



- Dasar Pemrograman – Pertemuan 9

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



Notasi Pengulangan



Notasi pengulangan

- Jenis notasi pengulangan:
 1. Berdasarkan jumlah pengulangan
 2. Berdasarkan kondisi berhenti
 3. **Berdasarkan kondisi pengulangan**
 4. **Berdasarkan dua aksi**
 5. **Berdasarkan pencacah**

3. Berdasarkan kondisi pengulangan

- Aksi akan dilakukan selama *kondisi-pengulangan* masih dipenuhi (bernilai true)
- Tes terhadap *kondisi-pengulangan* dilakukan setiap kali sebelum aksi dilaksanakan
- Pengulangan ini berpotensi untuk menimbulkan aksi “kosong” (tidak pernah melakukan apa-apa) karena pada tes yang pertama, *kondisi-pengulangan* tidak dipenuhi (bernilai false)

Notasi Algoritmik

while (kondisi-pengulangan) **do**
 <ekspresi>

Pengulangan berdasarkan kondisi pengulangan

- Non-Pengulangan

{program chatbot sederhana, tanya “apa mau belok kiri?” (y/t)}

if belokkiri \neq ‘y’ then

 output(“apa mau belok kiri? (y/t”)

if belokkiri \neq ‘y’ then

 output(“apa mau belok kiri? (y/t”)

.....

if belokkiri \neq ‘y’ then

 output(“apa mau belok kiri? (y/t”)

- Pengulangan

{program chatbot sederhana, tanya “apa mau belok kiri?” (y/t)}

output(“apa mau belok kiri?(y/t)”)

input(belokkiri)

while belokkiri \neq ‘y’ do

 output(“apa mau belok kiri?(y/t)”)

 input(belokkiri)

Program ChatbotSederhana

Program ChatbotSederhana

{program chatbot sederhana, tanya “apa mau belok kiri? (y/t)” dengan diberikan variabel belokkiri yang merupakan input user yang bertipe karakter}

KAMUS

belokkiri : char

ALGORITMA

output(“apa mau belok kiri? (y/t)”)

input(belokkiri)

while belokkiri \neq ‘y’ do

 output(“apa mau belok kiri? (y/t)”)

 input(belokkiri)

Program HalloNTimes

Program HalloNTimes
{program mencetak kata “Hallo” sebanyak n kali}

KAMUS

n : int

ALGORITMA

input(n)

while n \geq 0 and n<10 do

n \leftarrow n+1

output(“Hallo”)

4. Berdasarkan dua ekspresi

- Pengulangan ini seolah-olah “gabungan” antara bentuk pengulangan kedua dan ketiga
- Mekanisme yang dilakukan oleh pengulangan ini adalah dengan melakukan secara otomatis aksi-1 pada eksekusi yang pertama kemudian dilakukan tes terhadap kondisi berhenti
- Tergantung kepada kondisi berhenti yang dites :
 - Aksi-2 akan diaktifkan dan kemudian aksi-1 yang berikutnya diulang, atau
 - Pengulangan dihentikan karena efek neto dari aksi-1 menghasilkan kondisi berhenti
- Pengulangan ini berguna untuk kasus-kasus di mana Aksi-2 merupakan hal yang harus dilakukan tergantung dari hasil Aksi-1.

Notasi Algoritmik

iterate

<aksi-1>

stop (kondisi berhenti)

<aksi-2>

Program HalloNTimes

Program HalloNTimes
{program mencetak kata “Hallo” sebanyak n kali}

KAMUS

n, i : int

ALGORITMA

input(n)

i \leftarrow 0

iterate

i \leftarrow i+1

output(“Hallo”)

stop i \geq n:

print("stop")

break

5. Berdasarkan pencacah

- Aksi akan dilakukan dengan memperhitungkan nilai dari nama-pencacah yang di-"jelajahi"
- Dengan memakai pengulangan ini, pemrogram tidak perlu melakukan operasi terhadap suksesor/predesesor karena setiap kali selesai melakukan *Aksi*, otomatis mesin akan melakukan operasi untuk mendapatkan suksesor dari harga yang berlaku saat itu untuk nama
- *range* bisa dari kecil ke besar atau sebaliknya.
- setelah pelaksanaan pengulangan selesai, harga yang tersimpan pada nama-pencacah tidak terdefinisi: jika hendak dipakai, harus didefinisikan kembali

Notasi Algoritmik	Non- Pengulangan	Notasi
Nama-pencacah <u>traversal</u> [range] <ekspresi>	output("makan bang") output("makan bang") output("makan bang") output("makan bang") output("makan bang")	i <u>traversal</u> [1..5] output("makan bang")

Program CetakMahasiswa

Program CetakMahasiswa

{Program untuk mencetak mahasiswa berkali-kali dengan batasan n input oleh user}

KAMUS

n, i : integer

ALGORITMA

input(n)

i traversal [1 ... n]

 output("mahasiswa")

output("finish")

Contoh Persoalan Komputasional

Program CetakBilanganPositifRange

{Tulis semua bilangan positif lebih dari 0 mulai dari x sampai y. Perhatikan bahwa x dan y merupakan bilangan bulat positif yang di inputkan oleh user.}

KAMUS

x, y , i : integer {i merupakan bilangan yang akan di outputkan nantinya didalam pengulangan}

ALGORITMA

```
input(x,y)
i traversal [x ... y]
  output(i)
output("finish")
```

Contoh Persoalan Komputasional

Program CetakBilanganGenapPositifRange

{Tulis semua bilangan genap positif lebih dari 0 mulai dari 0 sampai n. Perhatikan bahwa n merupakan bilangan bulat positif yang di inputkan oleh user.}

KAMUS

n, i : integer {i merupakan bilangan yang akan di outputkan nantinya didalam pengulangan}

ALGORITMA

input(n)

i traversal [1 ... n]

i \leftarrow i+1 {variabel i akan digunakan sebagai jeda}

output(i)

output("finish")

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>