**LOGIKA DAN ALGORITMA**



**Penyusun:**

**Ali Ridho Barakbah, S.Kom., PhD.**

**NIP.197308162001121001**

**Tita Karlita, S. Kom., M.Kom.**

**NIP. 197910142002122002**

**Ahmad Syauqi Ahsan, S.Kom.**

**NIP. 197505302003121001**

**Program Studi Teknik Informatika**

**Departemen**

**Teknik Informatika dan Komputer**

**Politeknik Elektronika Negeri Surabaya**

**2013**

*Halaman ini sengaja dikosongkan*

**Kata Pengantar**

Dengan mengucapkan syukur Alhamdulillah kehadirat Allah SWT atas segala limpahan rahmat dan hidayah-Nya, sehingga penyusun dapat menyelesaikan buku ajar yang berjudul : **“Logika dan Algoritma”** dengan baik.

Buku petunjuk teori ini disusun sebagai pedoman khususnya bagi mahasiswa di Jurusan Teknik Informatika, Politeknik Elektronika Negeri Surabaya, dalam memberikan pengenalan, pemahaman serta mengasah kemampuan dalam bidang logika dan algoritma yang sudah menjadi kewajiban untuk dikenal dan dipahami sebagai dasar mata kuliah di bidang teknologi informasi. Diharapkan para mahasiswa nantinya bisa lebih terampil dan memahami segala bentuk teknologi informasi saat ini.

Bagimanapun penyusun telah berusaha membuat buku ini dengan sebaik-baiknya, namun tidak ada kesempurnaan dalam sebuah karya manusia. Penyusun menyadari masih banyak kekurangan dalam penyusunan buku ini. Untuk itu pula segala masukan, kritik dan saran dari pembaca dapat menjadikan acuan bagi penyusun dalam penyempurnaan dan pembuatan buku berikutnya.

Tiada untaian kata yang dapat penyusun sampaikan selain panjatkan doa, semoga Allah SWT selalu membuka hati kita dengan cahaya-NYA dan mengajarkan ilmu-NYA kepada kita, serta menghindarkan kita dari ilmu yang tidak bermanfaat.

Surabaya, Januari 2014

Penyusun

i

**Daftar Isi**

**KATA PENGANTAR.............................................................................................................. I DAFTAR ISI............................................................................................................................II DAFTAR GAMBAR............................................................................................................. VI DAFTAR TABEL ................................................................................................................VII BAB 1 PENGANTAR LOGIKA DAN ALGORITMA ........................................................1**

1.1. PENGERTIAN LOGIKA DAN ALGORITMA......................................................................1

1.2. CIRI-CIRI, SIFAT, STRUKTUR DASAR DAN CARA PENULISAN ALGORITMA..................4

*1.1.1 Runtunan (sequence)......................................................................................................... 5*

*1.1.2 Pemilihan (selection) ........................................................................................................ 6*

*1.1.3 Pengulangan (repetition) .................................................................................................. 6*

1.3. PERBEDAAN ALGORITMA DAN PROGRAM ...................................................................7

1.4. MEKANISME PELAKSANAAN ALGORITMA OLEH PEMROSES NOTASI PENULISAN

ALGORITMA ............................................................................................................................8

1.5. BELAJAR MEMBUAT PROGRAM DAN BELAJAR BAHASA PEMROGRAMAN .................10

1.6. LATIHAN SOAL .........................................................................................................12

**BAB 2 NOTASI PENULISAN ALGORITMA ...................................................................13**

2.1 KALIMAT DESKRIPTIF................................................................................................14

*2.1.1 Judul Algoritma............................................................................................................... 14*

*2.1.2 Bagian Deklarasi ............................................................................................................ 15*

*2.1.3 Bagian deskripsi.............................................................................................................. 16*

2.2 PSEUDOCODE ............................................................................................................17

2.3 FLOWCHART .............................................................................................................19

2.4 LATIHAN SOAL .........................................................................................................24

**BAB 3 STRUKTUR DATA DALAM ALGORITMA ........................................................28**

3.1 TIPE DATA ................................................................................................................29

*3.1.1 Tipe dasar........................................................................................................................ 29*

*3.1.2 Tipe data bentukan.......................................................................................................... 30*

*3.1.3 Tipe data abstrak (Abstract Data Type).......................................................................... 30*

3.2 KONSTANTA DAN VARIABEL ....................................................................................31

3.3 ARRAY ......................................................................................................................32

3.4 STACK.......................................................................................................................35

3.5 QUEUE ......................................................................................................................36

3.6 TREE .........................................................................................................................37

3.7 GRAPH ......................................................................................................................38

3.8 LATIHAN SOAL .........................................................................................................41

**BAB 4 STUDI KASUS PERMASALAHAN SEDERHANA .............................................42**

4.1 STUDI KASUS 1: MENGHITUNG LUAS DAN KELILING LINGKARAN .............................42

*4.1.1 Permasalahan ................................................................................................................. 42*

*4.1.2 Cara Penyelesaian Masalah ........................................................................................... 42*

*4.1.3 Struktur Data Yang Dibutuhkan...................................................................................... 43*

*4.1.4 Input ................................................................................................................................ 43*

ii

*4.1.5 Output.............................................................................................................................. 44*

*4.1.6 Proses Penyelesaian........................................................................................................ 44*

*4.1.7 Flowchart Keseluruhan................................................................................................... 45*

4.2 STUDI KASUS 2: KONVERSI SUHU .............................................................................45

*4.2.1 Permasalahan ................................................................................................................. 45*

*4.2.2 Cara Penyelesaian Masalah ........................................................................................... 45*

*4.2.3 Struktur Data Yang Dibutuhkan...................................................................................... 46*

*4.2.4 Deklarasi dan Inisialisasi................................................................................................ 46*

*4.2.5 Input ................................................................................................................................ 46*

*4.2.6 Output.............................................................................................................................. 46*

*4.2.7 Proses Penyelesaian........................................................................................................ 47*

*4.2.8 Flowchart Keseluruhan................................................................................................... 47*

4.3 STUDI KASUS 3: MENAMPILKAN BILANGAN GANJIL..................................................47

*4.3.1 Permasalahan ................................................................................................................. 47*

*4.3.2 CaraPenyelesaian Masalah ............................................................................................ 48*

*4.3.3 Struktur Data Yang Dibutuhkan...................................................................................... 48*

*4.3.4 Deklarasi dan Inisialisasi................................................................................................ 48*

*4.3.5 Input ................................................................................................................................ 48*

*4.3.6 Output.............................................................................................................................. 49*

*4.3.7 Proses Penyelesaian........................................................................................................ 49*

*4.3.8 Flowchart Keseluruhan................................................................................................... 50*

4.4 SOAL LATIHAN .........................................................................................................50

**BAB 5 STUDI KASUS PERBANDINGAN .........................................................................52**

5.1 CONTOH KASUS 1: TAHUN KABISAT .........................................................................52

*5.1.1 Permasalahan ................................................................................................................. 52*

*5.1.2 Proses Penyelesaian Masalah: ....................................................................................... 52*

*5.1.3 Input: ............................................................................................................................... 52*

*5.1.4 Output: ............................................................................................................................ 52*

*5.1.5 Struktur Data yang Dibutuhkan:..................................................................................... 53*

*5.1.6 Logika Pemrograman: .................................................................................................... 53*

*5.1.7 Flowchart:....................................................................................................................... 53*

5.2 CONTOH KASUS 2: DERET BILANGAN GENAP ............................................................53

*5.2.1 Permasalahan ................................................................................................................. 53*

*5.2.2 Cara Penyelesaian Masalah: .......................................................................................... 54*

*5.2.3 Input: ............................................................................................................................... 54*

*5.2.4 Output: ............................................................................................................................ 54*

*5.2.5 Struktur Data yang Dibutuhkan:..................................................................................... 54*

*5.2.6 Logika Pemrograman: .................................................................................................... 54*

*5.2.7 Flowchart:....................................................................................................................... 55*

5.3 SOAL LATIHAN: ........................................................................................................55

**BAB 6 STUDI KASUS KONVERSI ....................................................................................56**

6.1 CONTOH KASUS 1: KONVERSI JAM KE MENIT............................................................56

*6.1.1 Permasalahan ................................................................................................................. 56*

*6.1.2 Cara Penyelesaian Masalah: .......................................................................................... 56*

*6.1.3 Input: ............................................................................................................................... 56*

*6.1.4 Output: ............................................................................................................................ 57*

*6.1.5 Struktur Data yang Dibutuhkan:..................................................................................... 57*

*6.1.6 Logika Pemrograman: .................................................................................................... 57*

*6.1.7 Flowchart:....................................................................................................................... 57*

6.2 CONTOH KASUS 2: KONVERSI DETIK KE HARI, JAM, MENIT, DAN DETIK ....................57

*6.2.1 Permasalahan ................................................................................................................. 57*

*6.2.2 Cara Penyelesaian Masalah: .......................................................................................... 58*

*6.2.3 Input: ............................................................................................................................... 58*

*6.2.4 Output: ............................................................................................................................ 58*

*6.2.5 Struktur Data yang Dibutuhkan:..................................................................................... 58*

*6.2.6 Logika Pemrograman: .................................................................................................... 59*

*6.2.7 Flowchart:....................................................................................................................... 59*

6.3 SOAL LATIHAN: ........................................................................................................59

**BAB 7 STUDI KASUS PERCABANGAN DAN PERULANGAN....................................60**

7.1 CONTOH KASUS 1: KALKULATOR SEDERHANA .........................................................60

*7.1.1 Cara Penyelesaian Masalah: .......................................................................................... 60*

*7.1.2 Input: ............................................................................................................................... 61*

*7.1.3 Output: ............................................................................................................................ 61*

*7.1.4 Struktur Data yang Dibutuhkan:..................................................................................... 61*

*7.1.5 Logika Pemrograman: .................................................................................................... 61*

*7.1.6 Flowchart:....................................................................................................................... 62*

7.2 CONTOH KASUS 2: TUMPUKAN BILANGAN................................................................62

*7.2.1 Permasalahan ................................................................................................................. 62*

*7.2.2 Cara Penyelesaian Masalah: .......................................................................................... 62*

*7.2.3 Input: ............................................................................................................................... 63*

*7.2.4 Output: ............................................................................................................................ 63*

*7.2.5 Struktur Data yang Dibutuhkan:..................................................................................... 63*

*7.2.6 Logika Pemrograman: .................................................................................................... 63*

*7.2.7 Flowchart:....................................................................................................................... 63*

7.3 SOAL LATIHAN .........................................................................................................64

**BAB 8 STUDI KASUS TUMPUKAN (STACK).................................................................66**

8.1 CONTOH KASUS 1: MEMBALIK KALIMAT ..................................................................66

*8.1.1 Permasalahan ................................................................................................................. 66*

*8.1.2 Cara Penyelesaian Masalah: .......................................................................................... 66*

*8.1.3 Input: ............................................................................................................................... 67*

*8.1.4 Output: ............................................................................................................................ 67*

*8.1.5 Struktur Data yang Dibutuhkan:..................................................................................... 67*

*8.1.6 Logika Pemrograman: .................................................................................................... 67*

*8.1.7 Flowchart ........................................................................................................................ 67*

8.2 CONTOH KASUS 2: MEMBALIK BILANGAN ................................................................68

*8.2.1 Permasalahan ................................................................................................................. 68*

*8.2.2 Cara Penyelesaian Masalah: .......................................................................................... 68*

*8.2.3 Input: ............................................................................................................................... 69*

*8.2.4 Output: ............................................................................................................................ 69*

*8.2.5 Struktur Data yang Dibutuhkan:..................................................................................... 69*

*8.2.6 Logika Pemrograman: .................................................................................................... 69*

*8.2.7 Flowchart ........................................................................................................................ 69*

8.3 SOAL LATIHAN .........................................................................................................70

**BAB 9 STUDI KASUS KONVERSI BILANGAN..............................................................72**

9.1 . STUDI KASUS 1: KONVERSI BILANGAN BINER KE DESIMAL .....................................72

*9.1.1 Permasalahan ................................................................................................................. 72*

*9.1.2 Cara Penyelesaian Masalah ........................................................................................... 72*

*9.1.3 Struktur Data Yang Dibutuhkan...................................................................................... 73*

*9.1.4 Deklarasi dan Inisialisasi................................................................................................ 73*

*9.1.5 Input ................................................................................................................................ 73*

*9.1.6 Output.............................................................................................................................. 74*

*9.1.7 Proses Penyelesaian........................................................................................................ 74*

*9.1.8 Flowchart Keseluruhan................................................................................................... 74*

9.2 STUDI KASUS 2: KONVERSI BILANGAN DESIMAL KE BINER .......................................75

*9.2.1 Permasalahan ................................................................................................................. 75*

*9.2.2 Cara Penyelesaian Masalah ........................................................................................... 75*

*9.2.3 Struktur Data Yang Dibutuhkan...................................................................................... 76*

*9.2.4 Deklarasi dan Inisialisasi................................................................................................ 76*

*9.2.5 Input ................................................................................................................................ 76*

*9.2.6 Output.............................................................................................................................. 76*

*9.2.7 Proses Penyelesaian........................................................................................................ 77*

*9.2.8 Flowchart Keseluruhan................................................................................................... 78*

9.3 LATIHAN ...................................................................................................................78

*9.3.1 Permasalahan ................................................................................................................. 78*

*9.3.2 Cara Penyelesaian Masalah ........................................................................................... 78*

**BAB 10 STUDI KASUS OPERASI MATRIKS..................................................................80**

10.1 STUDI KASUS: OPERASI PENAMBAHAN MATRIKS ......................................................80

*10.1.1 Permasalahan ............................................................................................................. 80*

*10.1.2 Cara Penyelesaian Masalah ....................................................................................... 80*

*10.1.3 Struktur Data Yang Dibutuhkan ................................................................................. 81*

*10.1.4 Deklarasi dan Inisialisasi ........................................................................................... 81*

*10.1.5 Input............................................................................................................................ 81*

*10.1.6 Output ......................................................................................................................... 82*

*10.1.7 Proses Penyelesaian ................................................................................................... 82*

*10.1.8 Flowchart Keseluruhan .............................................................................................. 83*

10.2 10.2. LATIHAN ..........................................................................................................83

*10.2.1 Permasalahan ............................................................................................................. 83*

*10.2.2 Cara Penyelesaian Masalah ....................................................................................... 84*

**BAB 11 STUDI KASUS SHORTEST PATH PROBLEM.................................................86**

11.1 STUDI KASUS: SHORTEST PATH PROBLEM DENGAN ALGORITMA DIJKSTRA .............86

*11.1.1 Permasalahan ............................................................................................................. 86*

*11.1.2 Cara Penyelesaian Masalah ....................................................................................... 86*

*11.1.3 Struktur Data Yang Dibutuhkan ................................................................................. 90*

*11.1.4 Deklarasi dan Inisialisasi ........................................................................................... 91*

*11.1.5 Input............................................................................................................................ 91*

*11.1.6 Output ......................................................................................................................... 92*

11.2 LATIHAN ...................................................................................................................92

**Daftar Gambar**

Gambar 1.1. Penukaran gelas isi gelas A dan gelas B. ..............................................................2

Gambar 1.2. Langkah-langkah penukaran gelas isi gelas A dan gelas B. .................................3

Gambar 1.3. Komponen-Komponen Utama Komputer.............................................................9

Gambar 2.1. Contoh bagian judul algoritma............................................................................15

Gambar 2.3. Contoh bagian deklarasi algoritma. ....................................................................16

Gambar 2.4. Contoh bagian deskripsi algoritma. ....................................................................16

Gambar 2.5. Contoh penulisan algoritma menggunakan kalimat deskriptif. ..........................17

Gambar 2.5. Contoh pseudocode menggunakan gaya penulisan Fortran, Pascal dan C. ........18

Gambar 2.7. Contoh pseudocode konversi nilai angka menjadi nilai huruf. ...........................19

Gambar 2.8. Flowchart luas dan keliling lingkaran.................................................................23

Gambar 2.9. Flowchart Penentuan Bilangan Ganjil-Genap.....................................................24

Gambar 3.1. Ilustrasi pembuatan konstanta atau variabel .......................................................31

Gambar 3.2. Ilustrasi penggunaan variabel..............................................................................32

Gambar 3.3. Contoh penggunaan variabel...............................................................................32

Gambar 3.4. Ilustrasi array.......................................................................................................32

Gambar 3.5. Cara mengakses elemen array.............................................................................33

Gambar 3.6. Contoh penggunaan array ...................................................................................34

Gambar 3.7. Ilustrasi array 2 dimensi (kiri) dan array 3 dimensi (kanan)...............................35

Gambar 3.8. Ilustrasi stack.......................................................................................................36

Gambar 3.9. Ilustrasi queue .....................................................................................................36

Gambar 3.10. Ilustrasi tree.......................................................................................................38

Gambar 3.11. Ilustrasi graph....................................................................................................39

Gambar 3.12. List adjancency .................................................................................................41

**Daftar Tabel**

Tabel 2.1. Perbandingan beberapa kata yang biasa digunakan dalam penulisan algoritma dengan menggunakan kalimat deskriptif dan pseudocode...............................................17

Tabel 2.2.Simbol-simbol Flowchart ........................................................................................21

Tabel 2.2.Simbol-simbol Flowchart (lanjutan) ........................................................................22

*Halaman ini sengaja dikosongkan*

viii

**Bab 1**

**Pengantar Logika dan Algoritma**

**Pokok Bahasan**

1. Konsep logika dan algoritma

2. Ciri-ciri algoritma

3. Konsep algoritma, program, dan bahasa pemrograman

**Tujuan Pembelajaran**

Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:

1. Mengenal dan memahami logika dan algoritma.

2. Mampu membuat contoh penyelesaian masalah dengan menggunakan konsep logika

**1.1. Pengertian Logika dan Algoritma**

Pengertian algoritma sangat lekat dengan kata logika, yaitu kemampuan seorang manusia untuk berfikir dengan akal tentang suatu permasalahan menghasilkan sebuah kebenaran, dibuktikan dan dapat diterima akal, logika seringkali dihubungkan dengan kecerdasan, seseorang yang mampu berlogika dengan baik sering orang menyebutnya sebagai pribadi yang cerdas. Dalam menyelesaikan suatu masalahpun logika mutlak diperlukan.

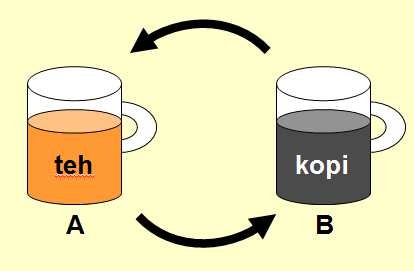
Logika identik dengan masuk akal dan penalaran. Penalaran adalah salah satu bentuk pemikiran. Pemikiran adalah pengetahuan tak langsung yang didasarkan pada pernyataan langsung pemikiran mungkin benar dan mungkin juga tak benar.Definisi logika sangat sederhana yaitu ilmu yang memberikan prinsip-prinsip yang harus diikuti agar dapat berfikir valid menurut aturan yang berlaku. Pelajaran logika menimbulkan kesadaran untuk menggunakan prinsip-prinsip untuk berfikir secara sistematis.

Logika berasal dari dari bahasa Yunani yaitu LOGOS yang berarti ilmu. Logika dapat diartikan ilmu yang mengajarkan cara berpikir untuk melakukan kegiatan dengan tujuan tertentu. Algoritma berasal dari nama seorang Ilmuwan Arab yang bernama Abu JafarMuhammad Ibnu Musa Al Khuwarizmi penulis buku berjudul Al Jabar Wal Muqabala.

Kata Al Khuwarizmi dibaca orang barat menjadi Algorism yang kemudian lambat laun menjadi Algorithm diserap dalam bahasa Indonesia menjadi Algoritma.Algoritma dapat diartikan urutan penyelesaian masalah yang disusun secara sistematis menggunakan bahasa yang logis untuk memecahkan suatu permasalahan.

Meski demikian terdapat beberapa definisi algoritma yang lain. Diantaranya menurut Rinaldi Munir, algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis. Sedang menurut Kamus Besar Bahasa Indonesia, definisi algoritma adalah urutan logis pengambilan keputusan untuk pemecahan masalah. Menurut tim Gunadarma:1988, algoritma adalah suatu himpunan berhingga dari instruksi-instruksi yang secara jelas memperinci langkah-langkah proses pelaksanaan, dalam pemecahan suatu masalah tertentu, atau suatu kelas masalah tertentu, dengan dituntut pula bahwa himpunan instruksi tersebut dapat dilaksanakan secara mekanik. Dari pengertian diatas maka dapat disimpulkan bahwa Logika danAlgoritmaadalah ilmu yang mempelajari cara penyelesaian suatu masalah berdasarkan urutan langkah-langkah terbatas yang disusun secara sistematis dan menggunakan bahasa yang logis dengan tujuan tertentu.

Untuk lebih mudah memahami arti dari algoritma dicontohkan sebuah permasalahan penukaran isi dari dua gelas. Diberikan dua buah gelas A dan B, gelasA berisi air teh dan gelasB berisi air kopi. Pertukarkan isi gelas tersebut sehingga menghasilkan gelasA yang semula berisi air teh menjadi berisi air kopi dan gelasB yang semula berisi air kopi menjadi berisi air teh. Ilustrasi permasalahan ini dapat dilihat pada Gambar 1.1.



Gambar 1.1. Penukaran gelas isi gelas A dan gelas B.

Cara penyelesaian permasalahan ini adalah sebagai berikut. Untuk mempertukarkan isi gelasdengan benar, maka diperlukan gelastambahan yang kita namakan gelasCsebagai tempat penampungan sementara.Berikut Algoritmanya:

1. Siapkan gelas cadangan C

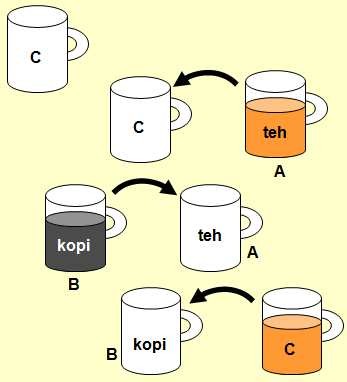
2. Tuangkan air teh dari gelas A ke dalam gelas C (gelas A menjadi kosong).

3. Tuangkan air kopi dari gelas B ke dalam gelas A (gelas B menjadi kosong).

4. Tuangkan air teh dari gelas C ke dalam gelas B.

Ilustrasi langkah-langkah algoritma dapat dilihat pada Gambar 1.2.

Gambar 1.2. Langkah-langkah penukaran gelas isi gelas A dan gelas B.



Dari contoh tersebut dapat dilihat bahwa penyelesaian permasalahan penukaran isi dua buah gelas sangat sederhana. Disini digunakan urutan langkah yang masuk akal atau logis sehingga isi dari kedua nya sudah berpindah media, dari A ke B dan B ke A. Inilah yang dinamakan “Algoritma”, urutan penyelesaian sebuah permasalahan dengan urutan dan langkah yang logis dan masuk akal menghasilkan sesuatu langkah yang benar.

Contoh lain penggunaan logika dan algoritma adalah membuat algoritma untuk menghitung luas lingkaran, caranya:

1. Menentukan nilai jari-jari (r) lingkaran.

2. Menentukan nilai phi.

3. Menghitung luas lingkaran dengan cara mengkalikan nilai jari-jari (r) dengan (r) lalu dikalikan dengan nilai phi.

4. Maka luas lingkaran ditemukan.

5. Selesai.

Saat menggunakan logika, sebaiknya jangan berfikir terlalu rumit tentang sebuah masalah, karena belum tentu masalah itu serumit yang kita pikir. Pikirkan hal yang paling sederhana untuk menyelesaikan masalah itu, sehingga tidak terjebak dalam pikiran rumit yang dibuat sendiri. Meski demikian jangan meremehkan masalah sekecil apapun, tapi berfikir sederhana untuk menghasilkan solusi yang efektif

Dalam menentukan algoritma untuk menyelesaikan suatu permasalahan, mungkin kita dihadapkan oleh beberapa pilihan algoritma. Oleh karena itu kita harus memiliki rambu- rambu dalam menentukan pilihan algoritma. Pertimbangan dalam pemilihan algoritmaadalah, pertama, algoritma haruslah benar. Artinya algoritma akan memberikan keluaran susuai seperti yang diharapkan dari sejumlah masukan yang diberikan. Tidak

peduli sebagus apapun algoritma, jika memberikan keluaran yang salah, maka sudah pasti algoritma tersebut bukanlah algoritma yang baik. Pertimbangan kedua yang harus diperhatikan adalah kita harus mengetahuiseberapa baik hasil yang dicapai oleh algoritma tersebut. Hal ini pentingterutama pada algoritma yang memerlukan aproksimasi hasil yaitu algoritma yang hasilnya hanya berupa pendekatan. Algoritma yangbaik harus mampu memberikan hasil yang sedekat mungkin dengan nilaiyang sebenarnya.Ketiga adalah efisiensi algoritma. Efisiensi algoritma dapat ditinjau dari dua hal yaitu efisiensi waktu dan memori. Meskipun algoritma memberikan keluaran yang benar atau paling mendekati, tetapi jika kita harus menunggu lama untuk mendapatkan hasil semisal berjam-jam untuk mendapatkan keluarannya maka biasanya algoritma tersebut biasanyatidak akan menjadi pilihan utama, setiap orang menginginkan keluaran yang relatif cepat. Begitujuga dengan memori, semakin besar memori yang terpakai maka semakin jelek algoritma tersebut.Dalam kenyataannya, setiap orang bisa membuat algoritma yang berbeda untuk menyelesaikan suatu permasalahan, walaupun terjadi perbedaandalam menyusun algoritma, tentunya kita mengharapkan keluaran yang mirip atau sama. Jika dihadapkan pada permasalahan seperti ini maka sebaiknya pilih algoritma yang paling efisien dan cepat.

Tujuan dari belajar logika dan algoritma adalah agar dapat membiasakan diri melakukan suatu perencanaan apabila menyelesaikan suatu masalah.Karena suatu permasalahan yang diselesaikan dengan suatu perencanaan yang matang maka akan mendapatkan solusi yang lebih optimal dibandingkan menyelesaikan masalah tanpa menggunakan suatu perencanaan.

**1.2. Ciri-ciri, Sifat, Struktur Dasar dan Cara Penulisan Algoritma**

Tidak semua urutan langkah penyelesaian masalah yang logis dapat disebut sebagai algoritma. Menurut Donald E. Knuth, algoritma mempunyai lima ciri penting yang meliputi:

1. Finiteness (keterbatasan), algoritma harus berakhir setelah mengerjakan sejumlah langkah proses.

2. Definiteness (kepastian), setiap langkah harus didefinisikan secara tepat dan tidak berarti ganda.

3. Input (masukan), algoritma memiliki nol atau lebih data masukan (input).

4. Output (keluaran), algoritma mempunyai nol atau lebih hasil keluaran (output).

5. Effectiveness (efektivitas), algoritma harus sangkil (efektif), langkah-langkah algoritma dikerjakan dalam waktu yang wajar.

Sedang sifat algoritma adalah:

1. Tidak menggunakan simbol atau sintaks dari suatu bahasa pemrograman tertentu.

2. Tidak tergantung pada suatu bahasa pemrograman tertentu.

3. Notasi-notasinya dapat digunakan untuk seluruh bahasa manapun.

4. Algoritma dapat digunakan untuk merepresentasikan suatu urutan kejadian secara logis dan dapat diterapkan di semua kejadian sehari-hari

Seperti telah dijelaskan di sub bab sebelumnya bahwa penyusun atau struktur dasar algoritma adalah langkah-langkah. Suatu Algoritma dapat terdiri dari tiga struktur dasar, yaitu runtunan, pemilihan dan pengulangan. Ketiga jenis langkah tersebut membentuk konstruksi suatu algoritma. Berikut adalah penjelasan dari tiga struktur tersebut :

**1.1.1 Runtunan (sequence)**

Sebuah runtunan terdiri dari satu atau lebih instruksi. Tiap instruksi dikerjakan secara berurutan sesuai dengan urutan penulisannya, yakni sebuah instruksi dilaksanakan setelah instruksi sebelumnya selesai dikerjakan. Urutan dari instruksi menentukan hasil akhir dari suatu algoritma. Bila urutan penulisan berubah maka mungkin juga hasil akhirnya berubah. Sebagai contoh perhatikan operasi aritmatika berikut ini, (4+3)\*7=49, tetapi bila urutan aksinya diubah maka hasil keluaran akan berbeda menjadi 4+(3\*7) =25.

Contoh lain dari runtunan aksi adalah algoritma penukaran dua bilangan bulat,yaitu:

1. Deklarasikan A, B, dan C sebagai bilangan bulat

2. Masukkan nilai A dan B

3. Masukkan nilai A ke dalam C

4. Masukkan nilai B ke dalam A

5. Masukkan nilai C ke dalam B

**1.1.2 Pemilihan (selection)**

Kadangkala terdapat suatu kejadian yang baru akan dikerjakan jika suatu kondisi tertentu telah terpenuhi. Pemilihan yaitu instruksi yang dikerjakan dengan kondisi tertentu. Kondisi adalah persyaratan yang dapat bernilai benar atau salah. Satu atau beberapa instruksi hanya dilaksanakan apabila kondisi bernilai benar, sebaliknya apabila salah maka instruksi tidak akan dilaksanakan. Contoh kasus pemilihan adalah dalam penentuan bilangan genap atau ganjil berikut ini:

1. Masukkan **bilangan** sebagai sebuah bilangan bulat

2. Bagi **bilangan** dengan angka 2, simpan nilai sisa pembagian dalam variabel **sisa**

3. Jika nilai **sisa** sama dengan 0 maka kerjakan langkah 4:

4. Tampilkan ”GENAP” ke layar

5. Jika nilai **sisa** tidak sama dengan 0 maka kerjakan langkah 6

6. Tampilkan ”GANJIL” ke layar

7. Selesai.

**1.1.3 Pengulangan (repetition)**

Salah satu kelebihan komputer adalah kemampuannya untuk mengerjakan pekerjaan yang sama berulang kali tanpa mengenal lelah.Kita tidak perlu menulis instruksi yang sama berulang kali, tetapi cukup melakukan pengulangan dengan instruksi yang tersedia. Pengulangan merupakan kegiatan mengerjakan sebuah atau sejumlah aksi yang sama sebanyak jumlah yang ditentukan atau sesuai dengan kondisi yang diinginkan. Beberapa statemen pengulangan di bahasa pemrograman yaitu for…, while()..., do...while(), repeat....until, for...downto...do, for...to...do dan lain-lain. Sebagai contoh adalah menampilkan huruf tertentu sebanyak n kali ke layar sebagai berikut:

1. Deklarasikan variabel **huruf** untuk menyimpan karakter yang akan ditampilkan.

2. Deklarasikan variabel **n** untuk menyimpan banyaknya perulangan

3. Deklarasikan variabel **counter** yang digunakan sebagai counter perulangan yang sudah dilakukan.

4. Masukkan sebuah karakter dan simpan dalam variabel **huruf**

5. Masukkan banyaknya perulangan yang diinginkan dan simpan dalam variabel **n**

6. Set nilai **counter** dengan 0

7. Tampilkan **huruf** ke layar

8. Lakukan penambahan **counter** dengan 1

9. Jika nilai **counter**<**n**, kerjakan langkah 6

10. Jika **nilai counter = n** selesai

**1.3. Perbedaan Algoritma dan Program**

Sebagaimana telah diuraikan di sub bab sebelumnya bahwa algoritma adalah urutan langkah-langkah terbatas yang disusun secara sistematis dan menggunakan bahasa yang logis dengan tujuan menyelesaikan suatu masalah tertentu. Sementara program adalah kumpulan instruksi berupa pernyataan yang ditulis dengan menggunakan bahasa pemrograman yang melibatkan pemilihan struktur data. Beberapa pakar komputer menyatakan program dengan formula Program = Algoritma + Bahasa Pemrograman.

Bahasa pemrograman dan algoritma berhubungan sangat eratpada sebuah program. Algoritma yang baik tanpa pemilihan struktur datayang tepat akan membuat program menjadi kurang baik, demikian jugasebaliknya. Jika dihubungkan dengan program, maka pembuatan algoritma harus memperhatikan kaidah:

1. Pembuatan atau penulisan algoritma tidak tergantung pada bahasapemrograman manapun, artinya penulisan algoritma independen daribahasa pemrograman dan komputer yang memprosesnya.

2. Notasi algoritma dapat diterjemahkan ke dalam berbagai bahasapemrograman.

3. Apapun bahasa pemrogramannya, output yang akan dikeluarkan samakarena algoritmanya sama.

Algoritma dibuat untuk membantu kita dalam mengkonversikan suatu permasalahan ke dalam bahasa pemrograman.Algoritma merupakan hasil pemikiran konseptual, supaya dapatdilaksanakan oleh komputer, algoritma harus diterjemahkan ke dalamnotasi bahasa pemrograman. Ada beberapa hal yang harus diperhatikanpada penerjemahan tersebut, yaitu:

1. Pendeklarasian variabel

Variabel dibutuhkan oleh program dalam rangka menyimpan data masukan, memproses dan mendapatkan hasil komputasi.

2. Pemilihan tipe data

Apabila dalam proses pembuatan program ternyata membutuhkanpendeklarasian variabel maka diwajibkam memilih tipe data, karena setiap variabel pasti membutuhkan tipe data ketika dideklarasikan.

3. Pemakaian atau pemilihan instruksi

Terdapat beberapa macam instruksi dalam bahasa pemrograman (sequence, selection dan repetition), urutan langkah dalam algoritma dapat diterjemahkan menjadi salah satu atau beberapa instruksi tersebut.

4. Aturan sintaksis

Pada saat menuliskan program kita terikat dengan aturan sintaksis dalam bahasa pemrograman yang akan digunakan. Setiap bahasa pemrograman memiliki aturan penulisan sintaks-nya sendiri.

5. Tampilan hasil

Pada saat membuat algoritma kita tidak memikirkan tampilan hasil yang akan disajikan. Halteknis semacam ini diperhatikan ketikamengkonversikannya menjadi program.

6. Cara pengoperasian compiler atau interpreter.

Bahasa pemrograman yang digunakan termasuk dalam kelompok compiler atau interpreter. Masing-masing memiliki cara pengoperasian yang bisa jadi berbeda.

Komputer adalah mesin pemroses. Agar dapat dikerjakan oleh komputer, algoritma harus ditulis dalam notasi bahasa pemrograman berupa instruksi yang dapat dipahami oleh komputer sehingga dinamakan program. Jadi program adalah perwujudan atau implementasi teknis algoritma yang ditulis dalam bahasa pemrograman tertentu sehingga dapat dilaksanakan oleh komputer. Program ditulis dengan menggunakan salah satu bahasa pemrograman. Kegiatanmembuat program disebut pemrograman (programming). Orang yangmenulis program disebut pemrogram (programmer). Tiap-tiap langkah didalam program disebut pernyataan atau instruksi. Jadi, program tersusunatas sederetan instruksi. Bila suatu instruksi dilaksanakan, maka operasi-operasiyang bersesuaian dengan instruksi tersebut dikerjakan komputer.

**1.4. Mekanisme Pelaksanaan Algoritma oleh Pemroses Notasi Penulisan**

**Algoritma**

Komputer sebagai mesin pemroses terdiri dari empat komponen utama, yaitu unit input, unit output, unit pemroses sentral (CPU: Central Processing Unit), dan memori seperti digambarkan pada Gambar 1.3. CPU adalahperangkat keraskomputer yang memahami dan melaksanakan perintah dan data dari perangkat lunak. Istilah lain, pemroses/prosesor

(processor) sering digunakan untuk menyebut CPU. CPU berperan sebagai otak komputer, yang berfungsi mengerjakan operasi-operasi aritmatika seperti kalkulator, hanya saja CPU jauh lebih kuat dan lebih besar daya pemrosesannya. Fungsi utama dari CPU adalah melakukan operasi aritmatika dan logika terhadap data yang diambil dari memori atau dari informasi yang dimasukkan melalui unit input berupa beberapa perangkat keras, seperti papan tombol, pemindai, tuas kontrol, maupun tetikus. CPU dikontrol menggunakan sekumpulan instruksi perangkat lunakkomputer. Perangkat lunak tersebut dapat dijalankan oleh CPU dengan membacanya dari media penyimpan, seperti cakram keras, disket, cakram padat, maupun pita perekam. Instruksi-instruksi tersebut kemudian disimpan terlebih dahulu pada memori fisik, yang mana setiap instruksi akan diberi alamat unik yang disebut alamat memori. Selanjutnya, CPU dapat mengakses data-data pada memori fisikdengan menentukan alamat data yang dikehendaki.Selama proses ini terjadi, sebuah unit dalam CPU yang disebut dengan penghitung program akan memantau instruksi yang sukses dijalankan supaya instruksi tersebut dapat dieksekusi dengan urutan yang benar dan sesuai sehingga didapatkan hasil yang dapat ditampilkan pada unit output, misalnya dengan menampilkannya di layar

monitor.

Memori

Unit Input

CPU

Unit Output

Gambar 1.3. Komponen-Komponen Utama Komputer

Seperti yang telah diketahui bahwa komputer memerlukan instruksi berupa langkah- langkah perintah sehingga sebuah prosedur dapat dijalankan. Prosedur yang berisi langkah- langkah penyelesaian masalah inilah yang disebut dengan algoritma. Jadi, sebelum memasuki tahap pemrograman komputer dengan menggunakan bahasa pemrograman,sebaiknya mempelajari algoritma yang merupakan pengantar kepada pemrograman komputer tersebut.

Dalam ilmu komputer,algoritma dikenal dengan langkah-langkah komputasi yang terdiri dari masukan dan keluaran. Karena itu Algoritma biasanya dijadikan dasar atau

pengantar bahasa pemrograman dalam studi yang berkecimpung atau berhubungan dengan ilmu komputer, misalnya Teknik Informatika.

**1.5. Belajar Membuat Program dan Belajar Bahasa Pemrograman**

Belajar membuat program berbeda dengan belajar bahasa pemrograman. Belajar membuat program adalah belajar tentang strategi atau cara menyelesaikan suatu masalah, yang diungkapkan dalam bentuk algoritma yang mudah dibaca dan dipahami yang selanjutnya dituangkan dalam bahasa pemrograman. Belajar memprogrambersifat pemahaman persoalan, analisis dansintesis. Titik berat belajar membuat program adalah desain program. Sedangkan belajar bahasa pemrograman adalah belajar mengenai tata cara atau tata aturan penulisan pernyataan atau statement untuk membuat program dalam bahasa tertentu saja. Tata aturan ini biasa disebut juga dengan sintaks bahasa. Sintaks bahasa ini meliputi cara mendeklarasikan variabel, cara melakukan perulangan, percabangan dan lain- lain. Tiap bahasa pemrograman memiliki aturan penulisan sendiri-sendiri walaupun terkadang ada kemiripan aturan antar bahasa. Titik berat belajar bahasa pemrograman adalah coding program.

Sampai saat ini terdapat puluhanbahasa pemrogram, antara lain bahasa Assembly, Fortran, Cobol, PL/I, Algol, Pascal, C, C++, Basic, Prolog, LISP, PRG, CSMP, Simscript, GPSS, Dinamo, dan lain-lain. Berdasarkanterapannya, bahasa pemrograman dapat digolongkan atas dua kelompokbesar:

1. Bahasa pemrograman bertujuan khusus. Yang termasuk kelompokini adalah Cobol untuk terapan bisnis dan administrasi. Fortran untuk terapan komputasi ilmiah, bahasa Assembly untuk terapan pemrogramanmesin, Prolog untuk terapan kecerdasan buatan, bahasa-bahasasimulasi, dan sebagainya.

2. Bahasa pemrograman bertujuan umum, yang dapat digunakan untukberbagai aplikasi.

Yang termasuk kelompok ini adalah bahasa Pascal, Basic, Java, dan C. Tentu saja pembagian ini tidak kaku. Bahasabertujuan khusus tidak berarti tidak bisa digunakan untukaplikasi lain. Cobol misalnya, dapat juga digunakan untuk terapanilmiah, hanya saja kemampuannya terbatas. Yang jelas, bahasapemrograman yang berbeda dikembangkan untukbermacam-macam terapan yang berbeda pula.

Berdasarkan pada tingkat kerumitan sebuah bahasa pemrograman, apakah notasi bahasa pemrograman lebih mendekati bahasa mesin atau ke bahasa manusia, maka bahasa

pemrograman dikelompokkan atas dua macam, yaitu bahasa tingkat tinggi dan bahasa tingkat rendah. Istilah "bahasa pemrograman tingkat tinggi" tidak serta merta menjadikan bahasa tersebut lebih baik dibandingkan dengan bahasa pemrograman tingkat rendah. Akan tetapi, maksud dari "tingkat tinggi" di sini merujuk kepada abstraksi yang lebih tinggi dibandingkan dengan bahasa tingkat rendah terhadap bahasa mesin. Dibandingkan dengan harus berurusan dengan register, alamat memori dan stack-stack panggilan, bahasa pemrograman tingkat tinggi akan berurusan dengan variabel, larik, dan ekspresi aritmetika atau aljabar boolean. Selain itu, tidak seperti bahasa rakitan, bahasa tingkat tinggi tidak memiliki kode operasi yang dapat secara langsung menjadikan bahasa tersebut menjadi kode mesin. Fitur lainnya seperti fungsi-fungsi penanganan string, fitur pemrograman berorientasi objek, input/output terhadap berkas juga terdapat di dalam jenis bahasa ini.

Secara umum, bahasa tingkat tinggi akan membuat pemrograman komputer yang kompleks menjadi lebih sederhana, sementara bahasa tingkat rendah cenderung untuk membuat kode yang lebih efisien. Dalam sebuah bahasa tingkat tinggi, elemen-elemen kompleks dapat dipecah ke dalam beberapa elemen yang lebih sederhana, meski masih dapat dianggap kompleks, di mana bahasa tersebut menyediakan abstraksi. Karena alasan ini, kode- kode yang harus berjalan dengan efisien dapat ditulis dalam bahasa pemrograman tingkat rendah, sementara bahasa tingkat tinggi digunakan untuk mempermudah pemrograman.

Bahasa tingkat rendah dirancang agar setiapinstruksinya langsung dikerjakan oleh komputer, tanpa harusmelalui penerjemah (translator). Contohnya adalah bahasa mesin.CPU mengambil instruksi dari memori, langsung mengerti danlangsung mengerjakan operasinya. Bahasa tingkat rendah bersifatprimitif, sangat sederhana, orientasinya lebih dekat ke mesin, dansulit dipahami manusia. Bahasa Assembly dimasukkan kedalam kelompok ini karena alasan notasi yang dipakai dalam bahasaini lebih dekat ke mesin, meskipun untuk melaksanakaninstruksinya masih perlu penerjemahan ke dalam bahasa mesin.

Bahasa tingkat tinggi membuat pemrograman lebih mudahdipahami, lebih “manusiawi”, dan berorientasi ke bahasa manusia(bahasa Inggris). Hanya saja, program dalam bahasa tingkat tinggitidak dapat langsung dilaksanakan oleh komputer. Ia perluditerjemahkan terlebih dahulu oleh sebuah translator bahasa, yangdisebut kompilator atau compiler, ke dalam bahasa mesin sebelumakhirnya dieksekusi oleh CPU. Contoh bahasa tingkat tinggi adalahPascal, PL/I, Ada, Cobol, Basic, Fortran, C, C++, dan sebagainya.

Dengan bertambah rumitnya arsitektur mikroprosesor modern, kompilator-kompilator bahasa pemrograman tingkat tinggi dapat membuat kode yang lebih efisien dibandingkan dengan para programmer bahasa pemrograman tingkat rendah yang melakukannya secara manual.Perlu dicatat bahwa istilah "tingkat tinggi" dan "tingkat rendah" adalah relatif. Pada awalnya, bahasa Assembly dianggap sebagai bahasa tingkat rendah, sementara COBOL, C, dan lain-lainnya dianggap sebagai bahasa tingkat tinggi, mengingat mereka mengizinkan abstraksi terhadap fungsi, variabel, dan evaluasi ekspresi. Akan tetapi, banyak programmer saat ini mungkin menganggap bahasa C sebagai bahasa pemrograman tingkat rendah, mengingat bahasa pemrograman tersebut mengizinkan akses memori secara langsung dengan menggunakan alamatnya, dan juga dapat menggunakan beberapa direktif bahasa Assembly.

**1.6. LatihanSoal**

1. Apakah yang dimaksud dengan algoritma?

2. Apa perbedaan antara algoritma dan program?

3. Suatu algoritma terdiri dari tiga struktur dasar, yaitu runtunan, pemilihan, dan perulangan.

Jelaskan masing-masing!

4. Apa perbedaan antara program dan bahasa pemrograman?

5. Buatlah algoritma menulis dan mengirimkan surat!

6. Buatlah algoritma mengambil uang di ATM!

7. Buatlah algoritma membuat kopi yang rasa manis dan pahitnya pas menurut anda!

8. Buatlah algoritma untuk menghitung luas segitiga!

9. Buatlahalgoritma untuk proses aktivitas dari pagi sampai malam!

10. Buatlah algoritma mengurutkan 3 bilangan acak!