

MODUL MATA KULIAH

BAHASA PEMROGRAMAN DASAR

PG168 - 3 SKS



**FAKULTAS TEKNOLOGI
INFORMASI
UNIVERSITAS BUDI LUHUR**

**JAKARTA
SEPTEMBER 2019**

TIM PENYUSUN

Dr. Achmad Solichin, M.T.I.
Agnes Aryasanti, M.Kom.
Joko Christian C, M.Kom.
Painem, M.Kom.
Tri Ika Jaya, M.Kom.



MODUL PERKULIAHAN #12

PENCARIAN NILAI TERBESAR DAN TERKECIL

Capaian Pembelajaran	:	Mahasiswa Mampu: <ol style="list-style-type: none">1. Teknik perbandingan sebagian dasar penentuan nilai terbesar dan terkecil.2. Teknik pencarian nilai terbesar atau terkecil pada array satu dimensi.3. Teknik pencarian nilai terbesar atau terkecil menggunakan algoritma pencarian sekuensial dan sentinel.
Sub Bahasan	Pokok :	<ol style="list-style-type: none">1. Teknik pencarian suatu nilai pada array satu dimensi menggunakan algoritma sentinel.2. Teknik pencarian nilai terbesar atau terkecil pada array satu dimensi menggunakan algoritma pencarian sekuensial
Daftar Pustaka	:	<ol style="list-style-type: none">1. Sjukani, Moh. (2014). Algoritma (Algoritma dan Struktur Data 1) Dengan C++, dan Java Edisi 9. Jakarta : Mitra Wacana Media.2. Hartono, J., 2001. Konsep Dasar Pemrograman Bahasa C 2nd ed., Yogyakarta: ANDI.3. Kristanto, Andi. (2009). Algoritma & Pemrograman dengan C++ Edisi 2. Yogyakarta : Graha Ilmu.

PRAKTIKUM 12

PENCARIAN NILAI TERBESAR DAN TERKECIL

12.1. Teori Singkat

Pencarian pada array dapat menggunakan banyak pendekatan. Pada praktikum kali ini kita akan membahas 2 pendekatan, yaitu sekuensial, dan sentinel.

Pendekatan sekuensial adalah pendekatan paling mudah yang menggunakan perulangan sebanyak jumlah elemen array dan mencari satu persatu hingga ditemukan elemen dengan isi yang sama dengan nilai yang dicari. Jika sudah ditemukan, maka akan dikembalikan nilai *i* (nilai indeks).

Pendekatan sentinel adalah pendekatan yang meletakkan nilai pencarian pada elemen terakhir array (ukuran array harus lebih besar satu dari jumlah elemen yang dicari), dari sinilah istilah sentinel muncul karena hanya berupa data sementara. Tujuannya agar loop pencarian bisa berhenti. Perulangan yang digunakan pada pendekatan sentinel lebih efisien.

// sekuensial

```
for (int i = 0; i < panjangArray; i++) {  
    if (array[i] == nilaiYangDicari) { return i; }  
} return -1;
```

// sentinel

```
while (array[i] != nilaiYangDicari) {  
    i++;}  
return i;
```

Pada pola pertama (`//sekuensial`), program harus melakukan perbandingan nilai *i* (yang digunakan sebagai indeks), dan nilai isi `array[i]`. Sehingga diperlukan 2 (dua) kali perbandingan pada setiap loop.

Pada pola kedua (`//sentinel`), program hanya melakukan satu kali perbandingan yaitu nilai isi `array[i]`. Sehingga hanya diperlukan 1(satu) kali perbandingan pada setiap loop.



Meskipun terkesan sepele, namun perbedaan performance pada array yang besar akan signifikan, sangat cocok untuk diaplikasikan pada linked list (materi linked list pada Algoritma 2).

Jadi penerapan algoritma sentinel adalah untuk meningkatkan performa pencarian, dan menurunkan beban processor.

Namun ada beberapa kekurangan :

1. Hasil kembalian program jika nilai yang dicari tidak ditemukan berbeda (dalam contoh diatas adalah -1 pada linier, dan nilai panjang elemen array (length) pada sentinel.
2. Array harus dibuat sedikit lebih besar, untuk menampung nilai sentinel.
3. Array harus bisa dibaca tulis jika panjang array pas-pasan.
4. Tidak cocok untuk pencarian paralel (multi tread) jika ada pencarian lebih dari satu nilai pada array yang sama.

12.2. Praktikum

 Belajar mandiri dengan video praktikum		
Playlist pertemuan 12:	http://tiny.cc/jokoccBPD12	
SUBSCRIBE :	https://www.youtube.com/subscription_center?add_user=realjokocc	

Langkah-langkah Praktikum

1. Buka Editor Bahasa C Dev-C++ 5.11.
2. Buatlah file baru dengan membuka menu File > New > Source File atau dengan
3. Shortcut Ctrl + N.

Perhatikan contoh berikut untuk memahami algoritma pencarian sentinel.

Program 12.0 : pencarian_sentinel.cpp

Tuliskan Program 12.0 berikut ini pada editor Dev-C++ :

```
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      int cari;
7      //cetak isi array
8      for (I=0; I<11; I++) {
9          printf("%3i", A[I]);
10     }
11     printf("\n Inputkan Nilai yang dicari: ");
12     scanf("%i",&cari);
13     // nilai cari dimasukkan di dalam Array agar perulangan bisa berhenti
14     A[11] = cari;
15     I = 0;
16     //metode sentinel, perhatikan bahwa variabel I tidak di cek
17     while(A[I]!=cari) {
18         I++;
19     }
20
21     if(I<11){
22         printf(" Data Ditemukan");
23         printf("\n Pada Indeks ke %i",I);
24     }else {
25         printf(" Data tidak ditemukan");
26     }
27
28     return 0;
29 }
```

Silahkan jalankan program di atas dan uji dengan memasukkan angka 15, 8, dan 16! Menurut kalian apakah penggunaan algoritma sentinel akan lebih efektif untuk array dengan jumlah besar sekali (misalnya diatas 1000 elemen)? Mengapa?

Berikutnya kita akan bahas bagaimana mencari nilai tertinggi dan terendah pada sebuah array dengan algoritma sequensial.

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.

Untuk menyelesaikan persoalan tersebut, ikuti langkah-langkah praktikum sebagai berikut.

1. Tuliskan Program 12.1 berikut ini pada editor Dev-C++. Kompilasi dan jalankan program tersebut!

Program 12.1 : array_max1.cpp

Tuliskan Program 12.1 berikut ini pada editor Dev-C++:

```
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     //metode 1 : simpan nilai terbesar ke variabel max
13     I = 1;
14     max = A[0];
15     while(I<=10) {
16         if(A[I] > max) {
17             max = A[I];
18         }
19         I++;
20     }
21     printf("\nNilai terbesar : %i", max);
22     return 0;
23 }
```

Tuliskan tampilan program di atas! Apakah program menampilkan nilai TERBESAR dengan benar? Berapa kali blok perulangan WHILE pada baris 15-18 dieksekusi?

2. Selanjutnya perhatikan Program 12.2 berikut ini. Kompilasi dan jalankan program tersebut!

Program 12.2: array_max2.cpp

Tuliskan Program 12.2 berikut ini pada editor Dev-C++:

```
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     //metode 2 : simpan index dari elemen dengan
13     //nilai terbesar ke variabel j
14     I = 0;
15     J = 0;
16     while(I<=10) {
17         if(A[I] > A[J]) {
18             J = I;
19         }
20         I++;
21     }
22     printf("\nNilai terbesar : %i", A[J]);
23     printf("\nyang terletak pada index ke : %i", J);
24     return 0;
25 }
```

Tuliskan tampilan program di atas! Apakah program menampilkan nilai TERBESAR dengan benar? Berapa kali blok perulangan WHILE pada baris 15-18 dieksekusi?

3. Program 12.3 berikut ini merupakan cara lain mencari nilai terbesar dari serangkaian nilai di dalam array. Kompilasi dan jalankan program tersebut!
- Program 12.3: array_max3.cpp**


```

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     //metode 3 : copy elemen dengan nilai
13     //terbesar ke elemen paling akhir
14     I = 1;
15     A[11] = A[0];
16     while(I<=10) {
17         if(A[I] > A[11]) {
18             A[11] = A[I];
19         }
20         I++;
21     }
22     printf("\nNilai terbesar : %i", A[11]);
23     return 0;
24 }

```

Tuliskan tampilan program di atas! Apakah program menampilkan nilai TERBESAR dengan benar? Berapa kali blok perulangan WHILE pada baris 15-18 dieksekusi?

4. Dari Program 12.1, 12.2 dan 12.3, untuk mencari nilai terbesar di dalam array, manakah yang menurut Anda paling efektif? Mengapa? Dan apakah terdapat teknik, algoritma atau metode pencarian nilai TERBESAR yang lebih efektif dan tidak perlu melakukan penelusuran ke seluruh elemen array?

5. Selanjutnya jalankan Program 12.4 berikut ini. Catat hasilnya!

Program 12.4: array_max4.cpp

Tuliskan Program 12.4 berikut ini pada editor Dev-C++:

```
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, jum=1;
6
7      //cetak isi array
8      for (I=0; I<11; I++) {
9          printf("%3i", A[I]);
10     }
11
12     //metode 1 update : hitung jumlah max yang muncul
13     I = 1;
14     max = A[0];
15     while(I<=10) {
16         if(A[I] > max) {
17             max = A[I];
18             jum = 1;
19         } else if (A[I] == max) {
20             jum++;
21         }
22         I++;
23     }
24     printf("\nNilai terbesar : %i", max);
25     printf("\nBanyaknya Nilai terbesar : %i", jum);
26     return 0;
27 }
```

Tuliskan tampilan program di atas!

6. Ubahlah program 12.4 di atas sedemikian hingga tidak hanya menampilkan nilai TERBESAR dan JUMLAH terbesar, namun juga menampilkan POSISI atau index dimana nilai terbesar berada! Tuliskan program pada form berikut ini!

12.3. Latihan

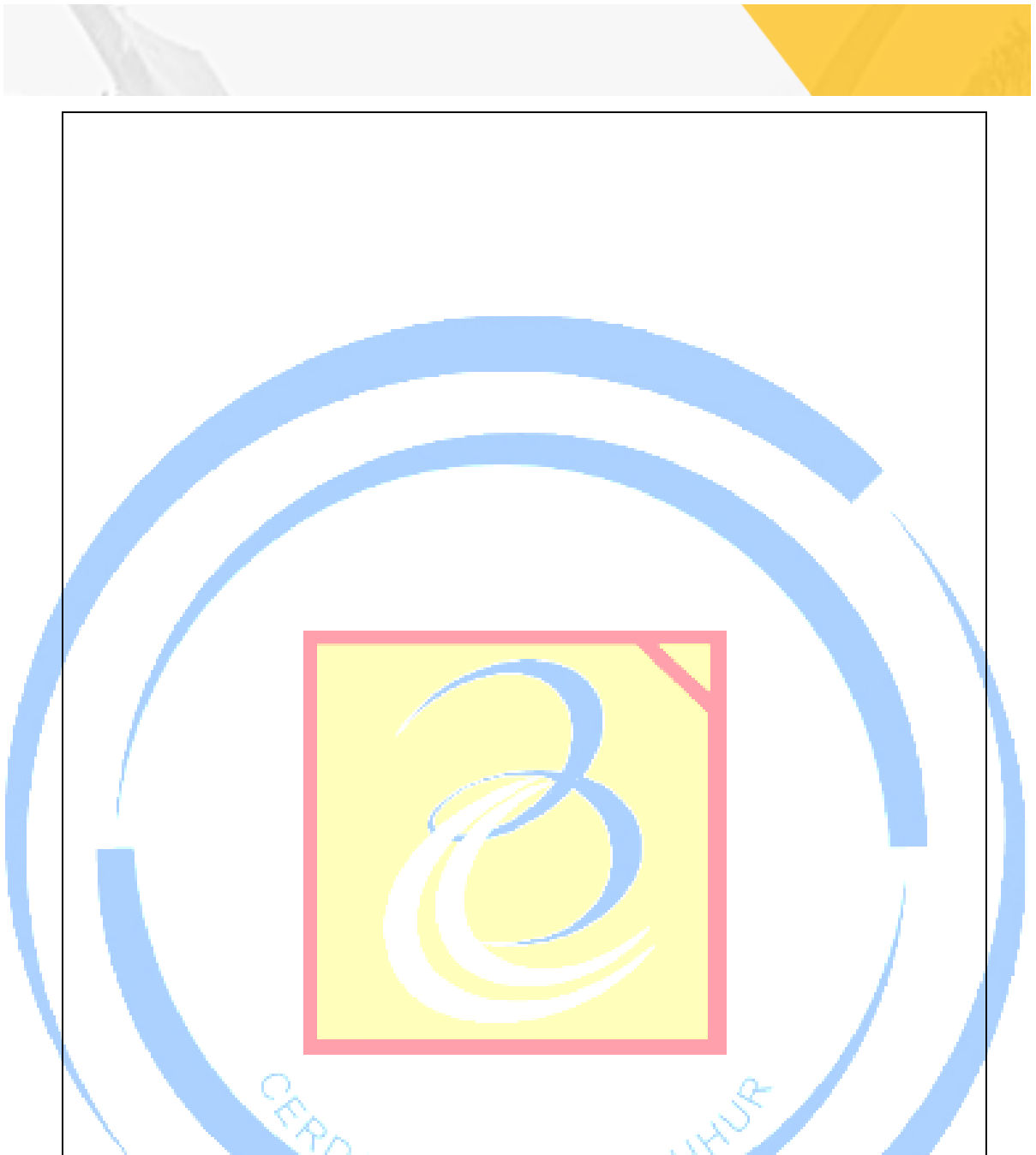
Sudah ada array satu dimensi yang dibuat dengan int A[11]. Sudah ada isinya dengan ilustrasi sebagai berikut :

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 :

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

Tuliskan program Anda pada form berikut ini.



12.4. Tugas Mandiri

1. 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 !
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 hitung dan cetak rata-rata nilai mahasiswa, serta cetak JUMLAH mahasiswa yang mendapat nilai di bawah rata-rata!



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>