1. FAKTORIAL

* DESKRIPSI

1. Masukkan angka yang akan difaktorialkan!, angka akan dinotasikan dengan “n”,
2. Hitung dengan rumus: n! = n x (n-1) x (n-2) x (n-3) … x (n-(n-1))
3. Tampilkan faktorial dari n.

* PSEUDOCODE

**Algoritma**

{Menghitung faktorial dari suatu angka jika input angka tersebut diberikan}

**Deklarasi**

Int n

**Deskripsi**

read n

f = n

While n > 1

Do

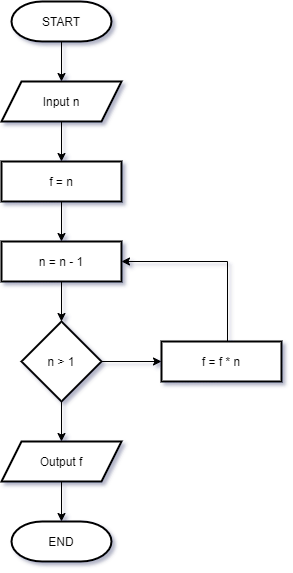
n = n – 1

f = f \* n

While end

write f

* FLOWCHART



1. KONVERSI WAKTU

* DESKRIPSI

1. Masukkan detik! Detik akan dinotasikan dengan “s”,
2. Detik akan dikonversi menjadi jam : menit : detik dengan rumus:

Jam = s / 3600;

Menit = (s - (jam \* 3600)) / 60;

Detik = (s - (jam \* 3600) - (menit \* 60));

1. Tampilkan jam : menit : detik.

* PSEUDOCODE

**Algoritma**

{Mengonversi detik ke format jam apabila detik tersebut diberikan}

**Deklarasi**

Int s, jam, menit, detik,

**Deskripsi**

read s

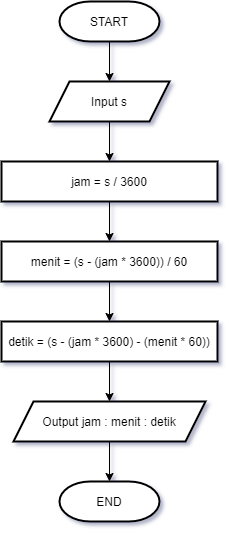
jam = s / 3600

menit = (s - (jam \* 3600)) / 60

detik = (s - (jam \* 3600) - (menit \* 60))

write s detik = jam : menit : detik

* FLOWCHART



1. SORTING

* DESKRIPSI

1. Input kumpulan bilangan (a)
2. Menentukan hasil = bilangan pertama (a[0])
3. Melakukan perulangan i dari 2 hingga banyak bilangan, jika hasil < a[i], maka ganti nilai hasil dengan a[i] tambahkan i = i + 1
4. Write (hasil/output).
5. Cetak hasil

* PSEUDOCODE

**Algoritma**

{Mengurutkan angka dari suatu kelompok bilangan secara ascending}

**Deklarasi**

int a[10]

**Deskripsi**

Read (a[])

Output = a[0]

For int i = 1 until (banyak bilangan) - 1

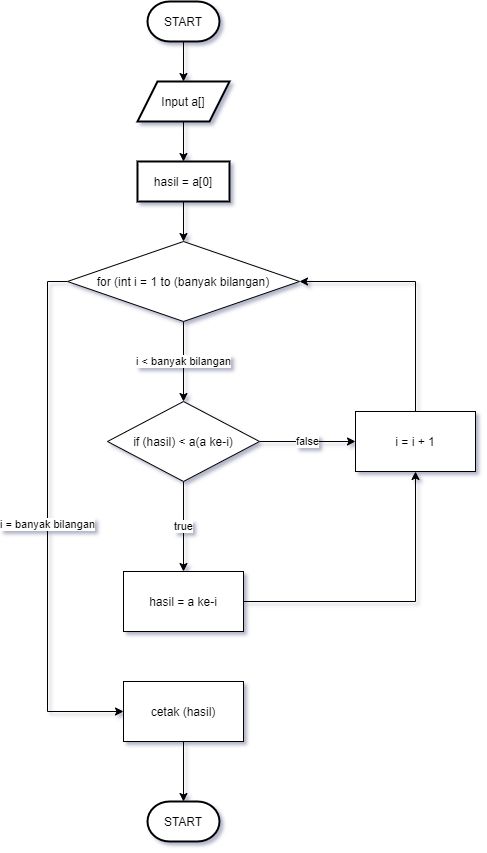
if (output < a[i])

output = a[1]

i = i + 1

write (hasil/output)

* FLOWCHART



1. PANGKAT

* DESKRIPSI

1. Masukan angka (n) dan pangkat (m)
2. Menentukan nilai hasil = 1
3. Jika m == 0 maka hasil akan = 1
4. Jika m > 0 maka lakukan:

Perulangan I dari 1 hingga m

Kalikan hasil = hasil \* n, tambahkan i = i + 1

1. Jika m < 0 maka lakukan perulangan I dari -1 hingga m

Kalikan hasil = hasil \* n, kurangi i = i – 1

Setelah akhir perulangan,ganti nilai hasil =1 / hasil

1. write hasil

* PSEUDOCODE

**Deklarasi**

int a, hasil

float b

**Deskripsi**

Read (b, a)

hasil = 1

if (a == 0)

hasil = 1

end if

else if (a > 0)

for i = 1 until a

hasil = hasil \* b

i = i - 1

else if (a < 0)

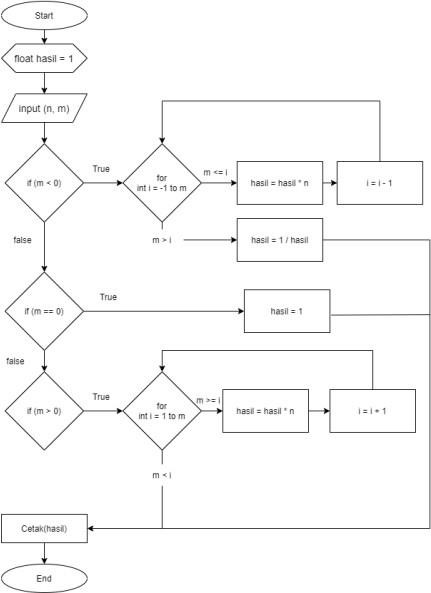
for i = - 1 until a

hasil = hasil \* b

i = i - 1

write (hasil)

* FLOWCHART



1. BILANGAN PRIMA (1-50)

* DESKRIPSI

1. Membuat perulangan dari i = 1 hingga 50
2. Jika i < 2, maka i bukan bilangan prima
3. Jika i = 2, maka i adalah bilangan prima
4. Jika I > 2, maka akan mengecek ke bilangan i,dengan melakukan

perulangan dari j = 2 hingga i

1. Jika angka i dibagi dengan j sisanya = 0, maka angka i bukan bilangan prima
2. Jika angka i dibagi dengan j sisanya tidak = 0, maka i adalah bilangan prima
3. Jika bilangan adalah prima maka output akan keluarkan bilangan prima

* PSEUDOCODE

**Deklarasi**

int n = 50

boolean isPrime

**Deskripsi**

For i = 1 until n = 50

isPrime = true;

if (i < 2)

isPrime = false;

if (i == 2)

isPrime = true;

if (i > 2)

for j = 2 until i

if (i % j = 0)

isPrime = false;

if (isPrime == true)

write (i)

* FLOWCHART

