

02

Introduction  
to  
Algorithm

Algoritma  
dan  
Pengetahuan Terkait

# Algoritma dan Pengetahuan Terkait

apakah  
Algoritma  
itu sebenarnya ?

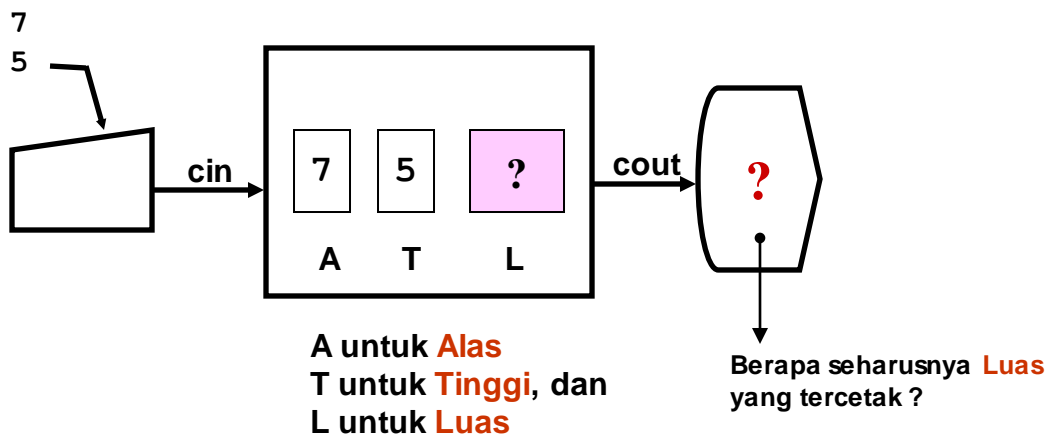
Sulitkah  
Algoritma  
itu ?

**Algoritma** Merepresentasikan apa yang Diketahui.

## Soal-5

Susun algoritma yang ditulis dalam Bahasa C++, untuk menginput **alas** dan **tinggi** sebuah segitiga, kemudian mencetak **luas** segitiga tersebut.

Untuk ilustrasi : Misal diinput 7 untuk **Alas** dan 5 untuk **Tinggi**



### Soal-5.

Susun **algoritma** untuk mencetak **luas** sebuah segitiga bila diketahui **alas** = 7 cm, dan **tinggi** = 5 cm.

Agar dapat menyusun **algoritma** untuk menjawab persoalan yang diberikan, ada **tiga** hal yang perlu dipahami/ dikuasai:

#### 1. Paham? (solving)

Beberapa konsep pemikiran dalam menghitung luas sebuah segitiga

$$\text{Luas} = \text{Alas} * \text{Tinggi} / 2$$

$$\text{Luas} = (\text{Alas} * \text{Tinggi}) / 2$$

$$\text{Luas} = 0.5 * \text{Alas} * \text{Tinggi}$$

Masih banyak konsep yang lain

#### 2. Design!

Dituangkan dalam bentuk **Algoritma** secara umum

```
A = 7
T = 5
L = (A * T) / 2
Cetak L
```

A untuk **Alas**  
T untuk **Tinggi**  
L untuk **Luas**

Orang lain paham membaca algoritma ini. Jangan menulis yang orang lain tidak paham.

#### 3. Application (coding)

Dituangkan dalam bentuk **Program**

Misal **Bahasa C++**

```
#include <iostream.h>
void main()
{ float A, T, L;
  A = 7;
  T = 5;
  L = ( A * T ) / 2;
  cout << L;
}
```

Tercetak : 17.5

7	5	17.5
A	T	L

1. Harus **mengerti maksud persoalan** yang akan diselesaikan. Tidak mengerti maksud persoalan, tidak ada yang dapat dikerjakan, walaupun hanya satu baris tulisan.

Untuk persoalan yang dicontohkan, harus mengerti apa yang dimaksud dengan luas segitiga. Kemudian harus mengetahui cara atau rumus menghitung luas sebuah segitiga bila alas dan tinggi segitiga tersebut diketahui.

Pengetahuan mengenai luas sebuah segitiga ini, bukan didapat dari pelajaran algoritma, tapi merupakan pengetahuan yang umum yang didapat dari sekolah sebelumnya.

Tidak paham rumus atau cara menghitung luas segi tiga maka tidak dapat membuat **algoritma**.

2. Setelah paham dan dapat menghitung bahwa untuk soal yang dicontohkan :

$$\begin{aligned}\text{Luas segitiga} &= (7 * 5) / 2, \text{ yaitu } (\text{alas} * \text{tinggi}) / 2 \\ &\text{atau} = 0.5 * (7 * 5)\end{aligned}$$

maka mulailah menuangkan jalan pikiran tersebut dalam bentuk langkah-langkah yang teratur secara :

**rinci,**  
 **lengkap**  
dan  **tersusun secara logis.**

Disinilah pelajaran **algoritma**, mulai berperan. Pada pelajaran algoritma, diajarkan dan dilatih bagaimana cara menuangkan jalan pikiran atau konsep penyelesaian pekerjaan kedalam suatu bentuk yang disebut **algoritma**.

Algoritma adalah langkah-langkah rinci dan lengkap yang tersusun secara logis untuk menyelesaikan suatu tugas / pekerjaan.

Sedemikian rupa, sehingga orang lain dapat ikut membaca dan memahaminya

Algoritma hanya dapat dibuat, bila jalan pikiran atau konsep penyelesaian sudah ada

Sudah tertuang dalam bentuk coret-coretan, walaupun belum rinci, belum lengkap, dan belum tersusun secara logis dan mungkin orang lain belum dapat ikut memahaminya

3. Membuat **Program** komputer.

```
#include <iostream.h>
void main()
{ float A, T, L;
  A = 7;
  T = 5;
  L = ( A * T ) / 2;
  cout << L;
}
```

Tercetak : 17.5

7.0	5.0	17.5
A	T	L

atau

```
#include <iostream.h>
void main()
{ float A, T;
  A = 7;
  T = 5;
  cout << (A*T) / 2;
}
```

Tercetak : 17.5

atau

```
#include <iostream.h>
void main()
{ cout << (7.0 * 5.0) / 2;
}
```

Tercetak : 17.5

```
#include <iostream.h>
void main()
{ int A, T, L;
  A = 7;
  T = 5;
  L = ( A * T ) / 2;
  cout << L;
}
```

Tercetak : 17



**Salah**, karena semua variabel ataupun nilai menggunakan tipe **int**

### 3. Membuat **Program** komputer.

Setelah jalan pikiran tertuang dalam bentuk **algoritma**, maka algoritma tersebut dapat diberikan ke komputer sebagai langkah-langkah penyelesaian pekerjaan. Untuk itu algoritma tersebut harus dituangkan kedalam suatu **bahasa pemrograman**, misal Bahasa C++. Algoritma yang ditulis dalam suatu bahasa pemrograman, disebut **program**.

Bila algoritma dituangkan kedalam Bahasa C++, maka harus diketahui karakteristik Bahasa C++ tersebut, antara lain :

Jenis dan syntax instruksi termasuk Fungsi Pustaka  
Tipe data dan penggunaannya.  
Identifier dan penggunaannya.

#### Harus paham bahwa :

Ditulis 7, bukan ditulis 7 cm  
Tercetak 17.5 satuannya adalah cm persegi

#### Harus paham mengapa :

- . Tercetak 17.5 bukan 17.50  
harus paham format cetakan
- . Tercetak 17, padahal seharusnya 17.5  
harus paham tipe data dan operasinya

#### Harus paham bahwa :

- . Banyak cara menulis program untuk maksud yang sama.

Angka 7 dibuat menjadi 7.0 agar tipenya Floating Point.  
Dalam operasi arithmetic, bila satu saja nilai bertipe float, maka hasil operasinya bertipe float.  
Bila semuanya bertipe int maka hasilnya bertipe int

```
#include <iostream.h>
void main()
{ int A, T;
  float L;
  A = 7;
  T = 5;
  L = 0.5 * ( A * T );
  cout << L ;
}
```

Tercetak : 17.5

Dalam menghitung L digunakan angka 0.5  
Nilai ini adalah Floating Point.  
Dalam operasi arithmetic, bila satu saja nilai bertipe float, maka hasil operasinya bertipe float.  
Bila semuanya bertipe int maka hasilnya bertipe int

# Beberapa Contoh Keterkaitan Pembuatan Algoritma dengan Pengetahuan Pendukung.

## Soal-6

Susun algoritma (dalam Bahasa C++) untuk menginput sebuah nilai yang menyatakan waktu dalam detik, kemudian cetaklah waktu tersebut dalam bentuk :  
**Jam, Menit dan Detik.**

Modal utama memahami soal ini adalah pemahaman mengenai :

1 jam = 60 menit,  
1 menit = 60 detik, dan  
1 jam = 3600 detik

Paham ?

Paham bahwa :

1 jam = 60 menit,  
1 menit = 60 detik, dan  
1 jam = 3600 detik

Paham saja tidak cukup,

Paham mengenai ini, belum tentu bisa menghitung sekian Detik = berapa jam, dan berapa menit, dan berapa detik

Tidak paham ini, tidak bisa menghitung.

Tidak bisa menghitung, tidak bisa membuat algoritma

Bagaimana bila diinput :

7425



● Bisakah menduga apa yang seharusnya tercetak : ?

● Bisakah menghitung :

7425 Detik = .... Jam, ..... Menit, ..... Detik ?

Bagaimana menghitung 7425 detik = 2 jam, 3 menit, dan 45 detik,

Ada yang menulis coret-coretan pada selembar kertas sebagai berikut :

Lalu ia mengatakan :

2 jam  
3 menit  
45 detik

3600 / 7425 \ 2  
7200  
60 / 225 \ 3  
180  
45

Tidak rinci, misal tidak dirinci bagaimana mendapatkan 7200

Orang lain belum tentu mengerti dari mana mendapatkan angka 7200

Ada juga yang menulis coret-coretan pada selembar kertas sebagai berikut

7425 / 3600 = 2 sisa 225  
225 / 60 = 3 sisa 45

Jadi :

2 Jam  
3 Menit, dan  
45 Detik

Kedua-duanya belum rinci, seharusnya dibuat sebagai berikut :

7425 / 3600 = 2  
2 \* 3600 = 7200  
7425 - 7200 = 225  
225 / 60 = 3  
3 \* 60 = 180  
225 - 180 = 45

Jadi: 7425 detik =  
2 Jam  
3 Menit  
45 Detik

Tidak rinci, misal bagaimana mendapatkan 225

Rinci, Lengkap, dan Tersusun logis

Inilah jalan pikiran yang akan dituangkan kedalam bentuk algoritma

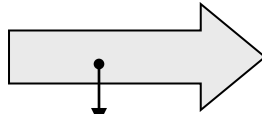
Ada 6 langkah

Proses berpikir dengan urutan seperti ini adalah orsinil milik tiap-tiap orang.

Inilah **Algoritma**, yang ada dalam pemikiran kita.

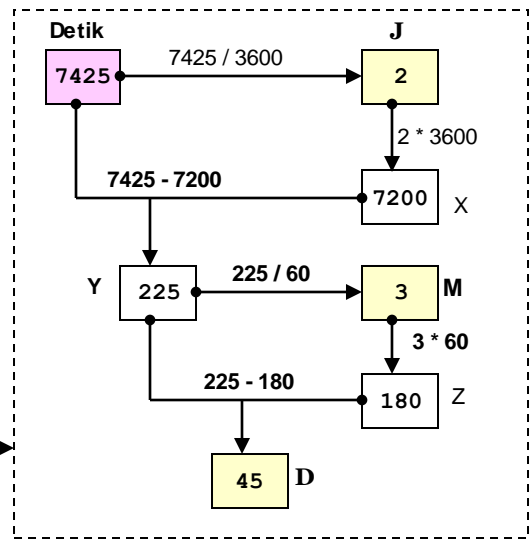
**Algoritma** (alur pikiran) dalam block diagram

- Hasil tahap 1.
1.  $7425 / 3600 = 2$
  2.  $2 * 3600 = 7200$
  3.  $7425 - 7200 = 225$
  4.  $225 / 60 = 3$
  5.  $3 * 60 = 180$
  6.  $225 - 180 = 45$



Mencontoh pembuatan algoritma menghitung luas segitiga, dimana setiap nilai disimpan dalam sebuah variabel, maka algoritma ini dapat digambarkan secara block diagram sebagai berikut:

**Diagram-1.**



Tahap

2.

Konsep pemikiran diatas dituangkan kedalam bentuk algoritma

### Algoritma

```
Input Detik
J = Detik / 3600
X = J * 3600
Y = Detik - X
M = Y / 60
Z = M * 60
D = Y - Z
Print J, M, D
```

Tahap

3.

Algoritma ini ditulis meniru jalan pikiran yang tertuang dalam block Diagram-2

### Program dalam Bahasa C++.

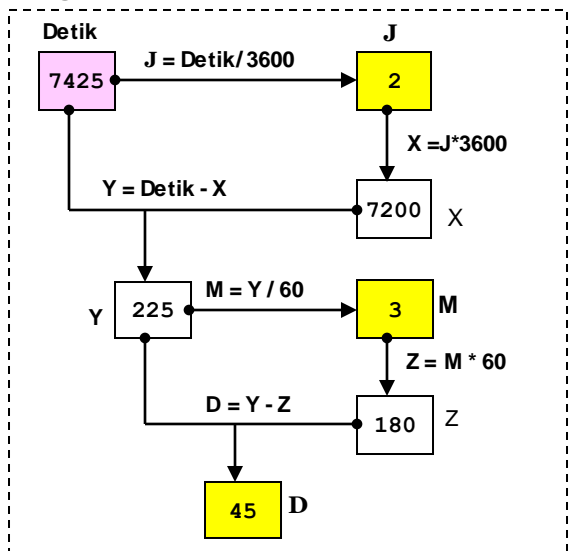
```
#include <iostream.h>
void main()
{ int Detik, J,M,D, X,Y,Z;
  cin >> Detik;
  J = Detik / 3600;
  X = J * 3600;
  Y = Detik - X;
  M = Y / 60;
  Z = M * 60;
  D = Y - Z;
  cout << J << " " << M << " " << D;
```

2

3

45

**Diagram-2.**



Tahap

2.

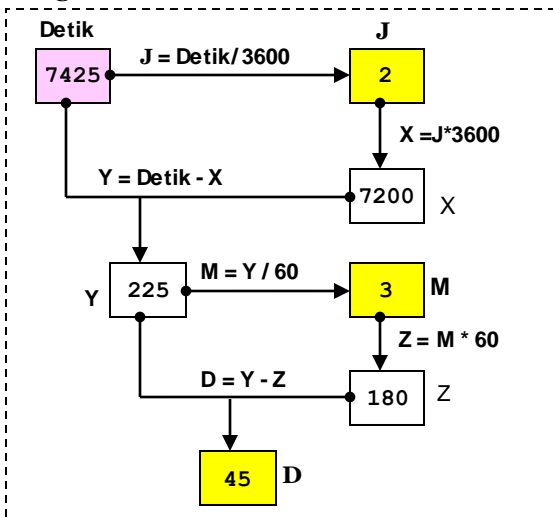
Tuliskan Algoritmanya

Tahap

3.

Tuliskan Programnya

Diagram-2.

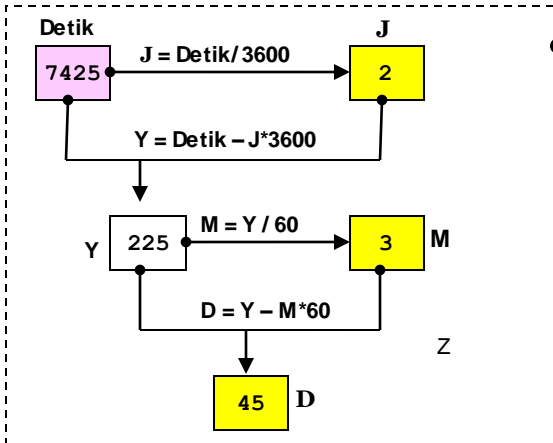


```

#include <iostream.h>
void main()
{ int Detik,J,M,D, X,Y,Z;
  cin >> Detik;
  J = Detik / 3600;
  X = J * 3600;
  Y = Detik - X;
  M = Y / 60;
  Z = M * 60;
  D = Y - Z;
  cout << J << " " << M << " " << D;
}
  
```

Bila diinput 7425  
Maka tercetak: 2 3 45

Diagram-3.

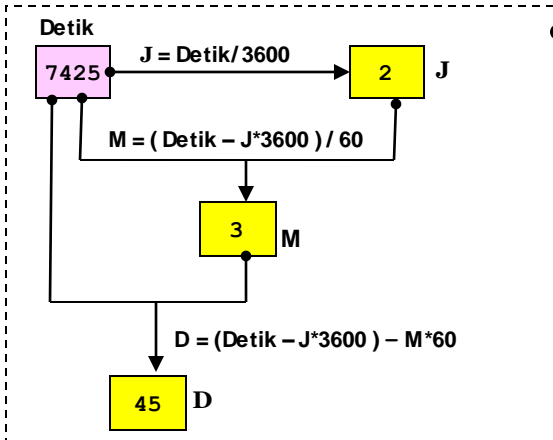


Tidak lagi menggunakan variabel X dan Z

```

#include <iostream.h>
void main()
{ int Detik,J,M,D,Y;
  cin >> Detik;
  J = Detik / 3600;
  Y = Detik - J*3600;
  M = Y / 60;
  D = Y - M*60;
  cout << J << " " << M << " " << D;
}
  
```

Diagram-4.



Tidak lagi menggunakan variabel X, Y dan Z

```

#include <iostream.h>
void main()
{ int Detik,J,M,D;
  cin >> Detik;
  J = Detik / 3600;
  M = (Detik - J*3600) / 60;
  D = (Detik - J*3600) - M*60;
  cout << J << " " << M << " " << D;
}
  
```

```
#include <iostream.h>
void main()
{ int Detik,J,M,D, X,Y,Z;
  cin >> Detik;
  J = Detik / 3600;
  X = J * 3600;
  Y = Detik - X;
  M = Y / 60;
  Z = M * 60;
  D = Y - Z;
  cout << J << " " << M << " " << D;
}
```

**ini yang paling mudah dipahami**

Dua instruksi ini dapat digabung menjadi satu dengan men-substitusi nilai X, Sehingga menjadi :  $Y = Detik - (J * 3600);$

Y ini sebenarnya adalah sisa pembagian D / 3600, yang dapat dihitung dengan :

$Y = Detik \% 3600;$

Dua instruksi ini dapat digabung menjadi satu dengan meng-substitusi nilai Z, Sehingga menjadi :

$D = Y - (M * 60);$

Atau :  $D = Y \% 60;$

Atau disingkat menjadi :

```
#include <iostream.h>
void main()
{ int Detik,J,M,D, Y;
  cin >> Detik;
  J = Detik / 3600;
  Y = Detik - (J*3600);
  M = Y / 60;
  D = Y - (M * 60);
  cout << J << " " << M << " " << D;
}
```

Disingkat lagi menjadi

```
#include <iostream.h>
void main()
{ int Detik, J,M,D, Y;
  cin >> Detik;
  J = Detik / 3600;
  M = (Detik - (J*3600)) / 60;
  D = (Detik - (*3600)) - (M*60);
  cout << J << " " << M << " " << D;
}
```

Disingkat lagi menjadi

```
#include <iostream.h>
void main()
{ int Detik,J,M,D;
  cin >> Detik;

  J = Detik / 3600;
  M = ( Detik - (Detik/3600*3600)) / 60;
  D = ( Detik - (Detik/3600*3600)) - ((( Detik - (Detik/3600*3600))/60)*60 );

  cout << J << " " << M << " " << D;
}
```

**Selain harus berpikir keras, kemungkinan salah tulis sangat besar.**

**Komputer-time pun lebih besar karena untuk setiap nilai, selalu harus menghitung ulang**

```
#include <iostream.h>
void main()
{ int Detik,J,M,D, Y;
  cin >> Detik;
  J = Detik / 3600;
  Y = Detik \% 3600;
  M = Y / 60;
  D = Y \% 60;
  cout << J << " " << M << " " << D;
}
```

Catatan :  $J = 7425 / 3600 = 2$   
 $Y = 7425 \% 3600 = 225$   
 (= sisa 7425/3600 )  
 $M = 225 / 60 = 3$   
 $D = 225 \% 60 = 45$   
 (= sisa 225 / 60 )



```
#include <iostream.h>
void main()
{ int Detik,J,M,D;
  cin >> Detik;
  J = Detik / 3600;
  M = ( Detik - (Detik/3600*3600)) / 60;
  D = ( Detik - (Detik/3600*3600)) - ((( Detik - (Detik/3600*3600))/60)*60 );
  cout << J << " " << M << " " << D;
}
```

**Selain harus berpikir keras,  
kemungkinan salah tulis sangat besar.**

**Komputer-time pun lebih besar  
karena untuk setiap nilai, selalu harus  
menghitung ulang**

Atau tidak menggunakan variabel lain selain **Detik**

```
#include <iostream.h>
void main()
{ int Detik;
  cin >> Detik;
  cout << Detik/3600 << " " << Detik-( Detik/3600*3600) )/60 << " " <<
    (Detik-( Detik/3600*3600))-(( Detik - (Detik/3600*3600))/60)*60 ) );
}
```

**Yang ini, lebih rumit lagi**

Tentu saja kita tidak akan memilih cara ini, walaupun benar.

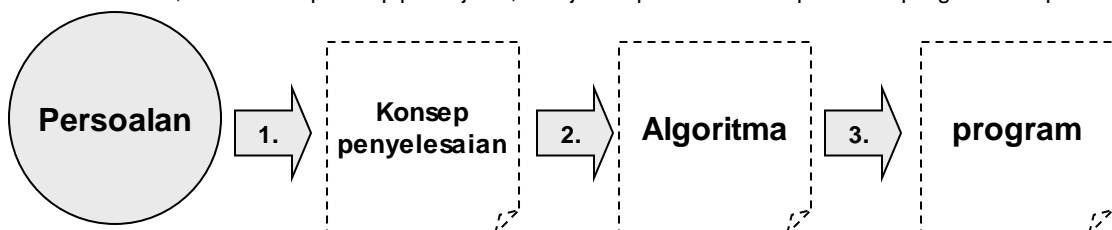
Copy dari halaman sebelumnya

```
#include <iostream.h>
void main()
{ int Detik,J,M,D, Y;
  cin >> Detik;
  J = Detik / 3600;
  Y = Detik % 3600;
  M = Y / 60;
  D = Y % 60;
  cout << J << " " << M << " " << D;
}
```

Bila tidak menggunakan variabel Y

```
#include <iostream.h>
void main()
{ int Detik,J,M,D;
  cin >> Detik;
  J = Detik / 3600;
  M = ( Detik % 3600 ) / 60;
  D = (Detik % 3600) % 60;
  cout << J << " " << M << " " << D;
}
```

Dari contoh diatas, terlihat tahap-tahap pekerjaan, menjawab persoalan sampai ditulis program komputer.



## Soal-7

Susun program (dalam Bahasa C++) untuk menginput dua buah bilangan integer yang nilainya berbeda satu dengan yang lainnya. Kemudian cetak kedua bilangan tersebut, dimana yang dicetak lebih dulu adalah bilangan yang lebih besar, kemudian bilangan yang lebih kecil.

Contoh :

1. Bila diinput: 6 dan 2  
Maka yang dicetak: 6 2
2. Bila diinput: -5 dan 2  
Maka yang dicetak: 2 -5

Konsep penyelesaian :

1. Untuk bilangan 6 dan 2.

$$\begin{aligned} 6 + 2 &= 8 \\ \text{Absolut } (6 - 2) &= 4 \\ 8 + 4 &= 12 \\ 12 / 2 &= 6 \rightarrow \text{besar} = 6 \end{aligned}$$

atau

$$\begin{aligned} 2 + 6 &= 8 \\ \text{Absolut } (2 - 6) &= 4 \\ 8 + 4 &= 12 \\ 12 / 2 &= 6 \rightarrow \text{besar} = 6 \end{aligned}$$

Untuk mencari nilai yang terkecil :

$$\begin{aligned} 8 - 4 &= 4 \\ 4 / 2 &= 2 \rightarrow \text{kecil} = 2 \end{aligned}$$

2. Untuk bilangan -5 dan 2.

$$\begin{aligned} -5 + 2 &= -3 \\ \text{Absolut } (-5 - 2) &= 7 \\ -3 + 7 &= 4 \\ 4 / 2 &= 2 \rightarrow \text{besar} = 2 \end{aligned}$$

atau

$$\begin{aligned} 2 + -5 &= -3 \\ \text{Absolut } (2 - -5) &= 7 \\ -3 + 7 &= 4 \\ 4 / 2 &= 2 \rightarrow \text{besar} = 2 \end{aligned}$$

Untuk mencari nilai yang terkecil :

$$\begin{aligned} -3 - 7 &= -10 \\ -10 / 2 &= -5 \rightarrow \text{cetak: } -5 \end{aligned}$$

**Catatan :** Dari cara penyelesaian diatas, terlihat bahwa beberapa masalah, terutama yang menyangkut angka, atau hitung menghitung, maka pengetahuan matematika sering mempengaruhi cara penyelesaian. Sebenarnya secara teknik algoritma yang biasa digunakan untuk penyelesaian diatas, orang lebih suka menggunakan algoritma sebagai berikut :

```

INPUT A,B;
IF(A > B) THEN
    PRINT A,B;
ELSE
    PRINT B,A;
ENDIF
    
```

Bila A lebih besar dari B, maka cetak A, B, Tapi bila tidak, cetak B,A

Yang dalam Bahasa C++ ditulis :

```

#include <iostream.h>
void main()
{ int A, B;
  cin >> A >> B;
  if(A > B)
      cout << A << " " << B;
  else
      cout << B << " " << A;
}
    
```

Algoritma seperti ini akan kita bahas setelah mempelajari **if ( ..... )**

Setelah melihat konsep penyelesaian, langsung saja membuat program dalam Bahasa C++.

```

#include <iostream.h>
#include<math.h>
void main()
{ int A,B, Total, Selisih;
  int Besar, Kecil;
  cin >> A >> B;
  Total = A + B;
  Selisih = abs(A - B);
  Besar = (Total + Selisih ) / 2;
  Kecil = (Total - Selisih ) / 2;
  cout << Besar << " " << Kecil );
}
    
```

**abs**

adalah fungsi pustaka yang penggunaannya harus meng-include file **math.h** (lihat bab 7 Fungsi Pustaka halaman 79 )

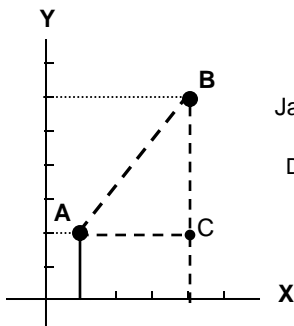
## Soal-8

Dalam sebuah bidang datar yang direpresentasikan atau dipetakan dengan koordinat sumbu **X** dan **Y** dimana sebuah titik **P** dinyatakan dengan **P(x,y)**, terdapat titik **A(x<sub>1</sub>,y<sub>1</sub>)** dan titik **B(x<sub>2</sub>,y<sub>2</sub>)**.

Susun program (dalam Bahasa C++) untuk menginput 4 buah bilangan, yang masing-masing secara berurutan, menyatakan x<sub>1</sub>,y<sub>1</sub>,x<sub>2</sub>, dan y<sub>2</sub>. Kemudian hitung dan cetak jarak kedua titik A dan B tersebut.

### Pemahaman

Bila A(1,2) dan B(4,6), maka jarak AB dapat diilustrasikan dengan gambar berikut:



$$\text{Jarak AB} = \sqrt{AC^2 + BC^2}$$

Dimana  $AC = 4 - 1 = 3$   
dan  $BC = 6 - 2 = 4$

$$\text{Jarak AB} = \sqrt{3.0^2 + 4.0^2} = 5.0$$

Setelah melihat konsep penyelesaian, langsung saja membuat program dalam Bahasa C++.

```
#include <iostream.h>
#include<math.h>
void main()
{ float x1, y1, x2,y2,Jarak;
  float ACkwd, BCkwd;
  cin >> x1 >> y1;
  cin >> x2 >> y2;
  ACkwd = pow( (x2 - x1), 2);
  BCkwd = pow( (y2 - y1), 2);
  Jarak = sqrt( ACkwd + BCkwd);
  cout << Jarak;
}
```

Bila diinput : 1 2      Maka tercetak : 5  
                  4 6

Karena hasil akar, kemungkinan bukan bilangan bulat (integer), maka semua data dibuat bertipe **float**

## Soa-9.

### Program Sederhana

Untuk uang kembalian, terdapat uang logam (koin) dalam 4 macam nilai, yaitu 25 sen, 10 sen, 5 sen dan 1 sen.

Susun program untuk menginput sebuah nilai integer, misal N, ( bernilai 1 - 99 ) yang menyatakan jumlah uang kembalian dalam sen. Kemudian cetak jumlah keping uang logam yang diperlukan.

Dari pengalaman menggunakan **operator %** (modulus) pada konversi 7425 detik = ... Jam, ... Menit, dan ... Detik, langsung dapat ditulis program yang diperlukan sebagai berikut:

```
#include <iostream.h>
void main()
{ int N, C25, C10, C5, C1 ;
  cin >> N ;
  C25 = N / 25;    N = N % 25;
  C10 = N / 10;    N = N % 10;
  C5 = N / 5;     N = N % 5;
  C1 = N;
  cout << "\n 25 sen = " << C25;
  cout << "\n 10 sen = " << C10;
  cout << "\n 5 sen = " << C5;
  cout << "\n 1 sen = " << C1;
}
```

Misal untuk N = <b>64</b> Akan tercetak :	Misal untuk N = <b>77</b> Akan tercetak :	Misal untuk N = <b>10</b> Akan tercetak :
25 sen = 2	25 sen = 3	25 sen = 0
10 sen = 2	10 sen = 0	10 sen = 1
5 sen = 0	5 sen = 0	5 sen = 0
1 sen = 4	1 sen = 2	1 sen = 0

## Soal-10

Sebuah fungsi kuadrat:

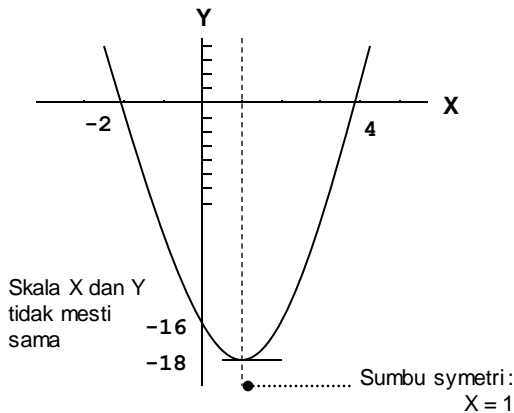
$$y = ax^2 + bx + c$$

Yang dapat dipetakan dalam bidang XY, (lihat Soal-8a). dibuat sedemikian rupa sehingga dipastikan memotong sumbu X dan Y.

Susun program untuk mencari dan mencetak titik-titik pada sumbu X dan Y yang dipotong oleh fungsi :

$$y = 2x^2 - 4x - 16$$

Ilustrasi fungsi dalam bentuk grafik pada sumbu XY:



$$\begin{aligned} X1 &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} \\ X1 &= \frac{-(-4) + \sqrt{(-4)^2 - 4 \cdot 2 \cdot (-16)}}{2 \cdot 2} \\ &= \frac{4 + \sqrt{16 + 128}}{4} \\ &= \frac{4 + \sqrt{144}}{4} \\ &= \frac{4 + 12}{4} \\ &= 4 \end{aligned}$$

Komputer mencetak: 4

$$\begin{aligned} X2 &= \frac{4 - 12}{4} \\ &= -2 \end{aligned}$$

Memotong sumbu Y bila  $x = 0$

$$y = 2x^2 - 4x - 16 \rightarrow \text{Untuk } X = 0, \text{ Didapat } Y = -16$$

$$y = 2x^2 - 4x - 16$$

$$y' = 4x - 4 \rightarrow \text{Titik minimum } y' = 0$$

Ini tercapai untuk  $x = 1$ .  
Jadi sumbu symetri  $X = 1$ .  
Untuk  $x = 1$  didapat  $y = -18$

Pemahaman :

$$y = 2x^2 - 4x - 16$$

Bila ditulis dalam bentuk umum dapat ditulis sebagai berikut :

$$y = (2)x^2 + (-4)x + (-16)$$

Memotong sumbu X berarti  $y = 0$ , sehingga

$$(2)x^2 + (-4)x + (-16) = 0$$

Titik potongnya dengan sumbu X dapat dicari dengan rumus **abc**

Sebagai berikut :

$$X1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$X2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

$$\begin{aligned} a &= 2 \\ b &= -4 \\ c &= -16 \end{aligned}$$

Dimana dalam hal ini, untuk soal yang diberikan :

Program :

```
#include <iostream.h>
#include<math.h>
void main()
{ float a,b,c,D, akarD, x1,x2;
  a = 2; b = -4; c = -16;
  D = b*b - (4 * a * c);
  akarD = sqrt(D);
  x1 = ( -b + akarD ) / ( 2*a);
  x2 = ( -b - akarD ) / ( 2*a);
  cout << "\n X1 = " << x1;
  cout << "\n X2 = " << x2;
}
```

Tercetak :  $x1 = 4$   
 $x2 = -2$

Karena dalam perhitungan menggunakan akar dan pembagian, maka digunakan tipe **float**

Untuk contoh ini, terlihat bahwa :

Memahami masalah, dan mendesign jawaban, terkadang lebih sulit dan membutuhkan waktu lama dibandingkan dengan menulis **algoritma** atau menulis **code program**

selesai  
Sessi-2