

## MODUL 10 ARRAY

---

### 1. Deskripsi Singkat

Pada modul 10 ini akan dibahas penggunaan array. Penggunaan array salah satunya bertujuan untuk efisiensi penyimpanan dan pengambilan data.

### 2. Tujuan Praktikum

Setelah praktikum pada modul 10 ini diharapkan:

1. Mahasiswa memahami dan menguasai struktur data array
2. Mahasiswa dapat menggunakan struktur data array dengan terampil pada program.

### 3. Material Praktikum

Kegiatan pada modul 10 ini memerlukan material berupa program Pascal.

### 4. Kegiatan Praktikum

#### A. Deklarasi Array 1 Dimensi

**Array** atau **larik** adalah suatu struktur data berindeks yang dapat digunakan untuk menyimpan sejumlah variabel yang memiliki tipe data yang sama.

Misalkan program kita akan menggunakan 10 buah data integer. Jika kita tidak menggunakan array, maka kita perlu mendeklarasikan 10 buah variabel dengan tipe integer seperti contoh berikut:

```
Var
    x1, x2, x3, x4, x5, x6, x7, x8, x9, x10 : integer;
```

Namun jika kita menggunakan array, maka kita cukup mendeklarasikan satu array yang berisi 10 data integer seperti contoh berikut:

```
Var
    x : array[1..10] of integer;
```

Dengan kata lain, untuk menyimpan 10 elemen variabel bertipe integer, kita cukup menggunakan satu identifier `x`.

Array merupakan sebuah struktur data yang statis. Statis artinya ketika *runtime* ukurannya tidak dapat berubah.

Deklarasi umum dari array adalah:

```
NamaArray : array[IndeksAwal..IndeksAkhir] of tipe_data;
```

Nilai IndeksAwal harus lebih kecil atau sama dengan nilai IndeksAkhir. Kita bisa menggunakan tipe data yang memiliki nilai ordinal sebagai IndeksAwal dan IndeksAkhir. Yang banyak digunakan sebagai indeks array adalah bilangan bulat.

Contoh deklarasi array:

```
Var
A: array[0..9] of Real;
B: array[-10..20] of String;
C: array['a'..'j'] of Boolean;
X : array[1..10] of Integer;
```

Simpan potongan program diatas dengan nama **Praktikum10A.pas**. Lengkapi bagian-bagian lainnya sehingga menjadi program yang utuh.

Untuk mengakses setiap elemen array, kita menggunakan indeksnya. Misal pada deklarasi variabel X di atas, indeks array dimulai dari 1 sampai 10 yang nantinya dituliskan pada kurung siku [ ]. Tambahkan potongan program di bawah ini pada **Praktikum10A.pas**, lalu jalankan. Apakah output yang dihasilkan?

```
...
X[1] := 10; {array X indeks pertama kita isi nilai 10}
X[2] := X[1] - 5; {array X indeks kedua kita isi nilai
array X indeks pertama dikurangi 5 yang mana hasilnya adalah
5}
X[3] := X[2] + X[1];
Writeln(X[3]);
...
```

Apakah kita bisa mengakses indeks array di luar range yang telah dideklarasikan? Tambahkan potongan program berikut masih pada **Praktikum10A.pas**. Output apakah yang dihasilkan?

```
Writeln(X[13]);
Writeln(A[10]);
```

## B. Penggunaan Array 1 Dimensi

Penggunaan array bertujuan untuk efisiensi manajemen data dengan memanfaatkan indeks array. Berikut adalah contoh pencarian nilai terbesar dari 3 bilangan tanpa menggunakan array:

```
if x > y then
  if x > z then

    terbesar := x
```

```

    else

        terbesar := z
else
    if y > z then

        terbesar := y
else
    terbesar := z;

```

Bagaimana jika bilangannya ada 100? Atau ada 1000? Jika kita menggunakan array, maka berapapun jumlah datanya tidak menjadi masalah. Perhatikan program berikut:

```

Var
    X : array[1..10] of Integer;
    Terbesar: Integer;
    i : Integer;
Begin
    For i:=1 to 10 do
        Begin
            Write('Input data ke-', i, '= ');
            Readln(X[i]);
        End;

        Terbesar := X[1];
        For i:=2 to 10 do
            Begin
                If X[i] > Terbesar then Terbesar := X[i];
            End;
        Writeln(Terbesar);
        Readln;
    End.

```

Simpan program dengan nama **Praktikum10B.pas**. Jalankan program, apakah output yang dihasilkan program? Jelaskan cara kerja program!

### C. Array 2 Dimensi

Pembahasan sebelumnya adalah untuk array satu dimensi. Array yang memiliki dimensi lebih dari 1 disebut array multidimensi. Array yang memiliki 2 dimensi disebut dengan array 2 dimensi.

Setiap elemen array dapat diakses menggunakan **indeks[baris, kolom]**

Misal kita mendeklarasikan array 2 dimensi sebagai berikut:

Var

```
X : array[1..3, 1..4] of integer;
```

Maka jumlah data yang disimpan pada variabel X adalah sebanyak baris kali kolom atau  $3 \times 4 = 12$ . Kira-kira memori akan menyimpan variabel X seperti berikut:

**X**

Baris/Kolom	1	2	3	4
1	[1,1]	[1,2]	[1,3]	[1,4]
2	[2,1]	[2,2]	[2,3]	[2,4]
3	[3,1]	[3,2]	[3,3]	[3,4]

Setiap elemen array dapat diakses menggunakan **indeks[baris, kolom]**. Misalnya:

Var

```
X : array[1..3, 1..4] of integer;
```

Begin

```
X[1,3] := 10;
```

```
Writeln(X[1,3]); {output 10}
```

End.

Array 2 dimensi banyak digunakan untuk operasi matriks. Perhatikan program untuk penjumlahan matriks berikut ini. Simpan program berikut dengan nama **Praktikum10C.pas**, lalu jalankan program.

Program penjum\_matriks;

var

```
a,b,c: array[1..3, 1..3] of integer;
```

```
i,j: integer;
```

begin

```
writeln('buat matriks A');
```

```
for i:=1 to 3 do
```

```
  for j:=1 to 3 do
```

```
    begin
```

```
      write('[' , i , ',' , j , ']=');
```

```
      readln(a[i,j]);
```

```
    end;
```

```
writeln('buat matriks B');
```

```
for i:=1 to 3 do
```

```
  for j:=1 to 3 do
```

```
    begin
```

```
      write('[' , i , ',' , j , ']=');
```

```
      readln(b[i,j]);
```

```

        end;

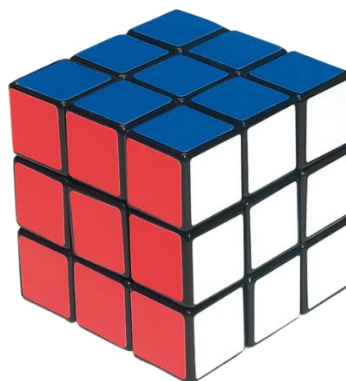
        writeln('Matriks A');
        for i:=1 to 3 do
            begin
                for j:=1 to 3 do
                    write(a[i,j], ' ');
                writeln;
            end;
        writeln('Matriks B');
        for i:=1 to 3 do
            begin
                for j:=1 to 3 do
                    write(b[i,j], ' ');
                writeln;
            end;
        readln;
    end.

```

Saat program di atas dijalankan, program menginput matriks A, menginput matriks B, kemudian menampilkan matriks A dan matriks B. Modifikasi program di atas dengan menambahkan potongan program untuk menampilkan matriks C sebagai hasil penjumlahan dari matriks A+B (tambahkan sebelum readln).

#### D. Array Multi Dimensi

Array dapat kita deklarasikan dalam sejumlah dimensi tergantung kebutuhan kita. Misal sebuah array yang memiliki 3 dimensi dengan masing-masing dimensinya adalah 3, dapat digambarkan seperti sebuah rubric cube.



Rubric Cube adalah array 3x3x3

Contoh deklarasinya adalah sebagai berikut:

```

Var
X : array[1..3, 1..3, 1..3] of integer;

```

Maka variabel X memiliki elemen sebanyak  $3 \times 3 \times 3 = 27$  buah. Untuk mengakses tiap elemen, indeksnya sebanyak 3 seperti contoh berikut ini.

```
Var
X : array[1..3, 1..3, 1..3] of integer;
Begin
    X[1,1,1] := 10;
    Writeln(X[1,1,1]); {output 10}
End.
```

Terdapat 27 elemen pada X yaitu:

X[1,1,1] , X[1,2,1] , X[1,3,1], X[1,1,2] , X[1,2,2] , X[1,3,2],  
X[1,1,3] , X[1,2,3] , X[1,3,3], X[2,1,1] , X[2,2,1] , X[2,3,1],  
X[2,1,2] , X[2,2,2] , X[2,3,2], X[2,1,3] , X[2,2,3] , X[2,3,3],  
X[3,1,1] , X[3,2,1] , X[3,3,1], X[3,1,2] , X[3,2,2] , X[3,3,2],  
X[3,1,3] , X[3,2,3] , X[3,3,3]

Dapatkah anda bayangkan array 4 dimensi, 5 dimensi, dan seterusnya?

Untuk array multidimensi terdapat cara lain untuk mengakses indeks elemen array selain menggunakan koma yaitu menggunakan kurung siku terpisah. Misalnya:

```
Var

    X : array[1..3, 1..4] of integer;
Begin

    X[1][3] := 10;

    Writeln(X[1][3]); {output 10}
End.
```

Buka Kembali program **Praktikum10C.pas**, lalu simpan ulang dengan nama **Praktikum10D.pas**. Modifikasi Array yang terdapat pada Praktikum10C.pas sehingga matriks A, B, dan C menjadi rubik berukuran  $3 \times 3 \times 3$ . Jalankan program dan pastikan hasil penjumlahan rubik yang ditampilkan sudah benar.

### E. Array sebagai Tipe Data Bentukan

Array dapat juga kita nyatakan pada bagian `type`, untuk selanjutnya kita gunakan pada bagian `var`. Perhatikan contoh berikut:

Type

```
Matrix = array[1..3, 1..3] of real;
```

Var

```
a,b,c : Matrix;
```

Buka Kembali program **Praktikum10C.pas**, lalu simpan ulang dengan nama **Praktikum10E.pas**. Deklarasikan Array berukuran 3 X 3 menjadi tipe data bentukan dengan nama `Matrix` seperti contoh di atas. Modifikasi deklarasi matriks A, B, dan C menggunakan tipe data bentukan yang baru saja dibuat. Jalankan program. Pastikan tidak ada error dan hasil penjumlahan matriks yang ditampilkan sudah sesuai.

## 5. Responsi

Kerjakan sesuai dengan yang dijelaskan pada bagian Kegiatan Praktikum. Simpan tangkapan layar hasil pekerjaan Anda untuk masing-masing kegiatan praktikum dalam file docx. Simpan ulang file tersebut dalam format pdf, dan beri nama dengan format `<<kelas>>_modul10_<<nim>>.pdf`, contoh: **1KS1\_modul10\_192191234.pdf**. Kumpulkan file tersebut sebagai responsi melalui Google Classroom