

MODULMATA KULIAH

ANALISIS DAN DESAIN ALGORITMA

PG167 – 3 SKS



**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

JAKARTA
SEPTEMBER

201
9
TIM PENYUSUN

Dr. Achmad Solichin, S.Kom.,
M.T.I

Atik Ariesta, S.Kom., M.Kom

Ita Novita, S.Kom., M.T.I



MODUL PERKULIAHAN #12

PENELUSURAN ARRAY SATU DIMENSI

Capaian Pembelajaran	:	Mahasiswa dapat memahami konsep penelusuran array untuk mencari nilai terbesar, nilai terkecil, dan kriteria lainnya di dalam array satu dimensi.
Sub Pokok Bahasan	:	<ol style="list-style-type: none">1.1. Konsep perbandingan sebagai dasar penentuan nilai terbesar dan terkecil1.2. Teknik pencarian nilai terbesar atau terkecil pada array satu dimensi.1.3. Teknik pencarian nilai terbesar atau terkecil menggunakan algoritma sekuensial1.4. Teknik pencarian nilai terbesar atau terkecil menggunakan algoritma sentinel
Daftar Pustaka	:	<ol style="list-style-type: none">1. Gaddis, nd.2011. Starting Out with C++ from Control Structures through Objects .8th. Boston: Addison-Wesley.2. Institue of Distance & Open Learning, n.d. UNIT I Algorithms, Flowcharts & Program Design in: INTRODUCTION TO C++. p. 2053. Sjukani,Moh .2014. Algoritma (Algoritma & Struktur Data 1) Dengan C, C++, dan Java Edisi 9", Mitra Wacana Media.

PENELUSURAN ARRAY SATU DIMENSI

1.1. KONSEP PERBANDINGAN

Perbandingan merupakan salah satu konsep algoritmik yang hampir selalu digunakan. Algoritme seperti penelusuran, pencarian dan pengurutan hampir selalu melibatkan proses perbandingan. Pada dasarnya perbandingan merupakan proses yang melibatkan operator perbandingan. Seperti sudah dijelaskan di modul pertemuan ke-3, operator perbandingan terdiri dari 6 (enam) jenis yang tersaji pada Tabel 1.

Tabel 1. Operator Perbandingan

Operator	Arti	Contoh	Keterangan
<	Kurang dari	$x < y$	Apakah x kurang dari y
<=	Kurang dari sama dengan	$x \leq y$	Apakah x kurang dari sama dengan y
>	Lebih dari	$x > y$	Apakah x lebih dari y
>=	Lebih dari sama dengan	$x \geq y$	Apakah x lebih dari sama dengan y
==	Sama dengan	$x == y$	Apakah x sama dengan y
!=	Tidak sama dengan	$x != y$	Apakah x tidak sama dengan y

Program 1 berikut ini merupakan contoh penerapan operator perbandingan untuk menampilkan bilangan terbesar dari dua buah bilangan bulat yang diinput.

Program 1. Contoh Penggunaan Operator Perbandingan

```
#include<stdio.h>
void main()
{
    int A,B;
    scanf("%i",&A);
    scanf("%i",&B);
    if(A>B)
        printf("%i",A);
    else
        printf("%i",B);
}
```



1.2. TEKNIK PENCARIAN NILAI TERBESAR / TERKECIL PADA ARRAY 1 DIMENSI

Pencarian nilai terbesar dan/atau terkecil dari sejumlah nilai yang tersimpan pada array 1 dimensi merupakan contoh kasus yang paling umum dalam menjelaskan konsep pencarian. Untuk melakukan pencarian nilai pada sebuah array, tentunya harus sudah menguasai 2 (dua) algoritme dasar, yaitu penelusuran array dan perbandingan nilai. Pada dasarnya, penelusuran array dapat dilakukan dengan perulangan baik menggunakan struktur `for()`, `while()`, maupun `do..while()`. Pada beberapa pemrograman, juga mengenal perulangan khusus untuk penelusuran array, yaitu `foreach()`, seperti ditemukan pada bahasa pemrograman PHP.

CONTOH SOAL 12.1 – NILAI TERBESAR

Sudah ada array satu dimensi yang dibuat dengan `int A[11]`. Sudah ada isinya dengan ilustrasi sebagai berikut :

0	1	2	3	4	5	6	7	8	9	10
12	17	10	5	15	25	11	7	25	16	19

Susun program untuk mencari dan mencetak isi array yang nilainya terbesar. Untuk contoh data diatas, bila program dijalankan maka akan tercetak : **25**.

Pembahasan Soal dan Jawaban

Untuk menyelesaikan soal 12.1 di atas, pertama-tama harus dilakukan penelusuran terhadap seluruh isi array. Penelusuran dapat dilakukan dengan memanfaatkan struktur perulangan. Nilai terbesar tidak diketahui posisinya, sehingga penelusuran dapat dilakukan dari sebelah kiri, maupun sebelah kanan. Selanjutnya untuk menentukan bilangan terbesar, dapat digunakan konsep perbandingan bilangan seperti sudah dijelaskan di atas.

Berikut ini algoritme atau kerangka pikir untuk menyelesaikan soal 12.1 di atas:

1. Inisialisasi array dan isinya
2. Inisialisasi variabel max untuk menyimpan nilai terbesar
3. Elemen pertama dari array anggap sebagai nilai terbesar
4. Lakukan penelusuran array dari elemen kedua hingga terakhir.
5. Bandingkan setiap elemen array dengan isi variabel max. Jika lebih besar dari max, maka ganti nilai variabel max dengan nilai tersebut.
6. Selesai penelusuran, cetak isi variabel max sebagai variabel terbesar.



Berdasarkan algoritme (kerangka pikir) di atas, selanjutnya dapat diimplementasikan ke dalam bentuk program seperti pada Program 12.1 berikut ini.

Program 2. Mencari dan mencetak nilai terbesar dari array satu dimensi

```
1  #include <stdio.h>
2  int main()
3  {
4      int A[11]={12,17,10,5,15,25,11,7,25,16,19};
5      int I, max;
6
7      //cetak isi array
8      for (I=0; I<11; I++) {
9          printf("%3i", A[I]);
10     }
11
12     I = 1;
13     max = A[0];
14     while(I<=10) {
15         if(A[I] > max) {
16             max = A[I];
17         }
18         I++;
19     }
20     printf("\nNilai terbesar : %i", max);
21     return 0;
22 }
```

1.3. TEKNIK PENCARIAN NILAI TERBESAR / TERKECIL DENGAN ALGORITME SEKUENSIAL

Teknik pencarian nilai pada array dengan algoritme sekuensial sebenarnya sudah dijelaskan pada bagian 1.2 di atas. Teknik sekuensial pada dasarnya melakukan penelusuran array secara berurut baik dari sebelah kiri (indeks terkecil) maupun dari sebelah kanan (indeks terbesar). Sebagai teknik dasar, kelebihan teknik ini tentunya dari sisi kemudahan dalam memahaminya. Sementara kekurangannya adalah tidak efektif secara komputasi, terutama pada data yang sangat banyak dan untuk posisi nilai berada di bagian akhir dari array.

Perhatikan kembali contoh Soal 12.1. Kita dapat menyelesaikan soal tersebut dengan cara yang sedikit berbeda, namun tetap menggunakan teknik penelusuran sekuensial. Perhatikan Program 3 berikut ini.



Program 3. Teknik pencarian nilai terbesar dengan algoritme sekuensial

```
1  #include <stdio.h>
2  int main()
3  {
4      int A[11]={12,17,10,5,15,25,11,7,25,16,19};
5      int I, J;
6
7      //cetak isi array
8      for (I=0; I<11; I++) {
9          printf("%3i", A[I]);
10     }
11
12     I = 0;
13     J = 0;
14     while(I<=10) {
15         if(A[I] > A[J]) {
16             J = I;
17         }
18         I++;
19     }
20     printf("\nNilai terbesar : %i", A[J]);
21     return 0;
22 }
```

Pada program 3 di atas, tidak menggunakan variabel tertentu yang menyimpan nilai terbesar, namun yang disimpan adalah nomor indeks (variabel J) yang mengandung nilai terbesar. Secara penelusuran, program 3 melakukan penelusuran dari awal array. Menurut Anda, antara program 2 dan program 3, mana yang lebih mudah dipahami?

1.4. TEKNIK PENCARIAN NILAI TERBESAR / TERKECIL DENGAN ALGORITME SENTINEL

Pencarian dengan teknik SENTINEL merupakan pengembangan dari algoritme pencarian sekuensial. Teknik ini tetap membutuhkan penelusuran array secara berurutan (sekuensial), baik dari depan maupun dari belakang. Perbedaannya adalah, pada teknik sentinel, ditambahkan satu elemen di akhir array untuk menyimpan nilai yang dicari. Berdasarkan soal nomor 12.1, posisi elemen sentinel dapat diilustrasikan pada gambar berikut ini. Elemen array indeks ke-11 merupakan elemen sentinel yang akan menyimpan nilai yang dicari.



0	1	2	3	4	5	6	7	8	9	10	11
12	17	10	5	15	25	11	7	25	16	19	0

SENTINEL

Program 4 berikut ini merupakan contoh program untuk mencari nilai TERKECIL dengan teknik SENTINEL.

Program 4. Teknik pencarian pada array dengan teknik SENTINEL

```

1  #include <stdio.h>
2  int main()
3  {
4      int A[12]={12,17,10,5,15,25,11,7,25,16,19};
5      int I;
6
7      //cetak isi array
8      for (I=0; I<11; I++) {
9          printf("%3i", A[I]);
10     }
11
12     I = 1;
13     A[11] = A[0];
14     while(I<=10) {
15         if(A[I] < A[11]) {
16             A[11] = A[I];
17         }
18         I++;
19     }
20     printf("\nNilai terkecil : %i", A[11]);
21     return 0;
22 }

```

Perhatikan deklarasi array pada baris ke-4 Program 4 di atas. Pada bagian deklarasi, ditambahkan satu elemen sentinel di bagian akhir array. Pada awal penelusuran, elemen sentinel diisi dengan isi dari elemen pertama array (A[0]) sebagai nilai terkecil. Selanjutnya dilakukan penelusuran sekuensial dan perbandingan setiap elemen dengan nilai yang tersimpan pada elemen sentinel. Pada akhir perulangan, nilai TERKECIL akan tersimpan pada elemen sentinel yaitu A[11].



KESIMPULAN

Penelusuran array merupakan salah satu proses dasar di dalam algoritme dan pemrograman. Penelusuran array diperlukan di banyak algoritme yang melibatkan data dalam jumlah banyak. Beberapa contoh algoritme yang memerlukan konsep penelusuran array antara lain algoritme pencarian (searching), pengurutan (sorting), penggabungan (merging) dan pembagian / pemecahan (splitting/dividing). Oleh karena itu, memahami konsep penelusuran array merupakan salah satu kompetensi yang wajib dimiliki oleh seorang programmer.

SOAL LATIHAN

Soal 1.

Sudah ada array satu dimensi yang dibuat dengan int A[11]. Sudah ada isinya dengan ilustrasi sebagai berikut :

0	1	2	3	4	5	6	7	8	9	10
12	17	10	5	15	25	11	7	25	16	19

Susun program untuk:

- Mencari dan mencetak isi array yang nilainya TERKECIL. Untuk contoh data diatas, bila program dijalankan maka akan tercetak : **5**.
- Mencetak ada berapa nilai TERKECIL dalam array tersebut. Untuk contoh data diatas, bila program dijalankan maka akan tercetak : **1**.
- Mencetak berada di posisi (index) berapa, nilai TERKECIL dalam array tersebut. Untuk contoh data diatas, bila program dijalankan maka akan tercetak : **3**.

Soal 2.

Sudah ada array satu dimensi yang dibuat dengan int A[10], belum ada isinya. Susun program untuk menginput nilai 10 mahasiswa ke dalam array tersebut. Selanjutnya



cetak keseluruhan isi array, cetak nilai TERBESAR, dan cetak JUMLAH mahasiswa yang mendapat nilai terbesar tersebut !

Soal 3.

Sudah ada array satu dimensi yang dibuat dengan int A[10], belum ada isinya. Susun program untuk menginput nilai 10 mahasiswa ke dalam array tersebut. Selanjutnya hitung dan cetak rata-rata nilai mahasiswa, serta cetak JUMLAH mahasiswa yang mendapat nilai di bawah rata-rata!

Soal 4 (*credit to HackerRank.com*)

Given five positive integers, find the minimum and maximum values that can be calculated by summing exactly four of the five integers. Then print the respective minimum and maximum values as a single line of two space-separated long integers.

For example, $arr = [1, 3, 5, 7, 9]$. Our minimum sum is $1 + 3 + 5 + 7 = 16$ and our maximum sum is $3 + 5 + 7 + 9 = 24$. We would print

```
16 24
```

Function Description

Complete the `miniMaxSum` function in the editor below. It should print two space-separated integers on one line: the minimum sum and the maximum sum of 4 of 5 elements.

`miniMaxSum` has the following parameter(s):

- `arr`: an array of 5 integers

Input Format

A single line of five space-separated integers.

Constraints

$$1 \leq arr[i] \leq 10^9$$

Output Format

Print two space-separated long integers denoting the respective minimum and maximum values that can be calculated by summing exactly four of the five integers. (The output can be greater than a 32 bit integer.)



Sample Input

```
1 2 3 4 5
```

Sample Output

```
10 14
```

Explanation

Our initial numbers are **1**, **2**, **3**, **4**, and **5**. We can calculate the following sums using four of the five integers:

1. If we sum everything except **1**, our sum is $2 + 3 + 4 + 5 = 14$.
2. If we sum everything except **2**, our sum is $1 + 3 + 4 + 5 = 13$.
3. If we sum everything except **3**, our sum is $1 + 2 + 4 + 5 = 12$.
4. If we sum everything except **4**, our sum is $1 + 2 + 3 + 5 = 11$.
5. If we sum everything except **5**, our sum is $1 + 2 + 3 + 4 = 10$.

Hints: Beware of integer overflow! Use 64-bit Integer.





**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

Jl. Raya Ciledug, Petukangan Utara, Pesanggrahan

Jakarta Selatan, 12260

Telp: 021-5853753 Fax : 021-5853752

<http://fti.budiluhur.ac.id>