

LAPORAN PRAKTIKUM TEKNOLOGI CLOUD COMPUTING

**MIGRASI DATABASE KE CLOUD
MEMANFAATKAN CLOUD STORAGE DAN CLOUD SQL
IF-E**



Disusun oleh

Nama : Deni Norman Zaky

NIM : 123220165

**PROGRAM STUDI INFORMATIKA
JURUSAN INFORMATIKA
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”
YOGYAKARTA
2025**

HALAMAN PENGESAHAN

LAPORAN PRAKTIKUM

MIGRASI DATABASE KE CLOUD

MEMANFAATKAN CLOUD STORAGE DAN CLOUD SQL

Disusun Oleh :

Deni Norman Zaky 123220165

Telah diperiksa dan disetujui oleh Asisten Praktikum.....

Pada tanggal :

Menyetujui.

Asisten Praktikum

Asisten Praktikum

Berlyandhica Alam Febriwantoro
NIM 123210060

Rafli Iskandar Kavarera
NIM 123210131

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan tugas praktikum Teknologi Cloud Computing dengan judul “Migrasi Database ke Cloud Memanfaatkan Cloud Storage dan Cloud SQL”. Laporan ini disusun sebagai bentuk dokumentasi dari hasil pembelajaran yang telah kami lakukan selama praktikum berlangsung.

Kami menyampaikan terima kasih yang sebesar-besarnya kepada Bapak Dosen, yang telah memberikan bimbingan serta ilmu yang sangat berharga selama proses pembelajaran. Tak lupa, kami juga mengucapkan terima kasih kepada Berlyandhica Alam Febriwantoro dan Rafli Iskandar Kavarera selaku asisten praktikum yang telah dengan sabar membimbing serta membantu kami dalam memahami materi dan menyusun laporan ini.

Kami menyadari bahwa laporan ini masih jauh dari kesempurnaan. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun demi penyempurnaan laporan ini di masa mendatang.

Kami menyadari bahwa laporan ini masih jauh dari kesempurnaan. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun demi penyempurnaan laporan ini di masa mendatang.

Yogyakarta, 10 Maret 2025

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
BAB I PENDAHULUAN.....	1
1.1 <i>Latar Belakang.....</i>	<i>1</i>
1.2 <i>Rumusan Masalah.....</i>	<i>2</i>
1.3 <i>Tujuan</i>	<i>2</i>
1.4 <i>Manfaat</i>	<i>2</i>
BAB II TINJAUAN LITERATUR.....	3
2.1 <i>Google Cloud Platform (GCP)</i>	<i>3</i>
2.2 <i>Cloud Storage</i>	<i>4</i>
2.3 <i>Cloud SQL.....</i>	<i>6</i>
BAB III METODOLOGI	8
3.1 <i>Analisis Permasalahan</i>	<i>8</i>
3.2 <i>Perancangan Solusi</i>	<i>9</i>
4.1.1 <i>Pembuatan Instance Cloud SQL di GCP</i>	<i>9</i>
4.1.2 <i>Pembuatan Bucket di Cloud Storage</i>	<i>14</i>
4.1.3 <i>Pembuatan Database di Instance</i>	<i>14</i>
4.1.4 <i>Ekspor Database</i>	<i>15</i>
4.1.5 <i>Pengunggahan File SQL ke Cloud Storage.....</i>	<i>16</i>
4.1.6 <i>Impor Database ke Cloude SQL</i>	<i>16</i>

4.1.7	Konfigurasi Koneksi ke Cloude SQL	17
4.1.8	Pengujian dan Validasi.....	18
BAB IV HASIL DAN PEMBAHASAN		19
4.1	<i>Hasil</i>	19
4.1.1	Pembuatan instance Cloud SQL di GCP.....	19
4.1.2	Pembuatan bucket di Cloud Storage	19
4.1.3	Pembuatan database di instance	20
4.1.4	Ekspor database dari sistem lokal ke format SQL	20
4.1.5	Pengunggahan file SQL ke Cloud Storage.....	21
4.1.6	Impor database ke Cloud SQL	21
4.1.7	Konfigurasi koneksi ke Cloud SQL	22
4.1.8	Pengujian dan validasi.....	22
4.2	<i>Pembahasan</i>	24
BAB V PENUTUP		26
5.1	<i>Kesimpulan</i>	26
5.2	<i>Saran</i>	26
DAFTAR PUSTAKA		28
LAMPIRAN.....		29
6.1	<i>Link Github</i>	29

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital yang semakin berkembang, kebutuhan akan pengelolaan data yang efisien, aman, dan skalabel menjadi tantangan utama bagi banyak perusahaan dan organisasi. Sistem penyimpanan dan pengelolaan database tradisional sering kali menghadapi berbagai kendala, seperti keterbatasan kapasitas penyimpanan, biaya operasional yang tinggi, serta risiko kehilangan data akibat kegagalan perangkat keras. Selain itu, banyak perusahaan mengalami kesulitan dalam mengakses dan mengelola data mereka secara fleksibel, terutama ketika skala bisnis mereka berkembang. Oleh karena itu, diperlukan solusi yang dapat mengatasi berbagai tantangan tersebut agar sistem database dapat berjalan secara optimal dan mendukung kebutuhan operasional secara efektif.

Untuk menjawab tantangan tersebut, teknologi komputasi awan (cloud computing) menawarkan solusi yang lebih fleksibel, efisien, dan aman dalam pengelolaan database. Salah satu solusi utama yang dapat diterapkan adalah migrasi database ke cloud menggunakan layanan Cloud Storage dan Cloud SQL. Dengan pendekatan ini, data dapat disimpan dan dikelola secara terpusat, memungkinkan aksesibilitas yang lebih baik serta meningkatkan efisiensi dalam proses backup dan pemulihan data. Google Cloud Platform (GCP) sebagai salah satu penyedia layanan cloud menawarkan berbagai fitur yang mendukung migrasi database, seperti Cloud Storage untuk penyimpanan data dalam jumlah besar dan Cloud SQL untuk pengelolaan database relasional yang terintegrasi dengan sistem cloud.

Pemanfaatan teknologi cloud, khususnya layanan Cloud Storage dan Cloud SQL di GCP, merupakan solusi yang tepat karena menawarkan skalabilitas yang tinggi, keamanan data yang lebih baik, serta kemudahan dalam integrasi dan manajemen sistem database. Dengan menggunakan teknologi ini, organisasi dapat mengurangi beban biaya operasional, meningkatkan efisiensi pengelolaan data,

serta memastikan ketersediaan dan keandalan sistem dalam jangka panjang. Oleh karena itu, implementasi migrasi database ke cloud menjadi langkah strategis yang relevan dalam mendukung kebutuhan transformasi digital di berbagai sektor industri.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, beberapa permasalahan yang dapat dirumuskan adalah:

1. Bagaimana proses migrasi database dari sistem lokal ke lingkungan cloud menggunakan Google Cloud Platform (GCP)?
2. Apa saja kendala yang dapat muncul selama proses migrasi database ke cloud, dan bagaimana cara mengatasinya?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah dijelaskan, tujuan dari laporan praktikum ini adalah sebagai berikut:

1. Menjelaskan proses migrasi database dari sistem lokal ke lingkungan cloud menggunakan Google Cloud Platform (GCP) sebagai solusi dalam meningkatkan efisiensi dan skalabilitas sistem.
2. Mengidentifikasi kendala yang dapat muncul selama proses migrasi database ke cloud serta menyusun strategi untuk mengatasinya guna memastikan kelancaran dan keberlanjutan sistem yang dimigrasikan.

1.4 Manfaat

Berdasarkan latar belakang, rumusan masalah, dan tujuan yang telah dijelaskan, laporan praktikum ini memiliki tiga manfaat utama sebagai berikut:

1. Meningkatkan efisiensi dan skalabilitas pengelolaan database dengan migrasi ke Google Cloud Platform (GCP), memungkinkan penyesuaian kapasitas penyimpanan dan sumber daya sesuai kebutuhan.
2. Mengidentifikasi kendala migrasi database ke cloud serta menyusun strategi untuk mengatasinya guna memastikan proses transisi berjalan lancar.

BAB II

TINJAUAN LITERATUR

2.1 Google Cloud Platform (GCP)

Google Cloud Platform (GCP) merupakan salah satu penyedia layanan cloud terkemuka yang menawarkan berbagai solusi untuk pengelolaan data dan aplikasi. GCP menyediakan infrastruktur yang kuat dan fleksibel, memungkinkan organisasi untuk membangun, mengelola, dan mengembangkan aplikasi dengan lebih efisien. Salah satu fitur unggulan dari GCP adalah kemampuannya dalam menyediakan layanan database yang terintegrasi, seperti Cloud SQL dan Cloud Firestore, yang dirancang untuk memenuhi kebutuhan pengelolaan data yang skalabel dan aman. Dengan menggunakan GCP, perusahaan dapat memanfaatkan teknologi mutakhir yang mendukung analisis data, machine learning, dan pengembangan aplikasi berbasis cloud, sehingga meningkatkan daya saing di era digital (Google Cloud, 2023).

Dalam konteks migrasi database, GCP menawarkan berbagai alat dan layanan yang memudahkan proses transisi dari sistem lokal ke cloud. Layanan seperti Database Migration Service memungkinkan pengguna untuk melakukan migrasi database secara otomatis dan minim gangguan, sehingga meminimalkan risiko kehilangan data dan downtime. Selain itu, GCP juga menyediakan dokumentasi yang komprehensif dan panduan langkah demi langkah untuk membantu pengguna dalam merencanakan dan melaksanakan migrasi database dengan sukses. Dengan pendekatan ini, organisasi dapat memastikan bahwa data

mereka tetap aman dan dapat diakses dengan mudah setelah migrasi (Google Cloud, 2023).

Keamanan data menjadi salah satu prioritas utama dalam pengelolaan database di GCP. Platform ini dilengkapi dengan berbagai fitur keamanan, termasuk enkripsi data saat transit dan saat disimpan, serta kontrol akses yang ketat untuk melindungi informasi sensitif. GCP juga mematuhi berbagai standar dan regulasi keamanan internasional, sehingga memberikan kepercayaan kepada organisasi dalam mengelola data mereka di cloud. Dengan demikian, GCP tidak hanya menawarkan solusi yang efisien dan skalabel, tetapi juga menjamin keamanan dan integritas data, yang sangat penting dalam mendukung transformasi digital di berbagai sektor industri (Google Cloud, 2023).

2.2 Cloud Storage

Cloud Storage adalah layanan penyimpanan data yang disediakan oleh penyedia layanan cloud, termasuk Google Cloud Platform (GCP), yang memungkinkan pengguna untuk menyimpan dan mengelola data secara terpusat di internet. Layanan ini menawarkan berbagai keuntungan, seperti skalabilitas yang tinggi, di mana pengguna dapat dengan mudah menambah atau mengurangi kapasitas penyimpanan sesuai dengan kebutuhan bisnis mereka. Selain itu, Cloud Storage juga memberikan kemudahan dalam akses data dari berbagai lokasi dan perangkat, sehingga mendukung kolaborasi yang lebih baik di antara tim dan meningkatkan efisiensi operasional. Dengan fitur-fitur seperti penyimpanan objek, penyimpanan file, dan penyimpanan blok, Cloud Storage dapat memenuhi berbagai kebutuhan penyimpanan data yang berbeda, mulai dari file kecil hingga data besar yang tidak terstruktur (Google Cloud, 2023).

Salah satu keunggulan utama dari Cloud Storage adalah tingkat keamanan yang ditawarkannya. Data yang disimpan di Cloud Storage dilindungi dengan enkripsi baik saat transit maupun saat disimpan, sehingga mengurangi risiko kebocoran data. Selain itu, GCP mematuhi berbagai standar keamanan dan kepatuhan internasional, memberikan kepercayaan kepada organisasi untuk menyimpan data sensitif mereka di cloud. Fitur kontrol akses yang ketat juga memungkinkan pengguna untuk mengatur siapa yang dapat mengakses data tertentu, sehingga meningkatkan keamanan dan integritas data. Dengan demikian, Cloud Storage tidak hanya menawarkan solusi penyimpanan yang efisien, tetapi juga menjamin perlindungan data yang lebih baik (Google Cloud, 2023).

Dalam konteks migrasi database, Cloud Storage berperan penting sebagai tempat penyimpanan data yang dapat diakses oleh layanan database lainnya, seperti Cloud SQL. Proses migrasi database ke cloud dapat dilakukan dengan lebih lancar ketika data telah disimpan di Cloud Storage, karena pengguna dapat dengan mudah memindahkan data ke dalam database relasional yang dikelola oleh Cloud SQL. Selain itu, Cloud Storage juga mendukung proses backup dan pemulihan data yang lebih efisien, sehingga organisasi dapat memastikan ketersediaan data mereka dalam jangka panjang. Dengan memanfaatkan Cloud Storage, perusahaan dapat mengoptimalkan pengelolaan data mereka dan mendukung transformasi digital yang sedang berlangsung (Google Cloud, 2023).

2.3 Cloud SQL

Cloud SQL adalah layanan database terkelola yang disediakan oleh Google Cloud Platform (GCP), yang memungkinkan pengguna untuk menjalankan dan mengelola database relasional seperti MySQL, PostgreSQL, dan SQL Server di lingkungan cloud. Layanan ini dirancang untuk menyederhanakan proses pengelolaan database, termasuk pengaturan, pemeliharaan, dan pengoptimalan kinerja. Dengan Cloud SQL, organisasi dapat fokus pada pengembangan aplikasi dan analisis data tanpa harus khawatir tentang infrastruktur yang mendasarinya. Cloud SQL juga menawarkan fitur-fitur seperti replikasi otomatis, backup terjadwal, dan pemulihan bencana, yang semuanya berkontribusi pada peningkatan ketersediaan dan keandalan sistem database. Dengan demikian, Cloud SQL menjadi solusi yang ideal bagi perusahaan yang ingin memanfaatkan keunggulan cloud tanpa mengorbankan performa dan keamanan data (Google Cloud, 2023).

Salah satu keuntungan utama dari Cloud SQL adalah kemampuannya untuk terintegrasi dengan layanan lain di GCP, seperti Cloud Storage dan BigQuery. Integrasi ini memungkinkan pengguna untuk memindahkan data dengan mudah antara berbagai layanan, mendukung analisis data yang lebih mendalam dan pengambilan keputusan yang lebih baik. Selain itu, Cloud SQL juga mendukung skala yang fleksibel, memungkinkan pengguna untuk menyesuaikan kapasitas database sesuai dengan kebutuhan bisnis yang berubah. Dengan fitur keamanan yang kuat, termasuk enkripsi data dan kontrol akses yang ketat, Cloud SQL memberikan jaminan bahwa data sensitif tetap terlindungi. Oleh karena itu, penggunaan Cloud SQL dalam migrasi database ke cloud tidak hanya

meningkatkan efisiensi operasional, tetapi juga memastikan bahwa data dikelola dengan aman dan efektif (Google Cloud, 2023).

BAB III

METODOLOGI

3.1 Analisis Permasalahan

Dalam penugasan praktikum ini, fokus utama adalah pada proses migrasi database dari sistem lokal ke lingkungan cloud menggunakan Google Cloud Platform (GCP). Proses migrasi ini melibatkan beberapa langkah teknis yang harus dilakukan dengan hati-hati untuk memastikan bahwa data yang dipindahkan tetap utuh, aman, dan dapat diakses dengan efisien setelah migrasi. Tujuan dari penugasan ini adalah untuk memahami dan menerapkan teknik migrasi yang tepat, serta mengidentifikasi dan mengatasi kendala yang mungkin muncul selama proses tersebut.

Berdasarkan identifikasi tersebut, terdapat beberapa tantangan teknis spesifik dalam proses migrasi database ke GCP:

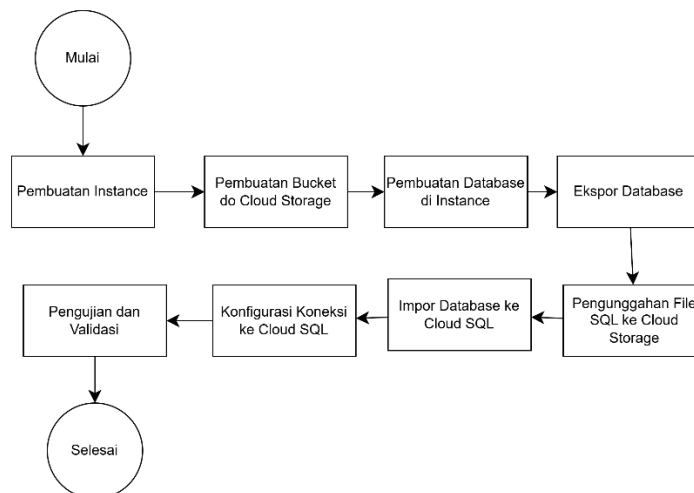
- a. Database perlu diekspor dalam format yang kompatibel dengan Cloud SQL, seperti SQL dump atau CSV, dengan tetap mempertahankan integritas data dan struktur relasional.
- b. Data yang telah diekspor memerlukan penyimpanan sementara di Cloud Storage dengan memperhatikan aspek keamanan melalui pengaturan hak akses yang tepat dan penerapan enkripsi.
- c. Proses impor dari Cloud Storage ke Cloud SQL harus memperhatikan kompatibilitas skema, kecepatan impor, serta antisipasi kesalahan yang mungkin terjadi.
- d. Selama proses migrasi berlangsung, integritas data harus tetap terjaga tanpa mengalami kehilangan atau perubahan yang tidak diinginkan, dengan menerapkan mekanisme enkripsi, autentikasi yang ketat, dan backup otomatis.
- e. Konfigurasi koneksi dari aplikasi ke database yang telah dimigrasikan memerlukan penyesuaian parameter koneksi agar aplikasi dapat

berkomunikasi dengan database cloud tanpa perubahan signifikan pada kode aplikasi.

Untuk mengatasi tantangan tersebut, diperlukan pendekatan migrasi yang sistematis dengan perencanaan yang matang, termasuk pengujian pra dan pasca migrasi, serta pemantauan performa secara real-time untuk memastikan keberhasilan proses migrasi dan optimalisasi sistem database di lingkungan cloud.

3.2 Perancangan Solusi

Untuk mengatasi permasalahan yang telah diidentifikasi, solusi yang dirancang dalam migrasi database dari lingkungan lokal ke Google Cloud Platform mencakup serangkaian langkah terstruktur. Pendekatan ini dirancang untuk memastikan proses migrasi berjalan lancar dengan meminimalkan risiko kehilangan data dan gangguan layanan.



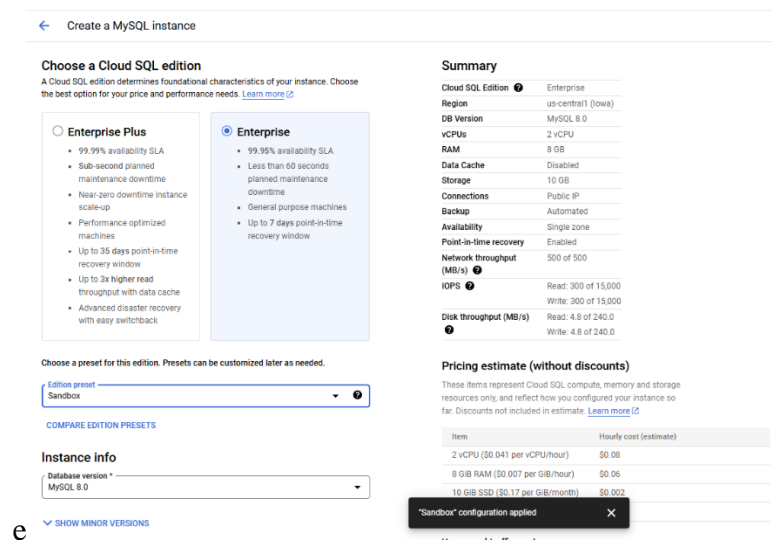
Gambar 3.1 Alur Kerja

4.1.1 Pembuatan Instance Cloud SQL di GCP

Instance Cloud SQL berfungsi sebagai basis data terkelola yang memungkinkan pengguna untuk menjalankan dan mengelola database SQL di cloud tanpa harus khawatir tentang infrastruktur fisik. Dengan menggunakan instance Cloud SQL, pengguna dapat fokus pada

pengembangan aplikasi tanpa harus mengelola server, patching, atau pemeliharaan database. Pembuatan instance ini sangat penting, terutama bagi pengembang dan perusahaan yang membutuhkan solusi database yang scalable, aman, dan mudah diakses. Oleh karena itu, saya memutuskan untuk membuat instance Cloud SQL untuk mendukung aplikasi yang sedang dikembangkan.

Pada langkah pertama, saya memilih SQL edition yang akan digunakan untuk instance Cloud SQL. Dalam hal ini, saya memilih Enterprise edition. Pilihan ini memberikan akses ke fitur-fitur premium yang ditawarkan oleh Google Cloud SQL, seperti kemampuan untuk melakukan pemulihan bencana, replikasi, dan pengelolaan yang lebih baik untuk aplikasi berskala besar.



Gambar 3.1 Memilih SQL edition

Setelah memilih SQL edition, langkah berikutnya adalah membuat instance ID dan password. Saya memberikan instance ID dengan nama deni-123220165. Pemilihan ID ini penting untuk identifikasi instance di

dalam proyek Google Cloud. Selain itu, saya juga menetapkan password yang kuat untuk memastikan keamanan akses ke database yang akan dibuat.

Create a MySQL instance

- Sub-second planned maintenance downtime
- Near-zero downtime instance scale-up
- Performance optimized machines
- Up to 35 days point-in-time recovery window
- Up to 3x higher read throughput with data cache
- Advanced disaster recovery with easy switchback
- Less than 60 seconds planned maintenance downtime
- General purpose machines
- Up to 7 days point-in-time recovery window

Choose a preset for this edition. Presets can be customized later as needed.

Edition preset: **Sandbox**

COMPARE EDITION PRESETS

Instance info

Database version: **MySQL 8.0**

Instance ID: **demo-123220165**

Use lowercase letters, numbers, and hyphens. Start with a letter.

Password: ********* **GENERATE**

Set a password for the root user. [Learn more](#)

☐ No password

Summary

Cloud SQL Edition	Enterprise
Region	us-central1 (Iowa)
DB Version	MySQL 8.0
vCPUs	2 vCPU
RAM	8 GB
Data Cache	Disabled
Storage	10 GB
Connections	Public IP
Backup	Automated
Availability	Single zone
Point-in-time recovery	Enabled
Network throughput (Mbps)	500 of 500
IOPS	Read: 300 of 15,000 Write: 300 of 15,000
Disk throughput (MB/s)	Read: 4.8 of 240.0 Write: 4.8 of 240.0

Pricing estimate (without discounts)

These items represent Cloud SQL compute, memory and storage resources only, and reflect how you configured your instance so far. Discounts not included in estimate. [Learn more](#)

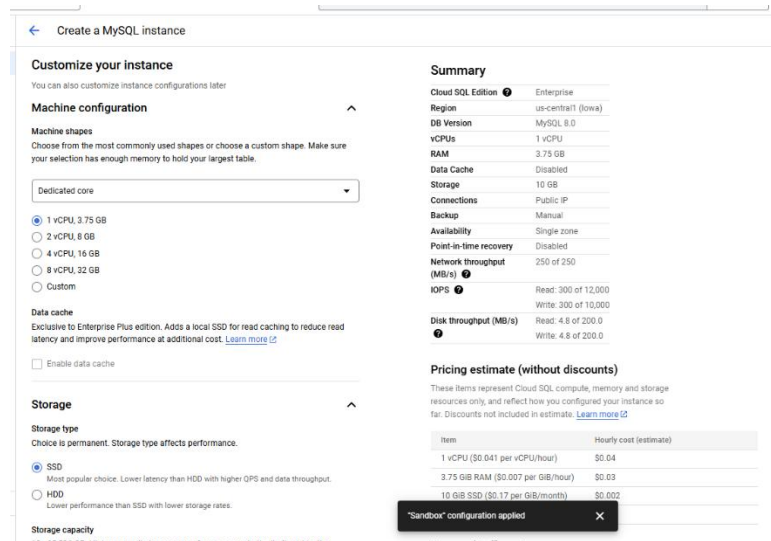
Item	Hourly cost (estimated)
2 vCPU (\$0.041 per vCPU/hour)	\$0.08
8 GB RAM (\$0.007 per GB/hour)	\$0.06
10 GB SSD (\$0.17 per GB/month)	\$0.003

"Sandbox" configuration applied

[Home](#) [View traffic logs](#)

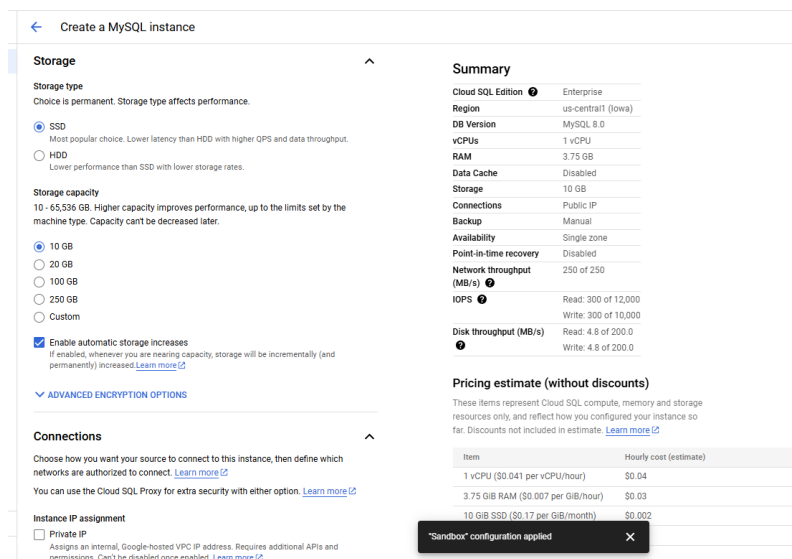
Gambar 3.2 Membuat instance ID dan password

Pada langkah ini, saya mengatur konfigurasi mesin untuk instance yang akan dibuat. Saya memilih konfigurasi 1 vCPU dengan 3.75 GB RAM. Konfigurasi ini dipilih berdasarkan kebutuhan aplikasi yang akan dijalankan, di mana saya memperkirakan bahwa spesifikasi ini cukup untuk menangani beban kerja yang diharapkan.



Gambar 3.3 Mengatur machine configuration

Pada langkah ini, saya mengatur konfigurasi mesin untuk instance yang akan dibuat. Saya memilih konfigurasi 1 vCPU dengan 3.75 GB RAM. Konfigurasi ini dipilih berdasarkan kebutuhan aplikasi yang akan dijalankan, di mana saya memperkirakan bahwa spesifikasi ini cukup untuk menangani beban kerja yang diharapkan.



Gambar 3.4 Mengatur besarnya storage

Pada langkah ini, saya mengatur konfigurasi mesin untuk instance yang akan dibuat. Saya memilih konfigurasi 1 vCPU dengan 3.75 GB RAM. Konfigurasi ini dipilih berdasarkan kebutuhan aplikasi yang akan dijalankan, di mana saya memperkirakan bahwa spesifikasi ini cukup untuk menangani beban kerja yang diharapkan.

The screenshot displays the 'Instance IP assignment' section of the Google Cloud Platform console. It features two radio button options: 'Private IP' (unchecked) and 'Public IP' (checked). Below these options, there is a section titled 'Authorized networks' with a warning icon and text stating: 'Allowing 0.0.0.0/0 opens your instance to all IPv4 clients, including those unintended. Use narrower IP ranges or Cloud SQL Auth Proxy for better security. If you must allow 0.0.0.0/0, ensure strong passwords.' Below this warning, there is a 'New network' section with a dropdown menu for 'Name' (showing 'allow-all-publicIP') and a dropdown menu for 'Network *' (showing '0.0.0.0/0'). A link 'Use CIDR notation' is also present. At the bottom, an example IP range '199.27.25.0/24' is provided.

Instance IP assignment

☐ Private IP
Assigns an internal, Google-hosted VPC IP address. Requires additional APIs and permissions. Can't be disabled once enabled. [Learn more](#)

☒ Public IP
Assigns an external, internet-accessible IP address. Requires using an authorized network or the Cloud SQL Proxy to connect to this instance. [Learn more](#)

Authorized networks
You can specify CIDR ranges to allow IP addresses in those ranges to access your instance. [Learn more](#)

⚠️ Allowing 0.0.0.0/0 opens your instance to all IPv4 clients, including those unintended. Use narrower IP ranges or Cloud SQL Auth Proxy for better security. If you must allow 0.0.0.0/0, ensure strong passwords. [Learn more](#)

New network

Name
allow-all-publicIP

Use [CIDR notation](#)

Network *
0.0.0.0/0

Example: 199.27.25.0/24

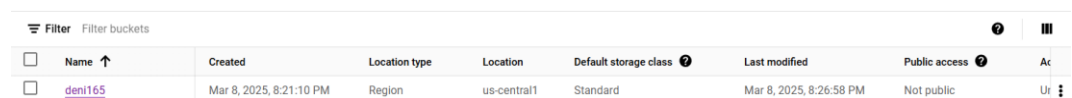
Gambar 3.5 Mengatur connections

Pada langkah terakhir, saya mengatur koneksi untuk instance Cloud SQL. Saya menambahkan network dengan nama allow-all-publicIP dan mengatur IP range menjadi 0.0.0.0/0. Pengaturan ini memungkinkan akses dari semua alamat IP publik, yang berguna untuk pengembangan dan pengujian. Namun, perlu diingat bahwa pengaturan ini harus dilakukan dengan hati-hati untuk menghindari potensi risiko keamanan.

4.1.2 Pembuatan Bucket di Cloud Storage

Pembuatan bucket di Google Cloud Storage (GCS) merupakan langkah penting dalam proses migrasi data dan pengelolaan file di cloud. Bucket berfungsi sebagai wadah untuk menyimpan objek, seperti file, gambar, video, dan data lainnya. Dalam konteks migrasi, bucket dapat digunakan untuk menyimpan data yang akan dipindahkan ke Cloud SQL atau untuk menyimpan cadangan (backup) dari database yang ada.

Berikut merupakan bucket yang telah dibuat dengan nama “deni165”. Bucket tersebut nanti digunakan untuk menyimpan file SQL yang akan dipindahkan ke GCP.



Filter buckets							
<input type="checkbox"/>	Name ↑	Created	Location type	Location	Default storage class	Last modified	Public access
<input type="checkbox"/>	deni165	Mar 8, 2025, 8:21:10 PM	Region	us-central1	Standard	Mar 8, 2025, 8:26:58 PM	Not public

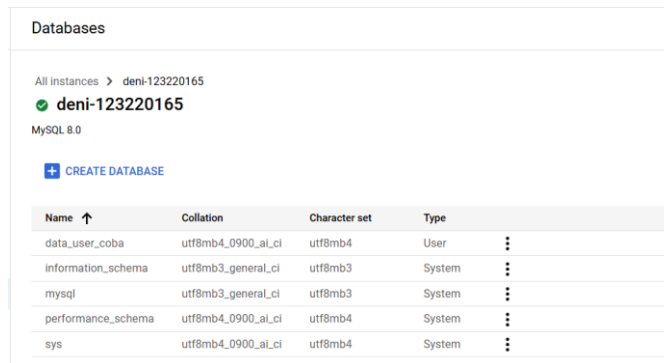
Gambar 3.6 Bucket

4.1.3 Pembuatan Database di Instance

Setelah instance Cloud SQL siap digunakan, langkah selanjutnya adalah membuat database di dalam instance tersebut. Database ini akan berfungsi sebagai tempat penyimpanan data yang akan dimigrasikan dari sistem lokal. Proses ini meliputi penamaan database dan pengaturan dasar lainnya.

Untuk membuat database di instance, pertama-tama masuk ke menu SQL, lalu pilih opsi Database dan klik Create Database. Selanjutnya, masukkan nama database sesuai dengan yang ada di server lokal (localhost).

Terakhir, klik Create Database untuk menyelesaikan proses pembuatan database di instance Cloud SQL.

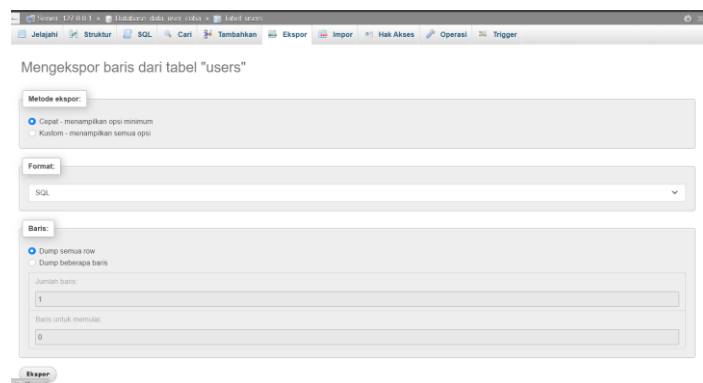


Name ↑	Collation	Character set	Type
data_user_coba	utf8mb4_0900_ai_ci	utf8mb4	User
information_schema	utf8mb3_general_ci	utf8mb3	System
mysql	utf8mb3_general_ci	utf8mb3	System
performance_schema	utf8mb4_0900_ai_ci	utf8mb4	System
sys	utf8mb4_0900_ai_ci	utf8mb4	System

Gambar 3.7 Database

4.1.4 Ekspor Database

Sebelum melakukan migrasi data, database yang ada di server lokal harus diekspor ke dalam format SQL. Proses ini dapat dilakukan dengan menggunakan alat seperti phpMyAdmin atau melalui perintah terminal yang sesuai dengan sistem manajemen database yang digunakan. File SQL yang dihasilkan akan mencakup struktur database beserta data yang terdapat di dalamnya.



Mengekspor baris dari tabel "users"

Metode ekspor:

☒ Cepat - menampilkan opsi minimum
☐ Kustom - menampilkan semua opsi

Format:

SQL

Baris:

☒ Dump semua row
☐ Dump beberapa baris

Jumlah baris:

1

Baris untuk memulai:

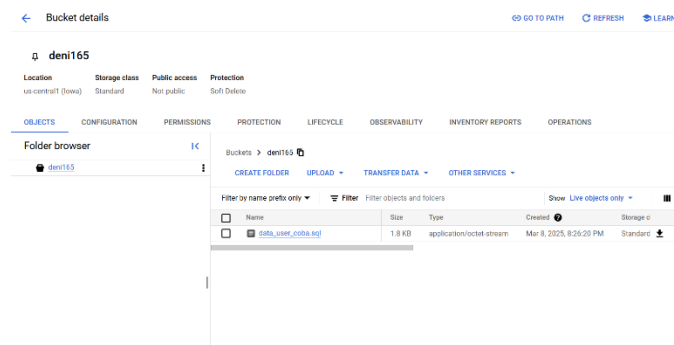
0

Ekspor

Gambar 3.8 Mengekspor Database SQL dari PhpMyAdmin

4.1.5 Pengunggahan File SQL ke Cloud Storage

Setelah proses ekspor dari PhpMyAdmin berhasil, file SQL kemudian diunggah ke ke bucket Cloud Storage yang telah dibuat sebelumnya.

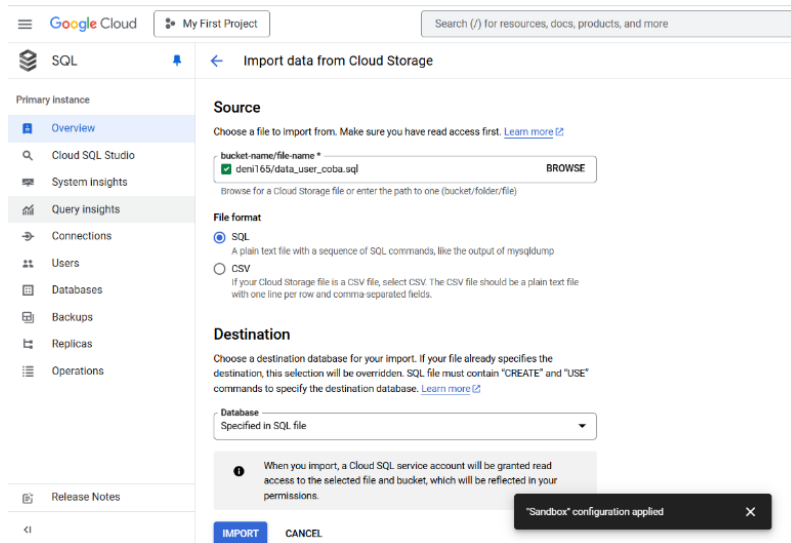


Gambar 3.9 Pengunggahan file SQL ke Cloud Storage

4.1.6 Impor Database ke Cloude SQL

Setelah file SQL disimpan di Cloud Storage, langkah selanjutnya adalah mengimpornya ke database yang telah dibuat di Cloud SQL. Proses ini memastikan bahwa semua data dari server lokal berhasil dipindahkan ke lingkungan cloud sambil mempertahankan integritas dan struktur aslinya.

Langkah-langkah yang perlu dilakukan meliputi memilih bucket tempat file disimpan, kemudian memilih file SQL yang akan diimpor. Selanjutnya, tentukan format SQL sebagai tipe file yang digunakan. Pada bagian tujuan, pilih database yang telah dibuat sebelumnya di instance Cloud SQL. Terakhir, lanjutkan proses impor hingga selesai.



Gambar 3.10 Import database dari Cloud Storage

4.1.7 Konfigurasi Koneksi ke Cloude SQL

Untuk memungkinkan aplikasi berkomunikasi dengan database yang telah dipindahkan ke Cloud SQL, diperlukan pengaturan koneksi. Konfigurasi ini mencakup pengaturan alamat IP, kredensial login (username dan password), serta parameter koneksi lainnya agar aplikasi dapat mengakses database dengan baik.

Untuk menghubungkan aplikasi Node.js ke database MySQL di Cloud SQL, digunakan Sequelize. Koneksi dibuat dengan menentukan nama database, username, password, host (IP Cloud SQL), dan dialek MySQL.

Tabel 3.1 Source Code Database.js

```
import { Sequelize } from "sequelize";

const db = new Sequelize("data_user_coba",
  "root", "password123", {
    host: "34.56.154.227",
    dialect: "mysql",
```

```
});  
export default db;
```

4.1.8 Pengujian dan Validasi

Langkah terakhir dalam proses migrasi adalah melakukan pengujian dan validasi untuk memastikan bahwa database berfungsi dengan baik di lingkungan cloud. Pengujian ini dilakukan dengan mengirimkan permintaan ke database, memeriksa integritas data, serta memastikan bahwa sistem dapat menangani operasi baca dan tulis dengan benar tanpa kehilangan atau perubahan yang tidak diinginkan.

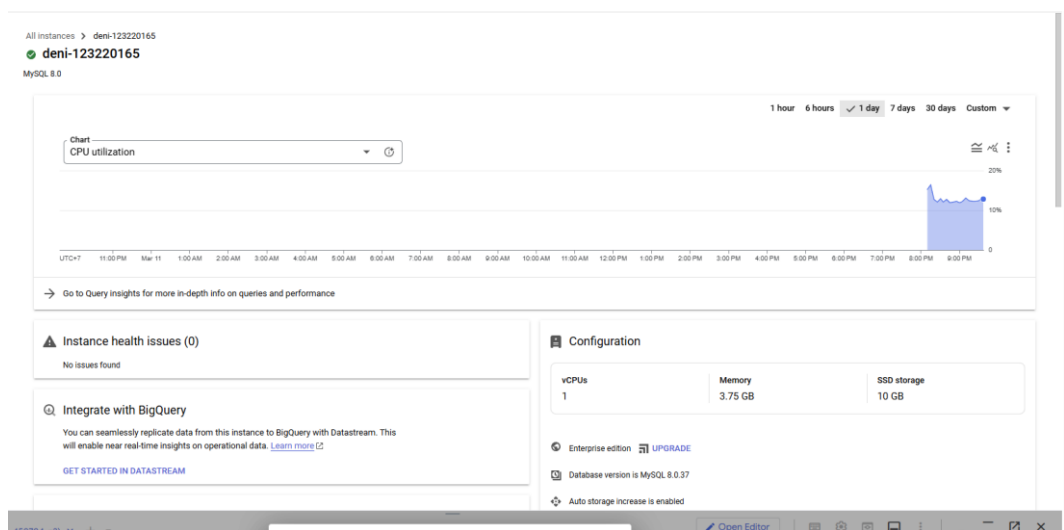
BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

4.1.1 Pembuatan instance Cloud SQL di GCP

Langkah pertama dalam migrasi SQL ke cloud adalah membuat instance SQL di GCP. Dalam hal ini, nama instance yang digunakan adalah "deni-123220165". Instance Cloud SQL telah berhasil dibuat dengan konfigurasi yang sesuai dengan kebutuhan. Gambar 3.12 menunjukkan bahwa instance tersebut telah berhasil dibuat dan siap digunakan untuk

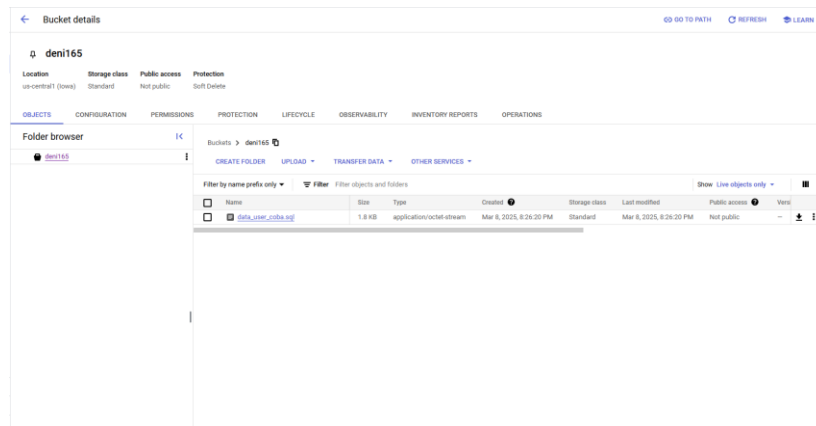


migrasi database.

Gambar 3.11 Instance berhasil dibuat

4.1.2 Pembuatan bucket di Cloud Storage

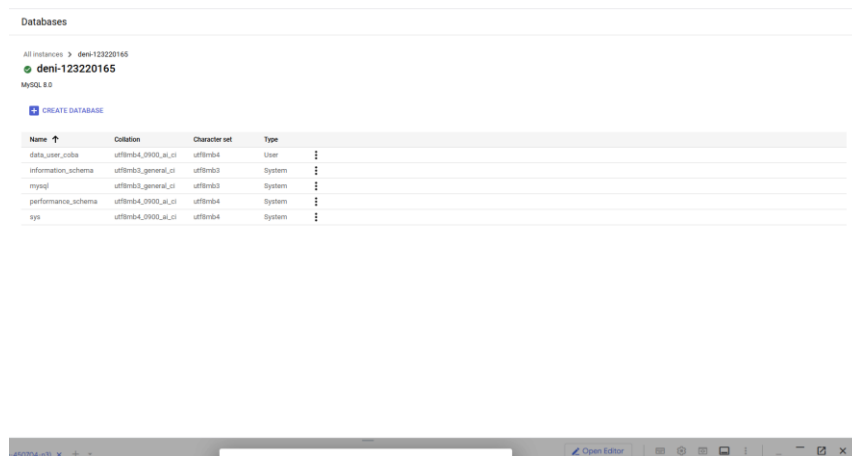
Bucket di Cloud Storage telah berhasil dibuat dengan nama "deni165" sebagai tempat penyimpanan sementara untuk file SQL yang akan diunggah. Gambar 3.13 menunjukkan bahwa bucket tersebut sudah siap untuk digunakan.



Gambar 3.12 Bucket berhasil dibuat

4.1.3 Pembuatan database di instance

Database catatan telah berhasil dibuat di dalam instance Cloud SQL. Seperti yang terlihat pada Gambar 3.14, database tersebut kini siap digunakan dan dapat diisi dengan data dari database lokal.

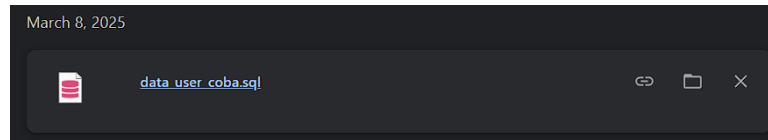


Gambar 3.13 Database instance berhasil dibuat

4.1.4 Ekspor database dari sistem lokal ke format SQL

Proses ekspor database dari localhost ke dalam file SQL telah berhasil dilakukan menggunakan phpMyAdmin. Gambar 3.15

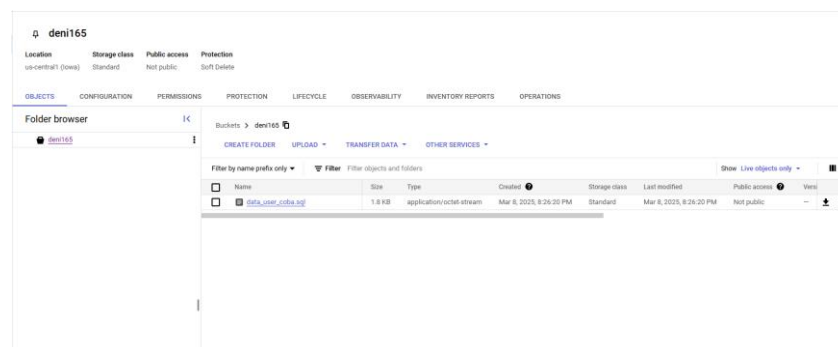
menunjukkan bahwa database telah diekspor, memastikan bahwa struktur dan data tersimpan dengan baik sebelum diunggah ke Cloud Storage.



Gambar 3.14 Database SQL berhasil di ekspor

4.1.5 Pengunggahan file SQL ke Cloud Storage

File SQL yang telah diekspor berhasil diunggah ke bucket Cloud Storage. Gambar 3.16 menunjukkan bahwa file catatan.sql telah tersimpan dengan aman di Cloud Storage dan siap digunakan dalam proses migrasi ke Cloud SQL.



Gambar 3.15 Database berhasil diunggah ke cloud storage

4.1.6 Impor database ke Cloud SQL

File SQL, catatan.sql, yang telah disimpan di Cloud Storage berhasil diimpor ke database yang telah dibuat di Cloud SQL. Gambar 3. menunjukkan proses pengimporan database, memastikan bahwa data telah berhasil dipindahkan dan siap untuk digunakan.

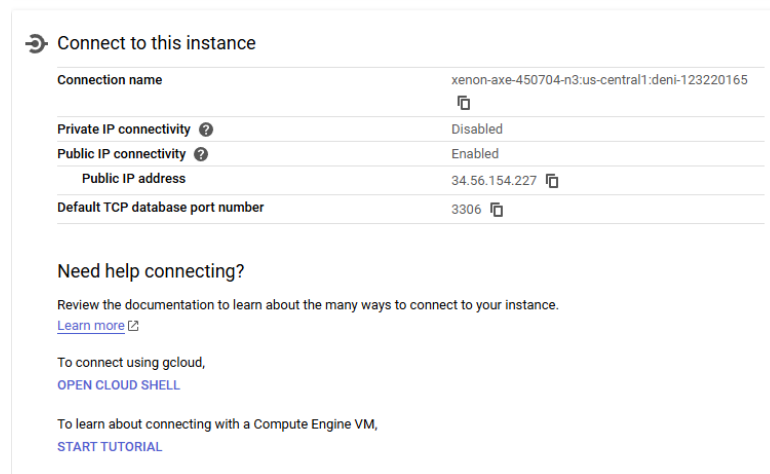
 CREATE DATABASE

Name ↑	Collation	Character set	Type	
data_user_coba	utf8mb4_0900_ai_ci	utf8mb4	User	⋮
information_schema	utf8mb3_general_ci	utf8mb3	System	⋮
mysql	utf8mb3_general_ci	utf8mb3	System	⋮
performance_schema	utf8mb4_0900_ai_ci	utf8mb4	System	⋮
sys	utf8mb4_0900_ai_ci	utf8mb4	System	⋮

Gambar 3.16 Database berhasil diimpor ke Cloud SQL

4.1.7 Konfigurasi koneksi ke Cloud SQL

Setelah database berhasil diimpor, langkah berikutnya adalah mengonfigurasi koneksi ke instance Cloud SQL. Gambar 3.18 menunjukkan hasil konfigurasi koneksi yang telah berhasil dilakukan, termasuk pengaturan IP, username, dan password, yang memastikan bahwa aplikasi dapat terhubung dengan database di Cloud SQL.



Gambar 3.17 Konfigurasi koneksi instance

4.1.8 Pengujian dan validasi

Pengujian dilakukan dengan mengirimkan permintaan ke database yang telah dimigrasikan. Tabel 3.2 menampilkan permintaan yang dikirim,

sedangkan Gambar 3.19 menunjukkan respons yang diterima. Hasil pengujian menunjukkan bahwa database terhubung dengan baik dan dapat menerima data baru.

Tabel 3.2 Request yang dikirim

```
POST http://localhost:5000/add-user
Content-Type: application/json

{
  "name": "tes13",
  "email": "tes13@tes.com",
  "gender": "tes13"
}
```

```
1 HTTP/1.1 201 Created
2 X-Powered-By: Express
3 Access-Control-Allow-Origin: *
4 Content-Type: application/json; charset=utf-8
5 Content-Length: 22
6 ETag: W/"16-8u22J4LBjqIw4Z43IYG71oVe2A8"
7 Date: Tue, 11 Mar 2025 15:03:49 GMT
8 Connection: close
9
10 {
11   "msg": "User Created"
12 }
```

Gambar 3.18 Response dari cloud

Permintaan berhasil diterima oleh database Cloud. Hal ini terlihat pada Gambar 3.20, yang menunjukkan data yang telah berhasil diterima dan disimpan dengan baik dalam database Cloud.

```

CLOUD SHELL
Terminal (xenon-axe-450704-n3) x + v

| sys |
+-----+
5 rows in set (0.20 sec)

mysql> use data user_coba;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables
+-----+
| Tables_in_data_user_coba |
+-----+
| users |
+-----+
1 row in set (0.21 sec)

mysql> select * from users;
+-----+-----+-----+-----+-----+-----+
| id | name | email | gender | createdAt | updatedAt |
+-----+-----+-----+-----+-----+-----+
| 1 | tes | tes@tes.com | tes | 2025-03-08 08:24:59 | 2025-03-08 08:24:59 |
| 2 | tes2 | tes2@tes.com | tes2 | 2025-03-09 16:46:05 | 2025-03-09 16:46:05 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.22 sec)

mysql> select * from users;
+-----+-----+-----+-----+-----+-----+
| id | name | email | gender | createdAt | updatedAt |
+-----+-----+-----+-----+-----+-----+
| 1 | tes | tes@tes.com | tes | 2025-03-08 08:24:59 | 2025-03-08 08:24:59 |
| 2 | tes2 | tes2@tes.com | tes2 | 2025-03-09 16:46:05 | 2025-03-09 16:46:05 |
| 3 | tes13 | tes13@tes.com | tes13 | 2025-03-11 15:03:47 | 2025-03-11 15:03:47 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.20 sec)

mysql>

```

Gambar 3.19 Data berhasil diterima di Cloud

4.2 Pembahasan

Implementasi migrasi database ke Cloud SQL telah berhasil dilaksanakan sesuai dengan langkah-langkah yang telah direncanakan. Seluruh proses, mulai dari ekspor database lokal, penyimpanan sementara di Cloud Storage, hingga impor ke Cloud SQL, telah berjalan dengan lancar. Dengan demikian, tujuan migrasi, yaitu memindahkan database dari sistem lokal ke Cloud SQL, telah tercapai dengan baik dan sesuai harapan.

Penggunaan Cloud Storage sebagai media penyimpanan sementara terbukti efektif dan efisien. Penyimpanan sementara ini memungkinkan akses dan pemrosesan data dengan mudah tanpa risiko kehilangan atau kerusakan data, karena proses transfer dilakukan secara terstruktur. Pemilihan Google Cloud Platform (GCP) dengan Cloud SQL dan Cloud Storage juga terbukti tepat, karena kedua layanan ini menawarkan integrasi yang mulus, kemudahan dalam pengelolaan, dan mengurangi kebutuhan konfigurasi manual.

Namun, perlu diperhatikan biaya yang dikenakan untuk penyimpanan dan penggunaan layanan cloud, yang bergantung pada kapasitas yang digunakan. Oleh karena itu, pemilihan kapasitas dan layanan harus disesuaikan dengan kebutuhan serta anggaran yang tersedia, agar migrasi dapat berjalan optimal tanpa membebani biaya yang tidak perlu.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pembahasan yang telah dilakukan, dapat disimpulkan bahwa proses migrasi database ke Cloud SQL berhasil dilaksanakan sesuai dengan tujuan yang telah ditetapkan. Semua tahapan, mulai dari ekspor database lokal, penyimpanan sementara di Cloud Storage, hingga impor ke Cloud SQL, telah berjalan dengan lancar dan efektif.

Penggunaan Cloud Storage sebagai media penyimpanan sementara terbukti meningkatkan fleksibilitas dan mengurangi risiko kehilangan data. Pemilihan Google Cloud Platform (GCP) dengan Cloud SQL juga terbukti tepat, karena mendukung integrasi yang mulus dan memudahkan pengelolaan tanpa memerlukan konfigurasi manual.

Namun, ada beberapa hal yang perlu diperhatikan, seperti biaya penyimpanan dan kapasitas layanan yang harus disesuaikan dengan kebutuhan. Secara keseluruhan, migrasi ini layak untuk diterap

Kesimpulan merupakan intisari dari pembahasan yang bersifat general, berisi hasil pembahasan yang dapat menjawab rumusan masalah dan mampu membuktikan capaian tujuan, menyimpulkan bukti-bukti yang diperoleh dan akhirnya menarik kesimpulan apakah hasil yang dikerjakan layak untuk digunakan (diimplementasikan).

5.2 Saran

Berdasarkan hasil evaluasi sistem, terdapat beberapa rekomendasi untuk pengembangan lebih lanjut. Pertama, penting untuk melakukan pemantauan kapasitas layanan cloud secara berkala agar biaya tidak meningkat secara tidak terkendali. Kedua, disarankan untuk menyiapkan cadangan data sebagai langkah antisipasi terhadap kemungkinan kehilangan informasi. Terakhir, perlu dilakukan

pemantauan dan pembaruan kebijakan keamanan data secara terus-menerus, terutama jika database menyimpan informasi yang bersifat sensitif.

DAFTAR PUSTAKA

Google Cloud. (2023). Google Cloud Platform Overview. Retrieved from <https://cloud.google.com/>

Google Cloud. (2023). Cloud Storage Overview. Retrieved from <https://cloud.google.com/storage>

Google Cloud. (2023). Cloud SQL Overview. Retrieved from <https://cloud.google.com/sql>

LAMPIRAN

6.1 Link Github

Link Github: <https://github.com/Indo86/Migrasi-Database.git>