LAPORAN PRAKTIKUM

PERTEMUAN KE : 6

KEAMANAN BASIS DATA

NAMA : Nauval Putra Widaya

NIM 4342411024

KELAS : TRPL 3A Malam

PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT

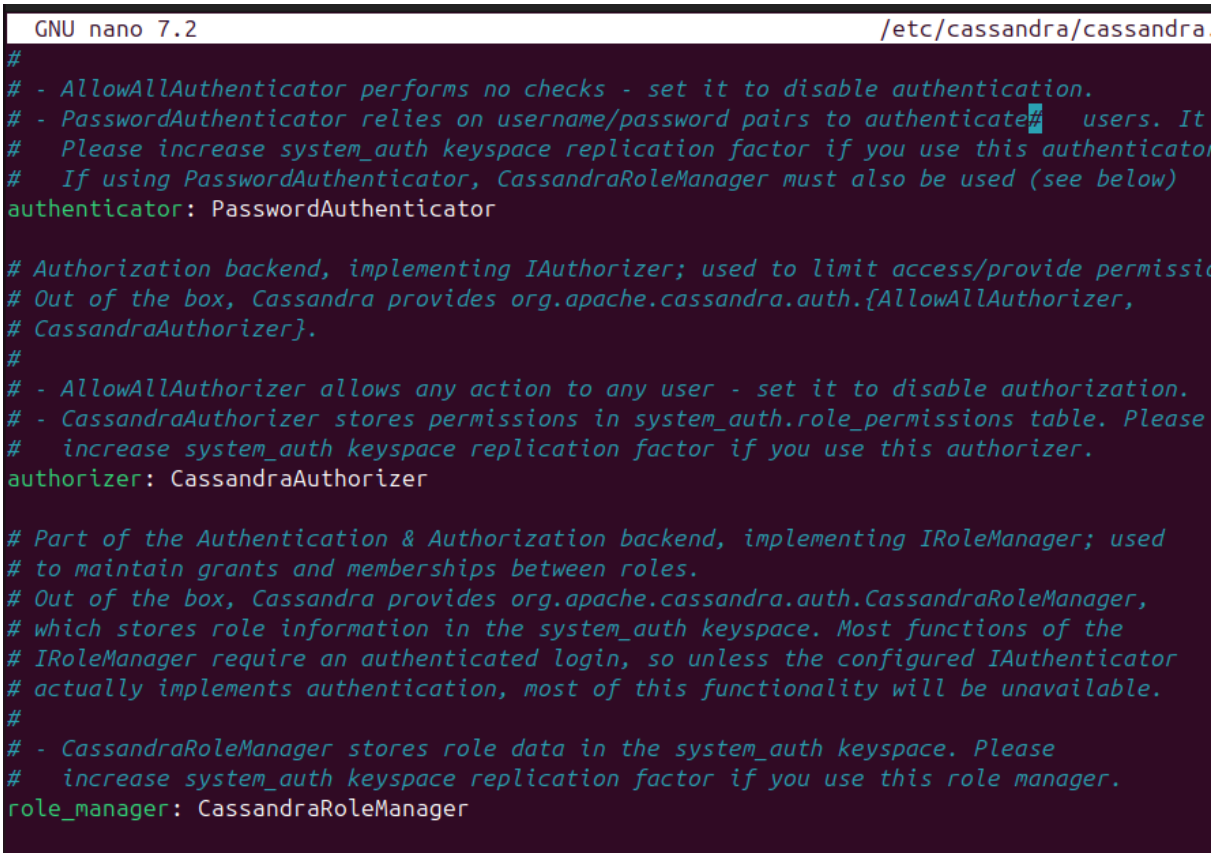
LUNAK JURUSAN TEKNIK INFORMATIKA

POLITEKNIK NEGERI

BATAM 2025

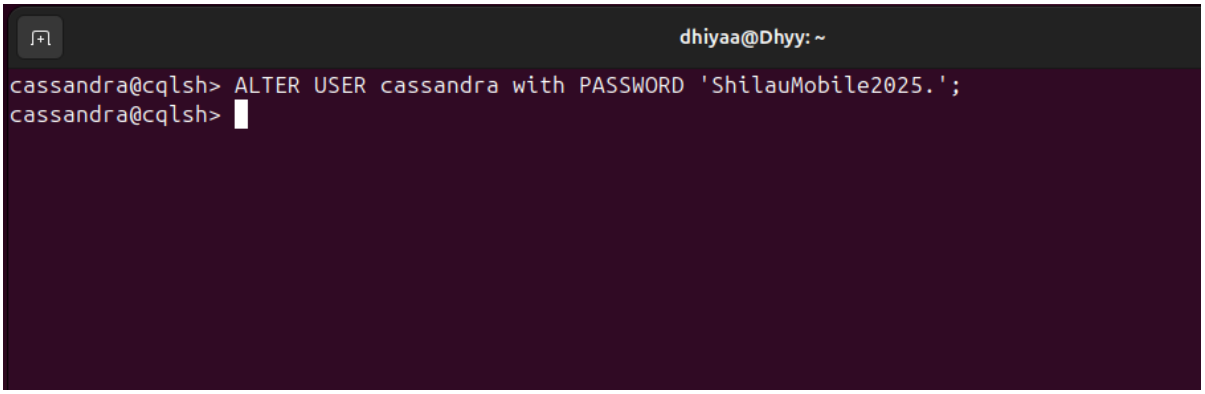
1. Konfigurasi Autentikasi dan Otorisasi

a. Melakukan konfigurasi pada Cassandra



Gambar 1. Melakukan beberapa konfigurasi pada Cassandra

b. Mengubah password default menjadi level 2 (strong).



Gambar 2. Mengubah password menjadi level 2

2. Implementasi role-based

a. Pada tahap ini dibuat user **admin_shilau** sebagai superuser agar dapat mengelola seluruh data dan konfigurasi sistem SHILAU. Selain itu, dibuat juga user psteam sebagai non-superuser yang hanya difokuskan untuk mengelola layanan tertentu sesuai tugasnya

```
cassandra@cqlsh> CREATE USER admin_shilau WITH PASSWORD 'AdminShilau2025.' SUPER
USER;
cassandra@cqlsh> CREATE USER psteam '
   ...
cassandra@cqlsh> CREATE USER psteam WITH PASSWORD 'Psteam2025.' NOSUPERUSER;
cassandra@cqlsh>
```

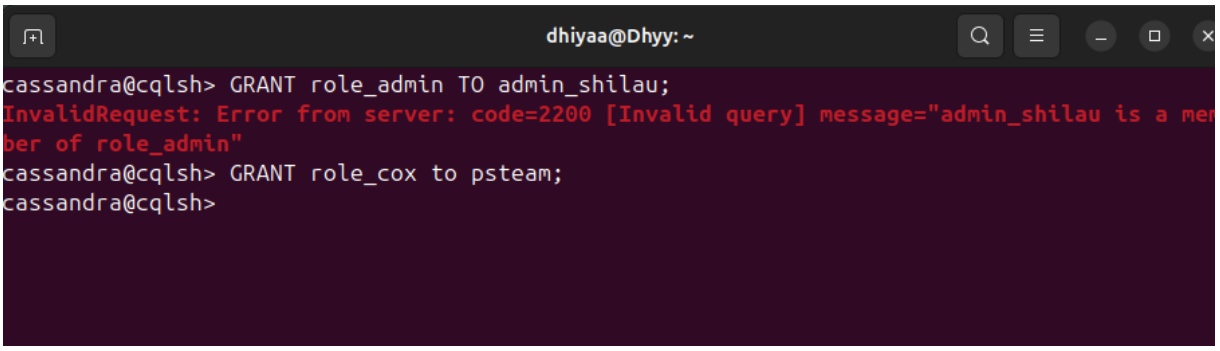
Gambar 3. Menambahkan user **admin_shilau** dan **psteam**

b. Dibuat dua role, yaitu **role_admin** sebagai superuser agar dapat mengelola seluruh aspek sistem SHILAU, serta **role_cox** sebagai non-superuser yang hanya diberikan hak login untuk mengakses data sesuai kebutuhan. Hal ini membedakan antara pengelola (admin) dan pengguna umum (klien).

```
cassandra@cqlsh> CREATE ROLE role_admin WITH SUPERUSER = true AND LOGIN = true;
cassandra@cqlsh> CREATE ROLE role_klien with LOGIN = true;
cassandra@cqlsh> DROP ROLE role_klien;
cassandra@cqlsh> CREATE ROLE role_cox WITH LOGIN = true;
cassandra@cqlsh>
```

Gambar 4. Membuat role_admin dan role_cox

c. Tahapan ini memberikan role kepada 2 user yang sudah dibuat sebelumnya.



Gambar 5. Menerapkan role ke 2 user yang dibuat

3. Enkripsi

a. Setup FQDN di `/etc/host`

```
dhiyaa@Dhyy:~$ echo "192.168.100.200 node1.shilau.local node1" | sudo tee -a /etc/hosts
[sudo] password for dhiyaa:
192.168.100.200 node1.shilau.local node1
dhiyaa@Dhyy:~$ ping -c2 node1.shilau.local
PING node1.shilau.local (192.168.100.200) 56(84) bytes of data.
64 bytes from node1.shilau.local (192.168.100.200): icmp_seq=1 ttl=64 time=0.563 ms
64 bytes from node1.shilau.local (192.168.100.200): icmp_seq=2 ttl=64 time=0.057 ms

--- node1.shilau.local ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1022ms
rtt min/avg/max/mdev = 0.057/0.310/0.563/0.253 ms
dhiyaa@Dhyy:~$
```

Gambar 5. Melakukan setup FQDN

Penjelasan:

Langkah ini memastikan node Cassandra memiliki identitas hostname (FQDN) yang jelas. FQDN akan dipakai untuk mencocokkan sertifikat SSL dengan alamat server sehingga proses verifikasi lebih aman.

b. Membuat root CA

```
dhiyaa@Dhyy:~$ sudo mkdir -p /etc/cassandra/ssl/ca
dhiyaa@Dhyy:~$ cd /etc/cassandra/ssl/ca
dhiyaa@Dhyy:/etc/cassandra/ssl/ca$ sudo openssl genrsa -out ca.key 4096
dhiyaa@Dhyy:/etc/cassandra/ssl/ca$ sudo openssl req -x509 -new -nodes -key ca.key -sha256 -da
ys 3650 \
  -subj "/CN=SHILAU-CA" -out ca.crt
dhiyaa@Dhyy:/etc/cassandra/ssl/ca$
```

Gambar 6. Membuat root CA

Penjelasan:

Root Certificate Authority (CA) dibuat sebagai pusat kepercayaan. Semua sertifikat node nantinya ditandatangani oleh CA ini agar bisa saling percaya.

c. Generate sertifikasi node

```
dhiyaa@Dhyy:/etc/cassandra/ssl/ca$ sudo mkdir -p /etc/cassandra/ssl/node1
dhiyaa@Dhyy:/etc/cassandra/ssl/ca$ cd /etc/cassandra/ssl/node1
dhiyaa@Dhyy:/etc/cassandra/ssl/node1$ sudo openssl genrsa -out node.key 2048
dhiyaa@Dhyy:/etc/cassandra/ssl/node1$ sudo openssl req -new -key node.key -out node.csr -subj
 "/CN=node1.shilau.local"
dhiyaa@Dhyy:/etc/cassandra/ssl/node1$ cat <<EOF | sudo tee san.ext
subjectAltName=DNS:node1.shilau.local,IP:192.168.100.200
extendedKeyUsage=serverAuth,clientAuth
EOF
subjectAltName=DNS:node1.shilau.local,IP:192.168.100.200
extendedKeyUsage=serverAuth,clientAuth
dhiyaa@Dhyy:/etc/cassandra/ssl/node1$ sudo openssl x509 -req -in node.csr \
  -CA /etc/cassandra/ssl/ca/ca.crt -CAkey /etc/cassandra/ssl/ca/ca.key -CAcreateserial \
  -out node.crt -days 825 -sha256 -extfile san.ext
Certificate request self-signature ok
subject=CN = node1.shilau.local
dhiyaa@Dhyy:/etc/cassandra/ssl/node1$
```

Gambar 7. Melakukan generate sertifikasi node

Penjelasan:

Sertifikat node berfungsi sebagai identitas server Cassandra. Dengan menambahkan FQDN dan IP di SAN, klien bisa memastikan mereka terhubung ke server yang benar

d. Buat Keystore dan Truststore

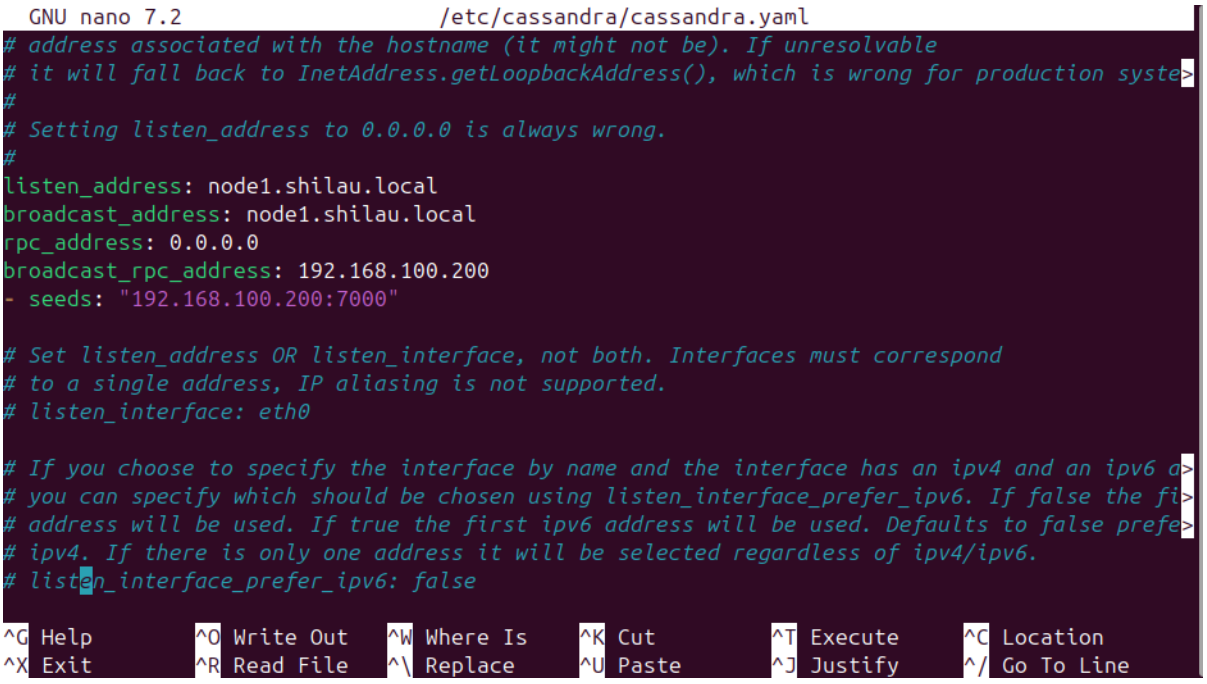
```
dhiyaa@Dhyy:/etc/cassandra/ssl/node1$ sudo openssl pkcs12 -export \
  -in node.crt -inkey node.key -certfile /etc/cassandra/ssl/ca/ca.crt \
  -name node1.shilau.local -out keystore.p12
Enter Export Password:
Verifying - Enter Export Password:
dhiyaa@Dhyy:/etc/cassandra/ssl/node1$ sudo keytool -importcert -noprompt -alias shilau-ca \
  -file /etc/cassandra/ssl/ca/ca.crt \
  -keystore truststore.p12 -storetype PKCS12 -storepass changeit
Certificate was added to keystore
dhiyaa@Dhyy:/etc/cassandra/ssl/node1$
```

Gambar 8. Membuat Keystore dan Truststore

Penjelasan:

Keystore menyimpan kunci privat dan sertifikat node. Truststore menyimpan CA yang dipercaya agar Cassandra dapat memverifikasi identitas pihak lain. Untuk kebutuhan pratikum percobaan ini menggunakan password default yang mudah diingat saja.

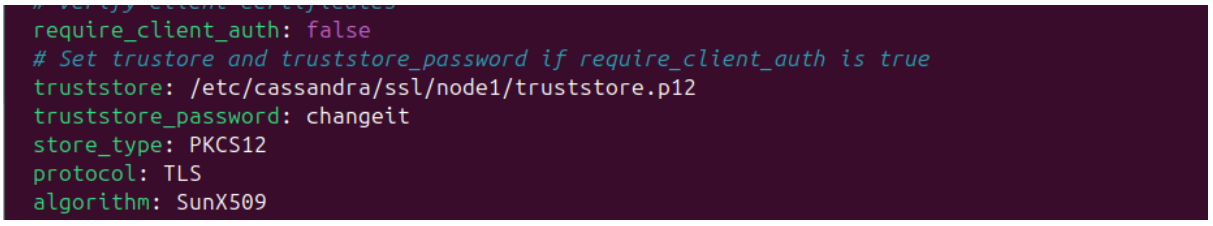
e. Konfigurasi cassandra alamat node



Gambar 9. Melakukan konfigurasi cassandra alamat node

Penjelasan:
Alamat ini harus sesuai dengan SAN sertifikat agar tidak terjadi error verifikasi. Dengan pengaturan ini, Cassandra akan mengenali dirinya menggunakan hostname dan IP yang sudah diproteksi SSL.

f. Aktifkan client ke node ssl



Gambar 10. Mengaktifan client ke node ssl

Penjelasan:
Opsi ini mengenkripsi semua komunikasi antara klien dan node Cassandra. Dengan begitu, data login dan query tidak lagi terbaca dalam bentuk teks biasa.

g. Pengujian dengan cqlsh

```
[ssl]
certfile = /etc/cassandra/ssl/ca/ca.crt
validate = true
version = TLSv1_2
```

Penjelasan :

File **`cqlshrc`** digunakan untuk mengarahkan klien **`cqlsh`** agar mempercayai sertifikat CA yang menandatangani sertifikat Cassandra. Dengan begitu, proses validasi SSL dapat berjalan tanpa error, memastikan komunikasi terenkripsi dengan server Cassandra.

h. Percobaan ssl

```
(venv-cassandra) dhiyaa@Dhyy:~$ cqlsh --ssl 127.0.0.1 9042 -u cassandra -p Shila
uMobile2025.

Warning: Using a password on the command line interface can be insecure.
Recommendation: use the credentials file to securely provide the password.

Warning: Explicit SSL and TLS versions in the cqlshrc file or in SSL_VERSION env
ironment property are ignored as the protocol is auto-negotiated.

WARNING: cqlsh was built against 5.0.0, but this server is 4.1.10.  All features
 may not work!
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.2.1 | Cassandra 4.1.10 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cassandra@cqlsh>
```

Penjelasan :

Percobaan SSL ini menunjukkan bahwa Cassandra berhasil dijalankan dengan enkripsi pada port 9042 dan dapat diakses menggunakan user yang sudah diatur. Hasil koneksi ke `cqlsh` menandakan autentikasi dan sertifikat SSL sudah berfungsi dengan benar. Dengan begitu, komunikasi client–server kini lebih aman karena data ditransmisikan melalui saluran terenkripsi.