

BAB 1

PENDAHULUAN

Pada Bab ini anda akan mempelajari

1. Pengertian algoritma
2. Hubungan algoritma dengan pemrograman
3. Penulisan algoritma dengan Teks Umum
4. Penulisan algoritma dengan Teks Standar
5. Penulisan algoritma dengan Flowchart

1.1. Pengertian Algoritma

Pada prinsipnya seluruh aspek kehidupan kita dipenuhi oleh berbagai permasalahan. Setiap orang tentunya berupaya menyelesaikan masalah tersebut dengan melakukan berbagai langkah. Misalnya, jika seseorang ingin kuliah di suatu perguruan tinggi maka yang ia lakukan tentunya mendaftar ke perguruan tinggi dimaksud, kemudian mengikuti tesnya, dan jika lulus ia melakukan pendaftaran ulang. Jika seseorang ingin memiliki SIM ia tentunya akan mendatangi kantor polisi, kemudian mendaftar, mengisi data, membayar sejumlah uang, mengerjakan tes, sampai pemotretan. Demikian pula halnya jika seseorang hendak menikmati mie instan, memiliki rumah, memiliki KTP, atau mengganti ban bocor, semuanya memiliki kesamaan : memerlukan proses.

Semua yang kita lakukan seperti contoh di atas memerlukan proses. Menjadi mahasiswa, memiliki SIM, membangun rumah dan menikmati mie instan tidak dapat terjadi begitu saja tanpa proses. Proses merupakan serangkaian tindakan (aksi) dilakukan untuk

mencapai suatu tujuan. Serangkaian tindakan itu kita sebut dengan langkah-langkah penyelesaian masalah. Langkah-langkah ini berjalan dengan urutan tertentu. Anda tidak bisa memiliki KTM (kartu tanda mahasiswa) sebelum anda membayar biaya kuliah. Anda tidak bisa membayar mendaftar ulang sebelum anda dinyatakan lulus dalam tes. Berarti urutan dalam langkah-langkah tersebut merupakan hal yang penting untuk menentukan keberhasilan menyelesaikan masalah. Langkah-langkah penyelesaian masalah dengan urutan tertentu ini kita sebut dengan algoritma. Ini adalah pengertian umumnya.

Baiklah, untuk jelasnya kita coba mendefinisikan sebuah masalah dan langkah-langkah penyelesaiannya atau algoritmanya. Misalnya, kita memiliki masalah tentang bagaimana cara agar bisa kuliah di suatu perguruan tinggi. Langkah apa yang harus dilakukan dan bagaimana urutannya. Salah satu solusinya dapat dijabarkan sebagai berikut.

1. Mulai
2. Memilih perguruan tinggi
3. Mengisi formulir pendaftaran
4. Mengikuti tes masuk
5. Jika lulus tes, lakukan lanjut ke 6. Jika tidak lanjut ke 8
6. Mendaftar ulang/bayar SPP
7. Ambil KTM
8. Selesai

Anda bisa lihat bahwa untuk bisa mencapai tujuan (menjadi mahasiswa) anda harus melewati beberapa tahapan tertentu. Perhatikan langkah-langkah menjadi mahasiswa berikut ini

1. Mulai
2. Mengikuti tes masuk
3. Memilih perguruan tinggi
4. Mengisi formulir pendaftaran
5. Membayar SPP
6. Jika lulus tes, lanjut ke 3. Jika tidak lanjut ke 8
7. Ambil KTM
8. Selesai

Bagaimana menurut anda langkah-langkah di atas? Kacau sekali, bukan? Urut-urutannya tidak tertata dengan benar. Kita sebut langkah-langkah tersebut di atas **tidak logis**, artinya ia tidak sesuai dengan kaidah berpikir yang benar. Dengan demikian, langkah-

langkah penyelesaian masalah tidak akan memberikan solusi jika urutannya tidak tertata dengan benar. Namun demikian, perlu diingat bahwa bisa jadi suatu persoalan dapat diselesaikan dengan beragam langkah dan urutan. Misalnya, untuk membuat mie instan dapat dilakukan dengan langkah

1. Mulai
2. Merebus air
3. Memasukkan mie ke dalam air yang mendidih
4. Menuangkan mie yang telah matang ke dalam mangkok
5. Masukkan bumbu masak
6. Aduk sampai rata
7. Selesai

Atau langkah ini

1. Mulai
2. Merebus air
3. Memasukkan mie ke dalam air yang mendidih
4. Masukkan bumbu masak
5. Aduk sampai rata
6. Menuangkan mie yang telah matang ke dalam mangkok
7. Selesai

Bagaimana menurut anda kedua langkah di atas? Tentunya keduanya berbeda, tetapi kedua cara tersebut dapat menyelesaikan persoalan yang sama. Dengan demikian bisa jadi terdapat berbagai variasi jawaban untuk menyelesaikan suatu masalah.

1.2. Pengertian Membuat Program

Barangkali ada yang bertanya, bukankan pernyataan-pernyataan algoritma di atas memiliki sub aksi yang lebih kecil? Misalnya, aksi merebus air bisa jadi terdiri dari beberapa aksi, seperti : mengambil panci, lalu membuka keran dan meletakkan panci di bawah keran. Kemudian jika telah cukup, matikan keran, lalu letakkan panci di atas

kompor, lalu hidupkan kompor, dan seterusnya. Bahkan sub pernyataan dari merebus air pun jika diurai lebih dalam akan terdiri dari beberapa sub pernyataan lebih banyak lagi.

Betul, demikian adanya. Namun, jika kita terus menuliskan semua sub pernyataan tersebut, bisa tak hingga banyaknya. Dalam hal ini ada banyak hal yang kita anggap sudah cukup mewakili atau jelas dengan satu pernyataan aksi saja. Misalnya, untuk menyatakan semua proses yang terjadi dalam aksi merebus air seperti yang telah disebutkan di atas, cukup diwakili oleh satu pernyataan yaitu : merebus air.

Bagaimana ukurannya apakah suatu pernyataan aksi sudah dianggap jelas atau belum? Hal ini bisa jadi sangat subyektif. Ia sangat tergantung kepada siapa perintah itu diberikan. Jika perintah “memasak rendang” diberikan kepada seseorang yang baru belajar memasak, rincian perintahnya haruslah cukup mendetail. Tetapi bagi seorang koki yang sudah berpengalaman, satu perintah “memasak rendang” sudah cukup baginya tanpa harus dirinci lebih jauh.

Demikian pula halnya komputer. Algoritma yang diterjemahkan ke dalam program komputer adalah cara kita memberikan perintah kepada komputer sampai pada taraf “dimengerti” oleh komputer. Kita tidak bisa memerintahkan kepada komputer “hitung 5 pangkat 10” begitu saja, tetapi hal itu bisa dilakukan jika kita merinci perintah-perintahnya.

Itulah gunanya bahasa pemrograman. Bahasa pemrograman adalah media untuk merinci perintah-perintah manusia kepada komputer. Bahasa pemrograman itu ada yang tingkat rendah seperti bahasa Assembly, dan ada pula yang tingkat tinggi seperti Pascal, Fortran, C, C++, Cobol, Basic, dll. Semakin tinggi level suatu bahasa pemrograman semakin dekat pula ia dengan bahasa manusia, artinya dengan sedikit perintah komputer sudah bisa menyelesaikan masalah yang kita minta untuk dikerjakan. Sebaliknya semakin rendah level bahasa pemrograman, semakin jauh ia dari bahasa manusia. Artinya, lebih

banyak perintah yang harus diberikan untuk menyelesaikan persoalan yang sama. Bekerja dengan bahasa pemrograman tingkat tinggi ibarat memberi perintah kepada koki yang sudah terlatih. Cukup dengan satu atau beberapa perintah saja, sang koki sudah tahu apa yang harus dikerjakannya. Sebaliknya, bekerja dengan bahasa pemrograman tingkat rendah ibarat memberi perintah kepada seorang calon koki yang masih belum mahir sehingga perintah harus diberikan secara lebih terperinci.

1.3. Antara Algoritma dan Pemrograman

Seperti telah dijelaskan sebelumnya algoritma secara umum berhubungan dengan semua langkah-langkah penyelesaian masalah dengan urutan tertentu. Tetapi istilah algoritma saat ini lebih sering dikaitkan dengan pemrograman komputer sehingga dikenal algoritma pemrograman. Pada dasarnya algoritma pemrograman merupakan kerangka dari program komputer. Program komputer sendiri merupakan sekumpulan perintah yang ditulis dalam bahasa yang bisa dimengerti komputer untuk menyelesaikan masalah tertentu.

Dalam konteks pemrograman komputer terdapat banyak definisi algoritma. Salah satunya definisi algoritma adalah *urutan langkah berhingga untuk menyelesaikan masalah matematika dan logika*. Dengan pengertian itu dapat kita simpulkan bahwa

1. Algoritma adalah kumpulan langkah, artinya algoritma merupakan proses di dalam penyelesaian masalah.
2. Langkah-langkah dalam algoritma berhingga banyaknya, artinya langkah-langkah tersebut memiliki akhir. Algoritma tanpa akhir, bukanlah termasuk algoritma karena tidak akan memberikan solusi apapun.
3. Digunakan untuk menyelesaikan masalah matematika dan logika, artinya tidak semua masalah bisa diselesaikan dengan algoritma. Algoritma hanya bisa digunakan untuk menyelesaikan masalah logika dan matematika, atau masalah-masalah yang dapat dinyatakan dalam bahasa logika dan matematika.

Komputer adalah alat bantu manusia untuk menyelesaikan masalah. Untuk bisa membantu manusia, komputer harus diberi perintah. Perintah itu dikenal dengan program. Namun demikian untuk memberi perintah kepada komputer tidak sama dengan memberi perintah kepada manusia. Komputer memiliki "logika berpikirnya" sendiri yang berbeda dengan manusia. Oleh karena itu untuk bisa memberi perintah kepada komputer, manusia harus bisa memberi perintah yang mampu "dipahami" komputer.

Algoritma adalah cara manusia untuk mengkomunikasikan perintahnya dengan paradigma komputer. Itu sebabnya semua orang yang belajar pemrograman komputer sangat dianjurkan memahami dulu algoritma, disamping tentunya juga bahasa pemrograman. Pemahaman terhadap algoritma sangat penting untuk melatih calon pemrogram berpikir secara terstruktur dan sistematis. Kemampuan berpikir terstruktur dan sistematis akan sangat membantu dalam memudahkan pencarian solusi dengan cara yang sesederhana mungkin. Pemrogram yang buruk adalah pemrogram yang meskipun mampu membuat program dan memecahkan masalahnya, namun programnya tidak terstruktur dengan baik. Programmer jenis ini akan menulis 100 baris perintah program untuk persoalan yang sebenarnya bisa diselesaikan dengan 20 atau 30 baris perintah saja.

1.4. Penulisan Algoritma Teks Umum

Sebagaimana telah dijelaskan algoritma adalah kumpulan perintah untuk menyelesaikan suatu persoalan. Dalam kaitannya dengan program, perintah algoritma sedapatnya merupakan perintah yang spesifik dan tidak mengandung ambiguitas (mengandung banyak penafsiran). Semakin spesifik suatu perintah akan semakin memudahkan kita mengkonversinya ke dalam bahasa pemrograman.

Perintah algoritma dapat dinyatakan dalam dua cara : dengan teks dan flowchart. Algoritma teks adalah algoritma yang dinyatakan dengan aksara / huruf (dan angka) yang lazim kita kenal selama ini. Algoritma ini juga dikenal dengan istilah *pseudo code*. Penulisan algoritma ini bersifat bebas (tidak baku), yang penting harus mudah dipahami

oleh pembacanya.. Berikut ini adalah contoh algoritma yang menghitung dan menampilkan hasil penjumlahan dua buah bilangan yang ditulis dengan teks umum.

1. Mulai
2. Baca bilangan a dan b
3. Hitung a ditambah b, simpan pada c
4. Tampilkan nilai c
5. Selesai

1.5. Penulisan Algoritma dengan Teks Standar

Sebagaimana telah disebutkan sebelumnya bahwa penulisan algoritma teks sebenarnya tidaklah baku, tidak seperti menulis program yang harus tepat sampai titik komanya. Namun dalam buku ini teknis penulisannya sengaja agak dibakukan agar proses konversi algoritma ke bahasa program menjadi lebih mudah. Struktur dan perintah algoritma teks standar yang kita pakai di sini dibuat mirip dengan yang ada pada bahasa pemrograman Pascal. Jadi, pengertian baku di sini sebenarnya tidaklah bersifat mutlak, tetapi sebagai pendekatan saja agar lebih tidak kesulitan saat mulai mempelajari bahasa pemrograman Pascal.

Struktur algoritma yang kita bakukan ini terdiri dari tiga bagian (sama dengan struktur penulisan dalam Pascal) :

1. Bagian Nama : bagian yang menuliskan nama algoritma serta keterangan lain perihal algoritma tersebut
2. Bagian deklarasi : bagian yang mendefinisikan konstanta dan variabel objek yang terlibat dalam algoritma tersebut
3. Bagian dekripsi (badan algoritma) : adalah bagian yang merupakan langkah-langkah penyelesaian masalah.

Berikut ini adalah contoh penulisan algoritma menghitung luas lingkaran dengan teks standar

```

Algoritma LuasLingkaran | ← {Bagian Nama}
{Algoritma ini menghitung luas lingkaran}

Deklarasi | ← {Bagian deklarasi}
  Pi=3.14
  Luas,Jari : real

Deskripsi | ← {Bagian deskripsi}
  Read(Jari)
  Luas←pi*Jari*Jari
  Write(Luas)

```

Jika algoritma di atas kita konversi ke dalam bahasa pemrograman Turbo Pascal maka dapat ditulis :

```

Program LuasLingkaran; | {Bagian Nama}

Uses crt;

Const | {Bagian deklarasi}
  Pi=3.14;
Var
  Luas, Jari : real;

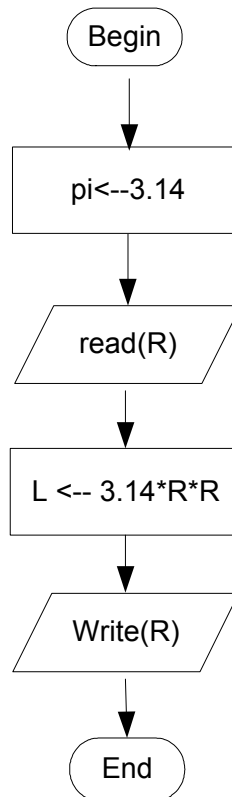
Begin | {Bagian deskripsi}
  Readln(Jari)
  Luas:=Pi*Jari*Jari;
  Writeln(Jari)
End;

```

1.6. Penulisan Algoritma dengan Flowchart



Bentuk penyelesaian masalah dengan algoritma teks terkadang agak sulit dimengerti terutama untuk persoalan yang relatif kompleks. Untuk itu sebagian orang terkadang menggunakan algoritma yang disusun dari menggunakan bagan-bagan yang lazim disebut flowchart. Dibandingkan dengan teks, cara ini struktur penyelesaian masalahnya lebih mudah dipahami tetapi penulisannya banyak menyita ruangan. Setengah halaman algoritma teks bisa jadi harus ditulis dengan berlembar-lembar algoritma flowchart.


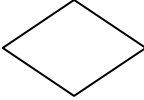


Algoritma menggunakan flowchart selalu dimulai dengan kata **begin** dan diakhiri dengan **end** sebagai tanda awal dan akhir dari algoritma. Perintah-perintahnya ditulis dalam ‘kotak-kotak’ dengan beragam bentuk. Setiap bentuk mencerminkan perintah tertentu. Berikut ini adalah contoh flowchart.



Gambar 1.1. Contoh Flowchart

Beberapa bentuk dasar flowchart berikut artinya dapat dilihat pada tabel dibawah ini.

Lambang Flowchart	Arti	Teks Perintah
	Awal / akhir program	Begin atau End
	Memasukkan (input) / Menampilkan keluaran (output)	Read(argumen)/ Write(argumen)

	Penugasan/perhitungan/ proses	(Tergantung prosesnya)
	Kotak keputusan	(tergantung prosesnya)
	Kotak penghubung	
	Arus Data	

1.7. Soal

1. Apa yang dimaksud dengan algoritma
2. Buatlah langkah-langkah penyelesaian persoalan berikut ini
 - a. mengganti ban mobil yang pecah
 - b. mengambil uang di ATM
 - c. membeli motor secara kredit
 - d. Mengisi pulsa pada telepon genggam
3. Jelaskan perbedaan algoritma dengan program
4. Jelaskan perbedaan algoritma teks dan flowchart! Sebutkan kelebihan dan kekurangan kedua metode ini masing-masing.
5. Jelaskan tentang struktur algoritma teks standar
6. Anda memiliki tiga buah ember A, B, dan C yang dua di antaranya yaitu A dan B terisi dengan air. Apa yang akan anda lakukan jika anda ingin menukar air yang ada di A dengan air yang ada di B?
7. Anda bermaksud mengisi bak air yang sumber airnya adalah sebuah keran yang terletak beberapa meter dari bak tersebut. Di dekat bak tersedia dua ember A dan B

yang masing-masing berkapasitas 3 dan 4 liter. Bagaimana caranya agar bak itu bisa terisi tepat 10 liter air dengan menggunakan ember yang tersedia dengan ketentuan :

- Hanya ember A yang bisa digunakan untuk mengangkut air ke bak
- tidak ada air yang terbuang
- tidak ada air yang tersisa di kedua ember tersebut.

8. Seorang petani bermaksud hendak menyeberangi sungai berikut barang bawaannya, yaitu sebakul sayur mayur, seekor domba dan seekor harimau. Di depannya terhampar sebuah rakit yang bisa dipakainya untuk menyeberangi sungai. Sayangnya rakit hanya bisa dipakai untuk membawa dirinya dan salah satu barang bawaannya. Jika ia menyeberang dengan membawa sayur mayur dan meninggalkan harimau dengan domba, maka habislah dombanya dimakan harimau. Tetapi jika ia pergi dengan harimau meninggalkan domba dengan sayur mayurnya, maka habislah sayur mayurnya dilahap domba. Dapatkah anda memikirkan bagaimana cara petani tersebut dapat menyeberangi sungai berikut barang bawaannya dengan selamat?