

Cara Mengakses Field Record dalam Pascal

Seperti yang telah kita pelajari, untuk mengakses isi *record*, atau dikenal dengan istilah **field record**, kita menggunakan tanda titik. Sebagai contoh, jika saya memiliki record '**siswa**' berikut:

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

Untuk mengakses **field** record siswa tersebut harus ditulis dengan format **nama_record.field_record**, seperti berikut ini:

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

Perintah **siswa.nama:= 'Anto'** berarti saya ingin mengisi *field* **nama** dari record **siswa** dengan nilai '**Anto**'.

Fungsi perintah With dalam Pascal

Selain cara pengaksesan diatas, pascal menyediakan *keyword* atau perintah **with**. Perintah **with** ini bisa digunakan untuk menyederhanakan pengaksesan isi record.

Sebagai contoh, proses pengisian record siswa sebelumnya bisa ditulis menjadi:

```
1 with siswa do
2   begin
3       nama:= 'Anto';
4       umur:= 17;
5       sekolah:= 'SMA 1 Durian Runtuh';
6       kota:= 'Pelembang';
7   end;
```

Kode program diatas diawali dengan perintah "**with siswa do**" ini merupakan instruksi kepada pascal bahwa kita ingin akan mengakses isi dari record **siswa**. Setelah perintah **with**, kita harus membatasi block kode program dengan perintah **begin** hingga **end**; Diantara blok inilah pengaksesan record dengan **with** dilakukan.

Perhatikan cara pengaksesan isi/field dari record siswa. Saya tidak perlu membuat nama variabel record "**siswa.nama**", tetapi cukup "**nama**" saja. Ini akan

memudahkan penulisan kode program, terutama jika record yang diisi cukup banyak.

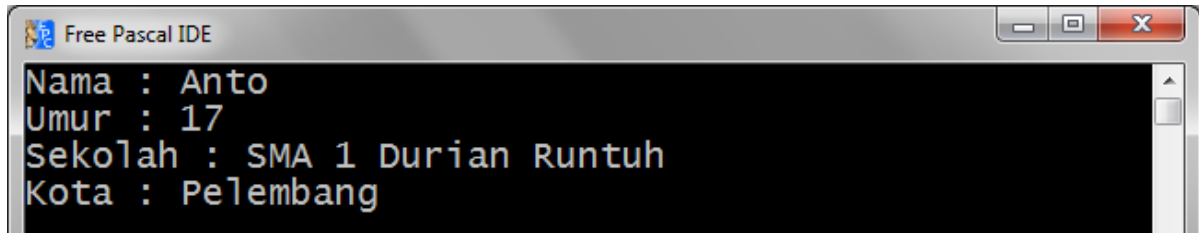
Berikut contoh lengkap kode program pascal dengan perintah with:

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

Lebih jauh lagi, kita bisa melingkupi seluruh kode program dengan perintah **with**.

```
1 program tipe_record;
2 uses crt;
3 var
4   siswa: record
5     nama: string[20];
6     umur: integer;
7     sekolah: string[30];
8     kota: string[20];
9   end;
10 begin
11   clrscr;
12   with siswa do
13     begin
14       nama:= 'Anto';
15       umur:= 17;
16       sekolah:= 'SMA 1 Durian Runtuh';
17       kota:= 'Pelembang';
```

```
17
18     writeln('Nama : ',nama);
19     writeln('Umur : ',umur);
20     writeln('Sekolah : ',sekolah);
21     writeln('Kota : ',kota);
22 end;
23
24 readln;
25 end.
26
```



Penggunaan perintah **with** seperti ini hanya bisa digunakan selama di dalam block kode program tersebut kita tidak mengakses record lain. Jika butuh untuk mengakses record lain, kita harus keluar dari block with terlebih dahulu (menggunakan perintah **end;**).

Perintah **with** yang kita pelajari disini cocok digunakan untuk mempermudah pengaksesan record, terutama jika record tersebut memiliki banyak *field*. Dalam tutorial berikutnya kita akan mempelajari cara membuat array dari struktur data record.

Cara Membuat Tipe Data Record Bentukan

Pada 2 tutorial **record pascal** sebelum ini, saya membuat **record** di bagian **var** kode program pascal. Kali ini saya akan memindahkannya ke bagian **type**. Dengan demikian, record akan menjadi tipe data bentukan yang nantinya bisa digunakan sebagai basis dari **array**.

Sebagai contoh, saya kembali menggunakan **record** data '**siswa**'. Sebelumnya kita mendefenisikan record siswa di bagian **var**, seperti contoh berikut ini:

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

Saya akan memindahkannya kepada bagian **type**, kemudian mengaksesnya dari bagian **var**:

```
1
2 type
3   rec_siswa =
4   record
5       nama: string[20];
6       umur: integer;
7       sekolah: string[30];
8       kota: string[20];
9   end;
10 var
11   siswa: rec_siswa;
```

Tampak sedikit perubahan, dimana kali ini saya menggunakan variabel '**rec_siswa**' sebagai penampung tipe data bentukan *record*, baru kemudian 'memasukkannya' ke dalam variabel '**siswa**'.

Berikut kode program lengkap cara pembuatan record di bagian **type** pascal:

```
1
2 program tipe_record;
3 uses crt;
4
5 type
6   rec_siswa =
7   record
8       nama: string[20];
9       umur: integer;
10      sekolah: string[30];
11      kota: string[20];
12   end;
```

```

13 var
14   siswa: rec_siswa;
15
16 begin
17   clrscr;
18   with siswa do
19     begin
20       nama:= 'Anto';
21       umur:= 17;
22       sekolah:= 'SMA 1 Durian Runtuh';
23       kota:= 'Pelembang';
24
25       writeln('Nama : ',nama);
26       writeln('Umur : ',umur);
27       writeln('Sekolah : ',sekolah);
28       writeln('Kota : ',kota);
29     end;
30   readln;
31 end.
32

```

Selain memindahkan *record* 'siswa' ke bagian **type**, saya juga menggunakan perintah **with** untuk memudahkan pengaksesan isi array. Ini sudah kita bahas pada tutorial sebelumnya: Fungsi dan cara penggunaan perintah with pada record.

Cara Membuat Array dari Struktur Data Record

Setelah memindahkan *record* ke bagian **type**, kita bisa membuat **array** berdasarkan record ini. Melanjutkan contoh record siswa, cara pembuatan array-nya adalah sebagai berikut:

```

1 var
2   siswa: array [0..5] of rec_siswa;

```

Kode program ini akan membuat variabel **siswa** berisi 5 element array yang setiap element-nya adalah **record**. Terasa membingungkan?

Untuk mengakses element ini, kita juga harus menggunakan **index** array.

Sebagai contoh, untuk mengakses element pertama, yang memiliki index 0, kita bisa menulis sebagai berikut:

```

1   siswa[0].nama:= 'Anto';
2   siswa[0].umur:= 17;
3   siswa[0].sekolah:= 'SMA 1 Durian Runtuh';
4   siswa[0].kota:= 'Pelembang';

```

Bagaimana untuk element kedua (ber-index 1)?

```

1  siswa[1].nama:= 'Santi';
2  siswa[1].umur:= 18;
3  siswa[1].sekolah:= 'SMA 1 Merdeka';
4  siswa[1].kota:= 'Makasar';

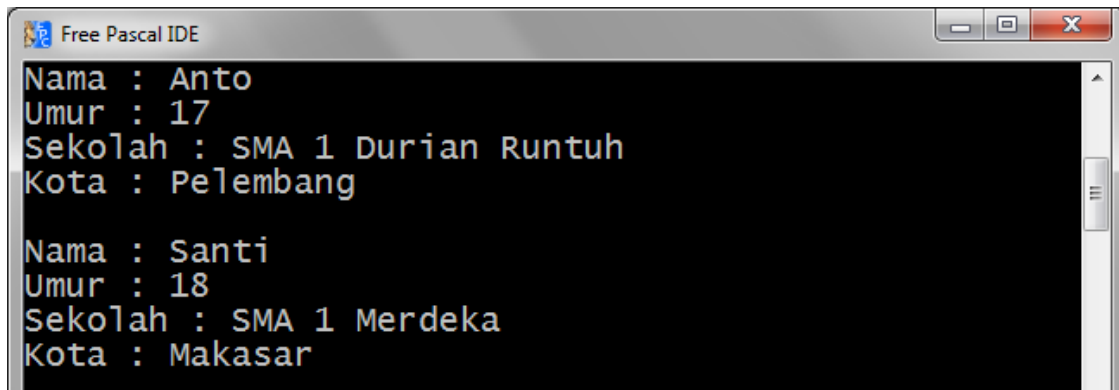
```

Seperti yang terlihat, kita mengkombinasikan pengaksesan *index* array dengan record. Berikut kode program lengkap cara menggabungkan array dengan record ini:

```

1
2
3 program tipe_record_array;
4 uses crt;
5
6 type
7   rec_siswa =
8     record
9       nama: string[20];
10      umur: integer;
11      sekolah: string[30];
12      kota: string[20];
13    end;
14 var
15   siswa: array [0..5] of rec_siswa;
16
17 begin
18   clrscr;
19
20   siswa[0].nama:= 'Anto';
21   siswa[0].umur:= 17;
22   siswa[0].sekolah:= 'SMA 1 Durian Runtuh';
23   siswa[0].kota:= 'Pelembang';
24
25   writeln('Nama : ',siswa[0].nama);
26   writeln('Umur : ',siswa[0].umur);
27   writeln('Sekolah : ',siswa[0].sekolah);
28   writeln('Kota : ',siswa[0].kota);
29
30   writeln;
31
32   siswa[1].nama:= 'Santi';
33   siswa[1].umur:= 18;
34   siswa[1].sekolah:= 'SMA 1 Merdeka';
35   siswa[1].kota:= 'Makasar';
36
37   writeln('Nama : ',siswa[1].nama);
38   writeln('Umur : ',siswa[1].umur);
39   writeln('Sekolah : ',siswa[1].sekolah);
40   writeln('Kota : ',siswa[1].kota);
41
42   readln;
43 end.

```

A screenshot of the Free Pascal IDE window. The window has a title bar with the text "Free Pascal IDE" and standard window controls (minimize, maximize, close). The main area is a black console with white text. The text is organized into two groups, each representing a person's data. The first group is for "Anto" and the second for "Santi". Each group contains four lines of data: Name, Age, School, and City.

```
Free Pascal IDE
Nama : Anto
Umur : 17
Sekolah : SMA 1 Durian Runtuh
Kota : Palembang

Nama : Santi
Umur : 18
Sekolah : SMA 1 Merdeka
Kota : Makasar
```