

Pengantar Algoritma dan Pemrograman

Alex Budiyanto

alex@ilmukomputer.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Bab 2 Aturan Penulisan Algoritma

Algoritma berisi langkah-langkah penyelesaian masalah. Langkah-langkah tersebut dapat ditulis dalam notasi apapun, asalkan mudah dibaca dan dimengerti, karena memang tidak ada notasi baku dalam penulisan algoritma. Tiap orang dapat membuat aturan penulisan dan notasi algoritma sendiri. Agar notasi algoritma mudah ditranslasi ke dalam notasi bahasa pemrograman, maka sebaiknya notasi algoritma tersebut berkoresponden dengan notasi bahasa pemrograman secara umum.

Aturan Penulisan Algoritma

Setiap Algoritma akan selalu terdiri dari tiga bagian yaitu :

- Judul (Header)
- Kamus
- Algoritma

Pada setiap bagian tersebut apabila akan dituliskan komentar mengenai setiap bagian tersebut dituliskan diantara tanda kurung kurawa contoh { Komentar }. Notasi algoritmis yang dituliskan diantara tanda ini tidak akan dieksekusi oleh program.

Contoh :

<u>Judul</u> { Komentar mengenai Algoritma seperti cara kerja program, Kondisi awal dan kondisi akhir dari algoritma }
<u>Kamus</u> { Pada bagian ini, didefinisikan nama konstanta, nama variable, nama prosedur dan nama fungsi }
<u>Algoritma</u> { Pada bagian ini algoritma dituliskan. Semua teks yang dituliskan tidak diantara tanda kurung kurawa akan dianggap sebagai notasi algoritma yang akan berpengaruh terhadap kebenaran algoritma }

Judul (Header)

Judul adalah bagian teks algoritma yang digunakan sebagai tempat mendefinisikan nama dengan menentukan apakah teks tersebut adalah program, prosedur, fungsi. Setelah judul disarankan untuk menuliskan spesifikasi singkat dari teks algoritma tersebut. Nama algoritma sebaiknya singkat namun cukup menggambarkan apa yang akan dilakukan oleh algoritma tersebut.

Contoh :

Program Luas_Kubus ← {Judul Algoritma} { Menghitung luas kubus untuk ukuran sisi yang dibaca dari piranti masukan lalu mencetak hasilnya kepiranti keluaran } ← {Spesifikasi Algoritma}

Catatan :

Untuk memisahkan antara kata dalam judul algoritma menggunakan tanda “_” bukanlah suatu keharusan. Anda dapat menuliskan LuasLingkaran atau Luas_Lingkaran. Tetapi sebaiknya anda tidak menggunakan spasi “ ” untuk memisahkan antara kata di dalam nama algoritma.

Kamus (Deklarasi)

Kamus adalah bagian teks algoritma sebagai tempat untuk mendefinisikan :

- Nama type
- Nama konstanta
- Nama variabel
- Nama fungsi
- Nama prosedur

Semua nama tersebut baru dapat dipakai di dalam algoritma jika telah didefinisikan terlebih dahulu didalam kamus. Penulisan sekumpulan nama dalam kamus sebaiknya dikelompokkan menurut jenis nama tersebut.

Nama variabel belum terdefinisi nilainya ketika didefinisikan. Pendefinisian nama konstanta sekaligus memberikan harga konstanta tersebut, pendefinisian nama fungsi dilakukan sekaligus dengan domain / range serta spesifikasinya. Pendefinisian nama prosedur sekaligus dengan pendefinisian parameter (jika ada) dan spesifikasi prosedur (kondisi awal “Initial State”, Kondisi akhir “ Final State” dan proses yang dilakukan).

Contoh :

Kamus

{Nama type, hanya untuk type yang bukan type dasar}

type jam : <hh,mm,ss :**integer**> {Type jam terdiri dari 3 masukan yaitu “hh” sebagai jam. “mm” sebagai menit dan “ss” sebagai detik}

{Nama konstanta, harus menyebutkan type dan nilai }

constant phi : **real** = 3,14159

constant nama : **string** = ‘Alex’

constant benar : **boolean** = **true**

{Nama Informasi, menyebutkan type}

x,y : **integer** {suatu nilai yang bertype bilangan bulat}

NMax : **real** {nilai maksimum yang bertype bilangan real}

Nama : **string** {suatu nilai yang merupakan kumpulan character}

P : **point** {suatu nilai pada bidang kartesian}

Cari : **Boolean** {suatu nilai logika}

{Nama fungsi, menyebutkan domain dan range}

function RealToInt (x :real) → integer

{mengubah harga x yang bertipe real menjadi harga ekuivalen yang bertipe integer}

{Nama prosedur, menyebutkan “IS” initial state, “FS” final state dan proses}

procedure tukar (input/output x,y : real)

{ IS x dan y terdefinisi, $x = a$ dan $y = b$

FS $x = b$ dan $y = a$

Proses : menukar isi informasi bilangan x dan y }

Algoritma (Deskripsi)

Algoritma adalah bagian inti dari suatu algoritma yang berisi instruksi atau pemanggilan aksi yang telah didefinisikan. Komponen teks algoritma dalam pemrograman procedural dapat berupa :

- Instruksi dasar seperti input/output, assignment
- Sequence (runtutan)
- Analisa kasus
- Perulangan

Setiap langkah algoritma dibaca dari “atas” ke “bawah”. Urutan deskripsi penulisan menentukan urutan langkah pelaksanaan perintah.

Contoh :

Algoritma

input (c,d) {menerima masukan 2 bilangan c dan d }

if $c < d$ **then** {operasi kondisional}

$e \leftarrow a + b$ { e di *assignment* oleh nilai a dan b }

else

$e \leftarrow a - b$

output (e) {hasil keluaran berupa bilangan e }

Catatan :

Untuk kata-kata input, output, if then else, output akan dipelajari lebih mendalam dalam bab-bab selanjutnya.

Contoh Penyelesaian Masalah Oleh Algoritma

Mencetak String “Selamat Belajar Algoritma dan Pemrograman” ke piranti Keluaran.

Program Cetak_string

{mencetak string “Selamat Belajar Algoritma dan Pemrograman” ke piranti
keluaran}

Kamus

{tidak ada}

Algoritma

Output (‘Selamat Belajar Algoritma dan Pemrograman’)

Menentukan nilai terbesar dari bilangan bulat yang dibaca dari piranti masukan dan menuliskan hasilnya ke piranti keluaran.

Program Nilai_Maksimal

{Menentukan nilai tertinggi yang dibaca dari piranti masukan dan hasilnya dicetak
ke piranti keluaran}

Kamus

hasil,x,y : **integer** {hasil merupakan variabel untuk menampung nilai keluaran}
{x,y adalah variabel untuk menampung nilai masukan}

Algoritma

input (x,y) {membaca nilai x dan y dari piranti masukan}

if x < y **then** {operasi kondisional}

hasil ← x {hasil di *assignment* oleh nilai terbesar}

else

hasil ← y

output (hasil) (nilai didalam variabel hasil dicetak ke piranti keluaran)}