



**UNIVERSITAS SANATA DHARMA  
YOGYAKARTA**

**JURUSAN TEKNIK INFORMATIKA  
PROGRAM HIBAH TIK-K1 2007**

PEMROGRAMAN  
BASIS DATA

Pertemuan V  
**FLOW CONTROL**

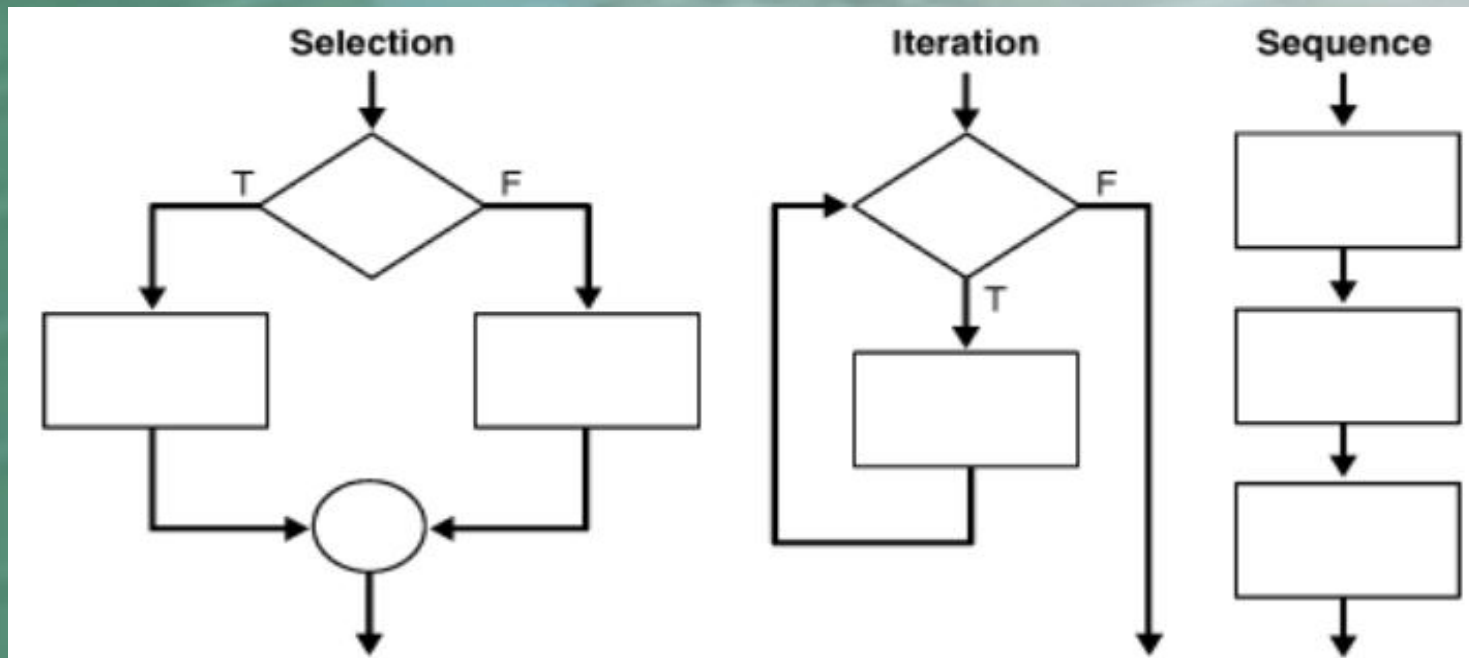
UNIVERSITAS SANATA DHARMA  
YOGYAKARTA

# TUJUAN

1. Mahasiswa dapat memahami berbagai struktur kontrol dalam bahasa pemrograman.
2. Mahasiswa dapat memahami berbagai jenis flow control yaitu IF, CASE, LOOP, WHILE, REPLACE ITERATE dan LEAVE
3. Mahasiswa dapat menggunakan dengan tepat berbagai jenis flow control.

# Struktur Kontrol

Struktur kontrol dalam bahasa pemrograman prosedural terdiri dari 3 yaitu: seleksi, iterasi dan sequential. Berikut flow chart untuk struktur kontrol



# Struktur Seleksi dengan IF dan CASE

## Syntak untuk IF Statement

```
IF search_condition THEN statement_list  
    [ELSEIF search_condition THEN  
statement_list] ...  
    [ELSE statement_list]  
END IF
```

IF Statement akan mengevaluasi *search\_condition* dan jika kondisinya TRUE maka akan dijalankan *statement\_list* setelah THEN.



# Struktur Seleksi dengan IF dan CASE (cont)

Dalam MySQL juga terdapat fungsi IF yang berbeda sintak dengan IF statement diatas. Fungsi IF memiliki sintak :

**IF** (*expr1* , *expr2* , *expr3* )

Jika *expr1* is TRUE (*expr1* <> 0 and *expr1* <> NULL) maka IF() akan mengembalikan *expr2* jika tidak akan mengembalikan *expr3*. IF() akan mengembalikan nilai numerik atau string, tergantung pada context yang digunakan.

# Struktur Seleksi dengan IF dan CASE (cont)

Contoh:

```
DELIMITER $$  
CREATE PROCEDURE `pbd`.`spContohCase` (v_grade char(1))  
BEGIN  
  declare v_appraisal varchar(20);  
  IF v_grade='A' THEN set v_appraisal='Excellent';  
    elseif v_grade='B' then set v_appraisal='Very Good';  
    elseif v_grade = 'C' then set v_appraisal='Good';  
    else set v_appraisal='No such appraisal';  
  END IF;  
  select concat('Grade ',v_grade,' Appraisal  
    ',v_appraisal) as hasil;  
END$$  
DELIMITER ;  
call spContohCase('C');
```

# Struktur Seleksi dengan IF dan CASE (cont)

## Syntax untuk CASE Statement

```
CASE case_value
```

```
    WHEN when_value THEN statement_list
```

```
    [WHEN when_value THEN statement_list] ...
```

```
    [ELSE statement_list]
```

```
END CASE
```

ATAU

```
CASE
```

```
    WHEN search_condition THEN statement_list
```

```
    [WHEN search_condition THEN statement_list] ...
```

```
    [ELSE statement_list]
```

```
END CASE
```



# Struktur Seleksi dengan IF dan CASE (cont)

BASIS DATA

Case statement digunakan untuk kondisi yang kompleks. Jika sebuah *search\_condition* dievaluasi menghasilkan nilai true, maka SQL statement yang sesuai akan dijalankan. Jika tidak ada kondisi yang sesuai, statement dalam klausa ELSE akan dijalankan. Setiap *statement\_list* dapat berisi satu atau lebih statement.

UNIVERSITAS SANATA DHARMA  
YOGYAKARTA

# Struktur Seleksi dengan IF dan CASE (cont)

## Contoh:

```
DELIMITER $$
DROP PROCEDURE IF EXISTS `pbd`.`spContohCase2` $$
CREATE PROCEDURE `pbd`.`spContohCase2` (v_grade char(1))
BEGIN
declare v_appraisal varchar(20);
    CASE v_grade
        when 'A' THEN set v_appraisal='Excellent';
        when 'B' then set v_appraisal='Very Good';
        when 'C' then set v_appraisal='Good';
    else set v_appraisal='No such appraisal';
END CASE;
select concat('Grade ',v_grade,' Appraisal
',v_appraisal) as hasil;
END $$
DELIMITER ;
call spContohCase2('A');
```

# Struktur Seleksi dengan IF dan CASE (cont)

atau :

```
DELIMITER $$
DROP PROCEDURE IF EXISTS `pbd`.`spContohCase3` $$
CREATE PROCEDURE `pbd`.`spContohCase3` (in v_grade
    char(1))
BEGIN
declare v_appraisal varchar(20);
CASE
    when v_grade = 'A' THEN set v_appraisal='Excellent';
    when v_grade = 'B' THEN set v_appraisal='Very Good';
    when v_grade = 'C' THEN set v_appraisal='Good';
    else set v_appraisal='No such appraisal';
END CASE;
select concat('Grade ',v_grade,' Appraisal ',v_appraisal)
    as hasil;
END $$
DELIMITER ;
call spContohCase3('F')
```

# Struktur Iterasi

## a. LOOP Statement

### Syntax untuk LOOP Statement

```
[begin_label:] LOOP  
    statement_list  
END LOOP [end_label]
```

LOOP mengimplementasikan sebuah bentuk loop sederhana, dapat mengulang eksekusi dari statement list, yang dapat terdiri dari satu atau beberapa statement. Statement dalam loop akan diulang sampai loop selesai. Untuk menghindari LOOP yang terus berulang tanpa berhenti gunakan statement LEAVE.

# Struktur Iterasi (cont)

## Syntak untuk LEAVE statement

**LEAVE** label

LEAVE statement ini digunakan untuk keluar dari label flow control. Bisa digunakan di antara BEGIN..END atau di dalam flow control.

Contoh:



# Struktur Iterasi (cont)

```
CREATE PROCEDURE `spLihatActor2`()
BEGIN
  DECLARE a smallint(5);
  DECLARE b varchar(45);
  DECLARE cur1 CURSOR FOR SELECT actor_id, last_name from
  actor order by last_name;
  set @x=0;
  OPEN cur1;
  test: LOOP
    FETCH cur1 INTO a, b;
    select concat(a,' ',b) as hasil;
    set @x=@x+1;
    if @x >= 5 then LEAVE test;end if;
  END LOOP ;
  CLOSE cur1;

END$$
DELIMITER ;
call spLihatActor2();
```

# Struktur Iterasi (cont)

## b. **ITERATE** Statement

Syntak untuk **ITERATE** statement

```
ITERATE label
```

**ITERATE** hanya dapat digunakan di dalam **LOOP**, **REPEAT**, and **WHILE** statement. **ITERATE** berarti 'DO THE LOOP AGAIN'

# Struktur Iterasi (cont)

Contoh :

```
CREATE PROCEDURE doiterate(p1 INT)
BEGIN
  labell1: LOOP
    SET p1 = p1 + 1;
    IF p1 < 10 THEN ITERATE labell1; END IF;
    LEAVE labell1;
  END LOOP labell1;
  SET @x = p1;
END
```

Berapakah nilai @x ????

# Struktur Iterasi (cont)

## Contoh Lain :

```
DELIMITER $$
DROP PROCEDURE IF EXISTS `pbd`.`spLihatActor3`$$
CREATE PROCEDURE `spLihatActor3`()
BEGIN
    DECLARE a smallint(5);
    DECLARE b varchar(45);
    DECLARE cur1 CURSOR FOR SELECT actor_id, last_name
    from actor order by last_name;
```

# Struktur Iterasi (cont)

```
OPEN curl;  
  set @x = 0;  
  test: LOOP  
    FETCH curl INTO a, b;  
    select concat(a,' ',b) as hasil;  
    set @x= @x+1;  
    IF @x < 5 THEN ITERATE test; END IF;  
    LEAVE test;  
  END LOOP ;  
  CLOSE curl;  
  
END$$  
DELIMITER ;  
call spLihatActor3();
```



# Struktur Iterasi (cont)

## c. REPEAT Statement

Syntak untuk REPEAT statement

```
[begin_label:] REPEAT  
    statement_list  
UNTIL search_condition  
END REPEAT [end_label]
```

Statement\_list dalam REPEAT akan diulang sampai dengan *search\_condition* benar. Jadi, REPEAT selalu masuk ke dalam loop minimal satu kali. *statement\_list* berisi satu atau lebih statement.

# Struktur Iterasi (cont)

Contoh :

```
DELIMITER $$
DROP PROCEDURE IF EXISTS `pbd`.`spREPEAT`$
CREATE PROCEDURE `pbd`.`spREPEAT` (in p1 int)
BEGIN
    SET @x = 0;
    REPEAT SET @x = @x + 1;
    UNTIL @x > p1
END REPEAT;
END$$ DELIMITER ;
call spREPEAT(1000);
select @x;
```

Berapakah nilai @x;

# Struktur Iterasi (cont)

## Contoh Lain :

```
DELIMITER $$
DROP PROCEDURE IF EXISTS `pbd`.`spLihatActor4`$$
CREATE PROCEDURE `spLihatActor4`()
BEGIN
    DECLARE a smallint(5);
    DECLARE b varchar(45);
    DECLARE cur1 CURSOR FOR SELECT actor_id, last_name from
    actor order by last_name;
```

# Struktur Iterasi (cont)

```
OPEN curl1;
  set @x = 0;
  REPEAT
    FETCH curl1 INTO a, b;
    select concat(a,' ',b) as hasil;
    set @x= @x+1;
  UNTIL @x>5
  END REPEAT;
CLOSE curl1;

END$$
DELIMITER ;
call spLihatActor4();
```

**BERAPAKAH jumlah hasil yang diperoleh ???**

# Struktur Iterasi (cont)

## d. WHILE Statement

Syntak untuk WHILE statement

```
[begin_label:] WHILE  
    search_condition  
DO statement_list  
END WHILE [end_label]
```

Statement list dalam WHILE statement akan diulang sampai *search\_condition* is true. *Statement\_list* dapat berisi satu atau lebih statement.



# Struktur Iterasi (cont)

Contoh :

```
DELIMITER $$  
DROP PROCEDURE IF EXISTS `pbd`.`spWHILE` $$  
CREATE PROCEDURE `pbd`.`spWHILE` (  
BEGIN  
    DECLARE v1 INT DEFAULT 5;  
    WHILE v1 > 0 DO  
        select v1;  
        SET v1 = v1 - 1;  
    END WHILE;  
END $$  
DELIMITER ;  
call spWHILE();
```

Berapakah nilai v1???

# Struktur Iterasi (cont)

## Contoh lain :

```
DELIMITER $$
DROP PROCEDURE IF EXISTS `pbd`.`spLihatActor5`$$
CREATE PROCEDURE `spLihatActor5`()
BEGIN
    DECLARE a smallint(5);
    DECLARE b varchar(45);
    DECLARE x int default 5;
    DECLARE cur1 CURSOR FOR SELECT actor_id, last_name
        from actor order by last_name;
```

## Struktur Iterasi (cont)

```
OPEN curl1;  
  WHILE x > 0  
  DO  
    FETCH curl1 INTO a, b;  
    select concat(a,' ',b) as hasil;  
    set x= x-1;  
  END WHILE;  
CLOSE curl1;
```

```
END$$
```

```
DELIMITER ;
```

```
call spLihatActor5();
```