



- Dasar Pemrograman – Pertemuan 4

Tim Bahan Ajar Dasar Pemrograman
Teknik Informatika - S1
Fakultas Ilmu Komputer



Review Materi





Capaian Kuliah Pertemuan 4



Capaian Pembelajaran

- Setelah mengikuti matakuliah ini mahasiswa dapat menjelaskan, membuat, dan mempraktikkan analisis satu kasus diterapkan pada aksi sekuensial permasalahan komputasional sederhana.



ANALISIS KASUS



Analisis Kasus

- Memungkinkan untuk membuat teks algoritma yang sama tetapi menghasilkan eksekusi yang berbeda-beda
- Analisa kondisi / Aliran kendali / Pencabangan / *Branching*
- Terdiri dari:
 - Kondisi: ekspresi yang bernilai benar / salah
 - Aksi: instruksi yang akan dilakukan jika kondisi yang dipasangkan dipenuhi, dapat berupa ekspresi atau perintah dasar lain

Analisis Satu Kasus

- Pengecekan terhadap satu kondisi yang memenuhi syarat.
 - Jika benar lakukan aksi, jika salah abaikan
- Menggunakan keyword if ... then ... ditutup dengan endif

```
if <kondisi> then  
    <aksi>  
    <aksi>  
endif
```



Perhatikan
indentasi !

Kondisi didalam If

- Merupakan ekspresi boolean, sehingga akan menghasilkan nilai True atau False.
- Untuk analisis satu kasus, **jika kondisi bernilai benar maka aksi akan dilakukan, tetapi jika kondisi bernilai salah maka tidak akan terjadi apa-apa (efek neto “kosong”)**

Ekspresi Boolean yang panjang dalam satu kondisi

- Sebagai contoh, suatu bilangan bulat a akan dicek kondisinya dengan syarat, dia adalah bilangan genap dan bilangan tersebut harus lebih dari 2.
- Bagaimana kita menuliskan ekspresi booleannya?
 - Asumsi a terdefinisi dengan input yang akan dimasukkan oleh user.
 - Pengecekan a bilangan genap? $a \bmod 2 \text{ eq } 0$ (dibaca: sisa bagi dari a dibagi 2 adalah 0)
 - Pengecekan a lebih dari 2? $a > 2$ (dibaca: a lebih dari 2)
 - Kedua ekspresi diatas perlu di gabung dengan operator “and” sehingga ekspresi boolean yang utuh adalah $((a \% 2 \text{ eq } 0) \text{ and } a > 2)$

Scope dalam pemrograman

- Scope atau lingkup disini penting sekali
- Untuk notasi algoritmik ditandai dengan tab atau 4 spasi, tetapi untuk pemrograman bahasa C++ ditandai dengan { dan }
- Dia bermanfaat untuk dimana suatu variabel itu dikenal dan kapan suatu variabel itu akan tidak terpakai lagi atau dihancurkan (terhapus dari memory)
- Peran scope di penggunaan analisis kasus adalah untuk membatasi aksi. Dalam hal ini setelah kondisi terpenuhi maka aksi akan dilakukan.
- **Jika seandainya kita mendefinisikan suatu variabel kedalam lingkup aksi, maka diluar itu variabel tersebut tidak dapat digunakan.**



Notasi Algoritmik untuk Aksi Sekuensial yang memanfaatkan Analisis satu kasus



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 ValidasiByConstant

Program ValidasiByConstant

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

KAMUS

a : integer

constant X : integer = 10

ALGORITMA

input(a)

if a>X then

output(a)

endif

output("selesai cek")

Apa yang terjadi jika input a kurang dari konstanta X?

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")

Apa yang terjadi jika input a kurang dari konstanta X?

Percobaan spesial case

- Seandainya terdapat variabel a yang bertipe bilangan bulat dengan nilai yang sudah terdefinisi.
- Pastikan nilainya lebih dari 0.
- Masukkan variabel tersebut secara langsung sebagai kondisi.
 - Contoh : if a then
- Ekspresi boolean apakah yang akan dihasilkan?

Referensi



Utama :

1. Liem, Inggriani. Diktat Pemrograman Prosedural Informatika ITB. IF-ITB. 2007
2. 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>