

# 1. Algoritma

Dilihat dari istilahnya algoritma, berasal dari nama seorang matematikawan muslim bernama Abu Ja'far Muhammad Ibnu Musa Al-Khuwarizmi (780 M-850), yang oleh orang barat menyebut Al- Khuwarizm sebagai *Algorism*, yang diartikan proses menghitung dengan angka arab. Salah satu karya beliau yang monumental adalah buku berjudul *Kitab Al Jabar Wal-Muqabala* yang artinya "Buku pemugaran dan pengurangan" (*The book of restoration and reduction*), yang menjadi cikal bakal istilah "Aljabar" (*Algebra*) yang dipakai hingga sekarang ini. Seiring dengan perkembangan jaman istilah "Algorism" berubah menjadi *algorithm* yang kemudian diartikan sebagai metode perhitungan (komputasi) secara umum. Dalam bahasa Indonesia, kata *algorithm* diserap menjadi *algoritma*.

Algoritma yang berbeda dapat diterapkan pada suatu masalah dengan syarat yang sama. Tingkat

kerumitan dari suatu algoritma merupakan ukuran seberapa banyak komputasi yang dibutuhkan algoritma tersebut untuk menyelesaikan masalah. Umumnya, algoritma yang dapat menyelesaikan suatu permasalahan dalam waktu yang singkat memiliki tingkat kerumitan yang rendah, sementara algoritma yang membutuhkan waktu lama untuk menyelesaikan suatu masalah membutuhkan tingkat kerumitan yang tinggi. Perhatikan algoritma sederhana berikut.

Contoh: ***Algoritma menghitung luas segitiga.***

- Start
- Baca data alas dan tinggi.
- Luas adalah alas kali tinggi kali 0.5
- Tampilkan Luas
- Stop

Algoritma di atas adalah algoritma yang sangat sederhana, hanya ada lima langkah. Pada algoritma ini tidak dijumpai perulangan ataupun

pemilihan. Semua langkah dilakukan hanya satu kali. Sekilas algoritma di atas benar, namun apabila dicermati maka algoritma ini mengandung kesalahan yang mendasar, yaitu tidak ada pembatasan pada nilai data untuk alas dan tinggi. Bagaimana jika nilai data alas atau tinggi adalah bilangan 0 atau bilangan negatif? Tentunya hasil yang keluar menjadi tidak sesuai dengan yang diharapkan. Dalam kasus seperti ini kita perlu menambahkan langkah untuk memastikan nilai alas dan tinggi memenuhi syarat, misalnya dengan melakukan pengecekan pada input yang masuk. Apabila input nilai alas dan tinggi kurang dari 0 maka program tidak akan dijalankan. Sehingga algoritma di atas dapat dirubah menjadi seperti contoh berikut.

Contoh: Hasil perbaikan algoritma perhitungan luas segitiga.

- Start
- Baca data alas dan tinggi.

- Periksa data alas dan tinggi, jika nilai data alas dan tinggi lebih besar dari nol maka lanjutkan ke langkah ke 4 jika tidak maka stop
- Luas adalah alas kali tinggi kali 0.5
- Tampilkan Luas
- Stop

Dari penjelasan di atas dapat diambil kesimpulan pokok tentang algoritma. Pertama, algoritma harus benar. Kedua algoritma harus berhenti, dan setelah berhenti, algoritma memberikan hasil yang benar.

## **2. Cara Penulisan Algoritma**

Ada tiga cara penulisan algoritma, yaitu:

### ***a. Structured English (SE)***

SE merupakan alat yang cukup baik untuk menggambarkan suatu algoritma. Dasar dari SE adalah Bahasa Inggris, namun kita dapat memodifikasi dengan Bahasa Indonesia sehingga kita boleh menyebut nya sebagai Structured

Indonesian (SI). Algoritma seperti pada Contoh 5.10 dan 5.11 merupakan algoritma yang ditulis menggunakan SI. Karena dasarnya adalah bahasa sehari-hari, maka SE atau SI lebih tepat untuk menggambarkan suatu algoritma yang akan dikomunikasikan kepada pemakai perangkat lunak.

### ***b. Pseudocode***

*Pseudocode* mirip dengan SE. Karena kemiripan ini kadang-kadang SE dan *Pseudocode* dianggap sama. *Pseudo* berarti imitasi atau tiruan atau menyerupai, sedangkan *code* menunjuk pada kode program. Sehingga *pseudocode* adalah kode yang mirip dengan instruksi kode program sebenarnya. *Pseudocode* didasarkan pada bahasa pemrograman yang sesungguhnya seperti *BASIC*, *FORTRAN* atau *PASCAL*. *Pseudocode* yang berbasis bahasa *PASCAL* merupakan *pseudocode* yang sering digunakan. Kadang-kadang orang menyebut

pseudocode sebagai *PASCAL-LIKE* algoritma Apabila Contoh 5.10 ditulis dalam pseudocode berbasis bahasa *BASIC* akan tampak seperti pada contoh 5.12.

*Contoh 5.12. Pseudocode.*

- Start
- READ alas, tinggi
- $\text{Luas} = 0.5 * \text{alas} * \text{tinggi}$
- PRINT Luas
- Stop










Pada Contoh 5.12 tampak bahwa algoritma sudah sangat mirip dengan bahasa BASIC. Pernyataan seperti *READ* dan *PRINT* merupakan *keyword* yang ada pada bahasa BASIC yang masing-masing menggantikan kata “baca data” dan “tampilkan”. Dengan menggunakan pseudocode seperti di atas

maka proses penerjemahan dari algoritma ke kode program menjadi lebih mudah.

### **c. Flowchart**

Flowchart atau bagan alir adalah skema/bagan (chart) yang menunjukkan aliran (flow) di dalam suatu program secara logika. Flowchart merupakan alat yang banyak digunakan untuk menggambarkan algoritma dalam bentuk notasi-notasi tertentu. Secara lebih detail bagian ini akan dibahas pada bagian berikutnya. Pada flowchart ada beberapa simbol penting yang digunakan untuk membuat algoritma sebagaimana tercantum pada Gambar 5.3.

## Simbol Flowchart :

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Pemulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inisialisasi/pemberian harga awal
	PROSES	Proses perhitungan/proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	PREDEFINED PROCESS (SUB PROGRAM)	Pemulaan sub program/proses menjalankan sub program
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	ON PAGE CONNECTOR	Penghubung bagian-bagian flow chart yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flow chart yang berada pada halaman berbeda

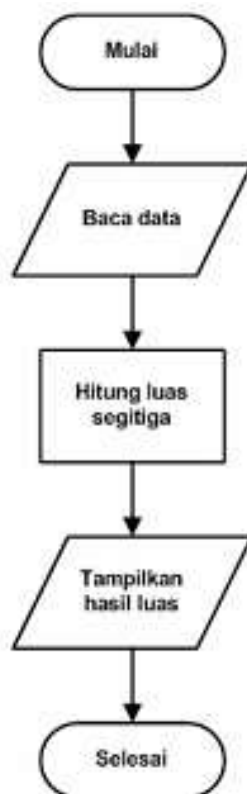
*Gambar Simbol-simbol yang digunakan dalam flowchart.*

Program Flowchart dapat terdiri dari dua macam, yaitu bagan alir logika program ( program logic flowchart ) dan bagan alir program komputer terinci (detailed computer program flowchart). Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara

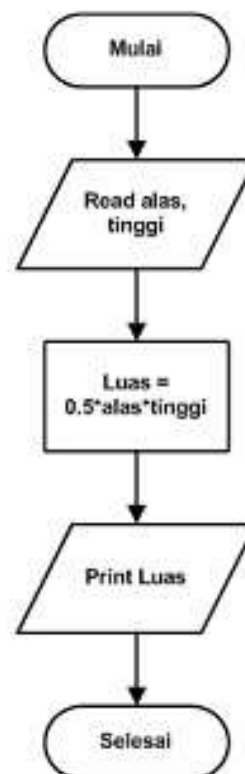


logika dan biasanya dipersiapkan oleh seorang analis system. Sedangkan bagan alir program komputer terinci digunakan untuk menggambarkan instruksi-instruksi program komputer secara terinci dan biasanya dipersiapkan oleh seorang programmer.

Apabila Contoh dibuat program flowchartnya maka akan tampak pada gambar



Bagan alir logika program



Bagan alir program komputer terinci

### 3. Struktur Algoritma Berurutan



Ada tiga struktur dasar yang digunakan dalam membuat algoritma yaitu struktur berurutan (sequencing), struktur pemilihan / keputusan / percabangan (branching) dan struktur pengulangan. Sebuah algoritma biasanya akan menggabungkan ketiga buah struktur ini untuk menyelesaikan masalah.

Pada bagian ini kita akan bahas lebih dulu struktur algoritma berurutan. Struktur berurutan dapat kita samakan dengan mobil yang sedang berjalan pada

jalur lurus yang tidak terdapat persimpangan. Mobil tersebut akan melewati kilometer demi kilometer jalan sampai tujuan tercapai.

Struktur berurutan terdiri satu atau lebih instruksi. Tiap instruksi dikerjakan secara berurutan sesuai dengan urutan penulisannya, yaitu sebuah instruksi dieksekusi setelah instruksi sebelumnya selesai dieksekusi. Urutan instruksi menentukan keadaan akhir dari algoritma. Bila urutannya diubah, maka hasil akhirnya mungkin juga berubah.

Menurut Goldshlager dan Lister (1988) struktur berurutan mengikuti ketentuan-ketentuan sebagai berikut:

- tiap instruksi dikerjakan satu persatu
- tiap instruksi dilaksanakan tepat sekali, tidak ada yang diulang
- urutan instruksi yang dilaksanakan pemroses sama dengan urutan aksi sebagaimana yang tertulis di dalam algoritmanya

- akhir dari instruksi terakhir merupakan akhir algoritma.

***Contoh 5.13. Flowchart untuk menghitung luas bangun.***

Buatlah flowchart untuk menghitung:

- volume balok
- luas lingkaran

*Penyelesaian:*

Soal ini merupakan permasalahan dengan algoritma struktur berurutan karena tidak ada proses pemilihan atau pengulangan. Untuk volume balok, kita harus menentukan variabel input dan output yang dibutuhkan. Untuk menghitung volume balok dibutuhkan variabel input panjang, lebar dan tinggi. Sedangkan variabel outputnya adalah volume. Pada luas lingkaran dibutuhkan variabel input radius dan variabel output luas. Untuk

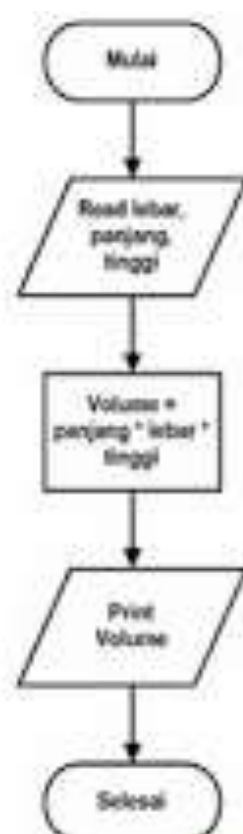
menghitung luas lingkaran ini kita juga membutuhkan konstanta phi.

***Contoh 5.14. Flowchart untuk konversi suhu.***

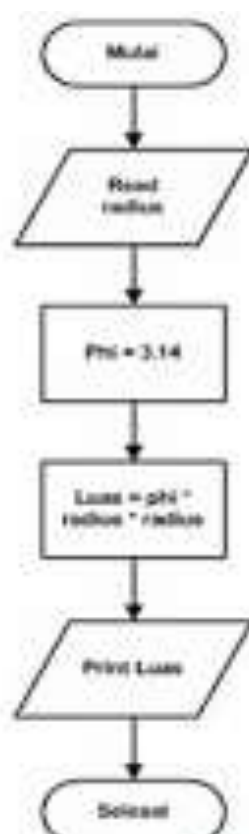
Buat flowchart untuk mengubah temperatur dalam Fahrenheit menjadi temperatur dalam Celcius dengan rumus  $C = 5/9 \times (F - 32)$ .

*Penyelesaian:*

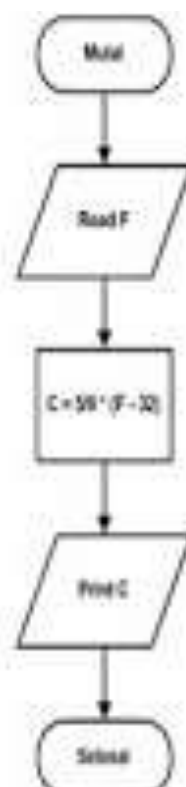
Soal ini juga masih menggunakan algoritma dengan struktur berurutan. Variabel input yang dibutuhkan adalah F dan variabel outputnya adalah C. Flowchart untuk dua masalah ini dapat dilihat pada Gambar 5.7.



a. flowchart menghitung volume balok



b. flowchart menghitung luas lingkaran



Gambar 5.7. Flowchart untuk konversi suhu.

Gambar 5.6. Flowchart menghitung volume balok dan luas lingkaran.

## 4. Struktur Algoritma Percabangan



Sebuah program tidak selamanya akan berjalan dengan mengikuti struktur berurutan, kadang-kadang kita perlu merubah urutan pelaksanaan program dan menghendaki agar pelaksanaan program meloncat ke baris tertentu. Peristiwa ini kadang disebut sebagai percabangan/pemilihan atau keputusan. Hal ini seperti halnya ketika mobil berada dalam persimpangan seperti pada Gambar 5.7. Pengemudi harus memutuskan apakah harus menempuh jalur yang kanan atau yang kiri. Pada struktur percabangan, program akan berpindah urutan

pelaksanaan jika suatu kondisi yang disyaratkan dipenuhi. Pada proses seperti ini simbol flowchart Decision harus digunakan. Simbol decision akan berisi pernyataan yang akan diuji kebenarannya. Nilai hasil pengujian akan menentukan cabang mana yang akan ditempuh.

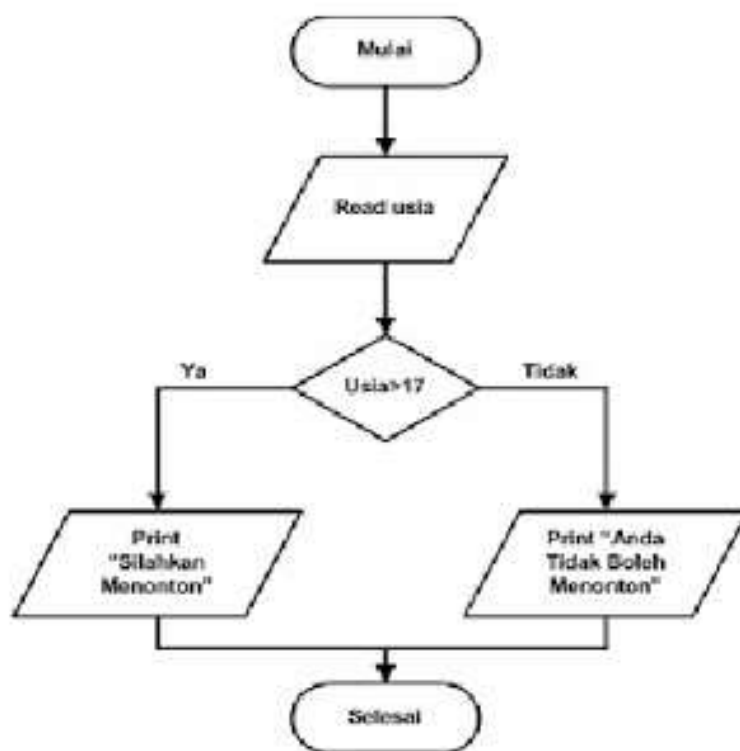
Struktur percabangan untuk masalah batasan umur. Sebuah aturan untuk menonton sebuah film tertentu adalah sebagai berikut, jika usia penonton lebih dari 17 tahun maka penonton diperbolehkan dan apabila kurang dari 17 tahun maka penonton tidak diperbolehkan nonton. Buatlah flowchart untuk permasalahan tersebut.

*Penyelesaian:*

Permasalahan diatas merupakan ciri permasalahan yang menggunakan struktur percabangan. Hal ini ditandai dengan adanya pernyataan jika .. maka ...(atau If ... Then dalam



Bahasa Inggris. Flowchart penyelesaian masalah tampak pada Gambar 5.9. Pada gambar tersebut, tampak penggunaan simbol Decision. Pada simbol ini terjadi pemeriksaan kondisi, yaitu apakah usia lebih dari 17 tahun atau tidak. Jika jawaban ya maka program akan menghasilkan keluaran teks “Silahkan Menonton”, sedangkan jika input usia kurang dari 17 tahun maka program akan menghasilkan keluaran teks “Anda Tidak Boleh Menonton”.

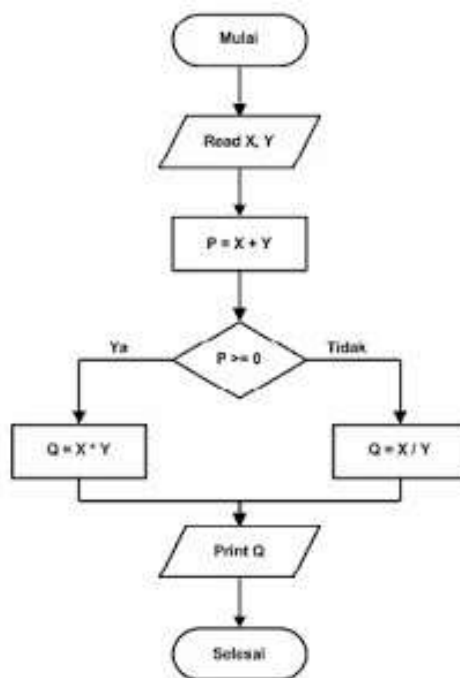


*Contoh 5.16. Struktur percabangan untuk perhitungan dua buah bilangan.*

Dalam suatu perhitungan nilai  $P = X + Y$ . Jika  $P$  positif, maka  $Q = X * Y$ , sedangkan jika negative maka nilai  $Q = X/Y$ . Buatlah flowchart untuk mencari nilai  $P$  dan  $Q$

*Penyelesaian:*

Pada contoh ini input yang dibutuhkan adalah nilai  $X$  dan  $Y$ , sedangkan proses pemeriksaan kondisi dilakukan pada nilai  $P$  apakah positif (termasuk 0) ataukah negative. Perhatikan flowchart penyelesaian masalah pada Gambar 5.10.



Kedua contoh di atas (5.15 dan 5.16) merupakan contoh struktur percabangan sederhana yang

melibatkan hanya satu percabangan. Pada masalah-masalah yang lebih rumit, kita akan menjumpai lebih banyak percabangan. Kita juga akan menjumpai suatu struktur percabangan berada di dalam struktur percabangan yang lain, atau yang biasa disebut nested (bersarang).

*Perhatikan contoh-contoh berikut.*

Sebuah usaha fotokopi mempunyai aturan sebagai berikut:

jika yang fotokopi statusnya adalah langganan, maka berapa lembar pun dia fotokopi, harga perlembarannya Rp. 75,- jika yang fotokopi bukan langganan, maka jika dia fotokopi kurang dari 100 lembar harga perlembarannya Rp. 100,-. Sedangkan jika lebih atau sama dengan 100 lembar maka harga perlembarannya Rp. 85,-.

Buat flowchart untuk menghitung total harga yang harus dibayar jika seseorang memfotokopi sejumlah X lembar.

*Penyelesaian:*

Pada contoh ini, masalah terlihat lebih rumit. Ada dua percabangan yang terjadi. Yang pertama adalah pemeriksaan apakah status seseorang pelanggan atau bukan. Kedua, apabila status seseorang bukan pelanggan, maka dilakukan pemeriksaan berapa jumlah lembar fotokopi, apakah lebih dari 100 lembar atau tidak.

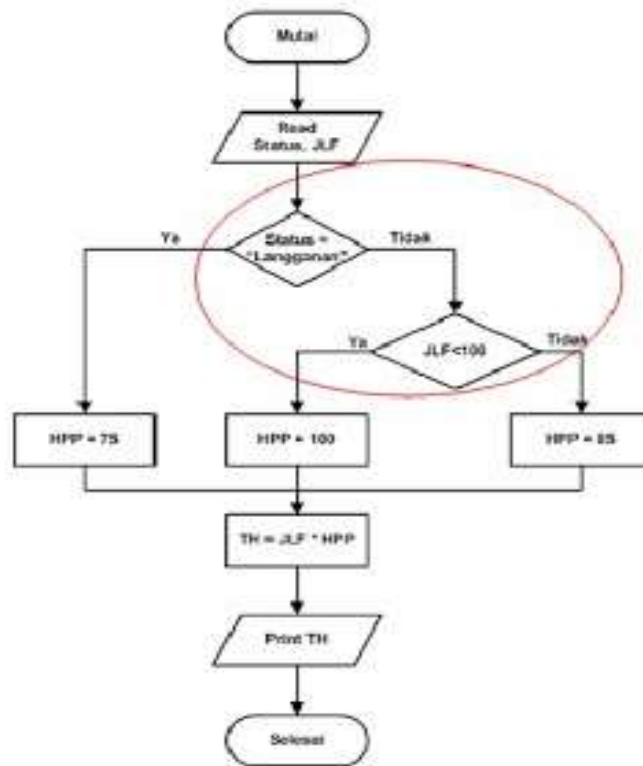
Pada soal ini kita juga menjumpai apa yang disebut sebagai nested. Perhatikan pernyataan pada syarat kedua dari persoalan di atas. Jika yang fotokopi bukan langganan, maka jika dia fotokopi kurang dari 100 lembar harga perlembaranya Rp. 100 pernyataan jika yang kedua berada di dalam jika yang pertama. Input

yang dibutuhkan untuk permasalahan ini adalah status orang yang fotokopi dan jumlah lembar yang difotokopi. Sehingga variable input yang digunakan adalah:

- Status untuk status orang yang fotokopi
- JLF untuk jumlah lembar yang difotokopi

Selain itu terdapat variable dengan nama HPP yang digunakan untuk menyimpan harga per lembar dan TH untuk menyimpan nilai total harga. Perhatikan, variable Status bertipe data char, sehingga penulisannya harus menggunakan tanda “ ”.

*Flowchart penyelesaian masalah ini dapat dilihat pada Gambar 5.11.*



Contoh 5.18. Struktur percabangan bersarang untuk masalah kelulusan siswa.

Aturan kelulusan siswa pada mata pelajaran Pemrograman Web diterapkan sebagai berikut:

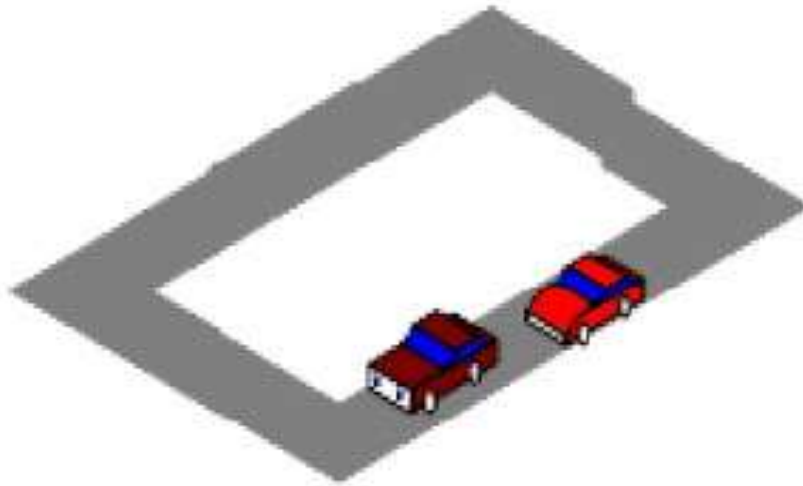
Jika nilai ujian tengah semester (UTS) lebih besar dari 70 maka siswa dinyatakan lulus dan Nilai Akhir sama dengan nilai UTS. Jika nilai UTS kurang atau sama dengan 70 maka siswa dinyatakan lulus jika Nilai Akhir lebih besar atau sama dengan 60 dimana Nilai Akhir = (nilai UTS x 40%) + (nilai UAS x 60%).

Buatlah flowchart penyelesaian masalah tersebut apabila output yang diinginkan adalah NIM, Nama Siswa, Nilai Akhir dan Status Kelulusan.

*Penyelesaian:*

Pada contoh ini, ada dua percabangan. Yang pertama adalah pemeriksaan apakah nilai UTS siswa lebih dari 70. Kedua, apabila nilai UTS tidak lebih dari 70, maka dilakukan pemeriksaan apakah nilai akhir lebih dari 60. Input yang dibutuhkan untuk permasalahan ini adalah NIM, nama siswa, nilai UTS, dan nilai UAS. Sehingga variable input yang digunakan adalah: NIM untuk Nomor induk siswa, nama untuk nama siswa, NUTS untuk nilai ujian tengah semester, dan NUAS untuk nilai ujian akhir semester. Sedangkan variabel output terdiri dari NA yang digunakan untuk menyimpan nilai akhir dan Status untuk menyimpan status kelulusan.

## 5. Struktur Algoritma Pengulangan



Dalam banyak kasus seringkali kita dihadapkan pada sejumlah pekerjaan yang harus diulang berkali. Salah satu contoh yang gampang kita jumpai adalah balapan mobil seperti tampak pada gambar 5.13. Mobil-mobil peserta harus mengelilingi lintasan sirkuit berkali-kali sesuai yang ditetapkan dalam aturan lomba. Siapa yang mencapai garis akhir paling cepat, dialah yang menang. Pada pembuatan program komputer, kita juga kadang-kadang harus mengulang satu atau sekelompok perintah berkali-kali agar memperoleh hasil yang diinginkan. Dengan menggunakan komputer, eksekusi pengulangan



mudah dilakukan. Hal ini karena salah satu kelebihan komputer dibandingkan dengan manusia adalah kemampuannya untuk mengerjakan tugas atau suatu instruksi berulang kali tanpa merasa lelah, bosan, atau malas. Bandingkan dengan pengendara mobil balap, suatu ketika pasti dia merasa lelah dan bosan untuk berputar-putar mengendarai mobil balapnya.

*Struktur pengulangan terdiri dari dua bagian :*

Kondisi pengulangan, yaitu syarat yang harus dipenuhi untuk melaksanakan pengulangan. Syarat ini biasanya dinyatakan dalam ekspresi Boolean yang harus diuji apakah bernilai benar (true) atau salah (false) Badan pengulangan (loop body), yaitu satu atau lebih instruksi yang akan diulang.

Pada struktur pengulangan, biasanya juga disertai bagian inisialisasi dan bagian terminasi. Inisialisasi adalah instruksi yang dilakukan sebelum pengulangan dilakukan pertama kali. Bagian

inisialisasi umumnya digunakan untuk memberi nilai awal sebuah variable. Sedangkan terminasi adalah instruksi yang dilakukan setelah pengulangan selesai dilaksanakan. Ada beberapa bentuk pengulangan yang dapat digunakan, masing-masing dengan syarat dan karakteristik tersendiri. Beberapa bentuk dapat dipakai untuk kasus yang sama, namun ada bentuk yang hanya cocok untuk kasus tertentu saja. Pemilihan bentuk pengulangan untuk masalah tertentu dapat mempengaruhi kebenaran algoritma. Pemilihan bentuk pengulangan yang tepat bergantung pada masalah yang akan diprogram.

Struktur pengulangan dengan For. Pengulangan dengan menggunakan For, merupakan salah teknik pengulangan yang paling tua dalam bahasa pemrograman. Hampir semua bahasa pemrograman menyediakan metode ini, meskipun sintaksnya mungkin berbeda. Pada struktur For kita harus tahu terlebih dahulu seberapa banyak badan loop akan

diulang. Struktur ini menggunakan sebuah variable yang biasa disebut sebagai loop's counter, yang nilainya akan naik atau turun selama proses pengulangan.