MODUL PRAKTIKUM

STRUKTUR DATA

Erna Kumalasari Nurnawati

ISBN: 978-602-7619-42-5

MODUL PRAKTIKUM

STRUKTUR DATA

ISBN: 978-602-7619-42-5

Hak cipta 2018 pada penulis dilarang keras mengutip, menjiplak, memfoto copy baik sebagian maupun keseluruhan isi buku ini tanpa mendapat izin tertulis dari pengarang dan penerbit

Penulis : Erna Kumalasari Nurnawati

Desain Cover : Erna Kumalasari Nurnawati

Diterbitkan oleh : AKPRIND PRESS

HAK CIPTA DILINDUNGI OLEH UNDANG -UNDANG

KATA PENGANTAR

Puji Syukur ke hadirat Allah Tuhan Yang Maha Esa yang melimpahkan karunianya sehingga perbaikan modul Struktur Data dapat direvisi dengan cukup signifikan.

Modul ini disusun sebagai pegangan pada praktikum Strukur Data dan diharapkan dapat membantu mahasiswa melaksanakan kegiatan praktikum dengan lebih baik. Mengingat kelengkapan modul ini, maka mahasiswa juga dapat menggunakan modul sebagai diktat kuliah Struktur Data di kelas. Dengan menggunakan modul ini diharapkan mahasiswa dapat mengembangkan logika berpikir dan kemampuan melakukan coding dalam bahasa Pascal yang bersifat lanjut.

Modul ini dilengkapi dengan banyak banyak latihan yang harus dikerjakan mahasiswa di labroratorium. Diharapkan dengan banyaknya latihan makin meningkatkan kemampuan pemrograman mahasiswa dan mengasah logika penrograman.

Materi pada modul ini meliputi pengingat kembali materi larik dan record. Record multi, pencarian (searching), pengelompkan(filtering), pengurutan data (sorting), stack (tumpukan), queue (antrian) dan ragam kasus dengan struktur gabungan.

Kami mengucapkan terimakasih atas semua pihak yang berkontribusi atas terselenggaranya revisi modul ini, terutama asisten di Lab Pemrograman Dasar. Jika ada pertanyaan atau perbaikan bias melalui email ernakumala@akprind.ac.id.

Yogyakarta, Februari 2019

Penyusun.

Erna Kumalasari Nurnawati ST., MT.

KARTU PRAKTIKUM STRUKTUR DATA

Nim : _____

Nama : _____

Minggu ke	Tanggal	Materi	Paraf Asisten dan Cap
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			
INHAL 1			
INHAL 2			
RESPONSI			

TATA TERTIB PRAKTIKUM

1. Praktikum hadir sebelum praktikum dimulai. Keterlambatan ditoleransi 15 menit. Jika keterlambatan lebih dari 15 menit maka nilai yang diperoleh pada praktikum pada sesi tersebut maksimal 75% dari nilai yang diperoleh
2. Praktikan berpakaian sopan, tidak diperkenankan memamakai kaos oblong, topi serta tidak diijinkan makan di ruang laboratorium (minum boleh tetapi membawa sendiri).
3. **Mengisi daftar hadir dan meminta paraf dan Cap asisten pada kartu praktikum** anda
4. Praktikan tidak diijinkan menggunakan flasdisk
5. Praktikan WAJIB memiliki dan menggunakan modul terbaru. Modul bisa diperoleh di laboratorium dengan mengganti biaya cetak (Kartu praktikum terdapat di dalam modul). **Modul WAJIB dibawa saat praktikum.**
6. Praktikan menjaga kebersihan ruangan dan merapikan komputer, kursi dan peralatan lain setelah praktikum selesai. Matikan komputer dengan prosedur yang benar, serta mematikan stabilizer
7. Demi keamanan, praktikan hendaknya memakai sandal yang disediakan lab. Kembalikan sandal ditempatnya setelah selesai praktikum. Sandal tidak diijinkan digunakan di luar ruangan.
8. Penilaian praktikum dilakukan selama 8 minggu, sejak minggu ke 2 hingga minggu ke 9. Nilai max 90 (karena anda dibimbing asisten dan tidak ada tugas yang 100% anda kerjakan sendiri) dengan bobot masing-masing 10%. Nilai responsi max 100 berbobot 20%. Nilai anda akan diakumulasi dan menjadi nilai tugas ke 4 untuk matakuliah masing-masing. Tidak ada KETIDAK LULUSAN dalam praktikum, tetapi nilai praktikum mempunyai bobot 20% dari nilai matakuliah
9. Kehadiran praktikum harus mencapai 75% (mutual dengan kehadiran kuliah), sehingga apabila kehadiran kurang, maka anda tidak bisa mengikuti ujian akhir. Jika anda tidak hadir, anda diharapkan mengikuti inhal. Mutual dengan perkuliahan, anda diijinkan mengikuti inhal 2x, dan nilai dan kehadiran akan disinkronkan dengan nilai yang anda peroleh. Jika ketidakhadiran lebih dari 2 kali, maka yang bisa diinhal hanya 2x saja, sisanya dianggap tidak hadir dan nilai nol. Nilai inhal max 75.
10. Demikian tata tertib praktikum harap mahasiswa menyesuaikan

DAFTAR ISI

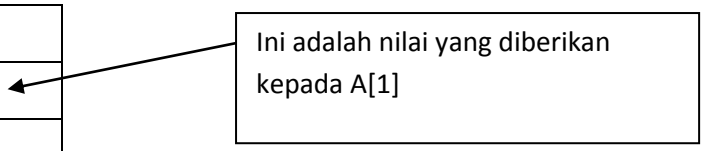
MODUL 1
LARIK DIMENSI SATU

Kompetensi: Mahasiswa mampu mengenali larik dimensi satu, mampu mendeklarasikan dan menggunakan larik dimensi satu dalam menyelesaikan masalah dalam pemrograman

1. Pendahuluan

Suatu array adalah sebuah struktur data yang terdiri atas banyak variabel dengan tipe data sama, dimana masing-masing elemen variabel mempunyai nilai indeks. Setiap elemen array mampu untuk menyimpan satu jenis data (yaitu: variabel). Suatu array dinyatakan dengan type, sehingga variabel yang bekerja akan dinyatakan dengan:
contoh type larik = array [1..10] of integer; var A:larik;

Data ke	value
1	35
2	-16
3	1
4	3
5	0
6	-100
7	13
8	43
9	56
10	67



Secara logika pendefinisian array di atas merupakan sekumpulan kotak , dimana tiap kotak mempunyai nilai indeks integer 1, 2, 3, ...,9, 10 tiap elemen array ditandai dengan:
A[1], A[2], A[3], A[4], A[5], A[6], A[7], A[8], A[9], A[10]

2. Sifat Array

Array merupakan struktur data yang statis, yaitu jumlah elemen yang ada harus ditentukan terlebih dahulu dan tak bisa di ubah saat program berjalan. Untuk menyatakan array dalam PASCAL kita harus terlebih dahulu:

Mendefinisikan jumlah elemen array,

Mendefinisikan tipe data dari elemen array

Contoh. const N=10;

type A= array [1..N] of integer;

pernyataan di atas merupakan penjelasan tentang array dengan satu dimensi. Pendefinisian array secara umum adalah sebagai berikut: jika kita ingin membuat beberapa array dengan tipe/jenis yang sama, kita lebih baik jika mendeklarasikan dengan type selanjutnya dengan deklarasi var.

SYNTAX

Type nama_array = ARRAY[bawah..atas] of tipe_data;
var variabel_array : nama_array;

atau dengan menggunakan statemen var :

var variabel_array : ARRAY[bawah..atas] of tipe_data;

Penjelasan: Bawah dan Atas menyatakan batas untuk array. tipe_data adalah merupakan tipe variabel yang dipunyai array (mis. Integer, char, real, dsb)

Contoh:

type int_array = ARRAY [1..20] of integer;

Pernyataan diatas adalah pernyataan untuk membentuk suatu array bernama int_array,yang berisi 20 tempat untuk bilangan integer. Setiap posisi disebut elemen, yang menyimpan suatu bilangan integer.langkah berikutnya adalah membuat suatu variabel kerja dengan tipe int_array yaitu,

var numbers : int_array;

kita bisa melakukan operasi pada setiap elemen dari numbers secara individual. Contoh kita bisa memberi nilai pada suatu elemen array seperti berikut:

numbers[2] := 10;

perintah ini memberikan suatu nilai integer 10 pada elemen ke-2 dari array numbers.

Nomor dari elemen ditempatkan didalam kurung tegak. Contoh berikut adalah merupakan array yang menyimpan variabel-variabel integer. Data dengan tipe integer hanya bisa dimasukkan satu persatu, kemudian baru bisa ditampilkan di monitor secara bersamaan

Contoh 1 :

```
program contoharraydimensi1;
uses wincrt;
type angka=array [1..10] of integer;

var x: angka;
   i,jum:byte;
   total:longint;
   rata:real;
   max,min:integer;
   lagi:char;

begin
{mengisi data nilai mhs}
i:=0; total:=0;lagi:='y';
repeat
begin
    i:=i+1;     write('isi angka ke  ',i,'  ');
    readln(x[i]);     total:=total+x[i];
    write('apakah akan mengisi lagi <y/t>?  ');
    readln(lagi);
end
until lagi <> 'y';
jum:=i-1;writeln;
writeln('DATA angka YANG TELAH DIMASUKKAN');
writeln('---------------------------------------');
writeln('NO          nilai');
writeln('---------------------------------------');
for i:=1 to jum do
   writeln(i:4,x[i]:8);
writeln('---------------------------------------');
{mencari rata-rata}
rata:=total/jum;
writeln('total nilai=  ',total:8);
writeln('nilai rata-rata dari ',jum:6, ' nilai di atas adalah  ',rata:8:2);
{mencari nillai terbesar dan terkecil}
max:=x[1];min:=x[1];
for i:=2 to jum do
begin
    if x[i] > max then max:=x[i];
    if x[i] < min then min:=x[i];
end;
writeln('nilai terbesar adalah  ',max:8);
```

```
writeln('nilai terkecil adalah  ',min:8);
end.
```

Contoh 2. Salinlah program pengelolaan nama dan nilai mahasiswa berikut untuk membantu memahami kinerja larik dimensi satu.

```
program week10_pdd3;
uses crt;
const max=10;
type       array1=array[1..max] of string;
  array2=array[1..max] of real;

var nama,jur:array1;
  ipk:array2;
  pil,i,n:integer;

procedure tambah_data(var x,y:array1);
begin
writeln('Menambah data baru');
inc(n);
write('masukkan nama baru      : ');readln(x[n]);
write('masukkan prodinya       : ');readln(y[n]);
writeln('Data sudah ditambahkan');
end;

procedure isi_ipk(var x,y:array1;var z:array2);
var nama_isi:string; ada:boolean;
label ulang;
begin
ada:=false;
writeln('Mengisi ip mahasiswa');
write('masukkan nama mhs yang akan diisi ipnya : ');readln(nama_isi);
for i:=1 to n do
begin
  if x[i]=nama_isi then
  begin
                    ada:=true;
                    if z[i]<>0 then writeln('Nama ',nama_isi,' sudah pernah diisi
ipknya')
                    else
                    begin
                              ulang:
                              write('masukkan ipk mhs tsb : ');readln(z[i]);
                              if (z[i]<=0.0) or (z[i]>4.0) then
                              begin writeln('Data tidak valid, ulangi pengisian');goto
ulang;end;
                              writeln('Pengisian berhasil');
                    end;
```

```
  end;
end;
if not ada then writeln('nama ',nama_isi,' tidak ditemukan pengisian gagal');
end;

procedure cetak_data(var x,y:array1;var z:array2);
begin
writeln('DAFTAR MAHASISWA DAN IPNYA');
writeln('-------------------------------------------');
writeln('No   Nama          Prodi         IPK');
writeln('-------------------------------------------');
for i:=1 to n do writeln(i:2,' ',x[i]:10,'    ',y[i]:14,'    ',z[i]:5:2);
writeln('-------------------------------------------');
writeln('Saat ini tercatat ',n, 'mahasiswa');
end;

begin
n:=0;
repeat
        clrscr;
        writeln('Pengelolaan data mahasiswa');
        writeln('1. Input data');
        writeln('2. Cetak data');
        writeln('3. Input IP');
        writeln('0. Selesai');
        write('Pilih menu  : ');readln(pil);
        case pil of
        1: if n=max then writeln('Kelas sudah penuh') else tambah_data(nama,jur);
        2: if n=0 then writeln('Data masih kosong') else cetak_data(nama,jur,ipk);
        3: if n=0 then writeln('Belum ada mhs terdaftar') else isi_ipk(nama,jur,ipk);
        0: writeln('Terimakasih') else writeln('Anda salah pilih menu');end;
        readln;
until pil=0;
end.
```

SOAL Minggu 1.

Soal array dimensi satu: Buatlah program untuk menginputkan data numeric bulat dengan jumlah tak tentu (gunakan while do atau repeat until), kemudian cetaklah data tersebut, dan temukan nilai terbesar dan terkecil dari data yang anda inputkan. Gunakan modul input data, cetak data dan modul maxmin

MODUL KE-2
LARIK DIMENSI DUA

Kompetensi: Mahasiswa mampu mengenali larik dimensi dua, mampu mendeklarasikan dan menggunakan larik dimensi dua dalam menyelesaikan masalah dalam pemrograman

1. Pengertian Larik Dimensi Dua

Larik dimensi dua dapat dipahami sebagai larik dengan elemen baris dan kolom. Pemanfaatan array tidak hanya dapat digunakan untuk menyimpan data dalam bentuk satu dimensi tetapi juga dapat digunakan untuk menyimpan data dalam bentuk 2 dimensi. Misal ada data dalam bentuk representasi sebagai berikut:

Tahun/Prodi	2012	2013	2014	2015	2016
Informatika	60	65	78	100	98
Kimia	100	97	102	76	94
Geologi	120	130	132	142	140
Mesin	200	180	210	187	176

Pada table di atas, maka index baris digunakan untuk menyimpan data prodi dan index kolom digunakan untuk menyimpan data tahun. Misal a[3][2] artinya jumlah mahasiswa pada prodi geologi pada tahun 2013. Atau dalam merepresentasikan matrix berorde mxn sehingga index baris menggunakan m dan index kolom menggunakan n. Pada pengelolaan larik berdimensi dua digunakan dua index, yaitu index baris dan index kolom.

Cara mendeklarasikan larik dimensi dua adalah sebagai berikut:

TYPE nama_tipe[jumbaris,jumkolom] of tipe data

Contoh:

TYPE larik2d= array[1..10,1..10] of string;

TYPE matrix= array[1..5,1..4] og integer;

VAR a,b,c:larik2d.

Mat_a, Mat_b:matrix;

Pada deklarasi di atas maka variable a,b,c mempunyai dimensi dua dengan maksimum baris 10 dan maksimum kolom 10. Sedangkan variable mat_a dan mat_B memiliki tipe matrix.

2. Cara memberikan nilai pada larik Dimensi Dua dan mencetaknya

Seperti halnya pada larik berdimensi satu, maka pengisian atau pencetakan data pada larik dimensi dua juga menggunakan loop. Akan tetapi karena index ada dua, yaitu baris dan kolom maka loop yang digunakan juga harus berupa loop bersarang (nested loop).

Struktur penulisan array dengan menggunakan perulangan adalah :

```
//index untuk baris
For (index awal baris to index akhir baris)
begin
//index untuk kolom nested di dalam index baris
  For (index awal kolom to index akhir kolom)
  begin
      pengisian larik
      Manipulasi larik
      Pencetakan larik
  End baris
  End kolom
```

Berikut adalah conroh program untuk memasukkan tiap elemen maka, diperlukan suatu procedure dengan menggunakan struktur pengulangan for ...do tersarang seperti berikut:

Latihan 1.

```
uses wincrt;

type matrix=array[1..10,1..10] of integer;

var A,B,T:matrix;
   bar_a,bar_b,kol_a,kol_b:byte;
   i,j:byte;

procedure isi_matrix(var X:matrix;m,n:byte);
begin
for i:=1 to m do
begin
    for j:=1 to n do
    begin
        write('masukkan elemen ke ',i,',',j,' = ');          readln(X[i,j]);
    end;
end;
end;
procedure cetak_matrix(X:matrix;m,n:byte);
begin
for i:=1 to m do
begin
    for j:=1 to n do write(X[i,j]:5);
    writeln;
end;
end;

procedure Transpose(var X:matrix;m,n:byte);
begin
for i:=1 to n do
begin
    for j:=1 to m do T[i,j]:=X[j,i];
end;
end;

begin{program utama}
      writeln('mengisi matrix A');
      write('baris A = ');readln(bar_a);
      write('kolom A = ');readln(kol_a);
      isi_matrix(A,bar_a,kol_a);
      writeln('mengisi matrix B');
      write('baris B = ');readln(bar_b);
      write('kolom B = ');readln(kol_b);
      isi_matrix(B,bar_b,kol_b);
      clrscr;
      writeln('MATRIX A=');
```

```
        cetak_matrix(A,bar_a,kol_a);
        writeln;
        writeln('MATRIX B=');
        cetak_matrix(B,bar_b,kol_b);
        writeln;
        writeln('MATRIX A TRANSPOSE');
        Transpose(A,bar_a,kol_a);
        cetak_matrix(T,kol_a,bar_a);
        writeln;
        writeln('MATRIX b TRANSPOSE');
        Transpose(b,bar_b,kol_b);
        cetak_matrix(T,kol_b,bar_b);
    end.
```

5. Operasi pada Array

Sifat masing-masing elemen array mengikuti jenis data yang dimilikinya, untuk array dengan tipe bilangan integer atau real kita bisa melakukan berbagai standar operasi aritmatika seperti penjumlahan, perkalian, pengurangan, dsb. Yang perlu di garis bawahi, bahwa sifat dari array dimanfaatkan untuk operasi matrik.

a. Mencari Harga Tertentu pada Array

Mencari suatu elemen data di dalam suatu data merupakan suatu kejadian yang sering kita alami, contoh: mencari nama mahasiswa dari daftar presensi. Pencarian beruntun (sequence), merupakan suatu teknik untuk mencari suatu elemen dalam suatu sistim yang lebih besar.

Contoh.

Misal array A[8], dengan elemen sbb:

A

60 12 76 23 11 42 18 42

Untuk menari apakah bilangan x=11 ada didalam tabel maka dilakukan pemeriksaan terhadap :

60 12 76 23 11

Sehingga ditemukan xpada elemen ke-5, dalam bahasa PASCAL diterjemahkan seperti berikut:

```
type PITA = ARRAY [1..8] of integer;
var AKU: PITA;
procedure CARI_MATRIK(AKU: PITA);
var
i: integer; {faktor pengulang}
```

```
begin
for i:=1 to 8 do
begin
if AKU[i]:= 11 then
writeln(‘ terdapat bilangan 11 dalam pita ini ‘);
else
writeln(‘ tidak ada bilangan 11, pencarian berhenti ‘);
end;
end;
```

b. Mencari Harga Maksimum pada Array

Misal array di atas kita cari harga yang tertinggi, maka kita perlu menentukan nilai tertinggi dahulu sebelum melakukan pencarian ; diawali dengan nilai maksimum=0

```
procedure CARI_MAKSIMUM(AKU: PITA);
var
i: integer; {faktor pengulang}
MAKS : integer;
begin
MAKS := AKU[1];
for i:=1 to 8 do
begin
if AKU[i]> MAKS then
MAKS:= AKU[i];
End;
Writeln(‘NILAI MAKSIMUM = ’,MAKS);
end;
```

b. Mencari Harga Minimum pada Array

Misal array di atas kita cari harga yang terendah, maka kita perlu menentukan nilai terendah dahulu sebelum melakukan pencarian ; diawali dengan nilai maksimum=3200

```
procedure CARI_MINIMUM(AKU: PITA);
var
i: integer; {faktor pengulang}
MIN : integer;
begin
MIN := 3200;
for i:=1 to 8 do
begin
if AKU[i]< MIN then
MIN:= AKU[i];
end;
writeln(‘NILAI MINIMUM = ’,MIN);
end;
```

c. Matrik

Sebagai perwujudan dari array dua dimensi, operasi aritmatika seperti penjumlahan, perkalian, dan pengurangan bisa dilakukan.

Contoh.

- Mendefinisikan Elemen

```
Program OPERASI_MATRIK;
uses wincrt;
type
matrik=array[1..100,1..100] of real;
var
m,n, p, q: integer; {dimensi dari matrik}
A,B,C: matrik; {matrik A, B sebagai input, C sebagai hasil}
```

- Membaca Elemen Matrik

```
procedure bacamatrik(var A:matrik; m,n:integer);
var
i,j: integer; {faktor pengulang}
begin {read}
for i:=1 to m do
begin {do}
for j:=1 to n do
read(A[i,j]);
readln;
end; {do}
end; {read}
```

- Menampilkan Elemen Matrik

```
procedure tulismatrik(A:matrik; m,n:integer);
var
i,j: integer; {faktor pengulang}
begin {write}
for i:=1 to m do
begin {tiap baris}
writeln;
for j:=1 to n do
write(A[i,j]:6:2);
end; {tiap baris}
writeln;
end; {write}
```

- Penjumlahkan Matrik

```
procedure check_matrik(A,B,C:matrik; m,n,p,q:integer);
var i,j :integer;
```

```
begin
if (m=p) and (n=q) then
begin
for i:=1 to m do
begin
for j:=1 to n do
begin
C[m,n]=A[m,n]+B[m,n])
end;
end;
end
else
writeln('DIMENSI MATRIK TIDAK COCOK')
end;
```

- Pengurangan Matrik

```
procedure check_matrik(A,B,C:matrik; m,n,p,q:integer);
var i,j :integer;
begin
if (m=p) and (n=q) then
begin
for i:=1 to m do
begin
for j:=1 to n do
begin
C[m,n]=A[m,n]- C[m,n])
end;
end;
end
else
writeln('DIMENSI MATRIK TIDAK COCOK')
end;
```

-. Perkalian Matrik

```
procedure perkalian_matrik(A,B,C:matrik; m,n,p,q:integer);
var i,j, k :integer;
C1: matrik;
begin
if (n=p) then
begin
for i:=1 to m do
begin
for j:=1 to p do
begin {inner product}
C1[i,j]:=0;
for k:=1 to n do
C1[i,j]:=C1[i,j]+A[i,k]*B[k,j];
```

```
end; {inner product}
end;
n:=q;
for i:=1 to m do
for j:=1 to n do
C[i,j]:=C1[i,j];
end
else
writeln('DIMENSI MATRIK TIDAK COCOK')
end;
```

- Transpose Matrik

```
procedure Transpose(A,B:matrik; m,n,p,q:integer);
var i,j:integer;
begin
for i:=1 to n do
begin
for j:=1 to m do
begin
B[m,n]=A[n,m]
end;
end;
end;
```

-. Mencari Elemen yang Kosong pada Matrik

```
procedure CHECK_ZERO_ELEMEN(A,matrik; m,n:integer);
var i,j:integer;
begin
for i:=1 to m do
begin
for j:=1 to n do
begin
if B[m,n]= 0 then
writeln (‘terdapat elemen yang kosong’)
else
writeln (‘tidak terdapat elemen yang kosong’)
end;
end;
end;
```

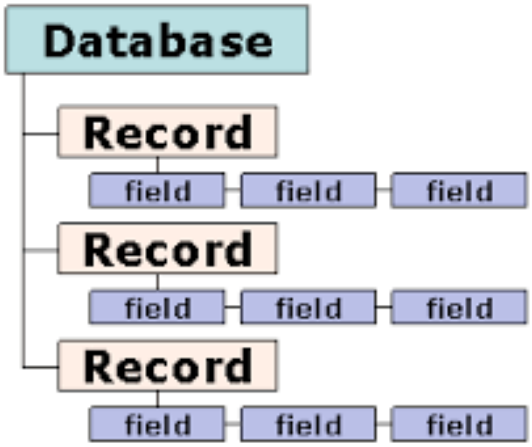
Soal Minggu ke dua.

Buatlah program untuk menginputkan dan mencetak data mahasiswa dari tahun 2015-2018 untuk 5 program studi, yaitu Teknik Kimia, Teknik Industri, Teknik Mesin, Teknik Elektro dan Teknik Informatika . Sajikan dalam bentuk tabel

MODUL KE -3
RECORD DAN ARRAY RECORD SEDERHANA

Kompetensi mahasiswa : Mampu memahami, menerapkan dan mengendalikan penggunaan record, larik reord dalam pemrograman tertruktur untuk menyelesaikan masalah

Sebuah record rekaman disusun oleh beberapa field. Tiap field berisi data dari tipe dasar / bentukan tertentu. Record mempunyai kelebihan untuk menyimpan suatu sekumpulan elemen data yang berbeda beda tipenya (di banding array). Contoh :



First Name	Last Name	Address	City	Age
Mickey	Mouse	123 Fantasy Way	Anaheim	73
Bat	Man	321 Cavern Ave	Gotham	54
Wonder	Woman	987 Truth Way	Paradise	39
Donald	Duck	555 Quack Street	Mallard	65
Bugs	Bunny	567 Carrot Street	Rascal	58
Wiley	Coyote	999 Acme Way	Canyon	61
Cat	Woman	234 Purrfect Street	Hairball	32
Tweety	Bird	543	Itotitaw	28

Records

Gambar 2.1 Gambaran Record dan larik record

Cara pendeklarasian dari record adalah sbb:

• Mendefinisikan tipe dari record (jumlah field, jenis tipe data yang dipakai),

• Mendefinisikan variabel untuk dilakukan operasi.

SYNTAX

```
type nama_record = record
identifier_1 : tipe_data_1;
:
:
identifier_n : tipe_data_n;
end;
var variabel : nama_record;
```

Contoh.

```
type Data_mahasiswa = record
    Nama : string;
    Usia : integer;
    Kota : String;
    Kodepos : integer; end;

Var
    x: Data_mahasiswa;
```

1. Pengaksesan Elemen Record

Cara mengacu pada tiap field pada record pada contoh di atas adalah sbb:

```
x.Nama
x.Usia
x.Kota
x.Kodepos  dimana x adalah nama variabel bertipe record
```

Contoh.

```
program RECORD_INTRO;
type tanggal = record
bulan, hari, tahun : integer;
end;
var waktu : tanggal;
begin
waktu.hari :=25;
waktu.bulan:=2;
waktu.tahun:= 2018;
writeln('hari ini adalah ',waktu.hari,':',waktu.bulan,':', waktu.tahun)
end.
```

2. Pengunaan With … do

Pernyataan with untuk lebih menyederhanakan pengaksesan field-field pada record.

Misalkan pernyataan :

x.Nama
x.Usia
x.Kota
x.Kodepos

menjadi

with x do
writeln(Nama,Usia,Kota,Kodepos);
(tidak perlu lagi menyebutkan variable)

contoh.

```
program RECORD_INTRO;
type tanggal = record
bulan, hari, tahun : integer;
end;
var waktu : tanggal;
begin {program utama}
with waktu do {mulai with}
begin
  hari :=25;
  bulan:=2;
  tahun:=2018;
writeln('hari ini adalah ',hari,':',bulan,':', tahun)
end {akhir with}
end.
```

3. Array dari Record

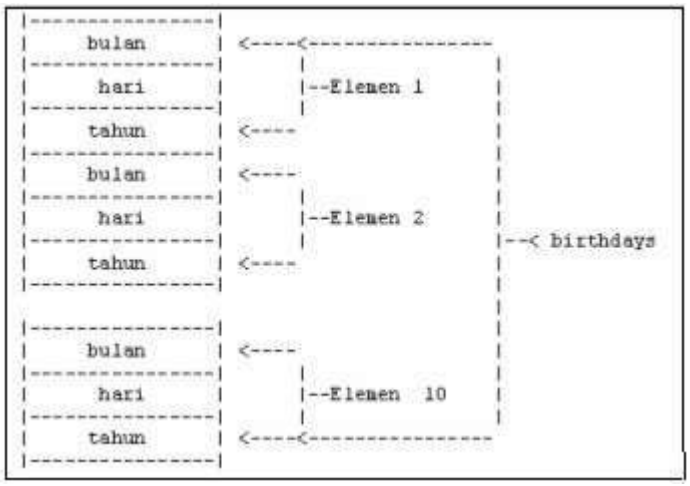
Suatu array dapat juga berisi record contoh suatu deklarasi record tanggal.

```
type tanggal = record
bulan, hari, tahun : integer;
end;
var waktu : tanggal;
```

kemudian kita membentuk suatu array dari record ini, namakan birthdays.

```
var birthdays : array[1..10] of tanggal;
```

pernyataan ini akan membentuk suatu array dengan 10 elemen. Dimana tiap elemen adalah sebuah record tanggal, yaitu, terdiri atas bulan, hari, tahun dengan tipe data Integer. Digambarkan seperti berikut:



Contoh Pemberian nilai awal dari masing-masing elemen birthdays:

```
Birthdays[1].hari :=25;
Birthdays[1].bulan:=2;
Birthdays[1].tahun:=1999;
```

4. Record di dalam Record

Record bisa berisi record lain sebagai field. Seperti contoh record tanggal dan jam dikombinasikan menjadi sebuah record saat ini,

```
type tanggal = record
bulan, hari, tahun : integer;
end;
type waktu =record
jam, menit, detik : integer;
end;
type waktu_ini =record
tanggal_ini : tanggal;
waktu_ini : waktu
end;
```

Kemudian kita perlu membuat variabel kerja

```
var saat_ini : waktu_ini;
```

pemberian nilai akan terjadi seperti di bawah ini:

```
saat_ini.tanggal.bulan:= 11;
saat_ini.tanggal.hari:= 2;
saat_ini.tanggal.tahun:= 1985;
saat_ini.waktu.jam:= 3;
saat_ini.waktu.menit:= 3;
saat_ini.waktu.detik:= 33;
```

Latihan 1: Program dibawah adalah program pengelolaan gudang dengan menggunakan record array tunggal. Salinlah dan kerjakan menu yang belum dibuat

```
program record_kelasB_week11;
uses crt;
const max=20;
type gudang=record
        kode,nama:string;
        harga:real;
        stok,keluar:integer;
        end;
        larik=array[1..max] of gudang;
var barang:larik;
        i,n,pil:byte;

procedure tambah_brg(var x:larik);
var kod:string;
        ya:char;
label ulang;

begin
writeln('Menambah Barang baru');
repeat
        ulang:
        write('masukkan kode barang yang baru : ');readln(kod);
        {cek ke dalam array apakah kode sudah digunakan atau belum}
        for i:=1 to n do
        begin if x[i].kode=kod then begin writeln('Kode sudah digunakan, masukkan kode
lain');
             goto ulang;end;
        end;
        inc(n);
        x[n].kode:=kod;
        write('masukkan nama barang                    : ');readln(x[n].nama);
        write('masukkan harga barang          : ');readln(x[n].harga);
        write('masukkan jumlahnya             : ');readln(x[n].stok);
        writeln('Barang yang baru sudah berhasil diinputkan');
        writeln;
        write('Apakah akan memasukkan barang baru yang lain<y/t> ? ');readln(ya)
until(ya<>'Y') and (ya<>'y');
end;
```

```
procedure cetak_data(x:larik);
begin
writeln('Laporan Kondisi Barang di Gudang Balapam Jaya');writeln;
writeln('-------------------------------------------------');
writeln(' No   Kode   Nama Barang     Harga   Stok   Keluar');
writeln('-------------------------------------------------');
for i:=1 to n do with x[i] do
writeln(i:2,'    ',kode:5,'    ',nama:12,'   ','Rp',harga:6:2,'  ',stok:3,'  ',keluar:4);
writeln('-------------------------------------------------');
end;

procedure tambah_stok(var x:larik);
var tambah:string; jum:byte;
        ada:boolean;

begin
ada:=false;
writeln('Menambah stok barang di gudang');
write('masukkan nama barang yang akan ditambah stok : ');readln(tambah);
{cek nama barang di gudang}
for i:=1 to n do
begin
        if x[i].nama=tambah then
        begin
                ada:=true;
                write('berapa stok tambahan untuk barang ',tambah,' : ');
                readln(jum);
                x[i].stok:=x[i].stok+jum;
        end;
end;
if not ada then writeln('Barang ',tambah,' belum ada di gudang, gunakan menu nomor 1');
end;

procedure ambil_brg(var x:larik);
var ambil:string; jum:byte;
        ada:boolean;
        ya:char;
begin
writeln('Mengambil barang di gudang Pengok Jaya');
write('masukkan nama barang yang akan diambil : ');readln(ambil);
write('Berapa yang akan diambil   : ');readln(jum);
ada:=false;
for i:=1 to n do
begin if x[i].nama=ambil then
        begin  ada:=true;
                {cek stok}
                if x[i].stok=0 then writeln('Barang ',ambil,' sedang kosong stoknya')
                else if x[i].stok<jum then
```

```
                    begin writeln('Maaf stok tinggal ',x[i].stok);
                                write('apakah akan diambi <y/t> ? '); readln(ya);
                        if (ya='y') or (ya='Y') then
                        begin
                                writeln('anda mengambil ',ambil,' sebanyak ',x[i].stok,' buah');
                                x[i].keluar:=x[i].keluar+x[i].stok;
                                        x[i].stok:=0;
                          end
                          else writeln('Barang tidak jadi diambil');
                 end
                else {ada dan stok cukup}
                begin
                writeln('anda akan mengambil ',ambil,' sebanyak ',jum);
                x[i].keluar:=x[i].keluar+jum;
                x[i].stok:=x[i].stok-jum;
                end;
        end;
end;
if not ada then writeln('Barang ',ambil,' tidak tersedia');
end;

procedure cetak_pendapatan(x:larik);
var inkam,subtotal:real;

begin
inkam:=0;
writeln('Pendapatan Gudang Pengok Jaya');writeln;
writeln('------------------------------------------------------------');
writeln('No   Kode     Nama Barang       Harga    Terjual   Subtotal');
writeln('------------------------------------------------------------');
for i:=1 to n do with x[i] do
begin
        subtotal:=x[i].keluar*x[i].harga;
        writeln(i:2,'   ',kode:4,'   ',nama:12,'   ','Rp',harga:8:2,'  ',keluar:4,'   ',
        'Rp ',subtotal:8:2);
        inkam:=inkam+subtotal;
end;
writeln('------------------------------------------------------------');
writeln('                       Pendapatan         Rp : ',inkam:8:2);
end;

procedure cek_data(x:larik);
var cek:string;cekharga:real;
        ada:boolean;
        pilih:byte;

begin
writeln('Mengecek kondisi barang tertentu di gudang');
writeln('1. Cek Berdasar Kode');
```

```
writeln('2. Cek Berdasar Nama barang');
writeln('3. Cek Berdasar Harga Barang ');
writeln('0. Kembali ke menu utama');
write('masukkan pilihan pencarian : ');readln(pilih);
case pilih of
1: begin
  ada:=false;
  write('masukkan kode barang yang akan dicari : ');readln(cek);
  for i:=1 to n do if x[i].kode=cek then
  begin ada:=true;
                writeln('Kode barang ',cek,' tersedia di posisi ke ',i);
                writeln('Nama barang   : ',x[i].nama,' dengan harga Rp ',x[i].harga:6:2);
                writeln('Sisa stok          : ',x[i].stok);
                writeln('Terjual            : ',x[i].keluar);
        end;
        if not ada then writeln('Kode tidak ditemukan');
        end;
2: begin
  ada:=false;
  write('masukkan nama barang yang akan dicari : ');readln(cek);
  for i:=1 to n do if x[i].nama=cek then
  begin ada:=true;
                writeln('Nama barang ',cek,' tersedia di posisi ke ',i);
                writeln('Kode barang   : ',x[i].kode,' dengan harga Rp ',x[i].harga:6:2);
                writeln('Sisa stok          : ',x[i].stok);
                writeln('Terjual            : ',x[i].keluar);
                writeln;
        end;
        if not ada then writeln('Nama barang ',cek,' tidak ditemukan');
        end;
3: begin
  ada:=false;
  write('masukkan Harga barang yang akan dicari : ');readln(cekharga);
  for i:=1 to n do if x[i].harga=cekharga then
  begin ada:=true;
                writeln('Kode barang ',x[i].kode,' tersedia di posisi ke ',i);
                writeln('Nama barang   : ',x[i].nama);
                writeln('Sisa stok          : ',x[i].stok);
                writeln('Terjual            : ',x[i].keluar);
                writeln;
        end;
        if not ada then writeln('Harga barang tsb  tidak ditemukan');
        end;
0: writeln('Enter untuk kembali ke menu') else writeln('Anda salah pilih nomor menu');
end;
end;

procedure cetak_order(x:larik);
begin
```

```
writeln('Laporan Kondisi Barang di Gudang Balapam Jaya yang harus segera
diorder');writeln;
writeln('-----------------------------------------------------');
writeln(' No   Kode   Nama Barang      Harga     Stok   ');
writeln('-----------------------------------------------------');
for i:=1 to n do with x[i] do if stok<=5 then
writeln(i:2,'    ',kode:5,'    ',nama:12,'     ','Rp',harga:8:2,'    ',stok:3);
writeln('-----------------------------------------------------');
end;

begin
n:=0;
repeat
        clrscr;
        writeln(' PENGELOLAAN GUDANG PADA TOKO BALAPAN JAYA');writeln;
        writeln('1. Tambah barang baru');
        writeln('2. Menambah stok');
        writeln('3. Mengambil barang dari gudang');
        writeln('4. Laporang Kondisi Gudang');
        writeln('5. Laporan pendapatan ');
        writeln('6. Cek ketersediaan barang di gudang');
        writeln('7. Cetak barang wajib order');
        writeln('0. Selesai');
        write('Pilih menu 0-7 : ');readln(pil);
        case pil of
        1: if n>=max then writeln('Gudang sudah penuh') else tambah_brg(barang);
        2: tambah_stok(barang);
        3: if n=0 then writeln('Gudang masih kosong') else ambil_brg(barang);
        4: if n=0 then writeln('Gudang masih kosong') else cetak_data(barang);
        5: if n=0 then writeln('Gudang masih kosong') else cetak_pendapatan(barang);
        6: if n=0 then writeln('Gudang masih kosong') else cek_data(barang);
        7: if n=0 then writeln('Gudang masih kosong') else cetak_order(barang);
        {8: edit barang hanya harga yang bs diedit
         9:hapus barang hanya yang stok nol.
         Dikumpulkan hard copy, source+output+penjelasan 9 menu}
        0: writeln('Terimakasih') else writeln('Anda salah memilih menu');
        end;
        readln;
until pil=0;
end.
```

Tugas: Kerjakan menu nomor 8,9 (dua saja)

MODUL KE-4
ARRAY RECORD MULTIPLE
(Searching, Sorting, Filtering)

Kompetensi mahasiswa : Mampu memahami, menerapkan dan mengendalikan penggunaan record, larik record multiple dalam pemrograman tertruktur untuk menyelesaikan masalah searching, sorting dan filtering yang lebih kompleks

Record multiple terdiri dari beberapa jenis, missal record di dalam record, array di dalam record. Record dalam array ataupun suatu program yang mempunyai banyak record.

Mendeklarasikan record di dalam record :

```
type rec_ttl=record
    tgl,bulan:byte;
    tahun:longint;
end;
type rec_mhs=record
    nim,nama:string;
    ttl:rec_ttl;
    prodi:string;
    ip:real;
end;
var mhs:rec_mhs;
```

Mendeklarasikan array di dalam record:

```
Type larik1=array[1..10] of string;
type rec_anggota=record
    no_anggota,nama:string;
    kd_bk,jd_bk,tgp,tgk:larik1;
    n1:byte;
    end;
```

larik_anggota=array[1..max] of rec_anggota;
var buku:larik_buku; member:larik_anggota;

Yang perlu diperharhatikan dalam pengaksesan record kompleks seperti contoh di atas adalah dalam penempatan index. Jika larik dimiliki oleh atribut, maka index ditempatkan pada atribut, dan jika larik dimilkiki oleh variable maka index ditempatkan pada variable. Atau jika keduanya memiliki larik maka masing-masing baik variable maupun atribut harus diberikan index yang berbeda.

Penanganan Record array muliple adalah menggunakan record array lebih dari satu dalam suatu program. Biasanya untuk masalah- masalah yang lebih kompleks. Misalnya, suatu perpustakaan akan menangani persoalan pendataan keanggotaan, peminjaman dan pengembalian buku. Jika dilakukan pengurutan data maka sorting dilakukan dengan array record bayangan. Pada program berikut sudah dibuat beberapa menu (silhakan dicoba dulu), kemudian lengkapilah menu-menu yang belum dibuat.

```
program perpustakaan_praktikum_SD;
uses wincrt;
const max=10;
type rec_buku= record
    kode,judul,pengarang: string;
    jb:byte;
    end;
    larik_buku=array[1..max] of rec_buku;

type anggota=record
    no_anggota,nama,kodepinjam,tgl_p,tgl_k: string;
    status:boolean;
    end;
    larik_anggota=array[1..max] of anggota;
var buku,bayangan:larik_buku;
   pinjam:larik_anggota;
   i,j,n,m,pil:byte;

procedure tambah_judul(var x:larik_buku);
var baru:string;
   ya:char;
label ul;

begin
repeat
begin
    writeln('Menambah Judul Baru ');
    ul:
```

```
    write('masukkan kode buku baru : ');readln(baru);
{validasi dulu kodenya harus unik}
    for i:=1 to n do
    begin if x[i].kode=baru then
          begin writeln('Kode sudah digunakan, ulang!! ');goto ul;end;
          end;
    {validasi diterima}
    inc(n);
    x[n].kode:=baru;
    write('Masukkan judul buku yang baru    : ');readln(x[n].judul);
    write('Masukkan pengarangnya            : ');readln(x[n].pengarang);
    write('Berapa jumlah bukunya            : ');readln(x[n].jb);
    writeln;write('Tambah judul lagi<y/t> ? ');readln(ya);
end
until(ya<>'y');
end;


procedure cetak_buku(var x:larik_buku);
begin
writeln('DAFTAR BUKU DI PERPUSTAKAAN SARI ILMU');
writeln('-----------------------------------------------');
writeln('No   Kode       Judul       Pengarang   Jumlah ');
writeln('-----------------------------------------------');
for i:=1 to n do with x[i] do
writeln(i:3,'   ',kode:4,'   ',judul:15,'   ',pengarang:10,'   ',jb:3);
writeln('-----------------------------------------------');
end;

procedure tambah_jumlah(var x:larik_buku);
var cari:string;
   pos,tambah:byte;
   ada:boolean;

begin
ada:=false;
writeln('Menambah jumlah buku ');
write('masukkan kode buku yang akan ditambah jumlahnya  : ');readln(cari);
for i:=1 to n do
begin
    if x[i].kode=cari then begin ada:=true;pos:=i;end;
end;
if ada then
begin
    writeln('Buku dengan kode ',cari);
    writeln('Judul       : ',x[pos].judul);
    writeln('Pengarang   : ',x[pos].pengarang);
    writeln('Jumlah buku semula : ',x[pos].jb);
    write('berapa tambahan buku untuk kode ',cari,' ? ');readln(tambah);
```

```
    if tambah>0 then x[pos].jb:=x[pos].jb+tambah;
    writeln('Buku dengan kode ',cari,' sekarang berjumlah ',x[pos].jb,' exemplar');
end
else writeln('Kode tersebut tidak ada dalam daftar di perpus kami');
end;
procedure edit_buku(var x:larik_buku);
var cari:string;
   pos,pilih:byte;
   ada:boolean;

begin
ada:=false;
writeln('Mengedit data buku ');
write('masukkan kode buku yang akan diedit  : ');readln(cari);
for i:=1 to n do
begin
    if x[i].kode=cari then begin ada:=true;pos:=i;end;
end;
if ada then
begin
    writeln('Buku dengan kode ',cari);
    writeln('Judul       : ',x[pos].judul);
    writeln('Pengarang   : ',x[pos].pengarang);
    writeln('Jumlah buku : ',x[pos].jb);
    writeln('Akan diedit : 1. Judul   2. Pengarang');readln(pilih);
    if pilih=1 then
    begin
        write('Masukkan judul buku setelah diedit : ');readln(x[pos].judul);
    end
    else if pilih=2 then
    begin
          write('Masukkan pengarang setelah diedit : ');readln(x[pos].pengarang);
    end
    else writeln('masukan pilihan salah ');
end
else writeln('Kode tersebut tidak ada dalam daftar di perpus kami');
end;

procedure cari_buku(var x:larik_buku); {searching}
var cari:string;
   ketemu:boolean;

begin
writeln('Mencari buku berdasar judul');
write('Masukkan judul buku yang akan dicari : ');readln(cari);
ketemu:=false;
for i:=1 to n do
begin
    if (x[i].judul=cari)then
```

```
    begin
        ketemu:=true;
        writeln('Buku dengan judul : ',cari);
        writeln('Pengarang         : ',x[i].pengarang);
        writeln('Stok saat ini     : ',x[i].jb);
        writeln;
    end;
end;
if not ketemu then writeln('Buku dengan judul ',cari,' tidak ditemukan');
end;

procedure hapus_buku(var x:larik_buku);
var jdl_hapus:string;
   pengarang_hapus,ya:string;
   pos:byte;
   hapus:boolean;

begin
hapus:=false;
writeln('Menghapus buku dari Perpustakaan berdasar Judul dan Pengarang');
write('Masukkan judul buku yang akan dihapus   : ');readln(jdl_hapus);
write('Pengarangnya                          : ');readln(pengarang_hapus);
for i:=1 to n do
begin
    if (x[i].judul=jdl_hapus) and (x[i].pengarang=pengarang_hapus) then
    begin
        write('Buku ditemukan, apakah yakin akan dihapus <y/t> ? ');
        readln(ya);
        hapus:=true;
        if ya='y' then
        begin
            pos:=i;
            writeln('Buku berhasil dihapus');
            for j:=pos to n-1 do x[j]:=x[j+1];
            dec(n);
        end
        else writeln('Anda tidak jadi menghapus buku dari perpustakaan');
    end;
end;
if not hapus then writeln('Buku dengan judul dan pengarang tsb tidak ada');
end;

procedure urut_judul(var X:larik_buku); {sorting}
var dummy: rec_buku;

begin
{kirim buku ke record bayangan}
for i:=1 to n do bayangan[i]:=x[i];
{proses sorting}
```

```
for i:=1 to n-1 do
begin
    for j:=i+1 to n do
    begin
        if bayangan[i].pengarang>bayangan[j].pengarang then
        begin
            dummy:=bayangan[i];
            bayangan[i]:=bayangan[j];
            bayangan[j]:=dummy;
        end;
    end;
end;
{cetak hasil sorting};
writeln('DATA BUKU SESUDAH DIURUTKAN BERDASAR JUDUL SBB:
');writeln;
cetak_buku(bayangan);
end;


procedure tambah_anggota(var Y:larik_anggota);
var nomor:string;
   ya:char;
label ul;

begin
repeat
begin
    writeln('Menambah anggota baru Perpustakaan');
    ul:
    write('Masukkan nomor anggota yang baru       : ');readln(nomor);
    for i:=1 to m do
    begin
        if y[i].no_anggota=nomor then
        begin writeln('nomor sudah digunakan, ulang');goto ul; end;
    end;
    inc(m);
    y[m].no_anggota:=nomor;
    write('Masukkan nama anggota yang baru        : ');readln(y[m].nama);
    writeln;write('Mau tambah anggota lagi<y/t> ? ');readln(ya);
end
until(ya<>'y');
end;

procedure peminjaman(var X:larik_buku;var Y:larik_anggota);
var kode_pinjam:string;
   kode_buku:string;
   adap,adab:boolean;
   posp,posb:byte;
```

```
begin
adab:=false;adap:=false;
writeln('Transaksi Peminjaman Buku');
write('masukkan nomor anggota yang akan meminjam buku   :
');readln(kode_pinjam);
{cek di larik anggota}
for i:=1 to m do
begin if y[i].no_anggota=kode_pinjam then begin posp:=i;adap:=true;end;end;
if adap then {setelah yakin no anggota ada di larik peminjam,lanjutkan ke
peminjaman}
begin
    writeln('Selamat datang saudara ', y[posp].nama);
    write('Masukkan kode buku yang akan dipinjam : ');readln(kode_buku);
    {ngecek kode buku yang akan dipinjam di larik buku termasuk stoknya}
    for j:=1 to n do
    begin if (x[j].kode=kode_buku) and (x[j].jb>0) then
        begin adab:=true;posb:=j;end;
    end;
    if adab then     {lakukan proses pinjam}
    begin
            writeln('Buku tersedia'); y[posp].kodepinjam:=kode_buku;
            write('Masukkan tanggal pinjam : ');readln(y[posp].tgl_p);
            dec(x[posb].jb);
            writeln('Terimakasih atas kunjungan anda!');
    end
        else writeln('Maaf buku tidak tersedia untuk anda');
end
else writeln('Maaf anda belum menjadi anggota perpustakaan');
end;

procedure pengembalian(var x:larik_buku;var y:larik_anggota);
var kode_pinjam,kode_buku:string;
   adap:boolean;
   posp:byte;

begin
writeln('Transaksi pengembalian buku');
write('Masukkan nomor anggota anda : ');readln(kode_pinjam);
for i:=1 to m do
begin if y[i].no_anggota=kode_pinjam then begin adap:=true;posp:=i;end;
end;
if adap then
begin
    writeln('Masukkan kode buku yang akan dikembalikan : ');readln(kode_buku);
    if y[posp].kodepinjam=kode_buku then
    begin
        write('Masukkan tanggal kembali : ');readln(y[posp].tgl_k);
        writeln('Terimakasih atas kunjungan anda');
        {cek posisi buku di larik buku dan lakukan inc}
```

```
        for j:=1 to n do
            if x[j].kode=kode_pinjam then inc(x[j].jb);
    end
    else writeln('kode buku yang anda masukkan salah');
end
else writeln('Anda bukan anggota atau tidak sedang meminjam buku');
end;

procedure cetak_peminjaman(var y:larik_anggota);
begin
writeln('   DAFTAR PEMINJAMAN BUKU DI PERPUSTAKAAN');
writeln('-------------------------------------------------------------------');
writeln('No  No anggota  Nama         Kode buku    Tgl Pinjam    Tgl Kembali');
writeln('-------------------------------------------------------------------');
for i:=1 to m do
writeln(i:2,'  ',y[i].no_anggota:4,'  ',y[i].nama:10,'  ',y[i].kodepinjam:4,'   ',
      y[i].tgl_p:12,'  ',y[i].tgl_k:12);
writeln('-------------------------------------------------------------------');
end;

begin{menu utama}
repeat
begin
    clrscr;
    writeln(' MANAJEMEN BUKU PERPUSTAKAAN');
    writeln('1. Tambah judul baru');
    writeln('2. Menambah jumlah buku yang sudah ada');
    writeln('3. Edit buku');
    writeln('4. Tampil buku');
    writeln('5. Cari buku');
    writeln('6. Hapus Buku');
    writeln('7. Urutkan Buku berdasar Judul');
    writeln('8. Tambah anggota');
    writeln('9. Transaksi Peminjaman Buku');
    writeln('10. Transaksi Pengembalian Buku');
    writeln('11. Cetak Peminjaman');
    writeln('0. Selesai');
    write('Pilih 0-11 :' );readln(pil);
    case pil of
    1: if n=20 then writeln('Perpustakaan sudah penuh') else tambah_judul(buku);
    2: if n=0 then writeln('Belum ada buku dalam perpus') else tambah_jumlah(buku);
    3: if n=0 then writeln('Belum ada judul') else edit_buku(buku);
    4: if n=0 then writeln('Perpus masih kosong, tidak bisa ditampilkan') else
       cetak_buku(buku);
    5: if n=0 then writeln('Perpus kosong') else cari_buku(buku);
    6: if n=0 then writeln('Data buku masih kosong, tidak bisa dilakukan proses
hapus') else
       hapus_buku(buku);
    7: if n=0 then writeln('Data buku kosong, tidak bisa dilakukan proses urut') else
```

```
        urut_judul(buku);
      8: if m=20 then writeln('Tidak menerima anggota baru lagi') else
tambah_Anggota(pinjam);
      9: if (n=0) or (m=0) then writeln('Belum ada data buku atau anggota') else
        peminjaman(buku,pinjam);
      10: if (n=0) or (m=0) then writeln('Belum ada buku atau anggota') else
         pengembalian(buku,pinjam);
      11: if m=0 then writeln('Belum ada anggota') else cetak_peminjaman(pinjam);
      0: writeln('Terimakasih');
      end;
      readln;
end
until(pil=0);
end.
```

Soal Minggu ke 4.

Buatlah program untuk mengelola Dealer suatu kendaraan dengan ketentuan sbb:

1. Data kendaraan disimpan dengan atribut no_plat (unik),nama kendaraan dan harga
2. Data pembelian disimpan dengan atribut nama pembeli, no_plat kendaraan yang dibeli dan tanggal pembelian

Menu yang diinginkan adalah input data kendaraan, cetak kendaraan, input pembelian (dari kendaraan yang telah diinputkan) dan cetak penjualan kendaraan

MODUL KE 5
STACK (TUMPUKAN)

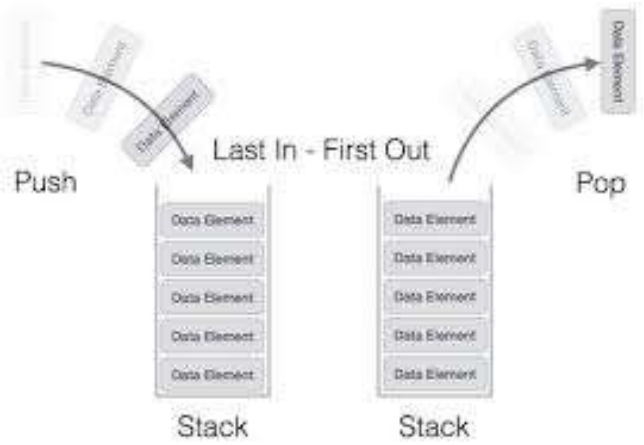
Kompetensi mahasiswa : Mampu memahami, menerapkan dan mengendalikan penggunaan struktur Tumpukan (stack). Membuat dan mendeklarasikan stack, operasi push, pop, cetak dan push acak dan pop acak

Secara sederhana, tumpukan bisa diartikan sebagai suatu kumpulan data yang seolah-olah ada data yang diletakkan diatas data yang lain. Satu hal yang perlu diingat adalah bahwa kita bisa menambah data, dan mengambil (menghapus) data lewat ujung yang sama, yang disebut sebagai ujung atas tumpukan (*top of stack*).

Untuk menjelaskan pengertian diatas kita ambil contoh sebagai berikut. Misalnya kita mempunyai dua buah kotak yang kita tumpukan, sehingga kotak kita letakkan diatas kotak yang lain. Jika kemudian tumpukan dua buah kotak itu kita tambah dengan kotak ketiga, keempat, kelima dan seterusnya, maka akan kita peroleh sebuah tumpukan kotak, yang terdiri dari N kotak.

Secara sederhana, sebuah tumpukan (stack) dapat digambarkan seperti diatas. Dari gambar ini kita bias mengatakan bahwa kotak B ada diatas kotak A dan ada dibawah kotak C. Dari gambar juga ditunjukan bahwa dalam tumpukan kita hanya bias menambah atau mengambil sebuah kotak lewat satu ujung, yaitu ujung bagian atas. Dapat dilihat pula bahwa tumpukan merupakan kumpulan data yang sifatnya dinamis, artinya kita bias menambah atau mengambil data darinya.

Penggambaran tumpukan juga tidak harus seperti diatas. Kita bias menggambarkan tumpukan dengan cara lain.



Dari gambar diatas menunjukan pengertian bahwa kotak yang terakhir dimasukkan bukan berarti memiliki posisi paling atas, tapi lebih ditekankan pada kotak yang paling dekat pada pintu masuk.

Dengan memperhatikan ilustrasi-ilustrasi yang disebutkan maka kita bias melihat bahwa tumpukan merupakan suatu senarai (*list*) yang mempunyai pengertian "masuk terakhir keluar pertama"(*Last In First Out* – LIFO).

OperasiPadaTumpukan

Ada dua operasi dasar yang bias kita lakukan pada sebuah tumpukan, yaitu Operasi menambahkan data, atau mem*push* data, dan operasi menghapus data atau mem*pop* data. Karena ke dalam tumpukan kita bias mem*push* data*,* maka tumpukan juga sering disebut dengan *pushdown list.*

Operasi Push

Sekarang kita akan menyusun sebuah prosedur untuk operasi push. Dengan menyajikan tumpukan seperti diatas, maka operasi push dengan mudah kita implementasikan sebagai berikut :

```
Procedure PUSH (varT :Tumpukan; X : integer) ;
Begin
        T.Atas := T.Atas + l;
```

```
        T.Isi [T.Atas] ;= x;
End;
```

Prosedur diatas akan menyiapkan tempat untuk x yang akan dipush ke dalam tumpukan, yaitu dengan menambah nilai medan T. Atas dengan l dan kemudian menyisipkan x ke dalam larik T.Isi.

Prosedur diatas belum cukup sempurna untuk melakukan operasi push, karena bila data yang dimasukkan (T.Atas) sama dengan Batas maximum (MaxElemen) dan kita akan mempush lagi maka akan terjadi *overlow* (melebihi batas maximum). Dengan demikian perlu ada penambahan untuk mengatasi overlow tersebut, agar prosedur selalu mengecek keadaan lokasi (T.Isi) sebelum suatu data dipush, apakah Max Elemen sudah tercapai atau belum, jika belum tercapai maka operasi push akan dilakukan, tetapi bila sudah mencapai Max Elemen operasi push akan ditolak.

Bentuk perubahan prosedurnya adalah :

```
Prosedure PUSH (varT :Tumpukan; X : integer) ;
Begin
  If T.Atas = MaxElemen then
Writeln ('TUMPUKAN SUDAH PENUH')
  Else
  Begin
     T.Atas := T.Atas + l;
     T.Isi [T.Atas] := X;
  End;
End;
```

Operasi Pop

Operasi Pop adalah operasi untuk menghapus elemen yang terletak pada posisi paling atas dari sebuah tumpukan. Sama halnya dengan operasi push, maka dengan deklarasi tumpukan seperti diatas, prosedur untuk operasi pop bias dengan mudah kita implementasikan sebagai berikut :

```
Procedure POP (varT :tumpukan) ;
Begin
```

```
        T.Atas := T.Atas – l;
End;
```

Prosedur diatas juga masih sangat sempurna, dan belum dapat digunakan untuk operasi pop, namun sudah mencakup pengertian dari pop. Hal initer jadi bila kita menggunakan prosedur diatas adalah bila data dalam lirik sudah kosong, karena adalah hal yang tidak mungkin melakukan operasi pop bila data sudah kosong.Maka perlu ada tambahan pada prosedur untuk mengecek keberadaan data dalam antrian, jika data masih ada maka dapat dilakukan operasi pop, namun bilasebaiknya akanditolak.
Penambahan dari prosedur pop adalah :

```
Procedure POP (varT :tumpukan) ;
Begin
        If T.Atas := 0 then
        Writeln ('TUMPUKAN SUDAH KOSONG')

        Else
        T.Atas := T.Atas – l ;
End;
```

Contoh Pemakaian Tumpukan Dalam Program

Untuk lebih memahami operasi yang terjadi pada tumpukan, berikut disajikan contoh program yang menggunakan metode tumpukan. Contoh Program dibawah mengelola tumpukan buku dengan atribut judul, pengarang dan harga. Adapun program dapat menangani penambahan buku ke stack (dengan syarat tumpukan belum penuh), mengambil buku dari tumpukan (dengan syarat tumpukan masih ada isinya), mencetak tumpukan dan mencari buku dalam tumpukan. Pelajarilah dan salinlah contoh program berikut untuk memahami stack:

```
program stack_SD_prak;
uses wincrt;
const max=10;
type stack=record
    judul,pengarang:array[1..max] of string;
    harga:array[1..max] of real;
```

```
    bawah,atas:0..max;
    end;
var buku:stack;
   i,j,n,pil:byte;
   jbaru,pbaru,jambil,pambil:string;
   hbaru:real;

function penuh(x:stack):boolean;
begin
if x.atas=max then penuh:=true else penuh:=false;
end;

function kosong(x:stack):boolean;
begin
if x.atas=0 then kosong:=true else kosong:=false;
end;

procedure push(var x:stack;var a,b:string;var c:real);
begin
    inc(x.atas);
    x.judul[x.atas]:=a;
    x.pengarang[x.atas]:=b;
    x.harga[x.atas]:=c;
    writeln('Buku yang ada di stack sekarang berjumlah ',x.atas,' buah');
end;

procedure pop(var x:stack);
begin
    jambil:=x.judul[x.atas];
    pambil:=x.pengarang[x.atas];
    writeln('Buku yang diambil adalah berjudul ',jambil,' pengarang ',pambil);
    dec(x.atas);
    writeln('Buku yang ada di stack sekarang berjumlah ',x.atas,' buah');
end;

procedure cetak_stack(var x:stack);
begin
writeln('Buku yang ada di tumpukan saat ini adalah sbb: ');
writeln('-----------------------------------------------------------');
writeln('Posisi ke | Judul      |   Pengarang      Harga');
writeln('-----------------------------------------------------------');
for i:=x.atas downto x.bawah do
writeln('|',i:4,'    |',x.judul[i]:15,'   |',x.pengarang[i]:15,'   ',
                'Rp ',x.harga[i]:6:2);
writeln('-----------------------------------------------------------');
end;
begin
buku.atas:=0;buku.bawah:=1;
repeat
```

```
begin
    clrscr;
    writeln(' MANIPULASI STACK BUKU');
    writeln('1. Tambah buku ke stack');
    writeln('2. Cetak buku dalam stack');
    writeln('3. Ambil buku dari stack');
    writeln('0. Keluar');
    write('Pilih 0-3: ');readln(pil);
    case pil of
    1: begin
      if penuh(buku) then writeln('Tumpukan sudah penuh, tidak bisa ditambah lagi')
      else
      begin
          writeln('Menambah buku ke dalam stack : ');
          write('Masukkan judul baru      : ');readln(jbaru);
          write('Masukkan pengarang       : ');readln(pbaru);
          write('Masukkan harganya        : ');readln(hbaru);
          push(buku,jbaru,pbaru,hbaru);
      end;
      end;
    2: if kosong(buku) then writeln('Belum ada buku di tumpukan')
      else cetak_stack(buku);
    3: if kosong(buku) then writeln('Stack sudah kosong, tidak bisa di pop')
      else pop(buku);
    0: writeln('Terimakasih') else writeln('Anda salah pilih menu');
     end;
     readln;
end
until(pil=6);
end.
```

SOAL MINGGU KE 6

Dari contoh soal di atas, buatlah program untuk mengelola tumpukan compact disk dengan atribut jenis cd (music, software,filem), judul dan tanggal_pembelian. Beri fasilitas untuk menambah data CD ke dalam tumpukan, mengambil CD dari tumpukan dan mencetaknya

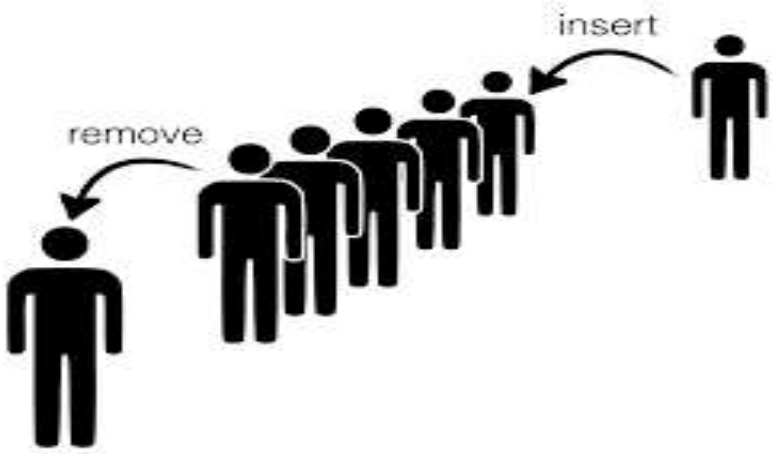
MODUL LIMA MINGGU KE ENAM
ANTRIAN (QUEUE)

Kompetensi mahasiswa : Mampu memahami, menerapkan dan mengendalikan penggunaan struktu antrian (queue). Membuat dan mendeklarasikan antrian, operasi push, pop, cetak dan pengecekan data dalam antrian

Antrian adalah suatu kumpulan data yang mana penambahan elemen hanya bisa dilakukan pada suatu ujung (belakang/rear), dan penghapusan dilakukan lewat ujung lainnya (depan/front). Sebagai contoh dapat kita ibaratkan dengan antrian membeli karcis di sebuah bioskop, para pengantri yang ingin membeli karcis hanya masuk ke antrian melalui satu pintu saja, demikian bagi yang telah selesai membeli karcis akan keluar satu pintu/ujung yang lain.

Seperti pada Stack/Tumpukan yang mengenal istilah masuk terakhir keluar pertama (LIFO – Last In First Out). Berbeda dengan antrian yang mengenal istilah masuk pertama keluar pertama (FIFO – First In First Out).

Ilustrasi Antrian :



Hal yang perlu diingat pada operasi Antrian adalah bahwa setiap penambahan data akan selalu dilakukan dari belakang, dan pengurangan/penghapusan data akan selalu dilakukan dari depan. Sehingga berdasarkan gambar bila ingin menambahkan suatu data harus diletakkan padasebelah kanan F, dan bila akan menghapus suatu data, maka akan dilakukan pada bagian sebelah kiri, dalam hal ini A.

Antrian dapat disajikan dalam bentuk larik/array. Berikut contoh pendeklarasian Antrian dengan array.

```
const max=10;
type antrian=record
        nama:array[1..max] of string;
        belanja:array[1..max] of longint;
        depan,belakang:0..max;
end;
var q:antrian;
```

Pada deklarasi diatas, elemen antrian dinyatakan dalam tipe strring dan longint. Variabel depan menunjukkan posisi elemen pertama dalam larik, variabel belakang menunjukkan posisi elemen terakhir dalam larik.

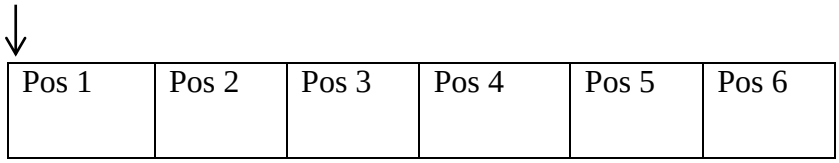
Dalam suatu permasalahan saat antrian sudah penuh, sementara kita masih ingin terus menambahan data lagi, maka akan terjadi offerflow (kelebihan kapasitas). Dengan mengabaikan akan terjadinya offerflow, maka penambahan data pada antrian dalam bentuk programnya adalah sebagai berikut :

belakang = belakang + 1 ;

antrian [belakang] : = x; (x: variabel untuk mengisi antrian)

Ilustrasi :

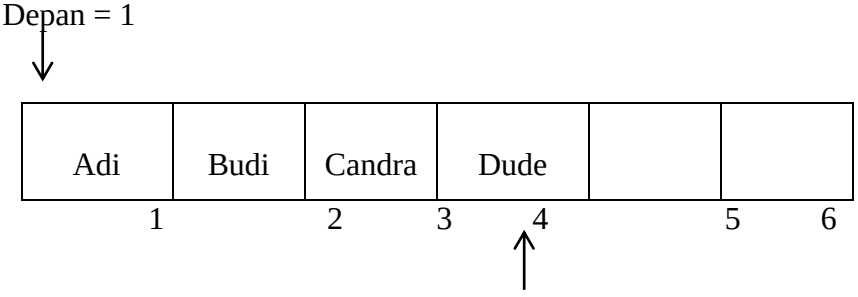
A. Depan = 1



belakang = 0

Program pada saat pertama kali dijalankan antrian masih dalam keadaan kosong, sehingga depan bernilai 1 dan belakang bernilai 0.

B. Depan = 1

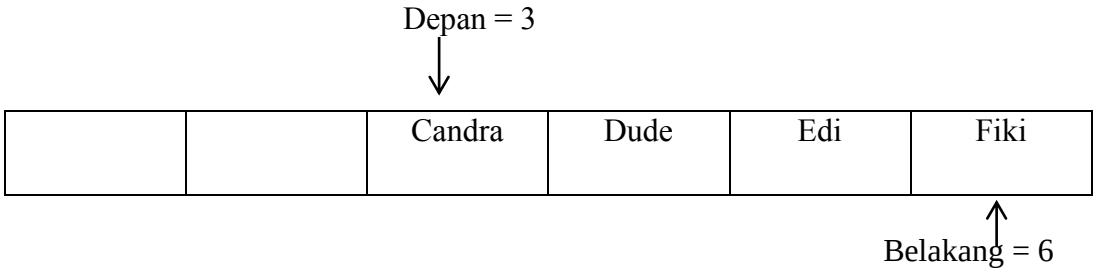


Belakang = 4

Pada saat dilakukan penambahan data maka belakang akan dijumlahkan dengan 1, kemudian antrian pada posisi belakang (antrian [belakamg]) akan diisi denfan elemen x.

Masih dengan mengabaikan akan terjadainya offerflow, untuk pengurangan /penghapusan pada antrian adalah sebagai berikut :

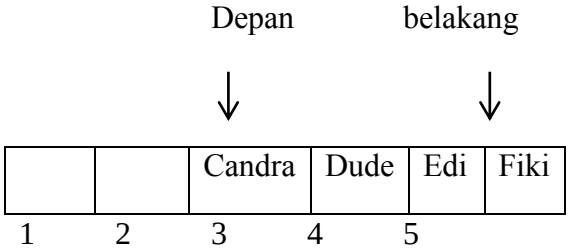
X := antrian [depan] ;
Depan := depan + 1 ;

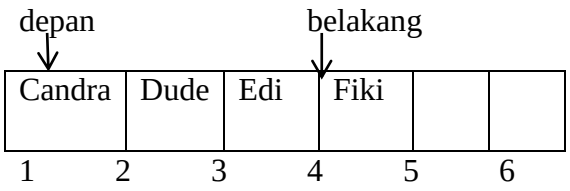


Pada saat dilakukan pengurangan data maka x akan diisi nilainya dengan kedudukan depan saat itu, kemudian depan dimajukan sebanyak 1 (depan + 1).

Dengan keadaan seperti diatas tentu saja dapat terjadi suatu keadaan dimana penambahan elemen baru tidak dapat dilakukan, misalnya karena nilai belakang sudah mencapai nilai maksimal akan tetapi antrian yang sesungguhnya masih banyak yang kosong karena nilai depan lebih besar dari 1, seperti pada gambar diatas.

Kemungkinan penyelesaian yang dapat dilakukan adalah dengan menggeser elemen-elemen lain maju kedepan jika ada elemen yang keluar (dihapus) seperti antrian manusia yang membeli karcis pada umumnya.

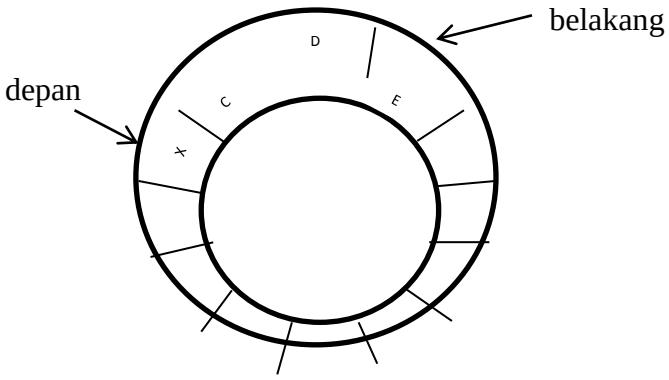




Akan tetapi hal ini akan sangat tidak efisien dilakukan pada elemen array, kalau

jumlah elemen array yang sdikit mungkin tidak terlalu jadi masalah, tetapi akan jadi masalah bila elemen array-nya sampai sebanyak seratus bahkan ribuan elemen, data akan dipindahkan satu-persatu sebanyak seribu elemen, hal ini akan memperlambat kinerja komputer.

Cara yang paling baik untuk memecahkan masalah ini adalah dengan menyimpan elemen larik seolah-olah larik berbentuk lingkaran.



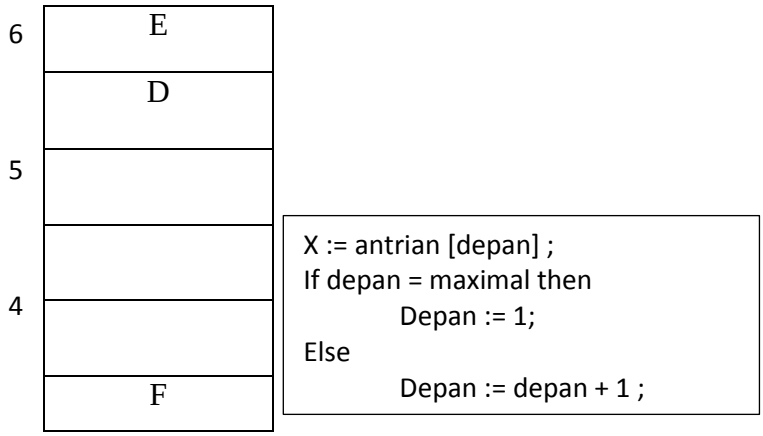
Elemen baru selalu dapat ditambahkan selama masih ada tempat kosong (terdapat tempat kosong antara belakang dan depan).

Pada kenyataan yang sebenarnya larik tidak benar-benar memutar melainkan masih seperti biasa yang memiliki ujung, namun perlakuannya di dalam penulisan progran dibuat seakan-akan melingkar/memutar.

Berikut adalah ilustrasi serta penulisan program untuk operasi pada antrian yang dibuat seakan-akan memutar :

6	E
	D
5	C
4	
	F

```
If belakang = maximal then
        Belakang := 1
Else
        Belakng := belakang + 1 ;
```



Contoh Program dan Soal:

Pada contoh berikut diperlihatkan suatu program antrian orang yang berbelanja di suatu supermarket. Antrean dilakukan di depan kasir saaat akan membayar. Pada saat membayar, maka total pembayaran akan disimpan dan nantinya akan dilaporkan sebagai rekap. Antrean yang digunakan masih sederhana.

```
uses wincrt;
const max=10;
type antrian=record
     nama:array[1..max] of string;
     belanja:array[1..max] of longint;
     depan,belakang:0..max;
end;
var q:antrian;
    masuk,keluar:string;
    bayarin,bayarout:longint;
    lagi:boolean;
    pilih:1..5;
    cacah:byte;hasil:real;

function penuh(q:antrian):boolean;
begin
if q.belakang=max then penuh:=true else penuh:=false;
end;

function kosong(q:antrian):boolean;
begin
if q.belakang=0 then kosong:=true else kosong:=false;
end;
```

```
procedure pushq(var q:antrian;m:string;b:longint);
begin
if penuh(q) then writeln('maaf antrian sedang penuh')
else
begin
    inc(q.belakang);
    q.nama[q.belakang]:=m;q.belanja[q.belakang]:=b;
end;
end;{end of procedure pushq}

procedure popq(s:antrian);
var i:byte;
begin
if kosong(q) then writeln('antrian sudah kosong')
else
begin
   keluar:=q.nama[q.depan];    {mengambil pengantri paling depan}
   bayarout:=q.belanja[q.depan];
   writeln('yang dilayani adalah Mr/Ms ',keluar,' dengan belanja Rp ',bayarout);
   inc(cacah);hasil:=hasil+bayarout;
   {pergeseran posisi pengantri ke kiri/depan}
   for i:=1 to q.belakang-1 do
   begin
       q.nama[i]:=q.nama[i+1];
       q.belanja[i]:=q.belanja[i+1];
   end;
   dec(q.belakang);    {mengurangi jumlah antrian}
end;
end;{end of procedure popq}

procedure cetak(var q:antrian);
var i:byte;
begin
if kosong(q) then writeln('daftar antrian kosong')
else
begin
writeln('DAFTAR PENGANTRI SAAT INI ==> ');
writeln('---------------------------------------');
writeln('posisi        nama         belanja');
writeln('---------------------------------------');
for i:=1 to q.belakang do
   writeln(i:4,q.nama[i]:15,'    ',q.belanja[i]:8);
end;
writeln('---------------------------------------');
end;{end of procedure cetak}

begin {program utama}
q.depan:=1;q.belakang:=0;
lagi:=true;
```

```
while lagi do
begin
    clrscr;
    writeln('ANTRIAN DI KASIR TOKO MAJU JAYA');
    writeln('1. Antrian masuk');
    writeln('2. Layanan kasir');
    writeln('3. Cetak antrian');
    writeln('4. Cek pendapatan sementara');
    writeln('5. Toko tutup dan tampilkan pendapatan hari itu');
    write('Pilih 1-5 --> ');readln(pilih);
case pilih of
1: begin
       writeln('nama baru ==> ');readln(masuk);
       writeln('jumlah belanja ==> ');readln(bayarin);
       pushq(q,masuk,bayarin);
  end;
2: popq(q);
3: cetak(q);
4: begin
  writeln('hasil sementara penjualan adalah Rp ',hasil:8:2);
  writeln('jumlah pengunjung sementara ',cacah,' orang');
  end;
5: begin
  writeln('Pendapatan total hari ini adalah Rp ',hasil:8:2);
  writeln('Pengunjung hari ini sebanyak ',cacah,' orang');
  lagi:=false;
  end;
end;
readln;
end;
end.
```

Tugas:

Buatlah program untuk menangani antrian dalam pendaftaran praktikum, dengan atribut nim,nama dan mata praktikum. Menu yang diminta adalah masuk antrian, layanan pendaftaran, cetak antrian dan cetak pendaftar yang sudah terlayani.

MODUL 7
TUMPUKAN DAN ANTRIAN

Kompetensi mahasiswa : Mampu memahami, menerapkan dan mengendalikan penggunaan struktur antrian (queue) dan tumpukan dalam suatu kasus yang lebih kompleks.

Dalam kasus yang lazim ditangani, maka ada kalanya kita harus menyelesaikan kasus antrian dan tumpukan yang tergabung menjadi satu. Missal dalam pengeloaan data pendaftar mahasiswa baru, maka pendaftar akan antre untuk membeli formulir atau mengisi formulir, kemudian formulir yang sudah diserahkan akan disimpan dalam tumpukan. Perhatikan contoh program berikut yang mengelola hal tersebut.

```
program pembelian_formulir;
uses crt;
const max=30;
type beli=record
     nomor:byte;
     nama:string;
     harga:longint;
     end;
     larik1=array[1..20] of beli;
type stack=record
     no:array[1..max] of byte;
     nama1,tgl :array[1..max] of string;
     depan,blkg:0..max;
     end;
type antrian=record
     nomer:array[1..max] of byte;
     front,tail:0..max;
     end;
var pembelian:larik1;
    formulir:stack;
    antri:antrian;
    pil: 1..7;
    nobaru,noambil,i,j,n,nn:byte;

function kosong(x:antrian):boolean;
begin
if x.tail=0 then kosong:=true else kosong:=false;
end;
function penuh(x:antrian):boolean;
```

```
begin
if x.tail=max then penuh:=true else penuh:=false;
end;

function empty(x:stack):boolean;
begin
if x.blkg=0 then empty:=true else empty:=false;
end;

function full(x:stack):boolean;
begin
if x.blkg=max then full:=true else full:=false;
end;


procedure beli_formulir(var x:larik1);
var nf:byte;
label ul;

begin
ul:
write('masukkan nomor formulir yang dibeli : ');readln(nf);
{cek}
for i:=1 to n do
begin
        if x[i].nomor=nf then
        begin
            writeln('Nomor sudah digunakan, ganti');
            goto ul;
        end;
end;
inc(n);
x[n].nomor:=nf;
write('masukkan nama pembeli : ');readln(x[n].nama);
x[n].harga:=75;
end;

procedure pushq(var x:antrian);
begin
    inc(x.tail); inc(nn);
    x.nomer[x.tail]:=nn;
    writeln('anda mendapat nomor urut antrian ',x.nomer[x.tail]);
end;

procedure layanan_antrian(var x:antrian);
var nf,pos:byte;
   ketemu,ada:boolean;
begin
ketemu:=false;
```

```
writeln('melayani pengembalian formulir');
writeln('----------------------------------');
noambil:=x.nomer[x.front];
{antrian digeser dulu}
for i:=1 to x.tail-1 do x.nomer[i]:=x.nomer[i+1];
dec(x.tail);
writeln('Sekarang melayani nomor urut antrian ',noambil);
{verifikasi nomor formulir, cek di pembelian}
write('masukkan nomor formulir yang akan dikembalikan : ');readln(nf);
for i:=1 to n do
begin if (pembelian[i].nomor=nf) then
     begin ketemu:=true;
          pos:=i;
     end;
end;
if ketemu then
begin
{Cek di stack}
    ada:=false;
    for j:=1 to formulir.blkg do begin if formulir.no[j]=nf then ada:=true;end;
    if ada then writeln('Formulir sudah pernah dikembalikan') else
    begin
        inc(formulir.blkg);
        formulir.no[formulir.blkg]:=nf;
        formulir.nama1[formulir.blkg]:=pembelian[pos].nama;
        writeln('Masukkan tanggal      : ');readln(formulir.tgl[formulir.blkg]);
    end;
end
else writeln('Nomor tersebut tidak ada di daftar formulir yang sudah dikeluarkan ');
end;

procedure cetak_form(var x:larik1);
begin
writeln('DAFTAR FORMULIR TERJUAL');
writeln;
writeln('--------------------------------------');
writeln('No   Nomor Formulir    Pembeli');
writeln('--------------------------------------');
for i:=1 to n do writeln(i:2,'   ',x[i].nomor:2,'     ',x[i].nama:10);
writeln('--------------------------------------');
writeln('Total formulir terjual ',n,' buah ');
writeln('Pendapatan adalah Rp ',n*75,'.000');
end;

procedure cetak_antrian(var x:antrian);
begin
writeln('DAFTAR FORMULIR TERJUAL');
writeln;
writeln('--------------------------------------');
```

```
writeln('Antrian ke    Nomor urut antrian');
writeln('-----------------------------------');
for i:=1 to antri.tail do writeln(i:4,'            ',x.nomer[i]:4);
writeln('-----------------------------------');
writeln('Jumlah yang sedang antri saat ini adalah ',x.tail);
end;

procedure cetak_stack(var x:stack);
begin
writeln('DAFTAR FORMULIR YANG SUDAH DIKEMBALIKAN');
writeln;
writeln('-------------------------------------------------');
writeln('No   Nomor Form       Pembeli     TANGGAL KEMBALI');
writeln('-------------------------------------------------');
for i:=x.blkg downto 1 do
writeln(i:2,'   ',x.no[i]:4,'      ',x.nama1[i]:10,'      ',x.tgl[i]:10);
writeln('-------------------------------------------------');
end;


begin{program utama}
antri.front:=1;antri.tail:=0; n:=0; formulir.depan:=1;formulir.blkg:=0;nn:=0;
repeat
begin
    clrscr;
    writeln('LAYANAN PEMBELIAN DAN PENGEMBALIAN FORMULIR
MABA');
    writeln('-----------------------------------------------');
    writeln('1. Beli formulir');
    writeln('2. Antri penyerahan');
    writeln('3. Layanan penyerahan formulir');
    writeln('4. Cetak  daftar pembelian formulir');
    writeln('5. cetak antrian yang akan menyerahkan formulir');
    writeln('6. Cetak tumpukan formulir yang sudah diserahkan');
    writeln('7. Selesai');
    write('pilih 1-7 : ');readln(pil);
    case pil of
    1: beli_formulir(pembelian);
    2: begin
      if penuh(antri) then writeln('antrian sedang penuh')
      else
         begin
            pushq(antri);
            writeln('anda berada di antrian nomor ',antri.tail);
         end;
      end;
    3: begin
      if kosong(Antri) then writeln('antrian sedang kosong')
      else
```

```
        layanan_antrian(antri);
        end;
      4: if n=0 then writeln('pembelian formulir belum ada') else
cetak_form(pembelian);
      5: if kosong(antri) then writeln('Antrian sedang kosong') else cetak_antrian(Antri);
      6: if formulir.blkg=0 then writeln('Belum ada yang menyerahkan formulir')
        else cetak_stack(formulir);
      7: writeln('Terimakasih');
      end;
      readln;
end
until (pil=7);
end.
```

Tugas:
Buatlah program untuk mengelola Dealer suatu kendaraan dengan ketentuan sbb:

1. Data kendaraan disimpan dengan atribut no_plat (unik),nama kendaraan dan harga
2. Data pembelian disimpan dengan atribut nama pembeli, no_plat kendaraan yang dibeli dan tanggal pembelian

Menu yang diinginkan adalah input data kendaraan, cetak kendaraan, antrean pembelian kendaraan, layanan pembelian (dari kendaraan yang telah diinputkan dan pembeli diambil dari antrian) dan cetak penjualan kendaraan

MODUL KE-8
POINTER DAN LINKED LIST

Kompetensi mahasiswa : Mampu memahami, menerapkan dan mengendalikan penggunaan pointer, pembuatan node, dan linked list sederhana untuk operasi standar : tambah awal, tambah akhir, tambah posisi tertentu, hapus awal, hapus akhir, hapus posisi tertentu dan cetak linked list

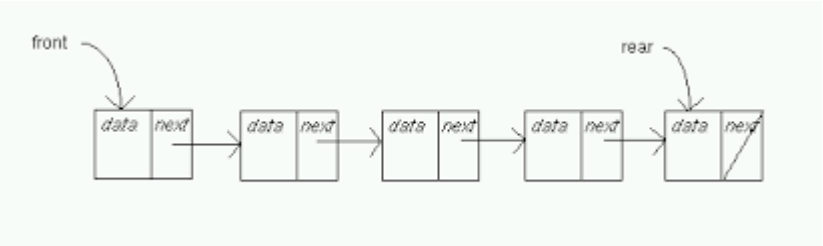
LINKED LIST (SENARAI BERANTAI)

A. Sigle Linked List

Salah satu struktur data dinamis yang paling sederhana adalah Senarai Berantai (Linked List),atau senarai satu arah. Senarai Berantai sendiri punya makna sebagai kumpulan komponen yang di susun secara berurutan dengan menggunakan bantuan pointer .Komponen-komponen yang tersusun tersebut akan di sebut sebagai simpul ,sehingga pada senarai akan terdapat banyak simpul, dan tiap simpulnya dapat dibagi menjadi dua bagian .Bagian pertama dari simpul disebut dengan **medan informasi,** yang berisi informasi /data yang dapat berupa record yang akan di olah.Sedangkan bagian yang kedua disebut sebagai medan penyambung (link field),yang berisi alamat simpul berikutnya.

Node

Untuk memudahkan setiap pembahasan selanjutnya ,maka akan kita tetapkan untuk sebelah kiri simpul sebagai medan informasi dan sebelah Kanan simpul disebut sebagai medsa penyambung. Perlu kita ketahui bahwa medan penyambung sebenarnya adalah suatu pointer yang akan menunjuk ke simpul berikutnya ,sehingga nilai dari medan ini dapat berupa alamat lokasi simpul berikitnya ataupun dapat bernilai *nil.* Perlu diingat bahwa setiap simpul terakhir dari linked list medan penyambung harus bernilai *nil.* Contoh dari senarai berantai dengan 4 simpul :



Tampak dari gambar bahwa medan penyambung menunjuk ke simpul berikutnya ,dan medan penyambung pada simpul terakhir akan bernilai nil karena medan penyambung tersebut tidak menunjuk kemampuan .Pada senarai berantai juga terdapat 2 buah pointer **awal** dan **akhir .** Kedua pointer ini berfungsi untuk mempermudah dalam melakukan operasi pengolahan data pada senarai berantai ,seperti : pembacaan ,pencarian, pengeditan, serta penghapusan data , pengertian kedua pointer ini mungkin akan lebih jelas pada pembahasan selanjutnya.

Bentuk umum pendeklarasian **single linked list :**

```
Type     pointer = ^nmrecord ;
         Nmrecord =recod;
         Nmvar 1 : tipe;
         next  : pointer ;  {penyambung ke node berikutnya}
         {Medan penghubung yang bertipe data pointer}
End ;
Var node :pointer
```

Penjelasan :

pointer : sebuah pointer yang menunjukan ke sebuah record.

Nmrecord : nama record yang berisi variabel penyimpanan data dan berisi medan penyambung.

Nmvar1 : variabel yang akan berfungsi sebagai penyimpanan data .

next : variabel yang berfungsi sebagai media penyambung.

Contoh penulisan deklarasi Single Linked List :

```
type pointer1=^recpointer;
        recpointer=record
```

```
        nama:string;
        kait:pointer1;
        end;
var awal,akhir:pointer1;
```

Contoh deklarasi:

```
   Program SinleLinkedList ;
Uses winert ;
Type simpul = ^data ;
        Data = record
Nama : string {25}                          Deklarasi simpul
        Kait   : simpul
        End ;
Var awal ,akhir,bantu : simpul ;
        ya : char ;
begin
        clrscr ;
        awal := nil ;                       Pembuatan nilai awal = nil pada saat
        program pertama                                 dijalankan
ya : = ‘ y ‘ ;
while ya in( ‘ y ‘ , ‘Y ‘) do
begin
        new (bantu) ;
        write (‘Masukan Nama : ‘ ) ;
        readln (bantu^ .nama ) ;
        write (‘ Tambah data lagi { Y/N } : ? ‘) ;
```

proses 1 : Proses untuk memberikan nilai pada variabel awal dan akhir saat pertama penginputan data. Pertama penginputan data. Tampak bahwa medan penyambung dari bantu (bantu^ .kait) dibuat nilainya nil karena data masih satu sehingga berperan sebagai awal dan akhir linked list.

Proses 2: Proses untuk memberikan nilai pada akhir untuk menunjuk ke bantu yang baru saja diinputkan.

Perlu diingat bahwa dalam linked list terdapat peraturan-peraturan sebagai berikut :

1. Awal akan selalu menunjuk ke awal linked list, dan akhir akan selalu menunjuk ke akhir ke linkd list, sekalipun penambahan data di lakukan .
2. Setiap simpul yang terkhir dari linked list nilai dari medan penyambung harus slalu bernilai nil .

Proses 3 : Pada proses ini bantu akan membaca simpul dan menampilkan hasilnya dari awal sampai akhir linked list .

Operasi pada Senarai Berantai (Single Linked List)

1. Mencetak Linked List

Mencetak linked list tidak sama dengan mencetak larik biasa. Pencetakan dapat dilakukan dengan menempatkan pointer bantu di posisi awal dan mencetak isinya. Selanjutnya pointer bantu digeser ke bantu^.next dan dicetak lagi, sampai pointer menunjuk nil (bantu=nil). Syarat pencetakan adalah linker list tidak kosong (x<>nil)/. Adapun procedurenya adalah

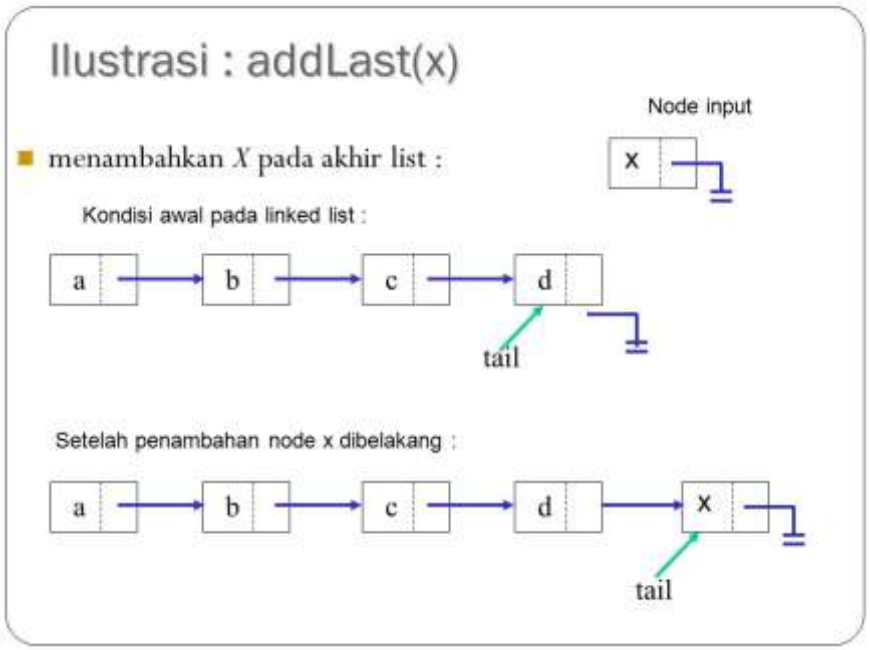
```
procedure cetak_list(var L:pointer);

begin
bantu:=L;i:=0;
writeln('-----------------------------------------');
writeln('Node ke     Nama          umur');
writeln('-----------------------------------------');
while bantu<>nil do
begin
   inc(i);
   writeln(i:4,'    ',bantu^.nama:10,'       ',bantu^.umur:4);
   bantu:=bantu^.next;
end;
writeln('-----------------------------------------');
writeln('jumlah node saat ini ',i,' buah');
end;
```

Pada procedure diatas dapat dilihat bahwa pencetakan dilakukan melalui pointer bantu yang ditempatkan pada awal (x). Procedure dipanggil dengan cetak_list(awal);

2. Menambah simpul di belakang

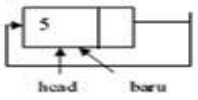
Menambah simpul di belakang dapat dilihat pada ilustrasi berikut:



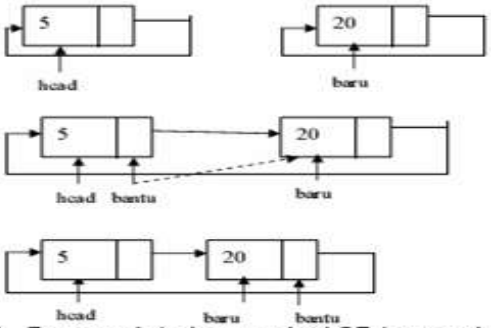
Adapun procedure untuk menambah data di belakang adalah sebagai berikut:

```
procedure tambah_akhir(var X,Y:pointer);
var baru:pointer;
begin
new(baru);
writeln('Menambah node ke larik');
write('masukkan nama yang baru : ');readln(baru^.nama);
baru^.next:=nil;
if X=nil then begin X:=baru;Y:=baru;end
else
begin Y^.next:=baru; Y:=baru;end;
writeln('Node sudah ditambahkan');
end;
```

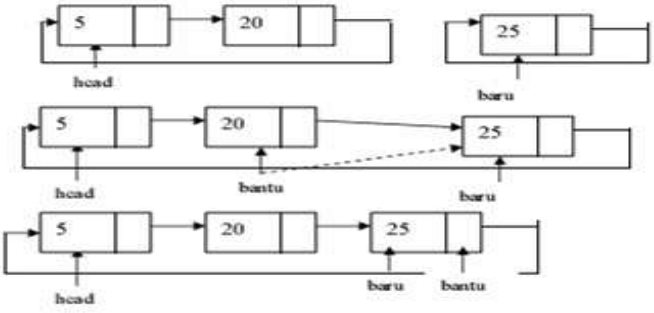
Procedure di atas dipanggil dengan perintah *tambah_akhir(awal,akhir)*; Dari procedure di atas dapat dilihat bahwa node yang akan ditambahkan diberi nama ‘baru’. Setelah node diisi, maka akan dicek apabila linked list masih kosong maka baru akan menjadi awal sekaligus akhir dari linked list. Akan tetapi jika linked list sudah ada maka node baru akan disambungkan ke posisi terakhir linked list dengan menghubunkan kait node terakhir (y^.next) ke node baru dan memindahkan node teakhir (y) ke baru.



3. Datang data baru, misalnya 20 (penambahan di belakang)

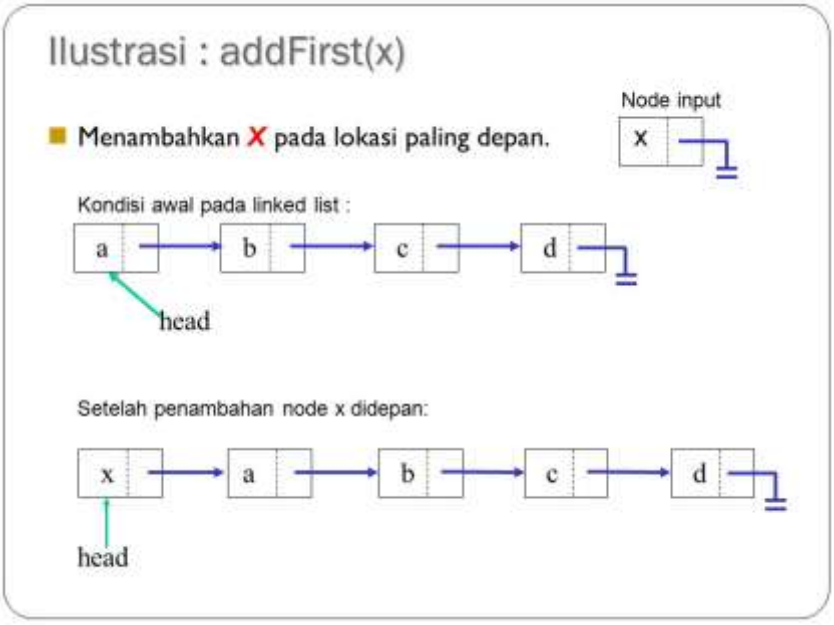


4. Datang data baru, misal 25 (penambahan di belakang)



//

3. Menambah simpul di depan



Menambah simpul di depan adalah menambahkan node ‘baru’ di posisi pertama, artinya di posisi sebelum node ‘awal’. Hal ini dapat dilakukan dilakukan dengan membuat dan mengisi node baru, kemudian dicek apabila linked list belum terbentuk maka node ‘baru’ akan menjadi awal dan akhir dari linked list, namun apabila node sudah ada maka pengait

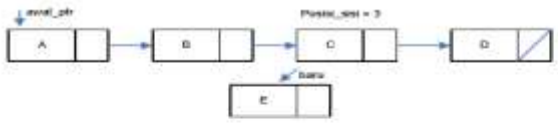
node baru (baru^.next) digubungkan dengan awal (x), dan memindahkan awal ke baru (x:=baru). Prosedur untuk proses penambahan awal adalah sbb:

```
procedure tambah_awal(var X,Y:pointer);
var baru:pointer;
begin
new(baru);
writeln('Menambah node ke larik');
write('masukkan nama yang baru : ');readln(baru^.nama);
baru^.next:=nil;
if X=nil then begin X:=baru;Y:=baru;end {jika belum ada node sebelumnya}
else
begin baru^.next:=X; X:=baru;end; {diletakkan di posisi pertama}
writeln('Node sudah ditambahkan');
end;
```

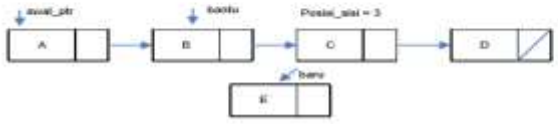
Procedure di atas dipanggil dengan tambah_awal(awal,akhir)

4. Menambah simpul di tengah

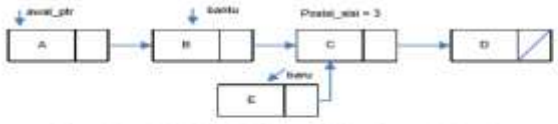
Ilustrasi :



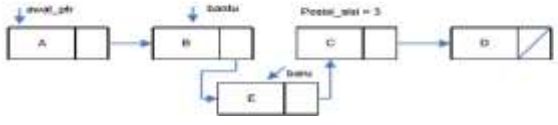
Langkah selanjutnya cari posisi elemen sebelum posisi sisip kemudian simpan dalam suatu variabel dengan nama bantu.



Kemudian sambungkan field next dari Baru ke posisi next dari bantu



Kemudian pindahkan field next dari bantu ke posisi data baru.



Menambah simpul (node) ditengah adalah penambahan berdasar kriteria atau nomor. Penyambungan node ini agak rumit. Pertama buat node ‘baru’ dan diisi, kemudian harus ditemukan posisi node yang berada tepat didepan node yang akan disisipkan, kemudian setelah ditemukan beri nama bantu. Kemudian hubungkan kait node baru ke node yang

ditunjuk oleh node bantu (baru^.next:=bantu^.next), setelah itu maka hubungkan kait node bantu ke node baru (supaya node tidak terlepas). Adapun procedure untuk proses penambahan node di tengah adalah sebagai berikut:

```
procedure tambah_nama(var X,Y:pointer);
var baru,bantu:pointer;
    ada1:boolean; sisip:string;
    label ulang;
begin
ada1:=false;ada:=false;
writeln('Menyisipkan nama sesudah nama tertentu');
write('Node baru akan diletakkan sesudah siapa ? ');readln(sisip);
if (Y^.nama=sisip) then begin ada1:=true;tambah_akhir(X,Y);end
else
begin
      {validasi}
  new(bantu); bantu:=X;
  ulang:
  write('masukkan nama baru yang akan disisipkan : '); readln(data_baru);
  validasi(X,Y,data_baru);
        if ada then begin writeln('data sudah digunakan, ulangi');goto ulang;end;
  bantu:=X;
  while bantu<>nil do
  begin
          if bantu^.nama=sisip then
          begin
                  new(baru); ada1:=true;
                  baru^.nama:=data_baru;baru^.next:=nil;
                  baru^.next:=bantu^.next;
                  bantu^.next:=baru;
                  inc(jum_nod);
                  writeln('node sudah disisipkan sesudah ',sisip);
          end;
          bantu:=bantu^.next;
          end;
end; {end else}
if not ada1 then writeln('nama ',sisip,' tidak ditemukan, penyisipan gagal');
end;
```

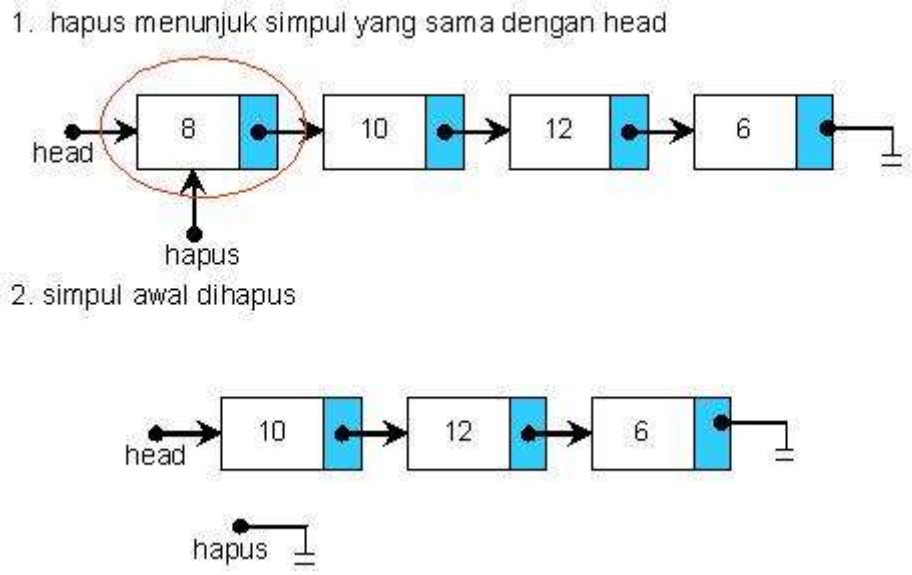
Procedure di atas dapat dipanggil dengan tambah_nama(awal,akhir

5. Menghapus simpul awal

Menghapus simpul (node) diawal adalah menghapus node posisi pertama. Caranya adalah node pertama (awal) diberi tanda dengan nama node hapus (hapus:=x), setelah itu maka penunjuk node pertama (awal) dipindahkan ke node sesudah awal yaitu node yang

ditunjuk oleh kait awal (awal:=awal^.next). Selanjutnya node hapus didispose sehingga terhapus dari memori. Ilustrasinya adalah seperti gambar di bawah:

Ilustrasi :



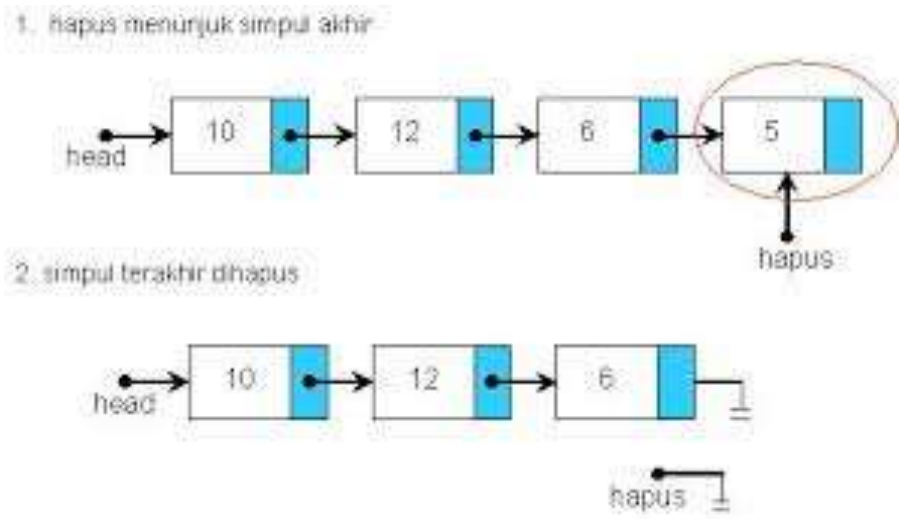
Adapun procedure untuk melakukan hapus awal adalah sebagai berikut:

```
procedure hapus_awal(var X,Y:pointer);
var hapus:pointer; ya:char;
begin
writeln('Menghapus node di posisi terdepan');
writeln('Data di posisi terdepan adalah ',X^.nama);
write('apakah yakin akan dihapus<y/t> ? ');readln(ya);
if (ya='y') or (ya='Y') then
begin
                new(hapus);hapus:=X;
                X:=hapus^.next;
                writeln('Data telah dihapus'); dec(jum_nod);
                dispose(hapus);
end
else writeln('data tidak jadi dihapus');
end;
```

Procedure di atas dipanggil dengan if awal<>nil then hapus_Awal(awal,akhir) dengan asumsi jika awal=nil maka node tidak ada isinya. Pada procedure di atas dibuat validasi penghapusan dengan variable ya.

6. Menghapus di akhir linked list

Menghapus node pada posisi terakhir dari linked list adalah menghapus node pada pointer akhir. Hal ini mudah dilakukan dengan menangkap pointer terakhir ke dalam pointer hapus. Persoalannya adalah, kita harus menandai pointer sebelum pointer terakhir karena pointer ini nantinya akan menjadi pointer terakhir setelah penghapusan. Caranya adalah, tempatkan pointer bantu pada posisi pertama (x) kemudian cek apakah kait dari node yang ditunjuk oleh bantu sudah bernilai nil (posisi terakhir).Jika belum maka letakkan bantu ke pointer yang ditunjuk oleh pointer bantu. Jika sudah memenuhi maka pointer bantu akan berada tepat di depan pointer terakhir. Setelahnya anda bisa menanghap pointer hapus dan mengubah bantu menjadi akhir.



Adapun procedure untuk menghapus node terakhir adalah sbb:

```
procedure hapus_akhir(var X,Y:pointer);
var bantu,hapus:pointer;ya:char;

begin
writeln('Menghapus node di posisi terakhir');
{mencari posisi node terakhir}
if X<>Y then
begin
new(bantu);bantu:=X; new(hapus);
while (bantu^.next)^.next <>nil do bantu:=bantu^.next;
end;
{posisi node terakhir sudah ketemu}
writeln('Data di posisi terakhir adalah : ',Y^.nama);
write('yakin akan dihapus <y/t> ? ');readln(ya);
if (ya='y') or (ya='Y') then
begin
                if X<>Y then begin hapus:=bantu^.next;Y:=bantu;Y^.next:=nil;end
                else begin hapus:=X;X:=nil;end;
```

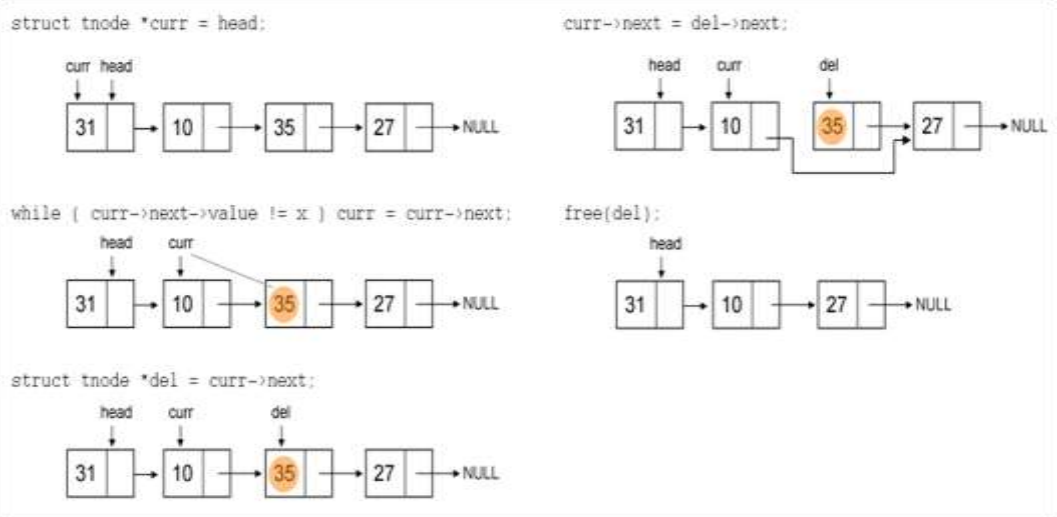
```
                dec(jum_nod); dispose(hapus);
                writeln('Data terakhir sudah dihapus');
    end
    else writeln('Data tidak jadi dihapus');
    end;
```

Procedure di atas dipanggil dengan if awal<>nil then hapus_akhir(awal,akhir) dengan asumsi jika awal=nil maka node tidak ada isinya. Pada procedure di atas dibuat validasi penghapusan dengan variable ya.

7. Menghapus simpul di tengah.

Menghapus simpul di tengah menggunakan logika yang mirip dengan menyisipkan data di tengah, hanya dibalik logikanya. Pertama tentukan dulu kriteria penghapusan, missal nomor node atau kriteria tertentu. Selanjutnya tempatkan pointer bantu di awal linked list dan cek apakah isi node di belakang node bantu sesuai kriteria yang dimaksud. Node bantu akan ditempatkan tepat sebelum node yang akan dihapus. Jika sudah ketemu, maka tangkap node hapus adalah node sesudah bantu (hapus:=bantu^.next). Selanjutnya hubungkan kait node bantu ke node susudah hapus (bantu^.next:=hapus^.next), sehingga node hapus tidak lagi menjadi bagian dari linked list. Syarat penghapusan adalah linked list tidak kosong (awal<>nil).

Ilustrasi :



Bentuk penulisan dalam bentuk program adalah seperti procedure berikut:

```
procedure hapus_nama(var L:pointer;var x:string);
var hapus:pointer;
   ada:boolean;

begin
ada:=false; new(bantu);
if L^.nama=x then begin ada:=true;hapus_awal(L);end
else
begin
    new(bantu);bantu:=L;
    while (bantu^.next <> nil) do
    begin
        if (bantu^.next)^.nama=x then
        begin
            ada:=true;
            new(hapus);hapus:=bantu^.next;
            writeln('Telah dilakukan penghapusan terhadap ',x);
            bantu^.next:=hapus^.next;
            dispose(hapus);
        end;
        bantu:=bantu^.next;
    end;
end;
if not ada then writeln('Nama ',x,' tidak ditemukan');
end;
```

Procedure di atas dipanggil dengan if awal<>nil then hapus_akhir(awal,akhir) dengan asumsi jika awal=nil maka node tidak ada isinya.

Latihan:

Berikut adalah contoh program yang bisa anda coba:

```
program linked_list_kelasA;
uses wincrt;
type pointer=^datapointer;
    datapointer=record
    nama:string;umur:byte;
    next:pointer;
    end;
var awal,akhir,bantu:pointer;
   pil,i:byte;
   namabr,sisip,nmhapus:string;umurbr:byte;

procedure tambah_depan(var L:pointer;var x:string;var y:byte);
var baru:pointer;
```

```
begin
new(baru);
baru^.nama:=x;baru^.umur:=y;baru^.next:=nil;
if L=nil then      {belum ada list sebelumnya}
begin
    L:=baru;akhir:=baru;
end
else {taruh list baru di depan list yang sebelumnya}
begin
    baru^.next:=L;
    L:=baru;
end;
end;

procedure tambah_akhir(var L:pointer; var x:string;var y:byte);
var baru:pointer;

begin
new(baru);
baru^.nama:=x;baru^.umur:=y;baru^.next:=nil;
if L=nil then       {belum ada list sebelumnya}
begin
    awal:=baru;L:=baru;
end
else
begin
    L^.next:=baru;  {taruh list sesudah list terakhir}
    L:=baru;
end;
end;

procedure cetak_list(var L:pointer);

begin
bantu:=L;i:=0;
writeln('-----------------------------------------');
writeln('Node ke      Nama          umur');
writeln('-----------------------------------------');
while bantu<>nil do
begin
    inc(i);
    writeln(i:4,'    ',bantu^.nama:10,'        ',bantu^.umur:4);
    bantu:=bantu^.next;
end;
writeln('-----------------------------------------');
writeln('jumlah node saat ini ',i,' buah');
end;
```

```
procedure tambah_tengah(var L:pointer;var x,y:string;var z:byte);
var baru:pointer;
   ada:boolean;

begin
bantu:=L; ada:=false;
if akhir^.nama=y then {nama disisipkan sesudah nama terakhir}
begin
    new(baru);ada:=true;
    baru^.nama:=x;baru^.umur:=z;baru^.next:=nil;
    akhir^.next:=baru;
    akhir:=baru;
end
else
begin
    while bantu^.next<>nil do
    begin
    if (bantu^.nama=y) then
    begin
        new(baru); ada:=true;
        baru^.nama:=x;baru^.umur:=z;baru^.next:=nil;
        baru^.next:=bantu^.next;
        bantu^.next:=baru;
    end;
    bantu:=bantu^.next;
    end;
end;
if not ada then writeln('nama tidak bisa disisipkan,karena tidak ada nama ',y,' di LIST');
end;

procedure hapus_awal(var L:pointer);
var hapus:pointer;

begin
    new(hapus);
    hapus:=L;
    writeln('Telah dilakukan penghapusan di awal node');
    writeln('Yang dihapus adalah : ',hapus^.nama,' umur ',hapus^.umur,' th');
    L:=hapus^.next;
    dispose(hapus);
end;

procedure hapus_akhir(var L:pointer);
var hapus:pointer;
begin
new(bantu);bantu:=L;
while (bantu^.next)^.next <> nil do bantu:=bantu^.next;
new(hapus);
hapus:=bantu^.next;
```

```
writeln('Telah dilakukan penghapusan di akhir node');
writeln('Yang dihapus adalah : ',hapus^.nama,' umur ',hapus^.umur,' th');
akhir:=bantu;
akhir^.next:=nil;
dispose(hapus);
end;

procedure hapus_nama(var L:pointer;var x:string);
var hapus:pointer;
   ada:boolean;

begin
ada:=false; new(bantu);
if L^.nama=x then begin ada:=true;hapus_awal(L);end
else
begin
    new(bantu);bantu:=L;
    while (bantu^.next <> nil) do
    begin
        if (bantu^.next)^.nama=x then
        begin
            ada:=true;
            new(hapus);hapus:=bantu^.next;
            writeln('Telah dilakukan penghapusan terhadap ',x);
            bantu^.next:=hapus^.next;
            dispose(hapus);
        end;
        bantu:=bantu^.next;
    end;
end;
if not ada then writeln('Nama ',x,' tidak ditemukan');
end;


begin {program utama}
new(awal);awal:=nil;new(akhir);akhir:=nil;
repeat
begin
     clrscr;
     writeln('Operasi sederhana pada linked list');
     writeln('1. Menambah Node di depan');
     writeln('2. Menambah di akhir node');
     writeln('3. Menambah di posisi tertentu');
     writeln('4. Menampilkan linked list');
     writeln('5. Menghapus list terdepan');
     writeln('6. Menghapus list terakhir');
     writeln('7. Menghapus node tertentu');
     writeln('0. Keluar');
     write('Pilih 0-7   : ');readln(pil);
```

```
    case pil of
    1:begin
          writeln('Tambah data di depan');
          write('Masukkan nama baru       : ');readln(namabr);
          write('masukkan umurnya         : ');readln(umurbr);
          tambah_depan(awal,namabr,umurbr);
      end;
    2: begin
          writeln('Tambah data di akhir');
          write('Masukkan nama baru       : ');readln(namabr);
          write('masukkan umurnya         : ');readln(umurbr);
          tambah_akhir(akhir,namabr,umurbr);
      end;
    3: begin
          writeln('Tambah data di posisi yang diinginkan');
          write('Masukkan nama baru       : ');readln(namabr);
          write('masukkan umurnya         : ');readln(umurbr);
          cetak_list(awal);
          write('Disisipkan sesudah siapa ? ');readln(sisip);
          tambah_tengah(awal,namabr,sisip,umurbr);
      end;
    4: if awal=nil then writeln('Linked list masih kosong') else cetak_list(awal);
    5: if awal=nil then writeln('Tidak ada linked list') else
       begin hapus_awal(awal);cetak_list(Awal);end;
    6: if awal=nil then writeln('Linked list masih kosong') else
       begin hapus_akhir(awal);cetak_list(awal);end;
    7: if awal=nil then writeln('Linked list masih kosong') else
       begin
          writeln('Mengapus node tertentu');
          writeln('ini adalah daftar linked list yang ada');
          cetak_list(awal);
          write('masukkan nama yang akan dihapus : ');readln(nmhapus);
          hapus_nama(awal,nmhapus);
          writeln('Setelah proses penghapusan');
          cetak_list(awal);
       end;
    0: writeln('Terimakasih');
    end;
    readln;
end
until(pil=0);
end.
```

Tugas

1. Buatlah program operasi linked list barang tambah depan, belakang, dan cetak dengan atribut kode_brg, nama_barang dan harga. Buat validasi kode_brg tidak boleh sama.

2. Lanjutkan program sebelumnya dengan menambah operasi hapus depan, belakang dan hapus barang tertentu (berdasar kode) dan operasi pencarian barang tertentu (nama).

MODUL KE-9
ANTRIAN DAN TUMPUKAN DENGAN LINKED LIST

Kompetensi mahasiswa : Mampu memahami, menerapkan dan mengendalikan penggunaan pointer dalam linked list untuk menyelesaikan persoalan pemrograman yang menggunakan konsep struktur data tumpukan dan antrian

A. Pendahuluan

Dalam kasus nyata, penggunaan konsep linked list (array dinamis) lebih banyak digunakan. Misal dalam kasus pelayanan pelanggan dan pengelolaan data di SPBU. Dalam antriannya, SPBU tidak mengenal pembatasan jumlah antrian, sehingga penggunaan array dinamis jauh lebih efektif dibanding penggunaan array statis. Kemudian pada transaksi pembelian BBM juga jumlahnya tak tentu, sehingga penggunaan linked list juga akan menambah efisiensi pennggunaan data. Atau dalam kasus antrian dalam bioskop yang mirip dengan kasus SPBU.

B. Program Latihan

Sebagai contoh, program pengelolaan bioskop dapat dijadikan contoh kasus antrian dinamis. Program ini menangani layanan antrian pembelian tiket bioskop dan menyimpan hasil pembelian (histori) dalam linked list. Linked list digunakan dalam antrian dan penyimpanan histori transaksi.

```
program bioskop_w13_F;
uses crt;
type pointer1=^rec_antri;
    rec_antri=record
    no_antrian:byte;
    next1:pointer1;
    end;
type pointer2=^rec_tiket;
     rec_tiket=record
               no_antri:byte; teater1:string; j_tiket:byte;subtot:longint;
               next2:pointer2;
    end;
type rec_bioskop=record
     teater,judul:string;stok,terjual,sisa:byte;
```

```
      harga:longint;
      end;
      larik_bioskop=array[1..4] of rec_bioskop;
var awal1,akhir1:pointer1;
    awal2,akhir2:pointer2;
    bioskop:larik_bioskop;
    n,i,pil:byte;

procedure isi_bioskop(var x:larik_bioskop);
begin
writeln('Mengisi Bioskop');
x[1].teater:='teater 1';x[1].judul:='Star Wars';
x[1].stok:=20;x[1].harga:=20000;x[1].terjual:=0;
x[1].sisa:=x[1].stok-x[1].terjual;
x[2].teater:='teater 2';x[2].judul:='Susah Sinyal';
x[2].stok:=20;x[2].harga:=25000;x[2].terjual:=0;
x[2].sisa:=x[2].stok-x[2].terjual;
x[3].teater:='teater 3';x[3].judul:='Ayat-Ayat Cinta';
x[3].stok:=20;x[3].harga:=15000;x[3].terjual:=0;
x[3].sisa:=x[3].stok-x[3].terjual;
x[4].teater:='teater 4';x[4].judul:='Anak Langit';
x[4].stok:=20;x[4].harga:=10000;x[4].terjual:=0;
x[4].sisa:=x[4].stok-x[4].terjual;
end;

function habis(x:larik_bioskop):boolean;
begin
if (x[1].sisa=0) and (x[2].sisa=0) and (x[3].sisa=0) and (x[4].sisa=0) then
habis:=true else habis:=false;
end;

procedure masuk_antrian(var x,y:pointer1);
var baru:pointer1;
begin
new(baru);inc(n);baru^.no_antrian:=n;baru^.next1:=nil;
writeln('Selamat datang di antrian bioskop XX4');
writeln('anda mendapat nomor antrian ',n,' mohon tunggu sebentar');
if x=nil then begin x:=baru;y:=baru;end
else begin y^.next1:=baru;y:=baru;end;
end;

procedure cetak_antrian(var x,y:pointer1);
var bantu:pointer1;
begin
writeln('Daftar antrian bioskop XX4');
writeln('----------------------------');
writeln('posisi     Nomor antrian');
writeln('----------------------------');
new(bantu);bantu:=x;i:=0;
```

```
while bantu<>nil do
begin
    inc(i);  writeln(i:4,'     ',bantu^.no_antrian:4);
    bantu:=bantu^.next1;
end;
writeln('----------------------------');
writeln('Saat ini terdapat ',i,' pengantre');
writeln('----------------------------');
end;

procedure cetak_bioskop(var x:larik_bioskop);
var tot:longint;
begin
tot:=0;
writeln(' REKAP PENJUALAN TIKET BIOSKOP XX4');
writeln('-----------------------------------------------------------------');
writeln('No  Teater  Judul Film            Harga  Stok  Terjual  Sisa');
writeln('-----------------------------------------------------------------');
for i:=1 to 4 do
begin tot:=tot+(x[i].harga*x[i].terjual);
      writeln(i:2,'  ',x[i].teater:8,'  ',x[i].judul:15,'   ',x[i].harga:8,'   ',
      x[i].stok:3,'   ',x[i].terjual:3,'   ',x[i].sisa:3);
end;
writeln('-----------------------------------------------------------------');
writeln('Total pendapatan hari ini Rp ',tot:10);
writeln('-----------------------------------------------------------------');
end;

procedure layanan(var x,y:pointer1;var x1,y1:pointer2;var z:larik_bioskop);
var pil1,j:byte;
    baru:pointer2;
    hapus:pointer1;
label ulang;
begin
writeln('Layanan Penjualan Bioskop');
new(hapus);hapus:=x;
writeln('antrian nomor ',hapus^.no_antrian,' silahkan menuju konter tiket');
X:=X^.next1;
cetak_bioskop(z);
ulang:
write('masukkan nomor teater yang akan dipilih : ');readln(pil1);
if (pil1<1) or (pil1>4) then
    begin writeln('Nomor salah, ulangi');goto ulang;end;
if z[pil1].sisa=0 then writeln('Maaf tiket habis ') else
begin
    writeln('anda memilih teatre ',z[pil1].teater);
    write('masukkan jumlah tiket yang dibeli : ');readln(j);
    if j>z[pil1].sisa then
    writeln('Stok tidak cukup')
```

```
    else
            begin
                    {stok cukup}
                    z[pil1].terjual:=z[pil1].terjual+j;
                    z[pil1].sisa:=z[pil1].stok-z[pil1].terjual;
                    {catat di transaksi}
                    new(baru);
                    baru^.no_antri:=hapus^.no_antrian;
                    baru^.teater1:=z[pil1].teater;
                    baru^.j_tiket:=j; baru^.subtot:=j*z[pil1].harga;
                    baru^.next2:=nil;
                    if x1=nil then begin x1:=baru;y1:=baru;end
                    else begin y1^.next2:=baru;y1:=baru;end;
                    writeln('Layanan pembelian tiket sudah dilakukan dan
dicatat');
                    dispose(hapus);
            end;
end;
end;

procedure cetak_transaksi(var x,y:pointer2);
var bantu:pointer2;
begin
writeln('Daftar transaksi pembelian tiket bioskop XX4');
writeln('-----------------------------------------------');
writeln('No Nomor antrian   Teater    Qtt    Subtotal');
writeln('-----------------------------------------------');
new(bantu);bantu:=x;i:=0;
while bantu<>nil do
begin
    inc(i);  writeln(i:4,'   ',bantu^.no_antri:4,'   ',bantu^.teater1:8,'  ',
    bantu^.j_tiket:4,'  ',bantu^.subtot:8);
    bantu:=bantu^.next2;
end;
writeln('-----------------------------------------------');
writeln('Saat ini terdapat ',i,' transaksi');
writeln('-----------------------------------------------');
end;


begin
isi_bioskop(bioskop);
new(awal1);new(akhir1);new(awal2);new(akhir2);
awal1:=nil;akhir1:=nil;awal2:=nil;akhir2:=nil;n:=100;
repeat
    clrscr;
    writeln('PENGELOLAAN PEMBELIAN TIKET BIOSKOP XX4');
    writeln('1. masuk antrian');
    writeln('2. Cetak antrian');
```

```
        writeln('3. Layanan pembelian tiket');
        writeln('4. Cetak rekap penjualan');
        writeln('5. cetak detil transaksi');
        writeln('0. Selesai');
        write('pilih menu : ');readln(pil);
        case pil of
        1: if habis(bioskop) then writeln('Maaf semua tiket habis')
                else masuk_antrian(awal1,akhir1);
        2: if awal1=nil then writeln('Belum ada antrian tiket') else
                cetak_antrian(awal1,akhir1);
        3: if awal1=nil then writeln('Tidak ada antrian') else
                layanan(awal1,akhir1,awal2,akhir2,bioskop);
        4: cetak_bioskop(bioskop);
        5: if awal2=nil then writeln('Belum ada transaksi hari ini')
        else cetak_transaksi(awal2,akhir2);
        0: writeln('Terimakasih') else writeln('Anda salah pilih menu');
        end;
        readln;
    until pil=0;
    end.
```

Sebagai contoh, program pengelolaan rak buku dapat dijadikan contoh kasus tumpukan dinamis.

```
program stack_week10;
uses crt;
type pointer=^rec_buku;
  rec_buku=record
  judul,pengarang:string;
  next:pointer;
   end;
var bawah,atas:pointer;
  pil,n:byte;

procedure tambah_buku(var X,Y:pointer);
var baru:pointer;
begin
writeln('Menambahkan buku baru ke tumpukan');
new(baru);
write('masukkan judul buku baru                  : ');readln(baru^.judul);
write('masukkan pengarang                        : ');readln(baru^.pengarang);
baru^.next:=nil;
if x=nil then begin x:=baru;y:=baru;end
else begin Y^.next:=baru;Y:=baru;end;
inc(n);
writeln('Buku berhasil ditempatkan di posisi teratas');
end;
```

```
procedure cetak_tumpukan(var X,Y:pointer);
var bantu:pointer;i:byte;
begin
writeln('Daftar buku di dalam tumpukan dari posisi terbawah');
writeln('-----------------------------------------');
writeln('posisi     Judul Buku        Pengarang');
writeln('-----------------------------------------');
new(bantu);bantu:=X;i:=0;
while bantu<>nil do
begin
  inc(i);
  writeln(i:3,'    ',bantu^.judul:15,'    ',bantu^.pengarang);
  bantu:=bantu^.next;
end;
writeln('-----------------------------------------');
writeln('Saat ini di tumpukan terdapat ',n,' buku');
end;

procedure ambil_urut(var x,y:pointer);
var bantu,hapus:pointer; ya:char;
begin
new(bantu);bantu:=X;
while (bantu^.next)^.next<>nil do bantu:=bantu^.next;
hapus:=Y;
writeln('Anda akan mengambil buku sbb:');
writeln('Judul buku       : ',hapus^.judul);
writeln('Pengarang        : ',hapus^.pengarang);
write('yakin akan diambil<y/t> ? ');readln(ya);
if (ya='y') or (ya='Y') then
begin
  Y:=bantu; Y^.next:=nil;
  dispose(hapus);
  writeln('Buku telah diambil dari tumpukan');
  dec(n);
end
else writeln('Buku tidak jadi diambil');
end;

procedure ambil_acak(var x,y:pointer);
var j_ambil,p_ambil:string; ada:boolean;
  bantu,hapus:pointer;

begin
ada:=false;new(hapus);new(bantu);
writeln('Ambil buku dari tumpukan');
write('masukkan judul buku  yang akan diambil : ');readln(j_ambil);
write('masukkan pengarang yang akan diambil  : ');readln(p_ambil);
if (X^.judul=j_ambil) and (X^.pengarang=p_Ambil) then  {ada di awal}
begin ada:=true; hapus:=X;X:=X^.next;end
```

```
else if (Y^.judul=j_ambil) and (Y^.pengarang=p_Ambil)  then {ada di akhir}
  begin
          ada:=true;
          bantu:=X;
          while (bantu^.next)^.next<>nil do bantu:=bantu^.next;
          hapus:=Y;Y:=bantu;Y^.next:=nil;
  end
else
begin {ada ditengah selain awal atau akhir}
 bantu:=X;
 while bantu^.next<>nil do
 begin
 if (((bantu^.next)^.judul)=j_ambil)
   and (((bantu^.next)^.pengarang)=p_ambil)  then
          begin
                  hapus:=bantu^.next;
                  bantu^.next:=hapus^.next;
                  ada:=true;
          end;
          bantu:=bantu^.next;
  end;
end;
if ada then
begin
  writeln('anda berhasil mengambil buku sbb');
  writeln('Jenis   : ',hapus^.judul);
  writeln('warna           : ',hapus^.pengarang);
  dispose(hapus);
  dec(n)
end
else writeln('buku dengan judul ',j_ambil,' dan pengarang ',
p_ambil,' tidak ditemukan');
end;

procedure ambil_buku(var X,Y:pointer);
var ambil:byte;
begin
writeln('Mengambil buku dari tumpukan');
writeln('1. ambil urut dari atas');
writeln('2. Ambil acak berdasar judul');
writeln('0. kembali ke menu utama');
write('Pilih : ');readln(ambil);
case ambil of
1: ambil_urut(X,Y);
2: ambil_acak(X,Y);
0: writeln('Tekan enter') else writeln('Salah pilih menu');
end;
end;
```

```
begin
new(bawah);new(atas);bawah:=nil;atas:=nil;n:=0;
repeat
  clrscr;
  writeln('Pengelolaan Tumpukan buku');
  writeln('1. Tambah buku ke tumpukan');
  writeln('2. Ambil buku dari tumpukan');
  writeln('3. Cetak buku di tumpukan');
  writeln('0. Selesai');
  write('pilihan anda : ');readln(pil);
  case pil of
  1: if n=10 then writeln('maksimum tumpukan berisi 10 buku') else
          tambah_buku(bawah,atas);
  2: if n=0 then writeln('Tumpukan kosong') else ambil_buku(bawah,atas);
  3: if n=0 then writeln('Tumpukan kosong') else cetak_tumpukan(bawah,atas);
  0: writeln('Selesai') else writeln('Anda salah pilih menu');
  end;
  readln;
until pil=0;
end.
```

C. Tugas

Buatlah program untuk mengelola Dealer suatu kendaraan bekas dengan menggunakan linked list dengan ketentuan sbb:

1. Data kendaraan disimpan dengan atribut no_plat (unik),nama kendaraan dan harga
2. Data pembelian disimpan dengan atribut nama pembeli, no_plat kendaraan yang dibeli dan tanggal pembelian

Menu yang diinginkan adalah input data kendaraan, cetak kendaraan, antrean pembelian kendaraan, layanan pembelian (dari kendaraan yang telah diinputkan dan pembeli diambil dari antrian) dan cetak penjualan kendaraan

Catatan

Catatan

ISBN 978-602-7619-42-5