

# SUBROUTINE PADA COBOL

---

## OBJEKTIF :

1. Mahasiswa Mengetahui Subroutine pada Bahasa COBOL.
  2. Mahasiswa Mampu Membuat dan Menjalankan Subroutine pada Bahasa COBOL.
  3. Mahasiswa Mampu Memahami Struktur Pernyataan Call pada Bahasa COBOL.
- 

## 11.1 SUBROUTINE PADA COBOL

Subroutines adalah program pada COBOL yang bisa di *compile* secara *independent* (mandiri) tapi tidak bisa di *execute* secara *independent* (mandiri). Maksudnya, terdapat suatu program utama (main program) dan satu atau lebih program bagian (subprogram atau subroutines). Program utama adalah program yang siap untuk dijalankan, program bagian dapat di tulis dan decompile secara terpisah tetapi hanya dapat dijalankan melalui program utama

Subroutines dapat berkomunikasi dan diproses melalui program utama. Ini dapat terjadi karena pada COBOL terdapat statement yang dapat menghubungkan modul program yang satu dengan modul program lainnya, sehingga antar program dapat saling berkomunikasi. Komunikasi tersebut dapat dilakukan dengan pernyataan CALL.

Cobol mendukung internal subroutine dan eksternal subroutine. Internal subroutines diimplementasikan dalam bentuk statement Perform dan eksternal subroutines diimplementasikan dalam bentuk statement CALL. Jika mempelajari BASIC language, maka sudah familiar dengan internal subroutines sedangkan jika mempelajari C, Modula-2, atau JAVA maka akan familiar dengan eksternal subroutines.

## 11.2 PERNYATAAN CALL

Ketika suatu tugas tertentu harus dikerjakan dengan memerlukan lebih dari satu program, maka solusinya adalah dengan membuat sebuah subprogram. Pernyataan Call dibutuhkan untuk memanggil sub program tersebut untuk melakukan tugas. Call digunakan untuk mengirimkan proses dari satu program ke program lain.

Program yang berisikan pernyataan call disebut *Calling program* (program utama) sedangkan program yang dipanggil disebut *Called program* (subprogram). Calling program yang dieksekusi akan berhenti sampai dengan called program selesai dieksekusi. Pada called program digunakan statement EXIT untuk mengembalikan proses.

### Hal-hal yang harus diperhatikan dalam *Called program*

- Linkage section, harus didefinisikan dalam *called program*. Yang berisi data elemen yang dilewatkan dalam program. item data tidak boleh memiliki Value clause. Dan Picture clause kompatibel dengan variable yang melewati *calling program*.
- Procedure division using, , berisikan daftar variabel yang diteruskan dari *calling program*, urutannya harus sama dengan yang di pernyataan Call.
- Exit program, untuk mengembalikan proses ke program utama dari program bagian, dan sebagai pengganti *STOP RUN*. Jika digunakan *STOP RUN* maka proses tidak bisa kembali ke program utama.

## Dua Tipe Jenis *Call* Berdasarkan Bagaimana Subprogram (*Called program*) Dipanggil

### Static Call

Static call terjadi ketika program di compile dengan *NODYNAM* compiler. Static call dapat ditentukan berdasarkan sintaks dari pernyataan Call.

Bentuk sintaks Static Call:

```
CALL 'sub-program-name'
```

### Dynamic Call

Dynamic call terjadi ketika program di compile dengan *DYNAM* dan *NODLL* compiler. Dynamic call dapat ditentukan berdasarkan sintaks dari pernyataan Call.

Bentuk sintaks Dynamic Call :

```
CALL 'nama-paragraph '
```

## Dua Tipe Jenis *Call* Untuk Melewatkan Parameter Dari Program Utama Ke Subprogram

### Call by Reference

Jika nilai variable dalam *Called Program* di modifikasi, maka nilai yang diubah tersebut juga akan ditampilkan pada *Calling Program*. Ini disebabkan karena penggunaan memori yang sama untuk parameter baik pada *Calling Program* maupun *Called Program*. Dengan kata lain setiap perubahan nilai pada sub program akan tercermin dalam program utama, karena program menggunakan memori yang sama.

Bentuk umum :

```
CALL nama-subprogram USING var-1, var-2
```

Call by reference berlaku untuk *static call* dan *dynamic call*.

### Call by Content

Jika nilai variable dalam *Called Program* di modifikasi, maka nilai yang diubah tersebut tidak akan ditampilkan pada *Calling Program*. Karena *Calling Program* membagikan content parameter ke *Called Program*.

Bentuk umum :

```
CALL nama-subprogram USING  
BY CONTENT var-1, BY CONTENT var-2
```

Call by Content berlaku untuk *static call* dan *dynamic call*.