

Algoritma dan Pemrograman I

Kondisi Komparasi (Perbandingan), Pemilihan dan IF Pemilihan Tersarang

Rosa A. S.



Rosa Ariani Sukamto

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



Pemilihan

- Pemilihan dalam pemrograman sering disebut dengan if
- Pemilihan digunakan jika ada lebih dari satu syarat atau kondisi tertentu yang perlu dipenuhi untuk melakukan proses



Operator Logika

- dan / and dalam bahasa C ditulis dengan `&&`
- Semua syarat yang dihubungkan dengan operator `&&` harus bernilai benar / *true*, jika ada yang bernilai salah atau tidak memenuhi maka hasilnya akan dianggap bernilai salah
- atau / or dalam bahasa C ditulis dengan `||`
- Salah satu syarat yang dihubungkan dengan operator `||` harus bernilai benar / *true*, jika semua bernilai salah atau tidak memenuhi maka hasilnya akan dianggap bernilai salah



Operator Pembandingan

- $a == b$ maksudnya jika a sama dengan b
- $a >= b$ maksudnya jika a lebih besar sama dengan b
- $a <= b$ maksudnya jika a lebih kecil sama dengan b
- $a != b$ maksudnya jika a tidak sama dengan b

Pemilihan Satu Kondisi

Penulisan dalam bahasa C

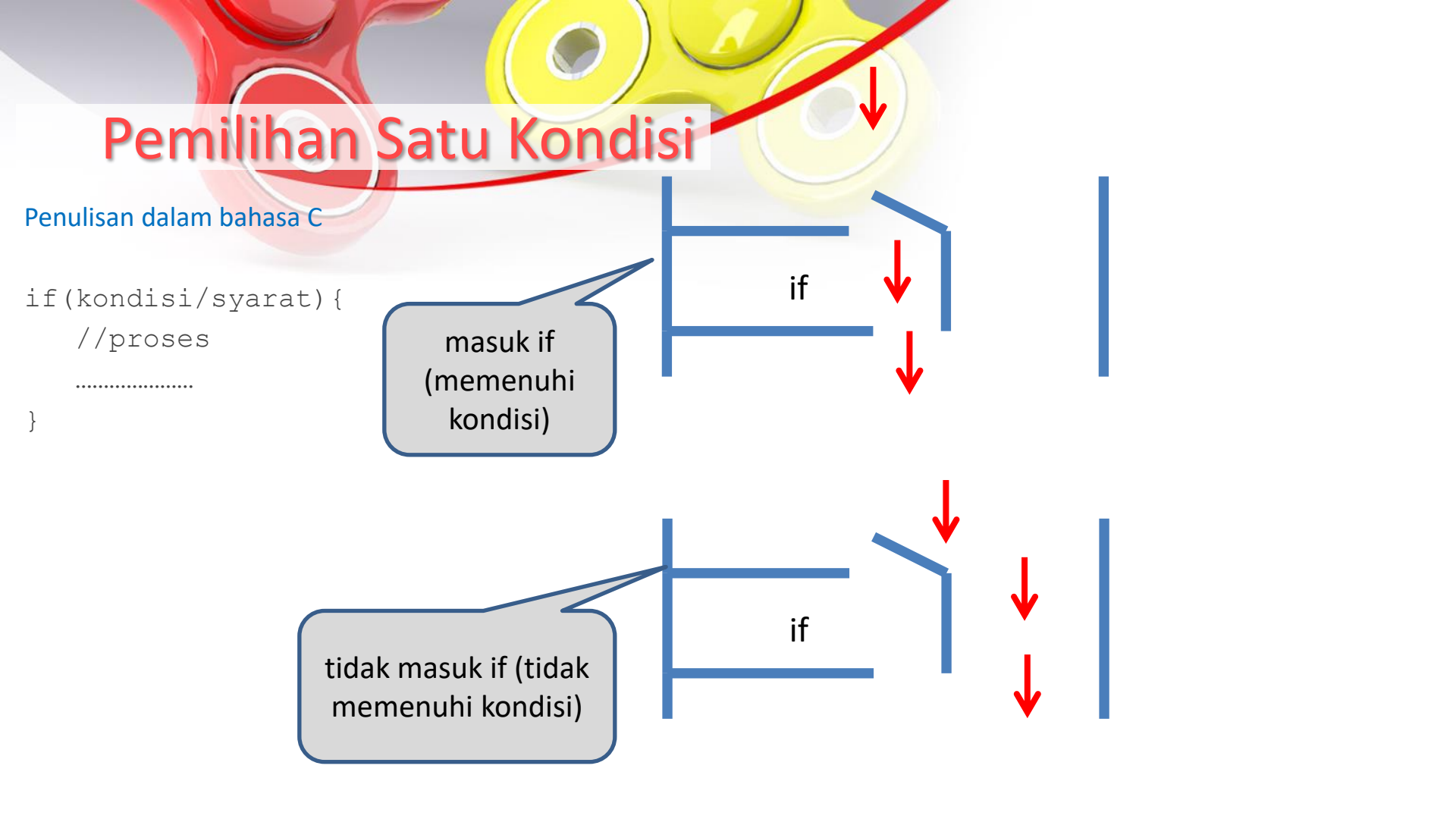
```
if(kondisi/syarat) {  
    //proses  
    .....  
}
```

masuk if
(memenuhi
kondisi)

if

tidak masuk if (tidak
memenuhi kondisi)

if





Pemilihan Satu Kondisi (2)

```
#include <stdio.h>

int main(){
    printf("masukkan angka:\n");
    int nilai;
    scanf("%d", &nilai);

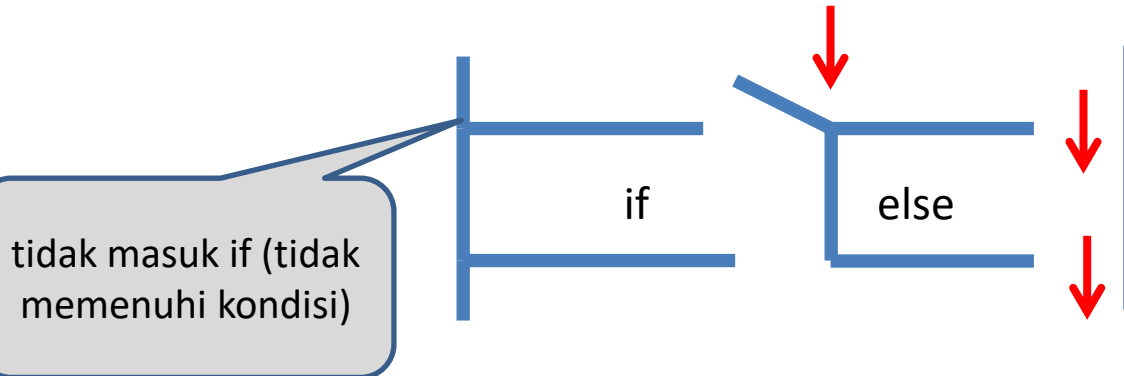
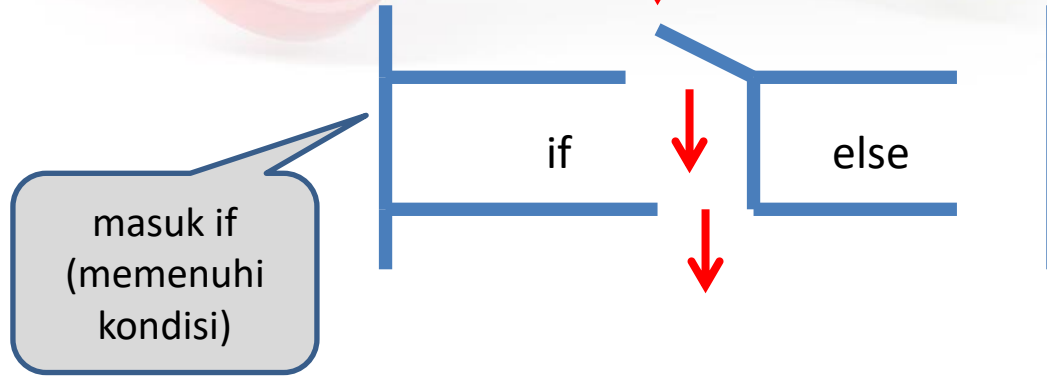
    if(nilai >= 65){
        printf("lulus");
    }
    return 0;
}
```

- Dari kode program di atas, maka tulisan “lulus” akan ditampilkan jika *user* memasukkan angka lebih besar atau sama dengan 65. Jika lebih kecil dari 65 maka tulisan “lulus” tidak akan ditampilkan ke layar

Pemilihan Dua Kondisi

Penulisan dalam bahasa C

```
if(kondisi/syarat){  
    //proses  
    .....  
}else{  
    //proses  
    .....  
}
```





Pemilihan Dua Kondisi (2)

```
#include <stdio.h>
```

```
int main(){
```

```
    printf("masukkan angka:\n");
```

```
    int nilai;
```

```
    scanf("%d", &nilai);
```

```
    if(nilai >= 65){
```

```
        printf("lulus");
```

```
    }else{
```

```
        printf("tidak lulus");
```

```
    }
```

```
    return 0;
```

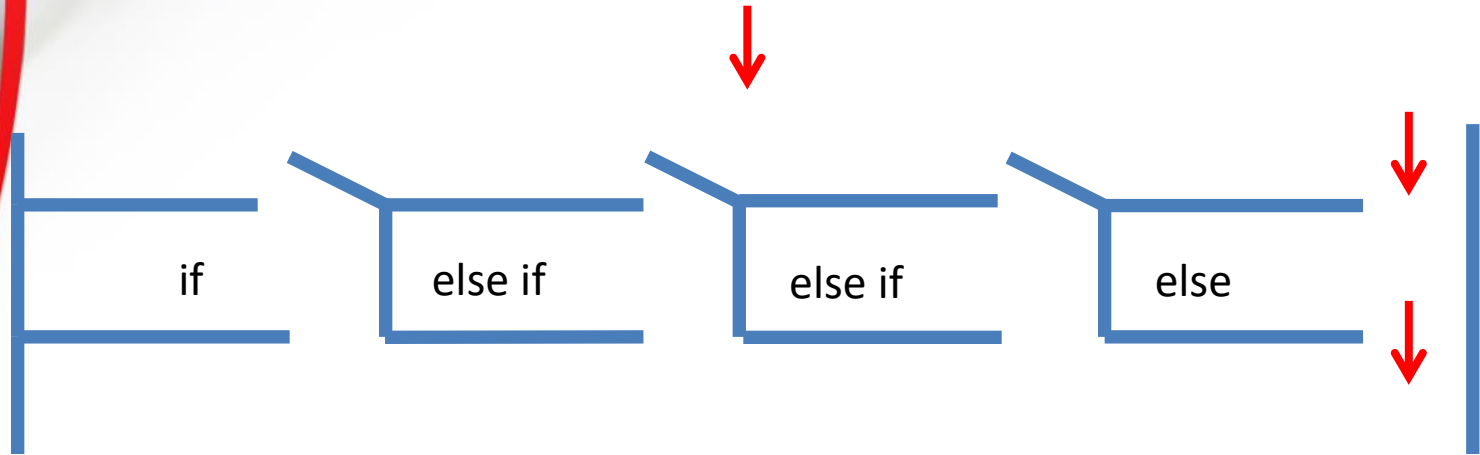
- Dari kode program di atas, maka tulisan “lulus” akan ditampilkan jika *user* memasukkan angka lebih besar atau sama dengan 65. Jika lebih kecil dari 65 maka tulisan “tidak lulus” yang akan ditampilkan ke layar

Pemilihan Lebih dari Dua Kondisi

Penulisan dalam bahasa C

```
if(kondisi/syarat){
    //proses
    .....
}else if(kondisi/syarat){
    //proses
    .....
} else if(kondisi/syarat){
    //proses
    .....
}else{
    //proses
    .....
}
```

Pemilihan Lebih dari Dua Kondisi (2)



Pemilihan Lebih dari Dua Kondisi (3)

```
#include <stdio.h>
int main() {
    printf("masukkan angka:\n");
    int nilai;
    scanf("%d", &nilai);

    if(nilai >= 80){
        printf("A");
    }else if((nilai < 80) && (nilai >= 70)){
        printf("B");

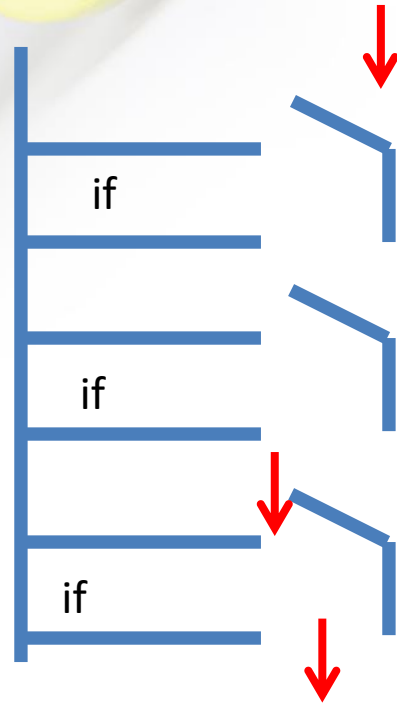
    }else if((nilai < 70) && (nilai >= 60)){
        printf("C");

    }else if((nilai < 60) && (nilai >= 50)){
        printf("D");

    }else{
        printf("E");
    }

    return 0;
}
```

Pemilihan Beruntun



```
if(kondisi/syarat){  
}
```

```
if(kondisi/syarat){  
}
```

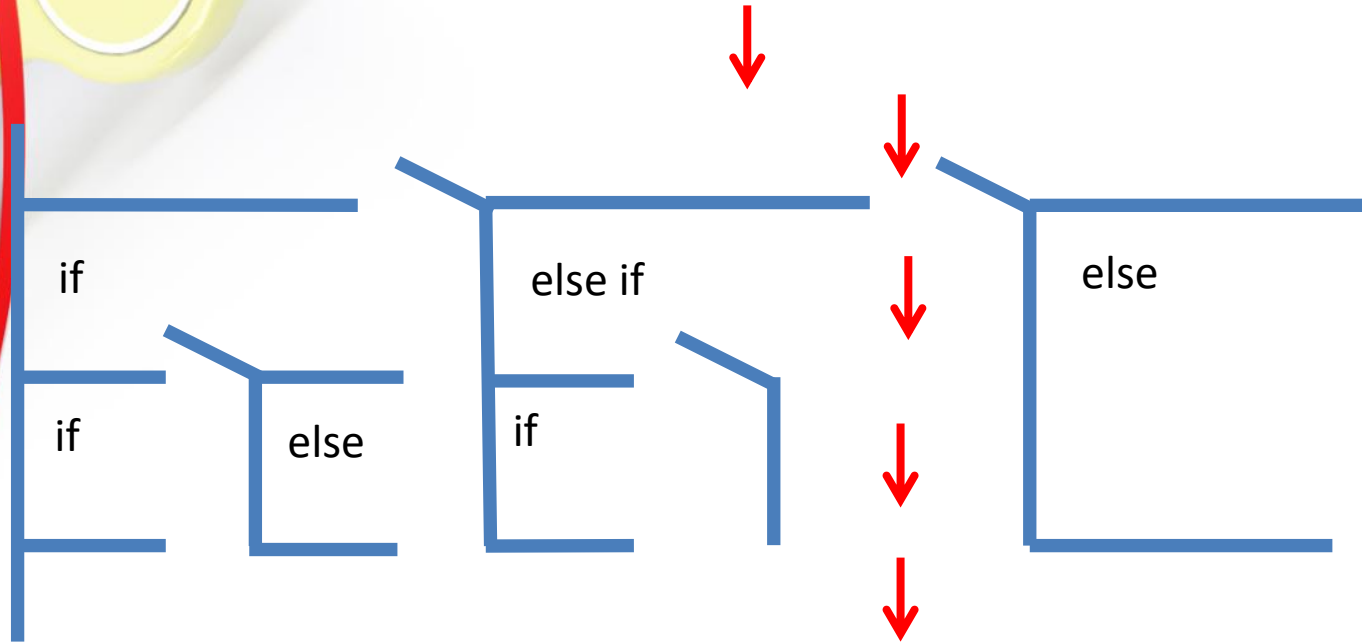
```
if(kondisi/syarat){  
}
```



Pemilihan Tersarang (*nested*)

```
if(kondisi/syarat){  
    if(kondisi/syarat){  
    }else{  
    }  
}else if(kondisi/syarat){  
    if(kondisi/syarat){  
    }  
}else{  
}
```

Pemilihan Tersarang (*nested*) (2)





Studi kasus (1)

- Diberikan 3 buah karakter, periksa apakah di dalam 3 karakter itu dengan urutan konsonan vokal konsonan (semua huruf kecil)
- Definisikan apa itu konsonan dan apa itu vokal agar dapat menjadikan syarat pemilihan

Algoritma Studi Kasus (1)

- Membuat kotak untuk menampung karakter pertama
- Membuat kotak untuk menampung karakter kedua
- Membuat kotak untuk menampung karakter ketiga
- Meminta masukan dari *user* untuk mengisi tiga kotak karakter yang disediakan
- Membuat tiga buah kotak integer sebagai penanda apakah karakter masukan *user* memenuhi syarat
- Semua kotak diisi dengan angka 0;
- Memeriksa 3 buah karakter:
 - Jika karakter pertama adalah konsonan maka nilai integer pertama diubah menjadi 1
 - Jika karakter kedua vokal maka nilai integer kedua diubah menjadi 1
 - Jika karakter ketiga adalah konsonan maka integer ketiga diubah menjadi 1
- Jika hasil integer adalah 1, 1, 1 maka masukan dinyatakan memenuhi syarat

Program Studi Kasus (1)

```
#include <stdio.h>
int main(){
    //tiga kotak karakter
    char ca, cb, cc;
    //meminta masukan user untuk 3 kotak
    karakter
    scanf("%c %c %c", &ca, &cb, &cc);

    //membuat tiga kotak integer sebagai penanda
    int a=0, b=0, c=0;

    //periksa konsonan
    if((ca != 'a') && (ca != 'i')
        && (ca != 'u') && (ca != 'e')
        && (ca != 'o')){
        a = 1;
    }

    //periksa vokal
    if((cb == 'a') || (cb == 'i') ||
        (cb == 'u') || (cb == 'e') ||
        (cb == 'o')){
        b = 1;
    }
}
```

```
//periksa konsonan
if((cc != 'a') && (cc != 'i') &&
    (cc != 'u') && (cc != 'e') &&
    (cc != 'o')){
    c = 1;
}

//memeriksa memenuhi syarat atau tidak
if((a == 1) && (b == 1) && (c == 1)){
    printf("memenuhi syarat\n");
}else{
    printf("tidak memenuhi syarat");
}

return 0;
}
```

Hasil Eksekusi (1)



y a h
memenuhi syarat

c i e
tidak memenuhi syarat



Studi Kasus (2)

- Diberikan 2 buah angka, periksa apakah kedua bilangan merupakan bilangan genap atau ganjil
- Definisikan apa yang disebut bilangan genap atau ganjil

Algoritma Studi Kasus (2)

- Membuat kotak untuk menampung angka pertama
- Membuat kotak untuk menampung angka kedua
- Meminta masukan dari *user* untuk mengisi dua kotak angka yang disediakan
- Memeriksa 2 buah angka apakah genap atau ganjil



Program Studi Kasus (2)

```
#include <stdio.h>
int main(){
    //membuat kotak angka
    int a, b;
    //meminta masukan dari user
    scanf("%d", &a);
    scanf("%d", &b);
    //memeriksa bilangan pertama untuk apakah
    genap atau ganjil
    if(a % 2 == 0){
        printf("bilangan pertama genap\n");
    }else{
        printf("bilangan pertama ganjil\n");
    }
}
```

```
//memeriksa bilangan kedua untuk apakah genap
atau ganjil

if(b % 2 != 0){
    printf("bilangan kedua ganjil\n");
}else{
    printf("bilangan kedua genap\n");
}

return 0;
}
```



Hasil Eksekusi (2)

```
3  
8  
bilangan pertama ganjil  
bilangan kedua genap
```



Studi Kasus (3)

- Diberikan 3 buah angka bilangan bulat, urutkan ketiga angka tersebut urut dari besar ke kecil (tidak ada angka yang bernilai sama)

Algoritma Studi Kasus (3)


- Membuat 3 kotak untuk menyimpan angka masukan
- Meminta masukan user untuk ketiga kotak angka
- Memeriksa kemungkinan yang mungkin terjadi
 - Jika bilangan pertama paling besar
 - Bandingkan bilangan kedua dan ketiga
 - jika bilangan kedua lebih besar dari bilangan ketiga
 - lainnya
 - Jika bilangan kedua paling besar
 - Bandingkan bilangan pertama dan ketiga
 - jika bilangan pertama lebih besar dari bilangan ketiga
 - lainnya
 - Jika bilangan ketiga paling besar
 - Bandingkan bilangan pertama dan kedua
 - jika bilangan pertama lebih besar dari bilangan kedua
 - lainnya

Program Studi Kasus (3)

```
#include <stdio.h>
int main(){
    //membuat kotak angka
    int a, b, c;
    //meminta masukan dari user
    scanf("%d", &a);
    scanf("%d", &b);
    scanf("%d", &c);
    //memeriksa setiap kemungkinan
    if((a > b) && (a > c)){
        //jika a paling besar
        if(b > c){
            printf("%d %d %d\n", a, b, c);
        }else{
            printf("%d %d %d\n", a, c, b);
        }
    }
}
```

```
else if((b > a) && (b > c)){
    //jika b paling besar
    if(a > c){
        printf("%d %d %d\n", b, a, c);
    }else{
        printf("%d %d %d\n", b, c, a);
    }
}
else if((c > a) && (c > b)){
    //jika c paling besar
    if(a > b){
        printf("%d %d %d\n", c, a, b);
    }else{
        printf("%d %d %d\n", c, b, a);
    }
}
return 0;
}
```

Hasil Eksekusi (3)



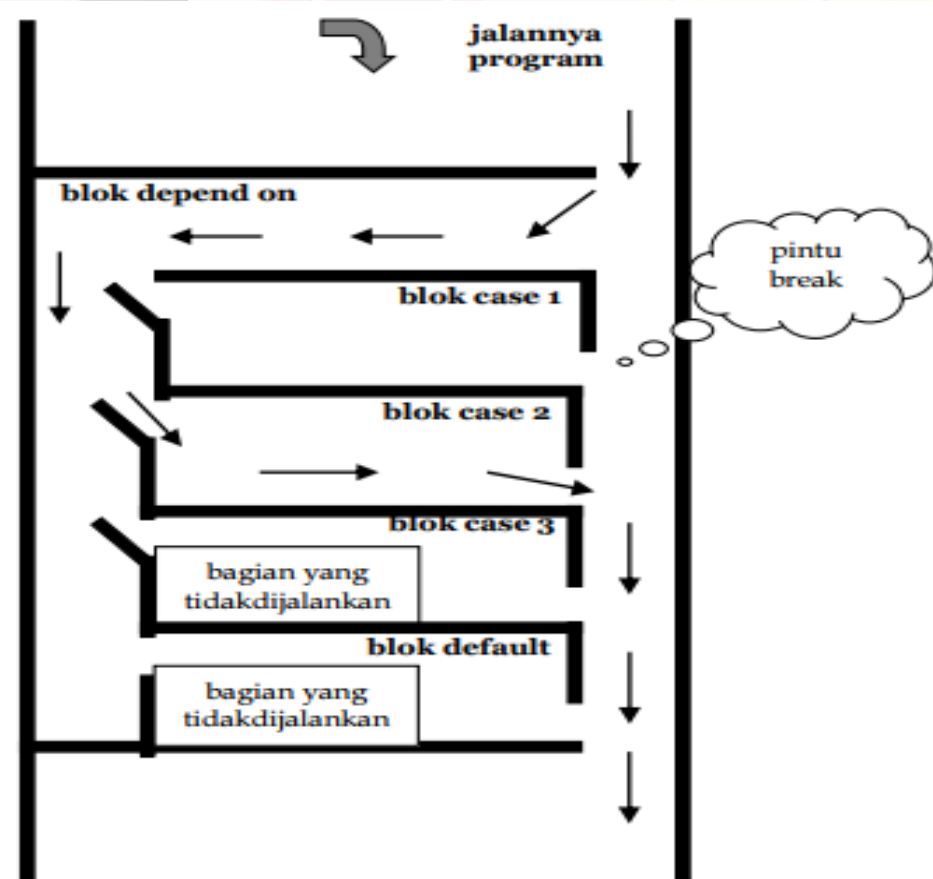
```
4  
66  
34  
66 34 4
```



Mari Mencoba

- Diberikan sebuah bilangan periksa apakah bilangan merupakan satuan, puluhan, ratusan, ribuan, atau selainnya
- Diberikan tiga buah bilangan bulat sebagai sisi segitiga, periksa apakah segitiga termasuk segitiga sama sisi, sama kaki, siku-siku, atau lainnya
- Diberikan dua buah bilangan, jika keduanya ganjil maka tampilkan hasil kali keduanya, jika keduanya genap maka tampilkan hasil pertambahan keduanya, dan jika selainnya maka tampilkan kedua bilangan.
- Diberikan sebuah bilangan float, periksa apakah nilai di depan koma adalah kelipatan dari nilai di belakang koma (gunakan mod untuk operasi kelipatan), asumsikan bahwa diambil dua angka untuk nilai depan dan belakang koma

Pemilihan Menggunakan Switch



```
switch(nama_variabel){  
    case nilai_variabel_1  
:  
    aksi_1  
    break;  
    case nilai_variabel_2  
:  
    aksi_2  
    break;  
    .....  
    case nilai_variabel_n  
:  
    aksi_n  
    break;  
    default      :  
    aksi_default  
    break;  
}
```

Pemilihan Menggunakan Switch (2)

```
switch(hari){  
    case 1 : {  
        printf("hari senin");  
        break;  
    }  
    case 2 : {  
        printf("hari selasa");  
        break;  
    }  
    case 3 : {  
        printf("hari rabu");  
        break;  
    }  
    .....  
    case 7 : {  
        printf("hari minggu");  
        break;  
    }  
    default : {  
        printf("tidak ada hari ke : %d\n", hari);  
        break;  
    }  
}
```

```
#include <stdio.h>
```

```
int main(){
```

```
    printf("Masukkan dua buah  
angka\n");
```

```
    int a, b;  
    scanf("%d %d", &a, &b);
```

```
    printf("Pilihlah Menu  
Berikut:\n");
```

```
    printf("1. tambah\n");  
    printf("2. kurang\n");  
    printf("3. kali\n");
```

```
    int menu;  
    scanf("%d", &menu);
```

```
    switch(menu){
```

```
        case 1 : {
```

```
            printf("hasil tambah: %d\n", (a + b));  
            break;
```

```
        }
```

```
        case 2 : {
```

```
            printf("hasil kurang: %d\n", (a - b));  
            break;
```

```
        }
```

```
        case 3 : {
```

```
            printf("hasil kali: %d\n", (a * b));  
            break;
```

```
        }
```

```
        default : {
```

```
            printf("Menu tidak valid\n");  
            break;
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

Daftar Pustaka

MD 100032

Rosa A. S.

LOGIKA ALGORITMA dan PEMROGRAMAN DASAR

- Pendahuluan Algoritma dan Pemrograman
- Komentar
- Tipe Data
- Masukan (input) dan Keluaran (Output)
- Operator
- Array (Larik)
- Percabangan / Pemilihan If
- Perulangan (Looping)
- Prosedur
- Fungsi
- Matriks dalam Larik (Array) Dua Dimensi
- Rekursif
- Pengurutan (Sorting)
- Penggabungan Larik
- Pencarian (Searching)
- Arsip Berurutan (Sequential File)
- Mesin Abstrak
- Flowchart

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 runtut agar komputer mengerti dan mampu mengeksekusinya. Analisis kasus sangat dibutuhkan dalam membuat sebuah algoritma, misalnya proses apa saja 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 pola-pola 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.

Komputer

ISBN 978-602-471912-6



Harga Rp. 180.000,-



Pemasaran: BI-OBSES
Pasar buku Palasan 82 Bandung 40264
Tel (022) 7317812 Fax: (022) 7317896
www.biobses.com



LOGIKA ALGORITMA dan PEMROGRAMAN DASAR

Rosa A. S.

LOGIKA ALGORITMA dan PEMROGRAMAN DASAR

Rosa A. S.

- Pendahuluan Algoritma dan Pemrograman
- Komentar
- Tipe Data
- Masukan (Input) dan Keluaran (Output)
- Operator
- Array (Larik)
- Percabangan / Pemilihan If
- Perulangan (Looping)
- Prosedur
- Fungsi
- Matriks dalam Larik (Array) Dua Dimensi
- Rekursif
- Pengurutan (Sorting)
- Penggabungan Larik
- Pencarian (Searching)
- Arsip Berurutan (Sequential File)
- Mesin Abstrak
- Flowchart

analisis

program

Activate Word
Click Here