
8 RECORD

Untuk merepresentasikan sebuah objek, sering tipe data dasar seperti integer, real, boolean, char, tidak dapat memenuhinya. Oleh karena itu, dibentuklah tipe bentukan yang merupakan gabungan dari beberapa tipe data dasar atau dari tipe bentukan lainnya. Salah satu tipe bentukan tersebut adalah *record*.

Record adalah salah satu tipe data terstruktur (*structured data type*) bentukan yang setiap recordnya terdiri dari beberapa elemen yang disebut *field*. Setiap *field* menggambarkan informasi tertentu dan tipe data pada masing-masing *field* dapat berbeda-beda namun sudah dikenal baik itu tipe dasar atau tipe bentukan lainnya.

| | | | | |
|----------------|----------------|----------------|-----|----------------|
| <i>field 1</i> | <i>field 2</i> | <i>field 3</i> | ... | <i>field n</i> |
|----------------|----------------|----------------|-----|----------------|

Ilustrasi sebuah record

Sebagai contoh, di dalam matematika untuk menggambarkan sebuah titik pada diagram kartesian digunakan dua bilangan integer atau real yaitu untuk menunjukkan koordinat titik yang ditunjuk pada sumbu x dan sumbu y. Untuk merepresentasikan tanggal juga digunakan tipe bentukan yang terdiri dari hari, bulan, dan tahun yang masing-masing bertipe integer. Data pegawai juga terdiri dari beberapa elemen seperti nama, tanggal lahir, dan alamat. Untuk menjawab semua kebutuhan pada beberapa contoh tersebut, dibuatlah tipe bentukan yang disebut record.

DEKLARASI RECORD

Seperti halnya tipe data lain, tipe data record juga harus dideklarasikan terlebih dahulu di bagian kamus jika kita ingin menggunakan sebuah variabel yang bertipe record pada bagian deskripsi algoritmanya. Deklarasi record pada algoritma adalah sebagai berikut :

```
type nama_record : record < nama_field1 : tipe_field1,
                             nama_field2 : tipe_field2
                             .....
                             nama_fieldn : tipe_fieldn
                             >
```

Contoh 1 : Type Titik {menyatakan absis dan ordinat pada diagram kartesian}

Titik dalam koordinat kartesian dinyatakan sebagai (x,y) dengan x adalah nilai absis dan y adalah nilai ordinat. Kita dapat menyatakan titik sebagai record dengan (x,y) sebagai field

| | |
|----------|----------|
| <i>x</i> | <i>y</i> |
|----------|----------|

```
type Titik : record < x : real,      {absis}
                    y : real      {ordinat}
                    >
```

Jika dideklarasikan sebuah variabel T sebagai berikut :

T : Titik {artinya : mendefinisikan variabel T bertipe Titik}

Maka cara mengacu atau mengakses nilai elemen yang tersimpan pada T yang telah terdefinisi adalah :

T.x dan T.y

Jika dideklarasikan sebuah variabel W sebagai berikut :

W : Waktu {artinya : mendeklarasikan variabel W bertipe Waktu}

Maka cara mengacu atau mengakses nilai elemen yang tersimpan pada W yang telah terdefinisi adalah :

W.hh , W.mm, dan dan W.ss

Contoh :

Kamus

```
type Waktu : record < hh : integer[0..23],  
                        mm : integer[0..59],  
                        ss : integer[0..59]  
>
```

W : Waktu

Deskripsi

```
W.hh ← 10  
W.mm ← 30  
W.ss ← 50  
read(W.hh, W.mm, W.ss)  
write(W.hh, W.mm, W.ss)
```

Contoh 4 : Record untuk pengolahan data nilai mahasiswa

Tipe terstruktur yang menyatakan nilai ujian seorang mahasiswa untuk mata kuliah yang diambil(MK) terdiri dari Nim, Nama, Kode MK dan Nilai

| <i>Nim</i> | <i>NamaMhs</i> | <i>KodeMK</i> | <i>Nilai</i> |
|------------|----------------|---------------|--------------|
|------------|----------------|---------------|--------------|

```
type NilaiMhs : record < Nim : integer,  
                        NamaMhs : string,  
                        KodeMK : string,  
                        Nilai : char  
>
```

Jika dideklarasikan sebuah variabel M sebagai berikut :

M : NilaiMhs {artinya : mendeklarasikan variabel M bertipe NilaiMhs}

Maka cara mengacu atau mengakses nilai elemen yang tersimpan pada W yang telah terdefinisi adalah :

M.Nim, M>NamaMhs, M.KodeMK, dan M.Nilai

Contoh :

Kamus

```
type NilaiMhs : record < Nim : integer,  
                        NamaMhs : string,  
                        KodeMK : string,  
                        Nilai : char  
>
```

M : NilaiMhs

Deskripsi

```
M.Nim ← 077006187  
M>NamaMhs ← 'Arjuna'  
M.KodeMK ← 'MKK1107'  
M.Nilai ← 'B'  
read(M.Nim, M>NamaMhs, M.KodeMK, M.Nilai)  
write(M.Nim, M>NamaMhs, M.KodeMK, M.Nilai)
```

LATIHAN

1. Tentukan baris instruksi yang salah pada algoritma di bawah ini!

Algoritma Contoh_Type

Kamus

```
type MyPoint : record <x : integer, y : integer>
P : integer
R : MyPoint
```

Deskripsi

```
{1} read(MyPoint)
{2} read(R)
{3} read(P)
{4} write(H.x , H.y)
{5} R ← R + 5
{6} R.x ← R.x + 5
{7} write(MyPoint)
{8} write(R.x, R.y)
```

2. Deklarasikan tipe Data_Karyawan yang terdiri dari ID, Nama Karyawan, Golongan, dan Gaji Pokok. Berikutnya buat algoritma untuk menerima masukan 1 buah data Karyawan dan menampilkannya!
3. Buat algoritma yang membaca dua waktu (jam, menit, dan detik) dan menghitung selisih kedua waktu tersebut dalam detik kemudian menampilkannya!

JAWABAN LATIHAN

1. Baris instruksi yang salah :

- (1) Karena MyPoint adalah sebuah type maka tidak bisa dijadikan sebagai variabel penampung hasil pembacaan dari piranti masukan
- (2) Karena R adalah variabel yang bertipe record jadi harus disebutkan nama fieldnya. Seharusnya read(R.x) atau read(R.y)
- (4) Karena variabel H tidak ada pada kamus
- (5) Karena R adalah variabel yang bertipe record jadi harus disebutkan nama fieldnya pada setiap pemrosesan. Seharusnya R.x ← R.x + 5 atau R.y ← R.y + 5
- (7) Karena MyPoint adalah sebuah type maka tidak bisa langsung dioutputkan

- 2.

Algoritma Rerata_Nilai_Mahasiswa

Kamus

```
type Data : record < ID : integer,
                    nama : string,
                    golongan : char,
                    gaji_pokok : real
                    >
```

D : Data

Deskripsi

```
read(D.ID, D.nama, D.golongan, D.gaji_pokok)
write(D.ID, D.nama, D.golongan, D.gaji_pokok)
```

3.

Algoritma Selisih_Waktu

Kamus

```
type waktu : record < jam : integer [0..23],  
                      menit : integer [0..59],  
                      detik : integer [0..59]  
                    >
```

```
W1, W2 : waktu  
selisih : integer
```

Deskripsi

```
read(w1.jam, w1.menit, w1.detik)  
read(w2.jam, w2.menit, w2.detik)  
selisih ← (w2.jam * 3600 + w2.menit * 60 + w2.detik) -  
          (w1.jam * 3600 + w1.menit * 60 + w1.detik)  
write(selisih)
```