

PART 19

PENGERTIAN DAN CARA PENGGUNAAN TIPE DATA ARRAY PASCAL



PENGERTIAN TIPE DATA ARRAY PASCAL

Tipe data array adalah tipe data bentukan yang terdiri dari kumpulan tipe data lain. Daripada membuat 10 variabel yang terdiri dari *nama1*, *nama2*, *nama3*, dst, akan lebih efisien jika variabel nama ini disimpan ke dalam array.

Sebagai contoh, perhatikan kode program berikut ini:

```
1  program tipe_array;
2  uses crt;
3  var
4      nilai1, nilai2, nilai3, nilai4: integer;
5  begin
6      clrscr;
7
8      nilai1:= 23;
9      nilai2:= 13;
10     nilai3:= 98;
11     nilai4:= 106;
12
13     writeln('nilai1: ',nilai1);
14     writeln('nilai2: ',nilai2);
15     writeln('nilai3: ',nilai3);
16     writeln('nilai4: ',nilai4);
17
18     readln;
19 end.
```

Dalam contoh tersebut saya membuat 4 variabel: *nilai1*, *nilai2*, *nilai3* dan *nilai4*. Keempat variabel ini bertipe integer.

```
1  program tipe_array;
2  uses crt;
3  var
4      nilai1, nilai2, nilai3, nilai4: integer;
5  begin
6      clrscr;
7
8      nilai1:= 23;
9      nilai2:= 13;
10     nilai3:= 98;
11     nilai4:= 106;
12
13     writeln('nilai1: ',nilai1);
14     writeln('nilai2: ',nilai2);
15     writeln('nilai3: ',nilai3);
16     writeln('nilai4: ',nilai4);
17
18     readln;
19 end.
```

Dalam contoh tersebut saya membuat 4 variabel: *nilai1*, *nilai2*, *nilai3* dan *nilai4*. Keempat variabel ini bertipe *integer*.

Tidak ada yang salah dari kode program diatas. Tapi bayangkan apabila kita ingin menyimpan lebih dari 4 nilai, bagaimana jika 10 atau 100 nilai? Tentu tidak efisien jika kita harus membuat variabel *nilai1*, *nilai2*, *nilai3*... sampai dengan *nilai100*. Untuk hal inilah tipe data array lebih cocok digunakan.

CARA PENGGUNAAN TIPE DATA ARRAY PASCAL

Untuk membuat tipe data array di pascal, kita harus menentukan seberapa banyak element array yang ingin dibuat. **Element** adalah sebutan untuk 'anggota' / isi dari array. Sebagai contoh, untuk membuat 10 element array bertipe integer saya bisa menggunakan kode berikut:

```
1 | var  
2 |   nilai: array[0..9] of integer;
```

Sekarang, variabel 'nilai' berisi array dengan 10 element bertipe integer. Perhatikan angka 0..9, ini berarti saya membuat element array dari element 0, element 1, element 2, element 3,.. hingga element 9 (total terdapat 10 element).

Bagaimana cara mengakses element ini? kita bisa mengaksesnya melalui nomor **index**. **Index** adalah urutan element di dalam sebuah array. Sebagai contoh, untuk mengakses element ke - 2, kita bisa menulis: `nilai[2]`. Untuk mengakses element ke-6, bisa menggunakan: `nilai[6]`.

Berikut contoh kode program pascal cara penggunaan tipe data array:

```
1  program tipe_array;
2  uses crt;
3  var
4      nilai: array[0..9] of integer;
5
6      begin
7          clrscr;
8
9          nilai[0]:= 23;
10         nilai[1]:= 13;
11         nilai[2]:= 98;
12         nilai[3]:= 106;
13
14         writeln('nilai0: ',nilai[0]);
15         writeln('nilai1: ',nilai[1]);
16         writeln('nilai2: ',nilai[2]);
17         writeln('nilai3: ',nilai[3]);
18
19         readln;
20     end.
```

Pada contoh diatas, saya membuat variabel '*nilai*' sebagai *array* yang berisi 10 element *integer*. Di dalam variabel '*nilai*' ini, index array dimulai dari 0 hingga 9, karena saya menuliskannya dengan `array[0..9] of integer`. Jika anda ingin membuat 100 element array, bisa menuliskannya sebagai `array[0..99] of integer`.

Walaupun saya membuat 10 element, tapi kita tidak harus mengisi semua element ini. Pada contoh tersebut, saya hanya mengisi 4 element. Bagaimana dengan element lainnya? ini akan menggunakan nilai *default* (bawaan) pascal, biasanya berisi angka 0 untuk tipe data integer.

Walaupun saya membuat 10 element, tapi kita tidak harus mengisi semua element ini. Pada contoh tersebut, saya hanya mengisi 4 element. Bagaimana dengan element lainnya? ini akan menggunakan nilai *default* (bawaan) pascal, biasanya berisi angka 0 untuk tipe data integer.

Selain itu, kita juga tidak harus mengisinya secara berurutan. Kita bisa mengisi element-element array ini secara acak, selama masih dalam batas yang ditetapkan. Berikut contohnya:

```
1  program tipe_array;
2  uses crt;
3  var
4      nilai: array[0..9] of integer;
5
6      begin
7          clrscr;
8
9          nilai[3]:= 23;
10         nilai[9]:= 13;
11         nilai[2]:= 98;
12         nilai[0]:= 106;
13
14         writeln('nilai3: ',nilai[3]);
15         writeln('nilai9: ',nilai[9]);
16         writeln('nilai2: ',nilai[2]);
17         writeln('nilai0: ',nilai[0]);
18
19         writeln('nilai1: ',nilai[1]);
20         writeln('nilai7: ',nilai[7]);
21
22         readln;
23     end.
```

Kita tidak hanya bisa membuat array bertipe integer saja, tapi juga bisa menggunakan tipe lain seperti `real`, `char` atau `string`. Berikut contohnya:

```
1  program tipe_array;
2  uses crt;
3  var
4      kata: array[20..29] of string[20];
5  begin
6      clrscr;
7
8      kata[24] := 'Sedang ';
9      kata[25] := 'belajar pascal ';
10     kata[26] := 'di ';
11     kata[27] := 'Duniailkom.com';
12
13     write(kata[24]);
14     write(kata[25]);
15     write(kata[26]);
16     write(kata[27]);
17
18     readln;
19 end.
```

Saya membuat variabel kata dengan array berjumlah 10 element yang masing-masing isinya adalah `string[20]`.

Anda bisa lihat bahwa saya menggunakan penomoran array mulai dari 20 hingga 29. Ini tidak menjadi masalah, selama kita juga mengaksesnya dengan index yang sesuai. Berikut hasil yang didapat: