

## BAB 3

### STRUKTUR KENDALI, SUBROUTINE, DAN FUNGSI

<b>TUJUAN</b>
---------------

**Tujuan Instruksi Umum:**

- § Menjelaskan kepada mahasiswa mengenai struktur kendali pada Fortran
- § Menjelaskan Kepada mahasiswa mengenai Function dan subroutine
- § Mengimplementasikan perintah struktur kendali pada Fortran

**Tujuan Instruksi Khusus:**

- § Agar praktikan dapat membuat program sederhana dengan menggunakan statemen struktur kendali
- § Agar praktikan dapat membuat program sederhana dengan menggunakan statement Function dan subroutine

<b>MATERI</b>
---------------

#### 3.1 Stuktur Kendali Dalam Bahasa FORTRAN

Ada beberapa statement stuktur kendali pada bahasa pemrograman fortran yang dapat digunakan dalam membuat program aplikasi. Statement tersebut antara lain:

- a) Statement percabangan
  - a.1) If Statement

Statement If merupakan statement yang dapat digunakan untuk menyeleksi suatu keadaan atau syarat dan proses akan melakukan suatu tindakan tertentu jika kondisi yang diseleksi benar dan melakukan tindakan lainnya jika kondisi salah. If Statement dapat digolongkan menjadi 3 yaitu:

- 1. Logical If

Bentuk Umum:

**IF** (<Ekspresi>) <Statement>      dimana  
<ekspresi>      adalah ungkapan logika yang akan diseleksi

<Statement> adalah *executeable statement* kecuali statement *Do, End, Endif, Elseif, Else, blok if* atau statement *if* logika lainnya.

## 2. Arithmetic If

Bentuk Umum:

**IF** (<ekspresi>) <slabel1>, <slabel2>, <slabel3>

<ekspresi> adalah ungkapan aritmatic real atau integer

<slabel1> adalah proses yang akan dilaksanakan jika ungkapan aritmatic yang diseleksi bernilai negatif

<slabel2> adalah proses yang akan dilaksanakan jika ungkapan aritmatic yang diseleksi bernilai nol

<slabel3> adalah proses yang akan dilaksanakan jika ungkapan aritmatic yang diseleksi bernilai positif

## 3. Block If

Statemen ini digunakan untuk menyeleksi suatu kondisi dan mengambil tindakan apa yang harus dilakukan dalam bentuk blok-blok statement.

Bentuk Umum:

Bentuk Umum **IF – THEN**

**IF** (<ungkapan>) **THEN**

Bentuk Umum **ELSE**

**ELSE**

Bentuk Umum **ELSEIF**

**ELSEIF** (<ungkapan>) **THEN**

Bentuk Umum **ENDIF**

**ENDIF**

Suatu statemen Blok If harus diawali dengan statemen **IF-THEN** dan diakhiri dengan statement **ENDIF**. Suatu loncatan menuju kedalam Blok If tidak diperbolehkan, tetapi loncatan keluar Blok If diperbolehkan.

### a.2) Statement Lompatan

#### 1. Statement GOTO

Statement GOTO merupakan statemen yang dapat digunakan untuk melompat menuju ke statemen yang lainnya. Statemen GOTO dapat dibedakan atas:

a. GOTO tak bersyarat

Bentuk Umum:

**GOTO** <Label>

Statement ini digunakan untuk mengontrol proses untuk menuju ke suatu statemen lainnya yang ditujukan oleh label tanpa ada syarat yang diberikan.

b. GOTO Pengerjaan

Bentuk Umum:

**GOTO** <nama> [[,] (<label1> [, <label2>]...)]

Statemen GOTO ini digunakan untuk melompat kesuatu label statemen yang ditunjukkan oleh isi dari <nama>

<nama> adalah variable integer yang diisi oleh label dengan perintah Assign

c. GOTO Bersyarat

Bentuk Umum:

**GOTO** (<label1> [, <label2>]...)[,] <i>

Statemen ini digunakan untuk mengontrol loncatan dari proses ke suatu label tertentu tergantung dari nilai ungkapan integer <i>

2. Statement Perulangan

a. Statement DO – CONTINUE

Kegunaan Statement DO adalah untuk melaksanakan proses berulang statement yang ada diantara statemen DO dan label

Bentuk Umum:

**DO** <Label> [,] <variable> = <ekspresi1>, <ekspresi2>, [, <ekspresi3>]

Sedangkan Statemen CONTINUE digunakan sebagai *Dummy Terminal* dalam statement DO

Disamping itu masih terdapat statement *Subroutine* dan *Function* yang fungsi dan kegunaannya hampir sama dengan statement *Procedure* dan *Function* pada bahasa pemrograman Palsal.

### 3.2 Subroutine Dalam Bahasa FORTRAN

Statement ini dapat digunakan untuk mengidentifikasi suatu unit program adalah suatu rutin bagian serta sekaligus memberikan nama dan argumen-argumennya

Bentuk umum:

**SUBROUTINE** <nama\_subroutine> [<parameter1>, <parameter2>..]

Keterangan :

<subroutine-name> adalah nama dari rutin bagian

<parameter> adalah nama dari argumen, atau disebut juga dengan *dummy argumen*

Berikut ini diberikan beberapa ketentuan dari *subroutine*:

- a) Subroutine merupakan unit program tersendiri yang diawali dengan statement SUBROUTINE dan diakhiri dengan statement END atau RETURN, serta dipergunakan oleh unit program yang lain dengan statement CALL.

Bentuk Pemanggilan:

**Program Utama**

....  
**Call A(X, HASIL)**

....  
....  
**END**  
**Subroutine A(X, Y)**

....  
....  
                    ↘ Argumen subroutine  
                    ↘ Nama subroutine

**END**

- b) Nama subroutine dapat berisi statement-statement apapun kecuali statement PROGRAM, statement SUBROUTINE lainnya, maupun statement FUNCTION
- c) Nama argument tidak boleh tampak di statemenat COMMON, EQUIVALENCE, INTRINSIC, atau DATA
- d) Argumen sesungguhnya yang tampak di statement CALL harus sesuai urutannya, jumlahnya dan tipenya dengan *dummy argument* yang tampak di statement Subroutine. Antara nama argument sesungguhnya dengan *dummy argument* bole sama maupun tidak.
- e) Di dalam suatu suroutine dapat memaggil subroutine yang lainnya.
- f) Subroutine dapat tidak mengandung *dummy argument* , yang berarti tidak ada data yang dikirim ke subrroutine dan tidak ada hasil yang dikirim balik ke pemanggil subroutine

Contoh :

```
CALL CETAK
WRITE (*, *) 'FORTRAN'
CALL CETAK
END
SUBROUTINE CETAK
WRITE (*, *) '-----'
END
```

Outputnya :

```
-----
FORTRAN
-----
```

### 3.3 Fungsi Dalam Bahasa FORTRAN

Fungsi dalam bahasa FORTRAN terdiri dari 3 macam, yaitu fungsi statement (*statement function*), fungsi eksternal (*external function*), dan fungsi intrinsik (*intrinsic function*).

#### 3.3.1 Fungsi Statement

Kegunaannya adalah mendefinisikan sebuah fungsi dalam bentuk sebuah statement tunggal.

Bentuk umum :

$\langle \text{fname} \rangle ( [\langle \text{farg} \rangle, [\langle \text{farg} \rangle] ] ) = \langle \text{expr} \rangle$

Keterangan :

$\langle \text{fname} \rangle$  adalah nama dari fungsi statement yang didefinisikan

$\langle \text{farg} \rangle$  adalah nama dari formal argumen fungsi yang berupa data yang diberikan kepada fungsi tersebut

$\langle \text{expr} \rangle$  adalah suatu ungkapan fungsi statement yang didefinisikan

Implementasinya ada pada contoh berikut ini :

Akan dilakukan konversi dari nilai derajat Celcius ke derajat Fahrenheit dalam rumus :

$$F = 1.8 * C + 32$$

Rumus matematika tersebut dikonversikan ke dalam fungsi statement sebagai berikut :

$$\text{FAH}(C) = 1.8 * C + 32$$

Tampak bahwa FAH adalah merupakan fungsi dari C yang besarnya ditunjukkan lewat ungkapan  $1.8 * C + 32$ .

### 3.3.2 Fungsi Eksternal

Kegunaannya adalah untuk mengidentifikasi suatu unit program sebagai suatu fungsi eksternal.

Bentuk umum :

[<type>] FUNCTION <fname> ([<farg>, [<farg>]...])

Keterangan :

<type> adalah tipe dari fungsi eksternal, dapat berupa :

INTEGER, INTEGER\*2, INTEGER\*4, REAL, REAL\*4, REAL\*8

DOUBLE PRECISION, LOGICAL\*2, LOGICAL\*4

Tipe dari fungsi eksternal tidak dapat berupa CHARACTER dan bila <type> dituliskan, maka nama dari fungsi eksternal tidak boleh didefinisikan ulang dengan statement type yang lainnya. Bila <type> tidak dituliskan, maka tipe dari fungsi eksternalnya adalah mengikuti tipe dari ungkapannya

<fname> adalah nama yang dipilih oleh pembuat program untuk nama fungsi eksternal

<farg> adalah nama dari argumen atau parameter dari fungsi eksternal. Nama dari fungsi eksternal harus digunakan sebagai suatu variabel di dalam unit program fungsi eksternal tersebut, yang menampung hasil akhir dari fungsi eksternal.

Implementasi fungsi eksternal ada dalam kasus konversi nilai derajat Celcius ke Fahrenheit, dalam ungkapan FUNCTION FAH(C). Potongan ungkapan ini bila dituliskan dalam program konversi menggunakan FORTRAN adalah :

**C fungsi eksternal menghitung konversi nilai derajat**

**FUNCTION FAH(C)**

**FAH = 1.8 \* C + 32**

**END**

Nama dari fungsi eksternal harus digunakan sebagai suatu variabel di dalam unit program fungsi eksternal tersebut, yang menampung hasil akhir dari fungsi. Nama dari fungsi eksternal tersebut adalah FAH, harus digunakan sebagai suatu variabel untuk menampung hasil akhir dari fungsi eksternal tersebut, yaitu :

$$FAH = 1.8 * C + 32$$

### 3.3.3 Fungsi Intrinsik

Fungsi intrinsik FORTRAN adalah fungsi yang telah tersedia dalam kamus bahasa FORTRAN, misalnya fungsi instrinsik yang dapat digunakan untuk mencari akar suatu nilai, yaitu memangkatkan nilai tersebut dengan 0.5 atau menggunakan fungsi SQRT. Keduanya dapat langsung disatukan dengan *source codenya*.

Untuk dapat mengetahui langkah-langkah pembuatan programnya, perhatikan dan ikuti dengan seksama Activity Lab yang telah disediakan.