

# BAB 12

---

## FUNCTION ALGORITMA PEMROGRAMAN

# BAB 12

# FUNCTION ALGORITMA

# PEMROGRAMAN

## Capaian Pembelajaran

Setelah mempelajari materi dalam bab ini, Mahasiswa diharapkan mampu menganalisis, menyampaikan pendapat, dan mengoperasikan fungsi algoritma pemrograman pada sebuah kasus operasional bisnis

## Pokok Bahasan

1. Konsep Dasar Fungsi
2. Cara Membuat dan Memanggil Fungsi
3. Parameter Fungsi
4. Pemanggilan Fungsi pada Fungsi Lainnya
5. Implementasi Fungsi dalam Algoritma Pemrograman

## Evaluasi Pembelajaran

Soal Latihan Fungsi Algoritma Pemrograman

---

Pre Test  
Fungsi Algoritma Pemrograman

1. Apa yang anda Ketahui Fungsi?
2. Sebutkan contoh penggunaan pada sebuah fungsi!
3. Apa yang anda ketahui dengan Parameter dalam Fungsi?
4. Sebutkan apa saja jenis dari parameter!
5. Apa yang anda ketahui bagaimana parameter fungsi diproses?
6. Jelaskan implementasi dari penggunaan sebuah Fungsi

Fungsi (*Function*) pada dasarnya memiliki kesamaan penggunaan dengan prosedur. Fungsi merupakan salah satu sub program yang melakukan satu atau lebih proses yang mengembalikan sebuah nilai untuk dikirimkan ke program utama (*return*). Fungsi juga berperan untuk melakukan pemilahan kode program agar penulisan kode tidak ditulis menjadi satu pada program utama (dipilah menjadi berdasarkan modul). Bab ini berisi penjelasan yang dituangkan ke dalam beberapa sub bab, diantaranya: Definisi Fungsi, parameter dalam fungsi, pemanggilan fungsi, fungsi standard, pengenalan method dalam bahasa java, implementasi fungsi dalam algoritma pemrograman.

## 12.1. Konsep Dasar Fungsi

Fungsi adalah sebuah blok program terpisah yang masih menjadi bagian dari program lain yang lebih besar. Fungsi pada dasarnya sama dengan prosedur. Jika prosedur didalamnya merupakan proses yang dijalankan, namun pada fungsi didalamnya serangkaian proses yang berjalan dan menghasilkan keluaran. Fungsi dapat dipanggil oleh program utamanya ataupun oleh prosedur atau fungsi lain yang juga merupakan bagian dari program utamanya. Pada sebuah program dapat terdiri dari satu atau lebih fungsi dan satu program utamanya.

Beberapa keuntungan yang didapatkan dari penggunaan fungsi hampir sama dengan keuntungan saat menggunakan prosedur. Penggunaan fungsi dapat diterapkan pada program utama yang lain, dengan menggunakan fungsi mudah untuk penelusuran terhadap kesalahan kode program, kemudahan dalam memodifikasi program, dan penggunaan fungsi bisa menyederhanakan kode program. Penyederhanaan ini berupa tidak perlunya melakukan kode yang berulang, namun cukup dengan pembuatan satu fungsi saja. Perbedaan mendasar dari perbedaan dari penggunaan prosedur dan fungsi adalah sebuah nilai yang dikembalikan oleh fungsi.

Salah satu kelemahan dari penggunaan fungsi adalah tidak bisa mencetak nilai balik di dalam fungsi itu sendiri, sehingga harus menggunakan variabel yang terletak di prosedur atau program utama. Fungsi tidak diijinkan memasukkan nilai

dan mencetak hasil keluaran dalam sebuah fungsi. Misalnya saja untuk melakukan cetak ke layar pada hasil penjumlahan dari proses aritmatika. Fungsi hanya bertugas sebagai penampungan nilai sementara. Jika pengguna ingin mencetak sebuah nilai, maka nilai yang dikembalikan dari hasil proses fungsi ditampung pada sebuah variabel di program utama, lalu pada program utama lah proses mencetak dilakukan.

Fungsi dapat dipanggil oleh program utamanya, prosedur maupun fungsi lainnya. Sebuah program dapat terdiri dari satu atau lebih fungsi, satu atau lebih prosedur dan satu program utamanya. Salah satu keunggulan fungsi adalah memiliki nilai balik (return value). Nilai balik yang dihasilkan dari sebuah fungsi harus ditampung terlebih dahulu pada sebuah variabel yang terletak pada program utama, kemudian nilai yang terdapat di dalam variabel tersebut dapat diolah atau diproses pada program selanjutnya.

## 12.2. Cara membuat dan Memanggil Fungsi

Pada contoh penggunaan fungsi pada bahasa java, fungsi harus ditulis atau dibuat didalam sebuah *class*. Sebagai contoh dapat dilihat sebagai berikut:

```
public static TypeDatayangDikembalikan namaFungsi(){  
    // statement atau kode fungsi  
}
```

Pada contoh diatas dapat dijelaskan bahwa *static* menunjukkan bahwa fungsi dapat dipanggil tanpa harus membuat instansiasi objek terlebih dahulu. *TypeDatayangDikembalikan* menunjuk bahwa tipe data yang ditentukan pada saat nilai dari fungsi dikembalikan. Sementara itu untuk *namaFungsi()* merupakan nama fungsi yang dapat diawali dengan huruf kecil tidak boleh ada pemisah dan dapat terdiri dari lebih dari satu kata.

Setelah berhasil membuat fungsi maka selanjutnya dapat dilakukan pemanggilan sebuah fungsi. Fungsi dapat dipanggil pada program utama (main) atau dapat juga dipanggil pada fungsi yang lainnya. Untuk melakukan pemanggilan cukup dengan dituliskan nama fungsinya saja *namaFungsi()*.

### 12.3. Parameter Fungsi

Seperti halnya dengan prosedur, fungsi juga memiliki parameter yang digunakan untuk melakukan pertukaran data. Penggunaan parameter fungsi juga sama dengan parameter prosedur yang telah dijelaskan pada bab prosedur, dimana memiliki dua jenis parameter, yaitu parameter aktual dan parameter formal. Begitu juga dengan bagaimana parameter diproses, yaitu dengan cara *pass by value* dan *pass by reference*.

Parameter merupakan sebuah variabel yang menampung nilai sementara untuk diproses di dalam sebuah fungsi. Seperti yang sudah dijelaskan dalam bab prosedur, parameter dapat sebagai nilai masukan untuk diproses dalam fungsi. Parameter sendiri ditulis diantara tanda kurung, dan parameter sendiri harus diberikan tipe data yang digunakan. Apabila parameter yang digunakan pada sebuah fungsi terdiri dari satu, maka masing masing parameter dipisahkan dengan tanda koma. Mengingat pada sebuah fungsi harus dapat mengembalikan sebuah nilai, maka di dalam fungsi harus ada kata kunci "return" agar nilai yang dihasilkan dari fungsi dapat dikirimkan ke program utama untuk diproses lagi. Berikut ini merupakan *pseudocode*, *flowchart*, dan pengkodean bahasa java untuk contoh penerapan pada sebuah fungsi dengan penggunaan parameter.

```
Algoritma: Menghitung luas persegi panjang dengan inputan oleh user}
```

```
FUNGSI hitungLuasPP(input p, l: integer)
```

```
Deklarasi
```

```
luas: integer
```

```
Deskripsi
```

```
luas ← p*l
```

```
return(luas)
```

```
ALGORITMA UTAMA
```

```
Deklarasi
```

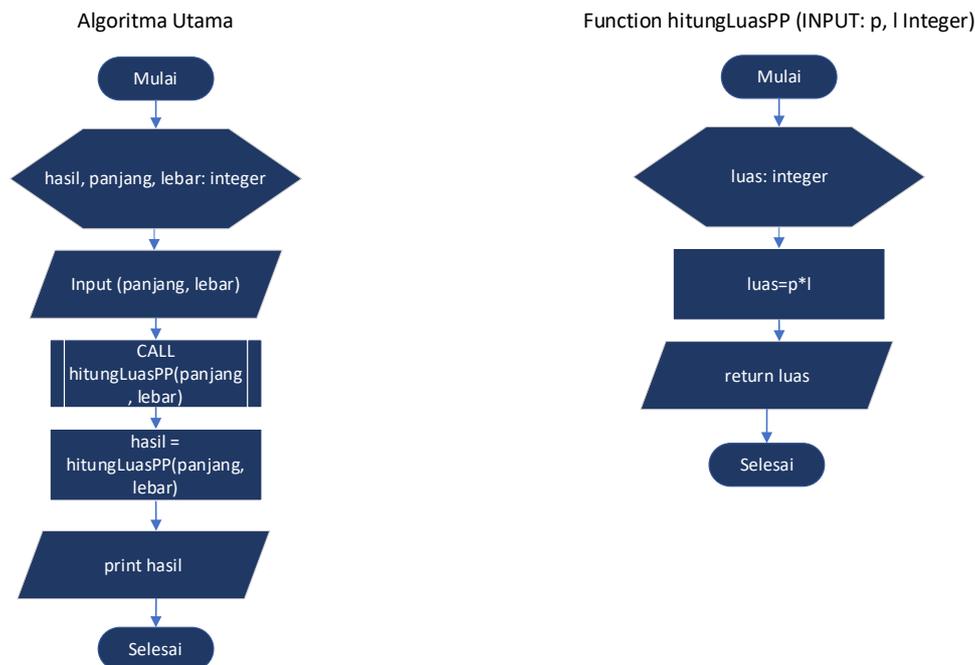
```
hasil, panjang, lebar: integer
```

```
Deskripsi
```

```
input (panjang)
```

```
input (lebar)
hasil ← Fungsi hitungLuasPP(panjang, lebar)
print (hasil)
END
```

Berikut ini merupakan hasil penggambaran flowchart pada contoh penerapan fungsi dengan penggunaan parameter.



Berikut ini merupakan hasil penerapan pengkodean java berdasarkan pseudocode dan flowchart pada penggunaan parameter.

```
import java.util.Scanner;
class ContohFungsiJava{

static int hitungLuasPP(int p , int l){
    int luas;
    luas = p*l;
    return luas;
}

public static void main(String args[]){
    int hasil, panjang, lebar;
    Scanner inputan = new Scanner(System.in);
    System.out.println("Perhitungan nilai Luas Persegi Panjang" );
    System.out.println("masukkan nilai panjang: " );
    panjang = inputan.nextInt();
    System.out.println("masukkan nilai lebar: " );
```

```
lebar = inputan.nextInt();

hasil = hitungLuasPP(panjang, lebar);
System.out.println("Luas Persegi Panjang = " + hasil);
}
}
```

## 12.4. Pemanggilan Fungsi pada Fungsi Lainnya

Pemanggilan sebuah fungsi dapat juga dilakukan dengan melakukan pemanggilan sebuah fungsi lainnya. Sebagai contohnya adalah fungsi perhitungan sebuah kubus, maka memerlukan fungsi lain yaitu fungsi perhitungan sebuah luas persegi. Berikut ini merupakan contoh pseudocode dan penerapan dalam program java untuk menghitung kubus.

Algoritma: Menghitung luas Kubus dengan inputan nilai sisi oleh user}

**FUNGSI hitungLuasPersegi**(input s: integer)

**Deklarasi**

luas: integer

**Deskripsi**

luas  $\leftarrow$  s\*s

return(luas)

**FUNGSI hitungLuasKubus**(input ss: integer)

**Deklarasi**

luasKubus: integer

**Deskripsi**

luasKubus  $\leftarrow$  hitungLuasPersegi(ss)

return(luasKubus)

**ALGORITMA UTAMA**

**Deklarasi**

hasil, sisi: integer

**Deskripsi**

input(sisi)

hasil  $\leftarrow$  Fungsi hitungLuasKubus(sisi)

print (hasil)

END

Berikut ini merupakan hasil penerapan pengkodean java berdasarkan pseudocode perhitungan luas kubus diatas.

```
import java.util.Scanner;
class ContohFungsiJava{
    static int hitungLuasPersegi(int s){
        int luas;
        luas = s*s;
        return luas;
    }
    static int hitungLuasKubus(int ss){
        int luasKubus;
        luasKubus = 6 * hitungLuasPersegi(ss);
        return luasKubus;
    }

    public static void main(String args[]){
        int hasil, sisi;
        Scanner inputan = new Scanner(System.in);
        System.out.println("Perhitungan nilai volume kubus" );
        System.out.println("masukkan nilai sisi: " );
        sisi = inputan.nextInt();

        hasil = hitungLuasKubus(sisi);
        System.out.println("Luas Kubus = " + hasil);
    }
}
```

## 12.5. Contoh Implementasi Fungsi pada Algoritma Pemrograman

Penerapan fungsi dapat digunakan untuk menyelesaikan beberapa aktifitas dalam kehidupan sehari hari. Beberapa contohnya adalah sebagai berikut:

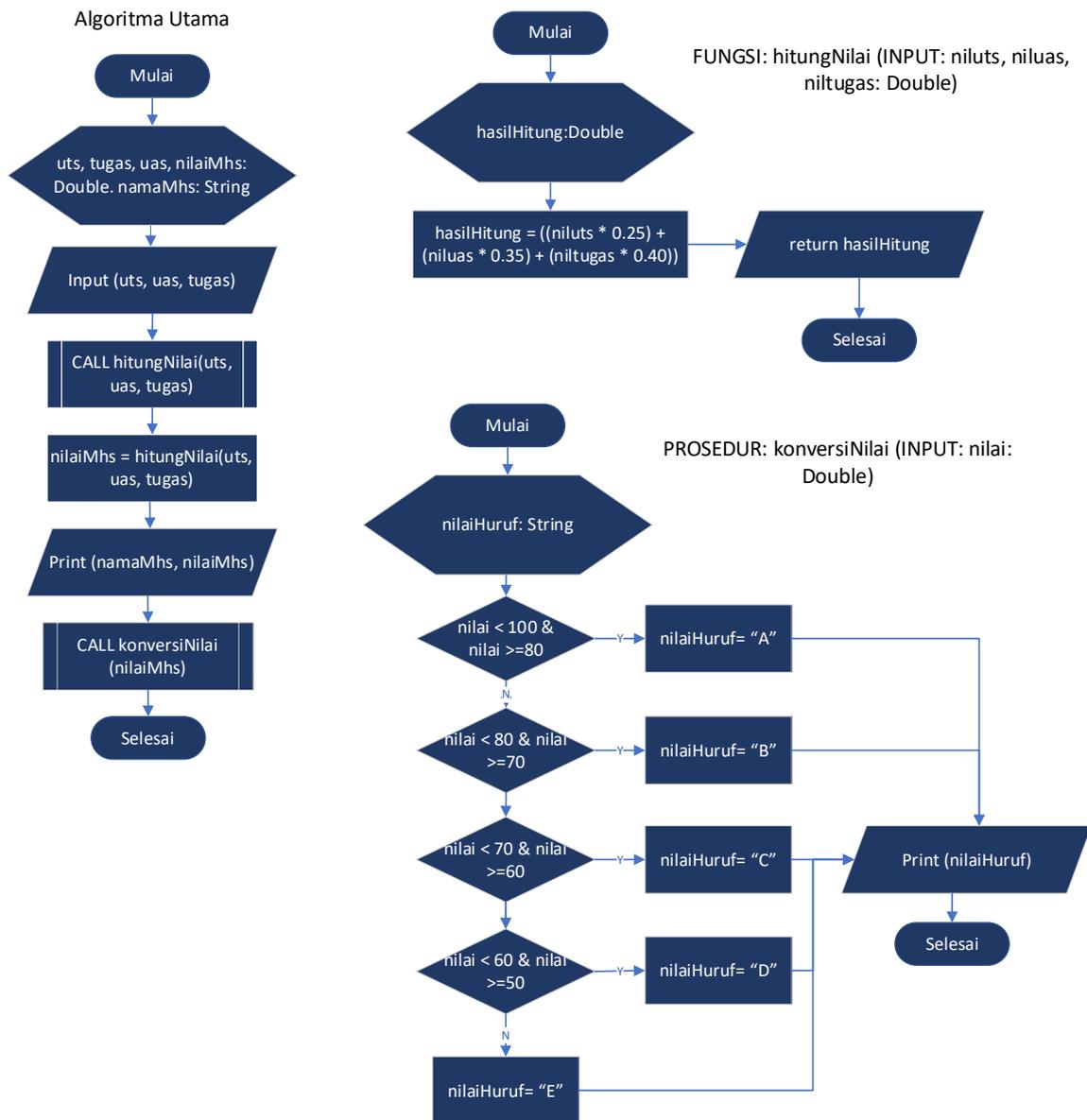
### Contoh Implementasi 1: Perhitungan Nilai Ujian

Dosen ingin melakukan perhitungan nilai hasil akhir pada matakuliah algoritma pemrograman kepada mahasiswanya. Nilai hasil akhir yang ingin ditampilkan adalah nilai dalam bentuk huruf. Ketentuan nilai huruf yang dimaksud diantaranya: Nilai A (80 - 100), Nilai B (70 - 79), C (60 - 69), D (50 - 59), dan E (0 - 50).

Nilai yang diperoleh mahasiswa berasal dari nilai inputan dosen yang terdiri atas nilai UTS (25%), UAS (35%), dan Tugas (40%).

Berdasarkan kasus diatas, maka dapat ditunjukkan hasil penerapan algoritma berupa *flowchart*, *pseudocode*, dan bahasa pemrograman java dengan menggunakan konsep fungsi.

a. Flowchart contoh kasus Perhitungan dan Konversi Nilai Akhir Mahasiswa



b. Pseudocode contoh kasus Perhitungan dan Konversi Nilai Akhir Mahasiswa

```

FUNGSI hitungNilai(niluts, niluas, niltugas: Double)
Deklarasi
hasilHitung: Double
Deskripsi
hasilHitung ← ((niluts * 0.25) + (niluas * 0.35) + (niltugas * 0.40))
return hasilHitung
END
    
```

```
PROSEDUR konversiNilai(input nilai: double)
```

**Deklarasi**

```
nilaiHuruf: String
```

**Deskripsi**

```
If nilai<100 and nilai>=80 then nilaiHuruf ← "A"  
Else if nilai<80 and nilai>=70 then nilaiHuruf ← "B"  
Else if nilai<70 and nilai>=60 then nilaiHuruf ← "C"  
Else if nilai<60 and nilai>=50 then nilaiHuruf ← "D"  
    Else then nilaiHuruf ← "E"  
Print (nilaiHuruf)  
end
```

**ALGORITMA UTAMA**

**Deklarasi**

```
uts, tugas, uas, nilaiMhs: Double. namaMhs: String
```

**Deskripsi**

```
FUNGSI hitungNilai(uts, uas, tugas)  
nilaiMhs ← hitungNilai(uts, uas, tugas)  
Print (namaMhs, nilaiMhs)  
PROSEDURE konversiNilai(nilaihs  
END
```

c. Bahasa pemrograman java contoh kasus Perhitungan dan Konversi Nilai Akhir Mahasiswa

```
import java.util.Scanner;  
class ContohFungsiJava{  
    double niluts, niluas, niltugas;  
    public static double hitungNilai(Double niluts, Double niluas, Double  
niltugas){  
        double hasilhitung;  
        hasilhitung = ((niluts * 0.25) + (niluas * 0.35) + (niltugas *  
0.40));  
        return hasilhitung;  
    }  
    static void konversiNilai(double nilai){  
        String nilaiHuruf;  
        if (nilai < 100 & nilai >=80){  
            nilaiHuruf = "A";  
        } else if (nilai < 80 & nilai >=70){  
            nilaiHuruf = "B";  
        } else if (nilai < 70 & nilai >=60){  
            nilaiHuruf = "C";  
        } else if (nilai < 60 & nilai >=50){
```

```
        nilaiHuruf = "D";
    } else {
        nilaiHuruf = "E";
    }
    System.out.println("Konversi Nilai Akhir Mahasiswa: " +
nilaiHuruf);
    }
    public static void main (String args[]){
        double uts, tugas, uas, nilaiMhs;
        String namaMhs;
        Scanner inputan = new Scanner(System.in);
        System.out.println("Masukkan Nama Mahasiswa: " );
        namaMhs = inputan.nextLine();
        System.out.println("Masukkan Nilai UTS: " );
        uts = inputan.nextDouble();
        System.out.println("Masukkan Nilai UAS: " );
        uas = inputan.nextDouble();
        System.out.println("Masukkan Nilai TUGAS: " );
        tugas = inputan.nextDouble();
        nilaiMhs = hitungNilai(uts, uas, tugas);
        System.out.println("");
        System.out.println("");
        System.out.println("=====Hasil Perhitungan Nilai
Mahasiswa=====");
        System.out.println("Nama Mahasiswa: " + namaMhs);
        System.out.println("Nilai Akhir Mahasiswa: " + nilaiMhs);
        konversiNilai(nilaiMhs);
    }
}
```

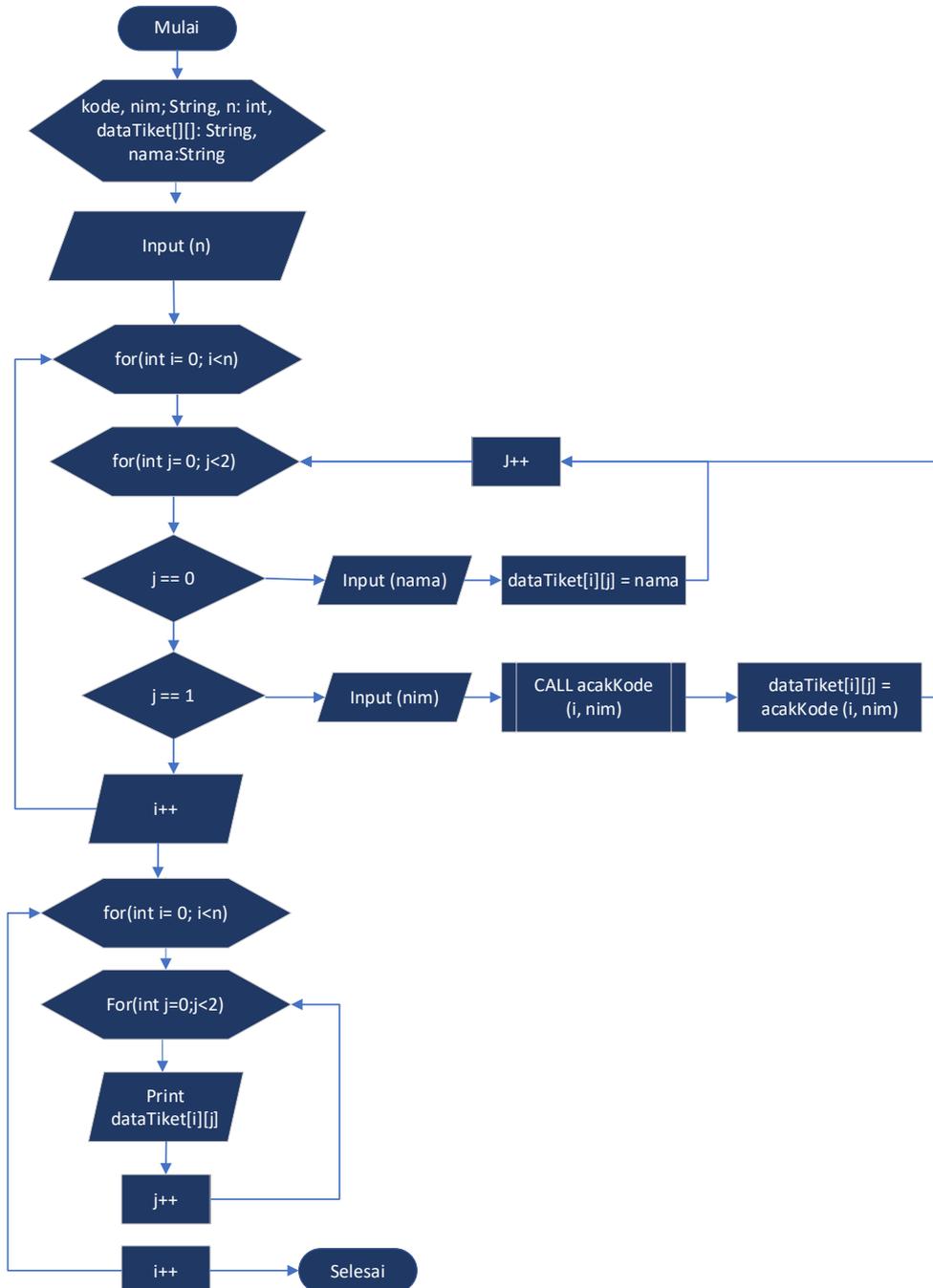
## Contoh Implementasi 2: Membuat Kode Tiket Seminar Internasional

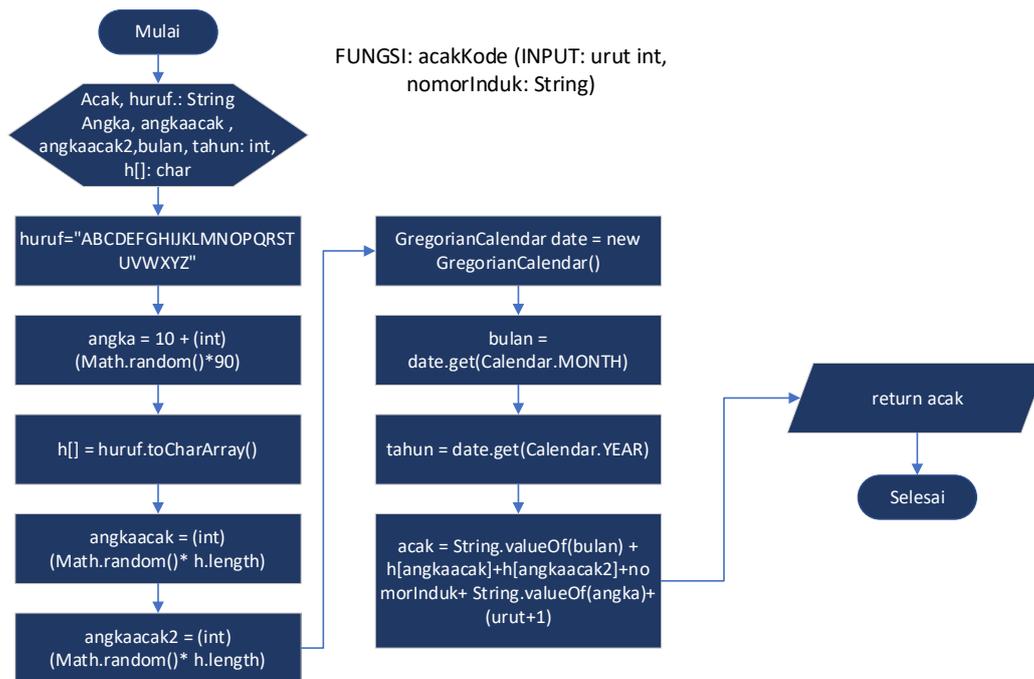
Panitia dari organisasi mahasiswa akan menyelenggarakan kegiatan seminar internasional bagi mahasiswa internal kampus. Bagi mahasiswa yang ingin mendaftarkan diri menjadi peserta harus mendaftarkan diri ke panitia. Pembelian tiket dapat dilakukan langsung untuk beberapa mahasiswa. Panitia akan memberikan kode tiket unik bagi mahasiswa yang sudah mendaftar. Kode tiket unik ini merupakan kombinasi dari:

1. Bulan pada saat mahasiswa mendaftar
2. Dua Huruf acak A - Z
3. Nomor induk Mahasiswa
4. Dua Digit Angka Acak
5. Nomor urut dari jumlah tiket yang dibeli oleh mahasiswa.

Berdasarkan studi kasus diatas, maka berikut ini merupakan hasil penerapan fungsi dari sebuah algoritma berupa *flowchart*, *pseudocode*, dan bahasa pemrograman java.

a. *Flowchart* contoh kasus Pembuatan Kode Tiket Seminar





## b. Pseudocode contoh kasus Pembuatan Kode Tiket Seminar

**FUNGSI** `acakKode` (urut int, nomorInduk: String)

**Deklarasi**

acak, huruf: String

angka, angkaacak, angkaacak2, bulan, tahun: integer

h[]: char

**Deskripsi**

huruf ← "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

angka ← 10 + (int) (Math.random()\*90)

h[] ← huruf.toCharArray();

angkaacak ← (int) (Math.random()\* h.length)

angkaacak2 ← (int) (Math.random()\* h.length)

GregorianCalendar date ← new GregorianCalendar()

bulan ← date.get(Calendar.MONTH)

tahun ← date.get(Calendar.YEAR)

acak ← String.valueOf(bulan) + h[angkaacak] + h[angkaacak2] + nomorInduk +  
String.valueOf(angka) + (urut+1)

return acak

END

#### ALGORITMA UTAMA

##### Deklarasi

```
kode, nim: String
dataTiket[][]: String
n: integer
```

##### Deskripsi

```
Input n
For (i=0;i<n;i++)
    For (j=0;j<2;j++)
        If j==0, then Input(nama)
            dataTiket[i][j] ← nama
        If j==1, then input (nim)
            dataTiket[i][j] ← acakKode(i, nim)
    END For
End for
For (i=0;i<n;i++)
    For (j=0;j<2;j++)
        Print (dataTiket[i][j])
    END For
End For
END
```

#### c. Bahasa pemrograman java contoh kasus Pembuatan Kode Tiket Seminar

```
import java.util.Scanner;
import java.util.*;
public class ContohFungsiJava {

    static String acakKode(int urut, String nomorInduk){
        String acak;
        String huruf = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        int angka = 10 + (int) (Math.random()*90);
        char h[] = huruf.toCharArray();
        int angkaacak = (int) (Math.random()* h.length);
        int angkaacak2 = (int) (Math.random()* h.length);

        GregorianCalendar date = new GregorianCalendar();
        int bulan = date.get(Calendar.MONTH);
        int tahun = date.get(Calendar.YEAR);

        acak = h[angkaacak]+h[angkaacak2]+nomorInduk+ String.valueOf(angka)+ (urut+1) ;

        return acak;
    }
}
```

```
public static void main(String[] args)
{
    Scanner masuk=new Scanner (System.in);

    String kode;
    String nim;

    System.out.print("Jumlah Mahasiswa : ");

    int n=masuk.nextInt();

    String dataTiket[][] = new String[n][2];

    for(int i= 0; i<n; i++)
    {
        System.out.println("");
        System.out.println("Data Pembeli Tiket:  "+(i+1));

        for(int j=0;j<2;j++)
        {
            if (j == 0){
                System.out.print("Nama      :");
                dataTiket[i][j] = masuk.next();
            }

            else if (j == 1){
                System.out.print("NIM      :");
                nim= masuk.next();
                kode = acakKode(i, nim);

                dataTiket[i][j] = kode;
            }
        }
    }

    System.out.println("Data Tiket yang Dibeli");
    System.out.println("-----");
    System.out.println("Nama \t Kode Tiket \t");

    for(int i=0;i<n;i++)
    {
        for(int j=0;j<2;j++)
        {
            System.out.print(dataTiket[i][j]+"\\t");
        }
        System.out.println();
    }
}
```

## POST TEST

### Soal Latihan tentang Materi Fungsi

1. Sebuah Maskapai penerbangan ABC sedang membangun sistem pembelian tiket untuk beberapa rute perjalanan dari surabaya. Rute Perjalanan yang tersedia pada bulan ini adalah Surabaya - lombok, surabaya - jakarta, dan surabaya bali. Harga Tiket pada masing masing adaalah sebagai berikut:

No	Rute	Dewasa	> 15 thn	Pilihan Jadwal
1	Surabaya - Lombok	150.000	75.000	11.00 WIB   13.00 WIB
2	Surabaya - Jakarta	300.000	150.000	14.00 WIB   16.00 WIB
3	Surabaya - Bali	200.000	100.000	12.00 WIB   18.00 WIB

*Nb: Jadwal dan harga juga berlaku untuk kepulangan*

Bagi penumpang yang pembelian lebih dari 1.000.000 akan mendapatkan Diskon Promo sebesar 10%. Khusus pada bulan ini jika terdapat penumpang yang ingin melakukan pemesanan penerbangan ke bali akan mendapatkan potongan 5% (tidak ada minimal pembelian). Harga Pada tabel diatas pulang dan pergi adalah sama. Promo potongan 5% pada maskapai ini juga berlaku jika pelanggan melakukan pemesanan diatas pukul 15.00 WIB.

Sebuah keluarga yang terdiri dari Ayah (45 Tahun), Ibu (40 Tahun), Anak 1 (17 Tahun), dan 1 Anak (10 Tahun) ingin melakukan perjalanan dari surabaya ke Jakarta pada bulan ini dan pada bulan depan ke Bali. Keluarga ini akan melakukan pemesanan pada bulan ini untuk perjalanan ke Jakarta maupun ke Bali. Keluarga ini melakukan pembelian untuk keberangkatan dan kepulangan menggunakan maskapai penerbangan yang sama, yaitu maskapai ABC. Jadwal keberangkatan yang dipilih oleh keluarga ini untuk Jakarta ( berangkat 16.00, pulang: 14.00), untuk bali (berangkat 12.00, pulang: 18.00).

Berdasarkan studi kasus diatas, maka tentukan lah penulisan algoritma berupa flowchart, pseudocode, dan panerapan implementasi pada java (gunakan konsep Fungsi).

- 
2. Sebuah Perguruan Tinggi membuka pendaftaran mahasiswa baru. Bagi calon mahasiswa yang sudah membayar SPP bulan pertama, maka sistem akan mengkonversi Nim Mahasiswa. Nim yang dibuat merupakan gabungan dari:
- Tahun Angkatan
  - Kode Fakultas
  - Kode Prodi
  - Nomor Urut Mahasiswa

Sebagai Contoh NIM yang dihasilkan 20220201001, 20220201002, dst  
Berdasarkan studi kasus diatas, maka tentukanlah penulisan algoritma berupa flowchart, pseudocode, dan penerapan implementasi pada java (gunakan konsep Fungsi)