



PENGANTAR KOMPUTER & SOFTWARE II

Array (Part II)

Tujuan

- Mahasiswa mampu memahami proses-proses yang dapat dilakukan terhadap array.
- Mahasiswa dapat menggunakan perintah operasi array yang sesuai dengan benar.
- Mahasiswa dapat memahami persoalan yang dapat diselesaikan dengan operasi array.

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;
}
```

Bagian 2

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

$N_{eff} = 5$

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

- Memungkinkan definisi array kosong, yaitu $N_{eff} = 0$

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)

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

```
// 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;
}
```

Tugas 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

Tugas 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

Tugas 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

Tugas 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 mengisi -999 (tidak termasuk sebagai elemen array)
- Tuliskan ke layar berapa banyak elemen array dan jumlah total seluruh elemen

Terimakasih