

BAB 8 SPESIFIKASI PROSES (PROCESS SPECIFICATION)

1. PENDAHULUAN

Pada bab ini, kita akan mencari tahu spesifikasi proses, dalam hal ini deskripsi dari apa yang terjadi di dalam setiap level paling dasar, bulatan konvensional pada diagram alir data. Kegunaan proses spesifikasi cukup penting untuk ke depannya, hal ini mendefinisikan apa yang harus dikerjakan untuk merubah input menjadi output. Hal tersebut merupakan gambaran detail kebijakan bisnis user yang dibawa oleh setiap lingkaran.

Spesifikasi Proses menggambarkan kejadian di dalam setiap *bubble* pada level terbawah pada data flow diagram. Spesifikasi proses mendefinisikan kegiatan yang harus dilakukan untuk mengubah input menjadi output (*Edward Yourdon, Modern Structured Analysis, hal. 203*).

Spesifikasi proses digunakan untuk mendeskripsikan proses yang terjadi pada level yang paling dasar dalam DFD. Model ini berfungsi mendeskripsikan apa yang dilakukan ketika masukan ditransformasi menjadi keluaran.

Ada berbagai macam tools yang dapat kita gunakan untuk menghasilkan suatu spesifikasi proses: tabel keputusan, Bahasa Inggris terstruktur, pre/post condition, flowcharts, diagram Nassi-shneiderman, dan lain sebagainya. Sedangkan kebanyakan analisis sistem mengarah ke Bahasa Inggris terstruktur, anda harus ingat bahwa setiap metode dapat digunakan, selama hal tersebut memuaskan dua keadaan penting:

1. Spesifikasi proses harus ditampilkan dalam suatu bentuk/form yang dapat diverifikasi oleh user dan sistem analis. Kondisi ini tepat untuk alasan ini, dimana kita mengurangi paparan Bahasa Inggris sebagai sebuah alat spesifikasi: hal ini dikenal sebagai ambigu, khususnya apabila menggambarkan tindakan (keputusan) alternatif dan tindakan berulang (loops). Secara alami, hal tersebut juga cenderung menyebabkan kebingungan yang sangat ketika mengekspresikan bagian-bagian kondisi Boolean (misalkan kombinasi dari operator Boolean AND, OR dan NOT).
2. Spesifikasi proses harus ditampilkan dalam suatu bentuk/form yang dapat mengkomunikasikan secara efektif berbagai keterlibatan berbagai latar belakang pendengar. Sedangkan hal tersebut akan menjadi tipe dari analisis sistem yang menuliskan spesifikasi proses, hal tersebut biasanya menjadi bermacam-macam pendengar dari para pengguna, manager, auditor, personil quality assurance, dan lainnya yang membaca spesifikasi proses. Suatu proses spesifikasi diharapkan dapat ditampilkan dalam perhitungan yang dapat diprediksi, atau dalam pascal, atau dalam pendekatan format diagram seperti software use-it; tetapi jika komunitas user menolak untuk melihat pada beberapa spesifikasi, mereka adalah tidak berharga.

Spesifikasi Proses berfungsi mendeskripsikan tahapan yang dilakukan untuk mentransformasikan input menjadi output. Bentuk penyajian spesifikasi proses adalah sebagai berikut :

No Proses : menyatakan nomor proses	
Nama Proses : menyatakan nama proses	
Deskripsi : penjelasan tujuan proses	
Source	Data
(menyatakan sumber data input menuju proses)	(menyatakan isi data yang masuk ke proses)
Destination	Data
(menyatakan tujuan data output dari proses)	(menyatakan isi data yang keluar dari proses)
Logika proses (menyatakan algoritma dari proses)	

Structured english (SE) merupakan alat yang cukup efisien untuk menggambarkan suatu algoritma. Pseudocode merupakan alternatif dari structured English. SE mirip dengan pseudocode, karena kemiripan ini, maka SE dan pseudocode sering dianggap sama.

Pseudo berarti imitasi/mirip atau menyerupai dan code menunjukkan kode dari program, berarti pseudocode adalah kode yang mirip dengan instruksi kode program yang sebenarnya. Pseudocode akan memudahkan programmer untuk memahami dan menggunakannya, karena mirip dengan kode-kode program sebenarnya. Pseudocode lebih terinci dibandingkan dengan SE, seperti menjelaskan juga tentang pemberian nilai awal dari suatu variabel, membuka dan menutup file, subscript atau tipe-tipe data yang digunakan (misalnya real, integer, Boolean) yang tidak disebutkan di SE.

Tidak ada satu alatpun yang sempurna untuk semua situasi. Demikian juga dengan SE dan pseudocode yang kurang baik untuk menggambarkan algoritma yang banyak mengandung penyeleksian kondisi atau keputusan (*decision*). Bagan alir (*flowchart*), tabel keputusan (*decision table*) dan pohon keputusan (*decision tree*) merupakan alat yang lebih baik untuk menggambarkan keputusan yang kompleks. Akan tetapi dalam beberapa proses yang harus didokumentasikan sebaliknya haruslah tidak terdiri dari proses-proses yang kompleks, cukup terdiri dari operasi-operasi keputusan dan perulangan yang sederhana. Hal ini tidaklah mengherankan, karena proses secara logika yang digambarkan secara terstruktur haruslah mudah dipahami dan dibentuk dengan bentuk struktur yang macamnya sedikit. Sedikitnya struktur ini menyediakan basis untuk pemrograman terstruktur, yang akan memberikan efektivitas dalam hal kesederhanaan dan standarisasi.

Kalimat dalam spesifikasi proses umumnya tersusun dari sejumlah komposisi seperti rumus matematis, kata kerja dan obyek (variable atau elemen data). Terminology dalam komputasi dideskripsikan dengan kata kerja sebagai berikut:

Cari (find, search atau locate),
Jumlahkan (add),
Kalikan (multiply),
Kurangi (subtract),
Bagi (divide),
Ambil (get, read atau accept),
Tulis (display atau write),
Hitung (compute),
Hapus (delete),
Cek (validate),
Pindahkan (move),
Gantikan (replace),
Set (set),
Urutkan (sort),
Buka (open),
Gandakan (copy), dan lain-lain.

Operator logic seperti:

Dan (and),
Atau (or), dan lain sebagainya

Obyek, dalam hal ini mendefinisikan elemen data yang telah diuraikan dalam kamus data atau variabel, yang hanya berlaku dalam proses tersebut (local term). Variabel tersebut dikenali, relevan dan mempunyai arti hanya dalam spesifikasi proses tertentu misalnya variabel total_harian dalam proses menghitung pesanan;

Total_harian = 0 *inisialisasi variabel*

buka pemesanan *membuka penyimpanan pemesanan*

do while (masih ada order) dan (tgl_pesanan=tgl_sekarang)

 ambil order berikutnya dan pemesanan dengan

 tgl_pesanan=tgl_sekarang

 buka keuangan *membuka penyimpanan keuangan*

 tulis ke keuangan nomor_pesanan. Nama_pelanggan, nilai_transaksi

 total_harian=total_harian+nilai_transaksi

enddo

Penggunaan do while dan enddo menunjukkan dilakukannya kombinasi antara spesifikasi proses dan konstruksi pemrograman terstruktur.

2. KONSTRUKSI-KONSTRUKSI YANG DIGUNAKAN

a. If-Then-Else

Digunakan untuk mendeskripsikan pilihan secara biner (pada satu saat hanya ada dua pilihan), bentuknya:

If kondisi	Atau,	If kondisi
Tindakan		Tindakan-1
Endif		Else
		Tindakan-2
		Endif

Contoh 1:

```
If pelanggan tinggal di Bandung
    Buka prospek_pemasaran
    *membuka penyimpanan pemasaran*
    Tulis pelanggan ke prospek_pemasaran
Endif
```

Contoh 2:

```
If umur_pelanggan > 65
    Set tarif_pemesanan ke pelanggan_senior
Else
    Set tarif_pemesanan ke tariff_normal
Endif
```

b. Do Case...Case...EndCase

Digunakan untuk mendeskripsikan pilihan pada satu saat secara jamak (multivalued decision).

Bentuknya:

```
Do case
case variabel=nilai(1)
    tindakan-(1)
case variabel=nilai(2)
    tindakan-(2)
```

...
case variabel=nilai(n)
 tindakan-(n)
otherwise
 kalimat(n+1)
Endcase

Contoh:

Do case

Case umur_pelanggan < 13

 Set tarif_pemesanan ke tarif_anak

Case umur_pelanggan > 12 dan umur_pelanggan < 20

 Set tarif_pemesanan ke tarif_remaja

Case umur_pelanggan > 19 dan umur_pelanggan < 65

 Set tarif_pemesanan ke tarif_dewasa

Otherwise

 Set tarif_pemesanan ke tarif_senior

Endcase

Catatan: otherwise digunakan untuk spesifikasi alternative yang tidak memenuhi semua kemungkinan yang sudah disediakan sebelumnya atau sudah tidak perlu didefinisikan lagi karena konsekuensi bagi pilihan (otherwise) tersebut sama.

c. For

Digunakan untuk mendefinisikan pengulangan instruksi sampai kondisi tertentu terpenuhi.

Bentuknya:

For each item

 Tindakan

Contoh:

Read jumlah_barang

For masing-masing barang

Read data nilai penjualan (unit dan harga)

Compute nilai_barang yaitu unit dikalikan harga

Display dalam satu baris hasilnya

Compute jumlah

Display nilai jumlah

d. Do While Condition...EndDo

Digunakan untuk mendefinisikan pengulangan instruksi selama memenuhi kondisi tertentu.

Bentuknya:

Do while kondisi_1

tindakan_1

Enddo

Contoh:

Do while (masih ada item_pemesanan)

Nilai_transaksi = harga_item x jumlah_item

Enddo

e. Repeat_Until Condition

Digunakan untuk mendefinisikan pengulangan instruksi selama memenuhi kondisi tertentu.

Bentuknya:

Repeat

tindakan_1

Until kondisi_1

Contoh:

Repeat

Read record dari file

Until akhir dari file

Sebagai salah satu model penting dalam perancangan pada dasarnya bentuk spesifikasi proses relative bebas, artinya dapat digunakan bentuk apapun asal mempermudah programmer dalam memahami rancangan. Justru yang paling penting adalah memilih bentuk yang tepat untuk spesifikasi proses.

3. PERANGKAT SPESIFIKASI PROSES LAINNYA

a. Model Pre/Post Conditions

Contoh:

Precondition1

Pelanggan mencocokkan nomor_rekening dengan nomor_rekening pada penyimpanan keuangan dan dilakukan jika kode_status sah.

Postcondition1

Dikeluarkan tanda_pembayaran dengan nomor_rekening dan nilai_penjualan

b. Model Tabel Keputusan

Model dari tabel keputusan terdiri dari empat bagian utama, yaitu condition stub, condition entry, action stub dan action entry. Seperti gambar di bawah ini

Gambar 1. Struktur dari table keputusan

	Rules				
	1	2	3	4	...N
Condition stub	Condition entry				
Section stub	Action entry				

Condition stub berisi kondisi-kondisi yang akan diseleksi. Condition entry berisi kemungkinan-kemungkinan dari kondisi yang diseleksi, yaitu terpenuhi (diberi symbol ‘Y’) dan tidak terpenuhi (diberi symbol ‘T’). setiap kondisi yang diseleksi akan mempunyai dua kemungkinan kejadian, yaitu terpenuhi dan tidak terpenuhi. Bila ada x kondisi yang diseleksi, maka akan terdapat N kemungkinan kejadian yaitu sebesar $N=2^x$. action stub berisi pernyataan-pernyataan yang akan dikerjakan baik kondisi yang diseleksi terpenuhi maupun tidak terpenuhi. Action entry digunakan untuk memberi tanda tindakan mana yang akan dilakukan dan mana yang tidak akan dilakukan.

Contoh:

Jika unit yang dipesan untuk suatu barang tertentu sama atau melebihi jumlah minimum yang mendapatkan potongan dan pemesannya adalah dealer, maka akan mendapatkan potongan. Sebaliknya bila unit yang dipesan kurang dari unit minimum yang mendapatkan potongan, walaupun yang membeli dealer, maka tidak akan mendapatkan potongan. Bila unit barang persediaan di gudang tidak mencukupi maka yang dikirim adalah unit yang ada dan kemudian dibuat catatan mengenai kekurangannya (back-order). Kalau unit persediaan mencukupi, maka semua pesanan dikirim.

Langkah-langkah membuat table keputusan:

1. Menentukan kondisi yang akan diseleksi

Dari permasalahan tersebut, terdapat 3 buah kondisi yang akan diseleksi, yaitu:

 - a. Apakah unit dipesan \geq unit minimum potongan?
 - b. Apakah pemesannya dealer?
 - c. Apakah unit persediaan mencukupi?
2. Menentukan jumlah kemungkinan kejadian yang akan terjadi, yaitu sebanyak $N = 2^3 = 8$
3. Menentukan tindakan yang akan dilakukan

Dari permasalahan tersebut terdapat 5 buah tindakan yang akan dilakukan, yaitu:

- a. Mendapatkan potongan
 - b. Tidak mendapatkan potongan
 - c. Kirim semua yang dipesan
 - d. Kirim yang ada saja
 - e. Buat catatan kekurangannya
4. Mengisi condition entry
 5. Mengisi action entry

dari langkah-langkah tersebut maka dihasilkan tabel keputusan adalah sebagai berikut:

	1	2	3	4	5	6	7	8
a. apakah unit dipesan \geq unit minimum potongan?	Y	Y	Y	Y	T	T	T	T
b. apakah pemesannya dealer?	Y	Y	T	T	Y	Y	T	T
c. apakah unit persediaan mencukupi?	Y	T	Y	T	Y	T	Y	T
a. Mendapatkan potongan	X	X	-	-	-	-	-	-
b. Tidak mendapatkan potongan	-	-	X	X	X	X	X	X
c. Kirim semua yang dipesan	X	-	X	-	X	-	X	-
d. Kirim yang ada saja	-	X	-	X	-	X	-	X
e. Buat catatan kekurangannya	-	X	-	X	-	X	-	X

Selain itu juga ada model-model lainnya diantaranya adalah model grafik dan model flowchart.