Praktikum 1

1. Menghitung biaya foto kopi

Jawab:

Untuk menghitung biaya fotokopi, perlu ditetapkan terlebih dahulu harga per lembar fotokopi. Kemudian kita perlu tahu berapa jumlah lembar fotokopi yang akan dihitung dan program dapat menanyakan hal ini kepada user untuk menginputkan jumlah lembar fotokopi. Setelah dihitung berapa besar biaya yang harus dibayar, selanjutnya program menampilkan informasi tersebut ke layar komputer.

Langkah-langkah:

* tanya, berapa jumlah lembar foto kopi
* hitung besar biaya fotokopi
* tampilkan informasi biaya fotokopi

Analisis kebutuhan

* Kebutuhan output
  + Tampilan:

Jumlah lembar : XXX lembar

Harga per lembar: Rp. XXX

Total biaya : Rp. XXX

* Kebutuhan input
  + Tampilan:

Ketikkan jumlah lembar: ??

* Kebutuhan proses
  + Menghitung total biaya fotokopi, dgn formula:

Biaya  lembar \* HARGA

* Kebutuhan variable
  + Untuk menyimpan nilai jumlah lembar fotokopi, nama variable ‘lembar’ bertipe integer
  + Untuk menyimpan nilai total biaya fotokopi, nama var ‘biaya’ bertipe real
* Kebutuhan konstanta
  + Untuk menetapkan harga per lembar fotokopi, konstanta bernama HARGA bertipe real

Algoritma BiayaFotoKopi (input, output)

{digunakan untuk menghitung total biaya yg harus dibayar sesuai jumlah lembar foto kopi}

Deklarasi

Konstanta:

HARGA = 200.00

Variable:

Lembar : integer

Biaya : real

Deskripsi

Start

Read(lembar)

Biaya  lembar \* HARGA

Write(biaya)

Stop

Implementasi dalam C

/\*

Nama File: kopi.c

Programmer: si Anu

digunakan untuk menghitung total biaya

yg harus dibayar sesuai jumlah lembar foto kopi

\*/

#include <stdio.h>

#include <stdlib.h>

//definisi konstanta

#define HARGA 200.00

Int main (){

int lembar;

double biaya;

printf("Program Menghitung Biaya Fotokopi\n\n");

//membaca banyaknya lembar foto kopi

printf("Ketikkan jumlah lembar foto kopi: ");

scanf("%d", &lembar); fflush(stdin);

//menghitung total biaya

biaya = lembar \* HARGA;

printf("Jumlah lembar : %d lembar\n", lembar);

printf("Harga perlembar : Rp. %0.2lf\n", HARGA);

printf("Total biaya : Rp. %0.2lf\n", biaya);

return 0;

}

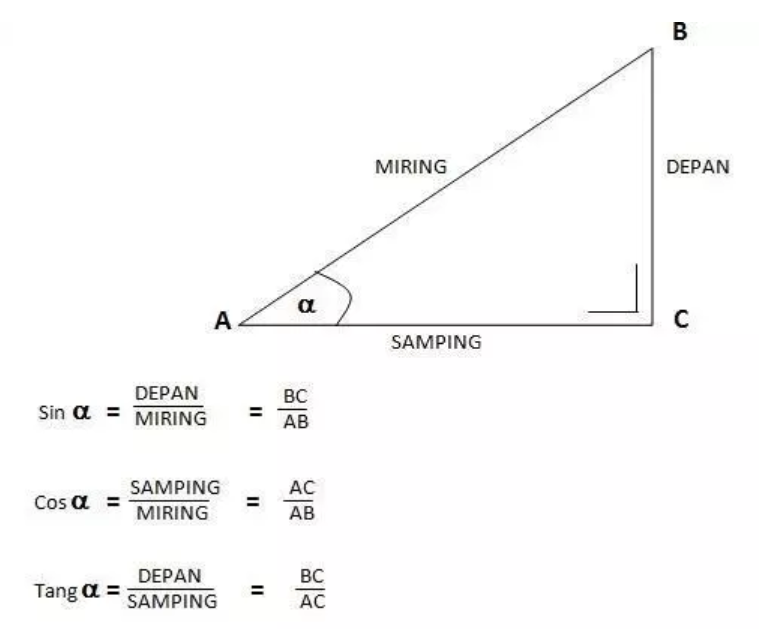
1. Menghitung tinggi layangan berdasarkan sudut tali (elevasi) dan panjang tali terulur

Implementasi dalam C

Langkah-langkah:

Untuk menghitung berapa tinggi sebuah layangan, kita perlu mengetahui berapa panjang tali terulur dan berapa sudut antara tali dan daratan tanah (sudut elevasi). Kedua nilai ini akan digunakan untuk menghitung tinggi layangan dari tanah daratan, tentu saja kita berasumsi bahwa tali dianggap lurus, artinya lengkung tali yang terulur diabaikan.

Kita tahu, bahwa formula yang bisa digunakan untuk menghitung tinggi layangan dari tanah dengan mengetahui nilai sudut dan panjang tali terulur adalah formula mencari tinggi sebuah segitiga siku-suka dengan mengetahui sudut elevasi dan panjang sisi miring, perhaitkan penjelasan berikut ini:



Apabila sisi miring dan sudut α diketahui, maka kita dapat mencari DEPAN atau tinggi dari segitiga melalui cara:

Tinggi = sin α \* MIRING

Catatan: dalam bahasa C, untuk mencari nilai sinus suatu sudut, nilai sudutnya harus dalam radian, bukan dalam derajat. Sehingga apabila sudut dalam derajat yang diketahui maka kita harus mengkonversi sudut dalam derajat ke nilai sudut dalam radian, sbb:

1 derajat = 180/π = 0.0174533

Sehingga apabila x derajat, kita tinggal kalikan dengan nilai tersebut untuk memperoleh nilai radian:

Radian = 180/π \* sudut derajat

Radian = 180/π \* x, dimana x adalah sudut dalam derajat

Dengan demikian untuk mencari tinggi layangan langkah-langkahnya adalah sebagai berikut:

- baca nilai panjang tali, misal y

- baca nilai sudut derajat, misal x

- konversi sudut derajat x ke radian dengan cara:

rad = 180/π \* x

- hitung tinggi dengan cara:

Tinggi = sin rad \* y

- tampilkan informasi

Analisis kebutuhan:

- kebutuhan output:

Tampilan:

Panjang Tali Terulur (dalam meter) : XXX meter

Sudut Tali dgn Tanah (dalam derajat: XX derajat

Tinggi layangan (dalam meter) : XXX meter

- kebutuhan input:

Tampilan:

Ketikkan Panjang Tali Terulur (dalam meter) : XXX

Ketikkan Sudut Tali dgn Tanah (dalam derajat: XX

- kebutuhan konstanta:

Konstanta bilangan phi bernilai 3.14, beri nama PHI = 3.14

- kebutuhan variabel:

- variabel untuk menyimpan nilai panjang tali, beri nama *tali,* bertipe *double*

*-* variabel untuk menyimpan nilai sudut tali dalam derajat, beri nama *sudut* bertipe *double*

*-* variabel untuk menyimpan nilai tinggi layangan, beri nama *tinggi* bertipe *double*

- variabel untuk menyimpan nilai sudut dalam radian, beri nama *radian* bertipe *double*

- Kebutuhan proses:

- rumus untuk mengkonversi sudut dalam derajat ke dlam sudut dalam radian:

*radian = 180/PHI \* sudut*

- rumus untuk menghitung tinggi layangan:

*tinggi = (sin(radian)) \* tali*

Algoritma tinggiLayangan(input, output)

{Menghitung tinggi layangan berdasarkan sudut tali (elevasi) dan panjang tali terulur }

Deklarasi

Konstanta

PHI = 3.14

Variabel

tali, sudut, radian, tinggi: real/double

Deskripsi:

Start

write(‘Program untuk menghitung tinggi layangan’)

write(‘Ketikkan panjang tali dalam meter: ‘)

read(tali)

write(‘Ketikkan sudut tali dalam derajat: ‘)

Read(sudut)

radian <- sudut/180 \* PHI

tinggi <- (sin(radian)) \* tali

write(‘Panjang tali terulur : ‘, tali, ‘meter’)

write(‘Sudut tali dengan daratan : ‘, sudut,’ derajat’)

write(‘Tinggi layangan dari daratan: ‘, tinggi, ‘ meter’)

Stop

/\*

Nama File: layang2.c

Menghitung tinggi layangan berdasarkan sudut tali (elevasi) dan panjang tali terulur

\*/

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

//konstanta PHI

#define PHI 3.14

main()

{

//rumus mencari tinggi segitiga siku2

//sin a = t/m

//t = (sin a)\*m

//t adalah tinggi

//m adalah panjang tali

//rumus mengkonversi sudut derajat ke radian

//1 derajat = PHI/180 radian

//30 derajat = 30 \* PHI/180 radian

//a derajat = a \* PHI/180 radian

//sin a = sin (a \* PHI/180) radian

//sehingga t = (sin (a\*PHI/180)) \* m

// atau t = (sin(PHI\*a/180)) \* m

double tali, sudut, tinggi, radian;

printf("Program untuk menghitung tinggi layangan\n\n");

printf("Ketikkan panjang tali dalam meter: ");

scanf("%lf", &tali); fflush(stdin);

printf("Ketikkan sudut tali dalam derajat: ");

scanf("%lf", &sudut); fflush(stdin);

//hitung atau konversikan sudut ke radian

radian = sudut/180 \* PHI;

//hitung tinggi layangan

tinggi = (sin(radian)) \* tali;

printf("Panjang tali terulur : %0.2lf meter\n", tali);

printf("Sudut tali dengan daratan : %0.2lf derajat\n", sudut);

printf("Tinggi layangan dari daratan: %0.2lf meter\n", tinggi);

}

1. Menghitung kebutuhan kalori manusia dewasa berdasarkan jenis kelamin, jenis pekerjaan, umur, tinggi badan, berat badan.

Jawab:

/\*

Nama File: kalori.c

Topik : Struktur sequential

Menghitung kebutuhan kalori manusia dewasa berdasarkan jenis

kelamin, jenis pekerjaan, umur, tinggi badan, berat badan.

Rumus Harris-Benedict:

Jumlah kebutuhan kalori per hari didapatkan dengan memperhitungkan

BMR dan tingkat aktivitas harian seseorang. Rumus yang paling banyak

digunakan oleh ahli gizi untuk menghitung BMR adalah Rumus Harris-Benedict.

Rumus ini dihitung berdasarkan usia, jenis kelamin, berat badan, dan tinggi badan.

Untuk laki-laki:

(88,4 + 13,4 x berat dalam kilogram) + (4,8 x tinggi dalam sentimeter)

- (5,68 x usia dalam tahun)

Untuk wanita: (447,6 + 9,25 x berat dalam kilogram) + (3,10 x tinggi

dalam sentimeter) - (4,33 x usia dalam tahun)

Hasil perhitungan BMR kemudian dikalikan dengan angka aktivitas harian

rata-rata orang tersebut. Angka ini berkisar antara 1,2–1,9 tergantung dari

seberapa tinggi aktivitas harian seseorang. Semakin jarang seseorang

melakukan aktivitas fisik, semakin rendah pula angka aktivitas hariannya.

- faktor aktivitas fisik:

rendah: 1.2

sedang: 1.3

berat : 1.4

- Perhitungan BMR:

pria = (88.4 \* 13.8 \* BB) + (4.8 \* TB) - (5.68 \* USIA)

wanita = (447.6 + 9.25 \* BB) + (3.10 \* TB) - (4.33 \* USIA)

- Total kebutuhan kalori:

kalori pria:

kalori aktivitas rendah = BMR pria \* 1.2

kalori aktivitas sedang = BMR pria \* 1.3

kalori aktivitas berat = BMR pria \* 1.4

kalori wanita:

kalori aktivitas rendah = BMR wanita \* 1.2

kalori aktivitas sedang = BMR wanita \* 1.3

kalori aktivitas berat = BMR wanita \* 1.4

Berikut perhitungan dasar untuk perempuan:

BMR = 655 + (1,8 x tinggi dalam cm) + (9,6 x berat dalam kilogram) - (4,7 x umur dalam tahun)

Untuk perempuan dengan usia 30, tinggi 167,6 cm dan berat badan 68 kilogram, maka perhitungan BMR-nya akan menjadi: 655 + 301,7 + 653 - 141 = sekitar 1.470 kalori per hari.

Berikut perhitungan dasar untuk pria:

BMR = 66 + (13,7 x berat dalam kilogram) + (5 x tinggi dalam cm) - (6,7 x umur dalam tahun).

Untuk contoh hasil perempuan, misalnya, maka bisa mengalikan hasil BMR tersebut dengan tingkat aktivitas. Kali jumlah BMR dengan 1,2 jika kamu tidak aktif; 1,375 jika melakukan olahraga ringan satu hingga tiga kali seminggu; 1,55 jika melakukan olahraga ringan 6-7 kali seminggu; kalikan dengan 1,75 jika sangat aktif (olahraga keras setiap hari atau berolahraga dua kali sehari) dan 1,9 jika ekstra aktif (latihan keras dua kali atau lebih per hari).

Jadi untuk seseorang dengan bobot 68 kilogram yang cukup aktif, maka kan menjadi: 1,470 x 1,55 = sekitar 2.280. Hasil itu merupakan berapa banyak kalori yang dibakar tubuh pada hari tertentu.

\*/

#include <stdio.h>

#include <stdlib.h>

//konstanta

#define RENDAH 1.2

#define SEDANG 1.3

#define BERAT 1.4

main()

{

double beratBadan, tinggiBadan, usia;

double bmrPria, bmrWanita;

double kaloriPriaRendah, kaloriPriaSedang, kaloriPriaBerat;

double kaloriWanitaRendah, kaloriWanitaSedang, kaloriWanitaBerat;

printf("\nMenghitung Kebutuhan Kalori Harian\n\n");

//kebutuhan masukkan

printf("Ketikkan nilai-nilai berikut:\n");

printf("Berat Badan dalam kg : ");

scanf("%lf", &beratBadan); fflush(stdin);

printf("Tinggi Badan dalam meter: ");

scanf("%lf", &tinggiBadan); fflush(stdin);

printf("Usia dalam tahun : ");

scanf("%lf", &usia); fflush(stdin);

//menghitung BMR

bmrPria = 66 + (13.7 \* beratBadan) + (5 \* tinggiBadan) - (6.7 \* usia);

bmrWanita = 655 + (1.8 \* tinggiBadan) + (9.6 \* beratBadan) - (4.7 \* usia);

//menghitung kebutuhan kalori pria

kaloriPriaRendah = bmrPria \* RENDAH;

kaloriPriaSedang = bmrPria \* SEDANG;

kaloriPriaBerat = bmrPria \* BERAT;

//menghitung kebutuhan kalori wanita

kaloriWanitaRendah = bmrWanita \* RENDAH;

kaloriWanitaSedang = bmrWanita \* SEDANG;

kaloriWanitaBerat = bmrWanita \* BERAT;

//menamp[ilkan informasi ke layar

printf("\nKebtuhan Kalori Pria:\n");

printf("\tPria Aktivitas Rendah : %0.2lf Kalori per hari\n", kaloriPriaRendah);

printf("\tPria Aktivitas Sedang : %0.2lf Kalori per hari\n", kaloriPriaSedang);

printf("\tPria Aktivitas Berat : %0.2lf Kalori per hari\n", kaloriPriaBerat);

printf("\nKebtuhan Kalori Wanita:\n");

printf("\tWanita Aktivitas Rendah: %0.2lf Kalori per hari\n", kaloriWanitaRendah);

printf("\tWanita Aktivitas Sedang: %0.2lf Kalori per hari\n", kaloriWanitaSedang);

printf("\tWanita Aktivitas Berat : %0.2lf Kalori per hari\n", kaloriWanitaBerat);

}

1. Menghitung biaya cicilan kredit barang berdasarkan harga pokok, bunga kredit, uang muka, banyaknya kali (waktu) cicilan.

Jawab:

Untuk membuat program ini, kita perlu mengetahui beberapa hal. Seperti, Besarnya hutang adalah harga barang dikurangi uang muka yang dibayarkan, besar cicilan bulanan kredit barang, adalah besar hutang ditambah besar bunga kredit bulanan. Sementara besar bunga bulanan dapat dihitung dari bunga tahunan dibagi 12 (jumlah bulan dalam setahun). Selain itu, pokok cicilan bulanan merupakan pokok hutnag dibagi dengan jumlah kali cicilan, sehingga besar cicilan bulanan merupakan besar pokok cicilan bulanan ditambah dengan bunga bulanan. Dengan demikian langkah-langkah yang harus dilakukan untuk menyelesaikan kasus ini adalah sebagai berikut:

Langkah-langkah:

- tetapkan besar persen BUNGA pertahun dan dinyatakan dalam nilai real, misal bunga 10 persen menjadi 0.10.

- baca input terkait:

- Harga Barang

- Uang muka

- jumlah kali cicilan

- pembayaran ke berapa

- hitung hal-hal berikut:

- Pokok hutang, dengan rumus:

Pokok hutang = harga barang - uang muka

- hitung bunga per bulan, dengan rumus:

Bunga bulanan = BUNGA tahunan / 12

- hitung pokok cicilan bulanan dengan rumus

Pokok cicilan = pokok hutang/jumlah kali cicilan

- hitung besar cicilan bulanan dengan rumus:

Cicilan bulanan = pokok cicilan + bunga bulanan

- hitung sisa hutang dengan rumus:

Sisa hutang = pokok hutang - pokok cicilan \* pembayaran yang telah dilakukan

- tampilkan informasi terkait:

Harga Barang : Rp. XXXX

Uang Muka : Rp. XXXX

Banyaknya bulan cicilan : XX kali cicilan

Pokok Hutang : Rp. XXXX

Besar bunga bulanan : Rp. XXXX

Besar Cicilan per bulan : Rp. XXXXX

Cicilan ke- : X

Sisa Pokok Hutang : Rp. XXXX

Analisis Kebutuhan:

- Kebutuhan Konstanta: konstanta bernama BUNGA dengan nilai (misal bunga 10 persen) 0.10

- kebutuhuan tampilan output:

INFORMASI KREDIT BARANG

Harga Barang : Rp. XXXX

Uang Muka : Rp. XXXX

Banyaknya bulan cicilan : XX kali cicilan

Pokok Hutang : Rp. XXXX

Besar bunga bulanan : Rp. XXXX

Besar Cicilan per bulan : Rp. XXXXX

Cicilan ke- : X

Sisa Pokok Hutang : Rp. XXXX

- Kebutuhan tampilan input:

KREDIT BARANG KOPERASI SUKASUKA

Ketikkan harga barang : Rp. ????

Ketikkan besar uang muka : Rp. ????

Ketikkan jumlah bulan kredit : ??

Ketikkan Cicilan kali ke berapa: ??

- Kebutuhan variabel:

- variabel untuk menyimpan harga barang, beri nama *harga* bertipe *double*

*-* variabel untuk menyimpan besar uang muka, beri nama *uangMuka* bertipe *double*

*-* variabel untuk menyimpan pokok hutang, beri nama *hutang*  bertipe *double*

- variabel untuk menyimpan bunga bulanan, beri nama *bunga* bertipe *double*

- variabel untuk menyimpan besar cicilan bulanan, beri nama *cicilan* bertipe *double*

- variabel untuk menyimpan besar sisa pokok utang, beri nama *sisa*  bertipe *double*

- variabel untuk menyimpan besar pokok bulanan, beri nama *pokok* bertipe *double*

*-* variabel untuk menyimpan jumlah kali cicilan, beri nama *bulan* bertipe *int*

- variabel untuk menyimpan kali ke berapa menyicil, beri nama *kaliKe* bertipe *int*

- Kebutuhan Proses/rumus/formula:

- Pokok hutang, dengan rumus:

*Pokok hutang = harga barang - uang muka*

- hitung bunga per bulan, dengan rumus:

*Bunga bulana = BUNGA tahunan / 12*

- hitung pokok cicilan bulanan dengan rumus

*Pokok cicilan = pokok hutang/jumlah kali cicilan*

- hitung besar cicilan bulanan dengan rumus:

*Cicilan bulanan = pokok cicilan + bunga bulanan*

- hitung sisa hutang dengan rumus:

*Sisa hutang = pokok hutang - pokok cicilan \* pembayaran yang telah dilakukan*

***Algoritma kredit(input, output)***

*{ pengolahan informasi kredit barang}*

*Deklarasi*

*Konstanta BUNGA = 0.10: real/double*

*Variabel:*

*harga, hutang, bunga, cicilan, uangMuka, sisa, pokok: real/double*

*Bulan, kaliKe: integer*

*Deskripsi*

*Start*

*write*(‘Ketikkan harga barang : Rp. ‘)

read(harga)

write(‘Ketikkan besar uang muka : Rp. ‘)

read(uangMuka)

write (‘Ketikkan jumlah bulan kredit : ‘)

read(bulan)

write(‘Ketikkan Cicilan kali ke berapa: ‘)

read(kaliKe)

//menghitung pokok hutang

hutang <- harga - uangMuka

//menghitung bunga perbulan

bunga <- (BUNGA \* hutang)/12

//menghitung pokok bulanan

pokok <- hutang/bulan

//menghitung besar cicilan perbulan

cicilan <- pokok + bunga

//menghitung sisa hutang

sisa <- hutang - pokok\*kaliKe

//menampilkan informasi

write(‘Harga Barang : Rp. ‘ + harga)

write(‘Uang Muka : Rp. ‘ + uangMuka)

write(‘Banyaknya bulan cicilan: “ + bulan + ‘ kali cicilan’)

write(‘Pokok Hutang : Rp. ‘ + pokok)

write(‘Besar bunga bulanan : Rp. “ + bunga)

write(‘Besar Cicilan per bulan : Rp. “ + cicilan)

write(‘Cicilan ke- : ‘ + kaliKe)

write(‘Sisa Pokok Hutang : Rp. ‘ + sisa);

*stop*

**Implementasi dalam C**

/\*

Nama File: kredit2.c

Menghitung besar cicilan kredit barang dan sisa hutang

\*/

#include <stdio.h>

#include <stdlib.h>

//konstanta bunga kredit pertahun dalam persen

#define BUNGA 0.09

main()

{

double harga, hutang, bunga, cicilan, uangMuka, sisa, pokok;

int bulan, kaliKe;

printf("Ketikkan harga barang : Rp. ");

scanf("%lf", &harga); fflush(stdin);

printf("Ketikkan besar uang muka : Rp. ");

scanf("%lf", &uangMuka); fflush(stdin);

printf("Ketikkan jumlah bulan kredit : ");

scanf("%d", &bulan); fflush(stdin);

printf("Ketikkan Cicilan kali ke berapa: ");

scanf("%d", &kaliKe); fflush(stdin);

//menghitung pokok hutang

hutang = harga - uangMuka;

//menghitung bunga perbulan

bunga = (BUNGA \* hutang)/12;

//menghitung pokok bulanan

pokok = hutang/bulan;

//menghitung besar cicilan perbulan

cicilan = pokok + bunga;

//menghitung sisa hutang

sisa = hutang - pokok\*kaliKe;

//menampilkan informasi

printf("\nHarga Barang : Rp. %10.1lf\n",harga);

printf("Uang Muka : Rp. %10.1lf\n",uangMuka);

printf("Banyaknya bulan cicilan: %d kali cicilan\n",bulan);

printf("Pokok Hutang : Rp. %10.1lf\n",pokok);

printf("Besar bunga bulanan : Rp. %10.1f\n",bunga);

printf("Besar Cicilan per bulan: Rp. %10.1lf\n", cicilan);

printf("Cicilan ke- : %d\n", kaliKe);

printf("Sisa Pokok Hutang : Rp. %10.1lf\n", sisa);

}

1. Menghitung kecepatan suatu kendaraan berdasarkan jarak tempuh dan waktu tempuh

Jawab:

Untuk menghitung kecepatan kendaraan, dapat dilakukan melalui jarak tempuh dibagi dengan waktu tempuh.

Analisis Kebutuhan

* Kebutuhan output
  + Tampilan:

Jarak Tempuh (dlm km): XXX Km

Waktu tempuh: XX:XX:XX

Kecepatan (km/jam) : XX km/jam

* Kebutuhan input
  + Tampilan:

Ketikkan jarak tempuh dalam kilometer: ??

Ketikkan waktu tempuh (format: jj:mm:dd): ??:??:??

* Kebutuhan proses
  + Menghitung total detik dari format jj:mm:dd menjadi detik

totalDetik  jj \* 3600 + mm \* 60 + dd

* + Menkonversi jarak dalam Km menjadi meter:

Meter  km \* 1000.0

* + Menghitung kecepatan:

Cepat  meter/totalDetik;

* Kebutuhan variable:
  + Untuk menyimpan nilai total detik, nama variable totalDetik bertipe integer
  + Untuk menyimpan nilai jam, nama variable jj bertipe integer
  + Untuk menyimpan nilai menit, nama variable mm bertipe integer
  + Untuk menyimpan nilai detik, nama variable dd bertipe integer
  + Untuk menyimpan nilai jarak dalam Km, nama variable km bertipe real
  + Untuk menyimpan nilai jarak dalam meter, nama variable meter bertipe real
  + Untuk menyimpan nilai kecepatan, nama variable cepat bertipe real
  + Untuk menyimpan tanda pemisah format waktu (:), nama variable titikDua bertipe char
* Algoritma HitungKecepatan (input, output)

{Menghitung kecepatan kendaraan berdasarkan jarak tempuh dan waktu tempu}

Deklarasi

Variable:

totalDetik, jj, mm, dd: integer

meter, km, cepat: real

titikDua: char

deskripsi

start

read(km)

read(jj, titikDua, mm, titikDua, dd)

//Menghitung total detik dari format jj:mm:dd menjadi detik

totalDetik  jj \* 3600 + mm \* 60 + dd

//Menkonversi jarak dalam Km menjadi meter:

Meter  km \* 1000.0

//Menghitung kecepatan:

Cepat  meter/totalDetik;

Write(cepat)

Stop

**Implementasi dalam C**

/\*

Nama File: cepat.c

Programmer: antu

menentukan kecepatan kendaraan berdasarkan jarak tempuh

dan waktu tempuh

\*/

#include <stdio.h>

#include <stdlib.h>

main(){

int totalDetik, jj, mm, dd;

double meter, km, cepat;

char titikDua;

printf("Program Menghitung Kecepatan\n\n");

printf("Ketikkan jarak tempuh dalam Km: ");

scanf("%lf", &km); fflush(stdin);

printf("Ketikkan waktu tempuh (ex: 02:05:15): ");

scanf("%d%c%d%c%d", &jj, &titikDua, &mm, &titikDua, &dd);

//Menghitung total detik dari format jj:mm:dd menjadi detik

totalDetik = jj \* 3600 + mm \* 60 + dd;

//Menkonversi jarak dalam Km menjadi meter:

meter = km \* 1000.0;

//Menghitung kecepatan:

cepat = meter/totalDetik;

printf("Jarak Tempu : %0.2lf Km\n", km);

printf("Waktu Tempu : %02d%c%02d%c%02d\n", jj, titikDua, mm, titikDua,dd);

printf("Kecepatan : %0.2lf m/detik\n", cepat);

}