
Pengertian Fixed Array Pascal

Fixed array adalah sebutan untuk array yang jumlah element-nya bersifat tetap. Contoh-contoh kode program dalam 2 tutorial sebelum ini, saya selalu menggunakan *fixed array*, dimana ukuran array sudah ditentukan pada saat array dibuat.

Sebagai contoh, kode program:

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

Berarti kita membuat array integer yang terdiri dari 10 element (0, 1, 2, 3 ... 9).

Contoh lain, kode program:

```
1 var
2  kata: array[0..2,0..2] of string[20];
```

Berarti kita membuat array string dengan 9 element, yakni 3 untuk dimensi pertama, dan 3 untuk dimensi kedua ($3 \times 3 = 9$). Mulai dari **kata[0,0]**, **kata[0,1]** hingga **kata [2,2]**.

Dalam kedua contoh ini ukuran array sudah diketahui pada saat array dideklarasikan.

Pengertian Dynamic Array Pascal

Dynamic array adalah tipe array yang jumlah element-nya belum ditentukan (masih nol). Contoh penulisannya adalah sebagai berikut:

```
1 var
2  nilai: array of integer;
3  kata: array of string[20];
```

Jadi, berapa jumlah element di dalam array ini? Jumlah element akan ditentukan menggunakan fungsi **setlength()** di bagian kode program utama. Sebagai contoh, jika saya ingin variabel '**nilai**' memiliki 10 element, saya tinggal menulis **setlength(nilai, 10)**. Langsung saja kita masuk kedalam contoh kode programnya:

```
program tipe_array;
1 uses crt;
2 var
3  nilai: array of integer;
4 begin
5  clrscr;
```

```

6  setlength(nilai,10);
7
8  nilai[0]:= 1;
9  nilai[6]:= 2;
10 nilai[9]:= 4;
11
12 writeln('nilai 0: ',nilai[0]);
13 writeln('nilai 6: ',nilai[6]);
14 writeln('nilai 9: ',nilai[9]);
15
16 readln;
17 end.
18

```



Pada saat variabel '**nilai**' dideklarasikan sebagai array, kita belum mengetahui berapa jumlah element array tersebut. Barulah di dalam kode program, fungsi **setlength(nilai,10)** akan menetapkan bahwa array '**nilai**' akan berisi **10 element**. Perhitungan 10 element ini dimulai dari index 0 hingga 9.

Kalau jumlah element array tetap ditulis, dimana letak dinamisnya?

Di dalam pascal, jumlah element suatu array memang harus ditulis, tapi dalam array dinamis, jumlah element ini ditentukan di dalam kode program utama, bukan pada saat di deklarasikan.

Oleh karena itu nantinya kita bisa menentukan jumlah element array berdasarkan kondisi tertentu (menggunakan **struktur logika IF**), misalnya jika syarat terpenuhi, set jumlah element 1000, jika tidak set jumlah element array menjadi 10.

Jika anda telah mempelajari **pointer**, array dinamis juga bisa dibuat menggunakan pointer, tapi caranya lebih rumit daripada menggunakan fungsi **setlength()**.

Untuk array 2 dimensi, kita tinggal men-set 2 buah batasan element array, seperti contoh berikut:

```

1 program tipe_array;
2 uses crt;
3 var
4   nilai: array of array of integer;
5 begin
6   clrscr;
7   setlength(nilai,4,4);
8
9   nilai[0,0]:= 34;
10  nilai[2,3]:= 99;

```

```
10 nilai[3,0]:= 15;
11
12 writeln('nilai [0,0]: ',nilai[0,0]);
13 writeln('nilai [2,3]: ',nilai[2,3]);
14 writeln('nilai [3,0]: ',nilai[3,0]);
15
16 readln;
17 end.
18
```



Perhatikan cara penulisan variabel nilai. Kita membuatnya sebagai **array of array of integer**, yang berarti ini adalah **array 2 dimensi** dengan tipe **integer**. Untuk men-set batasan jumlah element array, saya menggunakan fungsi **setlength(nilai,4,4)**. Fungsi ini akan membuat variabel 'nilai' berisi 25 element (hasil dari $5 \times 5 = 25$).

Membuat Array dari Tipe Data Standar

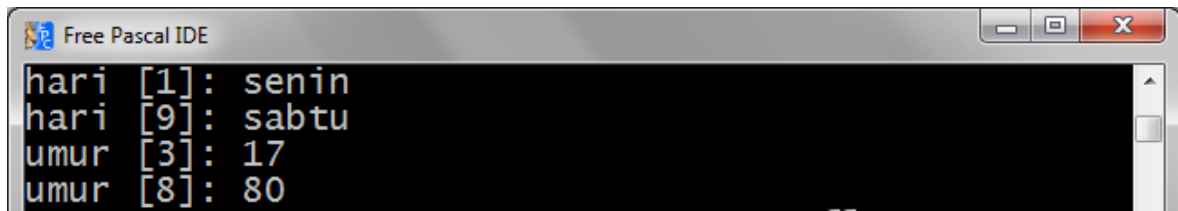
Dalam tutorial tentang array sebelumnya, secara tidak langsung kita telah membuat array menggunakan tipe data standar pascal, yakni **integer**, **string** atau **char**. Ditambah dengan tipe data **boolean** dan **real**, kelimaanya merupakan tipe data standar bawaan pascal (atau dikenal juga dengan **tipe data skalar**).

Membuat Array dari Tipe Data Bentukan

Tipe data bentukan (*user-defined data types*) adalah sebutan untuk tipe data yang dibentuk sendiri. Hingga tutorial ini, kita telah membahas 2 buah tipe data bentukan, yakni **enumeration** dan **subrange**.

Uniknya, tipe data bentukan ini juga bisa menggunakan sebagai element array. Langsung saja kita simak contoh kode program berikut ini:

```
1
2 program tipe_array;
3 uses crt;
4 type
5   nama_hari= (senin,selasa,rabu,kamis,jumat,sabtu,minggu);
6   usia= 0..99;
7 var
8   hari: array[0..9] of nama_hari;
9   umur: array[0..9] of usia;
10 begin
11   clrscr;
12
13   hari[1]:= senin;
14   hari[9]:= sabtu;
15
16   umur[3]:= 17;
17   umur[8]:= 80;
18
19   writeln('hari [1]: ',hari[1]);
20   writeln('hari [9]: ',hari[9]);
21   writeln('umur [3]: ',umur[3]);
22   writeln('umur [8]: ',umur[8]);
23
24   readln;
25 end.
```

A screenshot of the Free Pascal IDE window. The title bar says 'Free Pascal IDE'. The main text area shows four lines of code: 'hari [1]: senin', 'hari [9]: sabtu', 'umur [3]: 17', and 'umur [8]: 80'. The text is white on a black background. The window has standard Windows-style controls (minimize, maximize, close) in the top right corner.

```
hari [1]: senin
hari [9]: sabtu
umur [3]: 17
umur [8]: 80
```

Dalam kode diatas saya membuat 2 buah tipe data bentukan, yakni **nama_hari** yang merupakan tipe data *enumeration* dan **umur** yang merupakan tipe data *subrange*. Setelah deklarasi kedua tipe data ini, saya kemudian 'memasukkannya' ke dalam variabel **hari** dan **umur**. Keduanya merupakan array dari tipe data **nama_hari** dan **usia**.

Jika anda kurang paham tentang maksud keduanya, silahkan pelajari [tutorial tipe data enumeration pascal](#) dan [tutorial subrange pascal](#).

Selebihnya, cara penggunaannya sama seperti biasa, kecuali kita dibatasi kepada batasan yang dibuat. Sebagai contoh, variabel **hari** hanya bisa diisi dengan nama-nama hari yang telah dideklarasikan pada bagian type. Untuk variabel **umur**, juga tidak bisa diisi dengan angka lebih dari 100.

Pengertian Tipe Data Record Pascal

Tipe data record adalah tipe data khusus yang komponennya terdiri dari berbagai jenis tipe data lain. Sebuah **record** berisi beberapa *variabel lain* yang 'dipaketkan'. Konsep struktur data seperti ini sedikit mirip dengan konsep **object** dalam bahasa pemrograman modern (walaupun di dalam pascal juga terdapat konsep tentang object).

Record juga mirip dengan **array**, dimana kita bisa membuat sebuah variabel yang berisi berbagai element. Perbedaannya, *record* bisa menampung berbagai jenis tipe data, tidak hanya 1 tipe data seperti *array*.

Cara Penggunaan Tipe Data Record Pascal

Untuk membuat **record** di dalam *pascal*, kita bisa mendeklarasikannya di bagian **type** atau di bagian **var**. Jika digunakan di dalam bagian **var**, cara penulisannya adalah sebagai berikut:

```
1 var
2   nama_record:
3     record
4       variabel: tipe_data;
5       variabel: tipe_data;
6       variabel: tipe_data;
7   end;
```

Perhatikan bahwa untuk membuat *record*, diawali dengan **nama_record**. Ini adalah variabel yang akan menampung seluruh isi record. Setelah itu, pembuatan 'isi' *record* berada di antara perintah **record** dan **end**. Disinilah seluruh variabel yang menjadi 'isi' record di defenisikan.

Sebagai contoh, saya akan membuat sebuah *record* '**siswa**' yang terdiri dari **nama**, **umur**, **sekolah**, dan **kota**. Berikut cara penulisannya:

```
1 var
2   siswa: record
3     nama: string[20];
4     umur: integer;
5     sekolah: string[30];
6     kota: string[20];
7   end;
```

Sekarang, variabel '**siswa**' adalah tipe data **record** yang terdiri dari *nama*, *umur*, *sekolah*, dan *kota*. **Bagaimana cara mengakses record ini?**

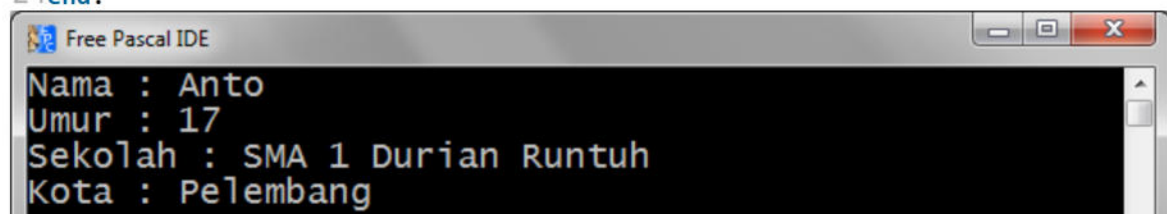
Untuk mengakses variabel di dalam record (atau disebut juga sebagai **field**), digunakan karakter titik (.), seperti contoh berikut:

```
1siswa.nama:= 'Anto';
2siswa.umur:= 17;
3siswa.sekolah:= 'SMA 1 Durian Runtuh';
4siswa.kota:= 'Pelembang';
```

Dalam kode diatas, saya mengisi beberapa data kedalam record '**siswa**'. Perhatikan bahwa isi masing-masingnya harus bersesuaian dengan tipe data sewaktu kita merancang *record*. Untuk **nama**, bisa diisi dengan **string[20]**, sedangkan untuk **umur** bisa diisi dengan angka bulat (**integer**).

Berikut kode lengkap cara penggunaan struktur record di dalam pascal:

```
1
2program tipe_record;
3uses crt;
4var
5  siswa: record
6      nama: string[20];
7      umur: integer;
8      sekolah: string[30];
9      kota: string[20];
10     end;
11begin
12  clrscr;
13
14  siswa.nama:= 'Anto';
15  siswa.umur:= 17;
16  siswa.sekolah:= 'SMA 1 Durian Runtuh';
17  siswa.kota:= 'Pelembang';
18
19  writeln('Nama : ',siswa.nama);
20  writeln('Umur : ',siswa.umur);
21  writeln('Sekolah : ',siswa.sekolah);
22  writeln('Kota : ',siswa.kota);
23
24  readln;
25end.
```



Pada kode diatas, saya membuat sebuah *record* **siswa**, mengisinya dengan beberapa data, kemudian menampilkan isi record menggunakan perintah **writeln**.