

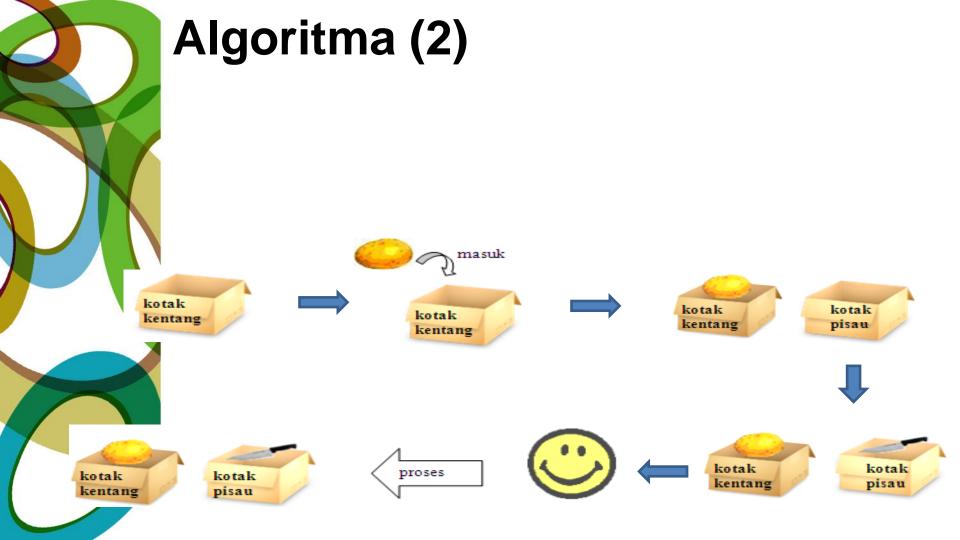


#### Rosa Ariani Sukamto

- Blog: <a href="http://hariiniadalahhadiah.wordpress.com">http://hariiniadalahhadiah.wordpress.com</a>
- Facebook: <a href="https://www.facebook.com/rosa.ariani.sukamto">https://www.facebook.com/rosa.ariani.sukamto</a>
- Email: <u>rosa.ariani@upi.edu</u>
- Website: <a href="https://rosa-as.id">https://rosa-as.id</a>



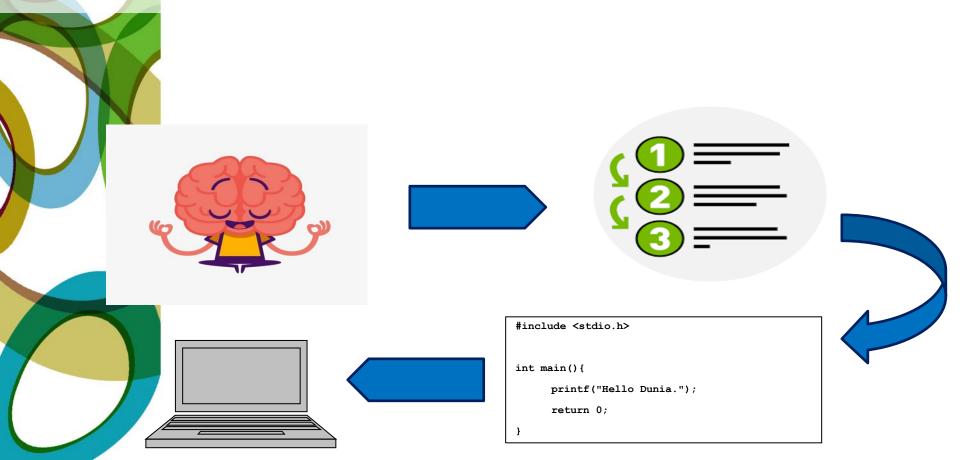
- Semua orang sebenarnya banyak memikirkan mengenai algoritma
- Algoritma adalah mengenai membuat solusi atas permasalahan dengan runut
- Algoritma sebenarnya sudah sering digunakan dalam kehidupan sehari-hari, hanya saja orang sering tidak mengingatnya dengan runut dan detail
- Dalam membuat sebuah program komputer dibutuhkan solusi permasalahan yang runut (algoritma)



## Pemrograman

- Pemrograman adalah proses untuk membuat sebuah program komputer
- Membuat program dilakukan dengan memetakan algoritma ke dalam bahasa pemrograman
- Misalkan membuat program untuk menampilkan kalimat, maka yang harus dimengerti adalah bagaimana cara bahasa pemrograman dalam menampilkan kalimat

#### Konversi Pemikiran – Algoritma - Pemrograman



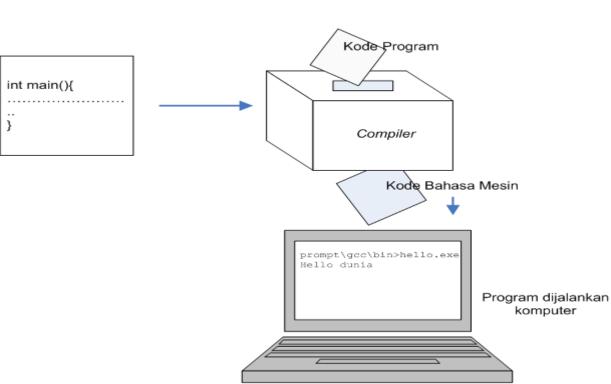


## Pemrograman (2)

```
#include <stdio.h>
int main() {
    printf("Hello Dunia.");
    return 0;
}
```



## Pemrograman (3)



## Pemrograman (4)

- Program dijalankan dengan menggunakan memori komputer
- Sebelum program dibuat, maka dapat dianggap memori komputer masih kosong, kita dapat membayangkannya sebagai lapangan kosong yang luas
- Oleh karena itu segala kebutuhan akan proses harus dibuat atau dideklarasikan terlebih dahulu agar dikenali oleh komputer



## variabel

- Variabel dapat diartikan suatu tempat untuk menyimpan sebuah nilai
- Dalam bayangan atau logika kita, kita bisa menganggap bahwa variabel adalah sebuah kotak untuk menyimpan nilai



 Kotak bernama "bilbulat" menyimpan sebuah nilai bilangan bulat yaitu angka 9

## Tipe data

- integer bilangan bulat (tidak memiliki angka di belakang koma)
- float bilangan riil (dapat menyimpan angka di belakang koma)
- char karakter

## integer

Dalam bahasa pemrograman C, deklarasi variabel integer adalah sebagai berikut:

```
int bilBulat = 9;
```

 Atau bisa juga dengan hanya mendeklarasikan kotaknya saja tanpa nilai

bilbulat

bilbulat

int bilBulat;

Jika hanya dideklarasikan tanpa nilai, maka sebenarnya nilai yang ada di dalam kotak bergantung pada kompilator yang digunakan

## Integer (2)

Jika setelah di deklarasikan, dan kemudian nilai diubah, maka nilai sebelumnya tidak akan disimpan lagi di dalam kotak

int bilBulat = 9;

bilBulat = 12;



bilbulat

### float

• Dalam bahasa pemrograman C, deklarasi variabel float adalah sebagai berikut:

```
float bilRiil = 59.0;
```

 Atau bisa juga dengan hanya mendeklarasikan kotaknya saja tanpa nilai

bilRiil

bilRiil

Jika hanya dideklarasikan tanpa nilai, maka sebenarnya nilai yang ada di dalam kotak bergantung pada kompilator yang digunakan

# float (2)

Jika setelah di deklarasikan, dan kemudian nilai di ubah, maka nilai sebelumnya tidak akan disimpan lagi di dalam kotak

float bilRiil = 59.0;

bilRiil = 99.99;





### char

- Untuk tipe data char, sama halnya dengan pendeklarasian integer maupun float.
- Cara mendeklarasikan adalah

```
tipe_data nama_variabel;
tipe_data nama_variabel = nilai;
```

• Tipe char dideklarasikan dengan menggunakan tanda petik satu (') char karakter = 'A';

## **Operator aritmatika**

- Dalam pemrograman juga dikenal operator aritmatika seperti (dalam bahasa pemrograman C)
  - Tambah (+)
  - Kurang (-)
  - Bagi (/)
  - Kali (\*)
  - Modula (sisa pembagian) (%)

### komentar

- Komentar tidak dieksekusi oleh kompilator
- Komentar hanya digunakan untuk menambahkan keterangan terhadap kode program
- Berikut adalah cara penulisan komentar dalam bahasa C

```
//ini adalah program pertama yang dibuat

/*
nama: Rosa Ariani Sukamto
  deskripsi program: menambahkan dua bilangan
```

#### Menampilkan ke layar (output)

Pada bahasa pemrograman C, menampilkan sesuatu ke layar menggunakan fungsi printf

```
printf ("Wilujeng Sumping.");
```

Menampilkan nilai variabel integer

```
printf("isi variabel bilangan bulat adalah %d\n", bilBulat
);
```

- %d digunakan sebagai penanda posisi nilai variabel bilBulat ditampilkan (khusus integer)
- \n adalah penanda berganti baris (newline)
- Menampilkan nilai variabel float

```
printf("isi variabel bilangan riil adalah %f\n", bilRiil);
```

Menampilkan nilai variabel char

```
printf("isi variabel karakter adalah %c\n", karakter);
```

#### Menerima masukan (input)

Pada bahasa pemrograman C, menerima masukan dari *user* dapat menggunakan fungsi scanf

```
scanf("%d", &bilBulat);
scanf("%f", &bilRiil);
scanf("%c", &bilKarakter);
```

yang berarti masukan user akan dimasukkan ke dalam variabel

- Membuat algoritma untuk menambahkan dua buah bilangan masukan user
- Membuat variabel untuk menampung bilangan pertama





- Membuat algoritma untuk menambahkan dua buah bilangan masukan user
- Membuat variabel untuk menampung bilangan pertama
- Membuat variabel untuk menampung bilangan kedua



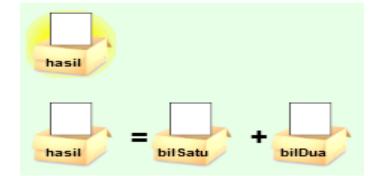
- Membuat algoritma untuk menambahkan dua buah bilangan masukan user
- Membuat variabel untuk menampung bilangan pertama
- Membuat variabel untuk menampung bilangan kedua
- Meminta masukan dari user untuk kedua bilangan







- Membuat algoritma untuk menambahkan dua buah bilangan masukan user
- Membuat variabel untuk menampung bilangan pertama
- Membuat variabel untuk menampung bilangan kedua
- Meminta masukan dari user untuk kedua bilangan
- Membuat variabel untuk menampung hasil pertambahan bilangan pertama dan kedua



- Membuat algoritma untuk menambahkan dua buah bilangan masukan user
- Membuat variabel untuk menampung bilangan pertama
- Membuat variabel untuk menampung bilangan kedua
- Meminta masukan dari user untuk kedua bilangan
- Membuat variabel untuk menampung hasil pertambahan bilangan pertama dan kedua
- Menampilkan hasil pertambahan kedua bilangan





```
#include <stdio.h>
int main() {
    return 0;
}
```



```
#include <stdio.h>
int main() {
   int bilSatu;//Membuat variabel untuk menampung bilangan pertama
   int bilDua;//Membuat variabel untuk menampung bilangan kedua
   return 0;
```





```
#include <stdio.h>
int main(){
         int bilSatu; //Membuat variabel untuk menampung bilangan pertama
         int bilDua; //Membuat variabel untuk menampung bilangan kedua
         //Meminta masukan dari user untuk kedua bilangan
         printf("masukkan bilangan pertama:\n");
         scanf("%d", &bilSatu);
         printf("masukkan bilangan kedua:\n");
         scanf("%d", &bilDua);
                                    "masukkan bilanga
                                                   pertama:\n"
                                                                "masukkan bilanga" kedua:\n"
         return 0;
```

```
#include <stdio.h>
int main(){
         int bilSatu; //Membuat variabel untuk menampung bilangan pertama
         int bilDua; //Membuat variabel untuk menampung bilangan kedua
         //Meminta masukan dari user untuk kedua bilangan
        printf("masukkan bilangan pertama:\n");
         scanf("%d", &bilSatu);
         printf("masukkan bilangan kedua:\n");
        scanf("%d", &bilDua);
         /* Membuat variabel untuk menampung hasil pertambahan bilangan pertama d
an kedua */
         int hasil = bilSatu + bilDua;
         return 0;
                                                         hasil
                                                         hasil
```

```
#include <stdio.h>
int main(){
        int bilSatu; //Membuat variabel untuk menampung bilangan pertama
        int bilDua; //Membuat variabel untuk menampung bilangan kedua
        //Meminta masukan dari user untuk kedua bilangan
        printf("masukkan bilangan pertama:\n");
        scanf("%d", &bilSatu);
        printf("masukkan bilangan kedua:\n");
        scanf("%d", &bilDua);
        /* Membuat variabel untuk menampung hasil pertambahan bilangan pertama d
an kedua */
        int hasil = bilSatu + bilDua;
        // Menampilkan hasil pertambahan kedua bilangan
                                                                  "hasil penambahan
        printf("hasil penambahan adalah %d\n", hasil);
        return 0;
```



#### Hasil eksekusi (1)

masukkan bilangan pertama: 7 masukkan bilangan kedua: 8 hasil penambahan adalah 15

 Membuat algoritma untuk menghitung luas persegi panjang dimana nilai panjang dan lebar diambil dari masukan user



- Membuat algoritma untuk menghitung luas persegi panjang dimana nilai panjang dan lebar diambil dari masukan *user*
- Membuat variabel untuk menampung nilai panjang





- Membuat algoritma untuk menghitung luas persegi panjang dimana nilai panjang dan lebar diambil dari masukan user
- Membuat variabel untuk menampung nilai panjang
- Membuat variabel untuk menampung nilai lebar



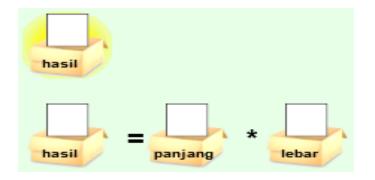
- Membuat algoritma untuk menghitung luas persegi panjang dimana nilai panjang dan lebar diambil dari masukan user
- Membuat variabel untuk menampung nilai panjang
- Membuat variabel untuk menampung nilai lebar
- Meminta masukan dari user untuk kedua bilangan





#### Algoritma studi kasus (2)

- Membuat algoritma untuk menghitung luas persegi panjang dimana nilai panjang dan lebar diambil dari masukan user
- Membuat variabel untuk menampung nilai panjang
- Membuat variabel untuk menampung nilai lebar
- Meminta masukan dari user untuk kedua bilangan
- Membuat variabel untuk menampung hasil kali keduanya karena luas persegi adalah panjang dikali dengan lebar



#### Algoritma studi kasus (2)

- Membuat algoritma untuk menghitung luas persegi panjang dimana nilai panjang dan lebar diambil dari masukan user
- Membuat variabel untuk menampung nilai panjang
- Membuat variabel untuk menampung nilai lebar
- Meminta masukan dari user untuk kedua bilangan
- Membuat variabel untuk menampung hasil kali keduanya karena luas persegi adalah panjang dikali dengan lebar
- Menampilkan hasil kali kedua bilangan



#### Kode program studi kasus (2)

```
#include <stdio.h>
int main(){
        int panjang; //Membuat variabel untuk menampung nilai panjang
        int lebar; // Membuat variabel untuk menampung nilai lebar
        //Meminta masukan dari user untuk kedua bilangan
        printf("masukkan panjang:\n");
        scanf("%d", &panjang);
        printf("masukkan lebar:\n");
        scanf("%d", &lebar);
        /* Membuat variabel untuk menampung hasil kali keduanya karena luas perse
qi adalah panjang dikali dengan lebar */
        int hasil = panjang * lebar;
        // Menampilkan hasil kali kedua bilangan
        printf("hasil perhitungan luas adalah %d\n", hasil);
        return 0;
```



### **HASIL EKSEKUSI (2)**

```
masukkan panjang:
6
masukkan lebar:
8
hasil perhitungan luas adalah 48
```

### Algoritma studi kasus (3)

- Membuat algoritma untuk menampilkan bilangan di belakang (2 angka)
   dan di depan koma dari float masukan
- Membuat variabel untuk menampung nilai float
- Membuat variabel untuk menampung nilai depan float
- Membuat variabel untuk menampung nilai belakang float
- Meminta masukan dari *user* untuk bilangan float
- Mengambil bagian depan float
- Mengambil bagian belakang float
- Menampilkan bagian depan float
- Menampilkan bagian belakang float
- (perhatikan kemampuan float menyimpan angka desimal)

#### Kode program studi kasus (3)

```
#include <stdio.h>
int main(){
  float bil; // Membuat variabel untuk menampung nilai float
  int depan; // Membuat variabel untuk menampung nilai depan float
  int belakang; //Membuat variabel untuk menampun nilai belakang float
  //Meminta masukan dari user untuk bilangan float
        printf("masukkan angka float:\n");
        scanf("%f", &bil);
        depan = bil;// Mengambil bagian depan float
        belakang = (bil - depan) * 100; //Mengambil bagian depan float
   //Menampilkan bagian depan float
        printf("bilangan depan %d\n", depan);
   //Menampilkan bagian belakang float
        printf("bilangan belakang %d\n", belakang);
        return 0;
```



#### Hasil eksekusi (3)

```
masukkan angka float:
5888.98
bilangan depan 5888
bilangan belakang 97
```

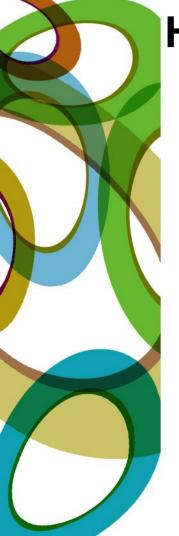
masukkan angka float: 899.90 bilangan depan 899 bilangan belakang 90

# Algoritma studi kasus (4)

- Membuat algoritma untuk menampilkan akar kuadrat dan hasil kuadrat dari sebuah bilangan integer
- Membuat variabel untuk menampung nilai integer
- Membuat variabel untuk menampung nilai akar kuadrat
- Membuat variabel untuk menampung nilai kuadrat
- Meminta masukan dari user untuk bilangan integer
- Menghitung akar kuadrat
- Menghitung kuadrat
- Menampilkan hasil akar kuadrat
- Menampilkan hasil kuadrat

#### Kode program studi kasus (4)

```
#include <stdio.h>
#include <math.h>
int main(){
        int bil;//Membuat variabel untuk menampung nilai integer
        float akar; // Membuat variabel untuk menampung nilai akar kuadrat
         int kuadrat; //Membuat variabel untuk menampung nilai kuadrat
        //Meminta masukan dari user untuk bilangan integer
        printf("masukkan angka integer:\n");
        scanf("%d", &bil);
        akar = sqrt(bil);//Menghitung akar kuadrat
        kuadrat = bil * bil; //Menghitung kuadrat
        //Menampilkan hasil akar kuadrat
        printf("bilangan akar %0.2f\n", akar);
        //Menampilkan hasil kuadrat
        printf("bilangan kuadrat %d\n", kuadrat);
        return 0;
```



# Hasil eksekusi (4)

masukkan angka integer: 56 bilangan akar 7.48 bilangan kuadrat 3136

## Mari mencoba

- Membuat algoritma dan program untuk menghitung luas segitiga
- Membuat algoritma dan program untuk menghitung jarak dua titik dengan masukan x1, y1, x2, y2 dimana jarak dua titik memiliki rumus

$$\sqrt{(x1-x2)^2}+(y1-y2)^2$$

 Membuat algoritma dan program untuk menampilkan angka satuan, p uluhan, ratusan, dan ribuan dari sebuah angka integer masukan, misal nya masukan adalah 5678

maka akan menampilkan

satuan: 8

puluhan: 7

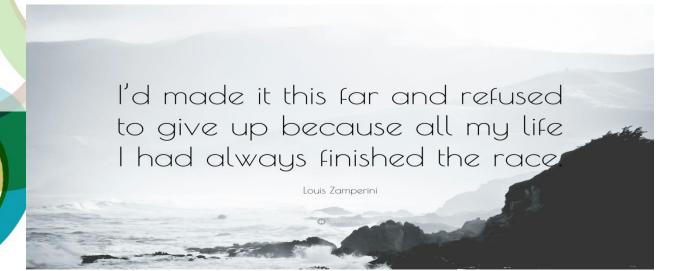
ratusan: 6

ribuan: 5

#### Rahasia Algoritma dan pemrograman

Aksi setiap minggu dengan latihan....

Latihan... latihan....



# Daftar pustaka



#### dan **PEMROGRAMAN**

Masukan (input) dan Keluaran (Output)

Algoritma berarti solusi. Ketika orang berbicara mengenai algoritma di bidang pemrograman, maka yang dimaksud adalah solusi dari suatu masalah yang harus dipecahkan dengan menggunakan komputer. Algoritma harus dibuat secara runut agar komputer mengerti dan mampu mengeksekusinya. Analisis kasus sangat dibutuhkan dalam membuat sebuah algoritma, misalnya proses apa saia yang sekiranya dibutuhkan untuk menyelesaikan masalah yang harus diselesaikan. Ketajaman dalam menganalisis sebuah kasus dapat dilatih dengan berlatih menyelesaikan kasus-kasus algoritma, mulai dari yang paling sederhana sampai rumit, Kesabaran sangat dibutuhkan dalam mempelajari algoritma.

Esensi dari belajar algoritma adalah membuat solusi untuk menyelesaikan permasalahan, jadi hasilnya adalah dapat menyelesaikan permasalahan. Maka sangat tidak sesuai dengan esensi jika algoritma dipelajari dengan hanya tahu polapola atau teknik-teknik algoritma tapi tidak mampu secara logika menyelesaikan permasalahan yang berbeda-beda. Maka dari itu sangat dibutuhkan latihan soal dalam mempelajari algoritma dan pemrograman agar logika terlatih untuk membuat solusi dari permasalahan.

Penulis berharap buku ini dapat membantu para pembaca pada umumnya dan mahasiswa, siswa SMP, SMK, SMA khususnya dalam memahami pemrograman secara lebih baik. Dalam buku ini banyak diberikan penggambaran/ilustrasi secara visual agar pembaca lebih mudah dalam memahami isi buku. Selain itu, dalam buku ini juga diberikan bagaimana mengimplementasikan suatu algoritma dalam bahasa algoritmik, bahasa Pascal, bahasa C, C++, dan Java agar pembaca dapat lebih mudah mengimplementasikannya secara langsung.



Pasar buku Palasari 82 Bandung 40264 Tel (022) 7317812 Fax (022) 7317896

Rosa

**ALGORITI** 

**PEMROGRAMAN** 

#### Rosa A. S.

- □ Pendahuluan Algoritma dan Pemrograman
- Masukan (Input) dan Keluaran (Output) □ Operator
- ☐ Array (Larik)
- □ Percabangan / Pemilihan If
- Perulangan (Looping) ■ Prosedur
- Matriks dalam Larik (Array) Dua Dimensi ■ Rekursif
- ☐ Pengurutan (Sorting)
- Penggabungan Larik
  Pencarian (Searching)
- Arsip Beruntun (Sequential File)
   Mesin Abstrak

