



PROGRAM STUDI
TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO

Mata Kuliah
Dasar Pemrograman



Analisis Kasus Tunggal

TIM DASAR PEMROGRAMAN
TEKNIK INFORMATIKA S1
UNIVERSITAS DIAN NUSWANTORO

Capaian Pembelajaran

- Setelah mengikuti kuliah ini mahasiswa dapat menjelaskan dan mempraktekkan analisis satu kasus dengan kondisi diterapkan pada aksi sekuensial permasalahan komputasional sederhana
- Setelah mengikuti kuliah ini mahasiswa dapat menjelaskan dan mempraktekkan analisis dua kasus komplemen dengan kondisi diterapkan pada aksi sekuensial permasalahan komputasional sederhana
- Setelah mengikuti kuliah ini mahasiswa dapat menjelaskan dan mempraktekkan analisis kasus bersarang diterapkan pada aksi sekuensial permasalahan komputasional sederhana



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO**

Analisis Kasus

Analisis Satu Kasus Bahasa C/C++

- Aturan bahasa C/C++ (Sintak):

```
if(<kondisi>)
```

```
{
```

```
    <aksi .... >
```

```
    <aksi .... bisa lebih dari satu>
```

```
}
```

Analisis Satu Kasus Bahasa C/C++

- Contoh untuk mencetak a jika lebih dari 0:

```
if(a>0)
```

```
{
```

```
    cout << a;
```

```
}
```

Perhatikan tidak ada titik koma dibagian kurung kondisi dan tidak ada titik koma di akhir scope dari if atau kurung kurawal penutup!

Program Validasi

Program Validasi

{Program untuk mengecek suatu bilangan bulat apakah lebih dari 0}

KAMUS

a : integer

ALGORITMA

input(a)

if a>0 then

output("valid")

endif

output("selesai cek")

Apa yang terjadi jika input a bernilai 0?

Program Validasi dengan bahasa C++

```
main.cpp x
1  /*
2  JUDUL: Program Validasi
3  {Program untuk mengecek suatu bilangan bulat apakah lebih dari 0}
4  */
5  #include <iostream>
6
7  using namespace std;
8
9  int main()
10 {
11     //Kamus
12     int a;
13
14     //Algoritma
15     cin >> a;
16     if(a > 0)
17     {
18         cout << "valid" << endl;
19     }
20     cout << "selesai";
21
22     return 0;
23 }
```

Program ValidasiByConstant

Program ValidasiByConstant

{Program untuk mengecek suatu bilangan bulat apakah lebih dari suatu nilai variabel konstanta}

KAMUS

a : integer

constant X : integer \leftarrow 10

ALGORITMA

input(a)

if a>X then

output(a)

endif

output("selesai cek")

Apa yang terjadi jika input a kurang dari konstanta X?

Program ValidasiByConstant dengan bahasa C++

```
main.cpp x
1  /*
2  JUDUL: Program ValidasiByConstant
3  {Program untuk mengecek suatu bilangan bulat apakah lebih dari suatu nilai variabel konstanta}
4  */
5  #include <iostream>
6
7  using namespace std;
8
9  int main()
10 {
11     //Kamus
12     int a;
13     const int X = 10;
14
15     //Algoritma
16     cin >> a;
17     if(a > X)
18     {
19         cout << a << endl;
20     }
21     cout << "selesai";
22
23     return 0;
24 }
```

Spesifikasi Program

- Terkadang notasi algoritmik diawali dengan penjelasan. Hal ini sangat penting jika notasi algoritmik cukup sulit dipahami secara langsung.
- Contoh:

Spesifikasi Program GenapLebihdariDua.

Input: a merupakan bilangan bulat

Proses: cek apakah a bilangan genap yang lebih dari 2.

Output: tulisan “<nilai a> adalah bilangan genap lebih dari 2”

Program GenapLebihdariDua

Program GenapLebihdariDua

{Program untuk mengecek apakah a bilangan genap yang lebih dari 2}

KAMUS

a : integer

ALGORITMA

input(a)

if ((a mod 2 eq 0) and (a>2)) then

output(a,"adalah bilangan genap lebih dari 2")

endif

output("selesai cek")

Program GenapLebihdariDua dengan bahasa C++

```
main.cpp X
1  /*
2  JUDUL: Program GenapLebihdariDua
3  {Program untuk mengecek apakah a bilangan genap yang lebih dari 2}
4  */
5  #include <iostream>
6
7  using namespace std;
8
9  int main()
10 {
11     //Kamus
12     int a;
13
14     //Algoritma
15     cin >> a;
16     if((a%2==0) && (a>2))
17     {
18         cout << a << endl;
19     }
20     cout << "selesai";
21
22     return 0;
23 }
24
```

Persoalan Komputasional

- Deskripsi: Mawar merupakan anak yang suka sekali dengan bilangan bulat. Dia ingin bilangan bulat yang dia temukan selalu bernilai positif. Mengingat bilangan bulat bisa juga bernilai negatif. Bantu Mawar untuk membuat program agar bilangan bulat negatif yang dia temui selalu menjadi bilangan bulat positif saat dioutputkan.
- Spesifikasi:
 - Input: x merupakan bilangan bulat bisa positif atau negatif
 - Output: x yang selalu bilang bulat positif
- Contoh input/output:
 - Input: -10
 - Output: 10

Cara menjawab

1. Dibutuhkan suatu variabel x yang bertipe bulat yang dapat di input oleh user
2. Proses yang terjadi adalah jika x bernilai negatif maka rubah x sebelumnya menjadi x yang baru dengan cara dikalikan dengan -1 .
3. Ingat bilangan bulat negatif dikali negatif hasilnya positif.
4. Jika user sudah menginputkan x bernilai positif tidak perlu pengecekan langkah 3 dan 4 dilakukan.

Program MawarAbsolut

Program MawarAbsolut

{Program untuk mengoutputkan bilangan bulat positif jika user menginputkan negatif}

KAMUS

x : integer

ALGORITMA

input(x)

if (x<0) then

 x = x * -1

endif

output(x)

Program MawarAbsolut Bahasa C++

```
main.cpp x
1  /*
2  JUDUL: Program MawarAbsolut
3  {Program untuk mengoutputkan bilangan bulat positif jika user menginputkan negatif}
4  */
5  #include <iostream>
6
7  using namespace std;
8
9  int main()
10 {
11     //Kamus
12     int x;
13
14     //Algoritma
15     cin >> x;
16     if(x<0)
17     {
18         x = x * -1;
19     }
20     cout << x;
21
22     return 0;
23 }
```




**FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO**

Analisis 2 Kasus

Analisis dua Kasus komplemen Bahasa C/C++

- Aturan bahasa C/C++ (Sintak):

```
if(<kondisi>)
```

```
{
```

```
    <aksi .... >
```

```
    <aksi .... bisa lebih dari satu>
```

```
}
```

```
else
```

```
{
```

```
    <aksi .... >
```

```
    <aksi .... bisa lebih dari satu>
```

```
}
```

Analisis Satu Kasus Bahasa C/C++

- Contoh untuk mencetak a jika lebih dari 0:

```
if(a>0)
{
    cout << "ya";
}
else
{
    cout << "tidak";
}
```

Analisis Dua Kasus Komplemen Bentuk Ternary Operator

- Dengan operator ?
- Dilakukan dalam satu baris program
- Contoh:
 `s = (x>0)? "ya":"tidak";`
 `cout << s;`
- s bertipe string atau char[5].

Program GenapGanjil

Program GenapGanjil

{Program untuk mengecek apakah a bilangan genap atau ganjil}

KAMUS

x : integer

ALGORITMA

input(x)

if x mod 2 eq 0 then

output("genap")

else

output("ganjil")

Program GenapGanjil di Bahasa C++

```
main.cpp x
1  /*
2  JUDUL: Program GenapGanjil
3  {Program untuk mengecek apakah a bilangan genap atau ganjil}
4  */
5  #include <iostream>
6
7  using namespace std;
8
9  int main()
10 {
11     //Kamus
12     int x;
13
14     //Algoritma
15     cin >> x;
16     if(x%2==0)
17     {
18         cout << "genap";
19     }
20     else
21     {
22         cout << "ganjil";
23     }
24
25     return 0;
26 }
```

Program MaxAB

Program MaxAB

{Program untuk mencetak bilangan dengan nilai maksimal dua bilangan bulat a dan b}

KAMUS

a , b : integer

ALGORITMA

input(a)

input(b)

if a >= b then

output(a)

else

output(b)

endif

Program MaxAB di Bahasa C++

```
main.cpp x
1  /*
2  JUDUL: Program MaxAB
3  {Program untuk mencetak bilangan dengan nilai maksimal dua bilangan bulat a dan b}
4  */
5  #include <iostream>
6
7  using namespace std;
8
9  int main()
10 {
11     //Kamus
12     int a,b;
13
14     //Algoritma
15     cin >> a;
16     cin >> b;
17     if(a>=b)
18     {
19         cout << a;
20     }
21     else
22     {
23         cout << b;
24     }
25
26     return 0;
27 }
```


Program Mawar5

Program Mawar5
{Program untuk penyelesaian kasus kelipatan 5 si Mawar}

KAMUS

a : integer

ALGORITMA

input(a)

if a < 0 then

 a ← a * -1

endif

if a mod 5 eq 0 then

output("hore ketemu bilangan kelipatan 5")

else

output("yah.. bukan bilangan kelipatan 5")

endif

Program Mawar5 di Bahasa C++

```
main.cpp x
1  /*
2  JUDUL: Program Mawar5
3  {Program untuk penyelesaian kasus kelipatan 5 si Mawar}
4  */
5  #include <iostream>
6
7  using namespace std;
8
9  int main()
10 {
11     //Kamus
12     int a;
13
14     //Algoritma
15     cin >> a;
16     if(a<0)
17     {
18         a = a * -1;
19     }
20
21     if(a%5==0)
22     {
23         cout << "hore ketemu bilangan kelipatan 5";
24     }
25     else
26     {
27         cout << "yah.. bukan bilangan kelipatan 5";
28     }
29
30     return 0;
31 }
```

Program Mawar5 di Bahasa C++

```
main.cpp x
1  /*
2  JUDUL: Program Mawar5
3  {Program untuk penyelesaian kasus kelipatan 5 si Mawar}
4  */
5  #include <iostream>
6
7  using namespace std;
8
9  int main()
10 {
11     //Kamus
12     int a;
13
14     //Algoritma
15     cin >> a;
16     if(a<0)
17     {
18         a = a * -1;
19     }
20
21     if(a%5==0)
22     {
23         cout << "hore ketemu bilangan kelipatan 5";
24     }
25     else
26     {
27         cout << "yah.. bukan bilangan kelipatan 5";
28     }
29
30     return 0;
31 }
```

Program Mawar5 di Bahasa C++ (lazy version)

```
main.cpp x
1  /*
2  JUDUL: Program Mawar5
3  {Program untuk penyelesaian kasus kelipatan 5 si Mawar}
4  */
5  #include <iostream>
6
7  using namespace std;
8
9  int main()
10 {
11     //Kamus
12     int a;
13     string hasil;
14
15     //Algoritma
16     cin >> a;
17
18     a = (a<0)? a * -1:a;
19
20     hasil = (a%5==0)? "hore ketemu bilangan kelipatan 5":"yah.. bukan bilangan kelipatan 5";
21
22     cout << hasil;
23
24     return 0;
25 }
```

Referensi

Utama:

1. Bjarne Stroustrup, 2014, Programming: Principles and Practice Using C++ (Second Edition), Addison-Wesley Professional

Pendukung:

1. Introduction to Computer Science and Programming in Python, MIT
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016>
2. Introduction to Computer Science and Programming, MIT
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00sc-introduction-to-computer-science-and-programming-spring-2011/index.htm>



TERIMA KASIH

ANY QUESTIONS?