Modul Praktikum

Algoritma dan Pemrograman

Program Studi Teknik Informatika

STMIK STIKOM Indonesia

DAFTAR ISI

MODUL 1
JENIS-JENIS DATA DAN VARIABEL
(Pertemuan 4)

Tujuan :

1. Mempraktekkan berbagai tipe data sederhana dalam bahasa pemrograman
2. Deklarasi data dalam bahasa pemrograman
3. Mempraktekkan perintah *input* dan *output* dalam bahasa pemrograman.

Tugas Pendahuluan :

1. Apa yang anda ketahui tentang pemrograman dasar? Jelaskan!
2. Sebutkan tipe data apa saja yang terdapat pada pemrograman Pascal!
3. Apa perbedaan *write* dengan *writeln*? Jelaskan!
4. Buatlah sebuah program sederhana yang dapat memasukkan dan menampilkan data pada Pascal!

DASAR TEORI

Dalam bahasa Pascal terdapat beberapa jenis tipe data yang bisa digunakan untuk sebuah variabel atau konstanta pada program. Tipe data tersebut antara lain :

1. Tipe Bilangan Bulat

   Tipe bilangan bulat terdiri atas :

   - **Shortint** : jangkauan nilai dari -128 sampai 127
   - **Integer** : jangkauan nilai dari -32768 sampai 32767
   - **Longint** : jangkauan nilai dari -2147483648 sampai 2147483647
   - **Byte** : jangkauan nilai dari 0 sampai 255
   - **Word** : jangkauan nilai dari 0 sampai 65535

2. Tipe Bilangan Lojik

   - **Boolean** : merupakan tipe data yang berisi nilai dengan kemungkinan hanya berupa *False* (nilai salah) dan *True* (nilai benar).

3. Tipe Bilangan Riil

   Tipe riil adalah tipe yang berkaitan dengan segala jenis bilangan pecahan yang membutuhkan ketelitian dengan adanya nilai di belakang koma. Tipe bilangan riil terdiri atas :

   - **Real** : jangkauan nilai dari $\pm 2{,}9 \times 10^{-39}$ s/d $1{,}7 \times 10^{38}$
   - **Single** : jangkauan nilai dari $\pm 5{,}0 \times 10^{-45}$ s/d $3{,}4 \times 10^{38}$
   - **Double** : jangkauan nilai dari $\pm 5{,}0 \times 10^{-324}$ s/d $1{,}7 \times 10^{308}$
   - **Extended** : jangkauan nilai dari $\pm 5{,}0 \times 10^{-4951}$ s/d $1{,}1 \times 10^{4932}$

- **Comp** : jangkauan nilai dari ± 2^{63} s/d 2^{63} -1 (-9,2x10^{18} s/d 9,2 x10^{18})

4. Tipe Karakter
   - **Char** : merupakan tipe yang berisi sebuah karakter.
5. Tipe String
   - **String** : merupakan pengembangan dari tipe char. Tipe string dapat berupa sederetan karakter.

Variabel biasa digunakan di dalam program dengan tujuan untuk menampung data. Nilai yang terdapat pada variabel sewaktu-waktu dapat diubah. Jumlah variabel yang dibuat dapat tidak terbatas, namun masing-masing variabel tersebut harus bersifat unik dan tidak boleh ada nama ymbol e yang sama. Selain ymbol e terdapat konstanta yang juga dapat menampung data.Hanya saja dalam kostanta nilai yang ada tidak dapat diubah atau bernilai pasti.Dalam program, variabel disingkat dengan Var, sedangkan konstanta disingkat dengan Const.

Bentuk Umum :

```
Const
    Pengenal : Tipe Data;
Var
    Pengenal : Tipe Data;
    ...
    Pengenal : Tipe Data;
```

Adapun aturan-aturan penamaan ymbol e dalam bahasa Pascal adalah sebagai berikut :

1. Tidak boleh mengandung spasi, ymbol, atau tanda.
2. Tidak boleh diawali dengan angka.
3. Tidak boleh menggunakan kata kunci yang sudah terdapat di dalam bahasa Pascal.

Contoh :

```
Var
    Nama : String[20];
    Umur : Byte;
    Kelas : Char;

Salah :
Var
    Nama lengkap : string;
    4asal : string;
    @email : string;
    For : integer;
```

Sebuah program dapat menampilkan kalimat ke layar. Hal ini biasanya dilakukan untuk menampilkan perintah untuk memasukkan masukan program kepada pemakai program (*user*). Berikut adalah cara untuk menampilkan kalimat ke layar :

```
write(string);
writeln(string);

Misal :
write('masukkan bilangan bulat :  ');
writeln('masukkan bilangan bulat');
```

KEGIATAN PRAKTIKUM

Membuat Program Biodata

Salinlah *coding* program berikut ini ke dalam Turbo Pascal :

```
program biodata;
uses crt;

var
nama : string[20];
umur : string[2];
alamat : string[50];

begin
      clrscr;
      write('masukkan nama anda : '); readln(nama);
      write('masukkan umur anda : '); readln(umur);
      write('masukkan alamat anda : '); readln(alamat);
      writeln('');
      writeln('Nama : ', nama);
      writeln('Umur : ', umur);
      writeln('Alamat : ', alamat);
      readln;
end.
```

Kompilasi program tersebut dengan menekan *Alt+F9* dan jalankan program tersebut dengan menekan *Ctrl+F9*, kemudian amati hasilnya bila dimasukkan masukan tertentu. Sekarang simpan program tersebut dengan memilih menu *File* lalu pilih *Save.* Simpan dengan nama**PRAK01.PAS**.

Membuat Program Hitung Luas Lingkaran

Salinlah *coding*program berikut ini ke dalam Turbo Pascal :

```
program luas_lingkaran;
uses crt;

const
     phi=3.14159;
var
     luas, jari2 : real;

begin
     clrscr;
     write('masukkan jari-jari lingkaran : '); readln(jari2);
     luas:= phi*sqr(jari2);
     writeln('luas lingkaran adalah : ', round(luas));
     readln;
end.
```

Kompilasi dan jalankan program tersebut, kemudian amati hasilnya bila dimasukkan masukan tertentu. Simpan pekerjaan anda dengan nama**PRAK02.PAS**. Amati kembali, apa yang terjadi bila nilai jari-jari sama dengan nol? Apa pula yang terjadi jika nilai jari-jari diisi dengan huruf?

TUGAS

1. Buatlah program Pascal untuk menentukan hasil penjumlahan dan pengurangan dari 2 bilangan bulat!
2. Buatlah program Pascal yang menerima sebuah masukan dan menampilkan hasil kuadrat dari bilangan masukan!
3. Buatlah program Pascal yang menerima 2 buah masukan bilangan bulat untuk menghitung keliling persegi panjang!

MODUL 2
OPERATOR LOGIKA, EKSPRESI RUNTUTAN, TEKNIK RUNTUTAN
(Pertemuan 5)

Tujuan :

1. Mempraktekkan jenis-jenis Operator Logika dalam bahasa pemrograman
2. Mempraktekkan jenis-jenis Ekspresi Logika dalam bahasa pemrograman
3. Mempraktekkan Struktur Runtutan dalam bahasa pemrograman

Tugas Pendahuluan :

1. Apa yang anda ketahui mengenaioperator logika? Sebutkan!
2. Buatlah program Pascaldengan menggunakan ekspresi logika!
3. Buatlah program Pascaldengan struktur runtutan yang benar!

DASAR TEORI

Operator adalah simbol atau tanda yang jika diletakkan pada dua buah operand dapat menghasilkan sebuah hasil, contohnya pada matematika dimana tanda tambah (‘+’) jika diletakkan di antara dua buah angka akan menghasilkan angka lain hasil pertambahan dari dua angka tersebut. Tanda tambah inilah yang disebut dengan operator.

Dalam logika, dua kalimat dapat digabungkan dengan operator logika.Operator logika dalam program digunakan untuk melakukan operasi-operasi yang menghasilkan nilai logik (*true* dan *false*).Adapun operand dalam operasi ini juga bertipe logik. Berikut adalah empat buah operator logika dalam Pascal :

- NOT : Negasi
- AND : Konjungsi
- OR : Disjungsi
- XOR : Eksklusif Disjungsi

Sebuah ekspresi merupakan kumpulan dari operand-operand (seperti : bilangan, konstanta, variabel, dan lain-lain) yang bersama-sama dengan operator membentuk suatu bentuk aljabar dan menyatakan suatu nilai.Ada dua jenis ekspresi dalam bahasa Pascal, yaitu :

1. Ekspresi numerik/aritmatika, yaitu suatu ekspresi yang menghasilkan nilai numerik/aritmatika.
2. Ekspresi Boolean/logika, yaitu suatu ekspresi yang menghasilkan nilai boolean (*true*/*false*).
   Contoh :

1. (b * b – 4 * a * c) / (2*a) / (2 * a) Þ ekspresi numerik, jika a,b dan c adalah bilangan (variabel bernilai numerik).
2. Upah < 1000.0 Þ ekspresi boolean (“upah” adalah suatu variabel bernilai real).

Dalam Pascal terdapat struktur algoritma yang paling sederhana dan mendasar yang disebut dengan struktur runtutan.Instruksi pada Pascal diproses secara sekuensial atau berkelanjutan.Jadi, sebelum membuat program pada Pascal sebaiknya membuat algoritmanya terlebih dahulu.Sebuah program Pascal yang lengkap tersusun atas 3 bagian :

1. Kepala Program
2. Bagian Deklarasi
3. Bagian Pernyataan

Urutan bagian-bagian di atas, letaknya di dalam program sesuai dengan penomorannya. Kepala program diawali dengan kata-tercadang **PROGRAM**, lalu diikuti dengan nama program dan tanda titik koma. Bagian deklarasi dalam program dapat saja tidak dilibatkan sama sekali. Bagian pernyataan selalu diawali dengan *begin* dan diakhiri dengan *end* dan titik(.). Dengan demikian :

```
begin
end.
```

merupakan contoh program terpendek pada Turbo Pascal, mengingat :

- Bagian kepala program hanya bersifat opsional,
- Bagian deklarasi boleh tidak ada,
- Pernyataan di dalam bagian pernyataan boleh tidak ada.

Berikut ciri dari struktur runtutan yang benar :

1. Tiap baris instruksi dikerjakan secara satu-persatu
2. Tidak ada pengulangan untuk setiap baris instruksi
3. Urutan instruksi yang dijalankan harus sama dengan urutan instuksi di algoritma
4. Akhir dari instruksi merupakan akhir dari algoritma.

KEGIATAN PRAKTIKUM

Menggunakan Operator NOT

Salinlah *coding* program berikut ini ke dalam Turbo Pascal :

```
program OperatorNOT;
uses crt;

var
   a : boolean;

begin
     clrscr;
     a := false;
     a := not a;
     writeln(a);
     readln;
end.
```

Kompilasi dan jalankan program tersebut, kemudian amati hasilnya. Simpan pekerjaan anda dengan nama**PRAK03.PAS**. Pada program di atas, variabel a yang telah diberi nilai *false* kemudian diberi pernyataan*not* a, maka nilai variabel tersebut akan menjadi *true* karena yang diinginkan adalah *not false*.

Menggunakan Operator AND, OR, dan NOT

Salinlah *coding* program berikut ini ke dalam Turbo Pascal :

```
program Boolean;
uses crt;

var
kar : char;

begin
clrscr;
write('masukkan sebuah karakter : '); readln(kar);
writeln('apakah merupakan huruf kapital?');
writeln( (kar>='A') AND (kar<='Z') );
writeln('bukan berupa huruf kapital?');
writeln( NOT ((kar>='A') AND (kar<='Z')) );

writeln('tekan enter');
readln;
end.
```

Kompilasi dan jalankan program tersebut, kemudian amati hasilnya dan pahami logikanya. Simpan pekerjaan anda dengan nama**PRAK04.PAS.** Setelah diamati kembali, bagaimana menurut anda? Apakah sudah bisa membedakan antara *and, or,* dan *not*?

Membuat Program dengan Ekspresi Aritmatika

Salinlah *coding* program berikut ini ke dalam Turbo Pascal :

```
program EkspresiAritmatika;
uses crt;

var
   x : real;

begin
     clrscr;
     x := 1 + 2 * 3 - 4 / 2;
     writeln(x:0:2);
     readln;
end.
```

Kompilasi dan jalankan program tersebut, kemudian amati hasilnya. Simpan pekerjaan anda dengan nama**PRAK05.PAS.** Berapakah hasilnya?Mengapa angka dibelakang koma dapat berjumlah dua buah? Cobalah mengganti angka yang akan dijumlahkan. Bagaimana hasilnya?

TUGAS

1. Buatlah algoritma dan program yang menerima masukan alas dan tinggi sebuah segitiga, dan akan mengeluarkan nilai luas segitiga tersebut!
2. Buatlah program yang menerima masukan Nama, Alamat, Tahun Lahir, dan Tahun Sekarang yangakan menampilkan Jumlah Usia dari tahun yang dimasukkan!
3. Buatlah program pengukuran suhu derajat Celcius ke derajat Fahrenheit dan Reamur dengan masukan derajat Celcius dan tampilkan hasilnya!

MODUL 3
TEKNIK PERCABANGAN (BAGIAN 1)
(Pertemuan 6)

Tujuan :

1. Mempraktekkan tentang pembacaan data secara percabangan dalam bahasa pemrograman
2. Mempraktekkan perintah-perintah percabangan dalam bahasa pemrograman

Tugas Pendahuluan :

1. Apa yang anda ketahui tentang percabangan pada Pascal? Jelaskan!
2. Mengapa percabangan begitu dibutuhkan pada bahasa pemrograman?
3. Buatlah program pernyataan *if*yangmenampilkan grade A untuk nilai di atas 7!

DASAR TEORI

Percabangan *if* merupakan sebuah blok program yang menyatakan bahwa sebuah aksi akan dijalankan jika kondisi percabangan dipenuhi jika tidak dipenuhi maka aksi tidak akan dijalankan. Percabangan *if* biasa digunakan untuk mengerjakan aksi yang memiliki syarat tertentu untuk menjalankannya. Pernyataan *if* diklasifikasikan lagi ke dalam tiga bagian, yaitu :

1. Pernyataan *if* dengan satu kondisi (*if* tunggal)
2. Pernyataan *if* dengan dua kondisi (*if - else*)
3. Pernyataan *if*bersarang (*if* di dalam *if*)

Pernyataan *If* tunggal hanya melibatkan satu kondisi yang akan diperiksa. Apabila kondisi yang diperiksa bernilai benar, maka program akan mengeksekusi bagian yang berada dalam blok. Jika sebaliknya, maka program akan mengabaikan statemen di dalam blok dan langsung melanjutkan eksekusi berikutnya.

Bentuk Umum :

```
{terdiri dari satu statemen}
if (kondisi) then
  statemen;

{terdiri atas beberapa statemen}
if (kondisi) then
  begin
    statemen1;
    statemen2;
    ...
  end.
```

Pernyataan *if* dengan dua kondisi (*if - else*) dipergunakan untuk menyatakan pernyataan percabangan dua kondisi dimana ada dua blok aksi yang dipilih untuk dikerjakan jika syarat kondisi aksi terpenuhi. Saat pembacaan program sampai pada blok*if-else* maka akan dilakukan pemeriksaan terhadap syarat kondisi percabangan yang ada pada deklarasi *if*, jika syarat dipenuhi maka yang akan dijalankan adalah aksi yang ada di dalam blok *if*, tapi jika syarat tidak dipenuhi maka aksi yang dikerjakan adalah yang ada di dalam blok *else*.

Bentuk Umum :

```
if (kondisi) then
  statemen1
else
  statemen2;

atau

if (kondisi) then
  begin
    statemen1a;
    statemen1b;
    ...
  end
  else
  begin
    statemen2;
  end;
```

KEGIATAN PRAKTIKUM

Penentuan Nilai Menggunakan Percabangan Sederhana

Salinlah *coding* program berikut ini ke dalam Turbo Pascal :

```
program satuif;
uses crt;

var
skor : integer;
nilai : char;

begin
clrscr;
write('masukkan skor : ');readln(skor);
if skor > 7 then nilai := 'A';
writeln('Nilai = ',nilai);
readln;
end.
```

Kompilasi program tersebut dengan menekan *Alt+F9* dan jalankan program tersebut dengan menekan *Ctrl+F9*, kemudian amati hasilnya bila dimasukkan masukan skor yang berbeda. Sekarang simpan program tersebut dengan memilih menu *File* lalu pilih *Save*. Simpan dengan nama**PRAK06.PAS**.

Penentuan Nilai dengan Banyak Statemen

Salinlah *coding* program berikut ini ke dalam Turbo Pascal :

```
program statemen;
uses crt;

var
skor : integer;
nilai : char;
lulus : boolean;
bonus : longint;

begin
clrscr;
write('masukkan skor : ');readln(skor);
if skor > 7 then
nilai := 'A';
lulus := True;
bonus := 50000;
writeln('Nilai = ',nilai);
writeln('Lulus = ',lulus);
writeln('Bonus = ',bonus);
readln;
end.
```

Kompilasi dan jalankan program tersebut, kemudian amati hasilnya dan pahami logikanya. Simpan pekerjaan anda dengan nama**PRAK07.PAS.**

Menentukan Kubus dan Bukan Kubus dengan *If*

Salinlah *coding* program berikut ini ke dalam Turbo Pascal :

```
program ifduakondisi;
uses crt;

var
   s1, s2, s3 : integer;

begin
clrscr;
write('masukkan sisi pertama : ');readln(s1);
write('masukkan sisi kedua : ');readln(s2);
write('masukkan sis ketiga : ');readln(s3);
writeln;
if (s1 = s2) and (s2 = s3) then
begin
writeln('Termasuk Kubus');
end
else
begin
writeln('Bukan Kubus');
end;
readln;
end.
```

Kompilasi program tersebut dan jalankan, kemudian amati hasilnya bila dimasukkan tiga buah bilangan sisi. Program tersebut menggunakan percabangan *if*dengan dua kondisi. Perlu

diingat bahwa *end* sebelum *else*tidak perlu diberi penutup titik koma (;).Sekarang simpan program tersebut dengan nama**PRAK08.PAS**.

TUGAS

1. Buatlah program yang menerima dua masukan bilangan yang memiliki syarat bilangan pertama tidak boleh lebih kecil dari 3 dan bilangan kedua tidak boleh lebih kecil dari 4. Jika syarat dipenuhi, maka akan muncul kalimat “Pascal”. Gunakan percabangan *if*satu kondisi yang disertai dengan *or*!
2. Buatlah program penentuan bilangan ganjil yang menerima masukan sebuah bilangan kemudian menampilkan apakah bilang tersebut adalah bilangan ganjil dengan menggunakan percabangan satu *if*!
3. Buatlah program bilangan terbesar di antara 3 buah bilangan yang dimasukkan dengan menggunakan *if* bersarang!

MODUL 4
TEKNIK PERCABANGAN (BAGIAN 2)
(Pertemuan 7)

Tujuan :

1. Mempraktekkan tentang pembacaan data secara percabangan dalam bahasa pemrograman
2. Mempraktekkan perintah-perintah percabangan dalam bahasa pemrograman

Tugas Pendahuluan :

1. Jelaskan menurut pengetahuan anda apa yang dimaksud dengan pernyataan *if multiple condition*!
2. Buatlah program pemilihan menu dengan menggunakan seleksi *case*!
3. Buatlah program kalkulator dengan menggunakan pernyataan *if* bersarang!

DASAR TEORI

Selain pernyataan *if* dengan dua kondisi, suatu pernyataan *if* dapat mengandung pernyataan *if* yang lain. Bentuk seperti ini biasa disebut *if* bersarang (*nested if*).Sebuah program mengijinkan blok percabangan *if* di dalam blok percabangan lainnya, dan tidak membatasi jenis percabangan apa yang boleh berada di dalam percabangan lainnya.

Bentuk Umum :

```
if (kondisi1) then
    if (kondisi2) then
      statemen1
else
      statemen2;

atau

if (kondisi1) then
    if (kondisi2) then
    begin
     statemen1;
    end
else
     statemen2;
```

Pernyataan *case* merupakan alternatif dari pernyataan *if* untuk masalah dengan pilihan berganda. Pada masalah tertentu, *case* lebih memberikan kejelasan daripada *if*. Namun perlu diketahui bahwa semua persoalan yang dapat ditangani *case* bisa ditangani oleh *if*, tetapi tidak sebaliknya. *Case* biasanya digunakan untuk memilih di antara lebih dari 2 pilihan. *Case* digunakan untuk menggantikan struktur *if-else-if* dimana kondisinya mengacu pada variabel yang sama.

Bentuk Umum :

```
case variabel of
    kondisi 1 : statemen 1;
    kondisi 2 : statemen 2;
    ...
    kondisi n : statemen n;
else
    statemen n+1;
end;
```

KEGIATAN PRAKTIKUM

Menentukan Harga Berdasarkan Status dan Jabatan

Salinlah *coding* program berikut ini ke dalam Turbo Pascal :

```
program if_tersarang;
uses
crt;

var
status : string;
jabatan : string;
harga : longint;

begin
clrscr;
write('masukkan status (1. mahasiswa/2. non mahasiswa) : ');
readln(status);
write('masukkan jabatan (1. panitia/2. non panitia) : ');
readln(jabatan);

if status='1' then
begin
if jabatan='1' then
begin
harga:=80000;
write('harga : ',harga);
end
else if jabatan='2' then
begin
harga:=120000;
write('harga : ',harga);
end
else
begin
writeln('inputan jabatan salah!');
end;
end
else if status='2' then
begin
if jabatan='1' then
begin
harga:=100000;
write('harga : ',harga);
end
else if jabatan='2' then
begin
harga:=150000;
write('harga : ',harga);
end
```

```
      else
begin
writeln('inputan jabatan salah!');
end;

end
else
begin
writeln('inputan status salah!');
end;
readln;
end.
```

Kompilasi dan jalankan program tersebut, kemudian amati hasilnya dan pahami logikanya. Bagaimana jika inputan yang dimasukkan adalah angka atau huruf yang berbeda?Mengapa hal tersebut bisa terjadi? Pahamilah proses yang terjadi pada program tersebut. Simpan pekerjaan anda dengan nama**PRAK09.PAS.**

Keterangan Grade Menggunakan *Case Of*

Salinlah *coding* program berikut ini ke dalam Turbo Pascal :

```
program case_of;
uses
    crt;

var
   grade : char;
   ket : string;

begin
     clrscr;
     write('masukkan grade (a/b/c) : '); readln(grade);

     case grade of
          'a' : begin
                     ket:=’amat baik';
                end;
          'b' : begin
                     ket:='baik';
                end;
          'c' : begin
                     ket:='cukup';
                end;
          else
              begin
                   ket:='inputan grade salah';
              end
     end;

     write('keterangan :', ket);
     readln;
end.
```

Kompilasi dan jalankan program tersebut, kemudian amati hasilnya dan pahami logikanya. Pahamilah proses yang terjadi pada program tersebut. Simpan pekerjaan anda dengan nama**PRAK10.PAS.**

TUGAS

1. Buatlah program kalkulator yang menerima masukan dua buah bilangan, kemudian menerima masukan pilihan menu berupa penjumlahan, pengurangan, dan perkalian. Selanjutnya kedua buah bilangan yang telah dimasukkan tersebut akan diproses sesuai dengan menu yang telah dipilih!
2. Buatlah program penentuan bonus bagi pembeli berdasarkan total pembelian yang dimasukkan, dimana kriterianya adalah jika total pembelian lebih dari 100.000 maka pembeli mendapatkan diskon sebesar 10%, jika total pembelian kurang dari 100.000 dan lebih dari 50.000 maka pembeli mendapatkan sebuah piring cantik, jika total pembelian kurang dari 50.000 dan lebih dari 10.000 maka pembeli mendapatkan sebuah gelas cantik, selanjutnya jika total pembelian kurang dari 10.000 maka pembeli tidak akan mendapatkan bonus!
3. Dapatkah soal nomor 2 dipecahkan menggunakan *case* (tanpa *if*)? Berikan penjelasan anda!

MODUL 5
TEKNIK PENGULANGAN (BAGIAN 1)
(Pertemuan 8)

Tujuan :

1. Mempraktekkan tentang pembacaan data secara berulang dalam bahasa pemrograman
2. Mempraktekkan perintah-perintah perulangan serta analisa kondisi dan aksi dengan perulangan dalam bahasa pemrograman.

Tugas Pendahuluan :

1. Apakah yang dimaksud dengan pengulangan pada pemrograman dasar? Jelaskan!
2. Ada berapa macam pengulangan yang terdapat pada Pascal? Sebutkan!
3. Buatlah program pengulangan sederhana dengan menggunakan *for* dan *while do*!

DASAR TEORI

Pengulangan (*looping*) adalah suatu bagian yang bertugas melakukan kegiatan mengulang suatu proses sesuai dengan yang diinginkan. Banyak dari aplikasi perangkat lunak yang melakukan pekerjaan berulang-ulang sampai sebuah kondisi yang diinginkan, oleh karena itu pengulangan merupakan bagian yang penting dalam pemrograman karena dengan adanya pengulangan pembuat program tidak perlu menulis kode program sebanyak pengulangan yang diinginkan.

Pascal menyediakan beberapa konstruksi perintah untuk melakukan proses-proses pengulangan, yaitu :

1. FOR
2. WHILE ... DO
3. REPEAT ... UNTIL

Struktur *for* digunakan untuk melakukan perulangan yang tidak berkondisi.Artinya jumlah perulangannya telah diketahui dengan pasti.

Bentuk Umum :

```
(Perulangan Positif)
for variabel := <awal> to <akhir> do
begin
     pernyataan
end;

(Perulangan Negatif)
for variabel := <awal> downto <akhir>
do
begin
     pernyataan
end;
```

Pada masalah tertentu ada kemungkinan *for* berada di dalam *for* yang lain. Bentuk seperti ini biasa disebut dengan *nested loop* (*for* bersarang).

Pengulangan *while do*biasa digunakan jika jumlah pengulangan tidak diketahui atau memiliki kemungkinan dapat dilakukan kurang dari batas pengulangan yang telah ditentukan. Pengulangan *while* hanya akan melakukan pengulangan selama kondisi pengulangan terpenuhi. Perintah-perintah akan dilaksanakan apabila ekspresi boolean dalam keadaan *true*. Di dalam *loop* ada nilai yang mengontrol *loop* dan nilainya harus berubah, sehingga pada akhir program akan keluar dari *loop.*

Bentuk Umum :

```
while kondisi do
begin
     pernyataan
end;
```

KEGIATAN PRAKTIKUM

Membuat Perulangan Positif

Salinlah *coding* program berikut ke dalam Turbo Pascal :

```
program positif;
uses crt;

var
   i : integer;
begin
     clrscr;
     for i := 1 to 5 do
     begin
          writeln('isi dari i adalah : ',i);
     end;
     readln;
end.
```

Kompilasi dan jalankan program tersebut.Apa yang terjadi? Amati dan pahamilah logikanya. Simpan pekerjaan anda dengan nama**PRAK11.PAS**.

Membuat Perulangan Negatif

Salinlah *coding* program berikut ke dalam Turbo Pascal :

```
program negatif;
uses crt;

var
   i : integer;
begin
     clrscr;
     for i := 1 downto 5 do
     begin
          writeln('isi dari i adalah : ',i);
     end;
     readln;
end.
```

Kompilasi dan jalankan program tersebut.Apa yang terjadi? Amati dan pahamilah logikanya.Apa perbedaan antara perulangan positif dan perulangan negatif menurut anda setelah menjalankan program di atas?

Menghitung Rata-rata dari Sejumlah Nilai

Salinlah *coding* program berikut ke dalam Turbo Pascal :

```
program rata2;
uses crt;

var
   nilai, jumlah : real;
   i : integer;
begin
     clrscr;
     jumlah := 0;
     for i := 1 to 5 do
     begin
          write('Nilai ke-',i,' : ');readln(nilai);
          jumlah := jumlah + nilai;
     end;
     writeln('Nilai rata-rata = ',jumlah/5 :0:2);
     readln;
end.
```

Kompilasi dan jalankan program tersebut.Apa yang terjadi? Amati dan pahamilah logikanya. Simpan pekerjaan anda dengan nama**PRAK12.PAS**.

Program *For*Bersarang

Salinlah *coding* program berikut ke dalam Turbo Pascal :

```
program nested_loop;
usescrt;

var
i, j : integer;
begin
clrscr;
for i:= 1 to 2 do
begin
for j:= 1 to 3 do
begin
write(i,' ',j);
writeln;
end;
end;

readln;
end.
```

Kompilasi dan jalankan program tersebut. Seperti apa tampilan yang muncul? Amati dan pahamilah logikanya. Simpan pekerjaan anda dengan nama**PRAK13.PAS**.

Mengurut Bilangan dengan *While Do*

Salinlah *coding* program berikut ke dalam Turbo Pascal :

```
program while1;
uses crt;

var
i : integer;

begin
clrscr;
i := 1;
while i <= 10 do
begin
writeln(i);
i := i + 1;
end;
readln;
end.
```

Kompilasi dan jalankan program tersebut.Amati dan pahamilah logikanya. Simpan pekerjaan anda dengan nama**PRAK14.PAS**.

TUGAS

1. Buatlah program yang menerima masukan batas awal dan batas akhir dan menampilkan perkalian deret dari bilangan yang telah dibatasi!
2. Buatlah program yang akan menampilkan deretan bilangan ganjil seperti ini : (1 3 5 7 9 11 13 15) menggunakan pengulangan *for*!
3. Buatlah program pengulangan *for* bersarang yang menampilkan tampilan seperti berikut!

```
*
**
***
****
*****
```

MODUL 6
TEKNIK PENGULANGAN (BAGIAN 2)
(Pertemuan 9)

Tujuan :

1. Mempraktekkan tentang pembacaan data secara berulang dalam bahasa pemrograman
2. Mempraktekkan perintah-perintah perulangan serta analisa kondisi dan aksi dengan perulangan dalam bahasa pemrograman.

Tugas Pendahuluan :

1. Jelaskan perbedaan dari *for*, *while ... do*, dan *repeat ... until*!
2. Apakah fungsi dari *repeat ... until* dan pada saat kondisi apa harus menggunakan *repeat ... until* pada Pascal?

DASAR TEORI

Pengulangan *repeat until* biasa digunakan jika jumlah pengulangan tidak diketahui atau belum pasti, namun berbeda dengan *while* karena kondisi pengulangan ada di bagian bawah blok pengulangan. Pengulangan *repeat* minimal selalu dilakukan sekali karena kondisi pengulangan ada di bagian bawah, berbeda dengan pengulangan *while* yang saat pertama kali masuk blok pengulangan dilakukan pemeriksaan kondisi pengulangan. Pada *repeatuntil* kita tidak perlu menggunakan*begin end* karena pernyataan *repeat until* pada Pascal diperlakukan sebagai sebuah blok.

Bentuk Umum :

```
repeat
    pernyataan_1;
    pernyataan_2;
    ...
    pernyataan_n;
until kondisi
```

Di dalam teknik pengulangan terdapat teknik counter yang berfungsi untuk mengontrol pengulangan proses. Pengontrolan ini dilakukan dengan memeriksa isi variabel yang digunakan sebagai counter, sehingga jumlah pengulangan dapat diketahui.

KEGIATAN PRAKTIKUM

Program *Repeat Until*

Salinlah *coding* program berikut ke dalam Turbo Pascal :

```
program repeat;
uses crt;

var
    i : integer;

begin
     i := 1;
     repeat
        i := i + 1;
     until i = 5;
     readln;
end.
```

Kompilasi dan jalankan program tersebut.Amati dan pahamilah logikanya. Simpan pekerjaan anda dengan nama**PRAK15.PAS**.

Membuat Tabel Derajat Celcius, Fahrenheit, Reamur

Salinlah *coding* program berikut ke dalam Turbo Pascal :

```
program tabel_suhu;
uses crt;

var
   celcius, reamur, fahrenheit : real;

begin
     clrscr;
     writeln('celcius':12,'reamur':12,'fahrenheit':12);
     writeln('-------------------------------------------------');
     celcius := 0;
     repeat
           reamur := 4 / 5 * celcius;
           fahrenheit := 9 / 5 * celcius + 32;
           writeln(celcius:12:2, reamur:12:2, fahrenheit:12:2);
           celcius := celcius + 0.5;
     until celcius > 10;
     writeln('-------------------------------------------------');
     readln;
end.
```

Kompilasi dan jalankan program tersebut.Amati dan pahamilah logikanya. Simpan pekerjaan anda dengan nama**PRAK16.PAS**.

Pilihan Pengulangan Tampilan

Salinlah *coding* program berikut ke dalam Turbo Pascal :

```
program repeat2;
uses crt;

var
    jawaban : char;

begin
     repeat
         writeln('Pemrograman Dasar');
         writeln;
         write('Apakah anda ingin menampilkan lagi? (Y/T)');
         readln(jawaban);
     until (jawaban = 't');
     readln;
end.
```

Kompilasi dan jalankan program tersebut.Amati dan pahamilah logikanya. Simpan pekerjaan anda dengan nama**PRAK17.PAS**.

TUGAS

1. Buatlah program untuk menampilkan bilangan genap dari 1 sampai dengan 100 menggunakan *repeat ... until*!
2. Buatlah program yang menerima masukan jumlah bintang dan menampilkan pola bintang sesuai dengan jumlah yang dimasukkan, seperti contoh di bawah ini!

```
Masukkan jumlah bintang : 5
*****
 *****
  *****
   *****
    *****
```

3. Selanjutnya buatlah program seperti nomor 2, namun dengan pola bintang seperti di bawah ini!

```
Masukkan jumlah bintang : 5
*****
 *****
  *****
 *****
*****
```

MODUL 7
PROSEDUR
(Pertemuan 10)

Tujuan:

- Mahasiswa diharapkan dapat mengetahui struktur serta cara menggunakan dan memanggil prosedur.
- Mahasiswa dapat mempraktekkan penggunaan prosedur dalam kasus yang berhubungan dengan prosedur.

Tugas Pendahuluan:

1. Apa yang Anda ketahui mengenai prosedur?
2. Apakah fungsi atau kegunaan dari sebuah prosedur?

DASAR TEORI

Prosedur merupakan bagian dari program yang berupa serangkaian *statement* yang memiliki nama. Sebuah prosedur dapat terdiri atas satu atau lebih dari satu prosedur. Penggunaan prosedur dapat memberikan beberapa keuntungan, sebagai berikut:

1. Dapat membagi atau memecah program menjadi lebih sederhana.
2. Dapat membantu menghemat penggunaan memori penyimpanan karena dapat dapat memperkecil ukuran *file.*
3. Dapat digunakan pada program lain cukup dengan mengopi prosedur yang hendak digunakan.

Oleh karena itu, prosedur pada umumnya digunakan pada program yang kompleks dan memiliki subprogram. Dalam penggunaannya, sebuah prosedur dapat digunakan dengan memanggil atau mengetikkan nama prosedur tersebut. Terdapat dua jenis prosedur, yaitu prosedur dengan parameter dan prosedur tanpa parameter. Berikut format penulisan pendeklarasian prosedur yang memiliki parameter:

```
procedure nama_prosedur (parameter1: tipe_data, parameter2:
tipe_data, ….);
        var (daftar variabel yang digunakan);
        begin
        (statement)
        end;
```

Selain prosedur dengan parameter, terdapat pula prosedur yang tidak memiliki parameter.Berikut format prosedur tanpa parameter:

```
procedure nama_prosedur;
        var (daftar variabel yang
digunakan);
        begin
        (statement)
        end;
```

KEGIATAN PRAKTIKUM

Salinlah program di bawah ini:

1. Prosedur tanpa parameter

```
program coba1;
uses crt;

procedure tampil;⟵ [prosedur]

begin
      Writeln('Hallo, ini percobaan pertama prosedur..');
end;


begin⟵ [program utama]
      clrscr;
      tampil;
      readln;
end.
```

2. Prosedur dengan parameter

```
programpagar;
uses crt;

var
   n1 : integer;

procedure tulispagar(n:integer);
var
      i : integer;

begin
      for i := 1 to n do
      write('# ');
end;

begin
      clrscr;
      write('Angka 1 : '); readln(n1);
      writeln;
      write(n1, ' | '); tulispagar(n1);
      readln;
end.
```

3. Menggunakan Lebih dari Satu Prosedur

```
program menu_prosedur;
uses crt;
var
   menu:integer;
   jawab:char;

procedure persegi;
var
         s,luas: integer;
begin
writeln;
        writeln('MENGHITUNG LUAS PERSEGI');
         write('Sisi: '); readln(s);
         luas:= s*s;
         write('Maka luas persegi adalah:
',luas);
end;


procedure persegi_panjang;
var
         p,l,luas: integer;
begin
        writeln;
        writeln('MENGHITUNG LUAS PERSEGI
PANJANG');
         write('Panjang: '); readln(p);
         write('Lebar: '); readln(l);
         luas:= p*l;
         write('Maka luas persegi panjang
adalah: ',luas);
end;
procedure jajargenjang;
 var
   alas, tinggi,luas: integer;

   begin
        writeln;
        writeln('MENGHITUNG LUAS
JAJARGENJANG');
        write('Alas: '); readln(alas);
        write('Tinggi: ' );readln(tinggi);
         luas:= alas*tinggi;
         write('Maka luas jajargenjang adalah:
',luas);
   end;


procedure layangan;
var
   d1,d2: integer;
   luas: real;
   begin
        writeln;
        writeln('MENGHITUNG LUAS LAYANG-
LAYANG');
        write('Panjang diagonal 1: ' );
         readln(d1);
        write('Panjang diagonal 2: ');
         readln(d2);
        luas:= 0.5*d1*d2;
        write('Maka luas layang-layang adalah:
',luas:2:0);
   end;
```

```
   begin
        clrscr;
        writeln('MENGHITUNG LUAS BANGUN
DATAR');
        writeln('1. Luas Persegi ');
        writeln('2. Luas Persegi Panjang');
        writeln('3. Luas Jajargenjang');
        writeln('4. Luas Layang-Layang');
         writeln('5. Keluar');
        writeln;
        writeln;
        write('Masukkan Pilihan anda
[1/2/3/4/5] : ' ); readln(menu);
        if (menu=1) then
                        begin
                        persegi;
                        end
        else if (menu=2) then
begin
                      persegi_panjang;
end
        else if (menu=3) then
                         begin
                         jajargenjang;
                         end
         else if (menu=4) then
                           begin
layangan;
end
        else
            begin
exit;
            end;
            writeln;
            readln;
            end.
```

Dalam sebuah program, bisa saja terdapat lebih dari satu *procedure* di dalamnya. Contoh di atas, merupakan contoh penggunaan lebih dari satu *procedure*.Dimana *procedure* yang ada dideklarasikan terlebih dahulu, untuk kemudian dapat dipanggil oleh program utama. Kumpulan *procedure* terdapat pada barisan *coding* sebelah kiri, sedangkan program utama yang memanggil *procedure-procedure* tersebut terdapat pada barisan *coding* di sebelah kanan.

TUGAS

1. Buatlah program untuk menghitung luas segi tiga dengan menggunakan prosedur yang memiliki parameter!

2. Buatlah program untuk menghitung luas persegi panjang dengan menggunakan prosedur tanpa parameter!

3. Buatlah program menggunakan lebih dari satu *procedure* dengan menu pilihan sebagai berikut:
    1) Menghitung Gaji
    2) *Booking* Kamar Hotel
    3) Karaoke
    4) Keluar

    Buatlah berdasarkan ketentuan seperti di bawah ini:

    1) Menghitung Gaji

        Gaji pokok setiap pegawai dilihat berdasarkan golongan kepegawaiannya, yaitu:
        - Golongan 1: Rp 2.000.000,-
        - Golongan 2: Rp 3.000.000,-
        - Golongan 3: Rp 4.000.000,-

        Selain itu, terdapat tunjangan bagi setiap pegawai yang telah menikah dengan ketentuan:
        - Tunjangan Rumah Tangga: 50% dari gaji pokok
        - Tunjangan Anak: jumlah anak * 25% dari gaji pokok

        Program yang dibuat diharapkan dapat menghasilkan output:
        - Nama karyawan:
        - Status:
        - Tunjangan:
        - Total Gaji:

2) *Booking* Kamar Hotel

Harga kamar hotel per-malamnya ditentukan dari jenis kamar hotel yang ada yaitu:

- *Single Room:* Rp 650.000,-/malam
- *Double Room:* Rp 900.000,-/malam
- *Deluxe Room:* Rp 1.200.000,-/malam
- *Suite Room:* Rp 1.500.000,-/malam

Petugas hotel selaku *user* nantinya akan bertugas untuk menginputkan nama tamu, jenis kamar yang diminta oleh tamu, serta lama menginap. Sehingga hasil atau *output* yang dihasilkan adalah jumlah biaya yang harus dibayarkan oleh tamu.

MODUL 8
FUNGSI
(Pertemuan 11)

Tujuan:

- Mahasiswa dapat mengetahui dan mempraktekkan tentang fungsi dan penggunaannya dalam pemrograman.
- Mahasiswa mengetahui struktur dan cara menggunakan fungsi.

Tugas Pendahuluan:

1. Apa yang Anda ketahui mengenai fungsi (*function*)?
2. Apakah kegunaan dari sebuah fungsi?
3. Apakah perbedaan antara fungsi dan prosedur?

DASAR TEORI

Pada dasarnya fungsi atau *function* maupun prosedur adalah sama. Fungsi maupun prosedur sama-sama merupakan subprogram.Selain itu, fungsi dan prosedur dapat digunakan atau dipanggil berkali-kali atau berulang-ulang dalam sebuah program.Namun, fungsi lebih cenderung digunakan untuk membuat perintah-perintah yang bersifat perhitungan.Selain itu, perbedaan antara fungsi dan prosedur adalah fungsi memiliki pengembalian nilai, sehingga pada saat dipanggil, sebuah fugnsi dapat langsung digunakan untuk mengisikan ekspresi yang ada.Dalam bahasa pemrograman Pascal, fungsi didefinisikan dengan *function.* Berikut format penulisan untuk mendeklarasikan sebuah *function:*

```
function nama_fungsi(parameter1: tipe_data, parameter2: tipe_data,
….):tipe_data;
        var (daftar variabel yang digunakan);
        begin
        (statement) ⟵ [pada bagian ini nantinya akan dilakukan pengembalian nilai]
        end;
```

KEGIATAN PRAKTIKUM

Salinlah *coding* program di bawah ini:

1. ***Function*tanpa input dari *user***

```
program fungsi;
uses crt;

function kali(d,s:integer):longint;
begin
kali:=d*s;
end;

begin
    clrscr;
    writeln(kali(10,30));
    readln;
end.
```

2. *Function* dengan input dari *user*

```
program fungsikali;
uses crt;

var
   angka1, angka2: integer;
   hasil: longint;

functionperkalian(a,d:integer):longint;
 begin
perkalian:=a*d;
     end;

begin
    clrscr;
    write('Angka pertama: '); readln(angka1);
    write('Angka kedua: '); readln(angka2);
    hasil := perkalian(angka1,angka2);
    write('Hasil perkaliannya= ',hasil);
    readln;
end.
```

TUGAS

1. Buat sebuah program untuk menghitung luas belah ketupat dengan ketentuan input dilakukan oleh *user*!

2. Buatlah sebuah program yang dapat menentukan bilangan terbesar di antara 3 angka yang di-inputkan oleh user menggunakan *function*!

MODUL 9
*ARRAY*1 DIMENSI
(Pertemuan 12)

Tujuan:

- Mahasiswa mengetahui tentang *array*dan jenis-jenis *array*, serta penggunaan *array* pada pemrograman.
- Mahasiswa dapat mempraktekkan penggunaan *array*1 dimensi dalam pemrograman.

Tugas Pendahuluan:

1. Apa yang Anda ketahui mengenai *array*?
2. Apakah kegunaan dari *array*?

DASAR TEORI

*Array*merupakan suatu tipe data yang terstruktur dan dapat digunakan untuk menyimpan data yang memiliki tipe data yang sama. Dengan kata lain, *array* adalah kumpulan data yang memiliki tipe yang sama. *Array* dapat diibaratkan seperti sebuah tabel.*Array*biasa juga disebut larik.*Array*dapat digunakan untuk menyimpan data yang berjumlah banyak namun masih memiliki suatu hubungan atau terdapat kesamaan antar data tersebut. Sebagai contoh, *array* dapat digunakan untuk menyimpan data 117 nama orang yang semuanya merupakan pelanggan sebuah toko obat, atau 65 nama yang kesemuanya merupakan nama mahasiswa yang mengambil kelas pada semester pendek. Data yang terdapat dalam sebuah *array*dapat diidentifikasikan atau dibedakan dengan menggunakan index.

Penggunaan *array* dapat mengurangi kerumitan dalam proses penyimpanan data dalam jumlah yang besar. Jika kita hendak menyimpan variabel nama yang berjumlah sepuluh nama, kita tidak harus membuat 10 variabel untuk nama tersebut. Dengan menggunakan *array* kita tidak perlu membuat banyak variabel untuk data yang memiliki tipe yang sama. Selain itu, dalam alokasi memori penyimpanan, tipe data *array* melakukan pemesanan tempat terlebih dahulu sesuai dengan kebutuhan yang ada.

Terdapat 2 jenis *array*, yaitu *array*1 dimensi dan *array*2 dimensi. Perbedaan mendasar antara kedua jenis *array* ini adalah, *array* dengan 1 dimensi merupakan *array* yang dapat digambarkan sebagai sebuah baris. Selain itu, dalam *array* 1 dimensi, elemen yang ada di dalamnya dapat diakses hanya dengan menggunakan 1 indeks saja.Sedangkan *array* 2 dimensi merupakan *array* yang dapat digambarkan seperti sebuah matrik. Selain itu elemen yang ada dalam *array* 2 dimensi dapat diakses dengan menggunakan 2 indeks, yaitu indeks kolom dan juga indeks baris. Berikut format pendeklarasian *array*:

```
NamaArray :array[indeks_awal..indeks_akhir]of tipe_data;
```

Contoh:

```
var
   abjad:array[1..26]ofchar;
```

KEGIATAN PRAKTIKUM

Salinlah *coding* program di bawah ini:

1. Pendeklarasian dan pengaksesan *array* 1 dimensi

```
program listmusik;
uses crt;

const
     maks = 5;

type
    jenis = array[1..maks] of string [20];

var
   musik : jenis;
   i : integer;

begin
    clrscr;
    writeln('-----*Jenis-jenis Musik*-----');
    musik[1] := 'JAZZ';
    musik[2] := 'FOLK';
    musik[3] := 'ALTERNATIVE';
    musik[4] := 'ROCK';
    musik[5] := 'POP';

    for i := 1 to maks do
        writeln(musik[i]);
    readln;
end.
```

Dengan melihat pendeklarasian *array* di atas maka diketahui bahwa, pendeklarasian *array* dilakukan dengan langsung mengisikan nilai dari tiap indeksnya.Lalu, bagaimana jika *user* sendiri yang diminta untuk mengisikan nilai dari tiap indeks yang ada?

2. Menghitung nilai rata-rata menggunakan *array*1 dimensi

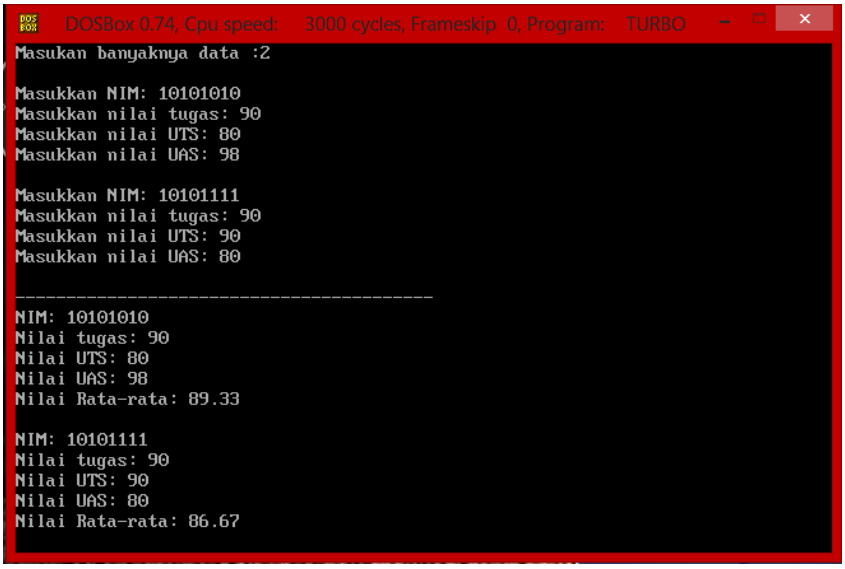
```
programratarata;
uses crt;
var
nim : longint;
nilai : array[1..4] of longint;
rata : array[1..10] of real;
i,byk : byte;

begin
clrscr;
write('Masukkan jumlah mahasiswa: '); readln(byk);
for i:=1 to byk do
begin
write('Masukkan NIM: ');readln(nim);
write('Masukkan nilai kuis: '); readln(nilai[1]);
write('Masukkan nilai tugas: '); readln(nilai[2]);
write('Masukkan nilai UTS: '); readln(nilai[3]);
write('Masukkan nilai UAS: '); readln(nilai[4]);
writeln;

rata[i]:=(nilai[1]+nilai[2]+nilai[3]+nilai[4])/4;
end;

writeln('----------------------------------------');
for i:=1 to byk do
begin
writeln('NIM: ',nim);
writeln('Nilai kuis:  ',nilai[1]);
writeln('Nilai tugas: ',nilai[2]);
writeln('Nilai UTS: ',nilai[3]);
writeln('Nilai UAS: ',nilai[4]);
writeln('Nilai Rata-rata: ',rata[i]:2:2);
readln;
end;
end.
```

Apabila diperhatikan dengan seksama, maka dapat dilihat bahwa *coding* program di atas merupakan contoh dari adanya penggunaan lebih dari 1 *array* dalam 1 program. Selain itu bahwa *coding* program di atas juga menunjukkan bagaimana cara menempatkan nilai sesuai dengan indeks yang ada.

TUGAS

1. Buatlah sebuah program yang dapat mencetak bilangan prima yang ada dalam bilangan 1-100 dengan menggunakan *array*!

2. Buatlah sebuah program untuk menghitung nilai rata-rata dengan menggunakan *array* 1 dimensi sehingga output atau tampilan yang dihasilkan seperti pada gambar di bawah ini!



3. Buatlah sebuah program yang dapat menentukan bilangan terbesar dari deretan angka! Dengan ketentuan, deretan angka tersebut di-inputkan oleh *user*!
4. Buatlah program untuk menentukan bilangan terkecil sesuai dengan ketentuan pada soal no.3!

MODUL 10
*ARRAY*2 DIMENSI
(Pertemuan 13)

Tujuan:

- Mempraktekkan tentang *array*dan jenis-jenis *array*, serta penggunaan *array* pada pemrograman.
- Mempraktekkan tentang *array*2 dimensi dalam pemrograman.

Tugas Pendahuluan:

1. Apa yang Anda ketahui tentang array 2 dimensi?
2. Apakah perbedaan antara *array* 1 dimensi dan *array* 2 dimensi?

DASAR TEORI

Seperti yang telah dijelaskan pada pertemuan sebelumnya, a*rray*merupakan suatu tipe data yang terstruktur dan dapat digunakan untuk menyimpan data yang memiliki tipe data yang sama. Dengan kata lain, *array* adalah kumpulan data yang memiliki tipe yang sama. Penggunaan *array* dapat mengurangi kerumitan dalam proses penyimpanan data dalam jumlah yang besar. Dengan menggunakan *array* kita tidak perlu membuat banyak variabel untuk data yang memiliki tipe yang sama.

Terdapat 2 jenis *array*, yaitu *array*1 dimensi dan *array*2 dimensi.*Array* dengan 1 dimensi dapat digambarkan sebagai sebuah baris, sedangkan *array* 2 dimensi merupakan dapat digambarkan seperti sebuah matrik. Berikut format pendeklarasian *array*2 dimensi:

```
NamaArray :array[1..banyak_baris,1..banyak_kolom]of tipe_data;
```

Sebagai contoh, apabila kita hendak mendeklarasikan *array* yang memiliki 4 baris dan 6 kolom dan memiliki tipe data char, maka yang harus kita ketikkan adalah:

```
var
     duadimensi :array[1..4,1..6]of char;
```

KEGIATAN PRAKTIKUM

Salinlah *coding* program di bawah ini:

- **Pendeklarasian dan pengaksesan *array* 2 dimensi**

```
program matrikduadimensi;
uses crt;

var
matrik:array[1..3,1..3] of integer;
i,j:byte;

begin
clrsccr;
writeln('Matrik 2 Dimensi :');
matrik[1,1] := 1;
matrik[1,2] := 2;
matrik[1,3] := 3;
matrik[2,1] := 4;
matrik[2,2] := 5;
matrik[2,3] := 6;
matrik[3,1] := 7;
matrik[3,2] := 8;
matrik[3,3] := 9;

for i:= 1 to 3 do
begin
for j:=1 to 3 do
write (matrik[i,j]:5);
writeln;
end;
readln;
end.
```

Koding di atas merupakan contoh koding yang dapat menampilkan sebuah matrik dengan ordo 3 x 3 menggunakan *array* 2 dimensi.

TUGAS

1. Buatlah sebuah program untuk menjumlahkan dua buah matrik menggunakan *array* dua dimensi dengan ketentuan:
   - Syarat dari penjumlahan matrik adalah jumlah ordo matrik A = jumlah ordo matrik B.
   - *User* yang menginputkan ordo matrik.
   - *User* yang menginputkan nilai dari tiap elemen matrik yang ada.
2. Buat program perkalian antara 2 buah matrik dengan menggunakan *array* dimana nilai tiap elemennya di-inputkan oleh *user*! Selain itu, perhatikan syarat perkalian antar matrik, yaitu jumlah kolom pada matrik A = jumlah baris pada matrik B!
3. Transpose matrik adalah terjadinya pertukaran tempat antara elemen baris dan elemen kolom dan sebaliknya pada sebuah matrik. Buatlah sebuah program transpose matrik dengan ordo yang ditentukan oleh *user* sendiri!

MODUL 11
RECORD
(Pertemuan 14)

Tujuan:

- Mahasiswa mengetahui definisi *record*.
- Mahasiswa mengetahui cara penggunaan *record*.
- Mahasiswa dapat mempraktekkan tentang *record* dan menggunakannya dalam pemrograman.

Tugas Pendahuluan:

1. Apa yang Anda ketahui tentang *record*?
2. Apakah perbedaan antara *array* dan *record*?

DASAR TEORI

Pada bahasa pemrograman Pascal, terdapat salah satu tipe data yang bernama *record*.*Record* merupakan salah satu tipe data selain array yang dapat menampung lebih dari 1 data. Perbedaan antara array dan *record* adalah setiap data yang tersimpan dalam array harus memiliki tipe data yang sama. Sedangkan dalam *record*, tipe datanya dapat berbeda. Setiap elemen dalam *record* disebut dengan *field*. Berikut cara mendeklarasikan *record*:

```
type nama_record=record
                  nama_field1  : tipe_data1;
                  nama_field2  : tipe_data2;
                  ……..
                  nama_fieldn  : tipe_datan;
                  end;
```

Sebagai contoh:

```
type kelas=record
                  nama         : string[20];
                  alamat       : string[50];
                  nomor_id     : integer;
                  end;
```

KEGIATAN PRAKTIKUM

Salinlah *coding* program di bawah ini:

1. Mendeklarasikan dan menggunakan tipe data *record*

```
program recordmhs;
uses crt ;

type
    mahasiswa = record
       nim    : string[8] ;
       nama   : string[30] ;
       kota   : string[20] ;
       konsentrasi  : string[30] ;
    end;

var
   recMhs : mahasiswa ;

Begin
     clrscr;
     writeln('Masukkan data Anda');
     write('NIM        : '); readln(recMhs.nim) ;
     write('Nama       : '); readln(recMhs.nama) ;
     write('Kota Asal  : '); readln(recMhs.kota) ;
     write('Konsentrasi: '); readln(recMhs.konsentrasi) ;
     writeln;

     writeln('Data yang anda inputkan');
     writeln('NIM        : ', recMhs.nim);
     writeln('Nama       : ', recMhs.nama);
     writeln('Kota Asal  : ', recMhs.kota);
     writeln('Konsentrasi: ', recMhs.konsentrasi);
     readln;
end.
```

Coding di atas menunjukkan cara mendeklarasikan dan menggunakan tipe data *record*. Berdasarkan *coding* di atas, mahasiswa menjadi tipe data dan recMhs menjadi variabel yang memiliki tipe data bentukan yang diberi nama mahasiswa. Jika memperhatikan *coding* di atas dengan seksama, maka dapat dilihat bahwa pemanggilan sebuah *record* dilakukan dengan mengetikkan variabel *record* dan indeks variabel *record* yang dimaksudkan. Sebagai contoh, untuk menampung nilai NIM pada variabel nim yang terdapat pada *record* mahasiswa, maka dilakukan dengan cara mengetikkan **recMhs.nim**. Selain cara tersebut, masih terdapat cara lain untuk memanggil tipe data *record,* yaitu dengan cara mengetikkan **with recMhs**. Contoh:

```
var
      mhs:mahasiswa;

begin
      with mhs do
      begin
      (statements)
      end;
readln;
end.
```

2. Menggabungkan *array* dan *record*

```
program recordmhs;
uses crt ;

type
    mahasiswa = record
       nim    : string[8] ;
       nama   : string[30] ;
       kota   : string[20] ;
       konsentrasi  : string[30] ;
    end;

var
   recMhs : array[1..50]of mahasiswa ;
   byk: byte;
   i,j: integer;

begin
     clrscr;
     write('Masukkan jumlah mahasiswa: '); readln(byk);
     for i:= 1 to byk do
     begin
     write('NIM        : '); readln(recMhs[i].nim) ;
     write('Nama       : '); readln(recMhs[i].nama) ;
     write('Kota Asal  : '); readln(recMhs[i].kota) ;
     write('Konsentrasi: '); readln(recMhs[i].konsentrasi) ;
     writeln;
     end;
     writeln;
     writeln('Data yang telah diinputkan');
     for j:= 1 to byk do
     begin
     writeln('NIM        : ', recMhs[j].nim);
     writeln('Nama       : ', recMhs[j].nama);
     writeln('Kota Asal  : ', recMhs[j].kota);
     writeln('Konsentrasi: ', recMhs[j].konsentrasi);
     writeln;
     end;
     readln;
end.
```

Coding program di atas merupakan *coding* yang sama dengan *coding* pada praktikum nomor 1, hanya saja dengan menambahkan beberapa baris kode *array*, maka *record* mahasiswa di atas dapat menampung lebih dari 1 mahasiswa. Berbeda dengan praktikum nomor 1 yang hanya dapat menampung data untuk 1 mahasiswa saja.*Coding* di atas merupakan contoh penggabungan antara *arra* dan *record.*

TUGAS

1. Buatlah sebuah program yang dapat mengolah data mahasiswa dengan ketentuan:
   - User menginputkan NIM, Nama, Nilai Kuis, Nilai UTS, dan Nilai UAS.
   - Hasil atau output yang diinginkan: NIM, Nama, Nilai Kuis, Nilai UTS, Nilai UAS, dan Nilai Akhir.

- Nilai akhir = 20% dari nilai kuis + 30% dari nilai UTS + 50% dari nilai UAS.