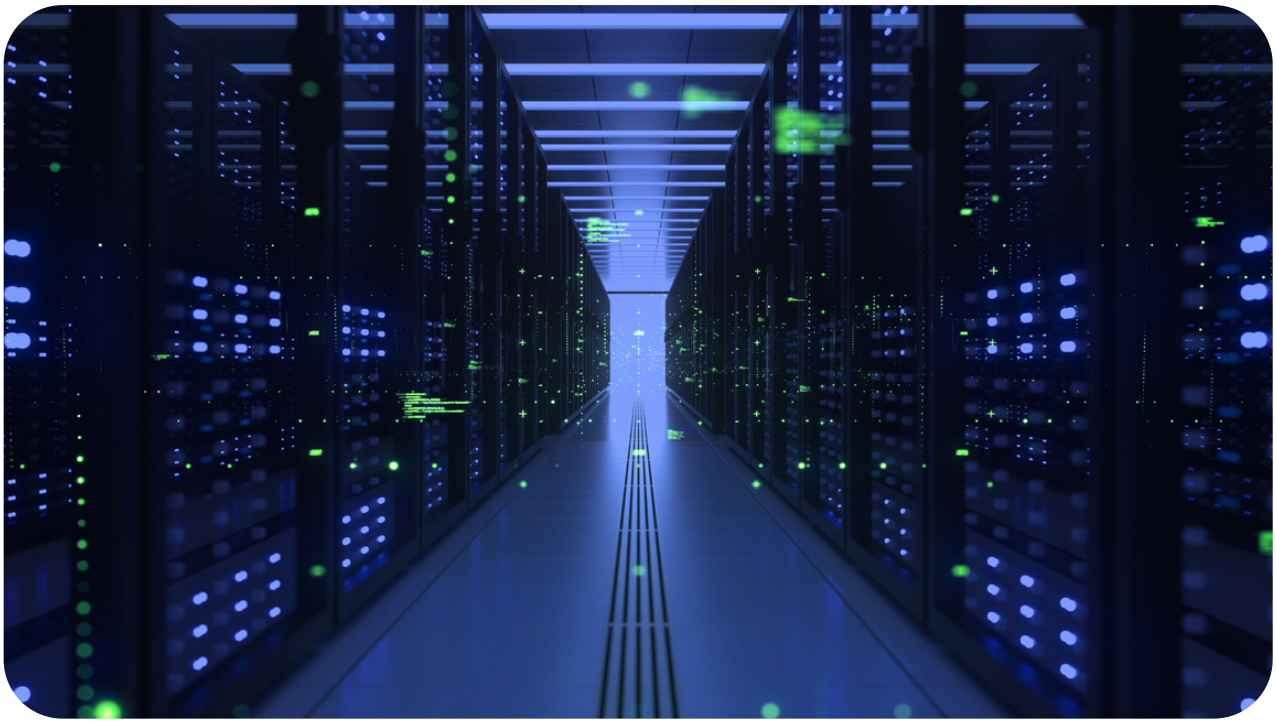
Keamanan Basis Data

Dosen Pengampu: Bapak Antoni Haikal, S.T., M.T

IMPLEMENTED VIRTUAL PRIVATE DATABASE ON POSTGRESQL



Submitted by:
Nisrina Amelia Putri
4332101006

Politeknik Negeri Batam
Teknik Informatika
Rekayasa Keamanan Siber

DAFTAR ISI

PENDAHULUAN

A. LATAR BELAKANG

Penggunaan teknologi digital kian hari kian meningkat. Banyak kegiatan, salah satunya kegiatan pengelolaan data sudah beralih ke digitalisasi demi kemudahan dalam mengolahnya. Hal ini menyebabkan bertambahnya kebutuhan akan alat pengolah data digital. Semakin besar data yang diolah maka semakin rentan pula data tersebut menjadi target pencurian peretas. Karena itu, sebuah alat pengolah data harus memiliki keamanan yang ketat dengan standar yang tepat dan pusat mitigasi yang terarah untuk mengurangi kerentanan kebocoran data.

Untuk melindungi suatu basis data, dapat menerapkan *grand access* atau *privacy policy* agar orang yang tidak berwenang tidak dapat mengakses data tersebut. Salah satu contoh *rules* yang dapat diterapkan yaitu VPD (*Virtual Private Database*). VPD yaitu sebuah aturan keamanan untuk mengontrol akses *databases* pada level baris dan kolom. Dengan menerapkan VPD, akses ke dalam sebuah basis data dapat diatur dengan ketat, sehingga kerentanan akan kebocoran data dapat dikurangi.

Hal ini juga dikarenakan ketersediaan informasi yang terus mengalami peningkatan pesat, metode untuk penyandian dan penyimpanan informasi juga mengalami peningkatan. Perkembangan jumlah sumber informasi ini membawa beberapa permasalahan, diantaranya tentang bagaimana mengkombinasikan tempat penyimpanan data yang terdistribusi dan berbeda. Informasi pada suatu organisasi atau perusahaan biasanya disimpan di lokasi yang terpisah dan berbeda-beda format. Ketika terjadi peningkatan kapasitas tempat penyimpanan dan besarnya biaya pencarian informasi, perusahaan dihadapkan pada masalah melimpahnya jumlah data. Dengan kerentanan yang besar ini, maka standar keamanan yang tepat harus diterapkan secara terencana.

B. TUJUAN

Adapun tujuan pembuatan laporan ini ialah untuk:

1. Mengetahui Dasar Teori pada *Virtual Private Database*.
2. Dapat Mengimplementasikan *Virtual Private Database*.
3. Mengetahui Performa yang dihasilkan oleh *Virtual Private Database*.
4. Dapat membuktikan implementasi dan performa yang ada pada *Virtual Private Database*.
5. Sebagai sumber untuk memenuhi nilai tugas Mata Kuliah Keamanan Basis Data.

C. LANDASAN TEORI

Virtual Private Database (VPD) adalah kebijakan keamanan database paling populer yang diperkenalkan oleh *Oracle Database Enterprise*. *Virtual Private Database* (VPD) yaitu suatu kebijakan keamanan untuk mengontrol akses database pada tingkat baris dan kolom. Pada dasarnya, VPD menambahkan klausa *WHERE* ke pernyataan SQL yang dikeluarkan terhadap tabel, tampilan, atau sinonim yang menerapkan *privacy policy* VPD. Kebijakan keamanan ini langsung ke diterapkan pada database secara otomatis setiap kali pengguna mengakses data, sehingga tidak ada cara untuk melewati keamanan (VPD).

Saat pengguna secara langsung atau tidak langsung mengakses tabel, tampilan, atau sinonim yang dilindungi dengan kebijakan keamanan VPD, Database secara dinamis mengubah pernyataan SQL pengguna. Modifikasi ini membuat kondisi *WHERE* dikembalikan oleh fungsi yang mengimplementasikan kebijakan keamanan. Kebijakan Virtual Private Database dapat digunakan untuk pernyataan *SELECT*, *INSERT*, *UPDATE*, *INDEX*, dan *DELETE*.

Dalam pemanfaatannya, VPD sangat menguntungkan sebagai proteksi pada sebuah database agar otoritas yang tidak berkepentingan tidak dapat mengakses database tersebut. Berdasarkan paparan dari *geeksforgeeks* yang ditulis pada 24 Mei 2022 lalu, VPD memiliki keuntungan dan kelemahan tersendiri. Berikut keuntungan dan kelebihan dari VPD.

VPD memiliki beberapa keuntungan, diantaranya:

- Aksesibilitas Lebih Tinggi, Pengguna dapat dengan mudah mengakses data dari mana saja.
- Fleksibilitas, Dapat dengan mudah dimodifikasi tanpa merusak aliran kontrol.
- Tingkat Pemulihan Lebih Tinggi, Data dapat diambil dengan sangat mudah.
- Diamankan Secara Dinamis, Tidak perlu mempertahankan peran yang kompleks.
- Tidak ada back doors, Kebijakan keamanan dilampirkan ke data sehingga tidak ada jalan pintas yang diizinkan.

VPD juga memiliki kelemahan, diantaranya:

- Keamanan tingkat kolom yang sulit.
- Sulit untuk diperiksa.

Contoh Penggunaan VPD:
Pengguna dapat melihat data kolom account_mgr_id “149” dari tabel pertama. Ini akan spesifik untuk dirinya sendiri kecuali lebih banyak pertanyaan disediakan. VPD dapat mengeksekusi data yang telah diperintahkan.

	CUS_LAST_NAME	CUS_FIRST_NAME	ACC_MGR_ID
	P	A	145
	Q	B	145
	R	C	147
	S	D	147
	T	E	149
	U	F	149

CUS_LAST_NAME	CUS_FIRST_NAME	CREDIT_LIMIT	ACC_MGR_ID
T	E	1200	149
U	F	1200	149

Source: geeksforgeeks

PEMBAHASAN

A. IMPLEMENTASI VIRTUAL PRIVATE DATABASE

Untuk mengimplementasikan VPD, *tool* yang digunakan adalah PostgreSQL karena bersifat *open source* dan mampu menerapkan *rules* sesuai dengan perintah yang ada pada VPD. Berikut langkah-langkah yang dapat dilakukan untuk mengimplementasikan VPD pada PostgreSQL:

1. Lakukan Instalasi PosgreSQL pada Ubuntu Server

Pastikan server sudah di-update agar layanan dapat digunakan dengan maksimal.

```
nisr@nisr:~$ sudo apt update
[sudo] password for nisr:
Hit:1 http://id.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://id.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://id.archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
43 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Lakukan instalasi PosgreSQL dengan perintah berikut.

```
nisr@nisr:~$ sudo apt install postgresql postgresql-contrib
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm14 libpq5
  libsensors-config libsensors5 libtypes-serialiser-perl postgresql-14
  postgresql-client-14 postgresql-client-common postgresql-common ssl-cert
  sysstat
Suggested packages:
```

Jalankan PostgreSQL pada server Ubuntu dengan perintah berikut.

```
nisr@nisr:~$ sudo systemctl start postgresql
[sudo] password for nisr:
```

Aktifkan PostgreSQL menggunakan perintah berikut.

```
nisr@nisr:~$ sudo systemctl enable postgresql
[sudo] password for nisr:
Synchronizing state of postgresql.service with SysV service script with /lib/sys
temd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable postgresql
```

Cek versi dan status PostgreSQL untuk lebih memastikan bahwa layanan sudah terinstal.

```
nisr@nisr:~$ psql --version
psql (PostgreSQL) 14.5 (Ubuntu 14.5-0ubuntu0.22.04.1)
nisr@nisr:~$ service postgresql status
● postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor>
     Active: active (exited) since Tue 2022-11-22 11:09:15 UTC; 1h 28min ago
   Main PID: 4009 (code=exited, status=0/SUCCESS)
        CPU: 3ms
```

Masuk ke dalam PostgreSQL dengan perintah berikut.

```
nisr@nisr:~$ sudo su postgres
[sudo] password for nisr:
postgres@nisr:/home/nisr$ psql
could not change directory to "/home/nisr": Permission denied
psql (14.5 (Ubuntu 14.5-0ubuntu0.22.04.1))
Type "help" for help.

postgres=#
```

2. Melakukan Konfigurasi pada PosgreSQL

Buat user baru dengan password sebagai wadah untuk membuat database yang akan diimplementasikan menggunakan VPD.

```
postgres=# alter role postgres with encrypted password 'nisr';
ALTER ROLE
postgres=# sudo systemctl restart postgresql
postgres-#
```

Keluar dari postgre untuk melakukan konfigurasi terlebih dahulu pada server ubuntu.

```
postgres=# \q
postgres@nisr:/home/nisr$ \q
Command 'q' not found, but can be installed with:
snap install q                      # version 1.6.3-1, or
apt  install python3-q-text-as-data  # version 3.1.6-1
See 'snap info q' for additional versions.
postgres@nisr:/home/nisr$ exit
exit
```

Ketikan perintah berikut untuk masuk ke dalam file yang akan dikonfigurasi.

```
nisr@nisr:~$ cd /etc/postgresql
nisr@nisr:/etc/postgresql$ ls
14
nisr@nisr:/etc/postgresql$ cd 14
nisr@nisr:/etc/postgresql/14$ ls
main
nisr@nisr:/etc/postgresql/14$ cd main
nisr@nisr:/etc/postgresql/14/main$ ls
conf.d       pg_ctl.conf  pg_ident.conf    start.conf
environment  pg_hba.conf  postgresql.conf
```

Gunakan perintah nano agar dapat mengedit file.

```
nisr@nisr:/etc/postgresql/14/main$ sudo nano pg_hba.conf
nisr@nisr:/etc/postgresql/14/main$
```

Lakukan perubahan pada Database administrative yang awalnya peer menjadi md5. Hal ini dilakukan agar ketika login ke dalam database, dapat menggunakan password dari user database bukan password dari server.

```
# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local   all             postgres                                md5

# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             127.0.0.1/32            scram-sha-256
# IPv6 local connections:
host    all             all             ::1/128                 scram-sha-256
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     peer
host    replication     all             127.0.0.1/32            scram-sha-256
host    replication     all             ::1/128                 scram-sha-256
```

Restart sistem agar konfigurasi dapat berjalan.

```
nisr@nisr:/etc/postgresql/14/main$ sudo systemctl restart postgresql
nisr@nisr:/etc/postgresql/14/main$
```

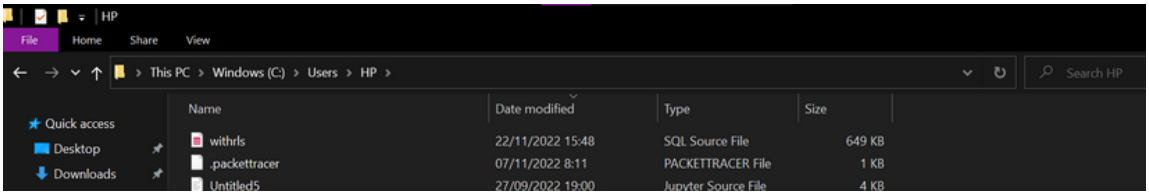
Selanjutnya, masuk ke database menggunakan user dan password yang telah dibuat tadi. \l digunakan untuk melihat list database dan perintah \du untuk melihat list user.

```
nisr@nisr:~$ psql -U postgres
Password for user postgres:
psql (14.5 (Ubuntu 14.5-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# \l
                                 List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
-----------+----------+----------+-------------+-------------+-----------------------
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
(3 rows)

postgres=# \du
                                   List of roles
 Role name |                         Attributes                         | Member of
-----------+------------------------------------------------------------+-----------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
```

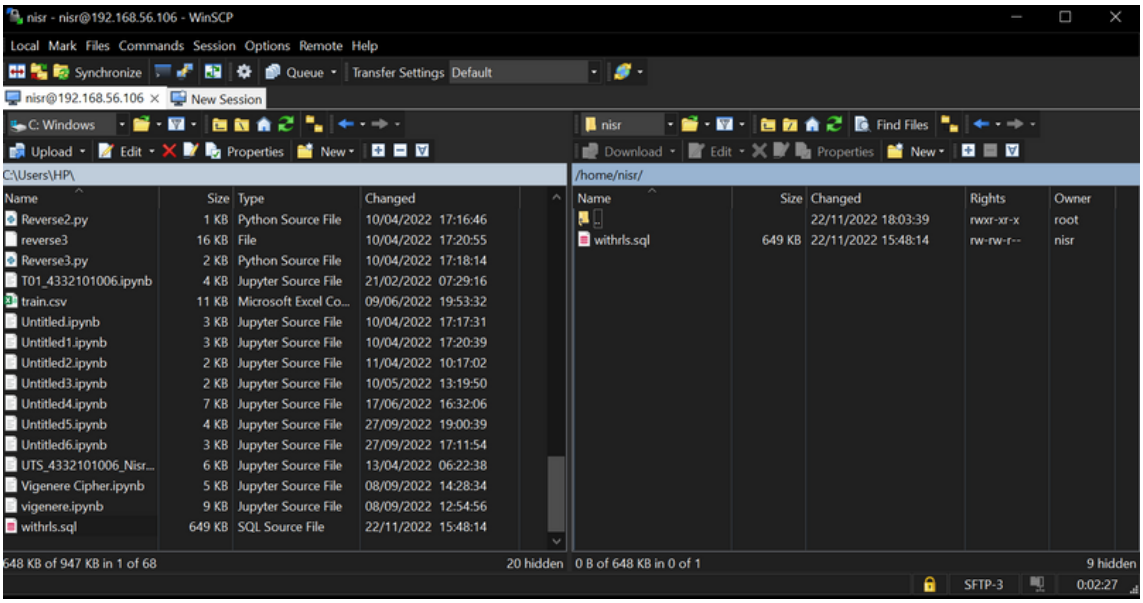
Buat database baru menggunakan perintah create database nama database;

```
postgres=# create database kambasdat;
```

Sebelumnya siapkan sebuah database yang terdiri dari minimal 5000 row sebagai bahan percobaan (menggunakan dummy data).



Pindahkan file tersebut dari windows ke ubuntu server menggunakan bantuan winscp.



file yang dipindahkan tadi ada berada pada home server, untuk memindahkannya ke postgres, dapat menggunakan perintah \i nama file dan untuk masuk ke database yang diinginkan dapat menggunakan perintah \c nama database.

```
postgres=# \c kambasdat
You are now connected to database "kambasdat" as user "postgres".
kambasdat=# \i withrls.sql
```

Gunakan perintah select count untuk melihat jumlah row dari file database tadi dan dapat menggunakan perintah select untuk melihat isi dari database.

```
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
kambasdat=# select count(*) from withrls;
 count
-------
  5000
(1 row)

kambasdat=# select * from withrls;
   first_name    |                email                |   ip_address    | kategory
-----------------+-------------------------------------+-----------------+----------
 Jacobo          | jdecleyne0@usgs.gov                 | 14.209.190.52   | red
 Leonore         | ltwycross1@unc.edu                  | 170.71.248.83   | green
 Dilly           | dpraten2@google.com                 | 218.95.86.59    | red
 Theresa         | tlayman3@epa.gov                    | 215.27.207.50   | yellow
 Lloyd           | lpetrina4@bbc.co.uk                 | 226.121.199.199 | yellow
 Grace           | ggrog5@acquirethisname.com          | 205.71.1.200    | red
 Jule            | jkennard6@zimbio.com                | 210.122.78.238  | yellow
 Hadrian         | heakle7@dailymotion.com             | 145.237.206.225 | green
```

Dapat juga melihat jumlah row berdasarkan kategori tertentu.

```
kambasdat=# select count(*) from withrls where kategory= 'red';
 count
-------
  1746
(1 row)

kambasdat=# select count(*) from withrls where kategory= 'green';
 count
-------
  1602
(1 row)

kambasdat=# select count(*) from withrls where kategory= 'yellow';
 count
-------
  1652
(1 row)
```

3. Membuat Privacy Policy

Pertama, dapat membuat role sesuai kategori yang akan dilakukan pengamanan.

```
kambasdat=# create role red;
CREATE ROLE
```

Buat policy user dengan perintah berikut. **user_view** digunakan sebagai penanda untuk membedakan role policy lain, sehingga namanya dapat diubah sesuai yang diinginkan. **current_user = kategori yang diinginkan** serta **for select** agar hanya bisa melakukan select.

```
kambasdat=# create policy user_view on withrls for select using (current_user = kategory);
CREATE POLICY
```

Berikan hak akses pada tabel tadi.

```
kambasdat=# GRANT SELECT (first_name, email, ip_address, kategory) ON withrls TO public;
GRANT
kambasdat=#
```

Enable role security dengan alter table serta set role yang telah diatur tadi. Lalu dapat memeriksa database yang tela di set, jika outputnya sesuai kategori yang ditentukan yaitu red, maka implementasi dari vpd sudah berhasil dilakukan.

```
kambasdat=# ALTER TABLE withrls ENABLE ROW LEVEL SECURITY;
ALTER TABLE
kambasdat=# SET ROLE red;
SET
kambasdat=> select * from withrls;
   first_name    |                email                |   ip_address    | kategory
-----------------+-------------------------------------+-----------------+----------
 Jacobo          | jdecleyne0@usgs.gov                 | 14.209.190.52   | red
 Dilly           | dpraten2@google.com                 | 218.95.86.59    | red
 Grace           | ggrog5@acquirethisname.com          | 205.71.1.200    | red
 Malissia        | mgobelf@360.cn                      | 40.211.25.233   | red
 Teresita        | tpalfriek@timesonline.co.uk         | 188.231.90.140  | red
 Pammi           | pmccanel@vkontakte.ru               | 214.211.116.247 | red
 Karry           | kcolthurstm@wired.com               | 63.7.28.44      | red
 Alvie           | apopplestonn@indiegogo.com          | 159.111.28.207  | red
 Jecho           | jpeaq@sciencedaily.com              | 64.178.128.226  | red
 Angelico        | adoersr@1und1.de                    | 0.149.18.119    | red
 Gizela          | gspatigu@reference.com              | 128.224.120.27  | red
 Ricky           | rcollynsv@sakura.ne.jp              | 208.195.217.185 | red
 Aggi            | awatkinsonx@tripadvisor.com         | 47.212.244.28   | red
 Collette        | cmckenney12@buzzfeed.com            | 216.44.203.60   | red
 Bobbie          | bpowis13@people.com.cn              | 239.93.155.176  | red
 Florentia       | fgrimsdale18@indiatimes.com         | 149.195.231.186 | red
```

Dapat membuat role baru dan policy user baru sesuai dengan kebutuhan.

```
nisrina=# create policy user_view on withrls for select using (current_user = kategory);
CREATE POLICY
nisrina=# GRANT SELECT (first_name, email, ip_address, kategory) ON withrls TO public;
GRANT
nisrina=# ALTER TABLE withrls ENABLE ROW LEVEL SECURITY;
ALTER TABLE
nisrina=# set role yellow;
SET
nisrina=> select * from withrls;
  first_name   |                email                |   ip_address    | kategory
---------------+-------------------------------------+-----------------+----------
 Theresa       | tlayman3@epa.gov                    | 215.27.207.50   | yellow
 Lloyd         | lpetrina4@bbc.co.uk                 | 226.121.199.199 | yellow
 Jule          | jkennard6@zimbio.com                | 210.122.78.238  | yellow
 Ruben         | rmold8@hexun.com                    | 60.154.157.255  | yellow
 Kynthia       | ksommerlingb@home.pl                | 101.195.109.5   | yellow
 Yancey        | yjerkec@ustream.tv                  | 125.210.167.243 | yellow
 Cary          | csponderh@umich.edu                 | 52.247.198.140  | yellow
 Bernarr       | banselmio@cnn.com                   | 234.107.167.159 | yellow
 Hedvige       | hdavionp@youtube.com                | 237.211.15.207  | yellow
 Debra         | decclestons@paginegialle.it         | 126.61.240.152  | yellow
 Jilli         | jobyrnez@ovh.net                    | 104.137.27.145  | yellow
 Humbert       | hgolthorpp10@nationalgeographic.com | 154.185.74.116  | yellow
 Robb          | rdevennie14@technorati.com          | 246.92.229.147  | yellow
 Leo           | lturbern15@vk.com                   | 87.249.72.25    | yellow
 Cathrin       | cpiesold16@youku.com                | 217.218.126.213 | yellow
 Mick          | mwitterick19@addtoany.com           | 239.142.237.221 | yellow
 Etty          | espondley1d@tripadvisor.com         | 5.219.134.200   | yellow
 Quinton       | qdumphreys1e@webmd.com              | 178.221.208.85  | yellow
 Ruthe         | rpeetermann1f@live.com              | 121.247.60.47   | yellow
 Norton        | nlocksley1i@walmart.com             | 171.90.46.78    | yellow
 Hy            | hocklandlk@lulu.com                 | 65.162.180.148  | yellow
```

B. PERFORMA VIRTUAL PRIVATE DATABASE

Pada bagian performa, akan mencoba membandingkan antara kecepatan akses dari database yang telah diterapkan dan yang tidak diterapkan VPD.

Kecepatan akses dari role yang telah diterapkan RLS:

```
nisrina=> \timing
Timing is on.
nisrina=> set role yellow;
SET
Time: 0.144 ms
nisrina=> select count (*) from withrls;
 count
-------
  1652
(1 row)

Time: 3.277 ms
```

Kecepatan akses dari role yang tidak diterapkan RLS:

```
nisrina=# alter table withrls disable row level security;
ALTER TABLE
nisrina=# \timing
Timing is on.
nisrina=# set role yellow;
SET
Time: 0.417 ms
nisrina=> select count (*) from withrls;
 count
-------
  5000
(1 row)

Time: 1.591 ms
```

Analisis:

Dari percobaan yang telah dilakukan, dapat disimpulkan bahwa role yang tidak diterapkan RLS dapat diakses lebih cepat daripada role yang diberikan privacy policy RLS. Hal ini disebabkan karena ketika diberikan privacy policy, maka saat diakses harus melewati sistem keamanan ini terlebih dahulu yang kemudian harus disesuaikan dulu dengan rules yang telah dibuat (dalam hal ini rules yang diterapkan adalah grant akses yang hanya dapat melihat beberapa kategori dan tidak dapat mengakses seluruh data yang ada pada database). Penyesuaian inilah yang membuat kecepatan akses pada data yang diterapkan RLS menjadi lebih lama.

Hal ini sesuai dengan hukum keamanan yang menyatakan bahwa keamanan akan berbanding terbalik dengan kenyamanan. Kenyamanan dalam hal ini yaitu mengorbankan kecepatan yang lebih lama dibandingkan ketika tidak diberikan proteksi tambahan.

PENUTUP

A. KESIMPULAN

Database sebagai layanan memiliki beberapa masalah dan perhatian utama, seperti keamanan data, kepercayaan, harapan, peraturan, dan kinerja. Pencegahan kehilangan data dan khususnya perlindungan data dari akses yang tidak sah tetap menjadi tujuan penting dari setiap sistem manajemen basis data. Pengguna yang memiliki hak istimewa pada tabel dasar dapat melewati penegakan keamanan yang disediakan oleh tampilan. Perhatikan bahwa ini adalah masalah umum dalam menyematkan keamanan dalam aplikasi alih-alih menegakkan keamanan melalui mekanisme database, tetapi diperparah saat keamanan diterapkan pada tampilan dan bukan pada data itu sendiri. VPD menggunakan KEBIJAKAN memberikan mekanisme yang fleksibel untuk membangun aplikasi yang menerapkan kebijakan keamanan hanya jika kontrol tersebut diperlukan dengan menambahkan pernyataan SQL secara dinamis dengan predikat, VPD membatasi akses ke data pada Tingkat Baris dan mengikat kebijakan keamanan ke tabel itu sendiri.

Pada praktikum kali ini, dilakukan implementasi berupa pemberian grant akses dengan kategori tertentu. Hal ini berhasil dilakukan dengan membuat beberapa role. Selain memberi grant access, dilakukan juga pengetesan performa dari role yang diterapkan RLS dengan yang tidak diterapkan RLS. Hasilnya adalah role yang diberikan RLS akan berjalan lebih lambat dibandingkan dengan role yang tidak diterapkan RLS. Hal ini sesuai dengan hukum keamanan yang berbanding terbalik dengan kenyamanan.

DAFTAR PUSTAKA

geeksforgeeks. 2022. Virtual Private Database (VPD) Diakses pada 12 November 2022. www.geeksforgeeks.org/virtual-private-database-vpd/

Gendron, François. 2004. Virtual Private Database Functionality, Diakses pada 12 November 2022. www.postgresql.org/message-id/001a01c48077%240b118e60%240200030a%40gendron.ca

Harauchi, Hajime dkk. 2002. Development of a Virtual Private Database for a Multi-institutional Internet-based Radiation Oncology Database Overcoming Differences in Protocols Vol.22. No.2.

Lakshmi, B. et. all. International Journal on Computer Science and Engineering (IJCSE) "Data Confidentiality and Loss Prevention using Virtual Private Database" Vol. 5 No. 03 Mar 2013.

Oracle Help Center. 7 Using Oracle Virtual Private Database to Control Data Access Diakses pada 12 November 2022. docs.oracle.com/cd/B28359_01/network.111/b28531/vpd.htm#DBSEG007

Oracle Technical White Paper. 2002. The Virtual Private Database in Oracle9iR2 Understanding Oracle9i Security for Service Providers.

Spendolini, S. (2013). Virtual Private Database. In: Expert Oracle Application Express Security. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4302-4732-6_12

Sugiantoro, Bambang dan Jazi Eko Istianto. 2010. Seminar Nasional Informatika 2010 (semnasIF 2010) "ANALISA KEAMANAN DATABASE SERVER MENGGUNAKAN TEKNOLOGI VIRTUAL PRIVATE DATABASE DAN NOTIFIKASI DATABASE SERVER MENGGUNAKAN AGENT BERGERAK".