurat.

I.3 Maksud dan Tujuan

Maksud dari pembuatan program aplikasi ini adalah untuk merancang dan mengimplementasikan program aplikasi pengolahan data pembelian tiket yang diambil dari informasi-informasi serta data-data yang telah di dapat.

Adapun tujuan dari pembuatan program aplikasi ini diantaranya:

1. Memberikan kemudahan kepada calon penumpang dalam proses pemesanan pendaftaran tiket kereta api.
2. Memberikan kemudahan kepada petugas tiket untuk melakukan pengolahan data pada transaksi pembelian tiket.
3. Memberikan kemudahan kepada petugas tiket untuk melakukan pengolahan data penumpang pada transaksi pembelian dan mempermudah dalam mencari data tiket yang terjual habis.

I.4 Batasan Masalah

Dalam pembuatan aplikasi ini penulis membatasi permasalahan yang disesuaikan dengan kemampuan yang ada, diantaranya:

Aplikasi yang akan dihasilkan yaitu aplikasi pengolahan data pemesanan tiket yang meliputi penyisipan data, penghapusan data, lihat data, dan keluar dari program.

BAB II

PEMBAHASAN TEORI LINKED LIST

A. Definisi

Double Linked List (DLL) adalah suatu cara pengolahan data yang bekerja dengan record dalam jumlah besar, sehingga membutuhkan alokasi memori dinamis yang besar pula. DLL biasanya digunakan pada saat alokasi memori konvensional tidak lagi bisa diandalkan. Sedangkan bekerja dengan data yang besar tidak dapat dihindari lagi, karena tidak jarang pula, data besar tersebut memiliki hubungan yang erat.

Di dalam DLL tidak hanya sekadar menampilkan setiap record-nya, melainkan dapat pula menambahkan record, menghapus beberapa record sesuai keinginan pengguna, sampai mengurutkan record. Kondisi tersebut memungkinkan dimilikinya satu rantai data yang panjang dan saling berhubungan. Pada Double Linked List, setiap node memiliki dua buah pointer ke sebelah kiri (prev) dan ke sebelah kanan (next). Gambar 1 memperlihatkan sebuah node dari Double Linked List.

Bertambah lagi komponen yang akan digunakan. Apabila dalam Single Linked List hanya memiliki head, curr dan node, maka untuk Double Linked List, ada satu penunjuk yang berfungsi sebagai akhir dari list: tail. Bagian kiri dari head akan menunjuk ke NULL. Demikian pula dengan bagian kanan dari tail. Setiap node saling terhubung dengan pointer kanan dan kiri.

B. Abstraksi Tipe Data Double Linked List

Abstraksi tipe data Double Linked List sedikit berbeda dengan Single Linked List, yaitu tinggal menambahkan pointer prev dan harus diawali dengan pembuatan struct tnode.

Kemudian, mendeklarasikan beberapa node yang akan digunakan sebagai head, tail, node aktif (curr) dan node sementara (node) seperti berikut:

Sama seperti pada pembuatan Single Linked List, dalam pembuatan Double Linked List ini, akan membuat sebuah perulangan sebanyak 5 kali untuk mengisi nilai 0 sampai 4 ke dalam field x untuk masing-masing node.

Secara umum, kode yang dibuat hampir sama dengan pembuatan Single Linked List. Hanya bedanya, pada Double Linked List, pointer kiri dan kanan dihubungkan dengan suatu node.

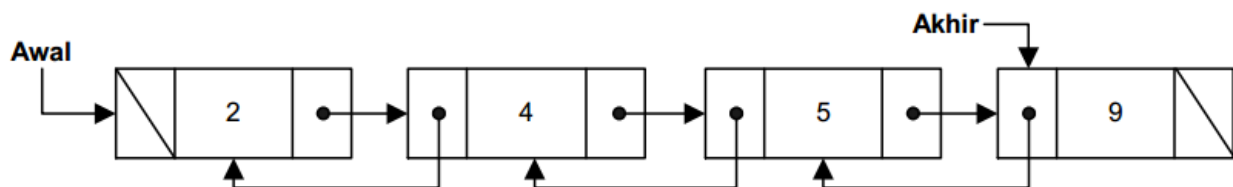
Pertama-tama, tentunya perlu diuji apakah head bernilai NULL yang artinya belum ada satu node pun yang tercipta. Apabila demikian, maka node yang dibuat akan menjadi head. Node aktif (curr) pun diset sesuai node yang dibuat. Dan sebagai konsekuensi dari Double Linked List, maka diatur pointer prev pada head menunjuk ke NULL.

Untuk menguji keberhasilan Double Linked List, awal list sampai akhir list akan dicetak dengan deklarasi:

Dan karena apa yang dibentuk adalah Double Linked List, maka juga mencetak dari tail sampai head, dengan deklarasi: Untuk membebaskan memori teralokasi, dilakukan dengan pemanggilan fungsi free(). Kode selengkapnya:

Operasi pada linked list tidak hanya pembuatan dan pencetakan. Suatu saat, mungkin perlu untuk menghapus node yang terletak di tengah-tengah list. Atau bahkan mungkin perlu menyelipkan node di tengah-tengah node.

Double linked list adalah suatu linked list yang mempunyai 2 penunjuk yaitu penunjuk ke simpul sebelumnya dan simpul berikutnya. Perhatikan gambar di bawah ini:



Deklarasi secara umum double linked list :

Type

nama_pointer = ↑Simpul

Simpul = Record

 medan_data : tippedata

 medan_sambungan_kiri, medan_sambungan_kanan : Namapointer

EndRecord

nama_var_pointer : nama_pointer

Contoh:

Type

Point = ↑Data

Data = Record

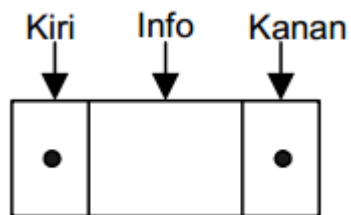
 < info : char

 prev, next : Point >

Endrecord

awal, akhir: Point

Jadi satu simpul di double linked list adalah sebagai berikut :



BAB III

ALGORITMA DARI KASUS

Berikut ini adalah Algoritma dari Program Tiket Kereta Api adalah sebagai berikut:

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 \leftarrow upcase(elemen.kodetujuan)

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

If (elemen.kodetujuan = 'CRB')

Then

elemen.tujuan \leftarrow 'Cirebon'

Else

If (elemen.kodetujuan = 'BDG')

Then

elemen.tujuan \leftarrow 'Bandung'

Else

If (elemen.kodetujuan = 'JKT')

Then

elemen.tujuan \leftarrow '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 \leftarrow 'Eksekutif'

2: elemen.kelas \leftarrow 'Bisnis'

3: elemen.kelas \leftarrow 'Ekonomi'

EndCase

If (elemen.kodetujuan = 'CRB')

Then

If (elemen.kodekelas = 1)

Then

elemen.harga \leftarrow 100000

Else

If (elemen.kodekelas = 2)

Then

elemen.harga \leftarrow 80000

Else

If (elemen.kodekelas = 3)

Then

elemen.harga \leftarrow 45000

EndIf

EndIf

EndIf

Else

If (elemen.kodetujuan = 'BDG')

Then

If (elemen.kodekelas = 1)

Then

elemen.harga \leftarrow 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
      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 \leftarrow awal

elemen \leftarrow phapus \uparrow .info

If (awal = akhir)

Then

awal \leftarrow nil

akhir \leftarrow nil

Else

awal \leftarrow awal \uparrow .next

awal \uparrow .prev \leftarrow 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 \leftarrow akhir

elemen \leftarrow phapus \uparrow .info

If (awal = akhir)

Then

awal \leftarrow nil

akhir \leftarrow nil

Else

akhir \leftarrow akhir \uparrow .prev

akhir \uparrow .next \leftarrow 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)

BAB IV

LISTING PROGRAM

Berikut ini adalah *listing* dari program pascal “Penjualan Tiket Kereta Api”.

```
program dltiket;
uses crt;

type
  Tiket = record
    nokursi    : integer;
    nama       : string[18];
    alamat     : string[14];
    nohape     : string[12];
    kodetujuan: string;
    tujuan     : string;
    kodekelas  : integer;
    kelas      : string;
    harga      : integer;
  end;

  RTiket = Tiket;

  Pointer = ^Simpul;
  Simpul = record
    prev : Pointer;
    info : RTiket;
    next : Pointer;
  end;

var
  data,awal,akhir : Pointer;
  elemen : RTiket;
  pilihan : integer;
  total : integer;

{Interface}
procedure Kotak(awalkolom,awalbaris,akhirkolom,akhirbaris:integer; latar:byte); //Kotak
var
  i : integer;

begin
  textbackground(latar);
  window(awalkolom,awalbaris,akhirkolom,akhirbaris);clrscr;window(1,1,80,25);
  gotoxy(awalkolom,awalbaris);write(#218);
  gotoxy(akhirkolom,awalbaris);write(#191);
```

```

gotoxy(akhirkolom,akhirbaris);write(#217);
gotoxy(awalkolom,akhirbaris);write(#192);

for i:=awalkolom+1 to akhirkolom-1 do
begin
    gotoxy(i,awalbaris);write(#196);
    gotoxy(i,akhirbaris);write(#196);
end;

for i:=awalbaris+1 to akhirbaris-1 do
begin
    gotoxy(akhirkolom,i);write(#179);
    gotoxy(awalkolom,i);write(#179);
end;
end;

procedure Header;
begin
    gotoxy(2,1);textbackground(CYAN);write(' NO ');
    gotoxy(7,1);textbackground(CYAN);write('  NAMA  ');
    gotoxy(26,1);textbackground(CYAN);write('  ALAMAT  ');
    gotoxy(41,1);textbackground(CYAN);write(' NO. HP ');
    gotoxy(54,1);textbackground(CYAN);write(' TUJUAN ');
    gotoxy(63,1);textbackground(CYAN);write(' KELAS ');
    gotoxy(73,1);textbackground(CYAN);write(' HARGA ');
    writeln();
end;

{Prototype}
procedure Isi(var elemen:RTiket);
begin
    clrscr();
    Kotak(18,5,62,21,5);
    gotoxy(31,6);writeln('INPUT DATA PEMBELI');
    gotoxy(19,7);writeln('-----');
    gotoxy(25,whereY+1);write('Nomor Kursi : '); readln(elemen.nokursi);
    gotoxy(25,whereY+1);write('Nama      : '); readln(elemen.nama);
    gotoxy(25,whereY+1);write('Alamat   : '); readln(elemen.alamat);
    gotoxy(25,whereY+1);write('No. HP   : '); readln(elemen.nohape);
    repeat
        gotoxy(25,whereY+1);write('Kode Tujuan : '); readln(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';

repeat
    gotoxy(25,whereY+1);write('Kode Kelas : '); readln(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';
end;

if(elemen.kodetujuan='CRB') then
begin
    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;
end
else if(elemen.kodetujuan='BDG') then
begin
    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;
end
else if(elemen.kodetujuan='JKT') then
begin
    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;
end;
end;

procedure Tampil(data:Pointer);
begin
    textbackground(0);clrscr();
    Header;
    data:=awal;
    while(data<>nil) do
    begin
        textbackground(0);
        gotoxy(3,whereY);write(data^.info.nokursi);
        gotoxy(7,whereY);write(data^.info.nama);
        gotoxy(26,whereY);write(data^.info.alamat);
        gotoxy(41,whereY);write(data^.info.nohape);
        gotoxy(54,whereY);write(data^.info.tujuan);
        gotoxy(63,whereY);write(data^.info.kelas);
        gotoxy(73,whereY);writeln(data^.info.harga);
    end;
end;

```

```

        data:=data^.next;
    end;
    readkey();
end;

{Penyisipan}
procedure SisipDepan(elemen:RTiket; var awal,akhir:Pointer); //Sisip Depan
var
    baru:Pointer;

begin
    new(baru);
    baru^.info:=elemen;
    baru^.prev:=nil;

    if(awal=nil) then
        begin
            baru^.next:=nil;
            akhir:=baru;
        end
    else
        begin
            baru^.next:=awal;
            awal^.prev:=baru;
        end;
    awal:=baru;
    Tampil(data);
end;

procedure SisipTengah(elemen:RTiket; var awal,akhir:Pointer); //Sisip Tengah
var
    baru,bantu:Pointer;
    ketemu:boolean;
    datasisip:integer;

begin
    new(baru);
    baru^.info:=elemen;

    if(awal=nil) then
        begin
            baru^.prev:=nil;
            baru^.next:=nil;
            awal:=baru;
            akhir:=baru;
        end
    else

```

```

begin
    clrscr();
    Tampil(data);
    Kotak(18,12,60,14,0);
    gotoxy(20,13);write('Data Akan Disisipkan Sebelum Nomor : '); readln(datisisip);
    bantu:=awal;
    ketemu:=false;

    while(not ketemu) and (bantu<>nil) do
    begin
        if(datisisip=bantu^.info.nokursi) then
        begin
            ketemu:=true;
        end
        else
        begin
            bantu:=bantu^.next;
        end;
    end;

    if(ketemu) then
    begin
        if(bantu=awal) then
        begin
            SisipDepan(elemen,awal,akhir);
        end
        else
        begin
            new(baru);
            baru^.info:=elemen;
            baru^.next:=bantu;
            baru^.prev:=bantu^.prev;
            bantu^.prev^.next:=baru;
            bantu^.prev:=baru;
            Tampil(data);
        end;
    end
    else
    begin
        gotoxy(22,16);write('Data dengan Nomor ',datisisip,' tidak ditemukan');
        readkey();
    end;
end;
end;

```



```

procedure SisipBelakang(elemen:RTiket; var awal,akhir:Pointer); //Sisip Belakang
var
    baru:Pointer;

begin
    new(baru);
    baru^.info:=elemen;
    baru^.next:=nil;

    if(akhir=nil) then
        begin
            baru^.prev:=nil;
            awal:=baru
        end
    else
        begin
            baru^.prev:=akhir;
            akhir^.next:=baru;
        end;
    akhir:=baru;
    Tampil(data);
end;

{Penghapusan}
procedure HapusDepan(var elemen:RTiket; var awal,akhir:Pointer); //Hapus Depan
var
    phapus:Pointer;

begin
    clrscr();
    phapus:=awal;
    elemen:=phapus^.info;

    if(awal=akhir) then
        begin
            awal:=nil;
            akhir:=nil;
        end
    else
        begin
            awal:=awal^.next;
            awal^.prev:=nil;
        end;
    dispose(phapus);
    gotoxy(33,14);write('Data telah dihapus');
    readkey();
end;

```

```

procedure HapusBelakang(var elemen:RTiket; var awal,akhir:Pointer); //Hapus Belakang
var
    phapus:Pointer;

begin
    clrscr();
    phapus:=akhir;
    elemen:=phapus^.info;

    if(awal=akhir) then
    begin
        awal:=nil;
        akhir:=nil;
    end
    else
    begin
        akhir:=akhir^.prev;
        akhir^.next:=nil;
    end;
    dispose(phapus);
    gotoxy(33,14);write('Data Telah Dihapus');
    readkey();
end;

```

```

procedure HapusTengah(var elemen:RTiket; var awal,akhir:Pointer); //Hapus Tengah
var
    phapus:Pointer;
    ketemu:boolean;
    datahapus:integer;

begin
    clrscr();
    if(awal=akhir) then
    begin
        phapus:=awal;
        elemen:=phapus^.info;
        awal:=nil;
        akhir:=nil;
        dispose(phapus);
        gotoxy(33,14);write('Data Telah Dihapus');
        readkey();
    end
    else
    begin
        Tampil(data);
        Kotak(23,12,54,14,0);
        gotoxy(25,13);write('Nomor Yang Akan Dihapus : '); readln(datahapus);
    end
end;

```

```

phapus:=awal;
ketemu:=false;

while(not ketemu) and (phapus<>nil) do
begin
    if(datahapus=phapus^.info.nokursi) then
    begin
        ketemu:=true;
    end
    else
    begin
        phapus:=phapus^.next;
    end;
end;

if(ketemu) then
begin
    elemen:=phapus^.info;

    if(phapus=awal) then
    begin
        HapusDepan(elemen,awal,akhir);
    end
    else if(phapus=akhir) then
    begin
        HapusBelakang(elemen,awal,akhir);
    end
    else
    begin
        phapus^.prev^.next:=phapus^.next;
        phapus^.next^.prev:=phapus^.prev;
        dispose(phapus);
        clrscr();
        gotoxy(33,14);write('Data Telah Dihapus');
        readkey();
    end;
end
else
begin
    gotoxy(22,whereY+1);write('Data dengan Nomor ',datahapus,' tidak ditemukan');
    readkey();
end;
end;
end;

```

```

{Pencarian}
procedure CariKursi(awal:Pointer);    //Cari Nomor Kursi(Unik)
var
    bantu:Pointer;
    ketemu:boolean;
    datacari:integer;

begin
    clrscr();
    Kotak(23,12,55,14,0);
    gotoxy(25,13);write('Nomor Kursi yang Dicari : '); readln(datacari);
    bantu:=awal;
    ketemu:=false;

    while(not ketemu) and (bantu<>nil) do
    begin
        if(datacari=bantu^.info.nokursi) then
        begin
            ketemu:=true;
        end
        else
        begin
            bantu:=bantu^.next;
        end;
    end;

    if(ketemu) then
    begin
        textbackground(0);clrscr();
        Header;
        textbackground(0);
        gotoxy(3,whereY);write(bantu^.info.nokursi);
        gotoxy(7,whereY);write(bantu^.info.nama);
        gotoxy(26,whereY);write(bantu^.info.alamat);
        gotoxy(41,whereY);write(bantu^.info.nohape);
        gotoxy(54,whereY);write(bantu^.info.tujuan);
        gotoxy(63,whereY);write(bantu^.info.kelas);
        gotoxy(73,whereY);write(bantu^.info.harga);
        readkey();
    end
    else
    begin
        gotoxy(22,16);write('Data dengan Nomor ',datacari, ' Tidak Ditemukan');
        readkey();
    end;
end;

```

```

procedure CariTujuan(awal:Pointer);  //Cari Jurusan(Tidak Unik)
var
    bantu:Pointer;
    datacari:string;

begin
    clrscr();
    Kotak(25,12,56,14,0);
    gotoxy(27,13);write('Tujuan yang Dicari : '); readln(datacari);
    clrscr();
    Header;
    textbackground(0);
    bantu:=awal;

    repeat
        if(datacari=bantu^.info.tujuan) then
            begin
                gotoxy(3,whereY);write(bantu^.info.nokursi);
                gotoxy(7,whereY);write(bantu^.info.nama);
                gotoxy(26,whereY);write(bantu^.info.alamat);
                gotoxy(41,whereY);write(bantu^.info.nohape);
                gotoxy(54,whereY);write(bantu^.info.tujuan);
                gotoxy(63,whereY);write(bantu^.info.kelas);
                gotoxy(73,whereY);writeln(bantu^.info.harga);
                bantu:=bantu^.next;
            end
        else
            begin
                bantu:=bantu^.next;
            end;
    until(bantu=nil);
    readkey();
end;

{Total Harga}
procedure TotalHarga(var total:integer);
begin
    data:=awal;
    total:=0;
    while(data<>nil) do
        begin
            total:=total+data^.info.harga;
            data:=data^.next;
        end;
    gotoxy(73,whereY);writeln('-----+');
    gotoxy(59,whereY);write('Total Harga = ',total);
    readkey();

```

```

end;

{Pengurutan}
procedure UrutKursi(awal,akhir:Pointer);
var
  min,i,j:Pointer;
  temo:string;
  temp:integer;

begin
  clrscr();
  i:=awal;
  while(i<>akhir) do
    begin
      min:=i;
      j:=i^.next;
      while(j<>nil) do
        begin
          if(j^.info.nokursi < min^.info.nokursi) then min:=j;
          j:=j^.next;
        end;

        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;
    end;
end;

{Tampil Data}
procedure TampilData;
begin
    if(awal<>nil) and (akhir<>nil) then
    begin
        UrutKursi(awal,akhir);
        Tampil(data);
        TotalHarga(total);
    end
    else
    begin
        clrscr();
        gotoxy(34,13);write('List Kosong');
        readkey();
    end;
end;

{Penghancuran}
procedure Penghancuran(var awal:Pointer);
var
    phapus:Pointer;

begin
    clrscr();
    phapus:=awal;
    while(awal<>nil) do
    begin
        phapus:=awal;
        dispose(phapus);
        awal:=awal^.next;
    end;
    awal:=nil;
    akhir:=nil;
    gotoxy(28,12);write('Semua Data Telah Dihapus');
    gotoxy(19,13);write('Terima Kasih telah menggunakan program ini');
    readkey();
end;

```

```

procedure Keluar;    //Keluar
begin
    Penghancuran(awal);
end;

{Menu}
procedure Menu(var pil:integer);    //Menu Utama
begin
    clrscr();
    Kotak(28,8,53,19,1);
    gotoxy(30,9);    writeln(' MENU PENJUALAN TIKET');
    gotoxy(30,whereY);writeln('PT KERETA API TORABIKA');
    gotoxy(29,whereY);writeln('=====');
    gotoxy(30,whereY);writeln('1. Sisip Data Pembeli');
    gotoxy(30,whereY);writeln('2. Hapus Data Pembeli');
    gotoxy(30,whereY);writeln('3. Cari Data Pembeli');
    gotoxy(30,whereY);writeln('4. Tampil Data Pembeli');
    gotoxy(30,whereY);writeln('5. Keluar');
    gotoxy(29,whereY);writeln('=====');
    gotoxy(30,whereY);write ('    Pilihan : '); readln(pil);
end;

procedure MenuSisip;    //Menu Sisip
var
    pil:integer;

begin
    repeat
        clrscr();
        Kotak(27,9,54,18,2);
        gotoxy(29,10);    writeln('    MENU SISIP DATA');
        gotoxy(28,whereY);writeln('=====');
        gotoxy(29,whereY);writeln('1. Sisip Depan');
        gotoxy(29,whereY);writeln('2. Sisip Tengah');
        gotoxy(29,whereY);writeln('3. Sisip Belakang');
        gotoxy(29,whereY);writeln('4. Kembali ke Menu Utama');
        gotoxy(28,whereY);writeln('=====');
        gotoxy(29,whereY);write ('    Pilihan : '); readln(pil);
        case pil of
            1: begin
                    Isi(elemen);
                    SisipDepan(elemen,awal,akhir);
                end;
            2: begin
                    Isi(elemen);
                    SisipTengah(elemen,awal,akhir);
                end;
        end;
    end;
end;

```



```

        3: begin
            Isi(elemen);
            SisipBelakang(elemen,awal,akhir);
        end;
    end;
until(pil=4);
end;

procedure MenuHapus; //Menu Hapus
var
    pil:integer;

begin
    repeat
        clrscr();
        Kotak(27,9,54,18,3);
        gotoxy(29,10); writeln(' MENU HAPUS DATA');
        gotoxy(28,whereY);writeln('=====');
        gotoxy(29,whereY);writeln('1. Hapus Depan');
        gotoxy(29,whereY);writeln('2. Hapus Tengah');
        gotoxy(29,whereY);writeln('3. Hapus Belakang');
        gotoxy(29,whereY);writeln('4. Kembali ke Menu Utama');
        gotoxy(28,whereY);writeln('=====');
        gotoxy(29,whereY);write (' Pilihan : '); readln(pil);
        case pil of
            1: HapusDepan(elemen,awal,akhir);
            2: HapusTengah(elemen,awal,akhir);
            3: HapusBelakang(elemen,awal,akhir);
        end;
    until(pil=4);
end;

procedure MenuCari; //Menu Cari
var
    pil:integer;

begin
    repeat
        clrscr();
        Kotak(27,9,54,17,4);
        gotoxy(29,10); writeln(' MENU CARI DATA');
        gotoxy(28,whereY);writeln('=====');
        gotoxy(29,whereY);writeln('1. Cari Nomor Kursi');
        gotoxy(29,whereY);writeln('2. Cari Rute Tujuan');
        gotoxy(29,whereY);writeln('3. Kembali ke Menu Utama');
        gotoxy(28,whereY);writeln('=====');
        gotoxy(29,whereY);write (' Pilihan : '); readln(pil);
    
```

```

        case pil of
            1: CariKursi(awal);
            2: CariTujuan(awal);
        end;
    until(pil=3);
end;

begin
    {Penciptaan List}
    awal:=nil;
    akhir:=nil;

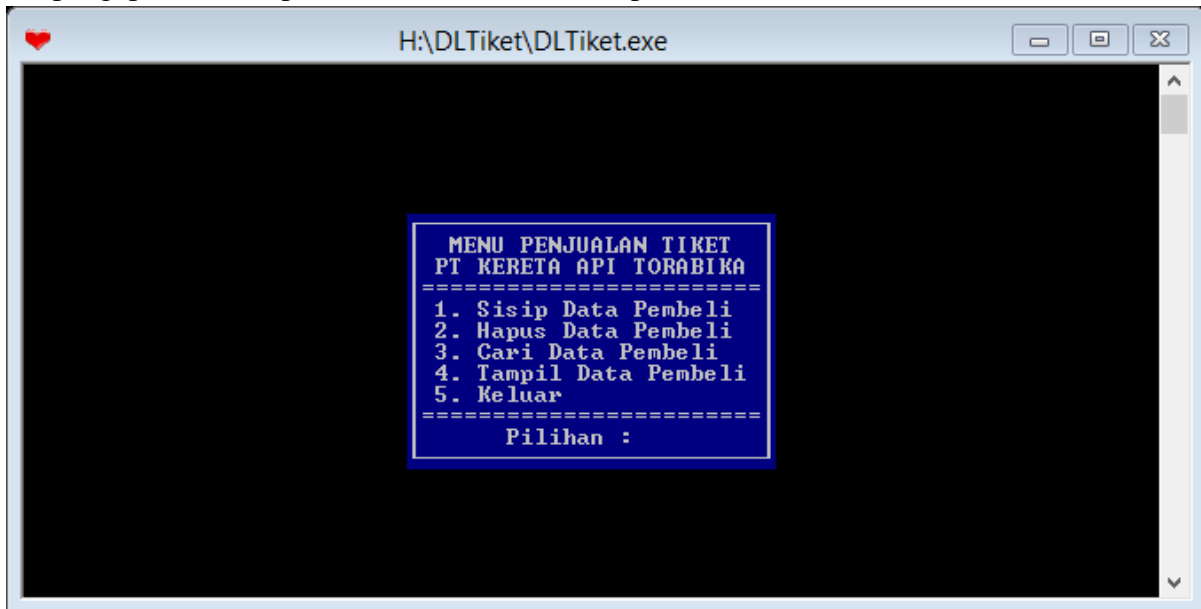
    {Menu Utama}
    repeat
        Menu(pilihan);
        case pilihan of
            1: MenuSisip;
            2: MenuHapus;
            3: MenuCari;
            4: TampilData;
            5: Keluar;
        end;
    until(pilihan=5);
end.

```

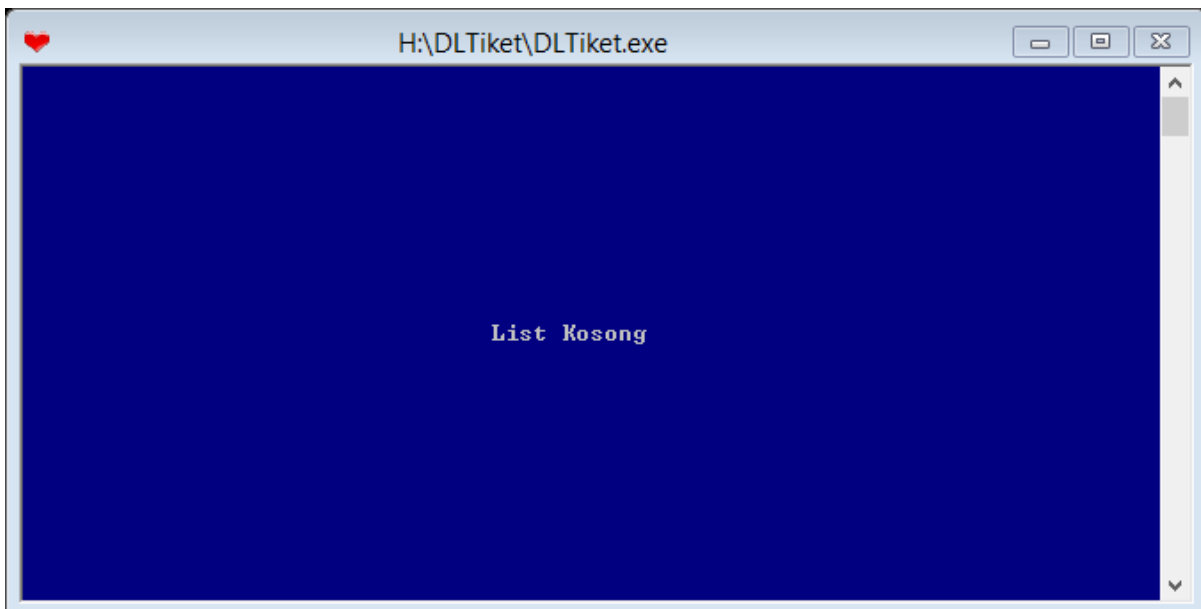
BAB V

LAYAR TAMPILAN

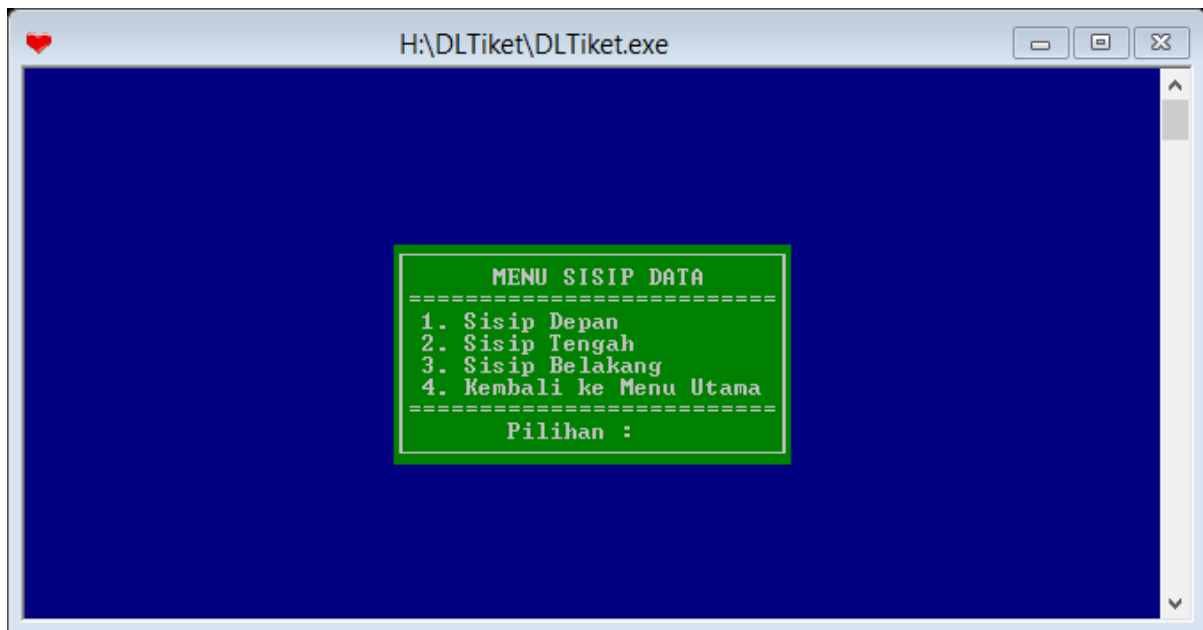
1. Saat menjalankan program, user akan dihadapkan langsung pada interface program. Pada interface program pertama, user akan dihadapkan pada list menu penyisipan data, pengapusan data, pencarian data, dan menampilkan data.



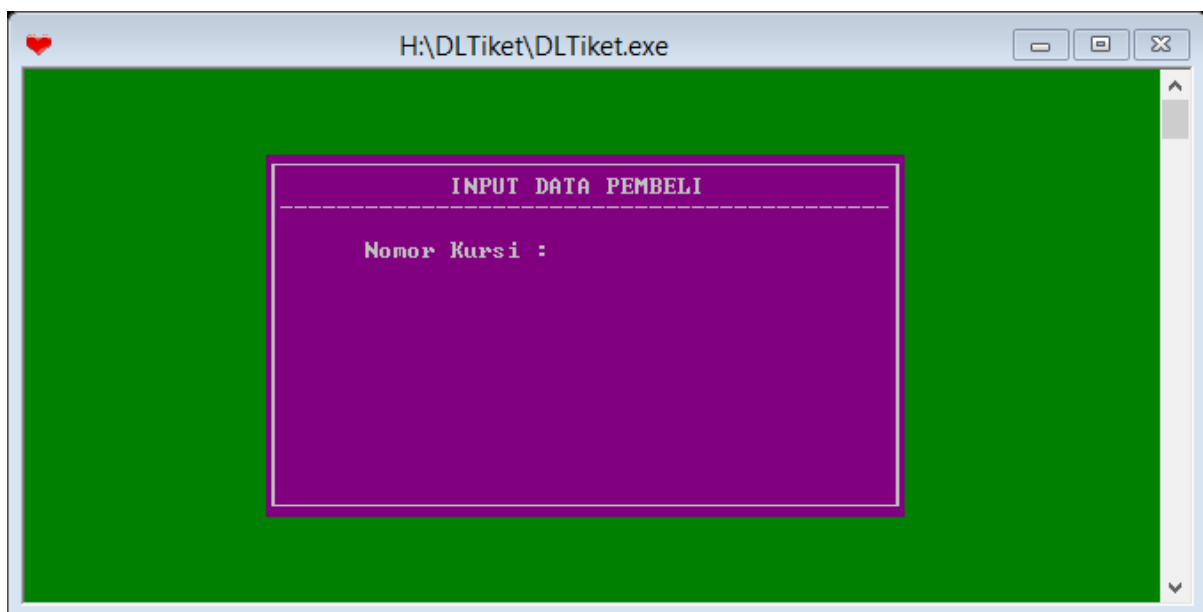
2. Masukkan nomor pilihan 4 untuk melihat bahwa list masih kosong, sehingga akan menampilkan seperti berikut ini :



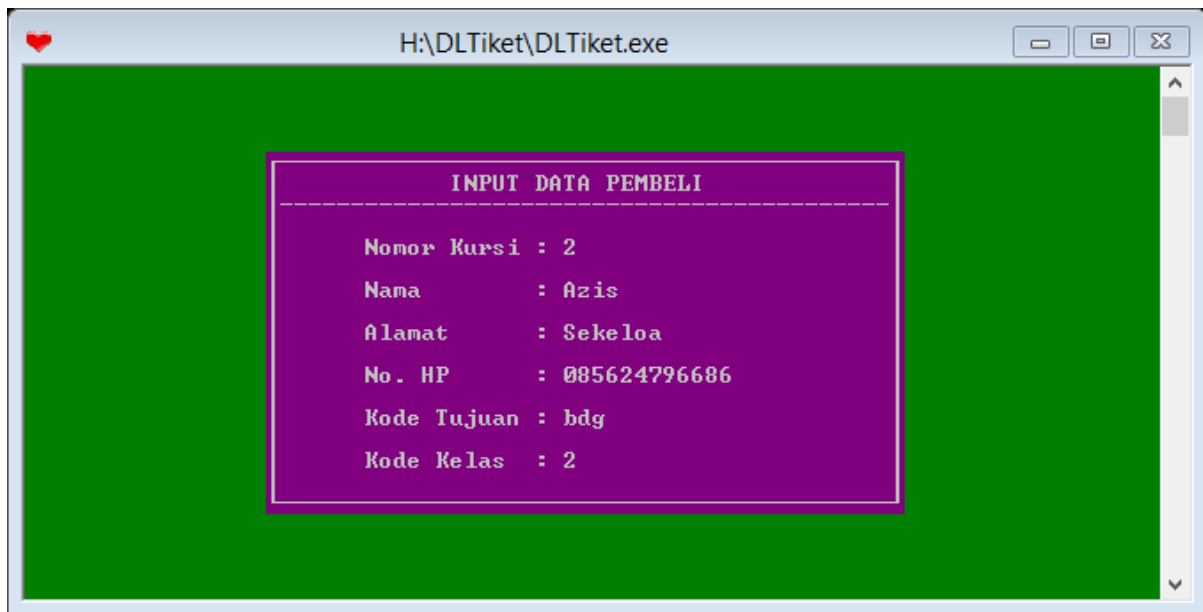
3. Kemudian tekan sembarang tombol untuk kembali ke menu utama. Kemudian masukkan sisip data yang terdapat pada nomor pilihan 1, maka akan menampilkan tampilan sebagai berikut.



4. Kemudian masukkan sisip depan untuk menginput data pembelian tiket. Maka pilihlah nomor 1, setelah itu akan muncul tampilan sebagai berikut.



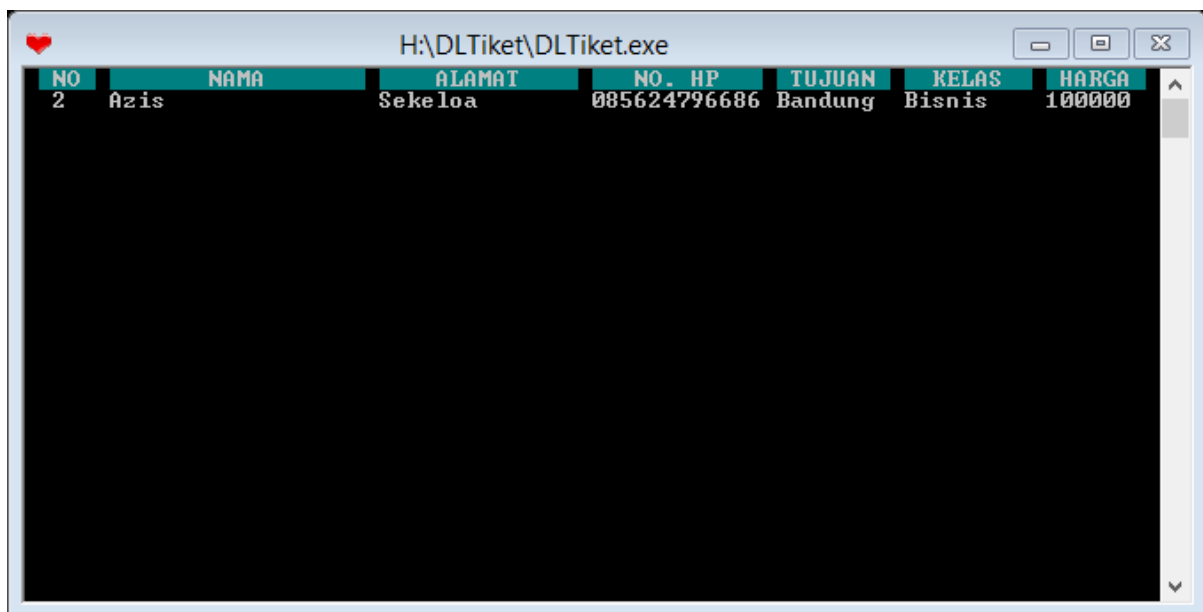
5. Kemudian isikan input data pembeli sesuai dengan data pembeli tiket.



H:\DLTiket\DLTiket.exe

INPUT DATA PEMBELI

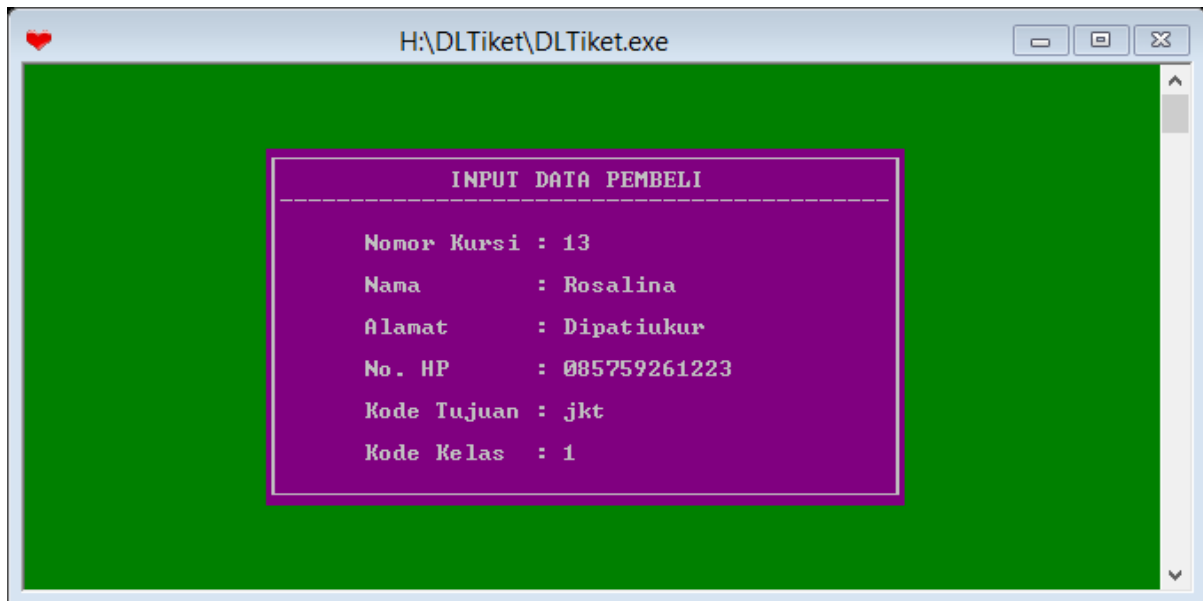
Nomor Kursi : 2
Nama : Azis
Alamat : Sekeloa
No. HP : 085624796686
Kode Tujuan : bdg
Kode Kelas : 2



H:\DLTiket\DLTiket.exe

| NO | NAMA | ALAMAT | NO. HP | TUJUAN | KELAS | HARGA |
|----|------|---------|--------------|---------|--------|--------|
| 2 | Azis | Sekeloa | 085624796686 | Bandung | Bisnis | 100000 |

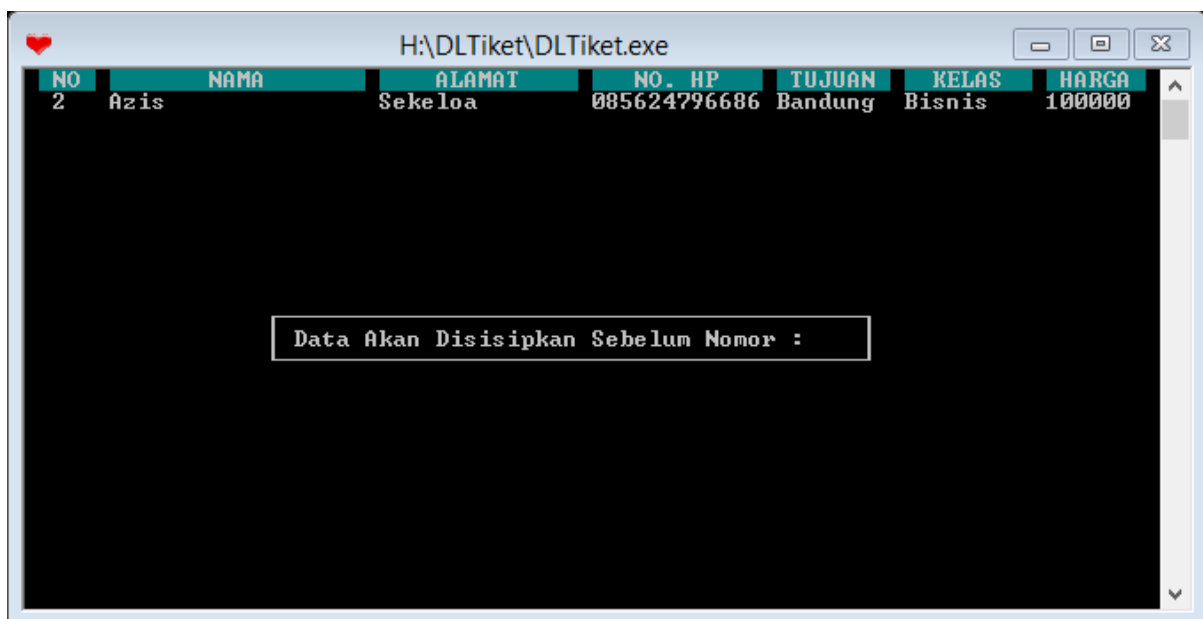
6. Kemudian tekan Enter dan tampilan akan kembali ke Menu utama. Lanjutkan menginputkan data pembelian tiket menggunakan sisip tengah seperti berikut ini.



H:\DLTiket\DLTiket.exe

INPUT DATA PEMBELI

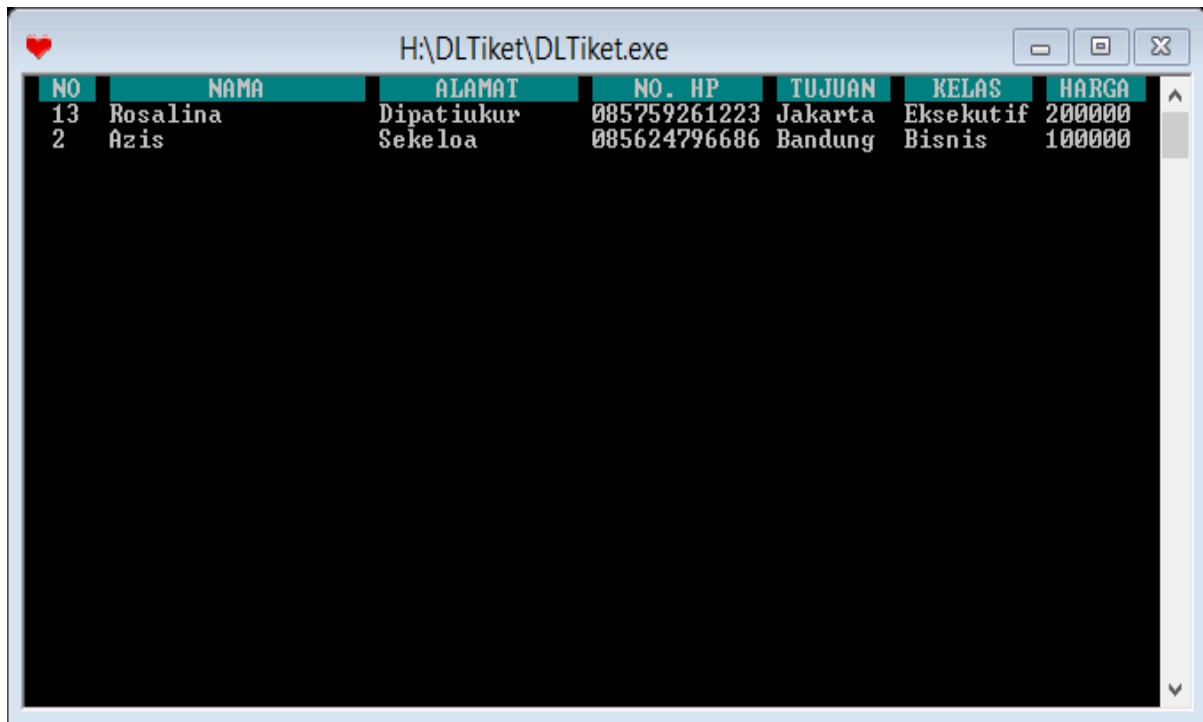
Nomor Kursi : 13
Nama : Rosalina
Alamat : Dipatiukur
No. HP : 085759261223
Kode Tujuan : jkt
Kode Kelas : 1



H:\DLTiket\DLTiket.exe

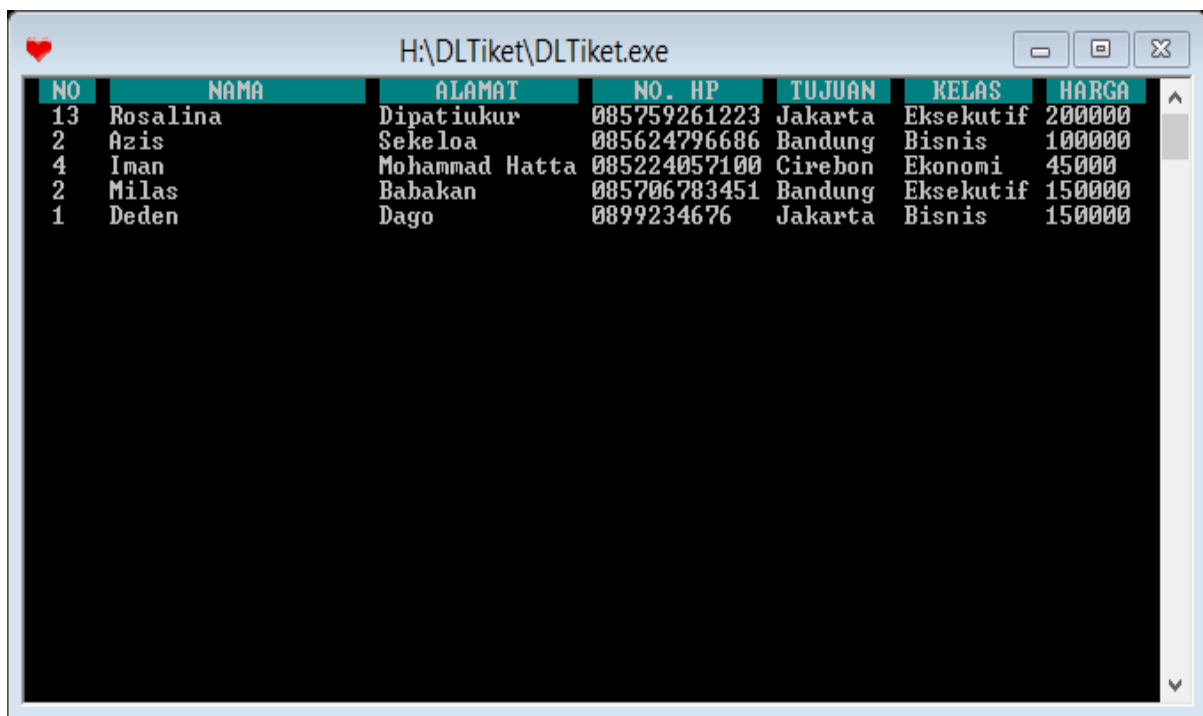
| NO | NAMA | ALAMAT | NO. HP | TUJUAN | KELAS | HARGA |
|----|------|---------|--------------|---------|--------|--------|
| 2 | Azis | Sekeloa | 085624796686 | Bandung | Bisnis | 100000 |

Data Akan Disisipkan Sebelum Nomor :



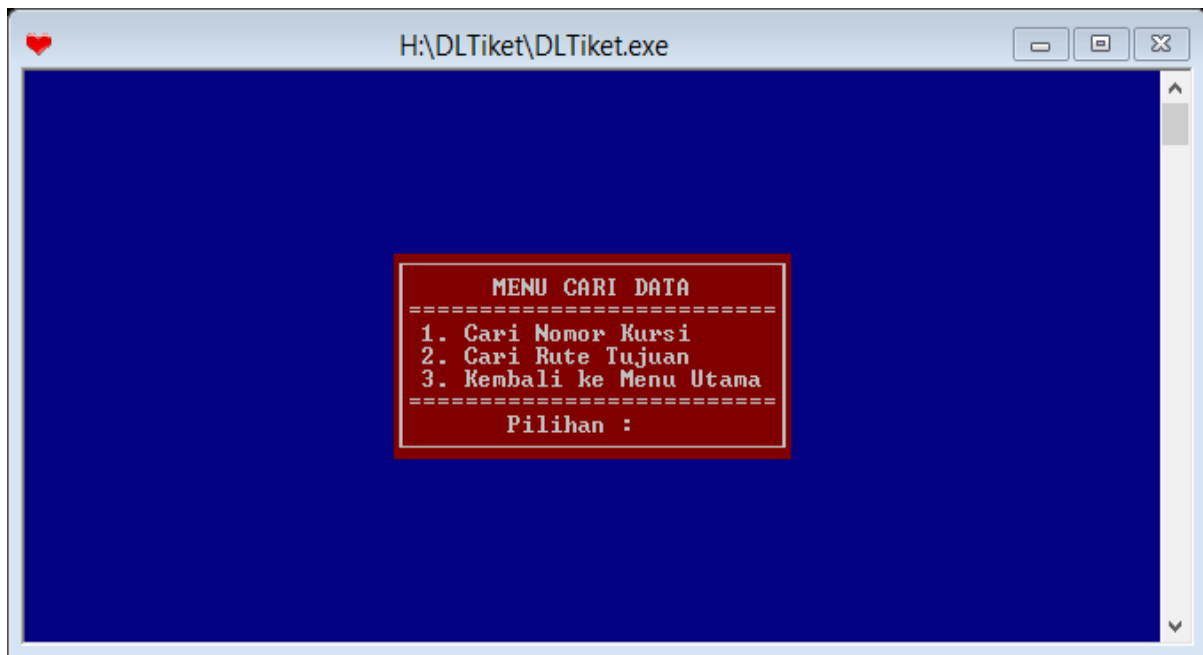
| NO | NAMA | ALAMAT | NO. HP | TUJUAN | KELAS | HARGA |
|----|----------|------------|--------------|---------|-----------|--------|
| 13 | Rosalina | Dipatiukur | 085759261223 | Jakarta | Eksekutif | 200000 |
| 2 | Azis | Sekeloa | 085624796686 | Bandung | Bisnis | 100000 |

7. Masukkan data sesuai dengan cara yang sebelumnya hingga mendapatkan jumlah tiket data yang diinginkan.
8. Setelah menginputkan data, maka seperti inilah tampilan dari list pembeli tiket kereta api.



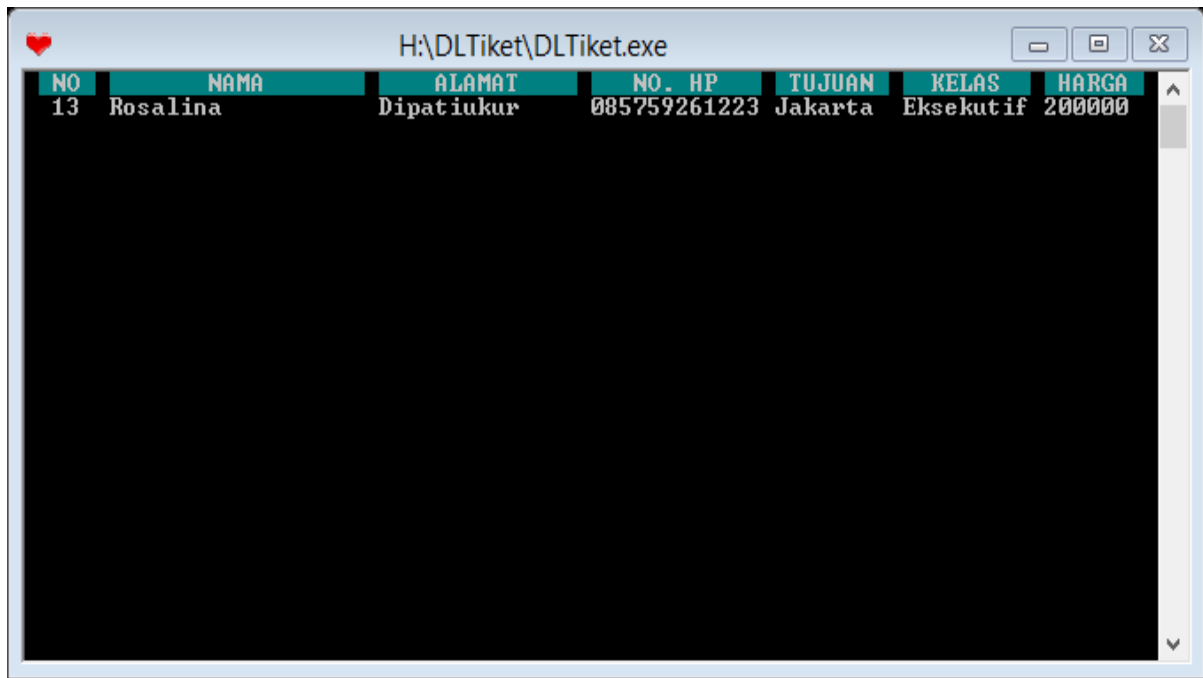
| NO | NAMA | ALAMAT | NO. HP | TUJUAN | KELAS | HARGA |
|----|----------|----------------|--------------|---------|-----------|--------|
| 13 | Rosalina | Dipatiukur | 085759261223 | Jakarta | Eksekutif | 200000 |
| 2 | Azis | Sekeloa | 085624796686 | Bandung | Bisnis | 100000 |
| 4 | Iman | Mohammad Hatta | 085224057100 | Cirebon | Ekonomi | 45000 |
| 2 | Milas | Babakan | 085706783451 | Bandung | Eksekutif | 150000 |
| 1 | Deden | Dago | 0899234676 | Jakarta | Bisnis | 150000 |

9. Kemudian kembali ke Menu utama untuk melakukan pencarian data. Maka setelah kembali ke menu utama, masukkan nomor pilihan 3. Dan akan menampilkan menu cari data sebagai berikut.



10. Inputkan kembali nomor 1 untuk mencari berdasarkan nomor kursi, sebagai berikut.

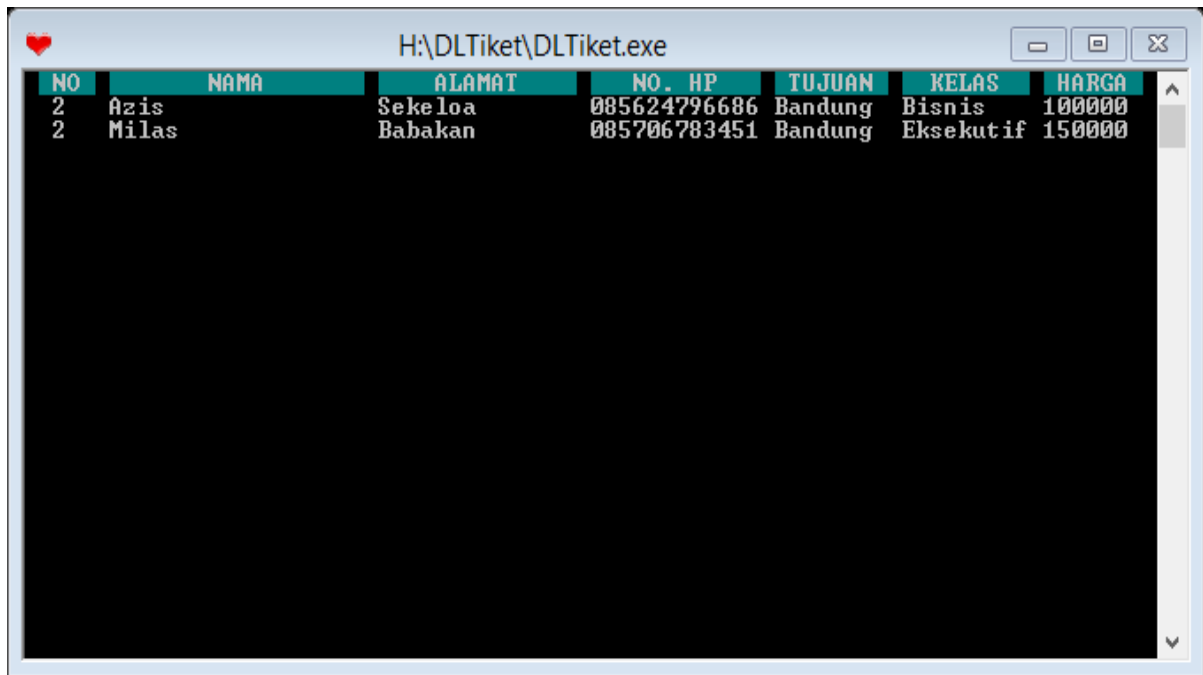




| NO | NAMA | ALAMAT | NO. HP | TUJUAN | KELAS | HARGA |
|----|----------|------------|--------------|---------|-----------|--------|
| 13 | Rosalina | Dipatiukur | 085759261223 | Jakarta | Eksekutif | 200000 |

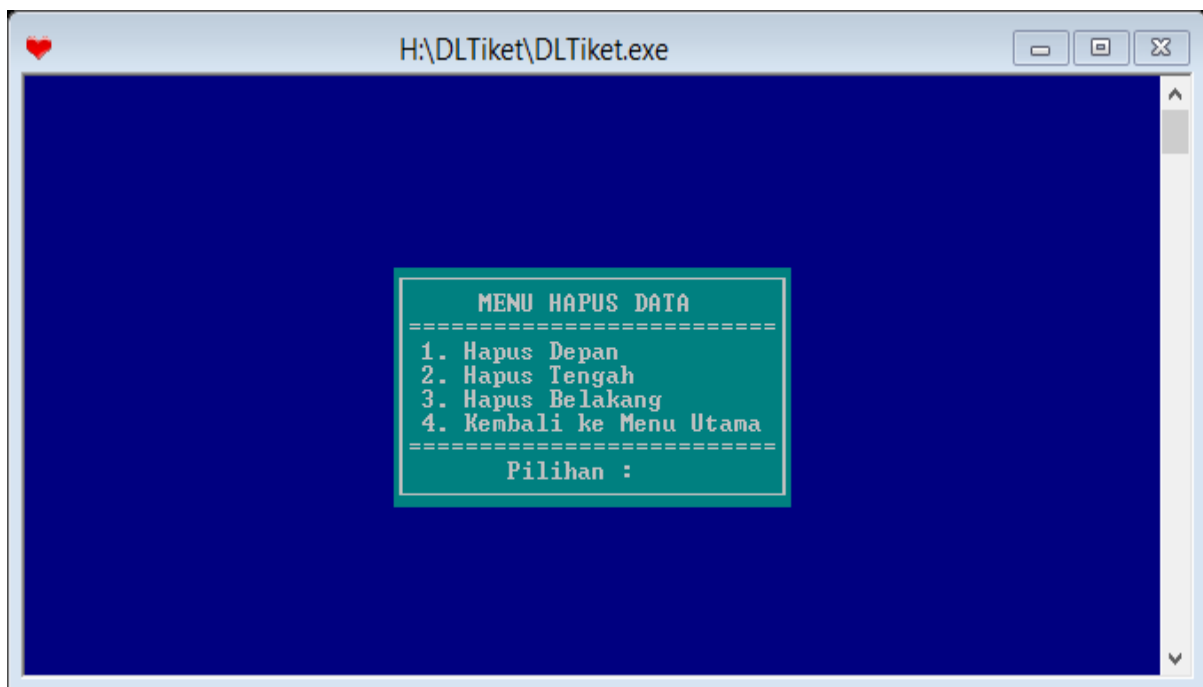
11. Kemudian coba kembali untuk mencari data berdasarkan rute tujuan pada nomor 2, maka akan menampilkan tampilan sebagai berikut.



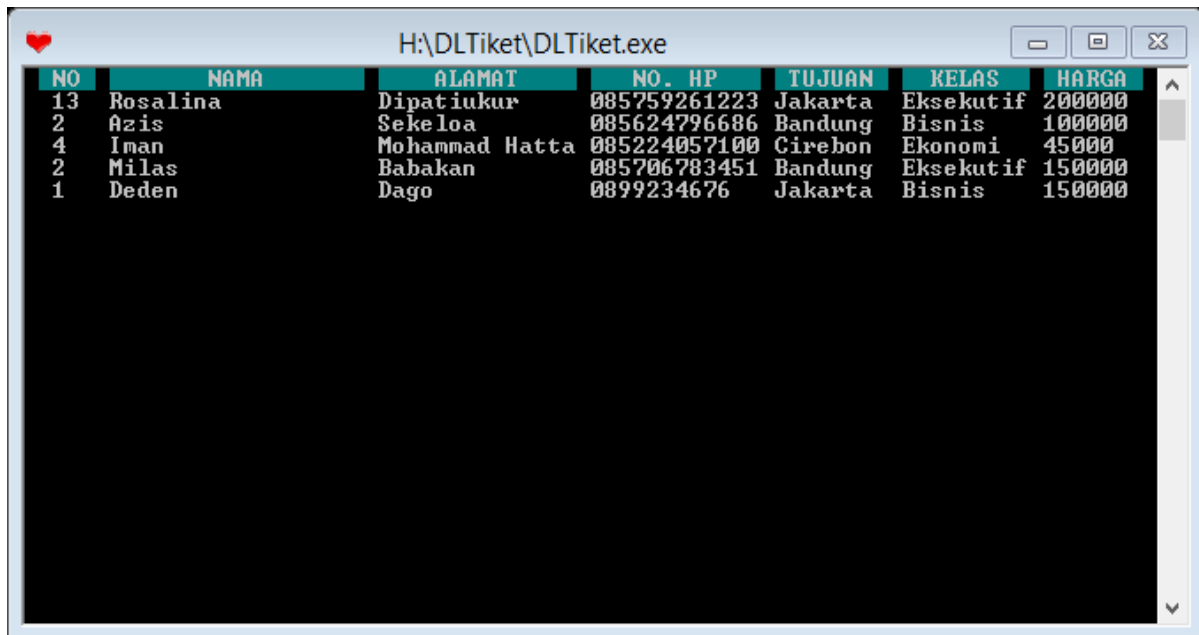


| NO | NAMA | ALAMAT | NO. HP | TUJUAN | KELAS | HARGA |
|----|-------|---------|--------------|---------|-----------|--------|
| 2 | Azis | Sekeloa | 085624796686 | Bandung | Bisnis | 100000 |
| 2 | Milas | Babakan | 085706783451 | Bandung | Eksekutif | 150000 |

12. Kembali ke Menu Utama untuk melakukan menu hapus data. Terdapat 3 daftar yang terdapat pada menu hapus, yaitu hapu depan, tengah, dan belakang.

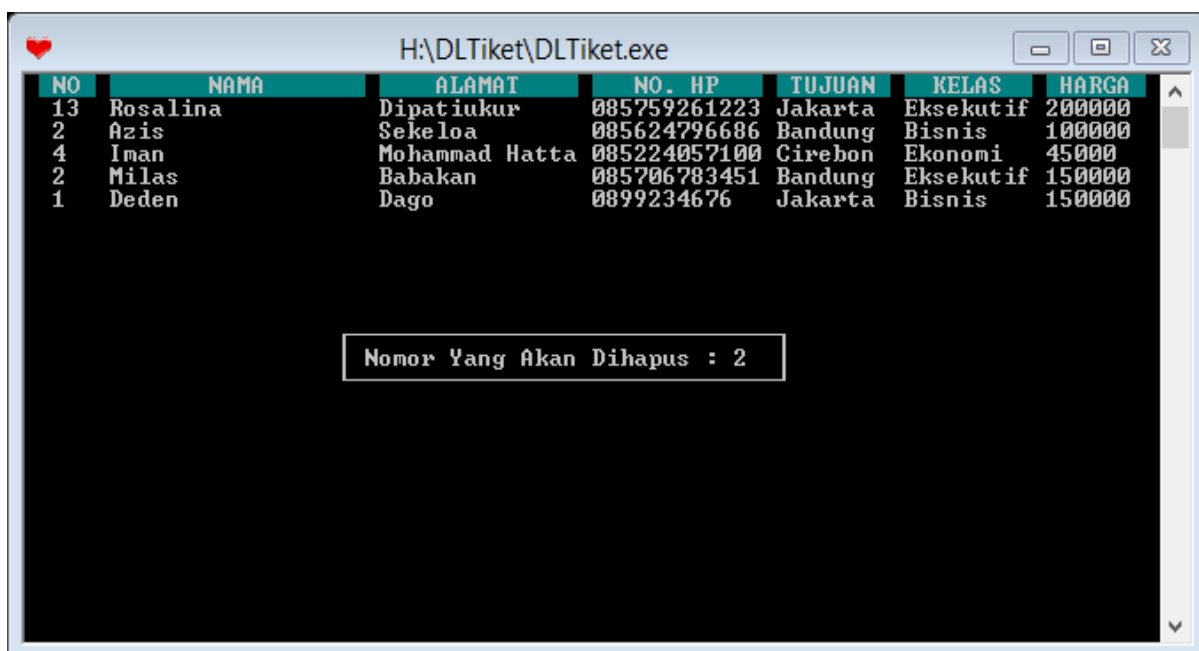


13. Pilih salah satu menu hapus, misalkan menu hapus tengah. Maka masukkan nomor 2 pada tombol pilihan. Maka akan menampilkan tampilan sebagai berikut.



The screenshot shows a window titled "H:\DLTiket\DLTiket.exe" with a red heart icon. It contains a table with 7 columns: NO, NAMA, ALAMAT, NO. HP, TUJUAN, KELAS, and HARGA. The data is as follows:

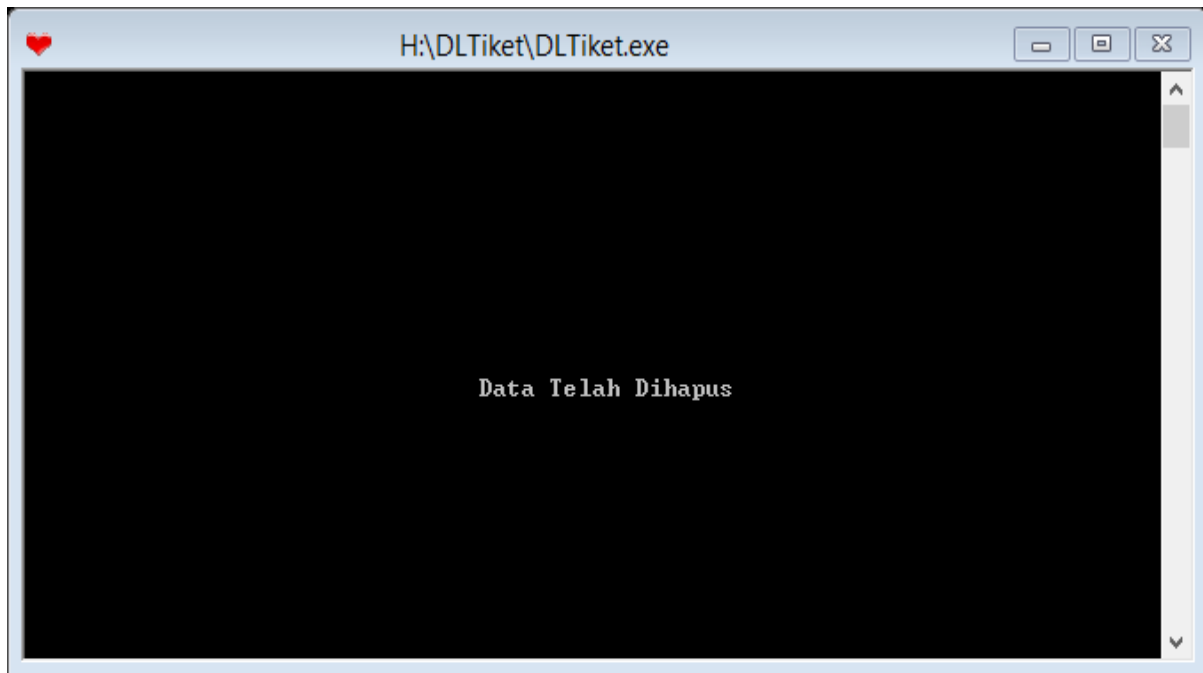
| NO | NAMA | ALAMAT | NO. HP | TUJUAN | KELAS | HARGA |
|----|----------|----------------|--------------|---------|-----------|--------|
| 13 | Rosalina | Dipatiukur | 085759261223 | Jakarta | Eksekutif | 200000 |
| 2 | Azis | Sekeloa | 085624796686 | Bandung | Bisnis | 100000 |
| 4 | Iman | Mohammad Hatta | 085224057100 | Cirebon | Ekonomi | 45000 |
| 2 | Milas | Babakan | 085706783451 | Bandung | Eksekutif | 150000 |
| 1 | Deden | Dago | 0899234676 | Jakarta | Bisnis | 150000 |



The screenshot shows the same window as above, but with a message box displayed in the center: "Nomor Yang Akan Dihapus : 2". The table data remains the same.

| NO | NAMA | ALAMAT | NO. HP | TUJUAN | KELAS | HARGA |
|----|----------|----------------|--------------|---------|-----------|--------|
| 13 | Rosalina | Dipatiukur | 085759261223 | Jakarta | Eksekutif | 200000 |
| 2 | Azis | Sekeloa | 085624796686 | Bandung | Bisnis | 100000 |
| 4 | Iman | Mohammad Hatta | 085224057100 | Cirebon | Ekonomi | 45000 |
| 2 | Milas | Babakan | 085706783451 | Bandung | Eksekutif | 150000 |
| 1 | Deden | Dago | 0899234676 | Jakarta | Bisnis | 150000 |

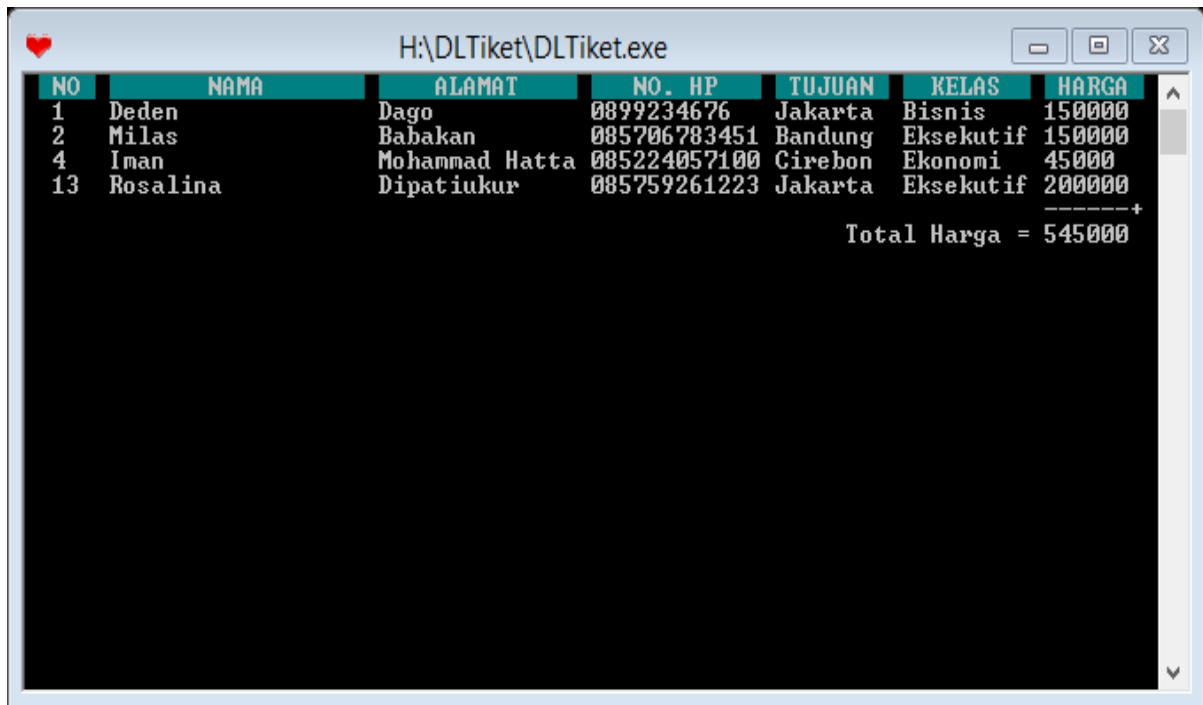
Nomor Yang Akan Dihapus : 2



14. Kemudian pilih kembali menu tampilkan data pada nomor pilihan nomor 4 untuk melihat data yang telah dihapus.

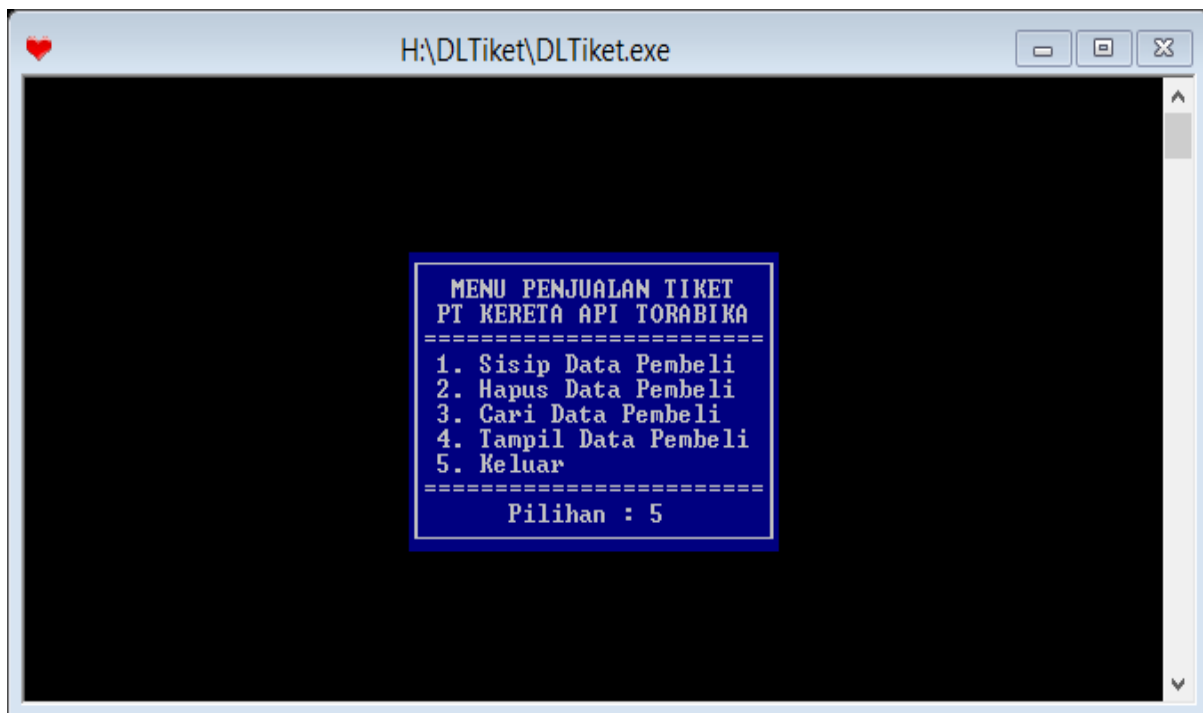
A screenshot of a Windows application window titled "H:\DLTiket\DLTiket.exe". The window displays a table with 7 columns: NO, NAMA, ALAMAT, NO. HP, TUJUAN, KELAS, and HARGA. The table contains 5 rows of data. The window has standard Windows controls (minimize, maximize, close) in the top right corner.

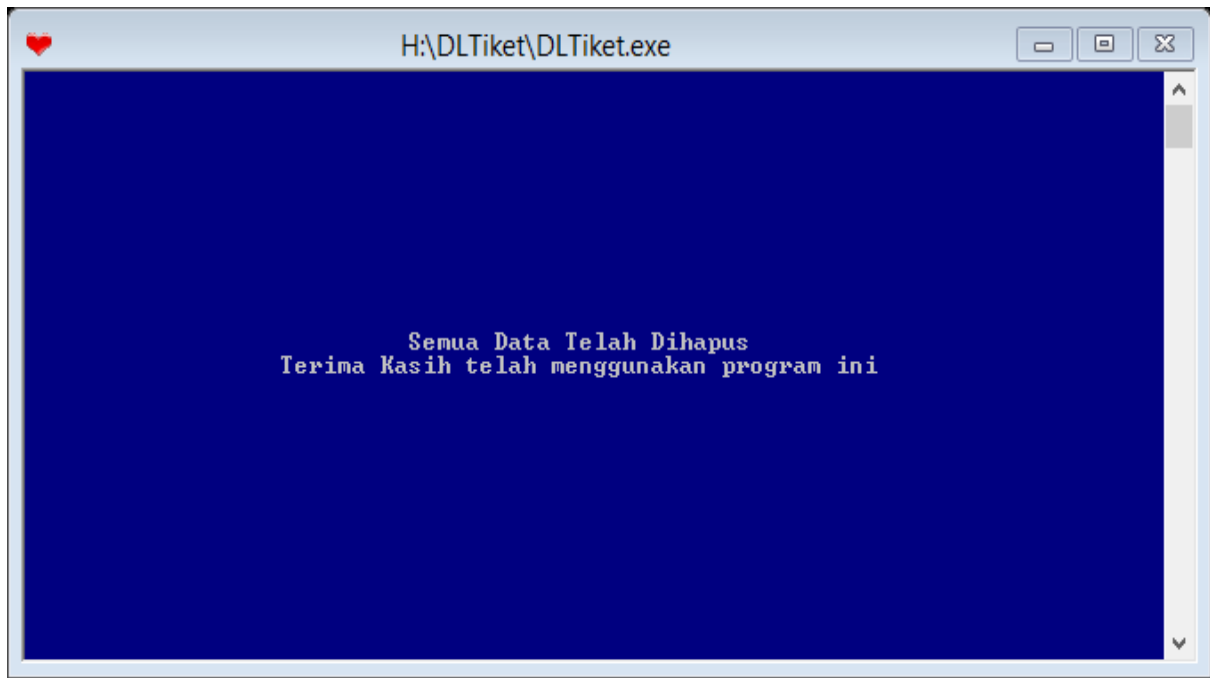
| NO | NAMA | ALAMAT | NO. HP | TUJUAN | KELAS | HARGA |
|----|----------|----------------|--------------|---------|-----------|--------|
| 1 | Deden | Dago | 0899234676 | Jakarta | Bisnis | 150000 |
| 2 | Milas | Babakan | 085706783451 | Bandung | Eksekutif | 150000 |
| 4 | Iman | Mohammad Hatta | 085224057100 | Cirebon | Ekonomi | 45000 |
| 13 | Rosalina | Dipatiukur | 085759261223 | Jakarta | Eksekutif | 200000 |



| NO | NAMA | ALAMAT | NO. HP | TUJUAN | KELAS | HARGA |
|----|----------|----------------|--------------|---------|-----------|----------------------|
| 1 | Deden | Dago | 0899234676 | Jakarta | Bisnis | 150000 |
| 2 | Milas | Babakan | 085706783451 | Bandung | Eksekutif | 150000 |
| 4 | Iman | Mohammad Hatta | 085224057100 | Cirebon | Ekonomi | 45000 |
| 13 | Rosalina | Dipatiukur | 085759261223 | Jakarta | Eksekutif | 200000 |
| | | | | | | -----+ |
| | | | | | | Total Harga = 545000 |

15. Kemudian inputkan nomor 5 untuk memilih menu keluar, sekaligus menghancurkan data.





BAB VI
DAFTAR KONTRIBUSI KELOMPOK

| NO | Nama Anggota | Tugas |
|----|----------------------|--|
| 1 | Azis | Membuat Algoritma |
| 2 | Deden Masmunandar | Membuat Algoritma |
| 3 | Milas Vernando Loby | Membuat Algoritma |
| 4 | Moh. Abdul Iman Syah | Membuat Program menggunakan Pascal dan membuat algoritma |
| 5 | Rosalina Fazriah | Membuat Algoritma |