



Array (Tabel) [Dalam Bahasa C++]

Tim Penyusun Materi KU1072



KU1072/Pengenalan Teknologi Informasi B
Tahap Tahun Pertama Bersama
Institut Teknologi Bandung



Tujuan Perkuliahan

- Mahasiswa memahami makna dan kegunaan array (tabel)
- Mahasiswa dapat menggunakan notasi pendefinisian dan pengacuan array dengan benar hingga proses pencarian terhadap elemen array
- Mahasiswa dapat membuat program dengan menggunakan array



BAGIAN 1

2013/10/18

KU1072/Pengenalan Teknologi
Informasi B

3

Mengolah 3 data

- Tuliskan program yang menerima 3 nama, lalu menampilkan semua kombinasi pasangan nama.
- Contoh keluaran:

```
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;
}
```

Mengolah 10 data

- Tuliskan program yang menerima 10 nama, lalu menampilkan semua kombinasi pasangan nama.
- Contoh keluaran:

```
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;
}
```

Bagaimana kalau...

Anda diminta menampilkan semua kombinasi pasangan nama yang mungkin dari ...

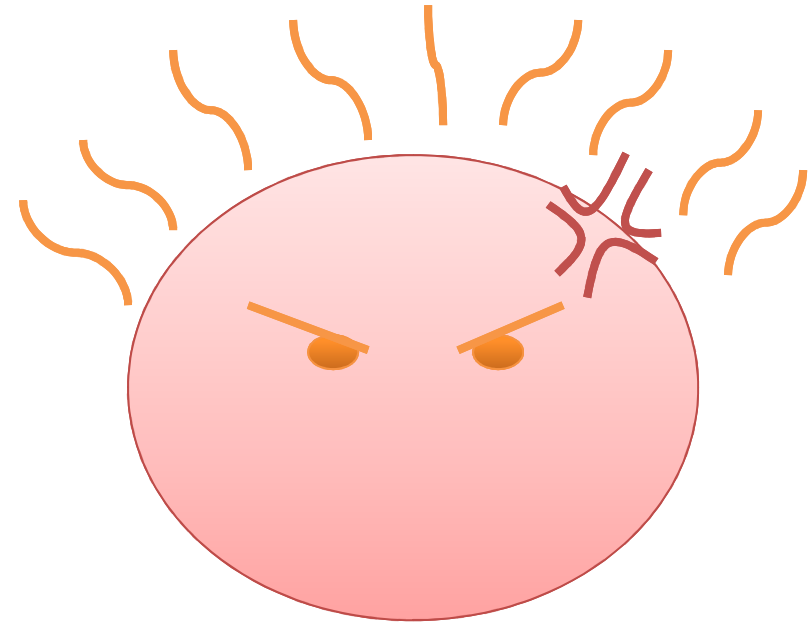
100 nama ???

1000 nama ???

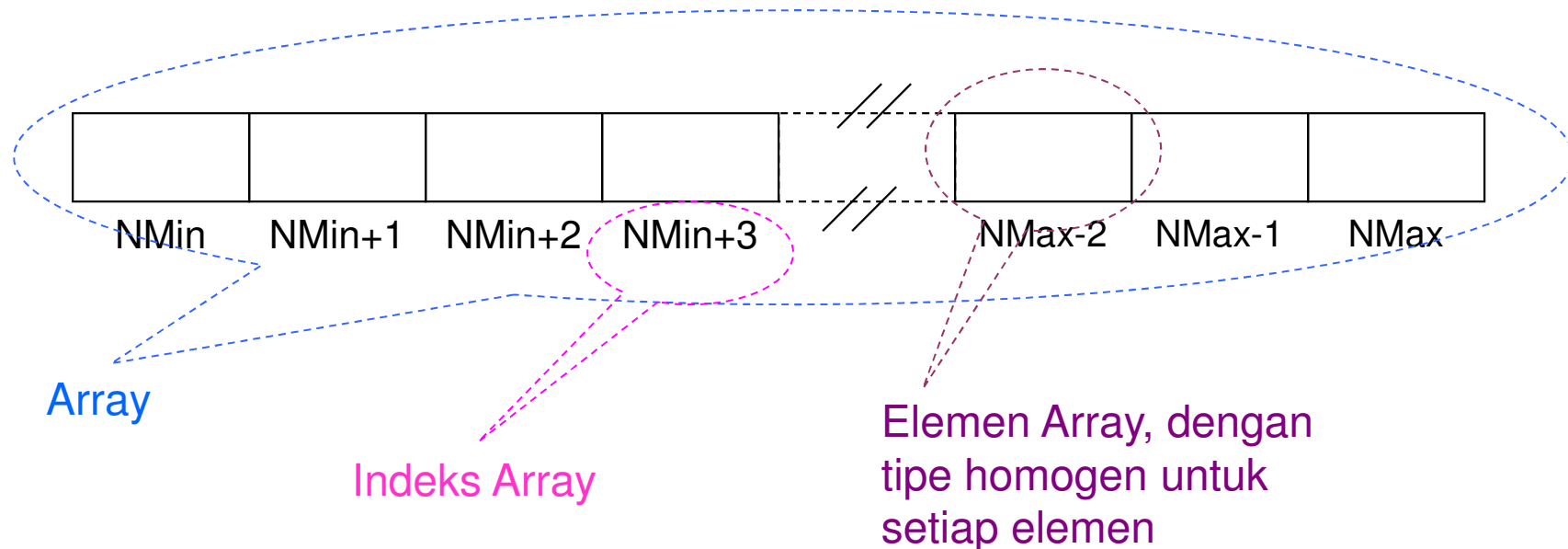
10000 nama ???

1000000 nama ???

....



Array / Tabel / Vektor / Larik



- Type array adalah type yang mengacu kepada sebuah atau sekumpulan elemen melalui indeks
- Merepresentasikan sekumpulan informasi yang bertipe sama dan disimpan dengan urutan yang sesuai dengan definisi indeks

Array dalam C/C++ (1)

- Variabel dapat dideklarasikan ber-type **array** dari suatu **type** tertentu
- Setiap elemen array diakses dengan alamat berupa **indeks** yang bertype integer
- Cara deklarasi: `<type> <namatabel>[<ukuran>];`
- Contoh: `int TabInt[10];`

Array bernama **TabInt** dengan setiap elemen bertype **integer**, dengan ukuran **10** elemen, dengan alamat setiap elemen array (indeks) adalah dari **indeks ke-0 s.d. 9**

Array dalam C/C++ (2)

- Cara akses elemen:
`<namatabel>[<indeks>]`
- Contoh: `int TabInt[10];`

1	2	4	-1	100	2	0	-1	3	9
0	1	2	3	4	5	6	7	8	9

`cout << TabInt[4]; // akan tercetak: 100`

`int x = TabInt[0] + TabInt[5]; // x bernilai 3`

`TabInt[9] = 8; // Elemen array indeks 9 menjadi 8`

`TabInt[10] ??? // Berada di luar range, tidak terdefinisi!!`

Contoh Deklarasi Array yang lain

```
int main() {  
    // KAMUS  
    int TabJumlahHari[12]; // indeks 0..11  
    float TabNilai[15];   // indeks 0..14  
    char TabHuruf[100];  // indeks 0..99  
    string TabKata[100]; // indeks 0..99  
    Point TabTitik[20];  // indeks 0..19, Point terdefinisi  
    // ALGORITMA  
    ...  
}
```

- Elemen dari array dapat diakses langsung jika dan hanya jika indeks terdefinisi
- Cara mengacu sebuah elemen:
 TabInt[2]
 TabInt[i] jika i terdefinisi

Mengisi Array

- Mengisi array merupakan aktifitas memberi nilai elemen array
 - Pemberian nilai satu elemen, contoh: `TabInt[0]=31;`
 - Pemberian nilai beberapa elemen, contoh:

```
for (i=0;i<10;i++) {  
    TabInt[i]=i*10;  
}
```
- Hati-hati!
 - Jangan mengakses elemen yang **indeks-nya berada di luar definisi**, misalnya `TabInt[10]` → index ke-10 tidak terdefinisi untuk TabInt
 - Jangan membaca elemen yang **belum** diisi nilainya

Mengisi dan membaca isi Array

- Elemen array yang telah diberi nilai dapat diakses kembali
- Contoh berikut menampilkan semua isi array ke layar

```
// File: isibacaarray.cpp
// Mengisi array dan menampilkan
// seluruh elemen pada array
#include <iostream>
using namespace std;
int main ()
{ // KAMUS
  int TabInt[10]; int i;

  // Algoritma

  // mengisi array
  for (i=0; i<10; i++) {
    TabInt[i]=i*10;
  }
  // membaca dan menuliskan isi
  // array ke layar
  for (i=0; i<10; i++) {
    cout << TabInt[i] << endl;
  }
  return 0;
}
```

Memroses Array

- Pemrosesan koleksi data pada array dilakukan secara sekuensial
- Asumsi : seluruh elemen array terdefinisi
- Contoh: menjumlahkan data dan menghitung rata-rata

```
// File: sumArray.cpp
// menghitung jumlah seluruh elemen pada array
#include <iostream>
using namespace std;
int main ()
{ // KAMUS
  int sum, i;
  int TabInt[10];
  // ALGORITMA
  // mengisi data nilai dari input user
  cout << "Isilah 10 data nilai dalam range 0-100:"
<< endl;
  for (i=0; i<10; i++) {
    cin >> TabNilai[i];
  }
  // menjumlahkan nilai dan menghitung rata-rata
  cout << "Data input:" << endl;
  sum=0;
  for (i=0; i<10; i++) {
    cout << TabInt[i] << endl;
    sum = sum + TabInt[i];
  }
  cout << "Rata-rata: ";
  cout << (float)sum/10.0 << endl;
  return 0;
}
```



Mencari Indeks Suatu Nilai (searching)

Dengan asumsi semua elemen array terdefinisi, dapat dilakukan pencarian *indeks* suatu nilai ditemukan pertama kali dalam array

```
// File: searchArray.cpp
// mencari indeks di mana nilai ditemukan
#include <iostream>
using namespace std;
int main ()
{ // KAMUS
  int X, i; bool found;
  int TabInt[10];
  // ALGORITMA
  // Pengisian data: asumsi array terisi

  // mencari suatu nilai, yaitu X
  cin >> X;
  i = 0; found = false;
  while ((i < 10) && (!found)) {
    if (TabInt[i]==X) {
      found = true;
    } else {
      i++;
    }
  } // i = 10 atau found
  if (found) { // X ada di
    cout << X << " ada di indeks " << i;
  } else {
    cout << X << " tidak ditemukan";
  }
  return 0;
}
```

Mencari Nilai Ekstrim Array

- Dengan asumsi array tidak kosong, dapat dilakukan pencarian elemen array bernilai ekstrim
- Contohnya: mencari nilai maksimum

```
// File: maxArray.cpp
// mencari nilai maksimum pada array
#include <iostream>
using namespace std;
int main ()
{ // KAMUS
  int max, i;
  int TabInt[10];
  // Algoritma
  // Pengisian data: Buat sebagai
  // latihan
  // mencari nilai maksimum
  max=TabInt[0]; //inisialisasi
                  //max dgn elemen ke-0
  for (i=1; i<10; i++) {
    //ganti max kalau ada nilai elemen
    //array yang lebih besar
    if (TabInt[i]>max) {
      max=TabInt[i];
    }
  }
  cout << "Nilai maksimum: " << max <<
endl;
  return 0;
}
```

Mencari Indeks Nilai Ekstrim Array

- Dengan asumsi array tidak kosong, dapat dilakukan pencarian *indeks* elemen array bernilai ekstrim
- Contohnya: mencari indeks nilai maksimum

```
// File: IdxMaxArray.cpp
// mencari indeks nilai maksimum array
#include <iostream>
using namespace std;
int main ()
{ // KAMUS
  int idxmax, i;
  int TabInt[10];
  // Algoritma
  // Pengisian data: Buat sebagai
  // latihan
  // mencari indeks nilai maksimum
  idxmax = 0;
  for (i=1; i<10; i++) {
    //ganti idxmax jika ada nilai
    //elemen array yang lebih besar
    if (TabInt[i]>TabInt[idxmax]) {
      idxmax = i;
    }
  }
  cout << "Index nilai maksimum: " << max
  << endl;
  return 0;
}
```


Latihan 1

- Deklarasikan sebuah array of character dengan ukuran 10.
- Isilah character dengan nilai dari keyboard (pengguna)
- Tuliskan ke layar semua elemen array
- Hitunglah ada berapa banyak character 'A' yang ada dalam array

Latihan 1



```
// File: arrChar.cpp
// membaca sejumlah data dalam array of character dan memrosesnya
#include <iostream>
using namespace std;
int main ()
{
    // KAMUS
    int i, countA;
    char TabChar[10]; // deklarasi array of character berukuran 10
    // ALGORITMA
    // mengisi array of character dari masukan pengguna
    for (i=0; i<10; i++) {
        cin >> TabChar[i];
    }
    // menuliskan kembali ke layar
    countA = 0;
    for (i=1; i<10; i++) {
        cout << TabChar[i] << endl;
        if (TabChar[i] = 'A') {
            countA++; //sekaligus menghitung kemunculan A
        }
    }
    cout << "Banyaknya A: " << countA << endl;
    return 0;
}
```

Latihan 2

- Lakukanlah modifikasi terhadap program `maxArray.cpp` untuk mencari elemen array bernilai minimum
- Lakukanlah juga modifikasi terhadap program `idxMaxArray.cpp` untuk mencari indeks elemen array bernilai minimum

Latihan 2: minArray



```
// File: minArray.cpp
// mencari nilai minimum pada array
#include <iostream>
using namespace std;
int main ()
{
    // KAMUS
    int i, min;
    int TabInt[10];

    // ALGORITMA
    // data nilai telah terisi
    // mencari nilai minimum
    min=TabInt[0]; //inisialisasi min dgn elemen ke-0
    for (i=1; i<10; i++) {
        if (TabInt[i]<min) {
            min=TabInt[i];
        }
    }
    cout << "Nilai minimum: " << min << endl;
    return 0;
}
```

Latihan 2: IdxMinArray



```
// File: IdxMinArray.cpp
// mencari indeks nilai minimum pada array
#include <iostream>
using namespace std;
int main ()
{
    // KAMUS
    int i, idxmin;
    int TabInt[10];

    // ALGORITMA
    // data nilai telah terisi
    // mencari nilai minimum
    idxmin=0; //inisialisasi idxmin dgn 0
    for (i=1; i<10; i++) {
        if (TabInt[i]<TabInt[idxmin]) {
            idxmin=i;
        }
    }
    cout << "Indeks nilai minimum: " << idxmin << endl;
    return 0;
}
```

Latihan 3

1. Buatlah sebuah program yang menghitung frekuensi kemunculan sebuah nilai X di dalam suatu array. Nilai X didapatkan dari input pengguna.
2. Buatlah sebuah program KemunculanTerakhir yang mencari indeks array terakhir IX yang berisi suatu nilai X. IX akan bernilai -1 jika X tidak terdapat di dalam TabInt.
Contoh: TabInt={4,3,7,6,3,8,4,3,6}, X=3, maka IX=7

Latihan 3: frekXArray



```
// File: frekXArray.cpp
// menghitung frekuensi kemunculan elemen X pada array
#include <iostream>
using namespace std;
int main ()
{
    // KAMUS
    int frek, i, X;
    int TabInt[10];
    // Algoritma
    // mengisi data nilai dari input user
    for (i=0; i<10; i++) {
        cin >> TabInt[i];
    }
    cin >> X; // Nilai yang dicari
    // menghitung frekuensi X
    cout << "Data input: ";
    frek = 0;
    for (i=0; i<10; i++) {
        if (TabInt[i]==X) {
            frek = frek + 1;
        }
    }
    cout << "Frekuensi " << X << " = " << frek << endl;
    return 0;
}
```

Latihan 3: Kemunculan Terakhir



```
// File: lastOccX.cpp
// mencari indeks kemunculan terakhir elemen X pada array; -1 jika tidak ada
#include <iostream>
using namespace std;
int main ()
{
    // Kamus
    int IX, X, i; //X adalah nilai yang akan dicari, IX adalah indeksnya
    int TabInt[10];
    // Algoritma
    // mengisi data nilai dari input user
    for (int i=0; i<10; i++) {
        cin >> TabInt[i];
    }
    cin >> X; // mencari suatu nilai, yaitu X
    i = 10; found = false; // Mencari mundur dari indeks terakhir
    while ((i >= 0) && (!found)) {
        if (TabInt[i]==X) {
            found = true;
        } else {
            i--;
        }
    }
    // i = -1 atau found
    IX = i;
    cout << "Indeks terakhir " << X << " : " << IX << endl;
    return 0;
}
```




BAGIAN 2

2013/10/18

KU1072/Pengenalan Teknologi
Informasi B

25

Array yang terisi sebagian (1)

- Dalam banyak persoalan, kita mendefinisikan sejumlah elemen array, namun tidak selalu harus menggunakan seluruhnya
 - Asumsi: array terisi secara kontigu dan “rata kiri” (mulai dari elemen ke-0)
- Contoh: Array di bawah ini hanya terdefinisi sampai elemen ke-4, elemen sisanya tidak terdefinisi

1	2	4	-1	100					
0	1	2	3	4	5	6	7	8	9

Array yang terisi sebagian (2)

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

1	2	4	-1	100					
0	1	2	3	4	5	6	7	8	9

Neff = 5

Artinya, hanya 5 elemen yang terdefinisi, yaitu dari indeks ke-0 s.d. ke-4

- Memungkinkan definisi array kosong, yaitu Neff = 0

Mengisi dan membaca isi Array

- Membaca sejumlah Neff elemen pada array dan menampilkannya ke layar
- Neff adalah nilai efektif masukan dari pengguna
- Apa yang terjadi jika Neff = 0??

```
// File: isibacaarray2.cpp
// Mengisi array dan menampilkan
// seluruh elemen pada array
#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 ditemukan pertama kali dalam array
- Neff adalah banyaknya elemen yang terdefinisi dalam array
- Apa yang terjadi jika Neff = 0??

2013/10/18

KU1

```
// File: searchArray2.cpp
// mencari indeks di mana nilai ditemukan
#include <iostream>
using namespace std;
int main ()
{ // KAMUS
  int X, i, Neff; bool found;
  int TabInt[10];
  // ALGORITMA
  // Pengisian data: asumsi array terisi,
  // Neff terdefinisi

  // mencari suatu nilai, yaitu X
  cin >> X;
  i = 0; found = false;
  while ((i < Neff) && (!found)) {
    if (TabInt[i]==X) {
      found = true;
    } else {
      i++;
    }
  } // i = Neff atau found
  if (found) { // X ada di
    cout << X << " ada di indeks " << i;
  } else {
    cout << X << " tidak ditemukan";
  }
  return 0;
}
```

Mencari Nilai Ekstrim Array

- Dengan asumsi array tidak kosong, dapat dilakukan pencarian elemen array bernilai ekstrim
- Contohnya: mencari nilai maksimum

```
// File: maxArray2.cpp
// mencari nilai maksimum pada array
#include <iostream>
using namespace std;
int main ()
{ // KAMUS
  int max, i;
  int TabInt[100];
  // Algoritma
  // Pengisian data: Buat sebagai
  // latihan
  // mencari nilai maksimum
  max=TabInt[0]; //inisialisasi
                  //max dgn elemen ke-0
  for(i=1; i<Neff; i++) {
    //ganti max kalau ada nilai elemen
    //array yang lebih besar
    if (TabInt[i]>max) {
      max=TabInt[i];
    }
  }
  cout << "Nilai maksimum: " << max <<
endl;
  return 0;
}
```

Latihan 4

- 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

Latihan 4



```
// File: BacaArray.cpp
// membaca masukan array of integer dari pengguna dan menghasilkan total nilai
#include <iostream>
using namespace std;
int main ()
{
    // KAMUS
    int i, sum, Neff;
    int TI[100];
    // ALGORITMA
    // mengisi data nilai dari input user
    cin >> X; i = 0;
    while ((X != -999) && (i < 100)) {
        TI[i] = X;
        cin >> X;
        i++;
    } // X = -999 atau i = 100
    Neff = i;
    // menghitung total isi array
    i = 0; sum = 0;
    while (i < Neff) {
        sum = sum + TI[i];
        i++;
    } // i = Neff
    cout << "Banyaknya elemen = " << Neff << endl;
    cout << "Jumlah total elemen = " << sum << endl;
    return 0;
}
```