

PENGERTIAN DASAR Dalam Pemrograman Prosedural

Pada bagian ini akan dijelaskan definisi beberapa pengertian dasar yang penting sehubungan dengan algoritma dan pemrograman, yang akan diberikan dalam contoh pada kehidupan sehari-hari. Mungkin pengertian-pengertian tersebut mula-mula terasa abstrak bagi beberapa pembaca, tapi baiklah coba dipahami.

Pengertian pertama yang akan dijelaskan adalah **aksi**. Suatu aksi adalah kejadian yang terjadi pada suatu **selang waktu terbatas** dan menghasilkan **efek neto** yang telah terdefinisi dengan baik dan memang **direncanakan**. Pada deskripsi tersebut, ditekankan benar efek tersebut harus “direncanakan”, maka berarti dititikberatkan pada kegunaannya. Jika seseorang tertarik pada suatu aksi, maka jelas bahwa minatnya adalah pada efek netonya.

Suatu aksi harus terjadi pada selang waktu yang terbatas, dimulai pada saat T_0 dan berakhir pada saat T_1 . Maka efek neto dari aksi dapat dijelaskan dengan membandingkan keadaan pada saat T_0 dan keadaan pada saat T_1 .

Contoh dari suatu aksi adalah Ibu Tati yang MENGUPAS KENTANG untuk mempersiapkan makan malam. Pernyataan ini mencakup hal yang luas ruang lingkupnya, misalnya :

- Apakah kentangnya harus dibeli dulu atau sudah ada di dapur ?
- Apakah yang dimaksud dengan mengupas kentang untuk makan malam berarti sampai dengan kentang terhidang ?
- Ketika kentangnya terhidang, jadi sup, digoreng atau direbus saja ?

Maka kita harus membatasi dengan jelas keadaan awal yang menjadi titik tolak mengupas kentang dan keadaan akhir yang ingin dicapai supaya dapat “merencanakan” efek neto yang diinginkan.

Untuk itu ditentukan :

- *Initial state* (I.S. keadaan awal), T_0 , adalah kentang sudah ada di kantong kentang, yang ditaruh di rak di dapur, di mana ibu Tati akan mengupasnya
- *Final state* (F.S. keadaan akhir), T_1 , kentang dalam keadaan terkupas di panci, siap untuk dimasak dan kantong kentangnya harus dikembalikan ke rak lagi.

Pengandaian yang lain adalah bahwa persediaan kentang di ibu selalu cukup untuk makan malam. Penambahan kentang ke dapur di luar tinjauan masalah ini. Ini adalah contoh bagaimana kita menentukan **batasan** dari persoalan yang akan diprogram.

Suatu kejadian dapat dipandang sebagai urutan dari beberapa kejadian, berarti dapat diuraikan dalam beberapa (sub) aksi yang terjadi secara sekuensial. Dengan sudut pandang ini maka efek kumulatifnya sama dengan efek neto dari seluruh kejadian. Dikatakan bahwa kejadian tersebut dianggap sebagai sequential process atau disingkat proses.

Penggolongan suatu kejadian sebagai proses atau aksi adalah relatif. Kejadian yang sama dapat dianggap sebagai aksi ataupun sebagai proses. Kalau lebih dititik beratkan efek netonya, yaitu keadaan “ sebelum dan sesudah” maka kejadian tersebut dianggap sebagai berarti kejadian tersebut dianggap sebagai proses. Dengan anggapan kejadian tersebut suatu proses, maka T_0 adalah awal dari sebuah sub-aksi dan setiap akhir dari suatu sub-aksi akan merupakan awal dari sub-aksi berikutnya, dengan suatu keistimewaan, akhir dari sub-aksi yang terakhir adalah T_1 yaitu akhir dari seluruh kejadian.

Penggolongan suatu kejadian menjadi proses atau aksi tidak ada hubungannya dengan sifat dari kejadian itu sendiri melainkan tergantung dari cara peninjauan. Jika cara peninjauan dilakukan dari sudut pandang yang berbeda, maka biasanya hasil antara yang ingin diperhatikan juga berbeda. Misalkan kejadian tentang ibu Tati mengupas kentang, dapat dijelaskan oleh urutan sub-aksi yang dilakukan oleh ibu tersebut, yaitu :

- *Ambil kantong kentang dari rak*
- *Ambil panci dari almari*
- *Kupas kentang*
- *Kembalikan kantong kentang dari rak*

Pada contoh tersebut, kejadian dijelaskan sebagai urutan dari empat sub-aksi yang diungkapkan berdasarkan suatu pengamatan. Jika dari hasil pengamatan **tidak dipandang perlu** untuk menjelaskan bahwa kantong kentang diambil dari rak sebelum panci diambil dari almari, maka cukup dituliskan :

- *Ambil kantong kentang dari rak dan panci dari almari*
- *Kupas kentang*
- *Kembalikan kantong kentang ke rak*

Ada kejadian yang mempunyai suatu **pola tingkah laku**, atau disingkat **pola**. Kejadian tersebut akan terjadi jika pola ini diikuti. Efek neto dari kejadian ditentukan sepenuhnya oleh pola tersebut dan (mungkin) oleh keadaan awal (yaitu keadaan pada saat T_0). Kejadian yang lain mungkin mengikuti pola yang sama. Jika dua kejadian dengan pola yang sama menghasilkan efek neto yang berbeda, maka efek neto tersebut pasti tergantung pada keadaan awal, dan dapat dipastikan pula bahwa keadaan awal dari keduanya berbeda.

Bagaimana cara mengamati pola yang sama dari berbagai kejadian, tidak dapat dijelaskan disini. Jika kita berjumpa dengan seorang teman, kita pasti segera dapat mengenalinya, apapun ekspresi yang sedang ditampilkannya. Mungkin ia sedang gembira, tertawa, menangis, atau bahkan ekspresi lain yang belum pernah ditampilkannya. Kita dapat mengenali teman tersebut karena kita kenal akan polanya.

Demikian juga dengan kejadian yang berbeda, dapat pula dikenal pola-pola yang sama, walaupun disarikan dari keadaan awal dan efek neto yang mungkin berbeda. Mengenali pola ini sama halnya nanti dengan mengenali pola-pola solusi algoritmik untuk kelas persoalan tertentu yang akan dipelajari, menjadi bagian dari belajar memprogram.

Kembali ke contoh ibu Tati yang mengupas kentang. pada suatu hari ibu Tati mengupas kentang untuk malam dan kejadian tersebut kita amati. Keesokan harinya, ia mengupas kentang lagi untuk makan malam juga. Pada pengamatan yang kedua, kita **amati hal-hal yang sama** dengan hari sebelumnya. Dapatlah kita katakan: "Jelas, pengamatan tentang kedua kejadian akan **sama** karena ibu itu **mengerjakan hal-hal sama?**"

Pernyataan terakhir tersebut dapat benar atau salah, tergantung pada apa yang dimaksud dengan "mengerjakan hal yang sama". Untuk menyatakan "hal yang sama" harus hati-hati. Tinjaulah murid-murid sekolah dasar yang berpakaian sama, karena mereka memakai seragam. Apa yang ingin dinyatakan sebagai "sama" adalah bahwa baju dari setiap murid terbuat dari bahan yang sama dan modelnya sama pula; tanpa memperhitungkan kemungkinan adanya perbedaan ukuran tergantung dari perawakan setiap murid. Demikian pula bahwa seorang murid dapat mempunyai lebih dari satu stel seragam yang sama.

Kedua dari aksi ibu Tati mengupas kentang pada dua hari yang berlainan tersebut juga dapat dipandang berbeda, seperti halnya baju murid sekolah dasar tersebut. Kejadian yang satu terjadi pada hari Sabtu dan yang lain pada hari Minggu. Karena setiap kentang hanya dapat dikupas satu kali, maka kentang yang terlibat pada kedua kejadian tersebut "berbeda" pula. Pada hari Minggu, mungkin kantong kentangnya berisi lebih sedikit dari kemarinnya, dan sebagainya.

Tetapi, kedua kejadian mengupas kentang pada hari Sabtu dan Minggu dapat pula dikatakan sama karena kemiripannya, dan sepakati untuk memberi nama yang sama untuk kedua aksi tersebut, misalnya "MENGUPAS KENTANG UNTUK MAKAN MALAM".

Algoritma adalah deskripsi dapat terdiri dari suatu pola tingkah laku, dinyatakan dalam **primitif**, yaitu aksi-aksi yang didefinisikan sebelumnya dan diberi nama, dan diasumsikan sebelumnya bahwa aksi-aksi tersebut dapat dikerjakan sehingga dapat menyebabkan kejadian yang dapat diamati.

Suatu algoritma dapat terdiri dari beberapa subalgoritma, jika setiap sub-aksi juga dapat diuraikan dalam urutan yang dapat dimengerti dengan baik dan terbatas.

Pengertian algoritma, yaitu sebagai suatu petunjuk untuk mewujudkan suatu efek neto, telah sangat dikenal dalam kehidupan sehari-hari. Petunjuk untuk merajut, resep-resep kue, aturan pakai suatu peralatan elektronik, adalah algoritma, yaitu deskripsi dari pola tingkah laku yang jika dikerjakan akan membawa ke tujuannya.

Aksi primitif **harus dapat dikerjakan**. “Pergi ke seberang jalan!” adalah aksi yang dapat dikerjakan, sedangkan “Pergi ke Neraka!” bukan algoritma, karena tidak dapat dikerjakan.

Urut-urutan langkah harus dapat dimengerti dengan baik, oleh pembuat algoritma maupun oleh yang akan mengerjakannya. Tidak boleh ada sedikitpun salah pengertian di antara keduanya supaya dapat dihasilkan efek yang diinginkan.

Jika pada suatu resep kue dituliskan “Panaskan dulu oven” maka instruksi tersebut tidak jelas, karena berapa lama dan sampai temperatur oven mencapai berapa derajat hal tersebut harus dilakukan, tidak ditentukan dengan pasti.

Sekarang perhatikanlah laporan pengamatan tentang kejadian ibu Tati yang mengupas kentang :

- *Ibu Tati mengambil kantong kentang dari rak*
- *Ibu Tati mengambil panci dari almari*
- *Ibu Tati mengupas kentang*
- *Ibu Tati mengembalikan kantong kentang ke rak*

Bandingkanlah hasil pengamatan di atas dengan teks berikut, yang merupakan algoritma, yaitu sekumpulan instruksi yang diberikan oleh ibu Tati kepada pembantu barunya :

- *Ambil kantong kentang dari rak*
- *Ambil panci dari almari*
- *Kupas kentang*
- *Kembalikan kantong kentang ke rak*

Jika teks algoritma tersebut diberikan kepada pembantunya yang bernama Ina, maka jika dilaksanakan akan menghasilkan pengamatan kejadian :

- *Ina mengambil kantong kentang dari rak*
- *Ina mengambil panci dari almari*
- *Ina mengupas kentang*
- *Ina mengembalikan kantong kentang ke rak*

Atau jika putri sulung Ibu Tati yang bernama Aida pada suatu hari dengan senang hati mengerjakan pengupasan kentang, maka akan dihasilkan pengamatan kejadian :

- *Aida mengambil kantong kentang dari rak*
- *Aida mengambil panci dari almari*
- *Aida mengupas kentang*
- *Aida mengembalikan kantong kentang ke rak*

Dengan membandingkan teks hasil pengamatan terhadap algoritma, dapat ditarik kesimpulan: algoritma ibu Tati menyatakan cara-cara untuk melakukan sesuatu sedangkan laporan pengamatan menjelaskan tentang kejadian itu sendiri. Adakah kesimpulan yang lain ? Tentu saja tidak ada, jika kita batasi bahwa algoritma yang diberikan adalah sederetan aksi-aksi bernama, yang harus dikerjakan dengan urutan tertentu. Dengan batasan ini, pengamat-pengamat dapat melaporkan dengan baik suatu aksi sesuai yang terjadi. Tetapi kelakuan dari ibu Tati (atau pembantu) dapat lebih rumit. Misalnya sehabis mengambil panci ia memakai celemek **jika perlu**, yaitu jika kebetulan ia memakai baju berwarna muda. Maka pada suatu hari ia memakai celemek, sedangkan pada hari lain tidak.

Secara umum dapat menyebut tentang celemek dan kondisi yang menyebabkan celemek tersebut dipakai, satu laporan pengamatan dapat ditulis untuk setiap kejadian:

Misalnya suatu hari, dihasilkan laporan pengamatan sebagai berikut :

- *Ibu Tati mengambil kantong kentang dari rak*
- *Ibu Tati mengambil panci dari almari*
- ***Ibu Tati memakai celemek***
- *Ibu Tati mengupas kentang*
- *Ibu Tati mengembalikan kantong kentang ke rak*

atau pada suatu hari yang lain, dihasilkan laporan pengamatan yang tidak sama dengan sebelumnya :

- *Ibu Tati mengambil kantong kentang dari rak*
- *Ibu Tati mengambil panci dari almari*
- *Ibu Tati mengupas kentang*
- *Ibu Tati mengembalikan kantong kentang dari rak*

Sekarang, masalahnya adalah bagaimana menuliskan teks pengamatan yang sama dari kedua laporan pengamatan yang berbeda tersebut, misalnya :

- *Ambil kantong kentang dari rak*
- *Ambil panci dari almari*
- ***Lakukan persiapan, tergantung pakaian***
- *Kupas kentang*
- *Kembalikan keranjang kentang ke rak*

Dengan pengertian implisit “lakukan persiapan tergantung pakaian” menyertakan tidak ada aksi jika pakaian tidak berwarna muda dan menyatakan pemakaian celemek jika pakaian berwarna muda.

Tetapi, jika diinginkan lebih terinci dan ingin menyebut secara eksplisit maka **lakukan persiapan tergantung pakaian** harus diganti dengan hasil pengamatan pada hari yang bersangkutan.

Maka pada hari Sabtu :

- *“Ibu Tati melihat bahwa bajunya tidak berwarna muda karena itu ia **tidak memakai celemek**” (berarti tidak ada aksi memakai celemek)*

Sedangkan laporan pada hari Minggu:

- *“Ibu Tati melihat bahwa bajunya berwarna muda karena itu ia **memakai celemek**”*

Pada derajat yang rinci tidak mungkin kedua kejadian ini dilaporkan dalam satu laporan pengamatan, karena terperinci, kedua kejadian tersebut berbeda. Inilah algoritma, yaitu menyatakan pola tingkah laku yang sama untuk dua, bahkan tak berhingga kejadian yang berbeda dan dengan menjelaskan pola tersebut memberikan sesuatu yang dapat terjadi pada suasana lingkungan apapun (dalam contoh tersebut, baju warna gelap ataupun muda). Apa yang sebenarnya terjadi jika suatu pola tingkah laku diikuti dapat ditentukan pola oleh keadaan yang berlaku ketika aksi tersebut mulai.

Ada dua hal yang penting. Pertama, **pengamatan apakah** baju si ibu berwarna muda, dan kedua berdasarkan pengamatan tersebut **aksi** “memakai celemek” bisa terjadi atau tidak (berarti aksi tersebut kondisional). Maka notasi untuk aksi kondisional dinyatakan oleh **kondisi** dan **aksi**).

- *Ambil kantong kentang dari rak*
- *Ambil panci dari almari*
- **if** *baju berwarna muda* **then**
Pakai celemek
- *Kupas kentang*
- *Kembalikan kantong ke rak*

Maka aksi kondisional mengandung dua aksi, aksi pertama harus suatu **pengamatan**. Hasil dari pengamatan ini adalah suatu keadaan benar (“true”) atau salah (“false”). Aksi kedua menghasilkan kejadian berlangsung sesuai dengan hasil pengamatan. Jika pengamatan memberikan hasil “true” maka aksi terjadi, jika pengamatan memberikan hasil “false” maka aksi tidak dilakukan.

Selain notasi untuk aksi kondisional, kita perlukan lagi beberapa notasi yang menunjukkan bahwa algoritma lebih tinggi tingkatannya dan menyangkut abstraksi dari pengamatan, yaitu notasi yang mewakili proses **pengulangan**. Misalnya kita ingin menyatakan bahwa “mengupas kentang” adalah suatu proses mengerjakan **satu** buah kentang pada suatu saat, maka aksi primitif kita adalah “kupas 1 kentang”. Jika jumlah kentang yang ingin dikupas selalu sama setiap hari, misalnya 25 maka sebagai ganti “kupas kentang” dapat dituliskan 25 kali “kupas 1 kentang”, masing-masing

pernyataan dituliskan per baris sehingga keseluruhannya dituliskan dalam 25 baris sekuensial. Tetapi jika kentang yang dikupas tidak selalu sama (dan hal ini lebih sering terjadi dalam kenyataan) sedangkan kita tetap menginginkan pola kelakuan yang sama apa yang harus dilakukan ? Setiap kali kita harus mengganti teks, satu jenis teks untuk satu kali pengamatan. Ini bukan tujuan dari penulisan algoritma yang mampu menghasilkan pengamatan yang berbeda-beda. Dianggap bahwa si ibu mampu untuk melongok ke panci dan dengan demikian mengamati apakah kentang yang dibutuhkan telah cukup.

Jika diketahui bahwa kasus yang ekstrem adalah mengupas 500 kentang (karena kentangnya sangat kecil-kecil dan ada pesta), artinya ibu Tati tidak mungkin mengupas lebih dari 500 kentang, kita dapat menuliskan algoritma umum untuk mengupas kentang dengan menuliskan 500 (lima ratus) kali secara sekuensial pernyataan berikut :

- **if** jumlah kentang yang sudah dikupas (belum cukup) **then**
 Kupas 1 kentang

Siapa pun pasti merasa keberatan dengan cara penulisan semacam itu, yaitu harus menuliskan hal yang sama 500 kali. Dengan asumsi dasar bahwa sebelumnya harus diketahui berapa jumlah kentang yang harus dikupas, batas seperti itu terlalu besar untuk rata-rata yang terjadi. Jika sebenarnya hanya diinginkan mengupas 25 buah kentang, maka pengamatan ke 26 akan memberikan hasil "false" yang pertama, dan 474 pengamatan berikutnya tidak akan memberikan hasil pengamatan yang baru. Sekali si ibu telah tahu bahwa kentang yang dikupasnya cukup, tidak perlu lagi memaksa dia melongok ke panci 474 lagi untuk meyakinkan dirinya sendiri.

Untuk mengatasi hal ini, diperkenalkan suatu notasi yang menjelaskan tentang suatu proses **pengulangan** sampai dijumpai keadaan tertentu, dan dituliskan sebagai :

while (kondisi.....) do
 Aksi.....

Dengan notasi ini algoritma mengupas kentang dapat dituliskan :

- Ambil kantong kentang dari rak
- Ambil panci dari almari
- **if** baju berwarna muda **then**
 pakai celemek
- **while** jumlah kentang terkupas belum cukup **do**
 Kupas 1 kentang
- Kembalikan kantong kentang ke rak

Contoh berikut, akan dijelaskan pola kelakuan dari ibu Tati yang menggunakan primitif sama, yang karena alasan tertentu selalu mengupas kentang dengan **jumlah genap** untuk masakannya. Maka dapat dituliskan algoritma sebagai berikut :

- *Ambil kantong kentang dari rak*
- *Ambil panci dari almari*
- **if** *baju berwarna muda then*
pakai celemek
- **while** *jumlah kentang terkupas belum cukup do*
Kupas 1 kentang
Kupas 1 kentang
- *Kembalikan kantong kentang ke rak*

Contoh di atas menunjukkan bahwa aksi primitif yang sama dapat menggambarkan pola kelakuan yang berbeda.

Berikut ini diandaikan bahwa ibu Tati dalah penggemar kentang sehingga ia selalu mempunyai beberapa kantong kentang di raknya. Kantong kentangnya kemungkinan ada yang berisi ataupun kosong.

Jika pengupasan kentang dapat dilakukan dari beberapa kantong, dapat dituliskan algoritma untuk mengupas sejumlah tertentu kentang sebagai berikut :

- *Ambil kantong kentang dari rak*
- *Ambil panci dari almari*
- *depend on baju*
berwarna muda : pakai celemek
tidak berwarna muda :-
- **while** *jumlah kentang terkupas belum cukup do*
 - *Depend on kantong kentang*
ada isinya : Kupas 1 kentang
tidak ada isinya : Ambil kantong kentang lain dari rak
Kupas 1 kentang

Dari contoh yang terakhir dapat pula ditarik kesimpulan bahwa suatu algoritma dapat dibangun dari aksi primitif dan gabungan dari notasi standard yang telah kita kenal. Satu algoritma mewakili beberapa kejadian yang berdasarkan pengamatan berbeda.

Algoritma adalah suatu sebuah teks yang tidak tergantung waktu, konsepnya statik. Di pihak lain ada realisasi kejadian yang dicakup oleh algoritma tersebut, yaitu suatu eksekusi yang dinamik, terjadi tergantung pada waktu, yang dijelaskan sebagai hasil dari pengamatan.

Telah dikatakan bahwa aksi harus terjadi dalam selang waktu yang terbatas, maka algoritma yang menjelaskan tentang aksi tersebut harus mencakup hal tersebut.

Kita tidak boleh menuliskan :

- *while pakaian berwarna muda do*
Kupas 1 kentang berikutnya

Karena pengupasan kentang tidak mempengaruhi warna pakaian, hanya ada dua kemungkinan: pakaian tidak berwarna muda dan pengupasan kentang **tidak pernah dilakukan** atau pakaian berwarna muda dan **proses pengupasan kentang akan dilakukan terus menerus**, yang berarti bahwa jika kebetulan pakaian berwarna muda, maka pengamatan akan menghasilkan sesuatu yang tidak pernah berhenti (looping). Contoh ini adalah sebuah algoritma yang salah karena dapat mengakibatkan pengulangan yang tidak pernah berhenti.

Tidak mudah untuk menentukan apakah suatu teks yang tampak seperti sebuah algoritma adalah memang algoritma yang benar. Bahkan **tidak mungkin** untuk membuat sebuah algoritma yang mampu memeriksa suatu teks dan menentukan apakah teks tersebut suatu algoritma yang benar. Maka adalah tanggung jawab moral orang-orang yang profesinya membuat algoritma untuk mempersiapkan bukti bahwa algoritma yang dibuatnya adalah sebuah algoritma yang benar.

Pengertian dasar yang lain adalah tentang **mesin**. Suatu mesin adalah sebuah **mekanisme** yang dapat menyebabkan suatu aksi terjadi mengikuti suatu pola tingkah laku yang dijelaskan oleh algoritma yang urutan pelaksanaannya dinyatakan dalam aksi primitif mesin tersebut.

Pada contoh-contoh di atas, diberikan beberapa algoritma tentang mengupas kentang. Semua algoritma dinyatakan dalam primitif yang sama yang kemudian melahirkan teks yang dinyatakan sebagai "pengamatan kejadian". Siapapun yang mampu untuk:

- Mengerjakan aksi primitif tersebut
- Menerima algoritma yang dinyatakan dengan primitif tersebut dan akan melaksanakan langkah-langkah dengan patuh, disebut sebagai **mesin**.

Jika saya dapat membuat teman saya, pembantu saya, tetangga kiri saya, atau tetangga kanan saya mengerjakan pengupasan kentang tersebut tergantung algoritma yang saya berikan, maka teman, pembantu, tetangga kiri maupun tetangga kanan saya adalah sebuah mesin.

Suatu mekanisme yang hanya dapat mengerjakan satu hal yang selalu sama (misalnya toilet flusher) tidak dapat disebut suatu mesin. Hal penting yang dapat dikerjakan oleh mesin adalah aksi-aksi yang sekuensial, kemampuan menerima suatu pola tingkah laku dan berkelakuan

berdasarkan pola tersebut. Mesin adalah **pengeksekusi atau pelaku dari algoritma**.

Algoritma yang mengontrol pola tingkah laku suatu mesin disebut program. Dengan perkataan lain, program adalah algoritma yang dimaksudkan untuk dieksekusi oleh mesin. Dalam pengertian algoritma yang harus dapat dimengerti dengan baik, tanpa menghiraukan bagaimana pengertian tersebut diwujudkan. Mesin terdiri dari sekumpulan peralatan. Berkat konstruksinya, mesin hanya dapat mengerjakan sekumpulan instruksi-instruksi tertentu yang terbatas jumlahnya yang telah terdefinisi.

Jika kita berikan suatu program kepada mesin, maka dengan patuh ia akan mengerjakan persis seperti yang dinyatakan dalam algoritma. Mesin sangat "patuh" terhadap instruksi yang kita berikan, dengan resiko kita harus mendefinisikan instruksi tersebut dengan rinci. Untuk seorang pemrogram yang belum berpengalaman, hal ini sering menimbulkan keluhan, tetapi dengan bertambahnya pengalaman, ia akan menyadari bahwa mesin akan selalu mengerjakan hal-hal yang dianggap "tidak umum" tanpa membantah. Bandingkanlah dengan pembantu yang seringkali mengadakan penafsiran sendiri terhadap instruksi kita supaya ia lebih enak, tetapi sering malahan menjengkelkan dan menyulitkan kita karena interpretasi dan tingkah lakunya yang tak terkontrol oleh kita.

Program yang mengontrol mesin harus disusun sedemikian rupa jika ingin dipakai sesuai keinginan. Misalnya kita ingin menyuruh mesin tersebut untuk memecahkan masalah yang kita hadapi, maka kita harus membuat program yang sesuai untuk masalah tersebut dan mesin akan mengerjakan program sesuai dengan algoritma yang ditulis. Dalam hal ini mesin tersebut adalah alat bantu untuk memecahkan dari sudut pandang ini.

Pada bagian berikutnya akan dijelaskan, apakah pemrograman itu, dan mesin pengeksekusi program yang selanjutnya disebut sebagai komputer. Primitif-primitif yang akan diuraikan pada bab-bab selanjutnya adalah primitif yang dapat dilakukan oleh komputer. Aksi primitif komputer terlalu detil untuk jalan pikiran manusia, sehingga terlalu sulit untuk menguraikan algoritma dalam primitif langsung komputer. Karena itu diperlukan mesin abstrak, yaitu mekanisme yang diasumsikan dapat dilakukan oleh sekumpulan primitif yang diberikan. Berangsur-angsur, secara bertahap, mesin abstrak akan dijabarkan menjadi mesin riil, yaitu sampai primitif yang dapat dilakukan oleh komputer. Keadaan awal dan keadaan akhir yang diceritakan di atas pada kejadian nyata, juga akan diwakili oleh keadaan komputer, dalam hal ini keadaan isi memori. Efek neto yang akan dihasilkan dinyatakan dalam spesifikasi program, yang menjadi bahan mentah dalam menuliskan programnya. Dari kejadian sehari-hari yang diuraikan pada bab ini, kita akan berbicara dalam notasi algoritmik. Notasi kondisional dan pengulangan di atas baru sebagian dari notasi algoritmik yang akan dipelajari secara bertahap (karena itu dituliskan dalam bahasa

Inggris), untuk membedakan dengan kalimat-kalimat ibu Tati dalam bahasa manusia.

Latihan Soal

1. Periksalah apakah masing-masing algoritma “mengupas kentang” berikut benar. Pada algoritma ini kita hanya tertarik pada aksi mengambil kentang dan kantong dan mengupas kentang.

1. *While jumlah kentang terkupas belum cukup do
depend on kantong
kantong berisi 1 kentang : Kupas 1 kentang*
2. *While kantong tidak kosong do
Kupas 1 kentang
Kupas 1 kentang*
3. *While kantong tidak kosong do
While jumlah kentang terkupas belum cukup do
Kupas 1 kentang*
4. *While (kantong tidak kosong) dan (jumlah kentang terkupas belum cukup) do
Kupas 1 kentang*
5. *While jumlah kentang terkupas belum cukup do
Depend on kantong
kantong tidak kosong : kupas 1 kentang
kantong kosong : ambil kantong lain*
6. *While jumlah kentang terkupas belum cukup do
While kantong tidak kosong do
Kupas 1 kentang*
7. *While jumlah kentang terkupas belum cukup do
depend on kantong :
kantong tidak kosong : Kupas 1 kentang
kantong kosong : -*
8. *While kantong ada isinya do
depend on jumlah kentang terkupas
jumlah kentang terkupas belum cukup : Kupas 1 kentang
jumlah kentang terkupas cukup :Ambil kantong lain
Kupas 1 kentang*
9. *While kantong ada isinya do
depend on jumlah kentang terkupas
jumlah kentang terkupas belum cukup : Kupas 1 kentang
jumlah kentang terkupas cukup : Stop*

10. *depend on kantong*

*kantong tidak kosong : while jumlah kentang terkupas belum cukup
do*

kupas 1 kentang

kantong kosong : beli kentang lagi

11. *Pada algoritma ini berlaku hypotesa : mula-mula kantong penuh*

Kupas 1 kentang

depend on jumlah kentang terkupas

jumlah kentang terkupas belum cukup : Kupas 1 kentang

jumlah kentang terkupas cukup: kembalikan kantong ke raknya

12. *While kantong ada isinya do*

Kupas 1 kentang

depend on jumlah kentang terkupas

jumlah kentang terkupas belum cukup: cari kentang lagi

jumlah kentang terkupas cukup : -

13. *While (jumlah kentang terkupas belum cukup) dan*

(kantong ada isinya) do

kupas 1 kentang

14. *depend on kantong*

kantong ada isinya : while jumlah kentang terkupas belum cukup do

Kupas 1 kentang

kantong tidak ada isinya :

Taruh kentang dalam kantong

while jumlah kentang terkupas belum cukup do

Kupas 1 kentang

II. Distribusi gula-gula

a. Ada sekantong gula-gula, hendak dibagikan merata ke empat orang anak. Tiap

anak harus mendapat jumlah yang sama, dan jika sisanya tidak cukup untuk

dibagikan ke empat anak tersebut, maka sisanya tidak dibagikan.

Tuliskanlah

algoritmanya.

b. Jika gula-gula tersebut mempunyai rasa jeruk, mentol, arbei dan durian, dan setiap

anak harus mendapat jumlah dan rasa yang sama, tuliskan pula algoritma untuk

membaginya.

Catatan : Definisikan dahulu aksi primitif untuk persoalan ini.