

{----- Subrutin Menyisipkan Angka Baru Di Depan -----}

**Procedure** SisipAngkaDepan ( I/O Awal, Akhir : PointAngka, Input AngkaBaru : integer)  
{*I.S. : List Angka sudah terdefinisi*}  
{*F.S. : menghasilkan List Angka yang sudah disisipkan satu simpul di depan*}

**Kamus:**

Baru : PointAngka

**Algoritma :**

Alloc(Baru)  
Baru↑.Angka ← AngkaBaru  
**If** (Awal = Nil)  
  **Then**  
    Baru↑.Next ← Nil  
    Akhir ← Baru  
  **Else**  
    Baru↑.Next ← Awal  
  **EndIf**  
Awal ← Baru

**EndProcedure**

{----- Subrutin Menyisipkan Angka Baru Di Belakang -----}

**Procedure** SisipAngkaBelakang ( I/O Awal,Akhir : PointAngka, Input AngkaBaru : integer)  
{*I.S. : List Angka sudah terdefinisi*}  
{*F.S. : menghasilkan List Angka yang sudah disisipkan satu simpul di belakang*}

**Kamus:**

Baru : PointAngka

**Algoritma :**

Alloc(Baru)  
Baru↑.Angka ← AngkaBaru  
Baru↑.Next ← Nil  
**If** (Awal = Nil)  
  **Then**  
    Awal ← Baru  
  **Else**  
    Akhir↑.Next ← Baru  
  **EndIf**  
Akhir ← Baru

**EndProcedure**

{----- Subrutin Menyisipkan Angka Baru Di Tengah -----}

**Procedure** SisipAngkaTengah ( I/O Awal,Akhir : PointAngka, Input AngkaBaru : integer)  
{*I.S. : List Angka sudah terdefinisi*}  
{*F.S. : menghasilkan List Angka yang sudah disisipkan satu simpul di tengah*}

**Kamus:**

Baru,Bantu : PointAngka  
Ketemu : boolean  
AngkaSisip : integer

**Algoritma :**

**If** (Awal = Nil)  
  **Then**  
    Alloc(Baru)  
    Baru↑.Angka ← AngkaBaru  
    Baru↑.Next ← Nil  
    Awal ← Baru  
    Akhir ← Baru  
  **Else**  
    {*Angka baru akan disisipkan setelah angka tertentu*}  
    **Input** (AngkaSisip)  
    Bantu ← Awal  
    Ketemu ← false  
    **While** (**Not** Ketemu) **and** (Bantu ≠ Nil) **do**  
      **If** (Bantu↑.Angka = AngkaSisip)  
        **Then**  
          Ketemu ← true  
        **Else**  
          Bantu ← Bantu↑.Next  
      **EndIf**  
    **EndWhile**

```

If (Ketemu)
  Then
    Alloc(Baru)
    Baru↑.Angka ← AngkaBaru
    If (Bantu = Akhir)
      Then
        Baru↑.Next ← Nil
        Akhir ← Baru
      Else
        Baru↑.Next ← Bantu↑.Next
    EndIf
    Bantu↑.Next ← Baru
  Else
    Output('Angka yang dicari tidak ada pada list')
  EndIf
EndIf
EndProcedure

```

{----- Subrutin Menghapus Angka Di Depan -----}

```

Procedure HapusAngkaDepan ( I/O Awal, Akhir : PointAngka, Output Elemen : integer)
{I.S. : List Angka sudah terdefinisi}
{F.S. : menghasilkan List Angka yang sudah dihapus satu simpul/angka di depan}
Kamus:
  Phapus : PointAngka

```

```

Algoritma :
  Phapus ← Awal
  Elemen ← Phapus↑.Angka
  If (Awal = Akhir)
    Then
      Awal ← Nil
      Akhir ← Nil
    Else
      Awal ← Awal↑.Next
    EndIf
  Dealloc(Phapus)
EndProcedure

```

{----- Subrutin Menghapus Angka Di Belakang -----}

```

Procedure HapusAngkaBelakang ( I/O Awal, Akhir: PointAngka, Output Elemen : integer)
{I.S. : List Angka sudah terdefinisi}
{F.S. : menghasilkan List Angka yang sudah dihapus satu simpul/angka di belakang}
Kamus:
  Phapus : PointAngka

```

```

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
  Elemen ← Phapus↑.Angka
  Dealloc(Phapus)
EndProcedure

```

{----- Subrutin Menghapus Angka Di Tengah -----}

**Procedure** HapusAngkaBelakang (**I/O** Awal, Akhir: PointAngka, **Output** Elemen : **integer**)  
{**I.S.** : List Angka sudah terdefinisi}  
{**F.S.** : menghasilkan List Angka yang sudah dihapus satu angka di tengah}

**Kamus:**

Phapus, Bantu : PointAngka  
Ketemu : **boolean**  
AngkaHapus : **integer**

**Algoritma :**

**Input**(AngkaHapus)

Ketemu  $\leftarrow$  false

Phapus  $\leftarrow$  Awal

**While** (**Not** Ketemu) **and** (Phapus  $\neq$  Nil) **do**

**If** (Phapus $\uparrow$ .Angka = AngkaHapus)

**Then**

            Ketemu  $\leftarrow$  true

**Else**

            Phapus  $\leftarrow$  Phapus $\uparrow$ .Next

**EndIf**

**EndWhile**

**If** (Ketemu)

**Then**

        Elemen  $\leftarrow$  Phapus $\uparrow$ .Angka

**If** (Awal = Akhir) {jika list hanya satu simpul}

**Then**

                Awal  $\leftarrow$  Nil

                Akhir  $\leftarrow$  Nil

**Else**

                {jika list lebih dari satu simpul}

**If** (Phapus = Awal) {jika ditemukan pada simpul pertama, berarti menghapus di depan}

**Then**

                        Awal  $\leftarrow$  Awal $\uparrow$ .Next

**Else**

                        Bantu  $\leftarrow$  Awal

**While** (Bantu $\uparrow$ .Next  $\neq$  Phapus) **do**

                            Bantu  $\leftarrow$  Bantu $\uparrow$ .Next

**EndWhile**

**If** (Phapus = Akhir) {jika ditemukan pada simpul terakhir, berarti menghapus di belakang}

**Then**

                                Akhir  $\leftarrow$  Bantu

                                Akhir $\uparrow$ .Next  $\leftarrow$  Nil

**Else**

                                Bantu $\uparrow$ .Next  $\leftarrow$  Phapus $\uparrow$ .Next {menghapus di tengah}

**EndIf**

**EndIf**

    Dealloc(Phapus)

**Else**

**Output**('Angka yang akan dihapus tidak ada pada list')

**EndIf**

**EndProcedure**

{----- Subrutin Menampilkan Isi List -----}

**Procedure** TampilIsiList(**Input** Awal : PointAngka)

{**I.S.** : List Angka sudah terdefinisi}

{**F.S.** : menampilkan isi list Angka}

**Kamus:**

Bantu : PointAngka  
i : **integer**

**Algoritma:**

**Output**('Menampilkan Isi List Angka')

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

Bantu  $\leftarrow$  Awal

i  $\leftarrow$  1

**While** (Bantu  $\neq$  Nil) **do**

**Output**('Angka ke-`,i,` : `, Bantu $\uparrow$ .Angka')

```

    Bantu ← Bantu↑.Next
    i ← i + 1
  EndWhile
EndProcedure

```

{----- Subrutin Menghitung Rata-rata Angka -----}

```

Function RataAngka (Input TotalAngka : real, Input BanyakData : integer) → real
  {I.S. : Angka sudah terdefinisi}
  {F.S. : menghasilkan fungsi rata-rata Angka}
Kamus :

```

**Algoritma:**

```

RataAngka ← TotalAngka/BanyakData

```

**EndFunction**

{----- Subrutin Menampilkan Rata-rata Angka Di List -----}

```

Procedure TampilRataAngka(Input Awal : PointAngka)
  {I.S. : List Angka sudah terdefinisi}
  {F.S. : menampilkan rata-rata angka di list}
Kamus:

```

```

  Bantu      : PointAngka
  TotalAngka : real
  BanyakData : integer

```

```

Function RataAngka (Input TotalAngka:real, Input BanyakData:integer) → real

```

**Algoritma:**

```

  Bantu ← Awal
  BanyakData ← 1
  TotalAngka ← 0

```

```

While (Bantu ≠ Nil) do
  TotalAngka ← TotalAngka + Bantu↑.Angka
  Bantu ← Bantu↑.Next
  BanyakData ← BanyakData + 1
EndWhile

```

```

  BanyakData ← BanyakData - 1
  Output ('Rata-rata Angka : ',RataAngka(TotalAngka,BanyakData))

```

**EndProcedure**

{----- Subrutin Mengurutkan Angka Secara Menaik Di List -----}

```

Procedure UrutAngkaAsc( I/O Awal, Akhir : PointAngka)
  {I.S. : List Angka sudah terdefinisi}
  {F.S. : menghasilkan List Angka yang sudah terurut secara ascending menggunakan Metode
  Minimum Sort}
Kamus:

```

```

  i,j,min : PointAngka
  Temp    : integer

```

**Algoritma :**

```

  i ← Awal
  While (i ≠ Akhir) do
    min ← i
    j ← i↑.Next
    While (j ≠ Nil) do
      If (j↑.Angka < min↑.Angka)
        Then
          min ← j
        EndIf
      j ← j↑.Next
    EndWhile

```

```

{tukar data}
Temp ← min↑.Angka
min↑.Next ← i↑.Next
i↑.Next ← Temp

```

```

i ← i↑.Next

```

```

EndWhile

```

```

EndProcedure

```

```

{----- Subrutin Mencari Satu Angka Tertentu Di List -----}

```

```

Procedure CariAngka ( Input Awal : PointAngka)

```

```

{I.S. : List Angka sudah terdefinisi}

```

```

{F.S. : menampilkan berapa banyak angka yang dicari ada pada list Angka (mencari data
yang tidak unik)}

```

```

Kamus:

```

```

Bantu          : PointAngka
CariAngka,i    : integer {i:banyaknya angka di list}
Ketemu         : boolean

```

```

Algoritma :

```

```

Input(CariAngka)

```

```

Bantu ← Awal

```

```

Ketemu ← false

```

```

While (Not Ketemu) and (Bantu ≠ Nil) do

```

```

    If (Bantu↑.Angka = CariAngka)

```

```

        Then

```

```

            Ketemu ← true

```

```

        Else

```

```

            Bantu ← Bantu↑.Next

```

```

    EndIf

```

```

EndWhile

```

```

If (Ketemu)

```

```

    Then

```

```

        i ← 1 {sudah ditemukan satu data}

```

```

        Bantu ← Bantu↑.Next {pointer Bantu menunjuk simpul sebelahnya}

```

```

        While (Bantu ≠ Nil) do

```

```

            If (Bantu↑.Angka = CariAngka)

```

```

                Then

```

```

                    i ← i + 1

```

```

            EndIf

```

```

            Bantu ← Bantu↑.Next

```

```

        EndWhile

```

```

        Output('Angka ',CariAngka,' ada ',i,' buah di list Angka')

```

```

    Else

```

```

        Output('Angka ',CariAngka,' tidak ada pada List Angka')

```

```

    EndIf

```

```

EndProcedure

```

```

{----- Subrutin Menghancurkan List -----}

```

```

Procedure Penghancuran( I/O Awal, Akhir : PointAngka)

```

```

{I.S. : List Angka sudah terdefinisi}

```

```

{F.S. : menghasilkan List Angka yang sudah dihancurkan dan menjadi kosong}

```

```

Kamus:

```

```

Phapus : PointAngka

```

```

Algoritma :

```

```

Phapus ← Awal

```

```

While (Phapus ≠ Nil) do

```

```

    Awal ← Awal↑.Next

```

```

    Dealloc(Phapus)

```

```

    Phapus ← Phapus↑.Next

```

```

EndWhile

```

```

Akhir ← Nil

```

```

EndProcedure

```

{----- Subrutin Menu Pilihan -----}

**Procedure** MenuUtama( **Output** Menu : integer)  
{I.S. : user memasukkan/memilih salah satu menu pilihan}  
{F.S. : menghasilkan nomor menu yang dipilih}

**Kamus:**

**Algoritma :**

**Output**('MENU PILIHAN')  
**Output**('-----')  
**Output**('1. Tambah Data Angka')  
**Output**('2. Hapus Data Angka')  
**Output**('3. Tampil Data Angka')  
**Output**('4. Tampil Rata-Rata Angka')  
**Output**('5. Cari Data Angka Tertentu')  
**Output**('0. Keluar')  
**Input**(Menu)  
{Beri Validasi untuk menu pilihan}

**EndProcedure**

{----- Subrutin Menu Tambah Data -----}

**Procedure** MenuTambahAngka( **Output** MenuTambah : integer)  
{I.S. : user memasukkan/memilih salah satu menu tambah data}  
{F.S. : menghasilkan nomor menu yang dipilih}

**Kamus:**

**Algoritma :**

**Output**('MENU TAMBAH DATA')  
**Output**('-----')  
**Output**('1. Tambah Data Angka Di Depan')  
**Output**('2. Tambah Data Angka Di Belakang')  
**Output**('3. Tambah Data Angka Di Tengah')  
**Output**('4. Keluar')  
**Input**(MenuTambah)  
{Beri validasi untuk menu tambah data}

**EndProcedure**

{----- Subrutin Menu Hapus Data -----}

**Procedure** MenuHapusAngka( **Output** MenuHapus : integer)  
{I.S. : user memasukkan/memilih salah satu menu hapus data}  
{F.S. : menghasilkan nomor menu yang dipilih}

**Kamus:**

**Algoritma :**

**Output**('MENU HAPUS DATA')  
**Output**('-----')  
**Output**('1. Hapus Data Angka Di Depan')  
**Output**('2. Hapus Data Angka Di Belakang')  
**Output**('3. Hapus Data Angka Di Tengah')  
**Output**('4. Keluar')  
**Input**(Menu Hapus)  
{Beri validasi untuk menu tambah hapus}

**EndProcedure**

{----- Algoritma Utama -----}

**ListAngka**

{I.S. : User memilih salah satu menu}  
{F.S. : Menampilkan hasil list sesuai menu yang dipilih}

**Kamus:**

**Type**

PointAngka = ↑SimpulAngka

SimpulAngka = **record**

Angka : **integer**,

Next : PointAngka

**EndRecord**

Awal, Akhir : PointAngka

AngkaBaru, Elemen, Menu, MenuTambah, MenuHapus : **integer**

*{Prototype Subrutin yang digunakan}*

**Procedure** SisipAngkaDepan( I/O Awal, Akhir : PointAngka, Input AngkaBaru : integer)  
**Procedure** SisipAngkaBelakang(I/O Awal, Akhir: PointAngka, Input AngkaBaru: integer)  
**Procedure** SisipAngkaTengah( I/O Awal, Akhir : PointAngka, Input AngkaBaru: integer)  
**Procedure** HapusAngkaDepan( I/O Awal, Akhir : PointAngka, Output Elemen : integer)  
**Procedure** HapusAngkaBelakang( I/O Awal, Akhir : PointAngka, Output Elemen: integer)  
**Procedure** HapusAngkaTengah( I/O Awal, Akhir : PointAngka, Output Elemen : integer)  
**Procedure** TampilIsiList(Input Awal : PointAngka)  
**Procedure** TampilRataAngka( Input Awal : PointAngka)  
**Procedure** UrutAngkaAsc(I/O Awal, Akhir : PointAngka)  
**Procedure** CariAngka(Input Awal : PointAngka)  
**Procedure** Penghancuran( I/O Awal, Akhir : PointAngka)  
**Procedure** MenuUtama(Output Menu : integer)  
**Procedure** MenuTambahAngka(Output MenuTambah : integer)  
**Procedure** MenuHapusAngka (Output MenuHapus : integer)

**Algoritma:**

*{Penciptaan List}*

Awal ← Nil  
Akhir ← Nil

**Repeat**

MenuUtama (Menu)

**Depend On** (Menu)

1 : **Repeat**

MenuTambahAngka (MenuTambah)

**Input** (AngkaBaru)

**Depend On** (MenuTambah)

1 : SisipAngkaDepan (Awal, Akhir, AngkaBaru)

TampilIsiList (Awal)

2 : SisipAngkaBelakang (Awal, Akhir, AngkaBaru)

TampilIsiList (Awal)

3 : SisipAngkaTengah (Awal, Akhir, AngkaBaru)

TampilIsiList (Awal)

**EndDepend**

**Until** (MenuTambah=0)

2 : **Repeat**

MenuHapusAngka (MenuHapus)

**If** (Awal = Nil) *{jika list masih kosong, maka tidak bisa menghapus data}*

**Then**

**Output** ('List Masih Kosong')

**Else**

**Depend On** (MenuTambah)

1 : HapusAngkaDepan (Awal, Akhir, Elemen)

TampilIsiList (Awal)

2 : HapusAngkaDepan (Awal, Akhir, Elemen)

TampilIsiList (Awal)

3 : HapusAngkaDepan (Awal, Akhir, Elemen)

TampilIsiList (Awal)

**EndDepend**

**EndIf**

**Until** (MenuHapus=0)

3 : **If** (Awal = Nil)

**Then**

**Output** ('Tidak Ada Data')

**Else**

UrutAngkaAsc (Awal, Akhir)

TampilIsiList (Awal)

**EndIf**

4 : **If** (Awal = Nil)

**Then**

**Output** ('Tidak Ada Data')

**Else**

TampilRataAngka (Awal)

**EndIf**

5 : **If** (Awal = Nil)

**Then**

**Output** ('Tidak Ada Data')

**Else**

CariAngka (Awal)

**EndIf**

**EndDepend**

**Until** (Menu = 0)

*{Menghancurkan List (setelah selesai dengan list, maka list harus dihancurkan)}*

Penghancuran(Awal,Akhir)

If (Awal = Nil)

Then

Output ('List sudah Kosong')

EndIf