

**PRAKTIKUM PEMROGRAMAN KOMPUTER
MINGGU 2**

A. Tujuan praktikum:

Mahasiswa mampu membuat algoritma untuk menyelesaikan masalah

B. Dasar Teori

B.1. Algoritma

Algoritma adalah langkah-langkah terstruktur yang digunakan untuk menyelesaikan suatu masalah. Algoritma bisa dinyatakan dalam bentuk : pseudocode atau flowchart.

Algoritma untuk menyelesaikan masalah harus memiliki :

1. Awal algoritma
2. Masukan data
3. Proses mengelola data yang telah dimasukkan
4. Keluaran dari hasil proses
5. Berhenti dari proses.

Algoritma yang dibuat harus bersifat :

1. Setiap langkah jelas dapat dikerjakan
2. Setiap langkah tidak ambigu
3. Rumus/ proses perhitungan dapat dikerjakan

Pseudocode

Langkah-langkah disajikan dalam bentuk kalimat. Dapat berupa bahasa sehari-hari, dapat berupa bahasa yang menyerupai program.

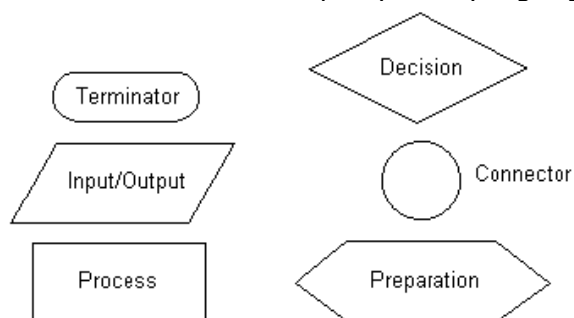
Contoh pseudocode :

1. Mulai
2. Masukkan panjang => p
3. Masukkan lebar => l
4. Hitung luas = $p \times l$
5. Tampilkan luas.
6. Selesai

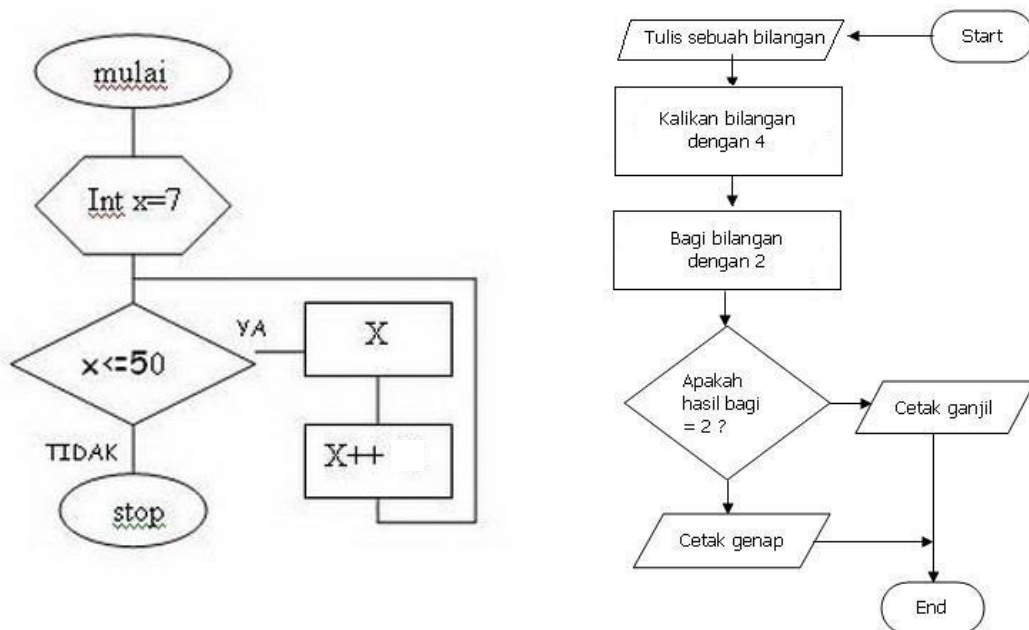
Flowchart

Algoritma dinyatakan dalam bentuk diagram.

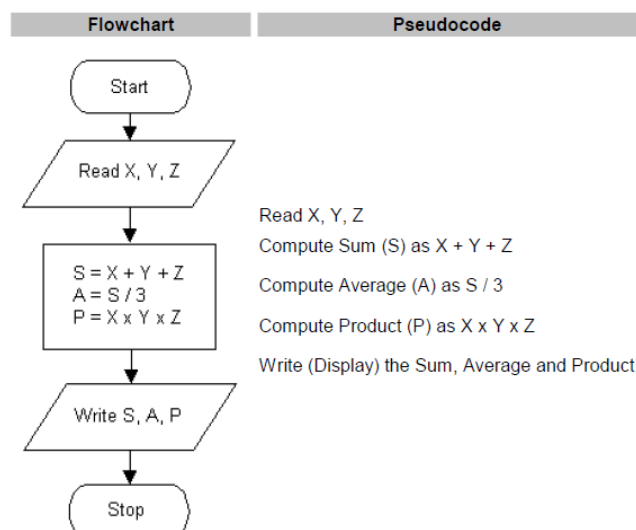
Berikut ini adalah beberapa symbol yang digunakan dalam flowchart.



Contoh flowchart :



Dibawah ini contoh pseudocode dan flowchart untuk menghitung jumlah, rata-rata dan perkalian 3 bilangan.



B.2. Struktur program bahasa C

Program dalam bahasa C terdiri dari :

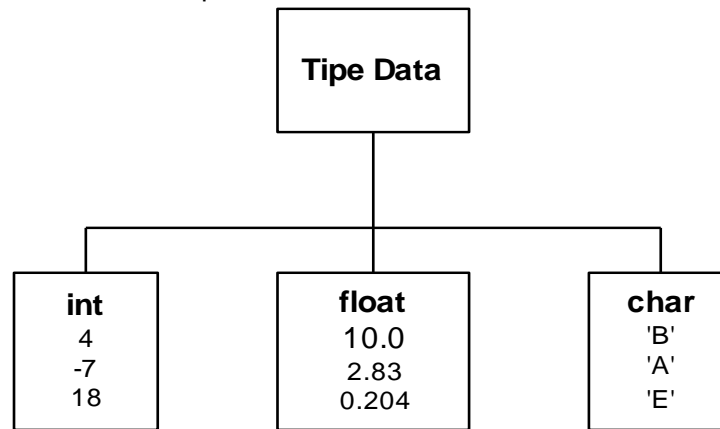
1. Header file
Berisi *preprocessor directive* yang akan memanggil file program bahasa C yang mempersiapkan program sehingga program kita dapat menjalankan perintah-perintah. Misal untuk menjalankan perintah `cout` kita membutuhkan `iostream.h`
2. Function `main()`
Setiap bahasa C harus memiliki `main()` karena program akan mulai dijalankan dari function ini.
3. Contoh program sederhana untuk mencetak "Bahasa C" ke layar.

```
#include <iostream.h>
void main()
{
    cout << "Bahasa C\n";
}
```

B.3. Tipe Data

Dalam bahasa C kita juga mengenal pengelompokan data menurut jenis / tipe seperti di atas. Data berupa bilangan bulat, sks misalnya, dikelompokkan dalam tipe **int**, dan data yang berupa

bilangan pecahan (misalnya, IP) disebut sebagai tipe **float**. Sedangkan data yang berbentuk karakter/huruf dikelompokkan dalam tipe **char**.



Gambar 4.1 Tipe Data Bahasa C

Ketiga tipe data ini – **int**, **float**, dan **char** – merupakan tipe data yang paling dasar dan paling umum digunakan. Tabel berikut menjelaskan beberapa contoh data dalam berbagai tipe data.

Tipe data	Deskripsi	Contoh
int	Semua bilangan bulat positif, negatif dan nol	-36 23 0 6547
float	Bilangan-bilangan yang mengandung pecahan (titik desimal) atau eksponen	-73.62 50.0 2.3000000000E02
char	Semua karakter seperti huruf, angka, tanda-tanda baca, atau spasi	'H' 'd' '5' '%' '!' ' '

Tabel 4.1 Beberapa contoh data dan tipenya.

Tipe Data Int (Bilangan Bulat)

Tipe data int mencakup semua bilangan positif, negatif dan nol yang tidak berisi bagian pecahan atau titik desimal.

Contoh:

-37 3454 0 -100

Karena keterbatasan memory komputer, tidak semua bilangan bulat dapat ditampung dalam memory. Bahasa C menyediakan tipe data lain yang merupakan perluasan dari tipe data int ini.

Beberapa varian tipe data integer:

Tipe	Batas	Ukuran (byte)
int	-32768 s/d 32767	2
long int	-2147483648 s/d 2147483648	4
unsigned int	0 s/d 65535	2

Dalam C, data bertipe **int** paling besar yang dapat disimpan adalah 32767 dan yang paling kecil adalah -32768. Bilangan-bilangan yang lebih kecil dari -32768 dan lebih besar dari 32767 tidak dapat disajikan. Operasi-operasi yang menghasilkan bilangan terlalu besar atau terlalu kecil (melampaui batas kapasitas penyimpanan) sehingga tidak dapat disajikan oleh komputer mengakibatkan keadaan *overflow* atau *underflow*.

Contoh program / operasi yang mengakibatkan *overflow*:

```

void main()
{
    int A, B, C;

    A = 10;
    B = 4000;
    C = A * B;          /* terjadi overflow */
}
  
```

Program 4.1.

Dalam contoh di atas, operasi $A * B$ menghasilkan nilai 40000 (empat puluh ribu). Tetapi karena variabel C yang menampung hasil operasi tersebut bertipe `int` dan nilai yang harus disimpan berada di luar batas tipe data `int` (yaitu lebih dari 32767), maka hasil tersebut tidak dapat disimpan di variabel C . Program pada contoh di atas dapat diperbaiki dengan mendeklarasikan variabel C sebagai tipe `long int` (tipe data integer yang mempunyai batas lebih besar) seperti berikut:

```
void main()
{
    int A, B, C;

    A = 10;
    B = 4000;
    C = A * B;           /* terjadi overflow */
}
```

Program 4.2.

Tipe Data Float (Bilangan Pecahan)

Tipe data **float** mencakup bilangan-bilangan yang dapat dinyatakan dengan titik desimal atau pecahan.

Contoh:

-36.9 3452 0.56 2.34358476E+02

Tipe data **float** mempunyai batas yang lebih besar dari pada tipe data integer. Dalam bahasa C, kita dapat menyatakan bilangan real / pecahan menggunakan **notasi titik kambang** (*floating-point notation*) dengan menggunakan huruf *e* atau *E* untuk menggantikan kalimat "kali 10 pangkat". Jadi bilangan $5.3e+02$ dapat dibaca $5,3 \times 10^2$ (lima koma tiga kali sepuluh pangkat dua). Seperti dalam tipe **int**, data dalam kelompok tipe **float** memiliki rentang nilai yang kecil (berukuran 32 bit). Tipe bilangan pecahan yang memiliki rentang nilai lebih besar dari pada **float** adalah tipe **double** (berukuran 64 bit).

Tipe Data Char

Tipe data `char` (baca "karakter") merupakan tipe data yang berupa huruf, angka, atau karakter lain. Dalam bahasa C, data karakter ditulis dengan diapit oleh apostrof (tanda kutip tunggal).

Contoh data tipe `char`:

'D' 'd' '%' '9' ' ' '\$'

Dalam contoh di atas, karakter 'D' berbeda dengan karakter 'd', demikian juga karakter '9' tidak sama dengan angka / bilangan 9. Karakter ' ' menyatakan karakter spasi. Dalam memory komputer, data karakter disimpan dalam bentuk bilangan bulat (sesuai kode ASCII-nya) dengan nilai antara 0 s/d 255. Misalnya huruf 'A' mempunyai nilai 65, sedangkan huruf 'a' mempunyai nilai 97.

PENGENAL

Kata atau barisan karakter / huruf yang membentuk nama suatu obyek disebut pengenalan atau *identifier*. Pada contoh di atas, kata/huruf seperti *phi*, *r* dan *L* serta kata-kata seperti *main* dan *printf* merupakan pengenalan atau *identifier*. Kata *phi* merupakan pengenalan yang dipakai sebagai nama konstanta; *r* dan *L* dipakai sebagai nama variabel, sedangkan *main* dan *printf* merupakan nama fungsi.

Aturan pemberian nama pengenalan dalam C:

1. Nama pengenalan **dimulai** dengan huruf besar A s/d Z, huruf kecil a s/d z dan dapat diikuti dengan huruf A sampai dengan Z, angka 0 sampai dengan 9 atau karakter underscore (garis bawah) "_".

2. Bahasa C bersifat case-sensitive artinya : penulisan dengan **huruf besar atau huruf kecil** dianggap **berbeda**.
3. Jumlah / panjang nama yang dikenali hanya 32 karakter / huruf (tergantung kompilernya).
4. Tidak boleh mengandung spasi.
5. Tidak boleh menggunakan *reserved word* (kata-kata cadangan) yaitu kata atau nama yang sudah digunakan oleh kompiler C seperti:
int double char float asm else new this auto enum operator throw bool explicit private true break export
protected try case extern public typedef catch false register typeid reinterpret_cast typename class for return
union const friend short unsigned

Contoh pemberian nama pengenalan yang **benar**:

Jarijari	Jari2	L	Harga_Total
R	Jari_jari	Rata2	Luas

Contoh pemberian nama pengenalan yang **tidak benar**:

Harga Total	(alasan: menggunakan spasi)
Jari-jari	(alasan: menggunakan tanda garis / strip tengah)
2bulan	(alasan: diawali dengan angka)
main	(alasan: merupakan kata cadangan atau <i>reserved word</i>)

B.4. Variabel, konstanta dan ekspresi

Konstanta akan berisi nilai yang tidak pernah berubah, sementara variabel akan berisi nilai yang berubah-ubah.

Contoh :

I=1; I => variabel, 1=> konstanta.

Untuk menuliskan variabel, menggunakan aturan pengenalan di atas.

Contoh penamaan variabel yang benar :

NIM, a, x, nama_mhs, f3098, f4, nilai, budi

Contoh penamaan variabel yang salah :

%nilai_mahasiswa, 80mahasiswa, rata-rata, ada spasi, penting!

Deklarasi Variabel

Variabel adalah obyek-obyek yang nilainya dapat berubah selama program dijalankan. Semua variabel yang akan digunakan dalam program harus dideklarasikan terlebih dahulu pada bagian deklarasi variabel. Setiap variabel dalam C mempunyai 3 komponen, yaitu nama, tipe dan nilai. Untuk mendeklarasikan variabel kita harus menentukan nama dan tipe dari variabel yang dimaksud. Nama variabel disusun dengan mengikuti aturan penamaan pengenalan.

Aturan deklarasi Variabel:

Nama_tipe daftar_variabel;

Contoh:

```
int r;
int alas;
int tinggi;
float luas;
```

Deklarasi di atas mempunyai arti bahwa kita menyiapkan 3 tempat dalam memori (yang kita beri nama *r*, *alas*, *tinggi*, dan *luas*) dan masing-masing tempat dapat menyimpan sebuah data. Variabel *r*, *alas*, dan *tinggi* masing-masing digunakan untuk menyimpan data bertipe bilangan bulat/integer (tipe **int**) dan variabel *luas* digunakan untuk menyimpan data bertipe bilangan pecahan (tipe **float**). Dengan demikian kita telah menyiapkan 3 variabel masing-masing dengan 2 komponen yaitu nama dan tipenya. Variabel yang bertipe sama dapat dideklarasikan menjadi satu dengan dipisahkan oleh tanda koma.

Contoh:

Deklarasi variabel dalam contoh di atas dapat juga ditulis sebagai berikut:

```
int r, alas, tinggi;
float luas;
```

Untuk sementara, keempat variabel tersebut belum memiliki nilai sampai ada perintah untuk menyimpan / mengisinya. Pemberian nilai terhadap variabel dilakukan dengan statemen *assignment* atau dengan mengisi langsung dari papan ketik (dengan perintah *scanf*, yang akan dibahas lebih lanjut pada bab berikutnya).

Contoh pemberian nilai variabel dengan statemen *assignment*:

```
alas = 5;
tinggi = 10;
luas = 0.5 * alas * tinggi;
```

Statemen *assignment*, sebagai contoh, **alas = 5** digunakan untuk mengubah / mengganti nilai yang tersimpan di lokasi memori sebuah variabel. Simbol "=" digunakan untuk melambangkan proses *assignment* dan disebut operator *assignment*.

Nama variabel yang diberi nilai harus dituliskan di sebelah kiri operator *assignment*. Di sebelah kanan operator kita tuliskan ekspresi yang menghasilkan nilai data yang akan disimpan.

Deklarasi Konstanta

Konstanta adalah variabel yang nilainya tetap (tidak berubah-ubah) selama program berjalan. Sebagai contoh, dalam program di atas konstanta **phi** digunakan untuk menggantikan nilai **3.14**. Maka nilai **phi** tidak berubah-ubah (tetap **3.14**) selama program berjalan. Ini berbeda dengan variabel biasa (bukan konstanta), misalnya **r**, yang nilainya dapat diganti-ganti 5, 8.5, 10, 100 atau yang lain selama program berjalan.

Suatu konstanta didefinisikan dengan statemen **#define** yang diletakkan di atas / sebelum statemen **main()**.

Contohnya :

```
#define PHI 3.14
#define NIM "0111500382"
#define NAMA "Sri Widhiyant"
```

B.5. Memasukkan dan menampilkan data

Untuk memasukkan data menggunakan perintah *cin*, sedangkan untuk menampilkan data menggunakan *cout*. Keduanya memerlukan *iostream.h*

Contoh :

```
cin >> a;
cout << "Tampilkan data";
```

setiap data yang digunakan dalam bahasa C harus memiliki tipe supaya bias dilakukan proses.

Tipe data yang dimiliki bahasa C beberapa diantaranya adalah :

No	Tipe Data	Ukuran	Range (Jangkauan)	Format	Keterangan
1	char	1 byte	- 128 s/d 127	%c	Karakter/string
2	int	2 byte	- 32768 s/d 32767	%i , %d	Integer/bilangan bulat
3	float	4 byte	- 3.4E-38 s/d 3.4E+38	%f	Float/bilangan pecahan
4	double	8 byte	- 1.7E-308 s/d 1.7+308	%lf	Pecahan presisi ganda
5	void	0 byte	-	-	Tidak bertipe

B.6. Escape Character

Terdapat beberapa karakter tertentu yang di perlukan untuk pem-format-an, sehingga cara menuliskannya juga berbeda. Gambar 3.3 merupakan daftar *escape character*

Character	What It Means
\a	Bell (alert)
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Tab
\v	Vertical tab
\'	Single quote
\"	Double quote
\?	Question mark
\\	Backslash
\ooo	Octal notation
\xhhh	Hexadecimal notation

Gambar 3.4. *Escape Character*

B.7. ASCII Character

Dengan menggunakan bahasa C kita dapat membuat format penulisan dengan tambahan berupa layout garis. Kita dapat menampilkan ASCII Character dengan menampilkan karakter dari suatu nilai decimal ASCII. Misal untuk menampilkan Ω menggunakan (char)234. Carilah daftar karakter dalam table Extended ASCII.

B.8. Ekspresi

Ekspresi merupakan sebuah statemen yang mengevaluasi sehingga menghasilkan suatu nilai. Misal $3+2$ merupakan ekspresi yang menghasilkan nilai 5. **Statemen penugasan (assignment statement)**

$x=5+3$

juga merupakan ekspresi, karena statemen tersebut mengevaluasi hasil dari $5+3$ dan meng-assign hasil tersebut ke variabel x , kemudian menghasilkan nilai x menjadi 8.

Statemen *assignment* bukan merupakan ekspresi kesamaan tetapi merupakan statemen perubahan / pemberian nilai kepada suatu variabel.

Sebagai contoh ekspresi

$k = k + 1;$

merupakan statemen *assignment* yang sah yang mempunyai arti bahwa nilai hasil penjumlahan dari ekspresi $k + 1$ disimpan ke variabel k . Apabila mula-mula variabel k menyimpan data 10, maka selanjutnya data yang tersimpan di variabel k akan berubah menjadi 11 yaitu hasil penjumlahan dari 10 (nilai k yang lama) dan 1.

Operator assignment

Operator *assignment* (penugasan) adalah $=$.

$x=5+3$

memiliki arti : nilai dari x akan berubah dan berisi hasil dari ekspresi $5+3$, sehingga nilai x akan berisi 8.

Kita tidak boleh menuliskan

$5=x$

Operator matematika

Operator matematika yang digunakan dalam bahasa C adalah :

- $*$: untuk perkalian
- $/$: untuk pembagian
- $\%$: untuk sisa pembagian (modulus)
- $+$: untuk pertambahan
- $-$: untuk pengurangan

Ordo Operasi Aritmatik

Ekspresi aritmatik, seperti $2 + 3 * 5$ sering melibatkan lebih dari satu operator. Ekspresi seperti ini dapat dievaluasi (dihitung nilainya) dengan beberapa cara tergantung operator mana yang didahulukan. Misalnya, ekspresi tersebut kalau dievaluasi dari sebelah kiri akan menghasilkan nilai 25, sedangkan bila dievaluasi dari sebelah kanan menghasilkan nilai 17. Untuk menghindari kerancuan ini, C menggunakan sistem ordo evaluasi untuk menentukan urutan evaluasinya. Seperti dalam aljabar aritmatika biasa, dalam C operator-operator perkalian, pembagian dan modulus ($*$, $/$, $\%$) memiliki ordo yang lebih tinggi dari pada operator penambahan dan pengurangan ($+$ dan $-$). Operator yang memiliki ordo lebih tinggi akan dievaluasi lebih dulu dibandingkan dengan operator yang ordonya lebih rendah. Operator yang memiliki ordo sama akan dievaluasi mulai dari sebelah kiri. Misalnya, dalam contoh di atas, ekspresi $3 * 5$ akan dievaluasi lebih dahulu (menghasilkan nilai 15), kemudian dilanjutkan operasi penambahan (ditambah dengan 2) sehingga menghasilkan nilai 17. Urutan evaluasi juga dapat ditentukan dengan memberikan tanda kurung buka dan kurung tutup pada bagian ekspresi yang harus didahulukan. Sebagai contoh, ekspresi $(2 + 3) * 5$ akan menghasilkan nilai 25.

Operator Aritmatik (Unary) ++ Dan --

Operator `'++'` dan `'--'` merupakan operator khusus yang ada dalam bahasa C dan jarang dijumpai dalam bahasa yang lain. Operator `'++'` akan menambahkan nilai 1 ke operannya, sedangkan operator `'--'` akan mengurangi operan dengan nilai 1. Operator-operator ini dapat ditulis sebelum atau sesudah operannya. Operator yang ditulis sebelum operannya akan

menghasilkan nilai yang berbeda dengan yang ditulis sesudah operannya. Sebagai contoh, A++ mempunyai arti yang sedikit berbeda dengan ++A. Apabila nilai variabel A = 5 maka ekspresi berikut masing-masing akan menghasilkan nilai:

Ekspresi	Keterangan	Hasil
B = A++;	Nilai A disimpan ke B lebih dahulu kemudian ditambah 1	A = 6 dan B = 5
B = ++A;	Nilai A ditambah satu lebih dahulu kemudian disimpan ke B	A = 6 dan B = 6

Menterjemahkan Ekspresi Aljabar Ke Program C

Banyak masalah dalam ilmu pengetahuan dan bisnis memerlukan penggunaan ekspresi dan rumus-rumus aljabar. Jika kita ingin menggunakan C untuk memecahkan masalah-masalah semacam ini maka kita harus menterjemahkan ekspresi aljabar tersebut ke bentuk ekspresi yang dikenal oleh C. Sebagai contoh, ekspresi $5x + y$ dapat diterima sebagai ekspresi aljabar, karena $5x$ secara implisit menyatakan operasi perkalian 5 kali x . Tetapi C tidak mengizinkan penggunaan operator implisit, sehingga ekspresi tersebut harus ditulis $5 * x$. Berikut ini beberapa contoh ekspresi aljabar dan konversinya dalam ekspresi C.

Ekspresi aljabar	Ekspresi program C
$A + \frac{B}{C} + D$	$A + B / C + D$
$A + \frac{B}{C + D}$	$A + B / (C + D)$
$\frac{A + B}{C} + D$	$(A + B) / C + D$
$\frac{A + B}{C + D}$	$(A + B) / (C + D)$
$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$	$X = (-b + \text{sqrt}(b*b - 4 * a * c)) / (2 * a)$

Catatan:

Dalam bahasa C, akar kuadrat dari x , yaitu \sqrt{x} , ditulis sebagai $\text{sqrt}(x)$ dan membutuhkan pustaka `math.h`

@ Materi diambil dari materi kuliah Algoritma Pemrograman I Program Studi Ilmu Komputer Universitas Sanata Dharma, tahun 2003/2004.

C. Tugas

1. Buatlah program dengan menggunakan bahasa C agar menampilkan keluaran berikut :

- a. Nama : Sri Hartati Wijono
 Nomor : 105314012
 Alamat : Jl. Tasura 1234 Paingan

- b.

Nama : Sri Hartati Wijono Nomor : 105314012 Alamat : Jl. Tasura 1234 Paingan
--

2. Tulislah ekspresi aljabar berikut ke dalam ekspresi C:

a. $p = 5a - 7b + 2(c - 4)$

b. $q = \frac{x - 10}{x + 12y}$

$$c. \quad r = (8 - a)(6 - b) + c^2$$

$$d. \quad s = \sqrt{2x^2 + 3y + 4} + 10x$$

$$e. \quad t = \frac{\sqrt{7a}}{c-10} - \sqrt{\frac{3b+25}{c+13}}$$

Buatlah program dalam bahasa C untuk menghitung nilai p, q, r, s, t dari soal nomor 1. Nilai p, q, r, s, t ditampilkan ke layar, dan nilai lain (input) berupa masukan dari user.

3. Mahmud mendapat tugas dari gurunya untuk menghitung isi kerucut dengan rumus:

$$I = \frac{1}{3} \times \pi \times R^2 \times H$$

Dengan:

I = isi kerucut

R = jari-jari alas kerucut

H = tinggi kerucut

Mahmud mencoba membuat program kecil dengan bahasa C++ untuk menghitung isi kerucut. Tetapi ternyata program tidak berjalan dengan baik; **hasil yang didapat tidak cocok dengan perhitungan secara manual**. Berikut ini *source code* program Mahmud:

```
#include <conio.h>
#include <iostream.h>
#define phi 3.14
void main (void) {
    // Menghitung isi kerucut
    int R, H, I, phi;
    clrscr();
    cout<<"Masukkan jari-jari alas kerucut: ";
    cin>>R;
    cout<<"Masukkan tinggi kerucut: ";
    cin>>H;
    I = 1/3 * phi * R * R * H;
    cout<<"Isi kerucut: "<<I<<endl;
}
```

Bantulah Mahmud untuk menyelesaikan masalahnya.

4. Buatlah algoritma dan program untuk mengkonversi menit menjadi jam. Masukan adalah menit dan keluaran adalah jam dan menit. Contoh :
Masukkan jumlah menit : 125
125 menit adalah 2 jam 5 menit
5. Buatlah algoritma dan program untuk mengkonversi kilometer menjadi mile. Masukan adalah besar kilometer dan keluaran besar mile. Contoh :
Masukkan besar kilometer : 1
1 kilometer adalah 0.621371 mile
6. Buatlah algoritma dan program untuk menentukan apakah suatu bilangan kelipatan 3 atau tidak.
7. Buatlah algoritma dan program untuk menghitung nilai akhir (dalam bentuk angka) untuk seorang mahasiswa. Aturan untuk menghitung nilai akhir :
 - Nilai kuis berbobot 10%.
 - Nilai tugas berbobot 20%
 - Nilai USIP 1 berbobot 30%.
 - Nilai UAS berbobot 40%

Nilai-nilai tersebut berupa masukan user dari keyboard. Nilai akhir ditampilkan di layar.

8. Terdapat algoritma di bawah ini. Ubah algoritma tersebut menjadi program.
 1. Mulai
 2. Masukkan bilangan a kurang dari 100
 3. Bagi bilangan a dengan 10. Masukkan hasil pembagian yang berupa bilangan integer(bulat) ke dalam variable puluhan.

4. Hitung sisa hasil bagi bilangan a dengan 10. Masukkan sisa hasil bagi tersebut ke dalam variable satuan.
5. Tampilkan : bilangan a dan puluhan serta satuan tersebut. Contoh : 56 = 5 puluhan dan 6 satuan
6. Selesai.

D. SOAL TANTANGAN

9. Buatlah algoritma dan program untuk menghitung nilai investasi kita setelah sekian bulan, jika diketahui bunga dalam satu tahun. Rumus untuk menghitung nilai investasi adalah :

$$\text{futureInvestment} = \text{investmentAmount} \times (1 + \text{monthlyInterestRate})^{\text{numberOfYears} \times 12}$$
10. Terdapat sebuah persamaan kuadrat $f(x)=ax^2+bx+c$. Diinginkan untuk memperoleh nilai x1 dan nilai x2. Buatlah algoritma dan program dengan nilai masukan a,b, dan c, kemudian keluaran adalah nilai x1 dan x2.
11. Terdapat masukan dari keyboard terdiri dari 4 nilai. Hitunglah rata-rata ke-4 nilai tersebut. Kemudian hitunglah standar deviasi dari ke 4 nilai tersebut. Buatlah terlebih dahulu algoritmanya, kemudian programnya.

E. Laporan Praktikum

Format Laporan Praktikum

1. Tujuan Praktikum (10%)
2. Tuliskan algoritma dan program serta capture output untuk semua soal. (40%)
3. Jangan lupa beri komentar pada program untuk identitas program dan penjelasan perintah-perintah penting. (40%)
4. Daftar Pustaka (10%)

Laporan dikumpulkan pada hari pertama minggu ke-2 September 2012. Laporan terlambat setelah 1 minggu tidak akan diberi nilai. Keterlambatan akan diberi nilai sebenarnya dan pengurangan 5 untuk setiap harinya.

@ You are nothing if you don't do something @