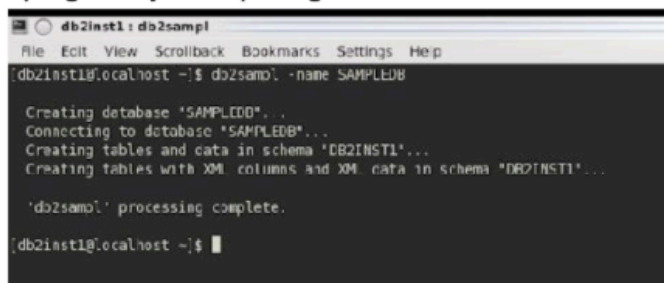


Nama : Nick Wilsan
NIM : 245150400111044
Kelas : D
Bab : 4
Asisten : Dimas Akmal Andrian

LANGKAH 1

A. Soal

1. Buat database sample dengan nama SAMPLEDB, jalankan perintah : “db2sampl -name SAMPLEDB” seperti yang ditunjukkan pada gambar 4.1



```
db2inst1: db2sampl
File Edit View Scrollback Bookmarks Settings Help
db2inst1@localhost ~]$ db2sampl -name SAMPLEDB

Creating database 'SAMPLEDB'...
Connecting to database 'SAMPLEDB'...
Creating tables and data in schema 'DB2INST1'...
Creating tables with XML columns and XML data in schema 'DB2INST1'...

'db2sampl' processing complete.
db2inst1@localhost ~]$
```

Gambar 4.1 Perintah membuat *database sampledb*

B. Screenshoot

```
$ db2sampl -name SAMPLEDB

Creating database "SAMPLEDB"...
Connecting to database "SAMPLEDB"...
Creating tables and data in schema "DB2INST1"...
Creating tables with XML columns and XML data in schema "DB2INST1"...
Creating tables with VECTOR columns and VECTOR data in schema "DB2INST1"...

'db2sampl' processing complete.
```

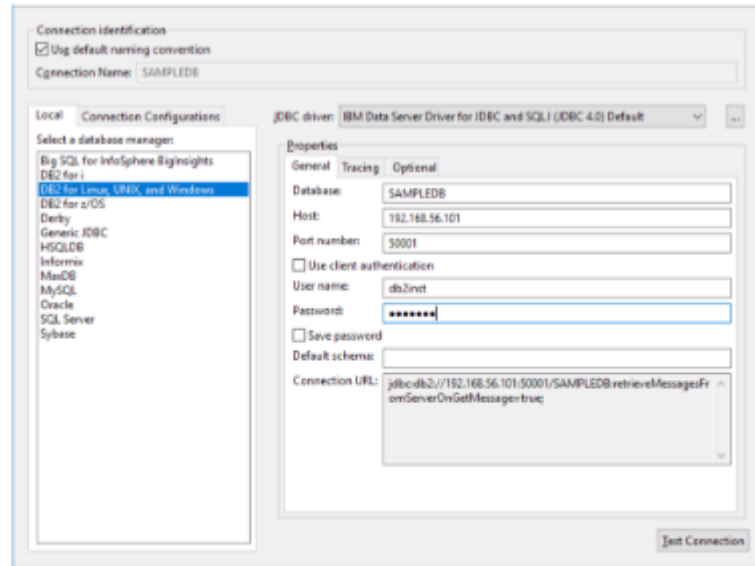
C. Penjelasan (Kesimpulan, Kendala, dll.)

Pada langkah ini kita membuat database bernama SAMPLEDB

LANGKAH 2

A. Soal

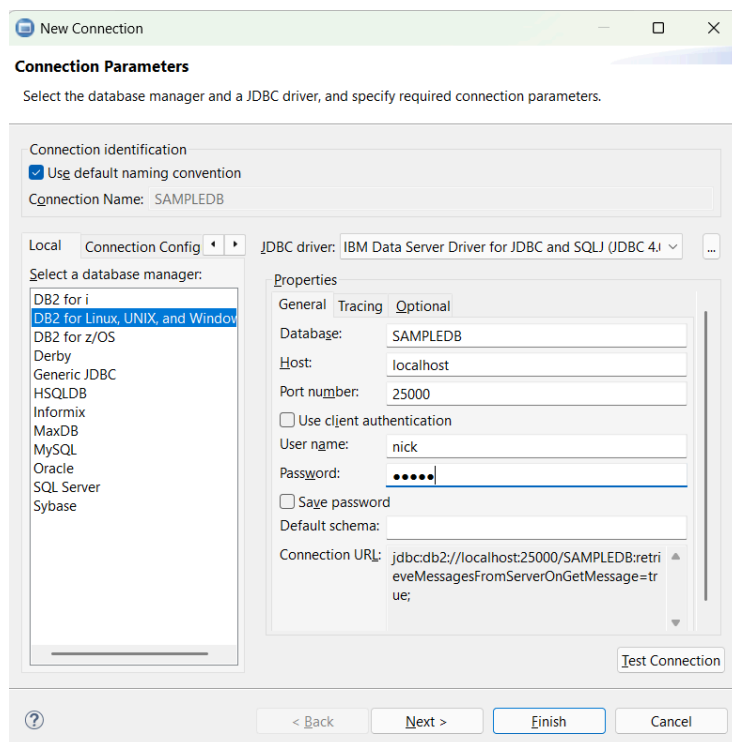
Buka Data Studio dan akses database SAMPLEDB, seperti yang terlihat pada gambar 4.2



Gambar 4.2 Akses database sampledb

Setelah berhasil mengakses database sampledb selanjutnya akan kita buat VIEW, MATERIALIZED QUERY TABLE, HISTORY TABLE dan TEMPORARY TABLE.

B. Screenshoot



C. Penjelasan (Kesimpulan, Kendala, dll.)

Pada langkah ini kita mengkoneksikan database SAMPLEDB yang sudah kita buat tadi ke Data Studio

LANGKAH 3

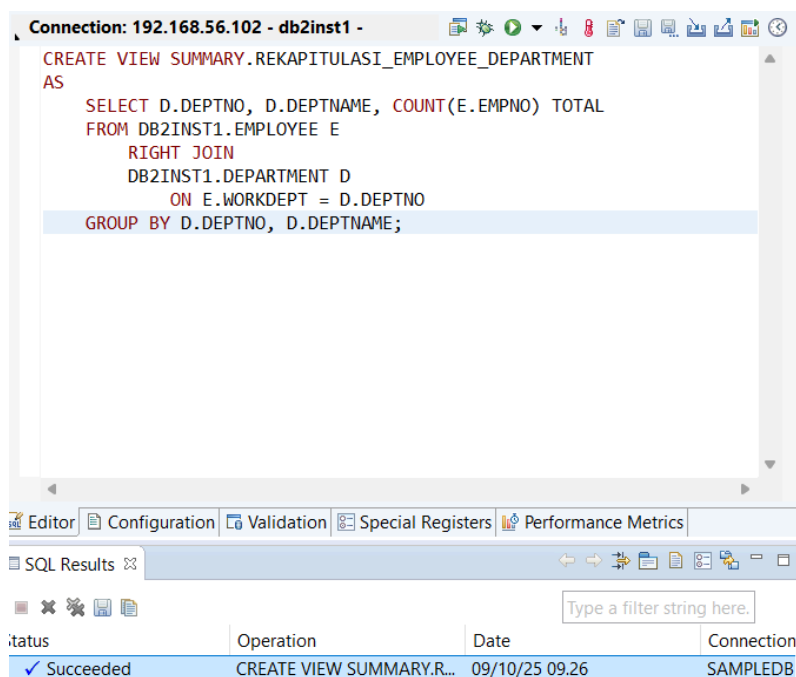
A. Soal

3. Buat View dapat menggunakan formulasi berikut :

```
CREATE VIEW NAMA_VIEW  
AS  
--DML
```

Implementasikan contoh view pada gambar 4.3. Pada gambar 4.3 menunjukkan sebuah view yang bertujuan untuk menyajikan data DEPTNO, DEPTNAME, TOTAL_EMPLOYEE. Untuk SCHEMA yang ada pada DML di VIEW disesuaikan dengan SCHEMA masing-masing.

B. Screenshoot



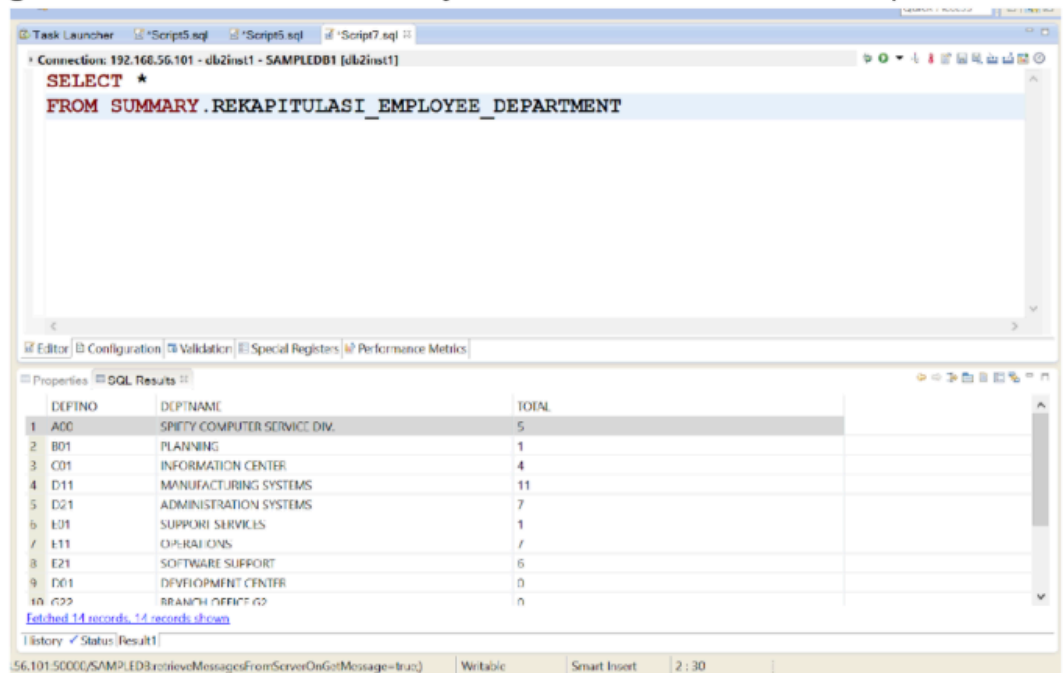
C. Penjelasan (Kesimpulan, Kendala, dll.)

Membuat view yang bertujuan untuk menyajikan data DEPTNO, DEPTNAME, TOTAL_EMPLOYEE.

LANGKAH 4

A. Soal

4. Selanjutnya *view* yang sudah dibuat pada nomor 3 dapat diakses seperti halnya kita mengakses *table*. Gambar 4.4 Menunjukkan cara untuk membaca data pada *VIEW*



Gambar 4.4 Cara mengakses data pada View

B. Screenshoot

```
SELECT *
FROM SUMMARY.REKAPITULASI_EMPLOYEE_DEPARTMENT;
```

DEPTNO	DEPTNAME	TOTAL
A00	SPIFFY COMPUTER SERVICE DIV.	5
B01	PLANNING	1
C01	INFORMATION CENTER	4
D11	MANUFACTURING SYSTEMS	11
D21	ADMINISTRATION SYSTEMS	7
E01	SUPPORT SERVICES	1
E11	OPERATIONS	7
E21	SOFTWARE SUPPORT	6
D01	DEVELOPMENT CENTER	0
G22	BRANCH OFFICE G2	0
H22	BRANCH OFFICE H2	0
F22	BRANCH OFFICE F2	0
I22	BRANCH OFFICE I2	0
J22	BRANCH OFFICE J2	0

C. Penjelasan (Kesimpulan, Kendala, dll.)

Menampilkan view yang telah dibuat

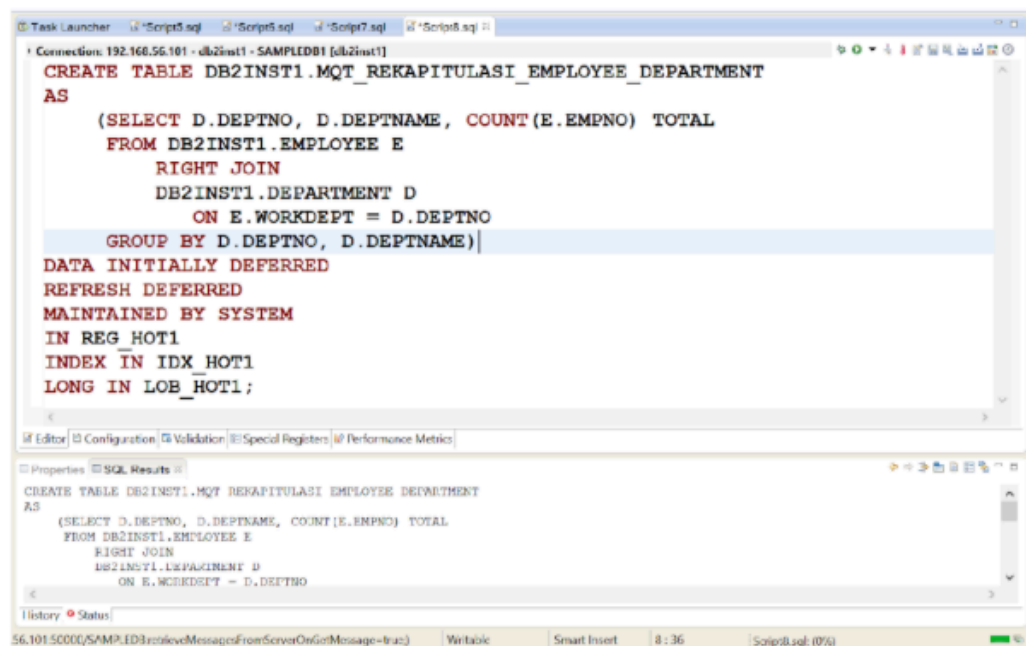
LANGKAH 5

A. Soal

5. Contoh untuk membuat MQTs dapat dilihat pada gambar 4.5. MQTs pada gambar 4.5 disusun dengan menggunakan DML yang sama seperti View pada gambar 4.3. Untuk membuat MQTs dapat mengikut sintaks berikut :

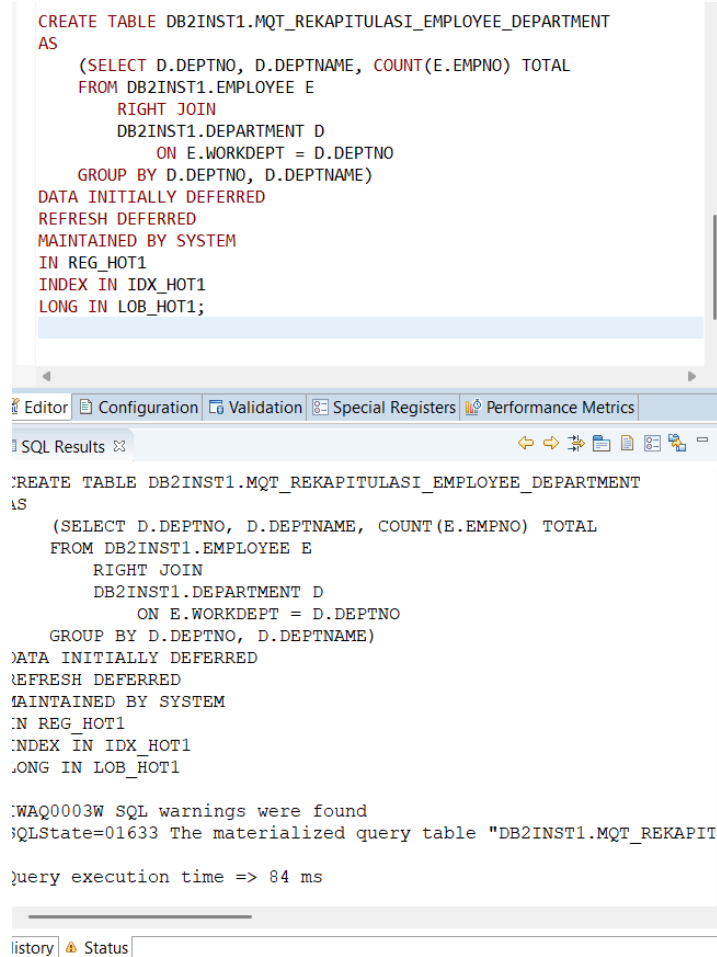
```
CREATE TABLE NAMA_SKEMA.NAMA_MQT
AS
(
    --DML
)
DATA INITIALLY DEFERRED
REFRESH DEFERRED
MAINTAINED BY SYSTEM
IN NAMA_REGULAR_TABLESPACE
INDEX IN NAMA_INDEX_TABLESPACE
LONG IN NAMA_LONG_TABLESPACE;
```

Implementasikan MQTs pada gambar 4.5



Gambar 4.5 Sintak untuk Membuat MQTs

B. Screenshoot



```
CREATE TABLE DB2INST1.MQT_REKAPITULASI_EMPLOYEE_DEPARTMENT
AS
  (SELECT D.DEPTNO, D.DEPTNAME, COUNT(E.EMPNO) TOTAL
   FROM DB2INST1.EMPLOYEE E
   RIGHT JOIN
     DB2INST1.DEPARTMENT D
     ON E.WORKDEPT = D.DEPTNO
   GROUP BY D.DEPTNO, D.DEPTNAME)
DATA INITIALLY DEFERRED
REFRESH DEFERRED
MAINTAINED BY SYSTEM
IN REG_HOT1
INDEX IN IDX_HOT1
LONG IN LOB_HOT1;
```

SQL Results

```
CREATE TABLE DB2INST1.MQT_REKAPITULASI_EMPLOYEE_DEPARTMENT
AS
  (SELECT D.DEPTNO, D.DEPTNAME, COUNT(E.EMPNO) TOTAL
   FROM DB2INST1.EMPLOYEE E
   RIGHT JOIN
     DB2INST1.DEPARTMENT D
     ON E.WORKDEPT = D.DEPTNO
   GROUP BY D.DEPTNO, D.DEPTNAME)
DATA INITIALLY DEFERRED
REFRESH DEFERRED
MAINTAINED BY SYSTEM
IN REG_HOT1
INDEX IN IDX_HOT1
LONG IN LOB_HOT1

WAQ0003W SQL warnings were found
SQLState=01633 The materialized query table "DB2INST1.MQT_REKAPIT
query execution time => 84 ms
```

History Status

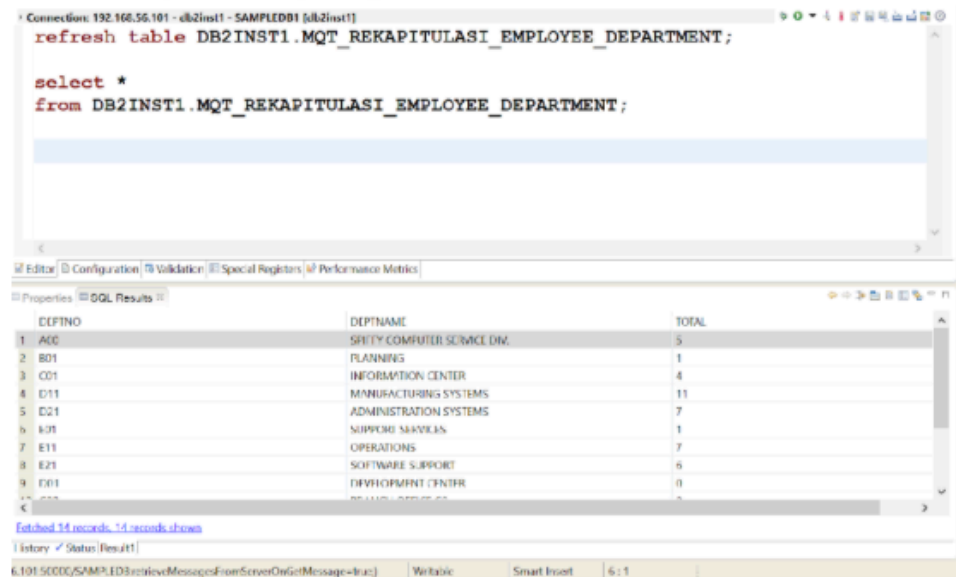
C. Penjelasan (Kesimpulan, Kendala, dll.)

Membuat tabel MQT, mirip dengan view tapi MQT ini dapat meningkatkan kinerja query sehingga lebih efisien

LANGKAH 6

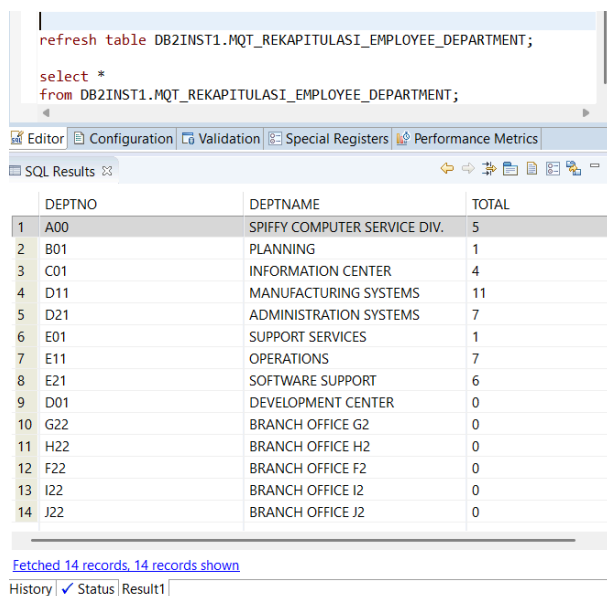
A. Soal

6. MQTs setelah dibuat tidak langsung dapat diakses, beda dengan View. Untuk bisa mengakses MQTs setelah dibuat, lakukan proses REFRESH TABLE. Untuk melakukan REFRESH TABLE dapat dilihat pada gambar 4.6.



Gambar 4.6 Refresh Tabel pada MQTs

B. Screenshoot



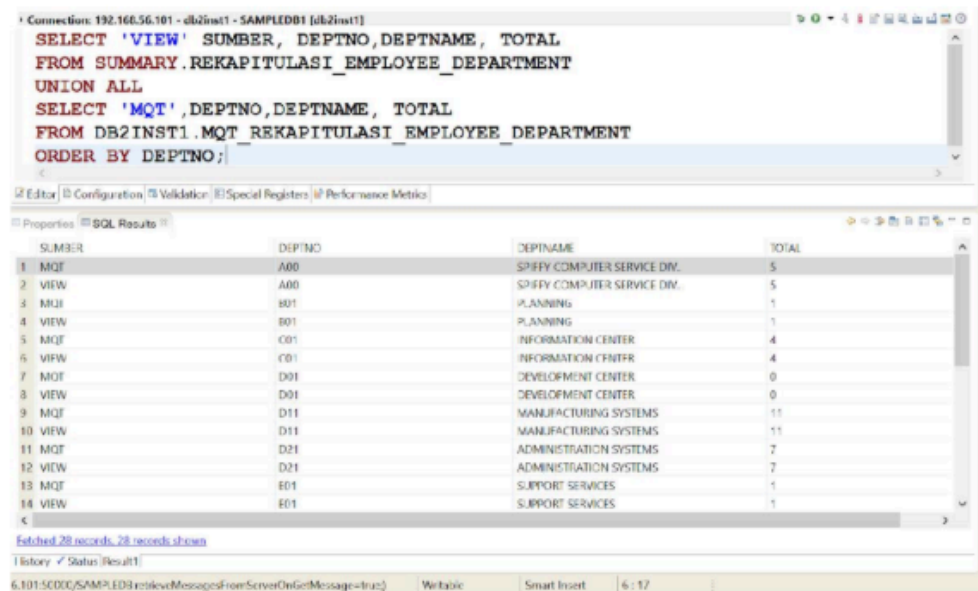
C. Penjelasan (Kesimpulan, Kendala, dll.)

Agar tabel MQT bisa diakses, lakukan refresh tabel terlebih dahulu dan tampilkan tabel tersebut agar dapat melihat perubahannya

LANGKAH 7

A. Soal

7. Hal lain yang membedakan VIEW dan MQTs adalah VIEW akan selalu mengikuti perubahan data dari tabel asalnya sementara MQTs tidak, namun MQTs dapat mengikuti perubahan data sesuai dengan tabel asalnya jika dilakukan REFRESH TABLE. Jalankan DML seperti pada gambar 4.7 perhatikan kondisi datanya, data dari VIEW dan MQT masih sama hal tersebut dikarenakan belum ada perubahan pada tabel asalnya.

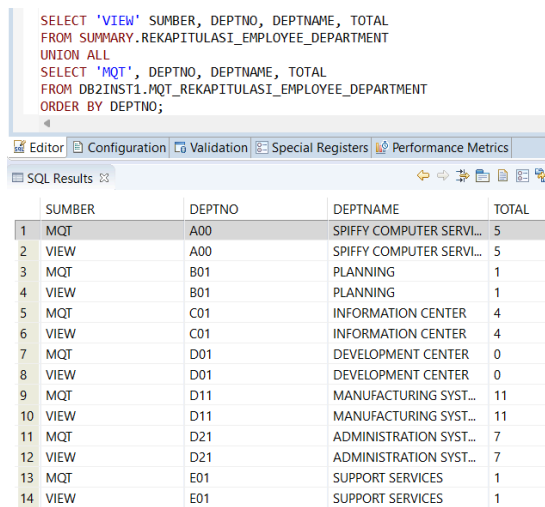


```
SELECT 'VIEW' SUMBER, DEPTNO, DEPTNAME, TOTAL
FROM SUMMARY.REKAPITULASI_EMPLOYEE_DEPARTMENT
UNION ALL
SELECT 'MQT', DEPTNO, DEPTNAME, TOTAL
FROM DB2INST1.MQT_REKAPITULASI_EMPLOYEE_DEPARTMENT
ORDER BY DEPTNO;
```

	SUMBER	DEPTNO	DEPTNAME	TOTAL
1	MQT	A00	SPIFFY COMPUTER SERVICE DIV.	5
2	VIEW	A00	SPIFFY COMPUTER SERVICE DIV.	5
3	MQT	B01	PLANNING	1
4	VIEW	B01	PLANNING	1
5	MQT	C01	INFORMATION CENTER	4
6	VIEW	C01	INFORMATION CENTER	4
7	MQT	D01	DEVELOPMENT CENTER	0
8	VIEW	D01	DEVELOPMENT CENTER	0
9	MQT	D11	MANUFACTURING SYSTEMS	11
10	VIEW	D11	MANUFACTURING SYSTEMS	11
11	MQT	D21	ADMINISTRATION SYSTEMS	7
12	VIEW	D21	ADMINISTRATION SYSTEMS	7
13	MQT	E01	SUPPORT SERVICES	1
14	VIEW	E01	SUPPORT SERVICES	1

Gambar 4.7 Perbandingan data MQTs dan View

B. Screenshoot



```
SELECT 'VIEW' SUMBER, DEPTNO, DEPTNAME, TOTAL
FROM SUMMARY.REKAPITULASI_EMPLOYEE_DEPARTMENT
UNION ALL
SELECT 'MQT', DEPTNO, DEPTNAME, TOTAL
FROM DB2INST1.MQT_REKAPITULASI_EMPLOYEE_DEPARTMENT
ORDER BY DEPTNO;
```

	SUMBER	DEPTNO	DEPTNAME	TOTAL
1	MQT	A00	SPIFFY COMPUTER SERVI...	5
2	VIEW	A00	SPIFFY COMPUTER SERVI...	5
3	MQT	B01	PLANNING	1
4	VIEW	B01	PLANNING	1
5	MQT	C01	INFORMATION CENTER	4
6	VIEW	C01	INFORMATION CENTER	4
7	MQT	D01	DEVELOPMENT CENTER	0
8	VIEW	D01	DEVELOPMENT CENTER	0
9	MQT	D11	MANUFACTURING SYST...	11
10	VIEW	D11	MANUFACTURING SYST...	11
11	MQT	D21	ADMINISTRATION SYST...	7
12	VIEW	D21	ADMINISTRATION SYST...	7
13	MQT	E01	SUPPORT SERVICES	1
14	VIEW	E01	SUPPORT SERVICES	1

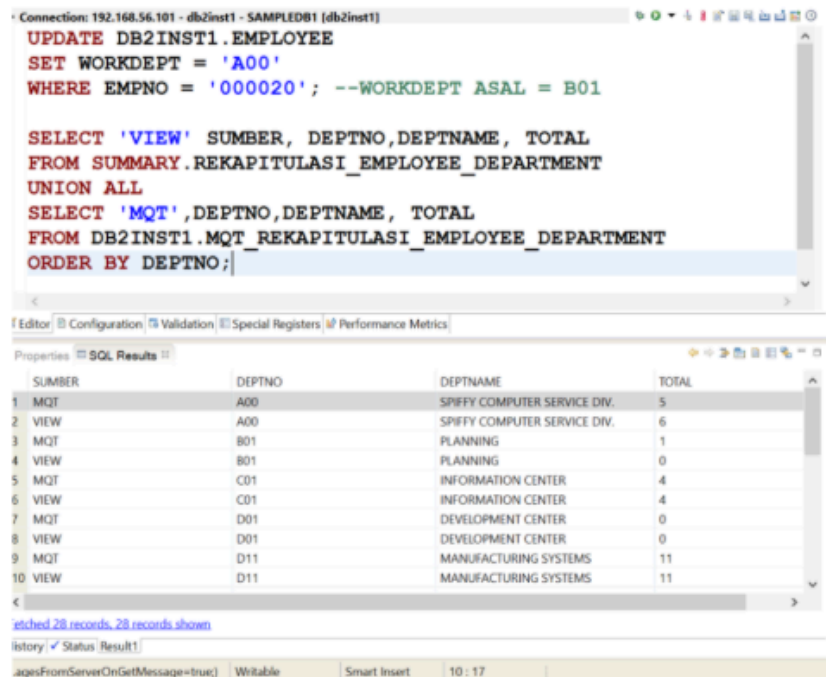
C. Penjelasan (Kesimpulan, Kendala, dll.)

Query tersebut digunakan untuk melihat perbedaan antara view dan MQT

LANGKAH 8

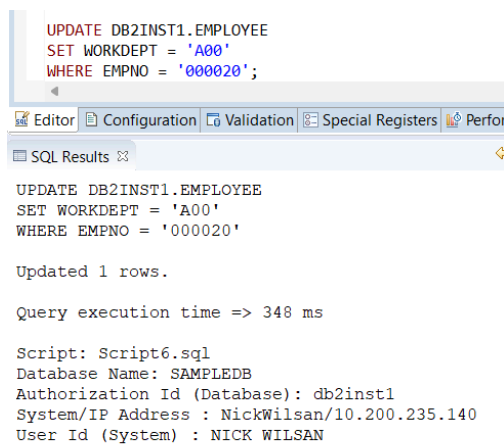
A. Soal

8. Lakukan update pada tabel EMPLOYEE dengan mengganti workdept dari salah satu pegawai. Perhatikan pada DEPTNO 'A00' dan 'B01'



Gambar 4.8 Update tabel Employee

B. Screenshoot



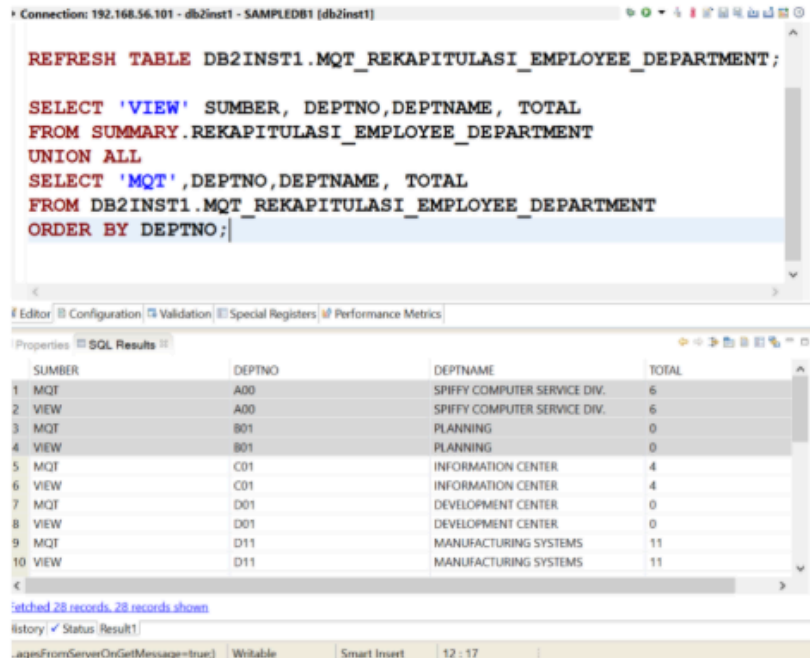
C. Penjelasan (Kesimpulan, Kendala, dll.)

Melakukan update pada tabel EMPLOYEE dengan mengganti workdept dari salah satu Pegawai. Kemudian, tampilkan tabel menggunakan query yang ada sebelumnya untuk melihat perbandingan antara view dan MQT. dapat dilihat, setelah melakukan update, Data pada MQT tidak otomatis mengikut perubahan pada tabel asal

LANGKAH 9

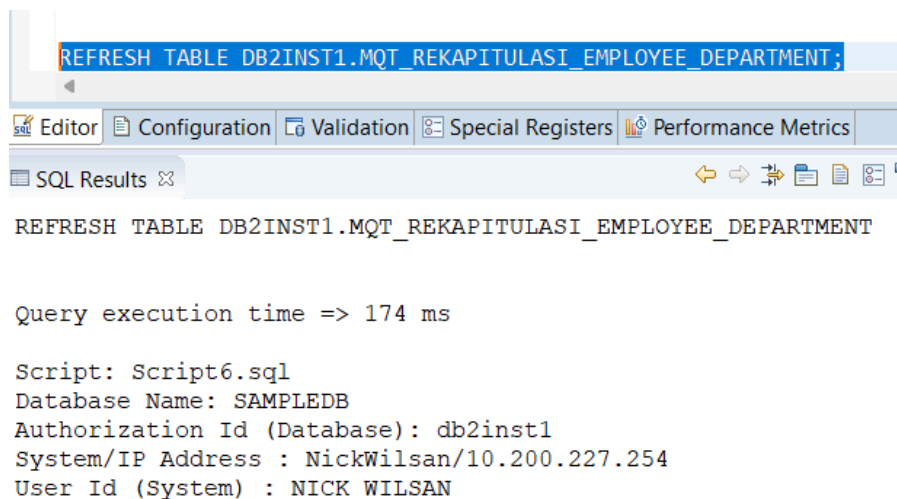
A. Soal

9. Data pada MQT tidak otomatis mengikut perubahan pada tabel asal, supaya bisa menyesuaikan datanya dengan tabel asal maka MQT harus dilakukan REFRESH TABLE kembali.



Gambar 4.9 Refresh Table pada MQTs

B. Screenshoot



C. Penjelasan (Kesimpulan, Kendala, dll.)

Agar bisa menyesuaikan data dengan tabel asal, lakukan refresh pada MQT dan tampilkan kembali hasilnya

LANGKAH 10

A. Soal

10. Sebelum pembuatan Buat *history table*, tambahkan kolom SYSTEM_START_TIME1, SYSTEM_END_TIME1, TRANS_ID1. Kemudian tambahkan system period SYSTEM_TIME seperti pada gambar 4.10 di bawah.

```
--ALTER TABLE YANG AKAN DIPASANG HISTORY DATANYA
--DENGAN MENAMBAHKAN 3 KOLOM
ALTER TABLE DB2INST1.EMPLOYEE
    ADD COLUMN SYSTEM_START_TIME1 TIMESTAMP(12)
        NOT NULL GENERATED ALWAYS AS ROW BEGIN
    ADD COLUMN SYSTEM_END_TIME1 TIMESTAMP(12) NOT NULL
        GENERATED ALWAYS AS ROW END
    ADD COLUMN TRANS_ID1 TIMESTAMP(12)
        GENERATED ALWAYS AS TRANSACTION START ID;

--LAKUKAN PERUBAHAN PADA TABLE EMPLOYEE
--UTK MENJADI SYSTEM-PERIOD TEMPORAL TABLE
ALTER TABLE DB2INST1.EMPLOYEE
    ADD PERIOD SYSTEM_TIME ( SYSTEM_START_TIME1,
                            SYSTEM_END_TIME1 );
```

Gambar 4.10 Query Penambahan Kolom pada Tabel Employee

Setelah berhasil , kemudian buat *history table* seperti pada gambar 4.11

```
--BUAT TABLESPACE UNTUK MENYIMPAN DATA PADA HISTOI
CREATE TABLESPACE USERSPACE1_HIST;

--BUAT TABEL HISTORY DENGAN MENGGUNAKAN
--TABLE SPACE HISTORY
CREATE TABLE DB2INST1_HIST.EMPLOYEE_HIST1
(
    EMPNO CHAR(6 OCTETS) NOT NULL,
    FIRSTNME VARCHAR(12 OCTETS) NOT NULL,
    MIDINIT CHAR(1 OCTETS),
    LASTNAME VARCHAR(15 OCTETS) NOT NULL,
    WORKDEPT CHAR(3 OCTETS),
    PHONENO CHAR(4 OCTETS),
    HIREDATE DATE,
    JOB CHAR(8 OCTETS),
    EDLEVEL SMALLINT NOT NULL,
    SEX CHAR(1 OCTETS),
    BIRTHDATE DATE, SALARY
    DECIMAL(9,2),
    BONUS DECIMAL(9,2),
    COMM DECIMAL(9,2),
    SYSTEM_START_TIME1 TIMESTAMP(12) NOT NULL,
    SYSTEM_END_TIME1 TIMESTAMP(12) NOT NULL,
    TRANS_ID1 TIMESTAMP(12)
) IN USERSPACE1_HIST;

--HUBUNGKAN TABEL UTAMA DENGAN HISTORY TABEL
ALTER TABLE DB2INST1.EMPLOYEE
    ADD VERSIONING USE HISTORY TABLE
    DB2INST1_HIST.EMPLOYEE_HIST1;
```

Gambar 4.11 Query Pembuatan History Tabel

B. Screenshoot

```
CREATE TABLE DB2INST1_HIST.EMPLOYEE_HIST1
(
  EMPNO CHAR(6 OCTETS) NOT NULL,
  FIRSTNAME VARCHAR(12 OCTETS) NOT NULL,
  MIDINIT CHAR(1 OCTETS),
  LASTNAME VARCHAR(15 OCTETS) NOT NULL,
  WORKDEPT CHAR(3 OCTETS),
  PHONENO CHAR(4 OCTETS),
  HIREDATE DATE,
  JOB CHAR(8 OCTETS),
  EDLEVEL SMALLINT NOT NULL,
  SEX CHAR(1 OCTETS),
  BIRTHDATE DATE,
  SALARY DECIMAL(9,2),
  BONUS DECIMAL(9,2),
  COMM DECIMAL(9,2),
  SYSTEM_START_TIME1 TIMESTAMP(12) NOT NULL,
  SYSTEM_END_TIME1 TIMESTAMP(12) NOT NULL,
  TRANS_ID1 TIMESTAMP(12)
) IN USERSPACE1_HIST;

ALTER TABLE DB2INST1.EMPLOYEE
  ADD VERSIONING USE HISTORY TABLE
    DB2INST1_HIST.EMPLOYEE_HIST1;
```

QL Results

```
PER TABLE DB2INST1.EMPLOYEE
  ADD VERSIONING USE HISTORY TABLE
    DB2INST1_HIST.EMPLOYEE_HIST1
```

every execution time => 18 ms

Script: Script6.sql
Database Name: SAMPLEDB
Authorization Id (Database): db2inst1
System/IP Address : NickWilson/10.200.227.10
User Id (System) : NICK WILSON

C. Penjelasan (Kesimpulan, Kendala, dll.)

Sebelum pembuatan history table, tambahkan kolom SYSTEM_START_TIME1, SYSTEM_END_TIME1, TRANS_ID1. Kemudian tambahkan system period SYSTEM_TIME. Lakukan perubahan pada tabel employee agar bisa melihat data karyawan versi sebelumnya. Setelah itu, buat tablespace untuk menyimpan data pada history dan setelah membuat tabel history, hubungkan tabel utama dengan history table

LANGKAH 11

A. Soal

11. Cara kerja history table

Masukan data seperti contoh pada gambar 4.12 dengan mengganti firstname, midinit, lastname dengan nama depan Anda, inisial nama depan Anda, dan nama belakang Anda.

```
INSERT INTO "USER-PC".EMPLOYEE (  
    EMPNO , FIRSTNAME , MIDINIT ,  
    LASTNAME , WORKDEPT , PHONENO ,  
    HIREDATE , JOB , EDLEVEL ,  
    SEX , BIRTHDATE , BONUS , COMM  
)  
VALUES (  
    '000099' , 'Yhouga' , 'Y',  
    'Evantio' , 'A00' , '3978' ,  
    '1995-01-01' , 'PRES' , 18,  
    'M' , '1963-08-24' , 0 , 4220.00)@
```

Gambar 4.12 Sintaks insert pada tabel Employee

Lalu cek pada tabel HIST.EMPLOYEE_HIST1. Data setelah proses insert pada tabel utama tidak akan tercatat dalam tabel history

```
SELECT *  
FROM DB2INST1_HIST.EMPLOYEE_HIST1@
```

The screenshot shows a database management tool interface. On the left, there is a 'SQL Results' tab and a 'Console' tab. The 'SQL Results' tab displays a table with columns: EMPNO, FIRSTNAME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, and HIREDATE. The table is currently empty. The 'Console' tab displays a log of operations, including several successful and failed queries, and a warning about a delete operation.

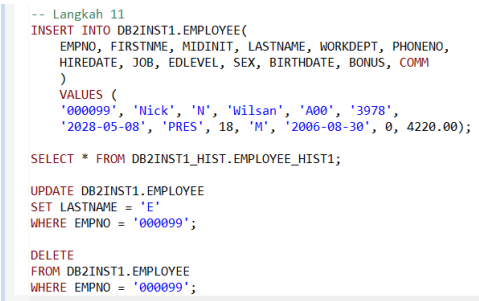
Status	Operation	Date	Connectio...
✓ Succeed	SELECT '000...	02/04/17 15...	SAMPLE
✗ Failed	DELETE FR...	02/04/17 15...	SAMPLE
✓ Succeed	DELETE FR...	02/04/17 15...	SAMPLE
✓ Succeed	INSERT INT...	02/04/17 15...	SAMPLE
✓ Succeed	SELECT * F...	02/04/17 15...	SAMPLE
⚠ Warnin	delete FRO...	02/04/17 15...	SAMPLE
✓ Succeed	delete from...	02/04/17 15...	SAMPLE
✓ Succeed	SELECT * F...	02/04/17 15...	SAMPLE
⚠ Warnin	delete FRO...	02/04/17 15...	SAMPLE
✓ Succeed	SELECT * F...	02/04/17 15...	SAMPLE
✓ Succeed	INSERT INT...	02/04/17 15...	SAMPLE
✓ Succeed	SELECT * F...	02/04/17 15...	SAMPLE

Lakukan proses update pada lastname pada data baru dengan nilai inisial pada lastname anda

```
UPDATE "USER-PC".EMPLOYEE  
SET LASTNAME = 'E'  
WHERE EMPNO = '000099'@
```

Proses update pada sebuah tabel yang memiliki history tabel akan mencatat proses update yang terjadi dengan menyimpan nilai sebelum proses update dilakukan ke dalam tabel history

B. Screenshoot



```
-- Langkah 11
INSERT INTO DB2INST1.EMPLOYEE(
EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO,
HIREDATE, JOB, EDLEVEL, SEX, BIRTHDATE, BONUS, COMM
)
VALUES (
'000099', 'Nick', 'N', 'Wilson', 'A00', '3978',
'2028-05-08', 'PRES', 18, 'M', '2006-08-30', 0, 4220.00);

SELECT * FROM DB2INST1.HIST.EMPLOYEE_HIST1;

UPDATE DB2INST1.EMPLOYEE
SET LASTNAME = 'E'
WHERE EMPNO = '000099';

DELETE
FROM DB2INST1.EMPLOYEE
WHERE EMPNO = '000099';
```

SQL Results

```
DELETE
FROM DB2INST1.EMPLOYEE
WHERE EMPNO = '000099'

Updated 1 rows.

Query execution time => 21 ms

Script: Script6.sql
Database Name: SAMPLEDB
Authorization Id (Database): db2inst1
System/IP Address : NickWilson/10.200.227.10
User Id (System) : NICK WILSON
```

C. Penjelasan (Kesimpulan, Kendala, dll.)

Melakukan insert pada tabel employee, saat di cek, data setelah proses insert pada tabel utama tidak akan tercatat dalam tabel history. Lakukan proses update pada lastname pada data baru, Proses update pada sebuah tabel yang memiliki history tabel akan mencatat proses update yang terjadi dengan menyimpan nilai sebelum proses update dilakukan ke dalam tabel history. Lakukan delete pada data baru, kemudian cek hasil, Data yang masuk adalah data employee sebelum dilakukan proses delete

LANGKAH 12

A. Soal

12. Time Travelling Query :

Time travelling query adalah query yang bisa menampilkan kondisi tabel sesuai dengan waktu yang kita inginkan. Berikut beberapa contoh penggunaan timetravelling query:

```
SELECT *  
FROM "USER-PC".EMPLOYEE  
    FOR SYSTEM_TIME AS OF '2017-03-27 09:20:10'@
```

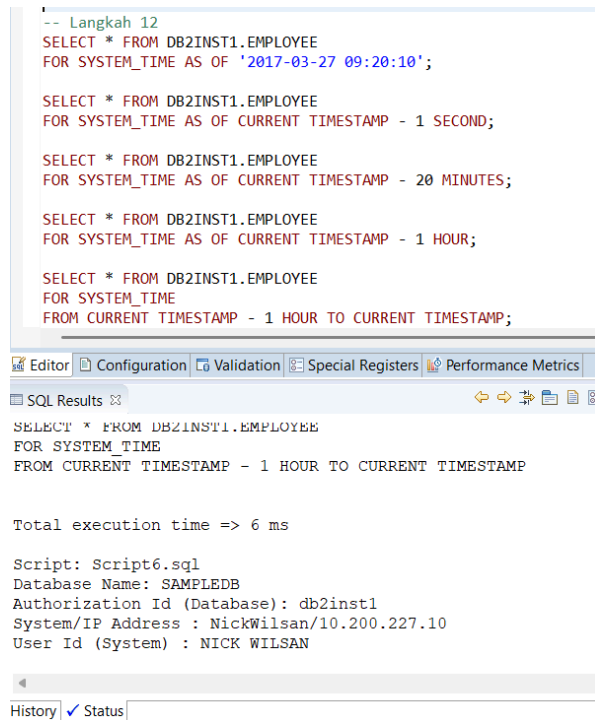
```
SELECT *  
FROM "USER-PC".EMPLOYEE  
    FOR SYSTEM_TIME AS OF CURRENT TIMESTAMP - 1 SECOND@
```

```
SELECT *  
FROM "USER-PC".EMPLOYEE  
    FOR SYSTEM_TIME AS OF CURRENT TIMESTAMP - 20 MINUTE@
```

```
SELECT *  
FROM "USER-PC".EMPLOYEE  
    FOR SYSTEM_TIME AS OF CURRENT TIMESTAMP - 1 HOUR@
```

```
SELECT *  
FROM "USER-PC".EMPLOYEE  
    FOR SYSTEM_TIME FROM CURRENT TIMESTAMP - 1 HOUR TO CURRENT TIMESTAMP @
```

B. Screenshoot



```
-- Langkah 12
SELECT * FROM DB2INST1.EMPLOYEE
FOR SYSTEM_TIME AS OF '2017-03-27 09:20:10';

SELECT * FROM DB2INST1.EMPLOYEE
FOR SYSTEM_TIME AS OF CURRENT_TIMESTAMP - 1 SECOND;

SELECT * FROM DB2INST1.EMPLOYEE
FOR SYSTEM_TIME AS OF CURRENT_TIMESTAMP - 20 MINUTES;

SELECT * FROM DB2INST1.EMPLOYEE
FOR SYSTEM_TIME AS OF CURRENT_TIMESTAMP - 1 HOUR;

SELECT * FROM DB2INST1.EMPLOYEE
FOR SYSTEM_TIME
FROM CURRENT_TIMESTAMP - 1 HOUR TO CURRENT_TIMESTAMP;
```

SQL Results

```
SELECT * FROM DB2INST1.EMPLOYEE
FOR SYSTEM_TIME
FROM CURRENT_TIMESTAMP - 1 HOUR TO CURRENT_TIMESTAMP
```

Total execution time => 6 ms

Script: Script6.sql
Database Name: SAMPLEDB
Authorization Id (Database): db2inst1
System/IP Address : NickWilson/10.200.227.10
User Id (System) : NICK WILSON

History ☒ Status

C. Penjelasan (Kesimpulan, Kendala, dll.)

Query tersebut adalah query time travelling, yang dimana bisa menampilkan kondisi tabel sesuai dengan waktu yang kita inginkan.