**KONFIGURASI REPLIKASI MASTER-SLAVE UNTUK DATABASE MYSQL**

**SISTEM TERDISTRIBUSI**



Disusun Oleh :

**NAMA : DEWI PURNAMA**

**NIM : 09011182126020**

**JURUSAN : SISTEM KOMPUTER**

**DOSEN : ADI HERMANSYAH, S. KOM., M.T.**

**FAKULTAS ILMU KOMPUTER**

**PROGRAM STUDI SISTEM KOMPUTER**

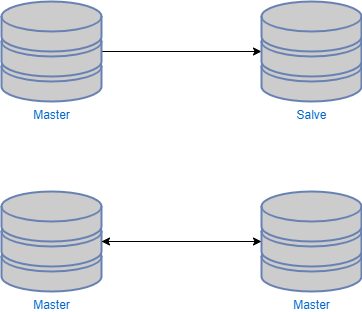
**UNIVERSITAS SRIWIJAYA**

**TAHUN AJARAN**

**2024**

**Jenis skema pada replikasi database ada 2 yaitu:**

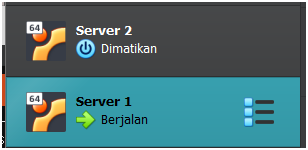
1. Master-Slave, hanya database master yang digunakan untuk operasi write, sedangkan operasi read dapat disebarkan di beberapa database slave. Biasa disebut dengan replikasi satu arah.
2. Master-Master, Skema ini memungkinkan semua server dapat digunakan untuk operasi write dan read. Replikasi ini biasa disebut dengan replikasi dua arah



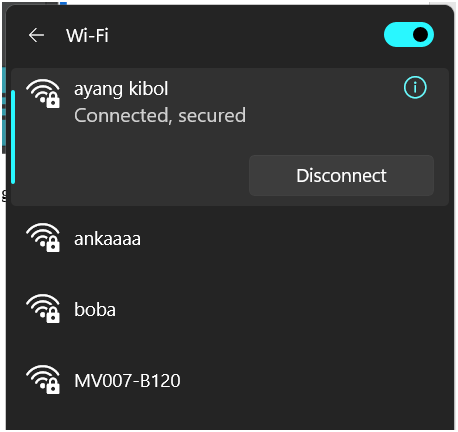
*Skema replikasi satu arah (master-slave), replikasi dua arah (master-master)*

**Yang perlu disiapkan :**

* 2 server ubuntu



* Koneksi internet untuk mengunduh package



**Disini ip dari server yang digunakan :**

* Server 1 (master) : 192.168.43.10
* Server 2 (slave) : 192.168.43.4

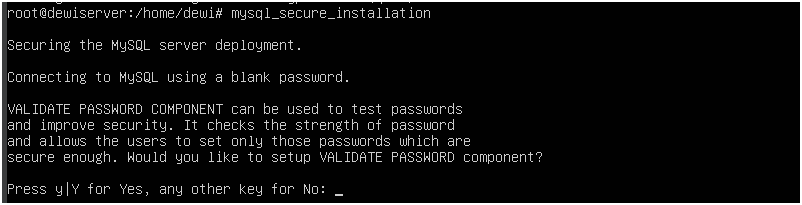
**Install MySQL Server pada Server 1**

Menginstall MySQL Server pada Ubuntu Server kita bisa menginstalnya menggunakan perintah apt / apt-get. sebelum menginstall MySQL lakukan update ubuntu package terlebih dahulu menggunakan perintah berikut :

apt install mysql-server -y

Setelah menginstal server MySQL, perlu mengatur kata sandi root MySQL di kedua server. Jalankan perintah berikut untuk mengatur kata sandi root:

mysql\_secure\_installation



Anda akan ditanyai hal berikut:

Press y|Y for Yes, any other key for No:

Cukup tekan **Enter** untuk mengatur kata sandi root:

Please set the password for root here.

New password:

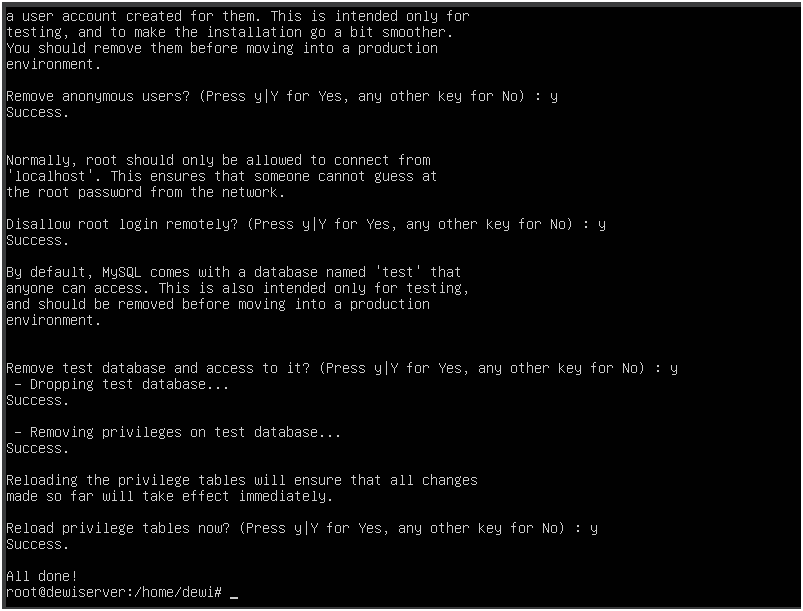
Re-enter new password:

Tetapkan kata sandi root MySQL Anda dan jawab pertanyaan selanjutnya seperti yang ditunjukkan di bawah ini:

Remove anonymous users? (Press y|Y for Yes, any other key for No) : Y  
Disallow root login remotely? (Press y|Y for Yes, any other key for No) : Y

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : Y

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : Y



**Konfigurasi Server Utama**

Pertama, Anda perlu mengedit file konfigurasi default MySQL di server Master untuk mengizinkan akses jarak jauh dan mengaktifkan log biner.

nano /etc/mysql/mysql.conf.d/mysqld.cnf

Tambahkan atau ubah baris berikut:

[mysqld]

pid-file = /var/run/mysqld/mysqld.pid

socket = /var/run/mysqld/mysqld.sock

datadir = /var/lib/mysql

bind-address = 0.0.0.0

log\_error = /var/log /mysql/error.log

server-id = 1

log\_bin = /var/log/mysql/mysql-bin.log

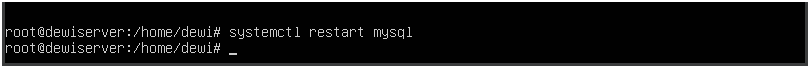
max\_binlog\_size = 500 juta

slow\_query\_log = 1



Simpan dan tutup file, lalu restart layanan MySQL untuk menerapkan perubahan.

systemctl restart mysql



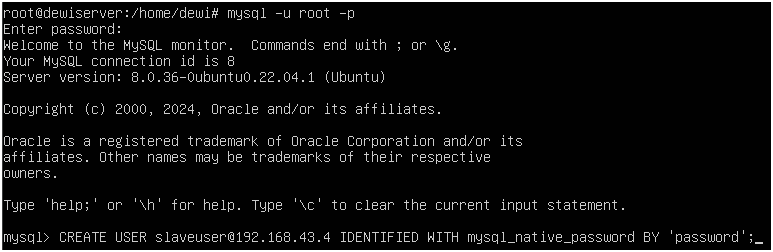
**Membuat Pengguna Replikasi di Server Master**

Selanjutnya, Anda perlu membuat pengguna replikasi di server Master untuk mengelola replikasi.Untuk melakukannya, sambungkan ke shell MySQL dengan perintah berikut:

mysql -u root -p

Berikan kata sandi root MySQL Anda, lalu jalankan perintah berikut untuk membuat pengguna replikasi:

mysql> BUAT USER slaveuser@192.168.43.4 DIIDENTIFIKASI DENGAN mysql\_native\_password DENGAN 'password';



Selanjutnya, berikan hak istimewa REPLICATION SLAVE kepada pengguna replikasi:

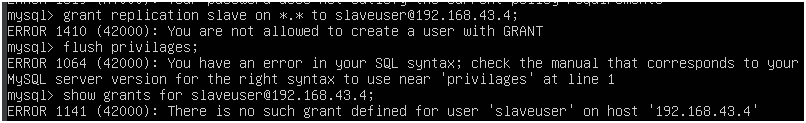
mysql> grant replication slave on \*.\* to slaveuser@192.168.43.4;

Selanjutnya, hapus hak istimewa untuk menerapkan perubahan:

mysql> flush privileges;

Selanjutnya, periksa hak istimewanya menggunakan perintah berikut:

mysql> show grants for [slaveuser@192.168.43.4](mailto:slaveuser@192.168.43.4);



Contoh keluaran:

+------------------------------------------------- ------------+   
| Grants slaveuser@192.168.43.4|   
+------------------------------------------------- ------------+   
| Grant REPLICATION SLAVE ON \*.\* TO `slaveuser`@`192.168.43.4` |   
+------------------------------------------------- ------------+

Selanjutnya, keluar dari shell MySQL dengan perintah berikut:

mysql> exit;



**Catatan** : Ganti **192.168.43.4** dengan alamat IP server **Slave** .

**Konfigurasikan Server Slave**

Selanjutnya, Anda perlu mengedit file konfigurasi utama MySQL dan membuat beberapa perubahan:

nano /etc/mysql/mysql.conf.d/mysqld.cnf

Tambahkan atau ubah baris berikut:

file pid = /var/run/mysqld/mysqld.pid   
soket = /var/run/mysqld/mysqld.sock   
bind-address = 0.0.0.0   
datadir = /var/lib/mysql   
log\_bin = /var/log/mysql/ mysql-bin.log   
server-id = 2   
read\_only = 1   
max\_binlog\_size = 500 juta   
slow\_query\_log = 1



Simpan dan tutup file, lalu restart layanan MySQL untuk menerapkan perubahan:

systemctl restart mysql

Inisialisasi Replikasi pada Server Slave

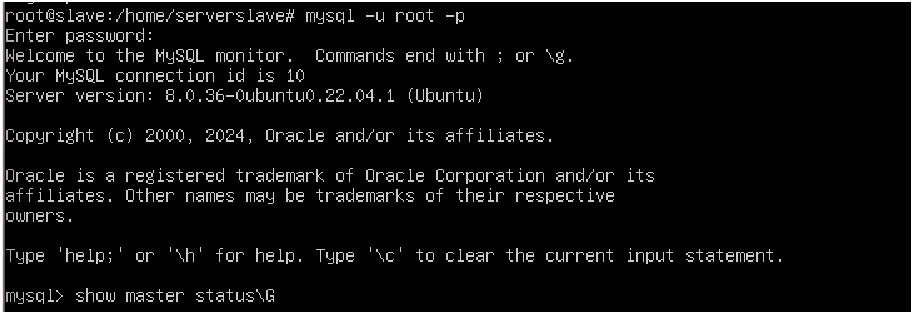
Selanjutnya, Anda perlu memulai proses Replikasi di server budak.

Pertama, sambungkan ke shell MySQL di server Master dengan perintah berikut:

mysql -u root -p

Selanjutnya cek status Master dengan perintah berikut:

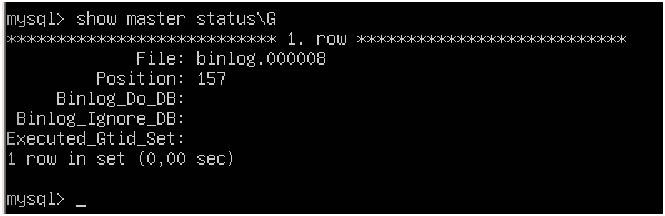
mysql> show master status\G



Contoh keluaran:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. baris

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*   
 File: mysql-bin.000002   
 Posisi: 1047   
 Binlog\_Do\_DB:   
Binlog\_Ignore\_DB:   
Executed\_Gtid\_Set:   
1 baris dalam set (0,00 detik)



Dari output di atas, catat file log master dan nomor posisinya

Selanjutnya, buka server Slave dan sambungkan ke shell MySQL:

mysql -u root -p

Selanjutnya, gunakan informasi yang diperoleh dari server Master dan konfigurasikan server Slave dengan perintah berikut:

mysql> GANTI MASTER KE MASTER\_HOST='192.168.43.10', MASTER\_USER='slaveuser', MASTER\_PASSWORD='password', MASTER\_LOG\_FILE='mysql-bin.000002', MASTER\_LOG\_POS=1047;

Selanjutnya, jalankan Slave menggunakan perintah berikut:

**Catatan** : ganti IP **192.168.43.10** dengan alamat IP server **Master** .

mysql> start slave;

Selanjutnya verifikasi status Slave dengan perintah berikut:

mysql> show slave status\G

akan mendapatkan hasil berikut:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. baris

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*   
 Slave\_IO\_State: Menunggu master mengirim acara   
 Master\_Host: 45.58.41.25   
 Master\_User: slaveuser   
 Master\_Port: 3306   
 Connect\_Retry: 60   
 Master\_Log\_File: mysql-bin.000002   
 Read\_Master\_Log\_Pos: 1047   
 Relay\_Log\_File: slave-relay-bin.000002   
 Relay\_Log\_Pos: 324   
 Relay\_Master\_Log\_File: mysql-bin.000002   
 Slave\_IO\_Running: Ya   
 Slave\_SQL\_Running: Ya

**Verifikasi Replikasi Master-Slave**

Pada titik ini, replikasi MySQL Master-Slave dikonfigurasi. Sekarang saatnya menguji apakah replikasi berfungsi atau tidak.

Pertama, masuk ke server Master dan buat database dengan perintah berikut:

mysql -u root -p   
mysql> membuat replika database;

Selanjutnya, verifikasi semua database menggunakan perintah berikut:

mysql> show databases;

Contoh keluaran:

+--------------------+   
| Basis Data |   
+--------------------+   
| informasi\_skema |   
| mysql |   
| skema\_kinerja |   
| replikadb |   
| sy |   
+------+

Selanjutnya, keluar dari shell MySQL:

mysql> exit;

Selanjutnya, buka server Slave dan sambungkan ke MySQL

mysql -u root -p

Selanjutnya, daftar semua database menggunakan perintah berikut:

mysql> show databases;

Contoh keluaran:

+--------------------+   
| Basis Data |   
+--------------------+   
| informasi\_skema |   
| mysql |   
| skema\_kinerja |   
| replikadb |   
| sy |   
+------+

Output di atas menunjukkan bahwa replikasi MySQL Master-Slave berfungsi seperti yang diharapkan karena database **'replicadb'** direplikasi di server 2.

:

1. Persiapan Data:
   * Bagi data menjadi data pelatihan dan data pengujian.
   * Gated Recurrent Unit (GRU) adalah jenis arsitektur jaringan saraf rekurensi (Recurrent Neural Network, RNN) yang digunakan dalam tugas klasifikasi untuk memproses urutan data dan menghasilkan prediksi kelas atau label. GRU dirancang dengan unit gating yang memungkinkan pengendalian aliran informasi dalam model, sehingga GRU dapat mengingat dan melupakan informasi yang relevan atau tidak relevan dalam urutan data.
   * Dalam konteks klasifikasi, GRU dapat digunakan untuk mengklasifikasikan berbagai jenis data urutan, seperti teks, waktu, atau sinyal suara. GRU memiliki kemampuan untuk memahami ketergantungan jarak jauh dalam urutan data, sehingga dapat mengenali pola dan fitur yang penting dalam tugas klasifikasi.Pastikan data telah diproses dan diubah menjadi format yang dapat diterima oleh arsitektur yang akan dievaluasi. Misalnya, jika menggunakan GRU, pastikan data dalam bentuk urutan yang sesuai.
2. Pembentukan Model:
   * Pilih arsitektur yang akan dievaluasi, seperti GRU.
   * Tentukan hyperparameter arsitektur, seperti jumlah unit GRU, ukuran embedding, lapisan dropout, atau lapisan Dense untuk output.
   * Inisialisasi model dengan arsitektur yang ditentukan.
3. Pelatihan Model:
   * Gunakan data pelatihan untuk melatih model dengan menggunakan metode optimisasi seperti SGD atau Adam.
   * Tetapkan fungsi kerugian (loss function) yang sesuai, seperti categorical cross-entropy untuk klasifikasi banyak kelas.
   * Latih model dengan melakukan iterasi melalui data pelatihan dan memperbarui bobot model sesuai dengan fungsi kerugian.
   * Ulangi langkah ini sampai model mencapai konvergensi atau batasan yang ditetapkan sebelumnya.
4. Evaluasi Model:
   * Gunakan data pengujian yang telah dipisahkan sebelumnya untuk menguji kinerja model yang dilatih.
   * Hitung prediksi model untuk data pengujian.
   * Bandingkan prediksi model dengan label sebenarnya pada data pengujian.
5. Metrik Evaluasi:  
   Ada berbagai metrik evaluasi yang dapat digunakan untuk mengevaluasi kinerja model klasifikasi. Berikut adalah beberapa metrik evaluasi yang umum digunakan:
   * Akurasi (Accuracy): Mengukur persentase keberhasilan prediksi yang benar dari semua sampel.  
     Rumus: Akurasi = (Jumlah prediksi yang benar) / (Jumlah total sampel)
   * Presisi (Precision): Mengukur sejauh mana prediksi positif yang dilakukan oleh model merupakan prediksi yang benar.  
     Rumus: Presisi = (True Positives) / (True Positives + False Positives)
   * Recall (Recall atau Sensitivity): Mengukur sejauh mana model dapat mengidentifikasi sampel positif dengan benar.  
     Rumus: Recall = (True Positives) / (True Positives + False Negatives)
   * F1-Score: Menggabungkan presisi dan recall menjadi satu skor yang mencerminkan keseimbangan antara keduanya.  
     Rumus: F1-Score = 2 \* (Presisi \* Recall) / (Presisi + Recall)
   * Matriks Kebingungan (Confusion Matrix): Merupakan tabel yang menunjukkan jumlah prediksi yang tepat dan salah untuk setiap kelas. Berikut adalah contoh matriks kebingungan:

|  | **Prediksi Negatif** | **Prediksi Positif** |
| --- | --- | --- |
| Aktual Negatif | True Negative | False Positive |
| Aktual Positif | False Negative | True Positive |

1. Terdapat juga metrik lain seperti Area Under ROC Curve (AUC-ROC) atau Area Under Precision-Recall Curve (AUC-PR) yang juga dapat digunakan tergantung pada kebutuhan dan sifat masalah klasifikasi.
2. Validasi Silang (Opsional):
   * Jika memungkinkan, gunakan validasi silang (cross-validation) untuk mendapatkan perkiraan yang lebih stabil tentang kinerja model.
   * Dalam validasi silang, data Anda dibagi menjadi beberapa lipatan (folds), dan model dilatih dan dievaluasi pada kombinasi lipatan yang berbeda.