**Mata Kuliah : TIK (Teknologi Informasi dan Komunikasi) Semester : Ganjil**

**Dosen Pembina : Triasasnova Kelas : XI**

|  |  |
| --- | --- |
| **Topik : logika, algoritma, dan pemograman** | **Waktu : 2 x 45 Menit** |

1. **TUJUAN PEMBELAJARAN**
2. Kognitif
3. Produk

* Siswa mampu menyebutkan definisi logika dengan benar.
* Siswa mampu menyebutkan definisi algoritma dengan benar.
* Siswa mampu menyebutkan definisi pemograman dengan benar.
* Siswa mampu menyelesaikan kasus dengan logika, algoritma, dan pemrograman dengan benar.

1. Proses

* Siswa mendefinisikan logika di depan kelas dengan tepat.
* Siswa mendefinisikan algoritma di depan kelas dengan tepat.
* Siswa mendefinisikan pemrograman di depan kelas dengan tepat.
* Siswa menyelesaian kasus dengan logika, algoritma, dan pemrograman dengan tepat.

1. Psikomotor

Siswa mampu menyelesaikan kasus dengan menerapan logika, algorima, dan pemrograman yang tepat di depan kelas.

1. Afektif

Siswa terbiasa menyelesaikan kasus dengan menggunakan logika dan algoritma dalam kehidupan sehari-hari.

1. **TEORI PEMBELAJARAN**
   * + 1. **Logika**

Logika merupakan sesuatu yang rasional. Logika berasal dari kata [Yunani kuno](http://id.wikipedia.org/wiki/Yunani_kuno) λόγος (*logos*) yang berarti hasil pertimbangan akal pikiran yang diutarakan lewat kata dan dinyatakan dalam bahasa. Sebagai ilmu, logika disebut dengan logike episteme (Latin: *logica scientia*) atau ilmu logika (ilmu pengetahuan) yang mempelajari kecakapan untuk berpikir secara lurus, tepat, dan teratur. Ilmu di sini mengacu pada kemampuan rasional untuk mengetahui dan kecakapan mengacu pada kesanggupan akal budi untuk mewujudkan pengetahuan ke dalam tindakan. Jadi logika adalah hasil pertimbangan akal pikiran yang diutarakan lewat kata dan dinyatakan dalam bahasa.

* + - 1. **Algoritma**

Algoritma adalah urutan logis langkah-langkah penyelesaian masalah. Sehingga program komputer merupakan realisasi teknis dari sebuah algoritma. Pengertian algoritma menurut (Webster dictionary) :

1. Langkah-langkah yang dilakukan agar solusi masalah dapat diperoleh.
2. Suatu prosedur yang merupakan urutan langkah-langkah yang berintegrasi
3. Suatu motode khusus yang digunakan untuk menyelesaikan suatu masalah yang nyata

Berikut kriteria pemilihan algoritma

1. Ada output

Mengacu pada definisi algoritma, algoritma harus mempunyai out put yang harus merupakan solusi dari masalah yang sedang diselesaikan.

1. Efektifitas dan efisiensi

Dikatakan efektif jika algoritma tersebut menghasilkan suatu solusi yang sesuai dengan masalah yang diselesaikan, dalam arti algoritma harus tepat guna.  
Dikatakan efisien jika waktu proses suatu algoritma relatif lebih singkat dan penggunakan memori komputernya lebih sedikit.

1. Jumlah langkahnya berhingga

Barisan instruksi yang dibuat harus dalam suatu urutan tertentu atau harus berhingga agar masalah yang dihadapi dapat diselesaikan dengan tidak memerlukan waktu relatif lama.

1. Terstruktur

Penyelesaian masalah menggunakan langkah-langkah tersusun.

Hubungan logika dengan algoritma, logika merupakan langkah awal atau kemampuan manusia merasionalkan masalah, sedangkan algoritma adalah bentuk susunan atau langkah-langkah atau perealisasian dari hasil berfikir rasionla tersebut.

* + - 1. **Pemograman**

Dalam pelajaran pemrograman, kita lebih memikirkan pada cara menyelesaikan masalah yang akan dipogram dengan menekankan pada desain atau rancangan yang mewakili pemecahan masalah tersebut. Desain ini dibuat sedemikian rupa sehingga independen dari bahasa pemrograman yang kelak digunakan dan komputer yang akan menjalankan program tersebut. Desain berisi langkah-langkah pencapaian solusi yang ditulis dalam notasi-notasi deskriptif/ notasi algoritmik. Setiap orang yang berbeda mungkin menghasilkan rancangan program yang berbeda pula. Karena belajar memprogram bukan belajar membuat program yang asal jadi, perlu difikirkan membuat program dengan menggunakan skema yang benar.

Bila desain program telah dibuat dengan skema yang benar, maka desain tersebut siap dikodekan dengan notasi bahasa pemrograman agar program bisa dieksekusi oleh komputer. Disinilah perlunya kita belajar bahasa pemrograman. Ada banyak bahasa pemrograman yang tersedia, namun desain program dapat diterjemahkan ke bahasa apapun. Sampai saat ini terdapat puluhan bahasa pemrograman, antara lain bahasa assembly, fotran, cobal, ada, algol, pascal, c++, basic, prolog, lips, prg, dan java.

* + 1. **Notasi algortimik**

Pada penjelasan sebelumnya telah dinyatakan bahwa notasi algoritmik dibuat independen dai spesifikasi bahasa pemrogram dan komputer yang mengeksekusinya. Notasi algoritmik ini dapat diterjemahkan ke dalam berbagai bahasa pemrogram. Analoginya sama dengan resep membuat kue. Sebuah resep dapat ditulis dalam bahasa manapun, bahasa ingris, prancis, indonesia, jepang, dan sebagainnya. Apapun bahasanya kue yang dihasilkan tetap sama, sebab algoritmanya sama (dengan catatan semua aturan pada resep tersebut diikuti). Mengapa demikian? Karena setiap juru masak (yang merupakan pemroses) mampu melakukan operasi dasar yang sam, seperti mengocok telur, menimbang berat gula, dan sebagainya. Jadi resep membuat kue tidak terikat pada bahasa dan juru masak yang mengerjakannya.

Demikiannya pula halnya komputer. Meskipun setiap komputer berbeda teknologinya, tetapi secara umum semua komputer dapat melakukan operasi-operasi dasar dalam pemrograman seperti operasi pembacaan data, operasi perbandingan, operasi aritmatika, dan sebagainya. Perkembangan tekonologi komputer tidak mengubah operasi-operasi dasar itu, yang berubah hanyalah kecepatan, biaya, atau tingkat ketelitian. Pada sisi lain, setiap program dalam bahasa tingkat tinggi selalu diterjemahkan kedalam bahasa mesin sebelum akhirnya dikerjakan oleh CPU. Setiap instruksi dalam bahasa mesin menyajikan operasi dasar yang sesuai, dan menghasilkan efek yang sama pada setiap komputer.

Yang perlu dicatat bahwa notasi algoritmik bukan notasi pemrograman sehingga siapapun dapat membuat notasi algoritmik yang berbeda. Hal yang penting mengenai notasi tersebut adalah ia mudah dibaca dan dimengerti. Selain itu, meskipun notasi algoritmik bukan notasi baku sebagaimana pada notasi bahasa pemrogram, namun ketaataasasan terhadap notasi perlu diperhatikan untuk menghindari kekeliruan.

Berikut ini beberapa jenis notasi :

1. Notasi 1 : menyatakan langkah-langkah algoritma dengan untaian kalimat deskriptif. Contoh :

Diberikan dua buah bilangan bulat tak negatif m dan n (m ≥ n). algoritma mencari pembagi bersama terbesar yang habis membagi m dan n???

Jawab :

* Jika n = 0 maka

M adalah jawabannya;

Stop

Tetapi jika n ≠ 0

Lanjutkan ke langkah 2

* Bagilah m dengan n, dan misalkan r adalah sisanya
* Ganti nilai m dengan nilai n, dan nilai n dengan nilai r, lalu ulang kembali ke langkah 1, hingga jawaban ditemukan.

**Dengan angka**, misal m = 7 dan n = 3

Jika n = 0, pembagi terbesarnya adalah m = 7 --- > tidak berlaku krn n ≠ 0

Jika n ≠ 0 maka m/n (7/3) sisa 1

M = n ---- > m = 3 dan n = r --- > n = 1

**Pengulang pertama**

Jika n = 0, pembagi terbesarnya adalah m = 3 --- > tidak berlaku krn n ≠ 0

Jika n ≠ 0 maka m/n (3/1) sisa 0

M = n ---- > m = 1 dan n = r --- > n = 0

**Pengulang kedua**

Jika n = 0, pembagi terbesarnya adalah m = 1 --- > berlaku krn n = 0

**Tidak ada pengulangan lagi karena jawabannya telah ditemukan yaitu 1**

1. Notasi 2 : menggunakan diagram alir (*flow chart)*

Dari soal nya sama :

Mulai

Selesai

Baca m dan n

Tulis m

r = m mod n

m = n

n = r

n = 0

Tidak

1. Notasi 3 : menggunakan *pseudo-code*

*Pseudo-code* artinya semu/ tidak sebenarnya merupakan notasi yang menyerupai bahasa pemrograman tingkat tinggi, khususnya pascal dan C. yang membedakanya adalah tidak menggunakan tanda titik koma, indeks, format keluaran, kata-kata khusus, dan sebagainnya. Sembarang versi *pseudo-code*  dapat diterima asalkan tidak membingungkan pembaca. Kelebihan *pseudo-code* adalah mudah menterjemahkan ke dalam bahasa pemograman.

Penyelesaian dengan *pseudo-code*  contoh soal yang sama :

Deklarasi :

m, n : integer (bilangan bulat yang akan dicari pembagi terbesarnya)

r : integer (sisa hasil bagi)

read (m,n)

while n ≠ 0

r < -- m mod n (hitung sisa hasil pembagian)

m < -- n

n < -- r

endwhile

(kondisi selesai, pengulangan hingga n = 0 maka pembagi terbesar adalah m)

writeln (m)

* + 1. **Kasus penyelesaian dengan logika, algoritma, dan pemrograman**

Logiak merupakan cara berfikir logis, Algoritma berisi langkah-langkah penyelesaian masalah, dan pemrograman merupakan rancangan penyelesaiannya sehingga realisasinya adalah algoritma/ langkah-langkah. Langkah-langkah tersebut dapat berupa runtutan aksi, pemilihan aksi, dan pengulangan aksi. Ketiga jenis langkah tersebut membentuk kostruksi suatu algoritma. Jadi algoritma dibangun dalam tiga buah struktur dasar, yaitu :

1. Runtutan

Sebuah runtutan terdiri dari satu atau lebih pernyataan. Tiap pernyataan dikerjakan secara berurutan sesuai dengan urutan penulisannya, yakni sebuah instruksi dilaksanakan setelah instruksi sebelumnya selesai dilaksanakan. Urutan isntruksi menentuka keadaan akhir algoritma. Bila urutannya diubah, maka hasil akhirnya mungkin juga berubah. Contoh : algoritma pertukaran air dari dua biah ember.

Diberikan dua buah ember, A dan B. A berisi air berwarna merah; B berisi air berwarna biru. Pertukarkanlah isi kedua ember ini sehingga ember A berisi warna biru dan ember B berisi warna merah.

Jawabannya.

Kita tambahkan satu ember lagi yaitu ember C.

* Tuangkan air dari ember A ke ember C
* Tuangkan air dari ember B ke ember A
* Tuangkan air dari ember C ke ember B

Maka ember A akan berisi air warna biru dan ember B berisi air warna merah.

1. Pemilihan

Pemilihan merupakan suatu keadaan dikerjakan jika kondisi tertentu dipenuhi. Misalnya kendaraan akan berhenti jika lampu merah menyala dan terus berjalan jika lampu hijau menyala.

Sintaknya :

if kondisi then

Aksi 1

If air dalam ketel mendidik then

Matikan api kompor

atau

if kondisi then

Aksi 1

Else

Aksi 2

If air dalam ketel mendidik then

Matikan api kompor

Else

Biarkan api kompor menyala

Semakin banyak pilihan kondisi yang terjadi maka pilihan else akan semakin bertambah.

If kondisi 1 then

Aksi 1

Else

If kondisi 2 then

Aksi 2

Else

Aksi 3

If lampu merah menyala then

Berhenti

Else

If lampu hijau menyala then

Jalan hati-hati

Else

Jalan terus

1. Pengulangan

Salah satu kelebihan komputer adalah kemampuannya untuk mengerjakan pekerjaan yang sama berulang kali tanpa mengenal lelah.

Sintaknya :

for pencacah pengulangan dari 1 sampai n do

aksi

menuliskan kalimat “saya berjanji tidak nakal dan malas lagi“ sebanyak 500 kali.

Jawabannya :

For i dari 1 sampai 500 do

Tulis “saya bejanji tidak akan nakal dan malas lagi”

1. **LATIHAN DAN TUGAS**
2. **Latihan**
3. Ada dua buah kotak yang berisi angka, kotak pertama berisi angka 100, kotak kedua berisi angka 50. Buatlah algoritma sehingga kotak pertama berisi angka 50 dan kotak kedua berisi angka 100!
4. Diketahui sebuah bilangan x, jika x habis dibagi 2 dan kelipatannya maka x adalah bilangan genab, jika x hanya habis dibagi oleh dirinya sendiri maka x adalah bilangan prima, jika tidak keduanya maka x adalah bilangan ganjil. Buatlah algoritma untuk kondisi diatas!
5. Seorang anak bernama Andi mendapatkan tugas dari guru piket untuk menuliskan kata yaitu “Saya Rajin” sebanyak 1000 buah karena ia sering terlambat datang ke sekolah. Buatlah algoritma yang dapat memudahkan tugas Andi!
6. **Penugasan**
7. Ada dua buah bungkusan makanan. Bungkusan pertama berisi goreng pisang, dan bungkusan kedua berisi buah mangga. Buatlah algoritma sehingga bungkusan pertama berisi buah mangga dan bungkusan kedua berisi goreng pisang! Bobot 30
8. Seorang anak, bernama susi pulang sekolah. Jika ia berjalan kaki maka ia sampai dirumah jam 2 siang, jika ia naik sepeda maka ia akan tiba dirumah jam 3 siang, tetapi jika ia tidak melakukan apapun maka ia akan menginap disekolah. Buatlah algoritma yang menyatakan kondisi diatas! Bobot 40
9. Disuatu sekolah, tepatnya kelas 1 sekolah dasar, seorang guru sedang mengajar huruf vokal “AIUEO” pada muridnya, agar murid-murid tersebut cepat mengingat huruf vokal itu, dia menugaskan pada siswanya untuk menuliskan huruf vokal tersebut sebanyak 500 kali. Buatlah algoritma yang akan membantu murid-murid tersebut! Bobot 30