

# **ALGORITMA PEMROGRAMAN**

## **Pertemuan XIII**

# **SUB PROSES - PROSEDUR**

**Oleh**  
**Achmad Arrosyidi**



# TUJUAN PEMBELAJARAN

## Umum:

- ✓ Mahasiswa dapat membuat Sub Program / Sub Proses - Prosedur / *Procedure* dalam bentuk *flowchart*.

## Khusus:

- ✓ Mahasiswa dapat menerapkan Sub Program / Sub Proses - Prosedur / *Procedure* dalam bentuk *flowchart*.



# MATERI KULIAH

1. Pengantar
2. Manfaat
3. Parameter
4. Prosedur
5. Latihan
6. Ringkasan



# 1. PENGANTAR



# 1. PENGANTAR

1. Seringkali sebuah proses besar diselesaikan dengan cara membaginya menjadi beberapa proses yang lebih kecil → sub proses.
2. Setiap sub-proses melakukan sebuah kegiatan spesifik yang berbeda dengan sub-proses lainnya, sehingga tidak boleh sampai terjadi overlapping tugas.
3. Sebuah sub-proses diperbolehkan memanggil sub-proses lain untuk melakukan kegiatan tertentu sesuai dengan kebutuhan. Sub-proses yang dipanggil ini boleh memanggil sub-proses lain, demikian seterusnya tanpa batas.
4. Setelah menyelesaikan kegiatan di dalam sub-proses, alur proses akan melanjutkan instruksi pada lokasi yang telah ditinggalkan. Address memory yang ditinggalkan tadi disimpan di dalam **STACK**.



## 2. MANFAAT



## 2. MANFAAT

1. Memudahkan proses penyelesaian masalah.
2. Memudahkan pembacaan program.
3. Memudahkan penelusuran alur program.
4. Memudahkan proses debugging, yaitu mencari dan memperbaiki kesalahan yang terjadi di dalam sistem.
5. Memudahkan pengembangan sistem.



# 3. PARAMETER



1. PARAMETER adalah data yang dikirimkan ke sub-proses untuk diolah di dalam sub-proses tersebut.
2. Keberadaan PARAMETER itu sendiri adalah tidak harus ada, tergantung sub-proses yang dibuat.
3. Sebuah sub-proses boleh tidak memiliki PARAMETER atau memiliki PARAMETER lebih dari satu dengan tipe beragam.
4. Sebuah PARAMETER bertindak sebagai variabel lokal bagi sub-proses tempat dimana dia dideklarasikan. Tipe data untuk parameter mengikuti aturan yang berlaku pada variabel.
5. Ada kalangan yang menyebutkan istilah PARAMETER dengan nama
6. ARGUMEN. Keduanya sama saja, hanya berbeda istilah.



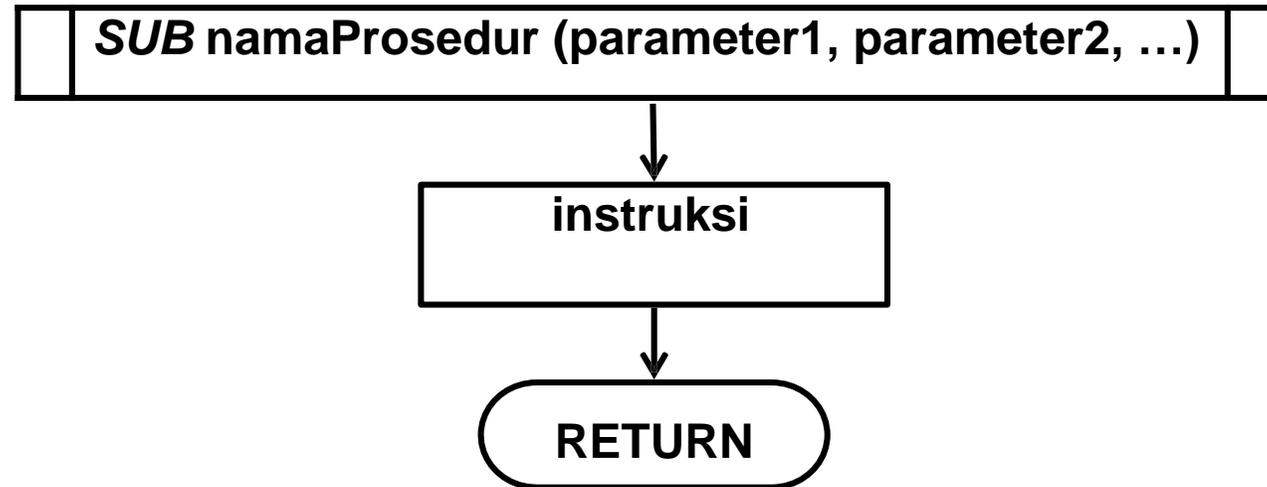
## 4. PROSEDUR



1. Sub-proses jenis prosedur tidak membawa nilai-balik setelah melakukan proses tertentu.
2. Sebuah prosedur dipanggil dengan cara menuliskan namanya secara langsung setelah keyword CALL.
3. Jika diperlukan, pemanggilan terhadap prosedur dilakukan dengan menyertakan sejumlah data yang dibutuhkan oleh prosedur tersebut agar bisa bekerja dengan baik.
4. Sebuah prosedur diawali dengan keyword SUB dan diakhiri dengan keyword RETURN.
5. Istilah procedure dalam Raptor adalah Subchart

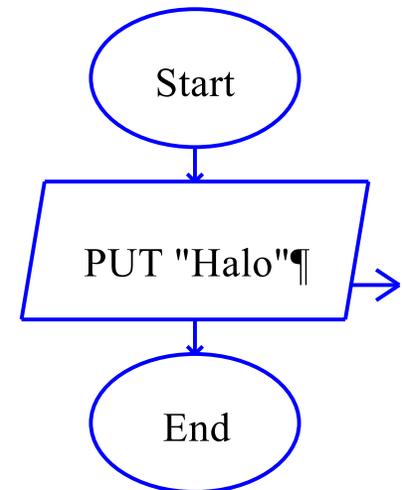
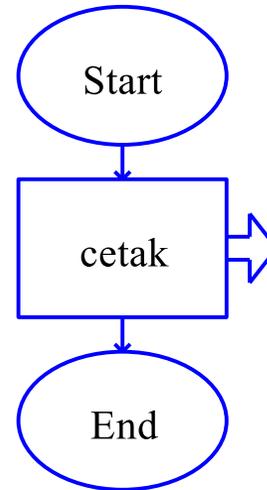
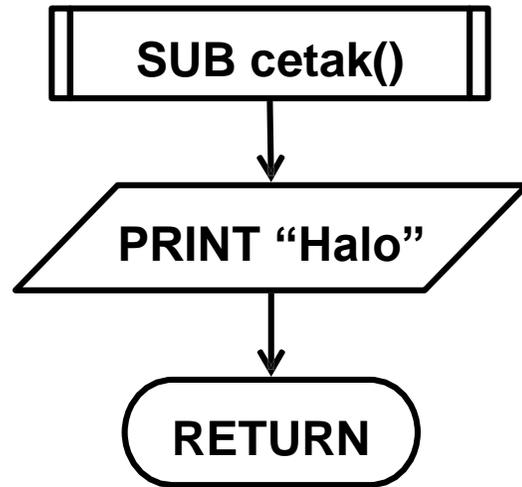
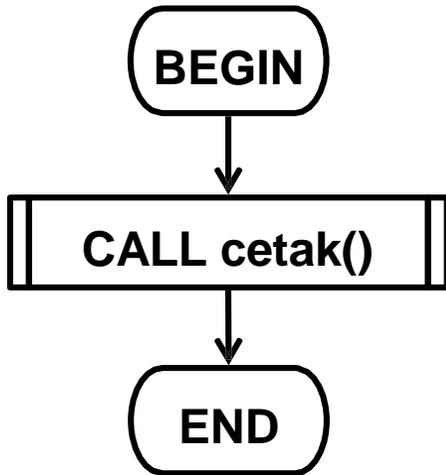


## 4. PROSEDUR



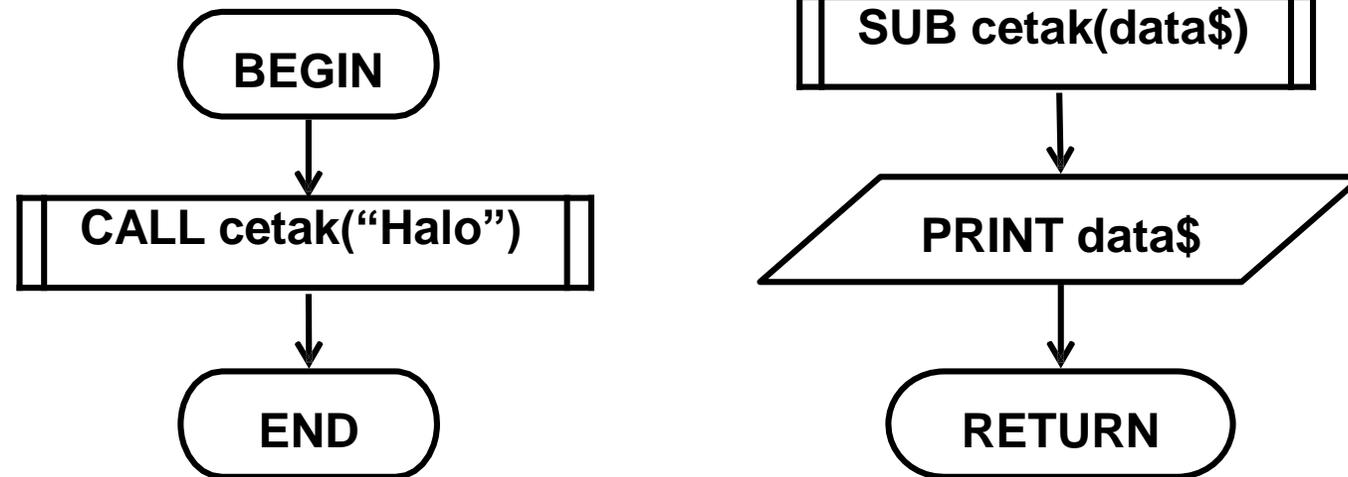
Contoh #1

## Tanpa Parameter Passing

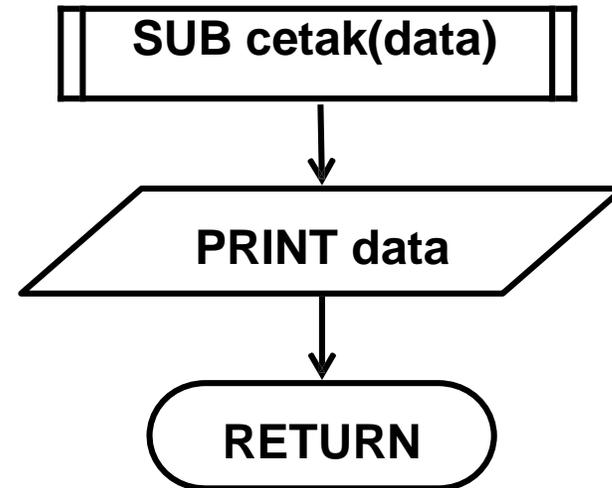
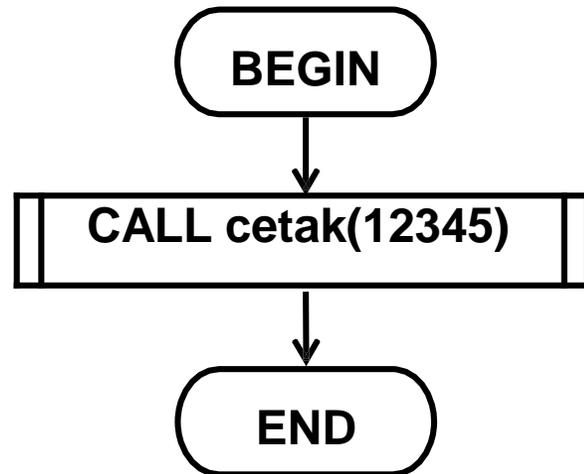


## Dengan Parameter Passing

Contoh #2



Contoh #3



## 5. LATIHAN



1. Menampilkan deret bilangan: cacah, genap, gasal, fibonacci
2. Menampilkan pola segitiga: pascal, genap, gasal
3. Perhitungan statistik: MAX, MIN, AVERAGE, TOTAL
4. Perhitungan matematika: perpangkatan, perkalian, FPB, KPK
5. Menampilkan pola bintang



## 6. RINGKASAN



FITUR	TEORI	RAPTOR
Istilah	Prosedur / Procedure	Subchart
Tanpa Parameter	√	√
Dengan Parameter	√	<b>X</b>
Jumlah Parameter Input	~	<b>X</b>
Return Value	<b>X</b>	<b>X</b>
Jumlah Parameter Output / Return Value	<b>X</b>	<b>X</b>



**SELESAI  
TERIMA KASIH**

