

Pertemuan 6



Albi Nur Rosif

3122522010

D3 PSDKU Sumenep

PRODI D3 TEKNIK INFORMATIKA

DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER

PENS PSDKU SUMENEP

Latihan 1

Tugas :

- Mencegah penggunaan password yang sederhana
- Kemampuan account untuk mengunci dalam waktu 10 menit ketika terjadi kesalahan login
- Membebaskan aplikasi login server dari perubahan password
- Kegagalan audit untuk koneksi ke database

- Mencegah penggunaan password yang sederhana

```
Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> CREATE OR REPLACE FUNCTION verify_password_complexity (
 2     username IN VARCHAR2,
 3     password IN VARCHAR2,
 4     old_password IN VARCHAR2
 5 ) RETURN BOOLEAN
 6 IS
 7     pw_length PLS_INTEGER;
 8     has_uppercase BOOLEAN := FALSE;
 9     has_lowercase BOOLEAN := FALSE;
10     has_digit BOOLEAN := FALSE;
11 BEGIN
12     -- Menghitung panjang password
13     pw_length := LENGTH(password);
14
15     -- Memeriksa panjang password minimal 8 karakter
16     IF pw_length < 8 THEN
17         RETURN FALSE;
18     END IF;
19
20     -- Memeriksa apakah password mengandung setidaknya satu huruf besar, satu huruf kecil, dan satu angka
21     FOR i IN 1..pw_length LOOP
22         IF ASCII(SUBSTR(password, i, 1)) BETWEEN ASCII('A') AND ASCII('Z') THEN
23             has_uppercase := TRUE;
24         ELSIF ASCII(SUBSTR(password, i, 1)) BETWEEN ASCII('a') AND ASCII('z') THEN
25             has_lowercase := TRUE;
26         ELSIF ASCII(SUBSTR(password, i, 1)) BETWEEN ASCII('0') AND ASCII('9') THEN
27             has_digit := TRUE;
28         END IF;
29     END LOOP;
30
31     -- Mengembalikan TRUE jika password memenuhi kriteria, FALSE jika tidak
32     RETURN (has_uppercase AND has_lowercase AND has_digit);
33 END;
34 /

Function created.
```

```
SQL> ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION VERIFY_PASSWORD_COMPLEXITY;

Profile altered.
```

Cek

Berhasil, password berisi huruf besar, kecil, angka, dan jumlah karakter tidak kurang dari 8

```
SQL> CREATE USER C##bee2 IDENTIFIED by "Password123";

User created.
```

Gagal, password terlalu sederhana

```
SQL> CREATE USER C##bee3 IDENTIFIED by "pass";
CREATE USER C##bee3 IDENTIFIED by "pass"
*
ERROR at line 1:
ORA-28003: password verification for the specified password failed

SQL>
```

- Kemampuan account untuk mengunci dalam waktu 10

menit ketika terjadi kesalahan login

```
SQL> ALTER PROFILE default LIMIT FAILED_LOGIN_ATTEMPTS 3 PASSWORD_LOCK_TIME 10/1440;  
Profile altered.
```

Cek

password yang saya gunakan untuk login ke user c##bee2 salah sebanyak 3 kali

maka akun dikunci

```
SQL> connect c##bee2/pass  
ERROR:  
ORA-01017: invalid username/password; logon denied  
  
Warning: You are no longer connected to ORACLE.  
SQL> connect sys as sysdba;  
Enter password:  
Connected.  
SQL> connect c##bee2/pass  
ERROR:  
ORA-28000: The account is locked.
```

- Membebaskan aplikasi login server dari perubahan

password

```
SQL> CREATE USER C##bee1 IDENTIFIED EXTERNALLY;
User created.
```

```
SQL> connect sys as sysdba;
Enter password:
Connected.
SQL> ALTER PROFILE DEFAULT LIMIT
      2 PASSWORD_REUSE_TIME UNLIMITED;

Profile altered.

SQL> ALTER USER c##bee2 IDENTIFIED BY Password1234;

User altered.

SQL> |
```

- Kegagalan audit untuk koneksi ke database

```
SQL> NOAUDIT CONNECT;
Noaudit succeeded.
```

```
SQL> SELECT audit_type,
2         unified_audit_policies,
3         action_name,
4         return_code,
5         COUNT (*)
6         FROM unified_audit_trail
7         WHERE unified_audit_policies = 'ORA_LOGON_FAILURES'
8 GROUP BY audit_type,
9         unified_audit_policies,
10        action_name,
11        return_code
12 ORDER BY COUNT (*);

AUDIT_TYPE
-----
UNIFIED_AUDIT_POLICIES
-----
ACTION_NAME                                RETURN_CODE
-----
COUNT(*)
-----
Standard
ORA_LOGON_FAILURES
LOGON                                     28800
1

AUDIT_TYPE
-----
UNIFIED_AUDIT_POLICIES
-----
ACTION_NAME                                RETURN_CODE
-----
COUNT(*)
-----
Standard
ORA_LOGON_FAILURES
LOGON                                     1045
3

AUDIT_TYPE
-----
UNIFIED_AUDIT_POLICIES
-----
ACTION_NAME                                RETURN_CODE
-----
COUNT(*)
-----
Standard
ORA_LOGON_FAILURES
LOGON                                     28809
4

AUDIT_TYPE
-----
UNIFIED_AUDIT_POLICIES
-----
ACTION_NAME                                RETURN_CODE
-----
COUNT(*)
-----
Standard
ORA_LOGON_FAILURES
LOGON                                     1017
31

SQL>
```


Latihan 2

Tugas :

- Audit SELECT pada kolom SALARY pada tabel

EMPLOYEES

- Audit perubahan pada kolom SALARY dari tabel

EMPLOYEES:

- Nilai lama
- Nilai baru
- User yang membuat perubahan
- Lokasi mana yang telah diubah dari yang dibuat

```
SQL> CREATE TABLE EMPLOYEES (  
2     ID INT PRIMARY KEY,  
3     NAME VARCHAR2(50),  
4     SALARY NUMBER(10, 2)  
5 );
```

Table created.

```
SQL> CREATE USER c##bee_audit IDENTIFIED BY Password123;
```

User created.

```
SQL> GRANT CREATE TABLE TO c##bee_audit;
```

Grant succeeded.

```
SQL> GRANT CREATE TRIGGER TO c##bee_audit;
```

Grant succeeded.

```
SQL> GRANT CREATE SESSION TO c##bee_audit;
```

Grant succeeded.

```
SQL> GRANT SELECT ON EMPLOYEES TO c##bee_audit;
```

Grant succeeded.

```
SQL> AUDIT INSERT, UPDATE, DELETE ON EMPLOYEES;
```

Audit succeeded.

```
SQL> GRANT INSERT ON EMPLOYEES TO c##bee_audit;
```

Grant succeeded.

```
SQL> GRANT UNLIMITED TABLESPACE TO c##bee_audit;
```

Grant succeeded.

```
SQL> GRANT UPDATE ON EMPLOYEES TO c##bee_audit;
```

Grant succeeded.

```
SQL> connect c##bee_audit/Password123
Connected.
```

```
SQL> CREATE TABLE EMPLOYEES (
  2     ID INT PRIMARY KEY,
  3     NAME VARCHAR2(50),
  4     SALARY NUMBER(10, 2)
  5 );
```

```
Table created.
```

```
SQL> INSERT INTO EMPLOYEES (ID, NAME, SALARY) VALUES (1, 'Bee', 10000);
```

```
1 row created.
```

```
SQL> INSERT INTO EMPLOYEES (ID, NAME, SALARY) VALUES (2, 'Eka', 9000);
```

```
1 row created.
```

```
SQL> INSERT INTO EMPLOYEES (ID, NAME, SALARY) VALUES (3, 'Hasna', 8000);
```

```
1 row created.
```

```
SQL> INSERT INTO EMPLOYEES (ID, NAME, SALARY) VALUES (4, 'Nur', 7000);
```

```
1 row created.
```

```
SQL> INSERT INTO EMPLOYEES (ID, NAME, SALARY) VALUES (5, 'Albi', 9000);
```

```
1 row created.
```


- Audit SELECT pada kolom SALARY pada tabel

EMPLOYEES

```
SQL> AUDIT SELECT ON EMPLOYEES BY ACCESS;

Audit succeeded.
```

```
SQL> SELECT * FROM USER_AUDIT_TRAIL WHERE obj_name = 'EMPLOYEES' AND action_name = 'SELECT';
OS_USERNAME
-----
USERNAME
-----
USERHOST
-----
TERMINAL
-----
TIMESTAMP
-----
OWNER
-----
OBJ_NAME
-----
ACTION ACTION_NAME
-----
NEW_OWNER
-----
NEW_NAME
-----
OBJ_PRIVILEGE          SYS_PRIVILEGE          A
GRANTEE
-----
AUDIT_OPTION          SES_ACTIONS          LOGOFF_TI
LOGOFF_LREAD LOGOFF_PREAD LOGOFF_LWRITE LOGOFF_DLOCK
COMMENT_TEXT
-----
SESSIONID ENTRYID STATEMENTID RETURNCODE
PRIV_USED
-----
CLIENT_ID
-----
ECONTEXT_ID          SESSION_CPU
EXTENDED_TIMESTAMP
-----
PROXY_SESSIONID GLOBAL_UID          INSTANCE_NUMBER
OS_PROCESS          TRANSACTIONID          SCN
```

- Audit perubahan pada kolom SALARY dari tabel

EMPLOYEES

```
SQL> SELECT * FROM USER_AUDIT_TRAIL WHERE obj_name = 'EMPLOYEES' AND action_name = 'SELECT';

no rows selected
```

```
SQL> CREATE TABLE SalaryAudit (
2     salary_lama NUMBER(10, 2),
3     salary_baru NUMBER(10, 2),
4     rubah_by VARCHAR2(100),
5     rubah_location VARCHAR2(100),
6     rubah_tgl TIMESTAMP DEFAULT CURRENT_TIMESTAMP
7 );
```

Table created.

```
SQL> CREATE OR REPLACE TRIGGER salary_audit_trigger
2 AFTER UPDATE OF SALARY ON EMPLOYEES
3 FOR EACH ROW
4 BEGIN
5     INSERT INTO SalaryAudit (salary_lama, salary_baru, rubah_by, rubah_location)
6     VALUES (:OLD.SALARY, :NEW.SALARY, USER, 'EMPLOYEES');
7 END;
8 /
```

Trigger created.

```
SQL> SELECT * FROM EMPLOYEES;
```

ID	NAME	SALARY
1	Bee	10000
2	Eka	9000
3	Hasna	8000
4	Nur	7000
5	Albi	9000

```
SQL> UPDATE EMPLOYEES SET NAME = 'Monspeet', SALARY = 6300 WHERE ID = 4;
```

```
1 row updated.
```

```
SQL> UPDATE EMPLOYEES SET NAME = 'Albi Nur Rosif', SALARY = 63000 WHERE ID = 5;
```

```
1 row updated.
```

```
SQL> SELECT * FROM EMPLOYEES;
```

ID	NAME	SALARY
1	Bee	10000
2	Eka	9000
3	Hasna	8000
4	Monspeet	6300
5	Albi Nur Rosif	63000

```
SQL> SELECT * FROM SalaryAudit;
```

```
SALARY_LAMA SALARY_BARU
```

```
RUBAH_BY
```

```
RUBAH_LOCATION
```

```
RUBAH_TGL
```

```
7000 6300  
C##BEE_AUDIT  
EMPLOYEES  
27-MAR-24 02.17.33.930000 PM
```

```
SALARY_LAMA SALARY_BARU
```

```
RUBAH_BY
```

```
RUBAH_LOCATION
```

```
RUBAH_TGL
```

```
9000 63000  
C##BEE_AUDIT  
EMPLOYEES  
27-MAR-24 02.17.35.307000 PM
```