

Chapter 13 & 14 : Record

Pengenalan Algoritma dan Pemrograman

Oleh : Fiftin Noviyanto

Rekaman

Disusun oleh satu atau lebih field. Tiap field menyimpan data dari tipe dasar tertentu atau dari tipe bentukan lain yang sudah didefinisikan sebelumnya. Nama rekaman ditentukan oleh pemrogram.

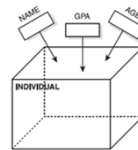
Rekaman disebut juga tipe terstruktur.

Beyond the Array: The Record

- Record bermanfaat untuk mengelompokkan sejumlah data dengan tipe yang berlainan.

- Contoh :

```
struct individual
{
    string name;
    int age;
    double g_p_a;
}; // the semicolon ends the declaration
```



- Struktur di atas bernama data_tanggal, yang tersusun atas 3 unit penyusunnya yaitu tahun, bulan, dan tanggal.

Contoh lain :

- `type Titik : record <x : real, y : real>`
jika P dideklarasikan sebagai Titik maka mengacu field pada P adalah P.x dan P.y.
- Didefinisikan tipe terstruktur yang mewakili Jam yang dinyatakan sebagai jam (hh), menit (mm) dan detik (ss), maka cara menulis type Jam adalah :
`type JAM : record`
 <hh : integer, {0...23}
 mm : integer, {0...59}
 ss : integer {0...59}
 >

Jika J adalah peubah (variabel) bertipe Jam maka cara mengacu tiap field adalah J.hh, J.mm dan J.ss

Algoritma 1 : jarak 2 titik

- Pseudocode
 - Input : koordinat titik $A(x,y)$ dan $B(x,y)$
 - Hitung jarak dengan rumus
 - Cetak jarak

Algoritma jarak 2 titik

Deklarasi

A, B : titik

jarak : real

Deskripsi

baca(A.x, A.y, B.x, B.y)

jarak $\leftarrow \text{sqrt}(\text{sqr}(B.x-A.x) + \text{sqr}(B.y-A.y))$

tulis(jarak)

Algoritma 2 : selisih 2 waktu

- Pseudocode :
 - Input : waktu akhir dan waktu awal
 - Proses :
 - waktu akhir dan awal dikonversi menjadi detik
 - Dihitung selisihnya
 - Hasilnya dikonversi menjadi waktu
 - Output : selisih waktu

Algoritma selisih 2 waktu

Deklarasi

w1, w2 : waktu

detik1, detik2, selisih_detik : integer

selisih : waktu

Deskripsi

baca(w1.jam, w1.menit, w1.detik)

baca(w2.jam, w2.menit, w2.detik)

detik1 $\leftarrow \text{konversi_ke_detik}(w1.jam, w1.menit, w1.detik)$

detik2 $\leftarrow \text{konversi_ke_detik}(w2.jam, w2.menit, w2.detik)$

selisih_detik $\leftarrow \text{detik2} - \text{detik1}$

selisih $\leftarrow \text{konversi_ke_waktu}(\text{selisih_detik})$

cetak(selisih)

Rekaman dalam rekaman

- Bisa saja terjadi, field dari suatu rekaman berupa rekaman juga.
- Misalkan, rekaman mahasiswa :
- Type mahasiswa : record <
 - Nama : string
 - Alamat : string
 - Tgl_lahir : tanggal >
- Tipe tanggal ternyata berupa rekaman :
- Type tanggal : record <
 - Tgl : integer
 - Bulan : integer
 - Tahun : integer >

Algoritma 3 : menghitung usia

Pseudocode

- Input : tanggal sekarang, dan tgl lahir mahasiswa
- Proses :
 - Masing-masing tanggal dikonversi ke bilangan
 - Dihitung selisihnya
- Output : selisih usia

Latihan

- Buatlah algoritma untuk mengkonversi tanggal menjadi bilangan integer
- Petunjuk :
 - Pertimbangkan lama hari dalam bulan
 - Contoh : Januari 31, Juni 30
 - Bisa menggunakan array untuk menyimpan data
 - Pertimbangkan pula tahun kabisat/non kabisat

Rekaman mewakili tabel

- Kadang kita mempunyai tabel, sebagai contoh tabel presensi mahasiswa.

No.	NIM	Nama	Hadir1	Hadir2	...	Hadir10
1	08018011	Adityo Wicaksono				
2	08018012	Vendi Rinanto				
3	08018013	Tana Raharjo				

- Tabel di atas dapat direpresentasikan menjadi :
- Type presensi : record <
 - No : integer
 - NIM : string
 - Nama : string
 - Tgl_kuliah : array [1..10] of tanggal >
- Namun, larik data akan digunakan untuk setiap entry tersebut, sehingga akan terdapat gabungan 2 tipe terstruktur yaitu :
- Larik_presensi : array [1..100] of presensi
- Bila terdapat 100 mahasiswayang mengikuti kuliah

Algoritma 4 : menghitung kehadiran dan presentase kehadiran mahasiswa

- Ide :
 - Kehadiran mahasiswa dapat diisi dengan ON (jika hadir) dan OFF (bila tidak hadir)
 - Untuk menghitung jumlah kehadiran tinggal menggunakan counter untuk isian ON
 - Untuk menghitung presentase = $(\text{jumlah seluruh peserta} - \text{jumlah kehadiran}) / (\text{jumlah seluruh peserta}) \times 100\%$

Latihan

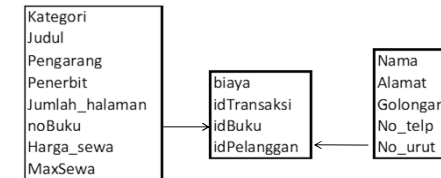
- Buatlah studi kasus berdasar tugas proyek tiap kelompok tabel-tabel yang ada menjadi tipe data baru

Studi kasus : persewaan buku

- Entitas : objek riil
 - Buku dengan atribut :
 - Kategori : string[25]
 - Judul : string[50]
 - Pengarang : string [50]
 - Penerbit : string [50]
 - Jumlah_halaman : integer
 - noBuku : integer
 - Harga_sewa : integer
 - MaxSewa : integer
 - Pelanggan dengan atribut :
 - Nama : string [25]
 - Alamat : string [30]
 - Golongan : string [20]
 - No_telp : string[15]
 - No_urut : integer

- Relasi : transaksi sewa
 - biaya : integer
 - idTransaksi : integer
 - idBuku : integer
 - idPelanggan : integer

Relasi antar tabel



Implementasi Entitas Buku

```

class Buku {
public:
    void setKategori( const char *
        const );
    void setJudul( const char * const );
    void setPengarang( const char *
        const );
    void setPenerbit( const char *
        const );
    void setJumlah_halaman( int );
    void setnoBuku( int );
    void setHarga_sewa( int );
    void setMaxSewa( int );

private:
    char Kategori[25];
    char Judul[50];
    char Pengarang[50];
    char Penerbit[50];
    int Jumlah_halaman;
    int noBuku;
    int Harga_sewa;
    int MaxSewa;
};
  
```

Implementasi Entitas Pelanggan

```

class Pelanggan {
public:
    void setName( const char * const );
    void setAlamat( const char * const );
    void setGolongan( const char * const );
    void setNo_telp( long int );
    void setNo_urut( int );

private:
    char Nama[25];
    char Alamat[30];
    char Golongan[20];
    char No_telp[15];
    int No_urut;
};
  
```

Implementasi Relasi Sewa

```
class Sewa {  
public:  
    void setidTransaksi(int);  
    void setbiaya( double );  
    void setidBuku( int );  
    void setidPelanggan( int );  
    int getidTransaksi() const;  
    double getbiaya() const;  
    int getidBuku() const;  
    int getidPelanggan() const;  
private:  
    int biaya;  
    int idTransaksi;  
    int idBuku;  
    int idPelanggan;  
};
```