



PENGANTAR PERKULIAHAN

ALGORITMA DAN STRUKTUR DATA



Institut Teknologi Sumatera

TUJUAN KULIAH

- Mahasiswa dapat memahami masalah secara menyeluruh dan persiapan data pada program.
- Mahasiswa dapat memahami operasi-operasi yang dilakukan terhadap data.
- Mahasiswa dapat memahami penyimpanan data-data pada memori sehingga tersimpan dan terstruktur secara logis serta beroperasi secara efisien.
- Mahasiswa dapat melakukan perancangan dan pemilihan struktur data yang tepat.

MOTIVASI

- SD merupakan materi dasar dalam berbagai jurusan di bidang teknologi informasi.
- Pemilihan struktur data yang tepat, dapat membuat program lebih efisien.
- Program efisien = performa = untung

PENILAIAN

- **Presensi**

- Minimum 80% kehadiran untuk dapat mengikuti ujian.

- **Komponen**

1. Ujian Tengah Semester (UTS)
2. Ujian Akhir Semester (UAS)
3. Kuis/Tugas Mandiri
4. Tugas Besar
5. Hasil praktikum

KEJUJURAN AKADEMIS

- **Kecurangan** (Setiap bentuk kecurangan akan mendapatkan sanksi tegas)
 - Kecurangan dalam tugas (menyalin hasil pekerjaan yang lain atau copy-paste dari internet)
 - Kecurangan dalam pencatatan kehadiran (titip tanda tangan)
 - Kecurangan saat ujian (menyontek jawaban teman atau berkerja sama)
- **Sanksi nilai E akan dikenakan baik pada si pelaku maupun yang memberikan kesempatan.**

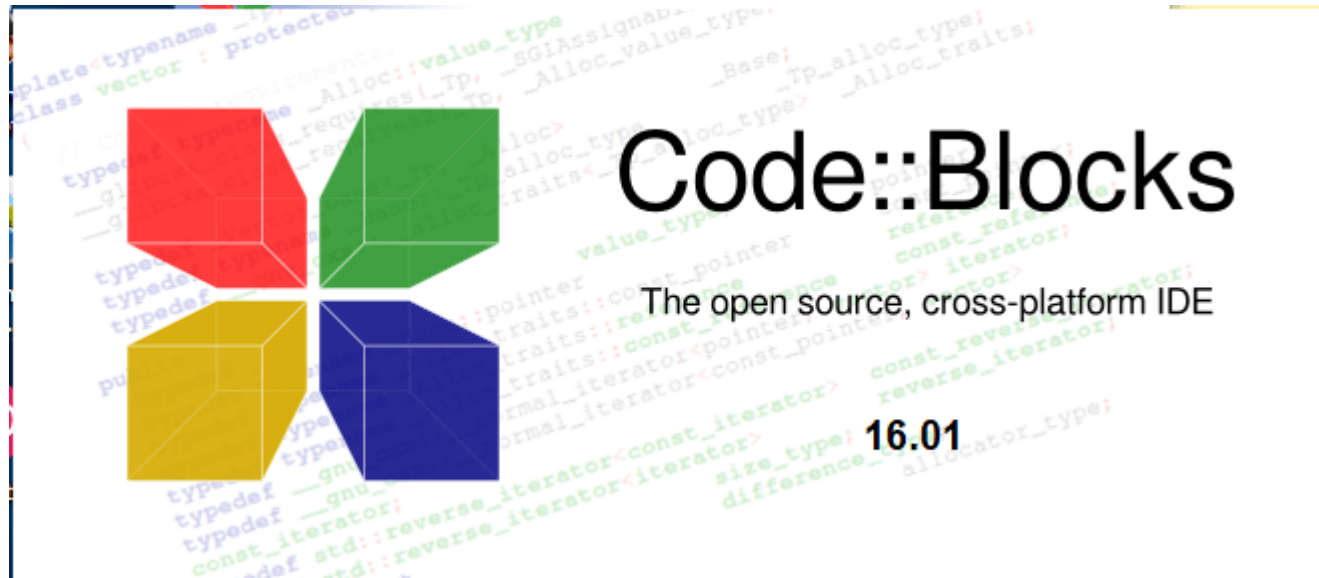
GRADING

- A : 80 ke atas
- AB : 70 sampai dengan 79.9
- B : 65 sampai dengan 69.9
- BC : 60 sampai dengan 64.9
- C : 50 sampai dengan 59.9
- D : 35 sampai dengan 49.9
- E : di bawah 35

PUSTAKA

- Abdul Kadir, “Teori dan Aplikasi Struktur Data Menggunakan C++
- Budi Raharjo, “Pemrogramman C++”
- Buku-buku/website/artikel relevan lain terkait struktur data dan bahasa C++

TOOLS



POKOK BAHASAN

- Pengantar
- Pengenalan array (NEFF)
- Pengenalan record

Program = Algoritma + Struktur Data

STRUKTUR DASAR PROGRAM C++

```
#header

int main() {
    // kamus data
    ...

    // algoritma
    ...
}
```

KAMUS DATA

- Kamus dipakai untuk mendeklarasi nama-nama yang digunakan dalam program
- Deklarasi nama yang didefinisikan pemrogram
 - type
 - variabel
 - konstanta
- Deklarasi BUKAN instruksi :

`[tipe_data] [nama];`
- Contoh :

`int l;
float JumlahUang
Point Titik;`

TIPE DATA

- Tipe data **primitif** atau tipe **dasar** (dalam C++)
 - Boolean (bool)
 - Integer (int)
 - Real (float)
 - Character (char)
 - String (string)
- Tipe data **turunan** atau **bentukan**
 - Dibentuk dari gabungan tipe dasar
 - Contoh
 - Tipe Data Mahasiswa
 - Dibentuk dari
 - » NIM: string
 - » Nama: string
 - » Umur: integer
 - » Kota: string
 - Tipe Array
 - Dibentuk dari kumpulan integer, misalnya 10 data tentang umur

LATIHAN 1

Buatlah program yang menerima masukan 20 buah bilangan integer (dari keyboard) dan menuliskan ke layar rerata dari ke-20 integer tersebut.

- Input : 20 bilangan
- Output : rerata

LATIHAN 2

Buatlah program yang menerima masukan 20 buah bilangan integer (dari keyboard) dan menuliskan ke layar 3 bilangan terbesar dari semua nilai integer tersebut.

- Input : 20 bilangan
- Output : 3 bilangan terbesar

ARRAY

OVERVIEW VARIABEL

- Satu buah variabel hanya bisa menampung satu buah nilai.
- Untuk menampung banyak nilai, maka harus menyediakan banyak variabel.



OVERVIEW VARIABEL

- Tuliskan program untuk menerima input 3 buah nama, kemudian tampilkan kombinasi semua pasangan nama?

Ali
Budi
Caca
Ali - Budi
Ali - Caca
Budi - Caca

```
int main () {  
    // KAMUS  
    string nama1, nama2, nama3;  
  
    //ALGORITMA  
    cin >> nama1;  
    cin >> nama2;  
    cin >> nama3;  
    cout << nama1 " - " nama2 << endl;  
    cout << nama1 " - " nama3 << endl;  
    cout << nama2 " - " nama3 << endl;  
}
```

OVERVIEW VARIABEL

- Tuliskan program untuk menerima input 10 buah nama, kemudian tampilkan kombinasi semua pasangan nama?

```
Nama-1: Ali
Nama-2: Budi
...
Nama-9: Ina
Nama-10: Jaja
Ali – Budi
Ali – Caca
...
Ina - Jaja
```

```
int main () {
    // KAMUS
    string nama1, nama2, nama3, nama 4, nama5;
    string nama6, nama7, nama8, nama9, nama10;
    //ALGORITMA
    cin >> nama1;
    cin >> nama2;
    ... // lanjutkan sendiri!!
    cin >> nama10;
    cout << nama1 << " - " << nama2 << endl;
    cout << nama1 << " - " << nama3 << endl;
    ... // lanjutkan sendiri!!
    cout << nama9 << " - " << nama10 << endl;
}
```

UNTUNGNYA....

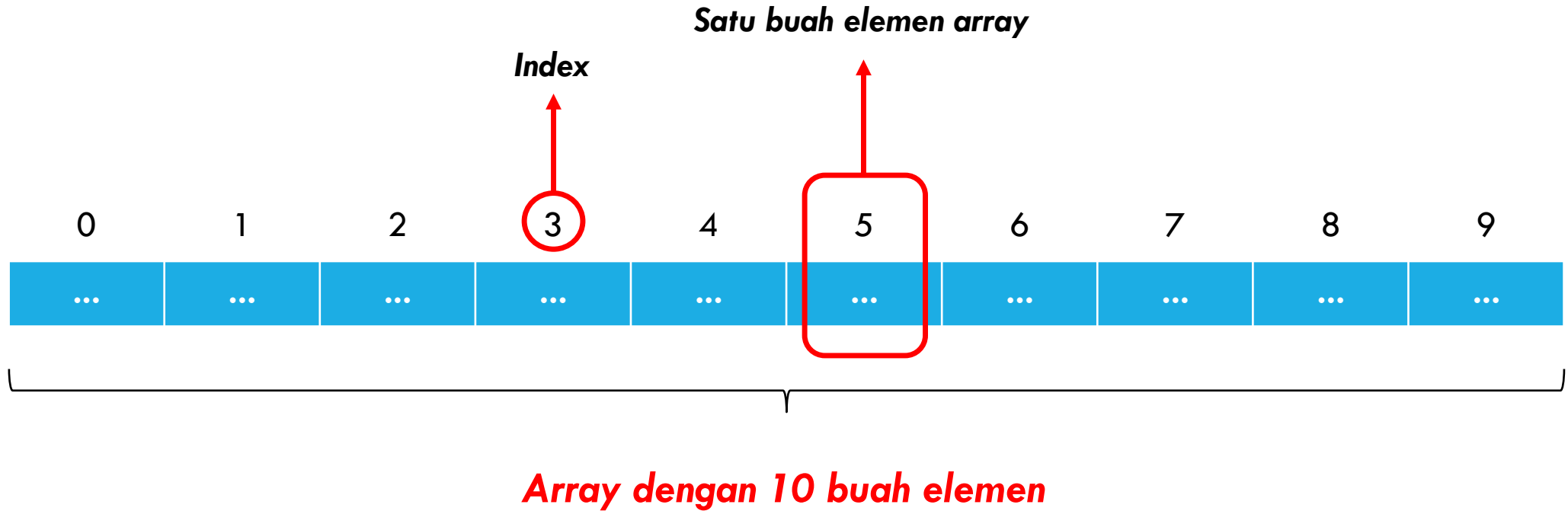
C++ mendukung fasilitas penyimpanan banyak data yang memiliki tipe yang sama ke dalam satu nama tertentu.



ARRAY

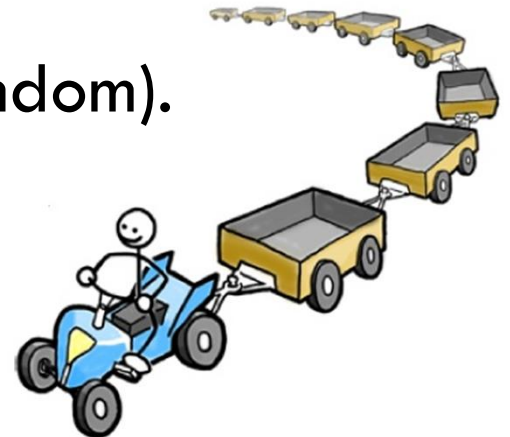
- Array adalah kumpulan nilai yang memiliki tipe yang serupa yang disimpan dalam satu tempat dengan nama yang sama.
- Masing-masing data yang disimpan dalam array memiliki index sebagai sebuah pengenalan.
- Array dengan satu buah indeks disebut array berdimensi satu, vektor, larik, atau tabel.

ARRAY



KARAKTERISTIK ARRAY

- Bersifat statis, jumlah maksimal elemen array harus sudah ditentukan sejak awal.
- Setiap elemen array dapat diakses dengan menyebutkan nama array disertai index-nya.
- Index pada array bertipe integer.
- Array dapat diakses secara berurutan maupun acak (random).



DEKLARASI ARRAY

- Array dapat dideklarasikan dengan cara menyebutkan type variabel dan jumlah elemen yang terkandung didalam array tersebut.
- Cara deklarasi: `<type> <namatabel>[<ukuran>];`
- Contoh:
 - `int tinggi[10];` // indeks 0..9
 - `float sesuatu[5];` // indeks 0..4
 - `char huruf[100];` // indeks 0..99
 - `string kata[100];` // indeks 0..99

PENGISIAN NILAI ARRAY

- Cara langsung

```
tinggi[0] = 165;  
tinggi[1] = 155;  
tinggi[2] = 163;  
tinggi[3] = 170;  
tinggi[4] = 155;
```

```
int tinggi[5] = {165, 155, 163, 170, 155};
```

- Dengan perulangan

```
for(i=0; i<10; i++) {  
    nilai[i] = i * 10;  
}
```

AKSES NILAI ARRAY

- Setiap elemen array dapat diakses dengan menyebutkan nama array disertai index-nya.
- Dengan cara langsung

```
cout << nilai[5];
```

- Dengan perulangan

```
for(i=0; i<10; i++){  
    cout << nilai[i];  
}
```

CONTOH

- Buatlah sebuah array yang dapat menampung data nilai Kimia dari 7 siswa SMA. Nilai diantara rentang 0-100.
- Tampilkan data pada index elemen ke 0, 3, dan 4.

LATIHAN 3

- Buatlah sebuah array yang dapat menampung N buah data:
 - Nilai Kimia
 - Nilai Fisika
 - Nilai Matematika
 - Nama siswa
- Nilai diantara rentang 0-100. Nilai N ditentukan pengguna
- Mintalah pengguna untuk menginputkan seluruh data.

LATIHAN 4

- Buatlah sebuah array yang dapat menampung N buah data:
 - Nilai Kimia
 - Nilai Fisika
 - Nilai Matematika
 - Nama siswa
- Nilai diantara rentang 0-100. Nilai N ditentukan pengguna
- Mintalah pengguna untuk menginputkan seluruh data.
- Tampilkan data:
 1. Siswa dengan nilai rerata tertinggi.
 2. Siswa yang harus remidi, ditandai dengan nilai rerata dibawah 60.

LATIHAN 6

- Buatlah sebuah array yang dapat menampung N buah data:
 - Nilai Kimia
 - Nilai Fisika
 - Nilai Matematika
 - Nama Siswa
- Nilai diantara rentang 0-100. Nilai N ditentukan pengguna
- Mintalah pengguna untuk menginputkan seluruh data.
- Buatlah agar program dapat melakukan pencarian berdasarkan nama siswa.

ARRAY YANG TERISI SEBAGIAN (1/2)

- Dalam banyak persoalan, kita mendefinisikan sejumlah elemen array namun tidak selalu harus menggunakan seluruhnya.
 - Asumsi : array terisi secara kontinyu dan “rata kiri” (mulai dari elemen ke-0)
- Contoh : array berikut hanya terdefinisi sampai elemen ke-6

1	3	4	-1	99	65				
0	1	2	3	4	5	6	7	8	9

ARRAY YANG TERISI SEBAGIAN (2/2)

- Kita membutuhkan sebuah nilai yang banyaknya elemen array yang terdefinisi.
- Nilai ini disebut sebagai nilai efektif array (Neff).
- Contoh :

1	3	4	-1	99	65				
0	1	2	3	4	5	6	7	8	9

Neff = 6, artinya hanya 6 elemen yang terdefinisi, yaitu dari indeks ke-0 s.d ke-5

- Memungkinkan definisi array kosong, yaitu Neff = 0.

MENGISI DAN MEMBACA ISI ARRAY

- Membaca sejumlah Neff elemen pada array dan menampilkan ke layar.
- Neff adalah nilai efektif masukkan dari pengguna



File : isiBacaArray2.cpp

```
#include <iostream>
using namespace std;
int main()
{ //KAMUS
int TabInt[100]; int i, Neff;
//ALGORITMA
cin >> Neff; //nilai
efektif[0..100]
//MENGISI ARRAY
i = 0;
while (i < Neff) {
```

```
cin >> TabInt[i];
i++;
} // i = Neff
//MENULIS ISI ARRAY
i = 0;
while (i < Neff) {
cout << TabInt[i] << endl;
i++;
} // i = Neff
return 0;
}
```

MENCARI INDEKS SUATU NILAI (SEARCHING)

- Dilakukan pencarian *indeks* suatu nilai dan ditemukan pertama kali dalam array.
- Neff adalah banyaknya elemen yang terdefinisi dalam array.

```
found = true;
} else {
    i++;
}
} // i = Neff atau found
if (found) { // X ada
    cout << X << "ada di indeks" << i;
} else {
    cout << X << "tidak ditemukan";
}
return 0;
}
```

File : searchArray2.cpp

```
#include <iostream>
using namespace std;
int main()
{ //iAMUS
    int X, i, Neff; bool found;
    Int TabInt[10];
    //ALGORITMA
    //Pengisian data : asumsi array
    terisi, Neff terdefinisi
    //MENCARI NILAI X
    cin >> X;
    i = 0; found = false;
    while ((i < Neff) && (!found)) {
        if (TabInt[i] == X){
```

1

2

RECORD

RECORD

- Dalam bahasa C/C++ seringkali record disebut dengan istilah ***structure*** atau ***struct*** saja.
- Struktur atau record adalah sebuah variabel bentukan yang terdiri dari sejumlah item dari tipe data yang berbeda-beda.

RECORD

- Masing-masing item data tersebut disebut sebagai *field* atau elemen struktur.
- Setiap field tersebut dapat memiliki tipe data yang berbeda.
- Setiap field dapat diakses dan dimanipulasi secara individu.

BENTUK UMUM

```
// deklarasi
struct sesuatu {
    tipe_data item1;
    tipe_data item2;
    ....
    ....
};
```

```
// penggunaan
```

```
sesuatu x;  Tipe data sesuatu, nama variabel x
```

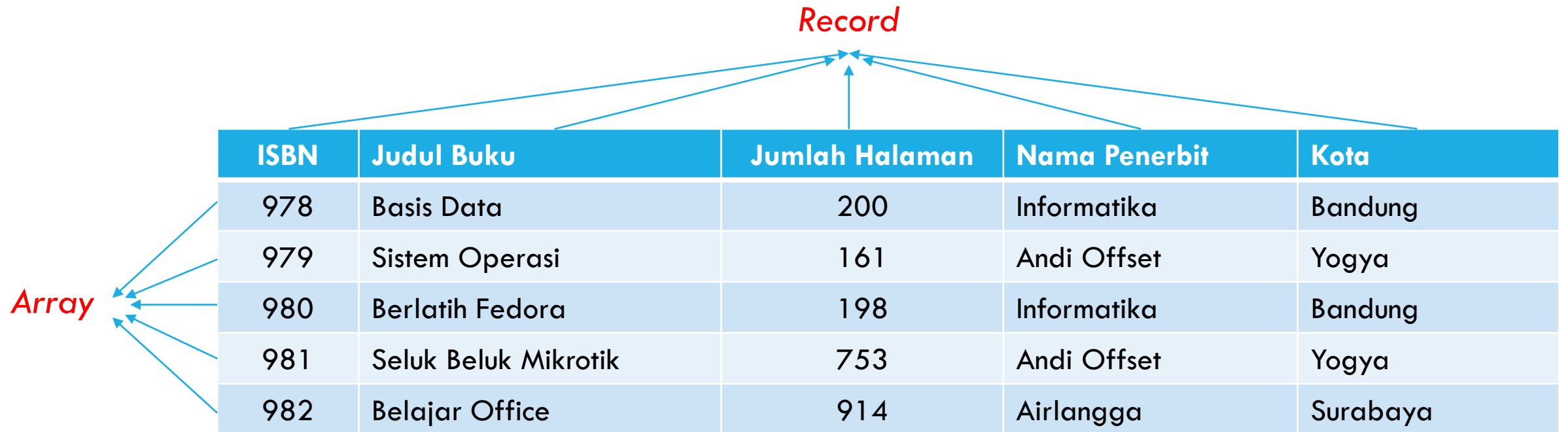
RECORD

- Deklarasi record akan membentuk sebuah tipe data baru.
- Tipe data baru berupa nama record tersebut dapat digunakan dalam semua bagian program sebagaimana tipe data lainnya seperti **int**, **float**, **char**, **short**, dll.

ARRAY DARI RECORD

- Record memiliki kelebihan dapat menyimpan jenis data yang berbeda dalam satu nama.
- Array memiliki kelebihan dapat menyimpan beberapa data yang memiliki jenis data yang sama.
- Array dari record menggabungkan kelebihan array dan record.

VISUALISASI ARRAY DARI RECORD



CONTOH

```
#include <iostream>
using namespace std;

struct mhs {
    int nim;
    string nama;
    float ipk;
};

int main(){
    mhs x;
    x.nim   = 14151003;
    x.nama  = "Andi Sugandi";
    x.ipk   = 3.25;

    cout << "NIM    : " << x.nim << endl;
    cout << "Nama    : " << x.nama << endl;
    cout << "IPK     : " << x.ipk;

}
```

LATIHAN 7

- Menggunakan array dari record, buatlah program yang dapat menyimpan data N buah buku.
- Inputan data berasal dari pengguna.
- Tampilkan data buku dengan format seperti dibawah ini.

ISBN	Judul Buku	Jumlah Halaman	Nama Penerbit	Kota
978	Basis Data	250	Informatika	Bandung
979	Sistem Operasi	400	Andi Offset	Yogya
980	Berlatih Fedora	125	Informatika	Bandung
981	Seluk Beluk Mikrotik	144	Andi Offset	Yogya
982	Belajar Office	320	Airlangga	Surabaya

LATIHAN 8

- Modifikasi Latihan 7, tambahkan fungsi pencarian berdasarkan ISBN.

ISBN	Judul Buku	Jumlah Halaman	Nama Penerbit	Kota
978	Basis Data	250	Informatika	Bandung
979	Sistem Operasi	400	Andi Offset	Yogya
980	Berlatih Fedora	125	Informatika	Bandung
981	Seluk Beluk Mikrotik	144	Andi Offset	Yogya
982	Belajar Office	320	Airlangga	Surabaya

LATIHAN 9

- Modifikasi Latihan 7, tambahkan fungsi pengurutan menurun, berdasarkan jumlah halamannya.

ISBN	Judul Buku	Jumlah Halaman	Nama Penerbit	Kota
978	Basis Data	250	Informatika	Bandung
979	Sistem Operasi	400	Andi Offset	Yogya
980	Berlatih Fedora	125	Informatika	Bandung
981	Seluk Beluk Mikrotik	144	Andi Offset	Yogya
982	Belajar Office	320	Airlangga	Surabaya

TERIMA KASIH

PR 1

Buatlah sebuah array dengan N buah elemen. Kemudian mintalah pengguna untuk menginputkan nilai dari masing-masing dalam bentuk bilangan bulat. Dari data yang dimasukkan tersebut, tentukanlah berapa banyak data yang termasuk kedalam tipe bilangan ganjil dan genap.

PR 2

- Buatlah sebuah program yang mendeklarasikan sebuah array of integer berukuran 100, mengisi array sampai salah satu dari 2 kondisi ini terpenuhi :
 - Array terisi penuh
 - Pengguna mengisikan -999 (tidak termasuk sebagai elemen array)
- Tuliskan ke layar berapa banyak elemen array dan jumlah total seluruh elemen.

PR 3

- Buatlah program yang dapat menyimpan data N buah mahasiswa.
- Jenis data yang disimpan adalah: NIM, nama, IPK, dan alamat.
- Program anda harus memiliki fasilitas:
 - Pencarian mahasiswa berdasarkan NIM
 - Menampilkan informasi rata-rata IPK, dan siapa pemilik IPK tertinggi dan terendah.

PR 4

- Modifikasi PR 3
- Program anda harus memiliki fasilitas:
 - Pengurutan berdasarkan IPK
 - Pengurutan berdasarkan NIM
- Gunakan algoritma pengurutan selain Bubble sort