

## MODUL 5

### DATABASE SERVER

---

#### **5.1 Topik Pembahasan**

1. Pengertian database server dan perannya.
2. Jenis-jenis database server.
3. Arsitektur database di *cloud*.
4. Manajemen database server di *cloud*.
5. Keunggulan dan tantangan database server.

#### **5.2 Tujuan Praktikum**

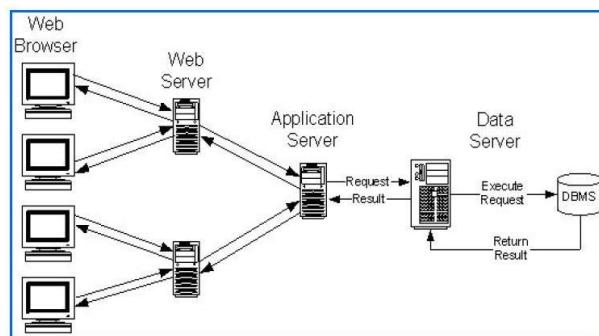
1. Memahami konsep dasar dan fungsi database server.
2. Mengetahui jenis-jenis database server serta perbedaannya.
3. Menjelaskan arsitektur dan komponen database di *cloud computing*.
4. Memahami aspek keamanan, backup, dan monitoring database.
5. Mengenali keunggulan dan tantangan penggunaan database server di *cloud*.

#### **5.3 Alat dan Bahan**

1. Laptop
2. Browser dengan koneksi internet

#### **5.4 Dasar Teori**

##### **5.4.1 Pengertian Database Server**

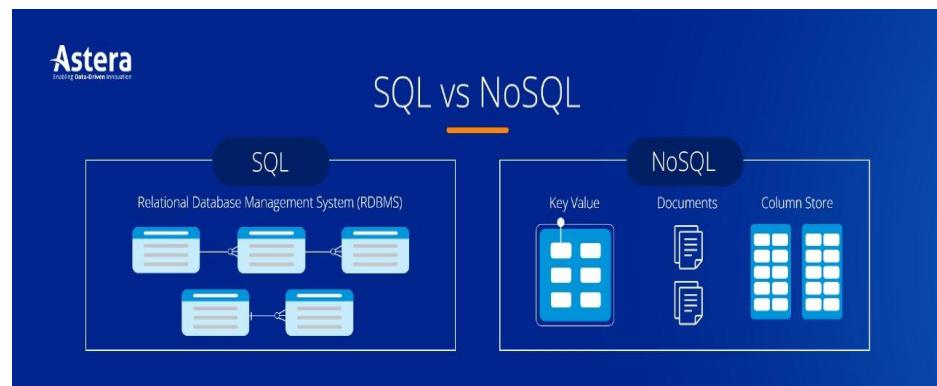


Gambar 1 Arsitektur *Client–Server* pada Database Server

Database server adalah sistem yang bertugas untuk menyimpan, mengelola, dan

menyediakan data kepada aplikasi atau pengguna lain melalui jaringan. Server ini menjadi pusat pengolahan data dan memungkinkan banyak *client* mengakses informasi secara bersamaan. Dalam lingkungan *cloud computing*, database server tidak lagi dijalankan di komputer lokal, melainkan di infrastruktur *cloud* yang menawarkan ketersediaan tinggi, fleksibilitas, dan skalabilitas. Model yang digunakan adalah *client-server*, dimana *client* mengirimkan perintah (*query*) ke server dan menerima hasilnya dalam bentuk data yang telah diproses.

#### 5.4.2 Jenis-Jenis Database Server



Gambar 2 Perbandingan RDBMS dan NoSQL Database

Database server secara umum dibagi menjadi dua kategori utama, yaitu *Relational Database Server* (RDBMS) dan *Non-Relational Database Server* (NoSQL).

a. *Relational Database Server* (RDBMS)

Jenis ini menyimpan data dalam bentuk tabel dan menggunakan bahasa SQL (*Structured Query Server*). Contohnya: MySQL, PostgreSQL, MariaDB, dan Microsoft SQL Server. Kelebihan RDBMS antara lain integritas data yang tinggi, dukungan transaksi, dan kemudahan dalam melakukan analisis terstruktur.

b. *Non-Relational Database Server* (NoSQL)

Database ini tidak menggunakan struktur tabel, melainkan dokumen, *key-value*, *graph*, atau *column-based*. Contohnya: MongoDB, Cassandra, dan DynamoDB. NoSQL cocok untuk data yang besar dan tidak terstruktur, serta mendukung skalabilitas horizontal yang tinggi.

#### 5.4.3 Arsitektur Database Server di *Cloud*

Dalam *Cloud Computing*, database dapat dijalankan melalui dua pendekatan

utama:

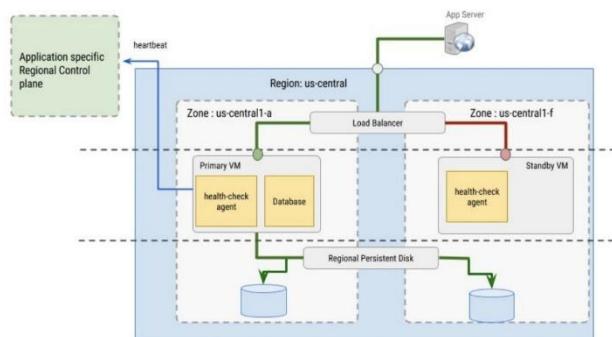
### 1. *Self-Managed Database*

Pengguna menginstal dan mengkonfigurasi database secara mandiri di mesin virtual. Seluruh tanggung jawab terkait update, patching, dan backup berada di tangan pengguna.

### 2. *Managed Database Service (DBaaS)*

Layanan seperti Amazon RDS atau Google Cloud SQL mengelola infrastruktur secara otomatis. Pengguna hanya mengatur parameter database dan data yang disimpan.

#### 5.4.4 Manajemen Database Server di *Cloud*



Gambar 3 Konsep *High Availability* dan Replikasi Database di *Cloud*

Manajemen database di *cloud* mencakup aspek penting seperti keamanan, backup, monitoring, dan skalabilitas.

- Keamanan: Gunakan password kuat, batasi IP akses, dan aktifkan enkripsi data.
- *Backup* dan *Recovery*: Gunakan *automated backup* dan *point-in-time* agar data dapat dipulihkan dengan mudah.
- *Monitoring*: Gunakan layanan seperti Amazon *CloudWatch* untuk memantau performa database dan aktivitas query.
- *Scability & High Availability*: Terapkan *replication* dan *multi-AZ deployment* agar database tetap aktif saat terjadi gangguan di salah satu zona.

#### 5.4.5 Keunggulan dan Tantangan Database di *Cloud*

Penggunaan database di *cloud* memberikan banyak keuntungan:

- Skalabilitas tinggi sesuai kebutuhan.

- Efisiensi biaya dengan model *pay-as-you-go*.
- *Backup* dan pemeliharaan otomatis.
- Keaman data yang terintegrasi.

Namun, terdapat pula beberapa tantangan:

- Ketergantungan terhadap penyedia layanan (*vendor lock-in*).
- Potensi biaya meningkat bila tidak dikontrol.
- Perlu konfigurasi keamanan yang hati-hati.

## LANGKAH PRAKTIKUM

---

1. Pilih **Start Lab**, tunggu sampai logo AWS dikiri atas berwarna hijau.
2. Jika sudah hijau, klik logo AWS tersebut.
3. Pada kotak pencarian, ketik dan pilih **VPC**.
4. Di panel navigasi sebelah kiri, pilih **Security Group**.
5. Klik tombol **Create Security Group**.
6. Kemudian atur Security group name: “DB Security Group”, Description: “Permit access from Web Security Group”, VPC: (Lab VPC).
7. Pada bagian **inbound rules**, klik add rule.
8. Atur type: **MySQL/Aurora**, source: pada sebelah kanan custom ketik sg kemudian pilih **Web Security Group**.
9. Kemudian klik tombol **Create Security Group**.
10. Pada kotak pencarian atas ketik RDS dan pilih **Aurora and RDS**.
11. Pada panel navigasi kiri pilih **Subnet Groups**, lalu klik **Create DB Subnet Group**.
12. Isi name dengan “**DB-Subnet-Group**”, description dengan “**DB Subnet Group**”, VPC “**Lab VPC**”.
13. Pada section Add subnets, expand availability zones, lalu pilih **us-east-1a** dan **us-east-1b**.
14. Expand subnets dan pilih CIDR **10.0.1.0/24** dan **10.0.3.0/24**, kemudian create.
15. Pada panel navigasi kiri pilih **Databases**, lalu pilih **Create Database**.
16. Pada engine options pilih **MySQL**.
17. Pada section template pilih **Dev-Test**.
18. Pada section Availability and durability pilih **Multi-AZ DB instance**.
19. Pada section settings, DB instance identifier: “lab-db”, Master username: “main”, Credentials management: “Self managed”, Master password: “lab-password”, Confirm password: “lab-password”.
20. Pada Instance configuration, pilih opsi **Burstable classes (includes t classes)** dan pilih **db.t3.micro**.
21. Pada section Storage pilih Storage type: General Purpose (SSD), Allocated storage: 20.
22. Pada section connectivity di Virtual private cloud (VPC) pilih **Lab VPC**.
23. Pada Existing VPC security groups hapus deafult, lalu pilih **DB Security Group** saja.
24. Pada section Monitoring expand Additional configuration, lalu di bagian Enhanced Monitoring uncheck **Enable Enhanced monitoring**.

25. Expand Additional configuration, isi Initial database name: “lab”, Uncheck **Enable automatic backups**, Uncheck **Enable encryption**, kemudian Create Database.
26. Tunggu hingga status database menjadi available atau modifying.
27. Setelah itu klik lab-db, lalu pada bagian Connectivity & security salin **endpoint** dan simpan.
28. Pada instruksi langkah-langkah terdapat **Aws Details** disamping kanan end lab, klik tersebut.
29. Salin ip address web server.
30. Buka tab browser baru dan paste ip address.
31. Buka tab **RDS**, masukkan data endpoint: “(yang telah disalin sebelumnya)”, Database: “lab”, username: “main”, password: “lab-password”, tekan **submit**.
32. Akan muncul pesan pesan bahwa aplikasi sedang mengirim data ke database. Setelah selesai, akan tampil aplikasi **Address Book**.
33. Website telah terhubung ke RDS MySQL, dan semua data tersimpan ke dalam database RDS.
34. **Submit** pekerjaan yang telah dilakukan, pilih **Yes**.
35. Tunggu beberapa menit hingga nilai keluar.
36. Jika sudah jangan lupa **End Lab**, pilih **Yes**.
37. Selesai.