
NAMA : Muhammad Zaqy Ulunnuha
NIM : 245150407111059
KELAS : SI - E
BAB : VI – DATABASE APPLICATION DEVELOPMENT

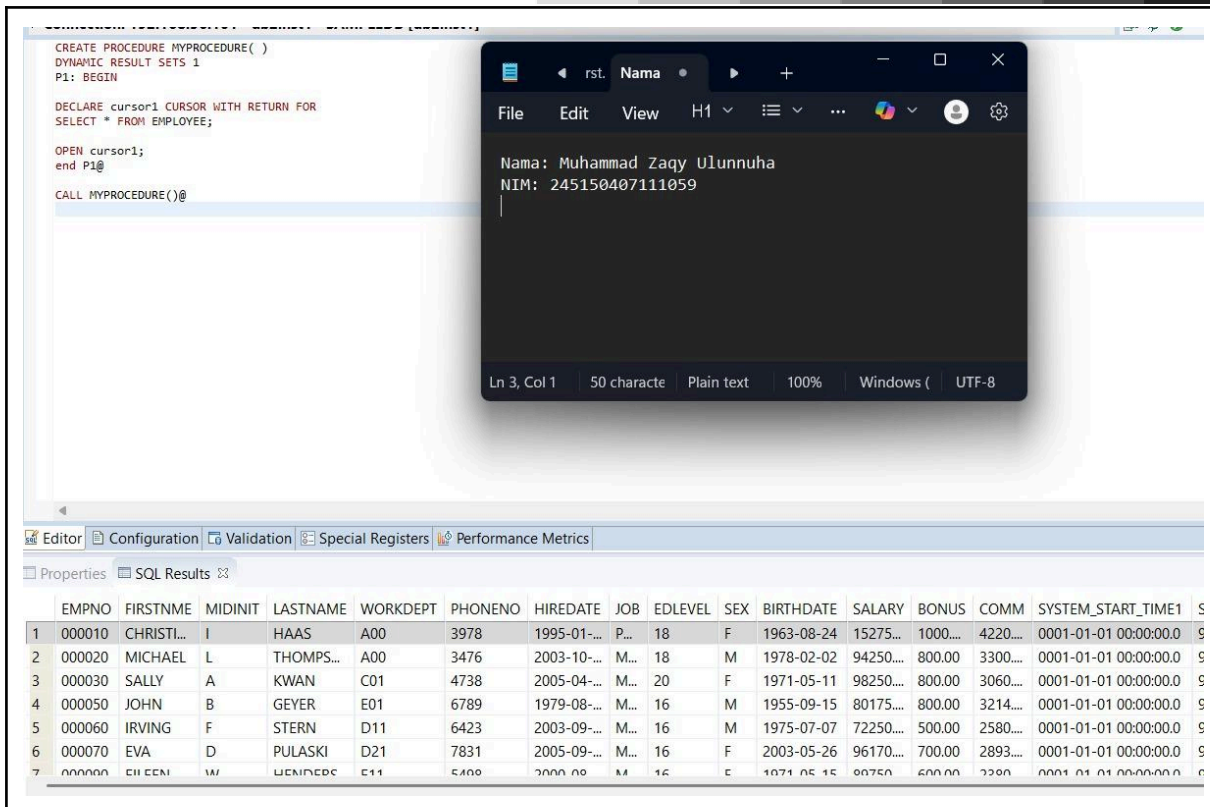
1. Langkah 1

1.1 Soal

1.2 Query

```
CREATE PROCEDURE MYPROCEDURE( )  
DYNAMIC RESULT SETS 1  
P1: BEGIN  
  
DECLARE cursor1 CURSOR WITH RETURN FOR  
SELECT * FROM EMPLOYEE;  
  
OPEN cursor1;  
end P1@  
  
CALL MYPROCEDURE()@
```

1.3 Screenshot



The screenshot shows a database IDE with a PL/SQL editor and a results window. The editor contains the following code:

```
CREATE PROCEDURE MYPROCEDURE( )
DYNAMIC RESULT SETS 1
P1: BEGIN

DECLARE cursor1 CURSOR WITH RETURN FOR
SELECT * FROM EMPLOYEE;

OPEN cursor1;
end P1@

CALL MYPROCEDURE()@
```

The results window, titled "Nama", displays the following text:

```
Nama: Muhammad Zaqy Ulunnuha
NIM: 245150407111059
```

Below the editor, the "SQL Results" tab is active, showing a table with the following data:

	EMPNO	FIRSTNAME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM	SYSTEM_START_TIME1	S
1	000010	CHRISTI...	I	HAAS	A00	3978	1995-01-...	P...	18	F	1963-08-24	15275...	1000...	4220...	0001-01-01 00:00:00.0	S
2	000020	MICHAEL	L	THOMPS...	A00	3476	2003-10-...	M...	18	M	1978-02-02	94250...	800.00	3300...	0001-01-01 00:00:00.0	S
3	000030	SALLY	A	KWAN	C01	4738	2005-04-...	M...	20	F	1971-05-11	98250...	800.00	3060...	0001-01-01 00:00:00.0	S
4	000050	JOHN	B	GEYER	E01	6789	1979-08-...	M...	16	M	1955-09-15	80175...	800.00	3214...	0001-01-01 00:00:00.0	S
5	000060	IRVING	F	STERN	D11	6423	2003-09-...	M...	16	M	1975-07-07	72250...	500.00	2580...	0001-01-01 00:00:00.0	S
6	000070	EVA	D	PULASKI	D21	7831	2005-09-...	M...	16	F	2003-05-26	96170...	700.00	2893...	0001-01-01 00:00:00.0	S
7	000000	CHEN	W	WENIGER	E11	5400	2000-09-...	M...	16	F	1971-05-15	90750...	600.00	2200...	0001-01-01 00:00:00.0	S

1.4 Penjelasan

Query ini pertama akan membuat prosedur bernama myprocedure yang akan menunjukkan semua data di tabel employee. Kemudian prosedur tersebut akan di panggil menggunakan call procedure untuk menjalankan prosedur.

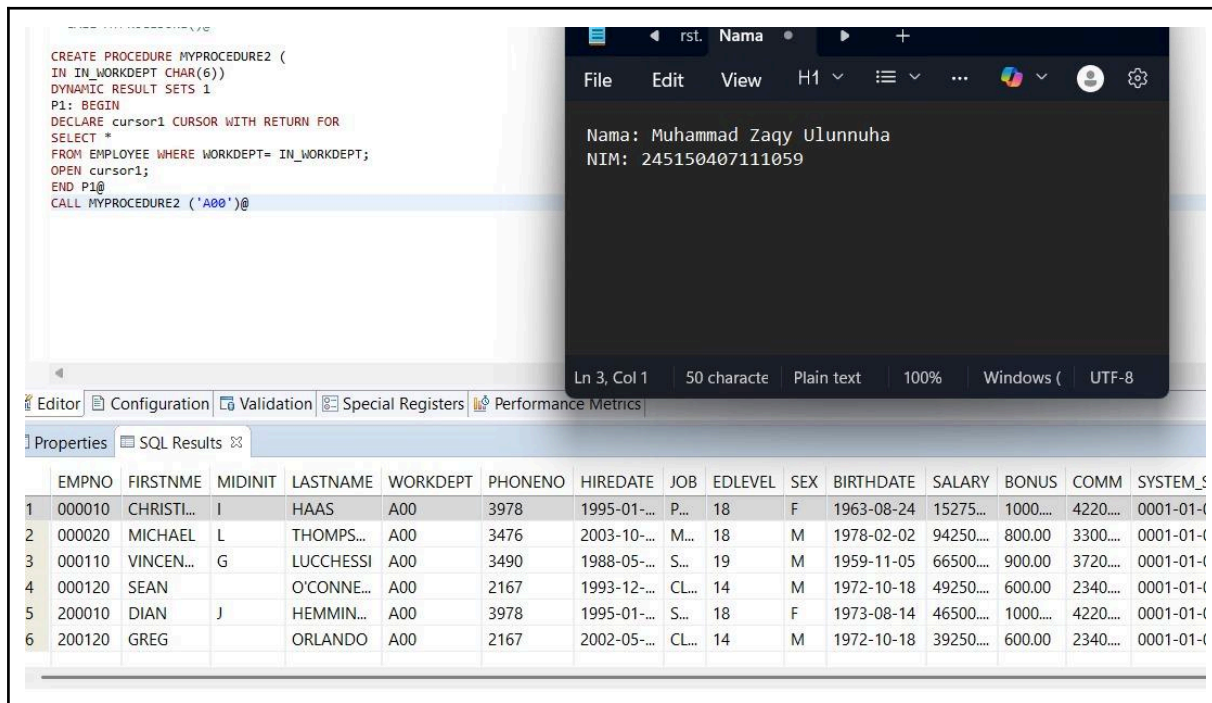
2. Langkah 2

2.1 Soal

2.2 Query

```
CREATE PROCEDURE MYPROCEDURE2 (
IN IN_WORKDEPT CHAR(6))
DYNAMIC RESULT SETS 1
P1: BEGIN
DECLARE cursor1 CURSOR WITH RETURN FOR
SELECT *
FROM EMPLOYEE WHERE WORKDEPT= IN_WORKDEPT;
OPEN cursor1;
END P1@
CALL MYPROCEDURE2 ('A00')@
```

2.3 Screenshot



The screenshot shows a database IDE with a SQL query editor on the left and a results window on the right. The query is the same as in section 2.2. The results window displays the output of the query, showing the name and NIM of the user.

Query Results:

```
Nama: Muhammad Zaqy Ulunnuha
NIM: 245150407111059
```

Below the results window, a table of employee data is visible. The table has 15 columns: EMPNO, FIRSTNAME, MIDDLEINITIAL, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB, EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM, and SYSTEM_ID. The first 6 rows of data are shown.

	EMPNO	FIRSTNAME	MIDDLEINITIAL	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM	SYSTEM_ID
1	000010	CHRISTI...	I	HAAS	A00	3978	1995-01-...	P...	18	F	1963-08-24	15275...	1000...	4220...	0001-01-4
2	000020	MICHAEL	L	THOMPS...	A00	3476	2003-10-...	M...	18	M	1978-02-02	94250...	800.00	3300...	0001-01-4
3	000110	VINCEN...	G	LUCCHESI	A00	3490	1988-05-...	S...	19	M	1959-11-05	66500...	900.00	3720...	0001-01-4
4	000120	SEAN		O'CONNE...	A00	2167	1993-12-...	CL...	14	M	1972-10-18	49250...	600.00	2340...	0001-01-4
5	200010	DIAN	J	HEMMIN...	A00	3978	1995-01-...	S...	18	F	1973-08-14	46500...	1000...	4220...	0001-01-4
6	200120	GREG		ORLANDO	A00	2167	2002-05-...	CL...	14	M	1972-10-18	39250...	600.00	2340...	0001-01-4

2.4 Penjelasan

Query ini akan membuat prosedur bernama MYPROCEDURE2 yang memiliki parameter CHAR(6). MYPROCEDURE2 akan menunjukkan semua data tabel employee yang memiliki nilai WORKDEPT sesuai dengan input parameter. Kemudian prosedur tersebut akan dipanggil menggunakan CALL prosedur dengan parameter yang sesuai.

3. Langkah 3

3.1 Soal

3.2 Query

```
CREATE OR REPLACE PROCEDURE SP_UPDATE(
```

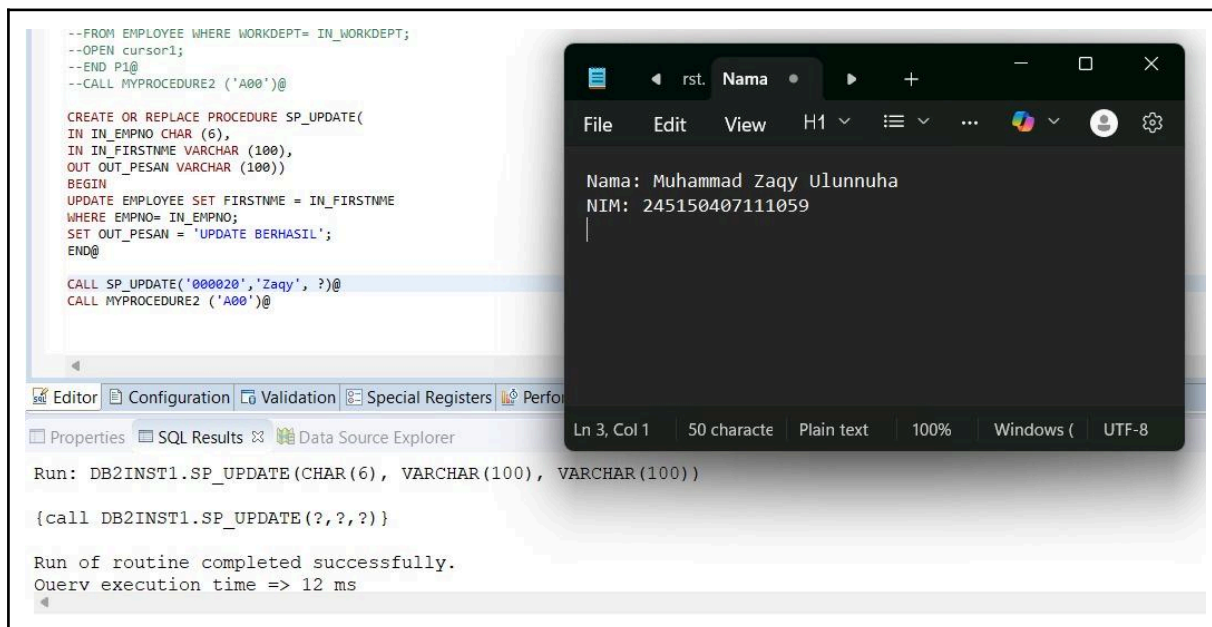
```

IN IN_EMPNO CHAR (6),
IN IN_FIRSTNME VARCHAR (100),
OUT OUT_PESAN VARCHAR (100))
BEGIN
UPDATE EMPLOYEE SET FIRSTNME = IN_FIRSTNME
WHERE EMPNO= IN_EMPNO;
SET OUT_PESAN = 'UPDATE BERHASIL';
END@

CALL SP_UPDATE('000020','Zaqy', ?)@
CALL MYPROCEDURE2 ('A00')@

```

3.3 Screenshot



3.4 Penjelasan

Query ini akan membuat prosedur bernama SP_UPDATE dengan 2 inputan dan output untuk menunjukkan keberhasilan prosedur. Prosedur ini akan mengupdate data tabel employee dengan firsrtnme sama seperti inputan parameter prosedur tersebut dan kemudian menggantinya dengan yang baru. Setelah pembuatan prosedur, query memanggil prosedur untuk mengupdate data firstname employee dengan empno '000020' menjadi nama 'zaqy'. MYPROCEDURE2 yang dibuat sebelumnya dapat dipanggil untuk memastikan data berhasil diupdate.

4. Langkah 4

4.1 Soal

4.2 Query

```

CREATE FUNCTION myfunction( )
RETURNS INTEGER

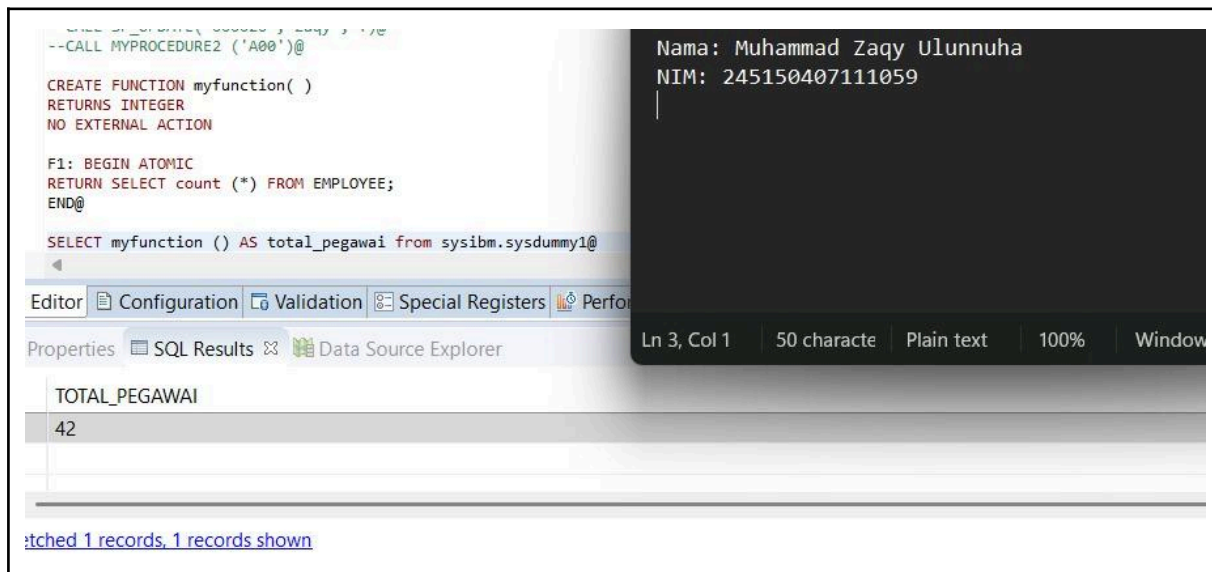
```

NO EXTERNAL ACTION

F1: BEGIN ATOMIC
RETURN SELECT count (*) FROM EMPLOYEE;
END@

SELECT myfunction () AS total_pegawai from sysibm.sysdummy1@

4.3 Screenshot



4.4 Penjelasan

Query ini akan membuat function bernama myfunction yang akan mengembalikan nilai integer jumlah data employee. Function ini kemudian akan dipanggil melalui SELECT untuk mendapatkan nilai kembali nya.

5. Langkah 5

5.1 Soal

5.2 Query

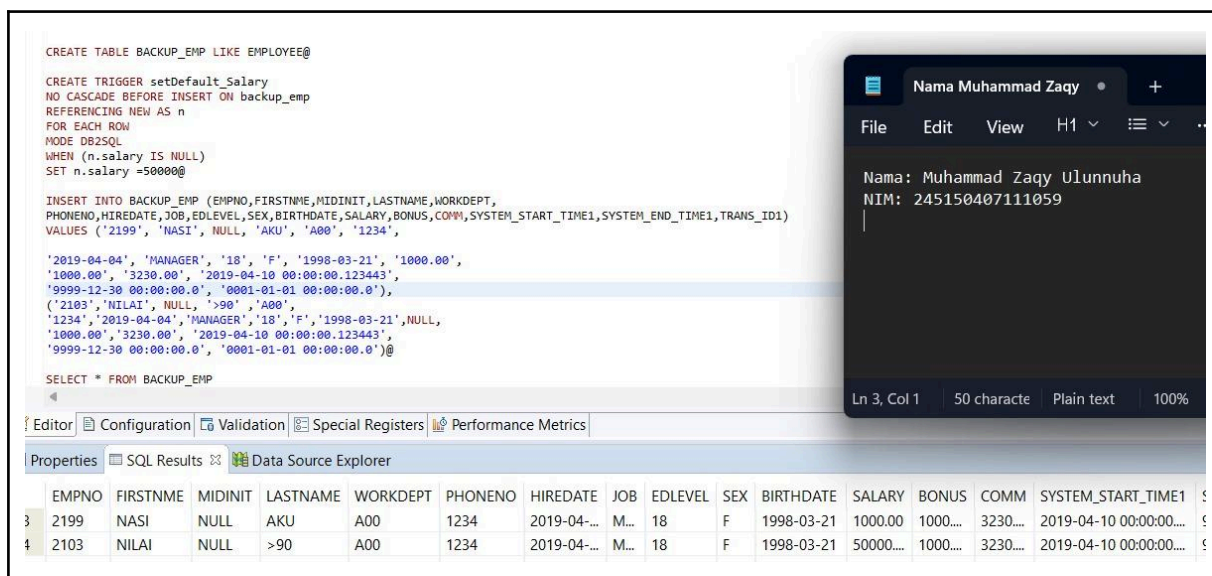
```
CREATE TABLE BACKUP_EMP LIKE EMPLOYEE@
CREATE TRIGGER setDefault_Salary
NO CASCADE BEFORE INSERT ON backup_emp
REFERENCING NEW AS n
FOR EACH ROW
MODE DB2SQL
WHEN (n.salary IS NULL)
SET n.salary =50000@
INSERT INTO BACKUP_EMP (EMPNO,FIRSTNME,MIDINIT,LASTNAME,WORKDEPT,
```

```
PHONENO, HIREDATE, JOB, EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM, SYSTEM_
_START_TIME1, SYSTEM_END_TIME1, TRANS_ID1)
VALUES ('2199', 'NASI', NULL, 'AKU', 'A00', '1234',
```

```
'2019-04-04', 'MANAGER', '18', 'F', '1998-03-21', '1000.00',
'1000.00', '3230.00', '2019-04-10 00:00:00.123443',
'9999-12-30 00:00:00.0', '0001-01-01 00:00:00.0'),
('2103', 'NILAI', NULL, '>90', 'A00',
'1234', '2019-04-04', 'MANAGER', '18', 'F', '1998-03-21', NULL,
'1000.00', '3230.00', '2019-04-10 00:00:00.123443',
'9999-12-30 00:00:00.0', '0001-01-01 00:00:00.0')@
```

```
SELECT * FROM BACKUP_EMP@
```

5.3 Screenshot



```
CREATE TABLE BACKUP_EMP LIKE EMPLOYEE@

CREATE TRIGGER setDefault_Salary
NO CASCADE BEFORE INSERT ON backup_emp
REFERENCING NEW AS n
FOR EACH ROW
MODE DB2SQL
WHEN (n.salary IS NULL)
SET n.salary =50000@

INSERT INTO BACKUP_EMP (EMPNO, FIRSTNAME, MIDINIT, LASTNAME, WORKDEPT,
PHONENO, HIREDATE, JOB, EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM, SYSTEM_START_TIME1, SYSTEM_END_TIME1, TRANS_ID1)
VALUES ('2199', 'NASI', NULL, 'AKU', 'A00', '1234',
'2019-04-04', 'MANAGER', '18', 'F', '1998-03-21', '1000.00',
'1000.00', '3230.00', '2019-04-10 00:00:00.123443',
'9999-12-30 00:00:00.0', '0001-01-01 00:00:00.0'),
('2103', 'NILAI', NULL, '>90', 'A00',
'1234', '2019-04-04', 'MANAGER', '18', 'F', '1998-03-21', NULL,
'1000.00', '3230.00', '2019-04-10 00:00:00.123443',
'9999-12-30 00:00:00.0', '0001-01-01 00:00:00.0')@

SELECT * FROM BACKUP_EMP
```

	EMPNO	FIRSTNAME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM	SYSTEM_START_TIME1	SYSTEM_END_TIME1	TRANS_ID1
3	2199	NASI	NUL	AKU	A00	1234	2019-04-04	MANAGER	18	F	1998-03-21	1000.00	1000.00	3230.00	2019-04-10 00:00:00.123443	9999-12-30 00:00:00.0	0001-01-01 00:00:00.0
4	2103	NILAI	NUL	>90	A00	1234	2019-04-04	MANAGER	18	F	1998-03-21	1000.00	1000.00	3230.00	2019-04-10 00:00:00.123443	9999-12-30 00:00:00.0	0001-01-01 00:00:00.0

5.4 Penjelasan

Query ini pertama akan membuat tabel seperti tabel employee. Kemudian query akan membuat trigger untuk memasukkan nilai 50000 ke salary apabila kolom salary bernilai null. Setelah itu query memasukkan 2 data ke tabel yang telah dibuat dengan data pertama memiliki salary senilai 100000 dan data kedua memiliki salary null. Karena terdapat Trigger sebelum data tersebut dimasukkan, nilai salary null diubah menjadi 50000. Untuk memastikan trigger berhasil berjalan, pemanggilan SELECT digunakan untuk mengecek perubahan data yang dimasukkan.

6. Langkah 6

6.1 Soal

6.2 Query

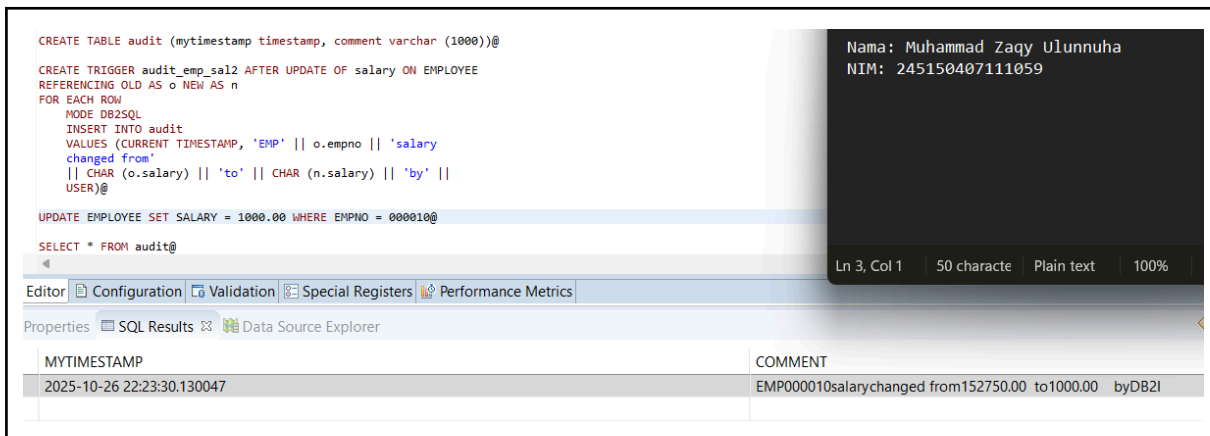

```
CREATE TABLE audit (mytimestamp timestamp, comment varchar (1000))@

CREATE TRIGGER audit_emp_sal2 AFTER UPDATE OF salary ON EMPLOYEE
REFERENCING OLD AS o NEW AS n
FOR EACH ROW
    MODE DB2SQL
    INSERT INTO audit
    VALUES (CURRENT TIMESTAMP, 'EMP' || o.empno || 'salary
changed from'
|| CHAR (o.salary) || 'to' || CHAR (n.salary) || 'by' ||
USER)@

UPDATE EMPLOYEE SET SALARY = 1000.00 WHERE EMPNO = 000010@

SELECT * FROM audit@
```

6.3 Screenshot



The screenshot shows a SQL IDE with the following SQL commands executed:

```
CREATE TABLE audit (mytimestamp timestamp, comment varchar (1000))@
CREATE TRIGGER audit_emp_sal2 AFTER UPDATE OF salary ON EMPLOYEE
REFERENCING OLD AS o NEW AS n
FOR EACH ROW
    MODE DB2SQL
    INSERT INTO audit
    VALUES (CURRENT TIMESTAMP, 'EMP' || o.empno || 'salary
changed from'
|| CHAR (o.salary) || 'to' || CHAR (n.salary) || 'by' ||
USER)@
UPDATE EMPLOYEE SET SALARY = 1000.00 WHERE EMPNO = 000010@
SELECT * FROM audit@
```

A tooltip shows the user information: Nama: Muhammad Zaqy Ulunnuha, NIM: 245150407111059.

The SQL Results pane shows the following data:

MYTIMESTAMP	COMMENT
2025-10-26 22:23:30.130047	EMP000010salarychanged from152750.00 to1000.00 byDB2I

6.4 Penjelasan

Query terakhir ini akan membuat tabel bernama audit serta trigger saat setelah UPDATE kolom salary di tabel employee dilakukan. Untuk mengetahui data mana yang lama dan yang baru, trigger menggunakan o untuk data lama dan n sebagai data baru. Kemudian data lam dan baru akan dimasukan kedalam tabel audit. Query kemudian mengupdate data salary dari tabel employee untuk mengecek keberhasilan trigger

KESIMPULAN

Stored Procedure merupakan sekumpulan perintah SQL yang disimpan di dalam database dan dapat dijalankan kembali secara langsung menggunakan perintah CALL.

Prosedur ini memudahkan proses otomasi pekerjaan di database, seperti menampilkan data, memproses input, hingga melakukan pembaruan (update) pada tabel.

Contohnya pada prosedur SP_UPDATE, proses perubahan data pada tabel employee dapat dilakukan hanya dengan satu pemanggilan prosedur tanpa harus menulis ulang perintah UPDATE.

Function berfungsi untuk mengembalikan satu nilai tertentu sebagai hasil dari proses di dalam database.

Fungsi dapat digunakan di dalam perintah SQL lainnya seperti SELECT, sehingga mempermudah pengambilan informasi yang bersifat perhitungan atau agregasi.

Misalnya pada myfunction, fungsi ini mengembalikan jumlah total data dari tabel employee.

Trigger adalah mekanisme otomatis yang dijalankan oleh database saat terjadi peristiwa tertentu, seperti INSERT, UPDATE, atau DELETE.

Pada contoh yang dibuat, trigger dijalankan setelah (AFTER) proses UPDATE pada kolom salary tabel employee.

Trigger tersebut menyimpan data lama dan data baru ke tabel audit, sehingga dapat digunakan untuk melacak perubahan data yang terjadi di sistem.