

Mgg 9

Praktikum Konsep Pemrograman Pembahasan Soal Post-Test UTS

Renovita Edelani, S.ST., M.Tr.Kom.



Renovita Edelani, S.ST., M.Tr.Kom. (Enok/Reno)

Tempat/Tanggal Lahir: Padang, 14 Oktober 1995

Email: renovita@pens.ac.id

No. HP: +6282384203252

LinkedIn: [renovita](#)

Google Scholar: [Link](#)

Formal Journey:

- D4 Teknik Informatika PENS (2013-2017)
- S2 Terapan Teknik Informatika dan Komputer (2017-2020)

Certification:

- Artificial Intelligence Engineer Associate (Solusi247)

Main Major:

- AI & ML Engineer
- Earthquake Data Analysis
- Text Analysis



Today's Material

01

Pengurutan
bilangan ascending

04

Penjumlahan
Bilangan Genap dan
Ganjil dari 1 sampai n

02

Pengecekan Zodiak

05

Validator Tanggal

03

Konversi bilangan
biner ke desimal

06

Bilangan sempurna
1 sampai n



00

Algoritma dan Computational Thinking

Review Mata Kuliah Logika dan Algoritma



Let's Watch the Video !



Exact Instructions Challenge

Yang dibutuhkan di Algoritma

- **Input : data** yang harus diberikan pada komputer
- **Output: informasi** yang akan **diperoleh** dari komputer
- **Logika: Merencanakan proses program.** Terdiri dari sejumlah instruksi yang mengubah input menjadi output yang diinginkan.

Test your CT

Ada dua stand (kios) penjual es krim warna warni, dengan 4 warna es krim



Es krim pada stand pertama dibuat dengan mengikuti instruksi sebagai berikut:

- 0) Mulai dengan corong kosong.
- 1) Ambil warna secara sembarang (random), tambahkan 2 bulatan berwarna sama.
- 2) Tambah 1 bulatan dengan warna berbeda.
- 3) Jika tingginya sudah sesuai yang diminta, berhenti. Jika belum, kembali langkah 1.

Es krim pada stand kedua tidak mengikuti instruksi tersebut.

Tantangan:

Yang mana merupakan es krim stand kedua?



A



B



C



D

Computational Thinking

- **Pemikiran Komputasional (CT):** Adalah pendekatan untuk menyelesaikan masalah **dengan cara yang sistematis** dan **dapat dikomputerisasi**.
- “**Cara berpikir** untuk **menguraikan suatu masalah** menjadi beberapa **bagian yang lebih kecil dan sederhana**, menemukan **pola** dalam dan masalah tersebut, serta **menyusun langkah-langkah logis** sebagai solusi mengatasi masalah.”

Penggunaan CT

Contoh:

- Kamu punya Cermin Persegi Panjang
- Coba hitung luasnya



Penggunaan CT

Contoh:

- Kamu punya Cermin Persegi Panjang
 - Coba hitung luasnya
-
- **Tentukan atribut** yang dibutuhkan untuk menghitung luas, yaitu panjang dan lebar.
 - Karena **luas persegi panjang** dihitung menggunakan **panjang kali lebar**, kita tidak perlu mengukur kelilingnya.
 - Kita tidak dapat menghitung luas tanpa mengetahui nilai lebar dan panjangnya.
 - Oleh karena itu, kita **perlu mengukur panjang dan lebarnya** sebelum menghitung luasnya. Kemudian, **kita dapat menghitung luasnya (Algoritma)**.





01

Pengurutan Bilangan Ascending

Syarat: nilai variable yang ditukar



Pengurutan Bilangan Ascending

Permasalahan:

Dengan menggunakan algoritma menukar dua variable, buatlah program yang melakukan pengurutan terhadap 3 buah bilangan yang diinputkan.

Contoh:

Input:

Masukkan bilangan 1: 66

Masukkan bilangan 2: 55

Masukkan bilangan 3: 44

Output:

Bilangan setelah diurutkan: 44 55 66

Pengurutan Bilangan Ascending

Cara Penyelesaian Masalah:

Cek bilangannya satu persatu. Mana yang lebih besar dibanding dengan bilangan lainnya.

Ilustrasi:

Bil1	Bil2	Bil3
------	------	------

66	55	44
----	----	----

If(bil1>bil2) → tukar kedua bilangan

Bil1	Bil2	Bil3
------	------	------

55	66	44
----	----	----

If(bil1>bil3) → tukar kedua bilangan

Bil1	Bil2	Bil3
------	------	------

44	66	55
----	----	----

If(bil2>bil3) → tukar kedua bilangan

Bil1	Bil2	Bil3
------	------	------

44	55	66
----	----	----

Pengurutan Bilangan Ascending

Struktur data atau variable yang dibutuhkan:

Variabel: bil1, bil2, bil3 dan....

temp

Input:

bil1, bil2, dan bil3

Output:

bil1, bil2, dan bil 3 yang isi variabelnya sudah ditukar sesuai urutan ascending

Proses Penyelesaian (Algoritma):

1. Inputkan nilai variable bil1, bil2 dan bil3.
2. Jika $bil1 > bil2$, maka tukar isi variable bil1 dengan bil2.
3. Jika $bil1 > bil3$, maka tukar isi variable bil1 dengan bil 3.
4. Jika $bil2 > bil3$, maka tukar isi variable bil2 dan bil3
5. Tampilkan isi variable bil1, bil2, dan bil3

Algoritma Tukar Bilangan:

(ingat tukar teh-kopi)

1. $temp = bil1$
2. $bil1 = bil2$
3. $bil2 = temp$

Apakah Algoritmanya **harus** seperti ini?

Pastinya **tidak**, semua orang punya cara masing-masing untuk menyelesaikan masalah.

Pengurutan Bilangan Ascending

Fungsi di C yang kita gunakan dalam koding ini:

- `printf()`
`printf("masukkan bilangan 1:");`
- `scanf()`
`scanf("%d", &bil1);`
- `if()`
`if(bil1 > bil2){`
`//coding`
`}`
- `printf("Bilangan 1 = %d", bil1);`

02

Pengecekan Zodiak

Syarat: if-else atau switch-case



Pengecekan Zodiak

Permasalahan:

Definisikan zodiak seseorang dengan menggunakan else-if atau switch-case













Contoh:

Input:

Masukkan Bulan dan Tanggal Lahir: 10 14

Output:

Bintang anda adalah Libra

Capricorn  22 Dec - 20 Jan	Aquarius  21 Jan -19 Feb	Pisces  20 Feb - 20 Mar	Aries  21 Mar- 19 Apr
Taurus  20 Apr - 20 May	Gemini  21 May - 21 Jun	Cancer  22 Jun - 23 Jul	Leo  24 Jul - 23 Aug
Virgo  24 Aug - 22 Sept	Libra  23 Sept - 22 Oct	Scorpio  23 Oct - 22 Nov	Sagittarius  23 Nov - 20 Dec

Pengecekan Zodiak

Cara Penyelesaian Masalah:

1. Cek bulannya apa
2. Cek rentang tanggalnya

Contoh:

bulan: 10, tanggal: 14

Zodiak yang ada **di bulan 10: Libra** dan **Scorpio**

Dimana, hasilnya **Libra** jika tanggalnya ≤ 22 atau **Scorpio** jika tanggalnya **selain itu**

Pengecekan Zodiak

Struktur data atau variable yang dibutuhkan:

Variabel: bulan dan tanggal

Input:

bulan dan tanggal

Output:

zodiaknya

Proses Penyelesaian (Algoritma):

1. Inputkan nilai variable bulan dan tanggal.
2. Jika bulan == 1, maka
 Jika tanggal <= 20, maka print Capricorn,
 Kalau tidak, maka print Aquarius
3. ... dst ... lanjutkan untuk bulan lainnya

Pengecekan Zodiak

Contoh Sebagian coding:

```
printf("Masukkan bulan dan tanggal:");  
scanf("%d %d", &bulan, &tanggal);
```

```
switch(bulan)  
{  
    case 1:  
        if(tanggal <=20){  
            printf("Bintang Capricorn");  
        }else{  
            printf("Bintang Aquarius");  
        }  
        break;  
    //...  
    default:  
        printf("Bulan tidak valid");  
}
```

Ingat cara penggunaan switch case!
Hanya untuk int

03

Konversi Bilangan Biner ke Desimal

Syarat: buat fungsi pangkat sendiri



Konversi Bilangan Biner ke Desimal

Permasalahan:

Dengan menggunakan fungsi, buatlah program untuk menampilkan hasil konversi bilangan biner ke bilangan decimal. Petunjuk buatlah fungsi pangkat sendiri untuk menghitung 2 pangkat n .

Contoh:

Input:

Masukkan bilangan biner: 101010

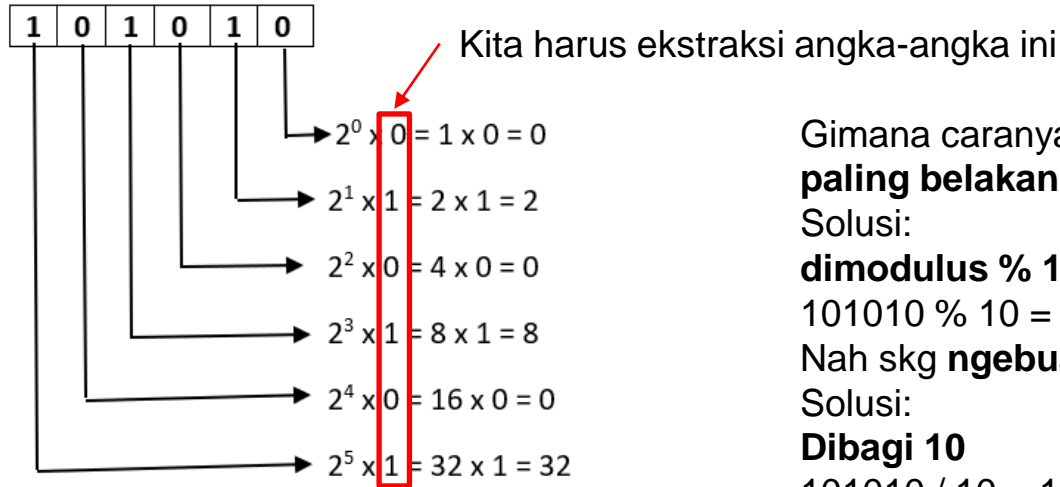
Output:

Bilangan decimal: 42

Konversi Bilangan Biner ke Desimal

Cara Penyelesaian Masalah:

Cara mengubah bilangan biner ke decimal:



Resultant decimal number = $0 + 2 + 0 + 8 + 0 + 32 = 42$

Gimana caranya, mengekstraksi nilai 0 paling belakang dari 101010 ?

Solusi:

dimodulus % 10

$101010 \% 10 = 0 \rightarrow$ butuh variabel

Nah skg ngebuang angka 0 dibelakang?

Solusi:

Dibagi 10

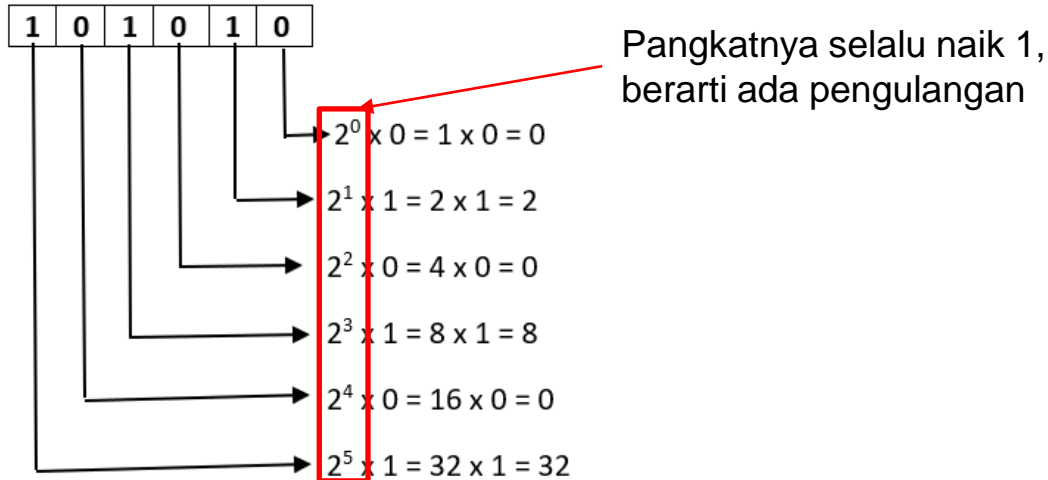
$101010 / 10 = 10101$

Dst..

Konversi Bilangan Biner ke Desimal

Cara Penyelesaian Masalah:

Cara mengubah bilangan biner ke decimal:



Resultant decimal number = $0 + 2 + 0 + 8 + 0 + 32 = 42$

Konversi Bilangan Biner ke Desimal

Struktur data atau variable yang dibutuhkan:

Variabel: biner, decimal, temp, i

Input:

biner

Output:

desimal

Proses Penyelesaian (Algoritma):

1. Inputkan nilai variable biner
2. $i = 0$
3. Jika $\text{biner} > 0$, maka,
 $\text{temp} = \text{biner} \% 10$
 $\text{desimal} = \text{temp} * \text{pangkat}(2, i)$
 $\text{biner} = \text{biner} / 10$
 $i++$
Kembali ke Langkah 3
4. Kalau Tidak, maka tampilkan nilai variable decimal.

Algoritma Fungsi Pangkat:

1. $\text{pangkat}(\text{base}, \text{exponent}) \rightarrow$ dapat inputan base dan exponent
2. $\text{hasil} = 1, i = 0$
3. jika $i < \text{exponent}$, maka
4. $\text{hasil} *= \text{base}$
5. $i++$
6. Kembali ke Langkah 3
7. Kalau tidak kembalikan nilai variable hasil ke fungsi utama

Konversi Bilangan Biner ke Desimal

Contoh coding:

```
for(int i=0; biner>0; i++){
    temp = biner % 10;
    desimal += temp * pangkat(2,i);
    biner /= 10;
}
```

```
int pangkat(int base, int exponent){
    int hasil = 1;
    for(int i =0; i < exponent; i++){
        hasil *=base;
    }
    return hasil;
}
```

04

Penjumlahan Bilangan Genap dan Ganjil dari 1 sampai n

Syarat: looping



Penjumlahan Bilangan Genap dan Ganjil dari 1 sampai n

Tampilkan nilai penjumlahan dari bilangan genap dan ganjil dari angka 1 sampai n. (Berarti ada proses pengecekan bilang genap atau ganjil)

Contoh:

Input:

n=10

Output:

Jumlah Ganjil = 25

Jumlah Genap = 30

Penyelesaian Masalah:

Dengan menggunakan looping, kita akan menggunakan nilai 1 sampai ke n dan mengecek nilai tersebut apakah ganjil atau genap. Dan mereka akan dijumlahkan sesuai dengan kelompok ganjil atau genap.

Penjumlahan Bilangan Genap dan Ganjil dari 1 sampai n

Struktur data atau variable yang dibutuhkan:

Variabel: n , i , $jmlGenap$, $jmlGanjil$

Input:

n

Output:

$jmlGanjil$, $jmlGenap$

Proses Penyelesaian (Algoritma):

1. Inputkan nilai n
2. $i = 1$
3. Jika $i \leq n$, maka
 4. Jika $i \% 2 == 0$, maka $jmlGenap$ ditambahkan nilai i
 5. Kalau tidak, maka $jmlGanjil$ ditambahkan nilai i
 6. $i++$
7. Kembali ke nomor 4
8. Kalau Tidak, maka tampilkan $jmlGanjil$ dan $jmlGenap$.

Penjumlahan Bilangan Genap dan Ganjil dari 1 sampai n

Contoh Sebagian coding

```
for(int i=1; i<=n; i++){  
    if(i%2 == 0){//genap  
        jmlGenap += i;  
    }else{//ganjil  
        jmlGanjil += i;  
    }  
}
```



05

Validator Tanggal

Syarat:Fungsi



Validator Tanggal

Dengan menggunakan fungsi, buatlah program yang melakukan validasi terhadap tanggal, bulan dan tahun yang diinputkan. Program juga melakukan pengecekan apakah sebuah tahun adalah tahun kabisat atau bukan (jika tahun bukan kabisat, bulan yang diinputkan bulan 2 tanggal di atas 28 maka tanggal tidak valid).

Contoh:

Input:

tanggal : 29

bulan : 2

tahun : 1999

Output : tanggal tidak valid

Validator Tanggal

Penyelesaian Masalah:

Cek bulan:

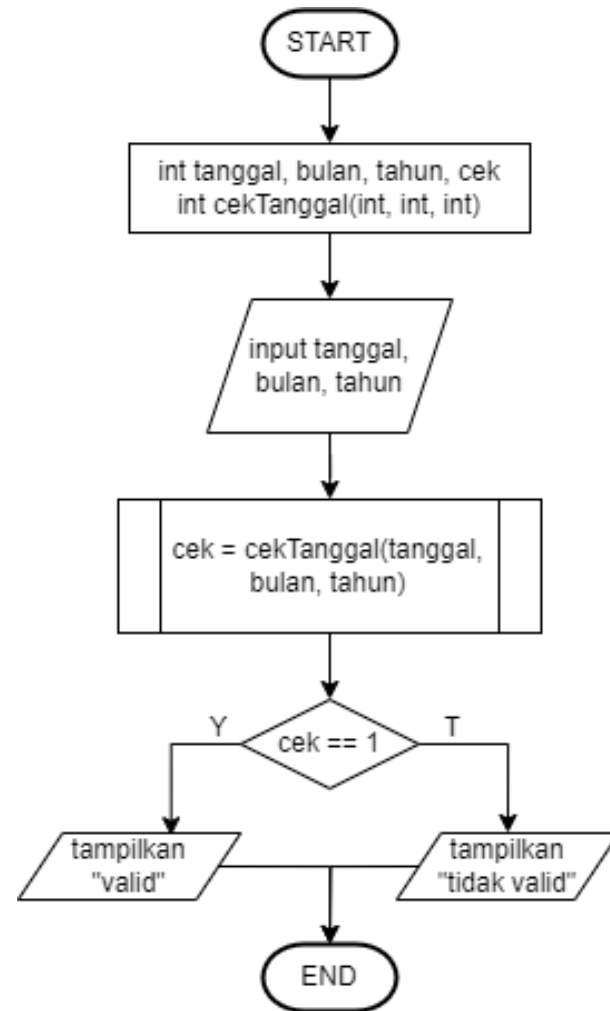
1. jika 1, 3, 5, 7, 8, 10, 12 maka tanggal dari 1 -31
2. Jika 4, 6, 9, 11 maka tanggal dari 1-30
3. Jika 2 maka cek jika tahun kabisat maka tanggal dari 1-29, jika tidak 1-28

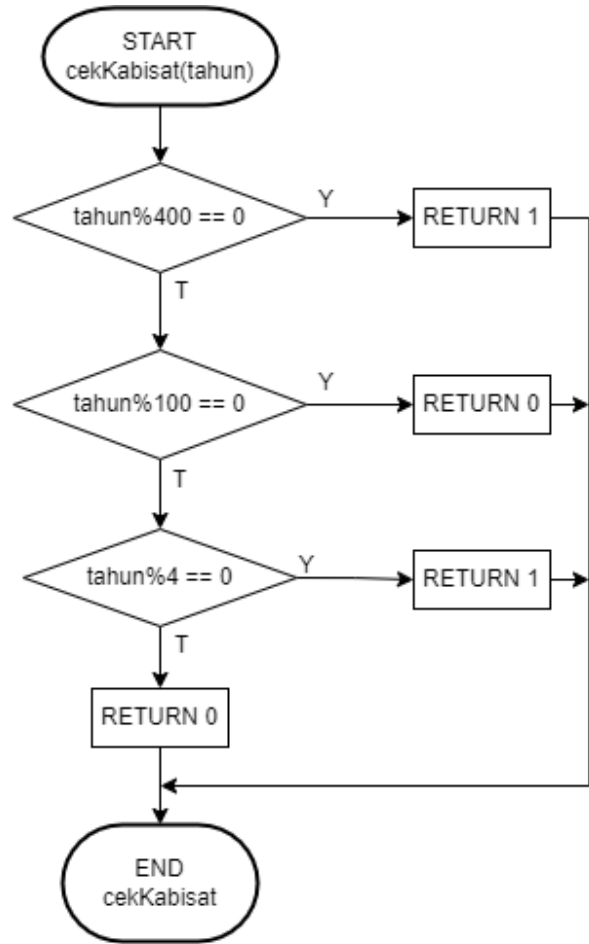
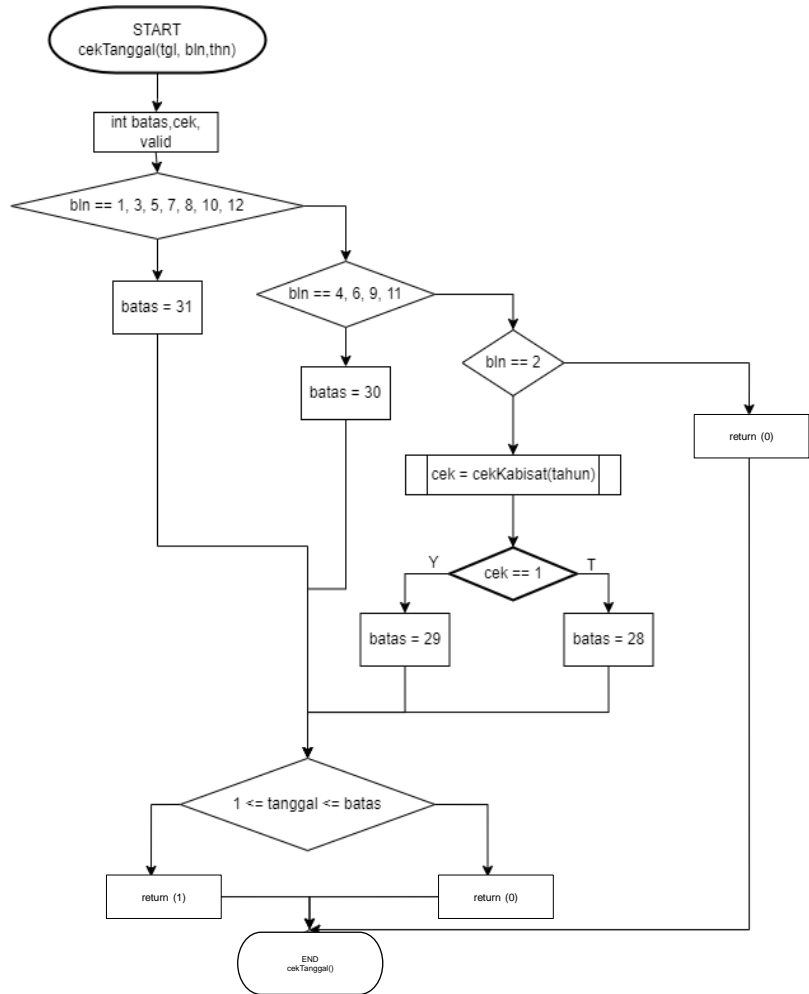
Cek tahun kabisat atau tidak dengan algoritma:

1. Jika angka tahun itu **habis dibagi 400**, maka tahun itu sudah pasti tahun **kabisat**.
2. Jika angka tahun itu **tidak habis dibagi 400 tetapi habis dibagi 100**, maka tahun itu sudah pasti **bukan** merupakan tahun kabisat.
3. Jika angka tahun itu **tidak habis dibagi 400, tidak habis dibagi 100** akan tetapi **habis dibagi 4**, maka tahun itu merupakan tahun **kabisat**.
4. Jika angka tahun **tidak habis dibagi 400, tidak habis dibagi 100**, dan **tidak habis dibagi 4**, maka tahun tersebut **bukan** merupakan tahun kabisat.

Validator Tanggal

<https://drive.google.com/file/d/1oXQavcf6ROab6JcuBDdOsPKpkUR97FDs/view?usp=sharing>





Validator Tanggal

contoh sebagian coding:

```
printf("Masukkan tanggal bulan tahun: ");
scanf("%d %d %d", &tanggal, &bulan, &tahun);

cek = cekTanggal(tanggal, bulan, tahun);
if(cek){
    printf("Tanggal Valid");
}else{
    printf("Tanggal Tidak Valid");
}
```

```
switch(bulan){
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12:
        batas = 31;
        break;

    case 4:
    case 6:
    case 9:
    case 11:
        batas = 30;
        break;

    case 2:
        cekTahun = cekKabisat(tahun)
        if(cekTahun) batas = 29;
        else batas=28;
        break;

    default:
        return(0);
        break;
}
```

```
if(1<=tanggal && tanggal <= batas){
    return 1;
} else{
    return 0;
}
```

```
int cekKabisat(int tahun){
    if((tahun % 400) == 0) return 1;
    else if(tahun %100 == 0) return 0;
    else if(tahun % 4 == 0) return 1;
    else return 0;
}
```



06

Bilangan Sempurna 1 sampai n

Syarat:Fungsi

Bilangan Sempurna 1 sampai n

Menampilkan bilangan sempurna dari 1 sampai n. Bilangan sempurna adalah bilangan yang jika seluruh faktor pembaginya dijumlahkan akan menghasilkan bilangan itu sendiri.

Contoh:

$$6 \rightarrow 1 + 2 + 3 = 6$$

$$28 \rightarrow 1 + 2 + 4 + 5 + 7 + 14 = 28$$

Input: n = 100

Output: Bilangan sempurna = 6 28

Kita akan menggunakan nested looping:

- dengan asumsi **nilai i** adalah **angka yang akan dicek**
- **nilai j** sebagai faktor pembaginya, dimana **nilai j** jika dia merupakan faktor pembagi maka akan ditambahkan dengan faktor pembagi lainnya.
- **Ingat 1 bukan bilangan sempurna**

Bilangan Sempurna 1 sampai n

Struktur data atau variable yang dibutuhkan:

Variabel: n, i, j, bilSmp

Input:

n

Output:

Bilangan sempurna

Proses Penyelesaian (Algoritma):

1. Inputkan nilai n
2. Lakukan pengulangan dari $i=1$ sampai ke n
3. Didalam pengulangan tersebut ada proses pengecekan nilai i apakah habis dibagi dengan nilai j. Dimana nilai j dimulai dari 1 sampai kecil dari i.
4. Jika nilai i habis dibagi nilai j maka variable bilSmp ditambahkan dengan nilai j-nya.
5. Jika variabel bilSmp nilainya sama dengan i maka ditampilkan.

Bilangan Sempurna 1 sampai n

Contoh Sebagian koding:

```
for(int i=1; i < n ; i++){
    bilSmp = 0;
    for(int j=1; j < i ;j++){
        if(i%j == 0){
            bilSmp += j;
        }
    }
    if(bilSmp == i){
        printf("%d ", bilSmp);
    }
}
```


Bisa dilanjutkan Praktikum Array

Sampai waktu selesai

Minggu Depan Proses Pengecekan
Praktikum satu persatu kedepan



END OF PRESENTATION



THANK YOU

makeameme.org



Resources

- Pengantar Logika dan Algoritma, Ali Ridho Barakbah;Tita Karlita
- <http://tita.lecturer.pens.ac.id/LA/>
- <http://yuliana.lecturer.pens.ac.id/Logika%20dan%20Algoritma/TEORI/>
- Algorithms, Yi-Shin Chen, Addison-Wesley Computing
- Algoritma dan pemrograman dalam bahasa Pascal dan C, Rinaldi Munir, penerbit Informatika Bandung
- Karl, Beecher. "Computational Thinking: A Beginner's Guide to Problem-Solving and Programming." Swindon, UK: BCS, The Chartered Institute for IT (2017).
- 4Knuth, Donald E. "Algorithms." Scientific American 236.4 (1977): 63-81.
- Konsep Pemograman, Umi Sa'adah, Entin Martiana Kusumaningtyas, Tri Hadiyah Muliawati