



PROGRAM STUDI
TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO

Mata Kuliah
Dasar Pemrograman



Analisis Kasus – Banyak Kasus

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

Analisis banyak Kasus Bahasa C/C++

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

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

```
{
```

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

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

```
}
```

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

```
{
```

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

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

```
}
```

```
...
```

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

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

```
{
```

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

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

```
}
```

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

```
{
```

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

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

```
}
```

```
...
```

```
else
```

```
{
```

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

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

```
}
```

Analisis banyak Kasus dengan ekspresi konstan Bahasa C/C++

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

```
switch(<name>)
```

```
{
```

```
    case <nilai-konstan>:    <aksi-1>  
                           break;
```

```
    case <nilai-konstan>:    <aksi-2>  
                           break;
```

```
    ....
```

```
    case <nilai-konstan>:    <aksi-n>  
                           break;
```

```
    default:                <aksi-n>  
                           break;
```

```
}
```

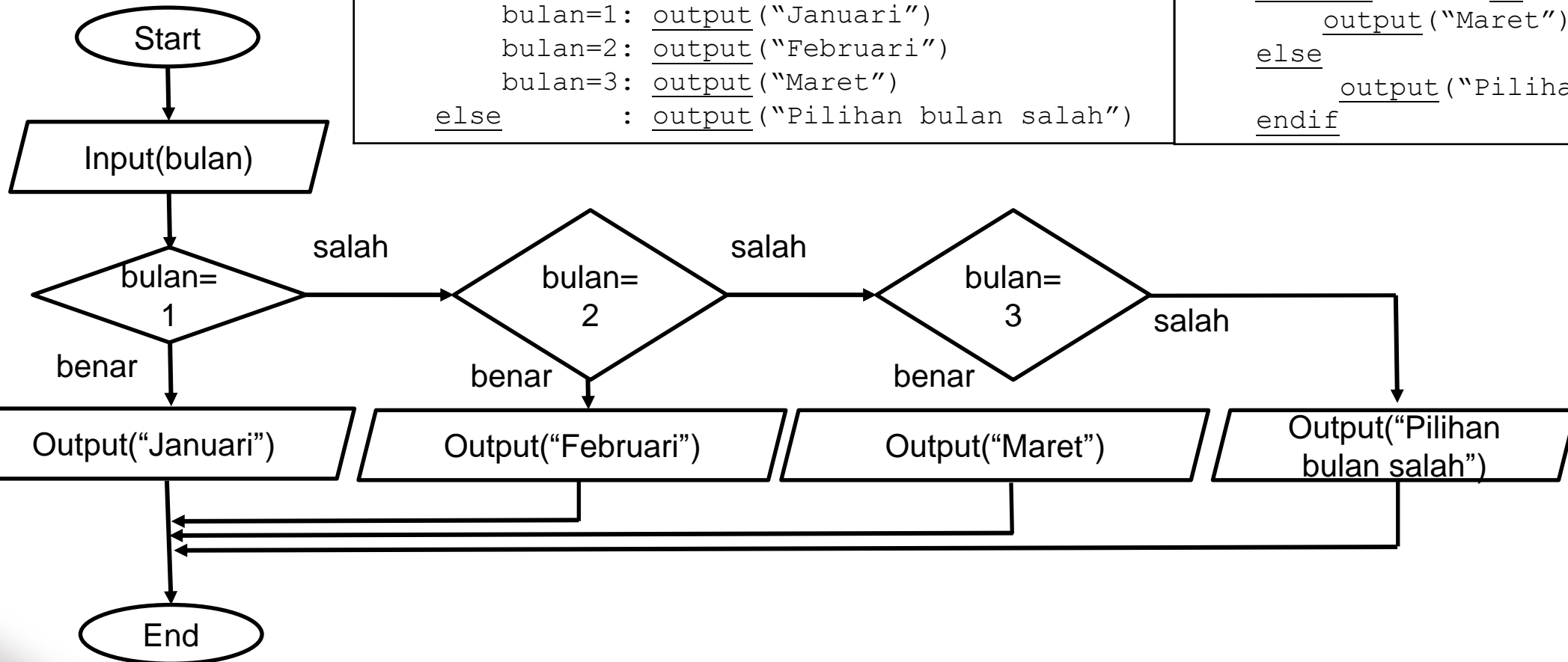
Contoh Ekspresi Konstan

Algoritma

```
input (bulan)
depend on bulan
    bulan=1: output ("Januari")
    bulan=2: output ("Februari")
    bulan=3: output ("Maret")
else      : output ("Pilihan bulan salah")
```

Algoritma

```
input (bulan)
if bulan eq 1 then
    output ("Januari")
else if bulan eq 2 then
    output ("Februari")
else if bulan eq 3 then
    output ("Maret")
else
    output ("Pilihan bulan salah")
endif
```



Contoh Ekspresi Konstan

Algoritma

```

input (bulan)
depend on bulan
    bulan=1: output ("Januari")
    bulan=2: output ("Februari")
    bulan=3: output ("Maret")
else      : output ("Pilihan bulan salah")

```

```

int bulan;
cin >> bulan;
switch (bulan) {
    case 1:{
        string b = "Januari"; //lebih dari 1 aksi
        cout << b;
        break;
    }
    case 2:
        cout << "Februari"; //1 aksi
        break;
    case 3:
        cout << "Maret";
        break;
    default:
        cout << "Pilihan bulan salah" << endl;
}

```

Algoritma

```

input (bulan)
if bulan eq 1 then
    output ("Januari")
else if bulan eq 2 then
    output ("Februari")
else if bulan eq 3 then
    output ("Maret")
else
    output ("Pilihan bulan salah")
endif

```

```

int bulan;
cin >> bulan;
if (bulan==1)
{
    string b = "Januari"; //lebih dari 1 aksi
    cout << b;
}
else if (bulan==2)
    cout << "Februari"; //1 aksi
else if (bulan==3)
    cout << "Maret";
else
    cout << "Pilihan bulan salah" << endl;

```

Contoh Switch Case menggunakan Char

```
char bulan;  
cin >> bulan;  
switch (bulan) {  
    case 'a':{  
        string b = "Januari"; //lebih dari 1 aksi  
        cout << b;  
        break;  
    }  
    case 'b':  
        cout << "Februari"; //1 aksi  
        break;  
    case 'c':  
        cout << "Maret";  
        break;  
    default:  
        cout << "Pilihan bulan salah" << endl;  
}
```

Notasi Algoritmik (Cek Bilangan Bulat)

Program CekBilanganBulat

{Program dengan notasi algoritmik untuk mengecek bilangan bulat positif/negatif/NOL}

KAMUS

x : integer

ALGORITMA

input(x)

if x > 0 then

output("positif")

else if x < 0 then

output("negatif")

else

output("no")

output("finish")

main.cpp x

```
1  /*
2  JUDUL: Program CekBilanganBulat
3  {Program dengan notasi algoritmik untuk mengecek bilangan bulat positif/negatif/NOL}
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         cout << "positif" << endl;
18     else if(x < 0)
19         cout << "negatif" << endl;
20     else
21         cout << "nol" << endl;
22     cout << "finish";
23
24
25     return 0;
26 }
```

Jika didalam blok if atau else if atau else berisi hanya 1 argumen/aksi boleh tidak menggunakan kurung kurawal

Program WujudAir

Program WujudAir

{menuliskan wujud air sesuai dengan nilai s, s bisa diasumsikan sebagai suhu}
{s dalam hal ini representasikan dengan integer}

KAMUS

s : integer

ALGORITMA

input(s)

depend on (s)

$s \leq 0$: output("beku")

$0 < s \leq 100$: output("cair")

$s > 100$: output("uap")

Program WujudAir

```
main.cpp x
1  /*
2  JUDUL: Program WujudAir
3  {menuliskan wujud air sesuai dengan nilai s, s bisa diasumsikan sebagai suhu}
4  {s dalam hal ini representasikan dengan integer}
5  */
6  #include <iostream>
7
8  using namespace std;
9
10 int main()
11 {
12     //Kamus
13     int s;
14
15     //Algoritma
16     cin >> s;
17     if(s <= 0)
18         cout << "beku" << endl;
19     else if((s > 0) && (s<=100))
20         cout << "cair" << endl;
21     else if(s > 100)
22         cout << "uap" << endl;
23     cout << "finish";
24
25
26     return 0;
27 }
```

Program WujudAir versi 2

Program WujudAir

{menuliskan wujud air sesuai dengan nilai s, s bisa diasumsikan sebagai suhu}
{s dalam hal ini representasikan dengan real}

KAMUS

s : real

ALGORITMA

input(s)

depend on (s)

s < 0 : output("beku")

s = 0 : output("beku-cair")

0 < s < 100 : output("cair")

s > 100 : output("uap")

s = 100 : output("cait-uap")

Kode c++? Silahkan di coba ^^

ng

Kasus Komputasional

- Deskripsi: Mawar merupakan mahasiswa teknik informatika, dia ingin mendapatkan beasiswa, syarat untuk mendapatkan beasiswa dilihat berdasarkan IPK. Hanya mahasiswa yang memiliki IPK tinggi yang dianggap lulus mendapat beasiswa. Jika IPK sedang maka dia dipertimbangkan. Jika IPK rendah tidak lulus. Buat sistem untuk permasalahan tersebut sehingga mawar bisa mengecek apakah dia lulus dapat beasiswa atau tidak.
- Keterangan:
 - $IPK \geq 3.5$ dianggap tinggi (berarti dia akan lulus)
 - IPK minimal 2.5 sampai 3.49 dianggap sedang (berarti dipertimbangkan)
 - IPK dibawah 2.5 dianggap rendah (berarti tidak lulus)
- Input: ipk merupakan bilangan real
- Output: tulisan “lulus” atau “dipertimbangkan” atau “tidak lulus”
- Contoh Input-Output:
 - Input: 3.6
 - Output: lulus

Program Beasiswa

Program Beasiswa
{Program untuk mengecek kelulusan beasiswa berdasarkan IPK}

KAMUS

ipk : real

ALGORITMA

input(ipk)

depend on (ipk)

$ipk \geq 3.5$: output("lulus")

$2.5 \leq ipk < 3.5$: output("dipertimbangkan")

$ipk < 2.5$: output("tidak lulus")

Program Beasiswa

```
main.cpp x
1  /*
2  JUDUL: Program Beasiswa
3  {Program untuk mengecek kelulusan beasiswa berdasarkan IPK}
4  */
5  #include <iostream>
6
7  using namespace std;
8
9  int main()
10 {
11     //Kamus
12     float ipk;
13
14     //Algoritma
15     cin >> ipk;
16     if(ipk >= 3.5)
17         cout << "lulus" << endl;
18     else if((ipk >= 2.5f) && (ipk<3.5f))
19         cout << "dipertimbangkan" << endl;
20     else if(ipk < 2.5f)
21         cout << "tidak lulus" << endl;
22
23
24     return 0;
25 }
```

Program PositifGanjilGenap

Program PositifGanjilGenap

{Program untuk mengecek suatu bilangan positif atau bukan, cek pula bilangan positif tersebut ganjil atau genap?}

KAMUS

x : integer

ALGORITMA

input(x)

if x > 0 then

if x mod 2 eq 0 then

 output("Bilangan positif genap")

else

 output("Bilangan positif ganjil")

else

 output("Bukan bilangan positif")

endif

output("finish")

main.cpp X

```
2  JUDUL: Program PositifGanjilGenap
3  {Program untuk mengecek suatu bilangan positif atau bukan, cek pula bilangan positif tersebut ganjil atau genap?}
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         if(x%2==0)
19         {
20             cout << "Bilangan positif genap" << endl;
21         }
22         else
23         {
24             cout << "Bilangan positif ganjil" << endl;
25         }
26     }
27     else
28     {
29         cout << "Bukan bilangan positif" << endl;
30     }
31     cout << "finish" << endl;
32
33     return 0;
34 }
```

Program PositifGanjilGenap

Kasus Komputasional

- Deskripsi: Mawar membutuhkan program pendataan uang saku mahasiswa. Syarat utamanya adalah seseorang yang menggunakan sistem tersebut adalah mahasiswa. Kemudian dia akan ditanya berapa uang sakunya. Jika uang sakunya lebih dari 1 juta maka dianggap mahasiswa mampu.. Jika kurang dari itu dianggap kurang mampu. Lalu jika seseorang yang menggunakan sistem tersebut bukan mahasiswa, maka akan mencetak tulisan bukan mahasiswa.
- Input: karakter 'y' atau 't' untuk verifikasi mahasiswa atau tidak dan uang saku dalam tipe integer
- Output: beberapa tulisan

Program Uangsaku

{karakter 'y' atau 't' untuk verifikasi mahasiswa atau tidak dan uang saku dalam tipe integer, digunakan untuk mengecek mahasiswa mampu atau kurang mampu}

KAMUS

c : char

u : integer

ALGORITMA

output("Apakah anda mahasiswa? (y/t) ")

input(c)

if c eq 'y' then

output("Masukkan uang saku per-bulan? ")

input(u)

if u > 1000000 then

output("Mahasiswa mampu")

else

output("Mahasiswa kurang mampu")

else

output("Bukan Mahasiswa")

Program UangSaku

```
main.cpp x
3  /*(karakter 'y' atau 't' untuk verifikasi mahasiswa atau tidak dan uang saku dalam tipe integer, digunakan untuk mengecek mahasiswa mampu atau kurang mampu)
4  */
5  #include <iostream>
6
7  using namespace std;
8
9  int main()
10 {
11     //Kamus
12     char c;
13     int u;
14     //Algoritma
15     cout << "Apakah anda mahasiswa? (y/t)";
16     cin >> c;
17     if(c == 'y')
18     {
19         cout << "Masukkan uang saku per-bulan? ";
20         cin >> u;
21         if(u > 1000000)
22         {
23             cout << "Mahasiswa mampu" << endl;
24         }
25         else
26         {
27             cout << "Mahasiswa kurang mampu" << endl;
28         }
29     }
30     else
31     {
32         cout << "Bukan mahasiswa" << endl;
33     }
34     return 0;
35 }
```

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?