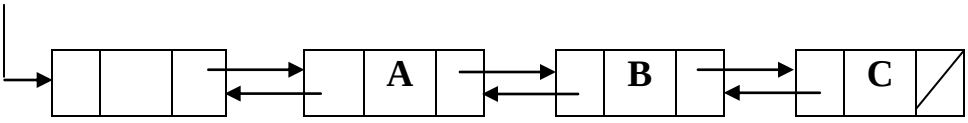
2 SENARAI/LIST BERANTAI GANDA

* **Senarai berantai yang sudah dipelajari hanya mempunyai sebuah pointer pada setiap simpul. Kerugiannya hanya bisa dibaca dalam satu arah saja. Jika ingin membacanya dari arah sebaliknya, tidak bisa melakukannya dengan cara menelusuri pointer, tetapi harus dengan cara rekursif atau terlebih dahulu mengubah arah pointernya.**
* **Senarai berantai ganda (*doubly linked-list*) atau senarai dua arah (*two-way list*), setiap simpulnya mempunyai 2 buah pointer, dengan pointer pertama menunjuk ke simpul sebelumnya dan pointer kedua menunjuk simpul sesudahnya.**
* **Jika senarainya mempunyai simpul kepala, dinamakan senarai berantai ganda berkepala (*headed doubly linked-list*).**

Kepala



*** Deklarasi simpulnya:**

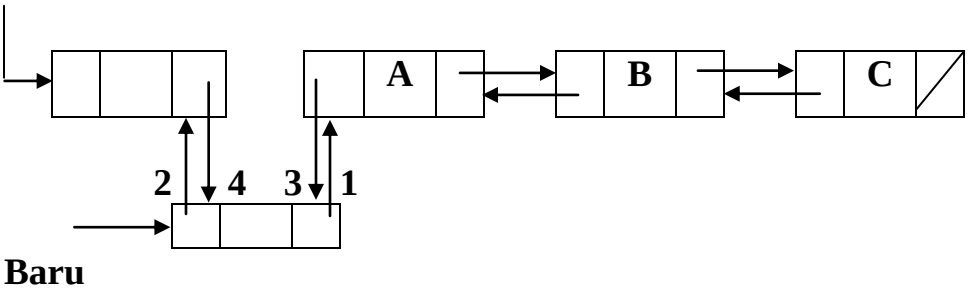
```
Type  Simpul  =  ^Data;
      Data =  record
                  Info          : char;
                  Kiri, Kanan : Simpul
              End;
Var   Kepala : Simpul;
```

(pointer Kiri adalah pointer untuk menunjuk ke simpul sebelumnya; pointer Kanan adalah pointer untuk menunjuk ke simpul sesudahnya)

*** Untuk inisialisasi kita menginginkan pointer Kiri dan Kanan pada simpul kepala bernilai nil.**

```
Procedure INISIALISASI (var Kepala : Simpul);
Begin
    New(Kepala);
    With Kepala^ do
        begin
            Kiri := nil;
            Kanan := nil
        end
End;
```

*** Ilustrasi penambahan simpul sesudah simpul kepala**




```
Baru^.Kanan := Kepala^.Kanan;
Baru^.Kiri := Kepala;
Kepala^.Kanan^.Kiri := Baru;
Kepala^.Kanan := Baru;
```

*** Untuk penambahan simpul baru di tengah senarai, memerlukan bantuan pointer lain, misalnya Bantu. Simpul Baru akan disisipkan setelah simpul yang ditunjuk oleh pointer Bantu. Dari ilustrasi diatas, gantilah pointer Kepala menjadi Bantu.**

```
Baru^.Kanan  :=  Bantu^.Kanan;
Baru^.Kiri  :=  Bantu;
Bantu^.Kanan^.Kiri  :=  Baru;
Bantu^.Kanan  :=  Baru;
```

*** Untuk penambahan simpul baru yang ditempatkan di akhir senarai maka pointer Bantu kita tempatkan di simpul akhir, sehingga penambahannya adalah:**

```
Bantu^.Kanan  :=  Baru;
Baru^.Kiri       :=  Bantu;
```

*** Prosedur untuk penambahan simpul baru pada senarai berantai ganda adalah sbb:**

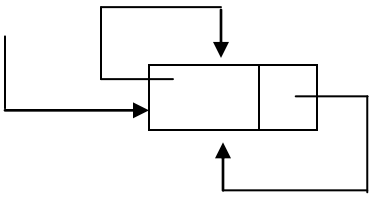
```
Procedure TAMBAH_SIMPUL (var Kepala  :  Simpul;
                                                Elemen : char);
Var  Baru, Bantu  :  Simpul;
Begin
      New(Baru);
      With  Baru^  do
          Begin
              Info  :=  Elemen;
              Kiri := nil;
```

```
            Kanan := nil
        End;
    Bantu  :=  Kepala;
    While Bantu^.Kanan^.Info  <  Elemen  do
        Bantu  :=  Bantu^.Kanan;
    If Bantu^.Kanan <> nil then
        Begin
            Baru^.Kanan := Bantu^.Kanan;
            Bantu^.Kanan^.Kiri := Baru
        End;
    Bantu^.Kanan  := Baru;
    Baru^.Kiri  :=  Bantu;
End;
```

* **Kelemahannya yaitu, apabila kita ingin menambah simpul baru dan ingin ditempatkan di akhir senarai, maka tetap harus mengunjungi setiap simpul dimulai dari simpul pertama sampai simpul terakhir untuk menempatkan pointer bantuan.**

* **Pada senarai berantai ganda, pointer Kiri dari simpul pertama dan pointer Kanan dari simpul terakhir masih bernilai nil. Bila pointer Kiri kita arahkan ke simpul terakhir dan pointer Kanan dari simpul terakhir kita arahkan ke simpul kepala, senarai yang demikian dinamakan <u>senarai berantai ganda berputar dan berkepala.</u>**
* **Untuk inisialisasi, kita menginginkan agar pointer Kiri dan Kanan dari simpul Kepala tidak bernilai nil, sehingga gambar simpul Kepala pada saat inisialisasi sebagai berikut:**

Kepala



Prosedur inisialisasi simpul kepala tersebut sebagai berikut:

```
Procedure INISIALISASI_SBGBB (var Kepala  :  Simpul);
Begin
      New(Kepala);
      With  Kepala^  do
          begin
              Kiri := Kepala;
              Kanan := Kepala
          end
End;
```

* **Penambahan simpul baru di tengah senarai berantai adalah sama seperti di atas, dengan mengganti pointer Kepala dengan Bantu.**

* **Penambahan simpul baru yang ditempatkan sesudah simpul Kepala adalah sama dengan di atas, dengan memasang pointer Bantu pada simpul Kepala.**

*** Penambahan simpul baru di akhir senarai berantai sebagai berikut:**

- **Arahkan pointer Kiri dari simpul Baru ke simpul terakhir.**
- **Pointer Kanan dari simpul Baru diarahkan ke simpul kepala.**
- **Pointer Kanan dari simpul terakhir diarahkan ke simpul Baru.**
- **Arahkan pointer Kiri simpul Kepala ke simpul Baru.**

Langkah-langkah di atas singkatnya sebagai berikut:

```
Baru^.Kiri  :=  Kepala^.Kiri;
Baru^.Kanan  :=  Kepala;
Kepala^.Kiri^.Kanan  :=  Baru;
Kepala^.Kiri  :=  Baru;
```

*** Prosedur untuk penambahan simpul pada senarai berantai ganda berputar berkepala sebagai berikut:**

```
Procedure TAMBAH_SIMPUL (var Kepala  :  Simpul;
                                        Elemen : char);
Var  Baru, Bantu  :  Simpul;
Begin
    New(Baru);
    Baru^.Info := Elemen;
    If Elemen > Kepala^.Kiri^.Info  then
        Begin
            Baru^.Kiri  :=  Kepala^.Kiri;
            Baru^.Kanan := Kepala;
            Kepala^.Kiri^.Kanan := Baru;
            Kepala^.Kiri  :=  Baru
        End;
    Else
    begin
    Bantu  :=  Kepala;
    While Bantu^.Kanan^.Info  <  Elemen  do
            Bantu  :=  Bantu^.Kanan;
            Baru^.Kanan := Bantu^.Kanan;
```

```
            Bantu^.Kanan^.Kiri := Baru;
            Bantu^.Kanan := Baru;
            Baru^.Kiri  :=  Bantu
        end
    End;
```

* **Penghapusan simpul sebagai berikut:**
  - **Pointer Bantu menunjukkan posisi simpul yang akan dihapus**
  - **Mulai dengan mengarahkan pointer Kanan dari simpul sebelum simpul Bantu ke simpul sesudah simpul Bantu.**
  - **Arahkan simpul Kiri sesudah simpul Bantu ke simpul sebelum simpul Bantu**
  - **Berikutnya simpul yang ditunjuk oleh pointer Bantu di dispose.**

* **Prosedur untuk menghapus simpul pada senarai berantai ganda berputar berkepala sebagai berikut:**

```
Procedure HAPUS_SIMPUL (var Kepala  :  Simpul;
                                       Elemen : char);
Var   Bantu  :  Simpul;
Begin
      If  Kepala^.Kanan  =  Kepala  then
           Writeln('Senarai kosong')
      else
          Begin
             Bantu  :=  Kepala;
             repeat
```

```
            Bantu := Bantu^.Kanan;
        Until (Bantu^.Info = Elemen) or (Bantu = Kepala);
        If  Bantu^.Info  =  Elemen  then
            begin
                Bantu^.Kiri^.Kanan  :=  Bantu^.Kanan;
                Bantu^.Kanan^.Kiri := Bantu^.Kiri;
                 Dispose(Bantu);
                Bantu  :=  Kepala
            End
        Else
            Writeln('Karakter di atas tidak ada')
    end
End;
```

*** Contoh program penggunaan senarai berantai ganda berputar adalah, untuk menjumlah 2 buah bilangan bulat positif yang mempunyai digit cukup panjang. Dalam Turbo Pascal versi 6.0 bilangan integer terbesar yang bisa ditangani adalah -2147483648 sampai 2147483647. Bilangan ini dikenal dengan bilangan bulat panjang (*long integer*).**

```
{ pemakaian senarai berantai ganda berputar untuk menjumlahkan }
{ dua bilangan bulat positif }

program jumlah_dua_integer;
uses WINcrt;
const max = 80;

type str80 = string [max];
     simpul = ^data;
     data = record
                 info : char;
                 kiri, kanan : simpul
            end;
```

```
var bilangan1,
    bilangan2,
    bilangan3 : simpul;
    angka1,
    angka2  : str80;
    I : integer;
    lagi : char ;

function cek_bilangan ( bil : str80) : boolean ;
 var I : integer;
     angka : set of char;
     valid : boolean;

 begin
      angka := ['0'..'9'];
      valid := true;
      for I := 1 to length (bil) do
          if not (bil[I] in angka ) then
          begin
               valid := false;
               I := length(bil)
          end;
      cek_bilangan := valid ;
 end;

 procedure awalan ( var baru : simpul );

 begin
      new(baru);
      with baru^ do
          begin
               info := chr(32);
               kiri := baru;
               kanan := kiri
          end;
 end;

 procedure buat_list( var list : simpul;
                          bil : str80);
 var I,cch_kar,j,kode : integer;
     baru : simpul;

 begin
      for I := 1 to length (bil) do
          begin
               awalan(baru);
               val(bil[I],j,kode);
               baru^.info := chr(j);
               baru^.kiri := list^.kiri;
               baru^.kanan := list;
               list^.kiri^.kanan := baru;
               list^.kiri := baru;
          end;
          list^.info := chr(length(bil))
 end;
```

```
procedure baca (kepala : simpul);
var bantu : simpul; kode : integer;

begin
     bantu := kepala^.kanan;

     repeat
           kode := ord(bantu^.info);
           if kode = 32 then
              write(' ')
           else
               write(kode);
           bantu := bantu^.kanan
     until bantu = kepala;
     writeln
end;

procedure cek_operand(var bil1, bil2 : simpul );
var jml1,
jml2 : integer;

procedure tambah_nol ( var T : simpul; c : integer);
var baru : simpul;
    I : integer;

    begin
         for I := 1 to c do
             begin
                  awalan (baru);
                  baru^.kiri := T;
                  baru^.kanan := T^.kanan;
                  T^.kanan^.kiri := baru;
                  T^.kanan := baru;
                  T^.info := chr(ord(T^.info)+1)
             end
    end;

    begin
         jml1 := ord(bil1^.info);
         jml2 := ord(bil2^.info);
         if jml1 <> jml2 then

            if jml1 > jml2 then

               tambah_nol(bil2,jml1-jml2)
            else
                tambah_nol(bil1,jml2-jml1)
    end;

    procedure hasil(var bil1, bil2, bil3 : simpul);
    var sisa,
        jumlah,
        dgt,
        dgt1 : integer;
        bantu1,
        bantu2,
        baru : simpul;
```

```
procedure oper;

     begin
          baru^.kanan := bil3^.kanan;
          baru^.kiri := bil3;
          bil3^.kanan^.kiri := baru;
          bil3^.kanan := baru
     end;

     begin
          bantu1 := bil1^.kiri;
          bantu2 := bil2^.kiri;
          sisa := 0;

          repeat
                dgt := ord(bantu1^.info);
                dgt1 := ord(bantu2^.info);
                if dgt1 = 32 then dgt1 := 0;
                if dgt = 32 then dgt := 0;

                jumlah := dgt + dgt1 + sisa;

                if jumlah >= 10 then
                   begin
                     jumlah := jumlah - 10;
                     sisa := 1
                   end

                else
                    sisa := 0;
                    new(baru);
                    baru^.info := chr(jumlah);
                    oper;
                    bantu1 := bantu1^.kiri;
                    bantu2 := bantu2^.kiri;
          until bantu1 = bil1;

          if sisa = 1 then
             begin
                  awalan(baru);
                  baru^.info := chr(sisa);
                  oper;
                  awalan(baru);awalan(bantu1);
                  baru^.kanan := bil1^.kanan;
                  baru^.kiri := bil1;
                  bil1^.kanan^.kiri := baru;
                  bil1^.kanan := baru;
                  bantu1^.kanan := bil2^.kanan;
                  bantu1^.kiri := bil2;
                  bil2^.kanan^.kiri := bantu1;
                  bil2^.kanan := bantu1;
                  bil1^.info := chr(ord(bil1^.info)+1);
             end
          end;
```

```
{*** PROGRAM UTAMA ***}

          begin
               repeat
                 clrscr;
                 write('contoh penggunaan senarai berantai');
                 writeln('ganda berputar');
                 write('untuk menjumlah dua bilangan bulat');
                 writeln('panjang ( long integer)');
                 write('(hanya bisa untuk dua bilangan');
                 writeln('bulat yang positif)');
                 write('-----------------------------------');
                 writeln('--------------------------');
                 writeln;
                 write('bilangan pertama :');readln(  angka1);
                 write('bilangan kedua : ');readln( angka2);
                 writeln;

                 if cek_bilangan(angka1) and
                    cek_bilangan(angka2)then

                    begin
                         awalan(bilangan1); awalan(bilangan2);
                         awalan(bilangan3);
                         buat_list(bilangan1,angka1);
                         buat_list(bilangan2,angka2);

                         cek_operand(bilangan1,bilangan2);

                         hasil(bilangan1,bilangan2,bilangan3);

                         writeln; writeln('hasil perhitungan :');
                         writeln;
                         baca(bilangan1);baca(bilangan2);
                            for I := 1 to ord(bilangan1^.info) do
                                write('-');

                                write('+');writeln;baca(bilangan3)
                            end
                            else
                              writeln('ada karakter tidak sah');
                       writeln; write('akan mencoba lagi?(Y/T):');
                       readln(lagi)
                 until not (lagi in['Y','y'])
end.
```