



JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
TAHUN AJARAN 2020/2021

MINGGU KE – 2

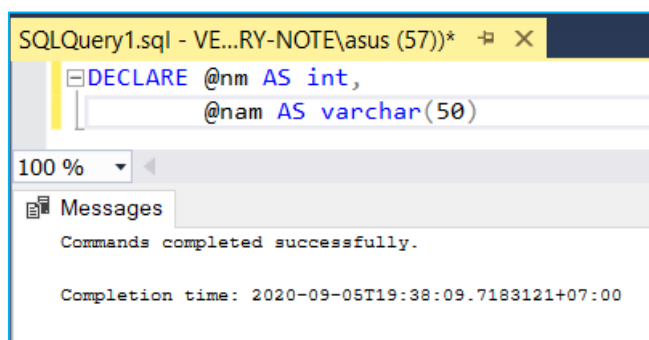
BASIS DATA LANJUT

Pokok Bahasan: T-SQL

A. Implementasi *Variable*

Sama seperti bahasa pemrograman lainnya, bahasa T-SQL juga memerlukan variable dalam scriptnya, bahkan dalam sebuah statement SELECT juga dapat diberi variable. Bila pada bahasa pemrograman lainnya variabel dideklarasikan terlebih dahulu, pada T-SQL juga demikian. Deklarasi variabel untuk T-SQL menggunakan kata kunci DECLARE. Bentuknya adalah sebagai berikut :

```
DECLARE nm_var tipe_var;
```



Tipe data variabel identik dengan tipe data field, hal yang perlu diingat adalah setiap variabel pada T-SQL didahului oleh @. Seperti halnya pada bahasa pemrograman umumnya, tiap variabel pasti memiliki ruang lingkup tersendiri. Pada ruang lingkupnya variabel dapat diakses dan dimanipulasi. Aturan pada T-SQL adalah variabel yang dideklarasikan pada sebuah batch hanya berlaku untuk batch tersebut, tidak dapat diakses oleh batch lainnya meskipun pada satu file.sql.

B. Flow elements

T-SQL memiliki statement-statement untuk mengatur jalannya alur program. Statement-statement tersebut adalah sebagai berikut :

1. IF

Ini adalah statement pengendali alur program yang paling umum dan sangat sering digunakan terutama pada bahasa-bahasa pemrograman lainnya. Bentuk IF yang paling sederhana adalah sebagai berikut:

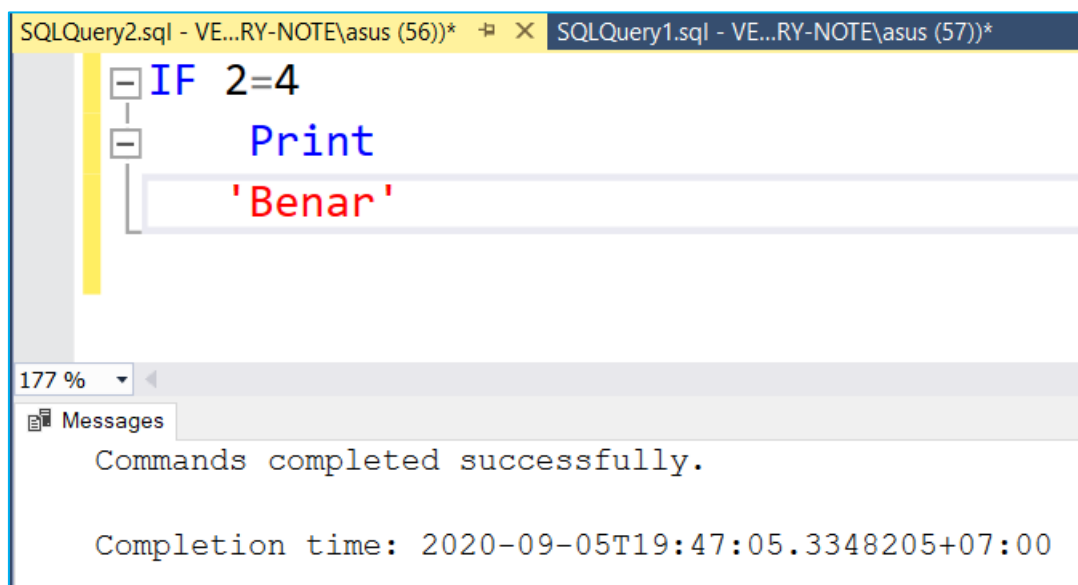
```
IF kondisi  
    statement
```

Pada contoh script dibawah ini, jika kondisi yang ada setelah IF bernilai False maka statement yang ada dibawahnya akan dihindari atau tidak dilaksanakan.

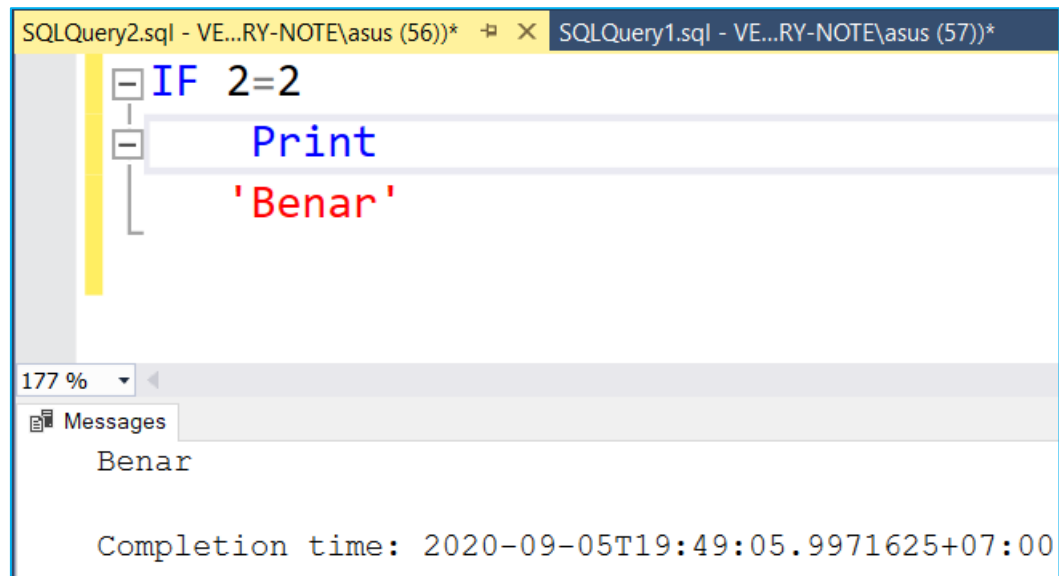
```
IF 2=4  
    Print  
    'Benar'
```

Hasilnya yang dicetak adalah Salah, karena 2 tidak sama dengan 4.

Contoh Hasil :



Kita akan membenarkan kondisinya agar statement benar ditampilkan, berikut adalah kondisi yang benar :

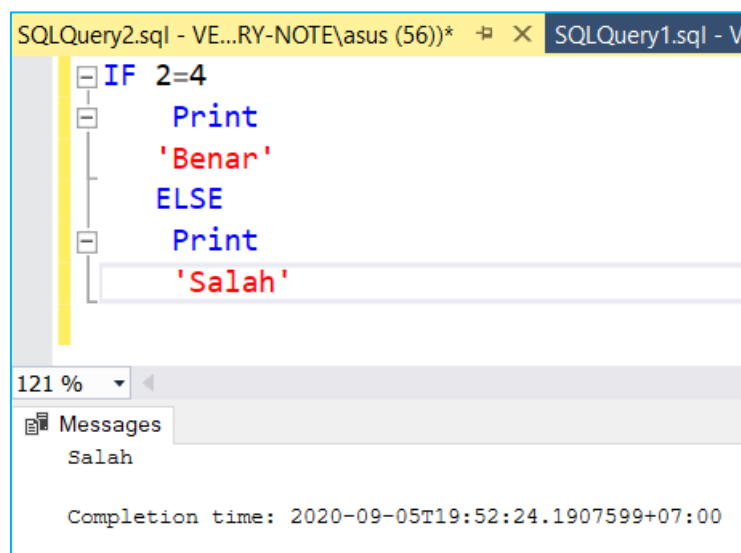


The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'SQLQuery2.sql - VE...RY-NOTE\asus (56))*' and 'SQLQuery1.sql - VE...RY-NOTE\asus (57))*'. The active tab displays the following SQL code:

```
IF 2=2
    Print
    'Benar'
```

Below the code editor, the 'Messages' pane shows the output 'Benar'. The completion time is '2020-09-05T19:49:05.9971625+07:00'.

2. IF ELSE

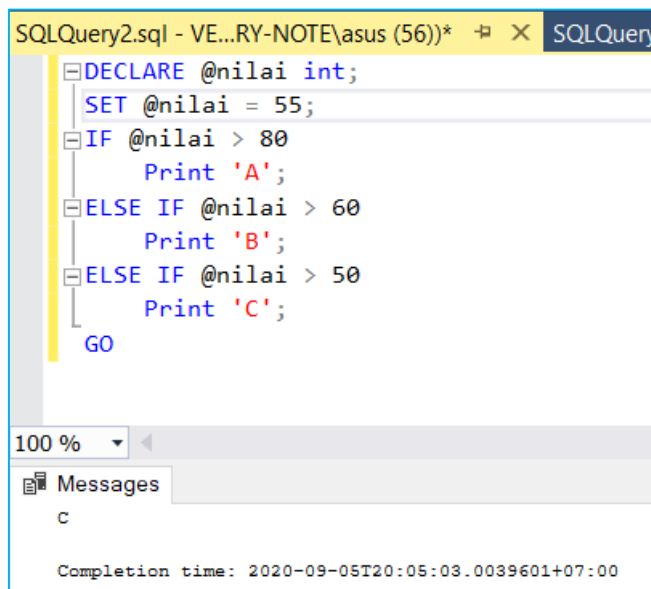


The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'SQLQuery2.sql - VE...RY-NOTE\asus (56))*' and 'SQLQuery1.sql - V'. The active tab displays the following SQL code:

```
IF 2=4
    Print
    'Benar'
ELSE
    Print
    'Salah'
```

Below the code editor, the 'Messages' pane shows the output 'Salah'. The completion time is '2020-09-05T19:52:24.1907599+07:00'.

3. Nested IF



```
SQLQuery2.sql - VE...RY-NOTE\asus (56))* X SQLQuery
DECLARE @nilai int;
SET @nilai = 55;
IF @nilai > 80
    Print 'A';
ELSE IF @nilai > 60
    Print 'B';
ELSE IF @nilai > 50
    Print 'C';
GO

100 %
Messages
c

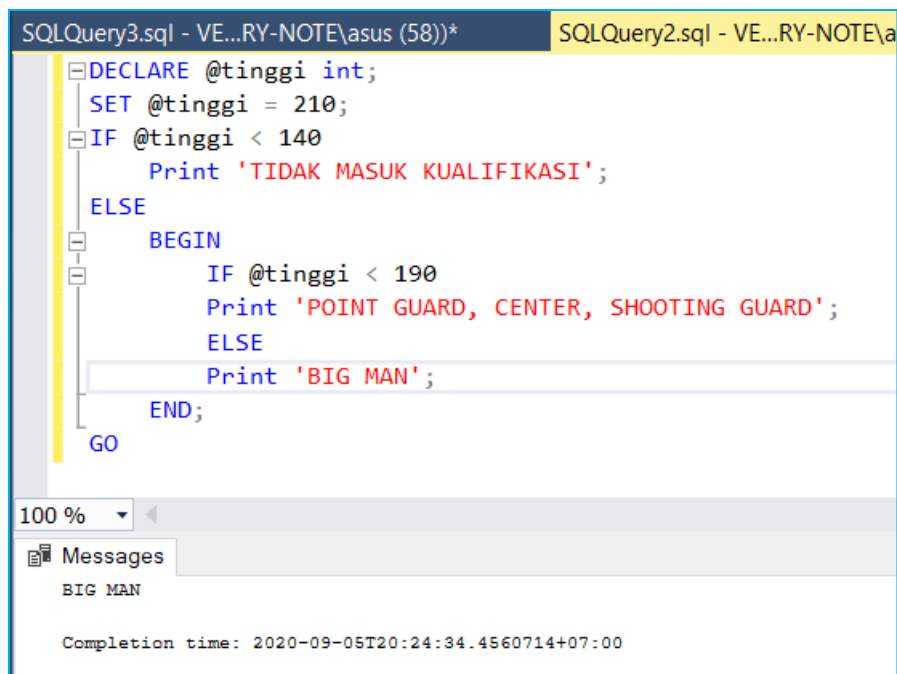
Completion time: 2020-09-05T20:05:03.0039601+07:00
```

4. IF Exists()

Salah satu bentuk If adalah if yang disertai dengan kata kunci exists (). Struktur If exists ini menggunakan keberadaan field-field yang dikembalikan dari statement SELECT. Statement SELECT harus menyertakan semua fieldnya.

5. Begin...End

Kadangkala statement yang akan dikerjakan apabila kondisi pada if mengembalikan nilai true lebih dari satu. Jika statement yang menyertai IF lebih dari satu maka digunakan kata kunci Begin untuk menandai statement End untuk menandai akhir statement. Begin dan End diperlukan supaya SQL Server mengetahui bahwa statement-statement yang ada diataranya adalah satu kumpulan. Bentuk penulisannya adalah sebagai berikut.



```
SQLQuery3.sql - VE...RY-NOTE\asus (58))*  SQLQuery2.sql - VE...RY-NOTE\asus (58)*
-- DECLARE @tinggi int;
-- SET @tinggi = 210;
-- IF @tinggi < 140
--     Print 'TIDAK MASUK KUALIFIKASI';
-- ELSE
--     BEGIN
--         IF @tinggi < 190
--             Print 'POINT GUARD, CENTER, SHOOTING GUARD';
--         ELSE
--             Print 'BIG MAN';
--     END;
GO
```

100 %

Messages

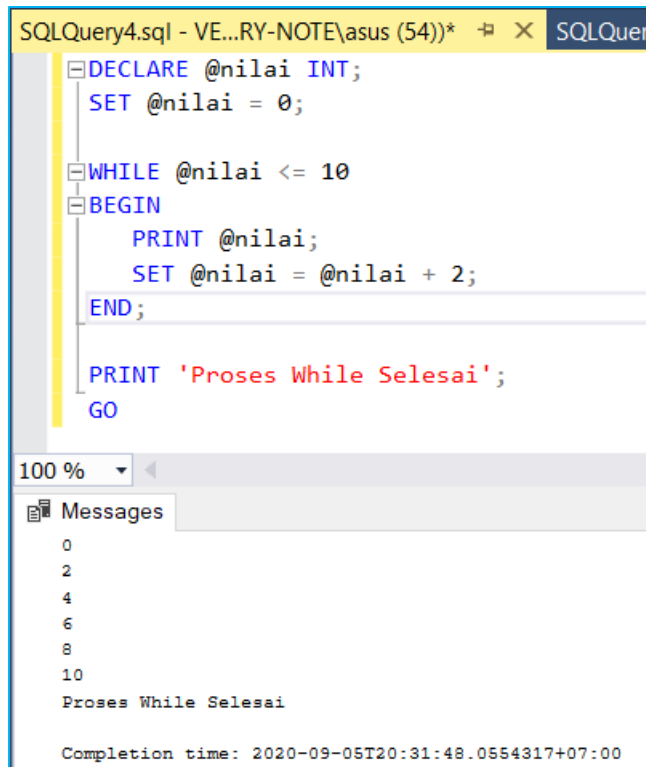
BIG MAN

Completion time: 2020-09-05T20:24:34.4560714+07:00

6. While

Perintah While digunakan untuk melakukan perulangan terhadap sebuah statement atau sekumpulan kode selama kondisi yang disyaratkan terpenuhi atau masih menghasilkan nilai true. Karena biasanya yang diulang adalah sekumpulan kode SQL maka kata kunci begin dan end diperlukan. Urutan pengerjaan statement while adalah sebagai berikut :

- Perintah While akan melakukan tes pada kondisi yang diberikan oleh user. Jika kondisi tersebut bernilai benar maka perintah While akan menjalankan blok kode SQL yang ada dibawahnya. Jika tes kondisi memberikan hasil false maka blok kode SQL yang ada dibawahnya tidak akan dikerjakan dan perulangan berhenti.
- Setelah mengerjakan perintah pada blok kode SQL maka alur program kembali ke bagian tes kondisi, hal ini terus-menerus dilakukan hingga kondisi memberikan nilai false.



```
SQLQuery4.sql - VE...RY-NOTE\asus (54))* X SQLQuer
DECLARE @nilai INT;
SET @nilai = 0;

WHILE @nilai <= 10
BEGIN
    PRINT @nilai;
    SET @nilai = @nilai + 2;
END;

PRINT 'Proses While Selesai';
GO
```

100 %

Messages

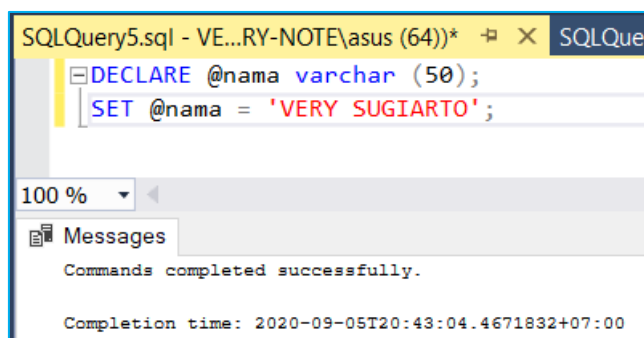
0
2
4
6
8
10
Proses While Selesai

Completion time: 2020-09-05T20:31:48.0554317+07:00

C. SELECT , SET, IF, Print dan Variable

1. SET

Statement SET digunakan untuk memberikan nilai pada sebuah variable yang sebelumnya sudah kita deklarasikan. Berikut adalah contoh penggunaan statement SET :



```
SQLQuery5.sql - VE...RY-NOTE\asus (64))* X SQLQue
DECLARE @nama varchar (50);
SET @nama = 'VERY SUGIARTO';
```

100 %

Messages

Commands completed successfully.

Completion time: 2020-09-05T20:43:04.4671832+07:00

2. SELECT

Statement *SELECT* digunakan untuk mengambil sebuah data. Berikut adalah contoh dari statement *SELECT* :

```

SQLQuery5.sql - VE...RY-NOTE\asus (64))* X S
-- DECLARE @nama varchar (50);
-- SET @nama = 'VERY SUGIARTO';
-- SELECT @nama;

100 %
Results Messages
(No column name)
1 VERY SUGIARTO

```

3. Begin...End

Berikut ini adalah contoh dari penggunaan Begin .. End :

```

SQLQuery5.sql - VE...RY-NOTE\asus (64))* X SQLQuery4
-- DECLARE @id int,
--          @nama varchar (50),
--          @alamat varchar (max);
-- SET @id = 1;
-- IF @id = 1
-- BEGIN
--     SET @nama = 'VERY SUGIARTO';
--     SET @alamat = 'MALANG';
-- END;
-- SELECT @id,@nama,@alamat;

100 %
Results Messages
(No column name) (No column name) (No column name)
1 1 VERY SUGIARTO MALANG

```

D. T-SQL pada Table

Berikut adalah implementasi T-SQL pada sebuah table .

```

SQLQuery6.sql - VE...RY-NOTE\asus (68))* X VERY-NOTE.polinema - dbo.r
-- IF EXISTS (SELECT * FROM mahasiswa)
-- BEGIN
--     SELECT * FROM mahasiswa WHERE nama LIKE 'VERY%';
-- END
-- ELSE
--     PRINT 'Nama VERY tidak ditemukan';

100 %
Results Messages
nim nama alamat
1 1 VERY MALANG
2 3 VERY SUGIARTO JAKARTA

```