



Pemrograman Prosedural

PENDAHULUAN

Tim Pengajar KU1071
Sem. 1 2008-2009

Tujuan Kuliah Pemrograman Prosedural



- Mahasiswa mampu:
 - Memecahkan masalah dengan paradigma prosedural dan menuliskan spesifikasi dan algoritmanya tanpa tergantung bahasa pemrograman
 - Menulis algoritma dengan metodologi dan skema standard yang diajarkan
 - Menuliskan program yang “baik”



Program Prosedural (1)

- Program dalam bahasa Pascal termasuk dalam program prosedural : Algoritma + Struktur Data
- Pemrograman prosedural (imperatif) :
 - Dihasilkan berdasarkan dekomposisi “aksional”, menjadi Aksi yang akan dijalankan secara berurutan (sekuensial).



Program Prosedural (2)

- Pemrograman prosedural (imperatif) :
 - Aksi :

Jelas Initial state, Final state dan harus dalam waktu terbatas

 - Dapat didekomposisi menjadi Sub Aksi
 - Aksi diterjemahkan menjadi sederetan instruksi (aksi primitif) yang dapat dijalankan oleh mesin
- Ilustrasi: MENGUPAS KENTANG (diktat hal. 17-25)



Aksi Mengupas Kentang (1)

Initial State (I.S.): T_0 , kentang di kantong, ada di rak dapur

Final State (F.S.): T_1 , kentang terkupas di panci, siap dimasak, kantong kembali ke rak dapur

Sub-aksi:

1. Ambil kantong kentang dari rak
2. Ambil panci dari lemari
3. Kupas kentang
4. Kembalikan kantong kentang ke rak

Aksi 1 dan 2 bebas urutan, bisa digabungkan

Aksi Mengupas Kentang (2)



1. Ambil kantong kentang dari rak
 2. Ambil panci dari lemari
 3. Kupas kentang
 4. Kembalikan kantong kentang ke rak
- Harus dikerjakan dengan urutan yang diinginkan → aksi sekuensial
 - Saat tertentu (jika baju warna muda) perlu celemek → aksi kondisional/ analisis kasus
 - Mengupas kentang dilakukan hingga pada kriteria tertentu (jumlah) → pengulangan
 - Mengupas kentang bagian dari aksi menyiapkan makan malam → subprogram

Notasi Algoritmik

- Teks Algoritma
 - Judul (Header)
 - Kamus
 - Algoritma
- Harus didefinisikan nama sebagai identifikasi
- Nama: sesuatu yang dipakai untuk identifikasi type, tempat penyimpanan (variable), konstanta, prosedur, fungsi, atau modul program, algoritma → harus unik

Notasi Algoritmik

- Template Program

<u>JUDUL</u>
{Spesifikasi teks algoritmik secara umum}
KAMUS
{Definisi konstanta, type, deklarasi variabel, spesifikasi prosedur, fungsi}
ALGORITMA
{Teks algoritma - tidak berada di antara tanda kurung kurawal}



Kamus

- Kamus dipakai untuk deklarasi
- To „declare“ = to make known formally, officially, or explicitly
- Deklarasi nama yang didefinisikan pemrogram : “type”, variabel, konstanta
- Deklarasi nama-nama lain (biasanya tidak harus): nama fungsi, prosedur dan spesifikasinya
- Deklarasi BUKAN instruksi
- Contoh deklarasi:

```
i : integer  
ff : real  
S : string  
P : Point
```



Algoritma

- Adalah bagian program dalam bentuk teks algoritmik yang berisi instruksi atau pemanggilan aksi
- Teks algoritmik tsb. dapat berupa:
 - Instruksi dasar: I/O, assignment
 - *Sequential statement* Analisis
 - kasus pengulangan
 -



Type

- Definisi Type (pola struktur informasi):
 - nama type
 - domain nilai, batasan nilai yang dapat disimpan dalam type tsb. (kelak, ketika menjadi variabel)
 - perjanjian cara menuliskan konstanta (literal) type tsb
 - operator
- Type:
 - primitif
 - bukan primitif, buatan pemrogram, harus dideklarasasi

Type

- Type dasar, primitif disediakan oleh bahasa
- Type enumerasi
- Type koleksi, misal: array, set, tuple, dictionary, String (akan dijelaskan kemudian)
- Type bentukan: nama type dibuat dan didefinisikan oleh pemrogram.



Type Dasar

- Disediakan oleh bahasa
- Disesuaikan oleh wijanarto
 - Boolean
 - Integer
 - Real
 - String
 - None



Type Bentukan

- Type yang dibentuk (dan diberi nama) dari beberapa komponen ber-type tertentu
- Disebut juga type komposisi, agregat
- Mengapa perlu type bentukan?
 - Perancang memutuskan bahwa keseluruhan (hasil komposisi) komponen type mempunyai makna **semantik**
- Operasi terhadap type bentukan:
 - Terhadap komponen bertipe dasar: seperti tipe dasar
 - Terhadap keseluruhan: bisa didefinisikan, bisa tidak



Type Bentukan(class)

```
type namatype: < elemen1 : type1,  
                elemen2 : type2,  
                ... >
```

- Membentuk type berarti menentukan struktur data
- Pembentuk keseluruhan: Konstruktor
- Pengacu elemen: Selektor

Contoh: Type Point

- Type Point

```
type Point: < X : real,  
                Y : real  
                >
```

- Deklarasi nama variable P

```
P: Point
```

- Cara acu/akses: P.X, P.Y

- Domain: <real, real>

- Konstanta: <1.0, 1.0>, <10.0, 5.0>

- Operator:

- Operasi terhadap Point harus dibuat
- operasi real terhadap P.X, P.Y



Kapan harus membuat type?

- Untuk mempermudah berpikir dalam abstraksi lebih tinggi (tidak pada tingkatan primitif).
- Kalau data yang diolah dapat diabstraksikan menjadi suatu data yang merupakan kesatuan, misalnya: Point, Jam, Date, Orang, Pegawai, SIM, Kartu, Window, Figure, Square,
Dengan memberi nama Type, pengolahan jadi lebih jelas.
- Biasanya, nama type mewakili nama „benda” dengan ciri tertentu.

Variabel

- Variabel menyimpan nilai ber-"type" sesuai dengan deklarasi, yang dapat dioperasikan sesuai operator
- Variabel :
 - deklarasi (supaya nama dikenal),
 - Alokasi/instansiasi(supaya punya memori), inisialisasi
 - nilai (siap dimanipulasi)
- Contoh
 - Deklarasi (dan alokasi): $i : \underline{integer}$
 - Inisialisasi: $i \leftarrow 0$
Disebut *assignment* = akan dijelaskan kemudian

Variabel: Scope & Life time



- Scope : lingkup
 - lingkup dimana nama variabel berlaku/dikenal.
- Lifetime : masa hidup
 - kapan terdefinisi memori & nilainya
 - kapan dihancurkan/tidak ada lagi
- Dalam mendeklarasi variabel, harus memahami betul mengenai scope dan life time, serta pemusnahannya.



Membuat Variabel yang “baik”

- Nama sesuai dengan artinya, tidak membingungkan.
- Memakai nama yang tidak berarti sesuai „konvensi“, misalnya:
 - i,j,k untuk indeks array, matriks
- Dirancang sesuai scope & life time yang diperlukan.



Konstanta

- Nilai, literal yang diberi nama
- Berbeda dengan Variable, tidak boleh diubah nilainya.
- Menambah robustness, *readability* (bandingkan dengan program yang mengandung literal di mana-mana)
- Contoh:

```
constant PI : real = 3.14159
```

Memakai konstanta

- Memang untuk nilai yang *constant*
- Nama sesuai dengan nilainya, tidak membingungkan. Misalnya

constant Satu: integer = 1

bukan

constant **Satu**: integer = **7**

- Sebagai „parameter“ program.



Perintah paling dasar

- Pemberian nilai (assignment) sesuai dengan type.
- Perbandingan (kesamaan, ketidak-samaan)
- Operasi relasional lain (lebih besar, lebih kecil,....)
- Operasi aritmetika (khusus untuk nilai numerik)



Nilai, Input+Output

- Nilai atau harga: suatu besaran bertipe yang telah dikenal
- Pengisian nilai:
 - Assignment
 - Dibaca dari piranti masukan
- Lihat contoh input, output notasi untuk assignment, (cheat-sheet)

Ekspresi

- Ekspresi:
 - “rumus perhitungan”
 - Terdiri atas operator dan operan
- Notasi yang dipakai: infix
 - Contoh: $27 * 5 - 2$
- Jenis Ekspresi: logika/boolean, numerik, karakter, dan string



Ekspresi

- Ekspresi :
 - ketat terhadap type
 - “loose” terhadap type
- Usahakan menulis ekspresi ketat type

Assignment (=)

- Ruas kiri = Ruas Kanan
- Ruas **kiri** harus **variable**
- Ruas **kanan** harus **<ekspresi>**
- Ekspresi :
 - operan (nama variabel, konstanta, aplikasi fungsi) dan operator harus kompatibel, ketat type.
 - ekspresi bukan hanya ekspresi aritmetika, ada ekspresi boolean, ekspresi relasional



Aksi Sekuensial

- Adalah struktur kontrol algoritmik paling sederhana
- Dilaksanakan oleh komputer berdasarkan urutan penulisannya
- Initial State vs Final State. Final State sebuah aksi ke- i menjadi Initial State bagi aksi selanjutnya, aksi ke $i+1$

Aksi Sekuensial

- Adalah urutan perintah/statement dalam bahasa yang **logic**.
 - **Logic** artinya keterurutan langkah **tidak tumpang tindih** dan sesuai **aturan** untuk menghasilkan **keluaran** yang **diinginkan**
- Merupakan struktur paling dasar dalam algoritma

Contoh

- Dalam dunia nyata-menggoreng telur ceplok
 - Siapkan kompor, alat penggoreng, telur, minyak dan garam
 - Taruh penggoreng diatas kompor dan nyalakan apinya
 - Tuang minyak secukupnya, tunggu hingga panas merata
 - Pecah telur dan masukan ke penggoreng, lalu taburi garam secukupnya
 - Tunggu hingga telur padat dengan pinggiran golden brown
 - Matikan kompor dan angkat telur dengan spatula
 - Menggoreng selesai

Contoh

- Dalam pemrograman – FPB dari dua bilangan

Notasi Algoritmik	Python
<pre>input(a,b) i ← 1 fpb ← 0 while(i ≤ a and i ≤ b) do if a mod i == 0 and b mod i == 0 then fpb = i i ← i + 1 output(fpb)</pre>	<pre>a = float(input("Masukan nilai a: ")) b = float(input("Masukan nilai b: ")) i = 1 fpb = 0 while(i <= a and i <= b): if(a % i == 0 and b % i == 0): fpb = i i = i + 1 print("FPB {0},{1}={2}".format(a, b, fpb))</pre>

Latihan Soal

- Buatlah program kecil dalam notasi algoritmik untuk persoalan-persoalan berikut:
- **Menghitung volume gas ideal** (V) dalam liter dengan masukan tekanan (P) dalam *kiloPascal*, banyaknya mol gas (n) dalam *mol*, dan temperatur (T) dalam *derajat Kelvin*, serta diketahui konstanta gas ideal (R) yaitu $8.314 \text{ JK}^{-1}\text{mol}^{-1}$ dengan rumus: **$P V = n R T$**
- **Menghitung tegangan listrik** (V) dalam volt dari masukan besarnya arus listrik (I) dalam ampere dan hambatan listrik (R) dalam ohm, dengan hukum Ohm: **$V = I R$**
- **Menghitung luas sebuah trapesium** (L) berdasarkan masukan a dan b yang merupakan panjang dua sisi sejajar trapesium dan h yang merupakan tinggi trapesium, rumus: **$L = \frac{1}{2} * h * (a+b)$**