

Penerapan Sistem Terdistribusi pada Database Website Forum Menggunakan Google Cloud Platform

Ikrar Bagaskara¹, Anggun Berlian Agustina^{2*}

^{1,2}Teknik Informatika/Fakultas Ilmu Komputer

Jl. Bhayangkara No.55, Tipes, Kec. Serangan, Kota Surakarta, Jawa Tengah 57154

¹210103101@mhs.udb.ac.id, ^{2*}210103178@mhs.udb.ac.id

Abstrak— Penelitian ini bertujuan untuk menerapkan sistem terdistribusi pada database website forum menggunakan Google Cloud Platform (GCP). Penerapan ini melibatkan teknik read replica untuk meningkatkan kinerja server MySQL. Dengan menggunakan layanan Cloud SQL di GCP, replika baca (read replica) dibuat untuk mempercepat akses ke database utama. Penelitian ini juga mencakup pengujian yang berhasil dilakukan untuk memverifikasi keberhasilan implementasi sistem terdistribusi. Pengujian melibatkan simulasi beban tinggi dan pengukuran kinerja, termasuk latensi akses dan waktu respon sistem. Hasil penelitian menunjukkan bahwa penerapan sistem terdistribusi dengan teknik read replica pada database website forum menggunakan GCP mampu meningkatkan kinerja dan ketersediaan data. Penelitian ini memberikan kontribusi dalam pengembangan sistem terdistribusi untuk aplikasi web dengan memanfaatkan layanan Cloud SQL di GCP. Hasilnya dapat digunakan sebagai panduan bagi pengembang dan organisasi yang ingin mengimplementasikan sistem terdistribusi pada database mereka, khususnya pada website.

Kata kunci— Google Cloud Platform, database, sistem terdistribusi, read replica.

Abstract— This research aims to implement a distributed system on the forum website database using Google Cloud Platform (GCP). This implementation involves the read replica technique to improve the performance of the MySQL server. Using the Cloud SQL service in GCP, read replicas were created to speed up access to the main database. This research also includes successful testing to verify the success of the distributed system implementation. The tests involved high-load simulations and performance measurements, including access latency and system response time. The results show that the implementation of a distributed system with read replica techniques on a forum website database using GCP is able to improve performance and data availability. This research contributes to the development of distributed systems for web applications by utilizing Cloud SQL services in GCP. The results can be used as a guide for developers and organizations who want to implement a distributed system on their database, especially on websites.

Keywords— Google Cloud Platform, database, distributed system, read replica.

I. PENDAHULUAN

Media digital merupakan cara untuk mengintegrasikan informasi, layanan, budaya dan hiburan, komunikasi dan interaksi [3], dan salah satu untuk bertukar informasi pada media digital salah satunya melalui forum online. Forum online adalah platform yang memungkinkan pengguna untuk berbagi informasi, bertukar ide, dan berdiskusi tentang topik yang relevan. Pada penggunaan dibidang pendidikan, forum diskusi online membawa pengajaran dan pembelajaran di luar lingkungan belajar yang terbatas seperti ruang kelas, membawa fleksibilitas dan kenyamanan dalam mengajar dan pembelajaran [1]. Namun, dengan pertumbuhan pesat jumlah pengguna dan lalu lintas yang tinggi, website forum sering mengalami tantangan dalam menghadapi beban kerja yang besar dan memastikan ketersediaan dan responsivitas yang baik.

Dalam upaya untuk mengatasi tantangan tersebut, implementasi sistem terdistribusi telah menjadi solusi yang efektif. Sistem terdistribusi

memungkinkan distribusi beban kerja di antara beberapa sumber daya komputasi, mempercepat pemrosesan data, dan meningkatkan ketersediaan sistem. Salah satu metode yang digunakan untuk ketersediaan sistem adalah menggunakan teknik *fail-over*, Teknik failover bekerja ketika server utama mengalami kerusakan dan mengakibatkan kegagalan layanan, sehingga server cadangan mengambil alih layanan [2]. Salah satunya layanan yang dapat mendukung hal ini yaitu, google cloud. Google Cloud menawarkan berbagai layanan yang mencakup mesin komputasi, yang merupakan infrastruktur yang memungkinkan pengguna menjalankan mesin virtual seperti Microsoft Windows dan Linux. Selain itu, mereka juga menyediakan aplikasi engine yang merupakan platform layanan komputasi di awan [4].

Selain meningkatkan kinerja dan skalabilitas, implementasi sistem terdistribusi menggunakan Google Cloud juga memiliki manfaat lainnya. Keamanan adalah aspek penting yang harus dipertimbangkan dalam pengoperasian forum

online, dan Google Cloud menyediakan fitur keamanan yang kuat untuk melindungi data pengguna dan mencegah serangan cyber. Selain itu, layanan Google Cloud juga menawarkan beberapa komponen keamanan umum untuk melindungi data termasuk manajemen kunci, pencegahan kehilangan data, pemantauan aktivitas, dan perlindungan data di titik akhir [5].

a. Landasan Teori

1. Cloud Computing

Komputasi awan adalah kumpulan dari komputasi cluster dan grid yang digunakan untuk mengumpulkan sumber daya di satu tempat sentral dan memanfaatkannya untuk komputasi berkinerja tinggi [6].

a. Pay as you go

Referensi [7] mengatakan komputasi awan sangat populer karena menyediakan layanan dan sumber daya komputasi dengan sistem bayar per penggunaan kepada pengguna sesuai dengan kebutuhan mereka.

b. Skalabilitas

Skalabilitas adalah kemampuan sistem cloud untuk meningkatkan kapasitas pelayanan perangkat lunak dengan menambah jumlah layanan perangkat lunak yang tersedia [8].

c. Elastic

Penyimpanan data harus dapat beradaptasi dengan meningkatnya beban serta dapat mengurangi beban dengan melepaskan sumber daya cloud yang tidak terpakai, sekaligus memastikan kepatuhan terhadap Perjanjian Tingkat Layanan (SLA) [9].

2. Google Cloud

Google Cloud terdiri dari berbagai aset fisik, seperti komputer dan hard disk drive, serta sumber daya virtual, seperti mesin virtual (VM), yang tersedia di pusat data Google yang terletak di berbagai lokasi di seluruh dunia. Setiap pusat data berada di wilayah tertentu [10].

3. Database

Basis data dapat diartikan sebagai kumpulan data tentang suatu benda / kejadian yg saling berhubungan satu sama lain [11].

4. Read Replica

Dalam sistem ini, terdapat satu komputer yang berperan sebagai master dan yang lainnya sebagai

slave. Komputer server bertugas untuk mengakses dan menulis ke database selama operasi berlangsung. Di sisi lain, komputer slave hanya bertugas untuk membaca dari database [12].

5. MySQL

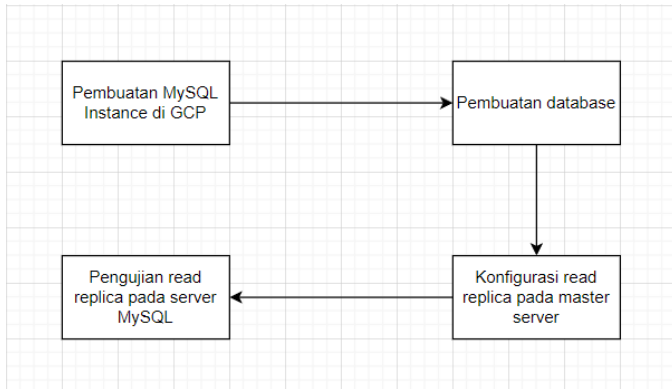
MySQL adalah sebuah database yang dikembangkan oleh Michael Widenius dan selesai pada bulan Mei 1995. Michael Widenius menciptakan MySQL sebagai alat yang diperlukan untuk menyelesaikan proyek yang sedang ia kerjakan pada saat itu [14]. Referensi [13] mengatakan MySQL adalah salah satu teknologi basis data yang sangat populer dan digunakan secara luas di berbagai tempat. Keandalannya telah terbukti, meskipun ada banyak pilihan teknologi basis data lain selain MySQL yang juga tersedia.

Penelitian ini akan mengeksplorasi arsitektur dan teknologi terkait yang diperlukan untuk mendukung sistem terdistribusi yang efisien dan handal. Dalam implementasi ini, peneliti akan menggunakan Google Cloud Platform sebagai infrastruktur komputasi yang mendukung distribusi beban kerja dan manajemen sumber daya.

Tujuan dari penelitian ini adalah untuk mengimplementasikan sistem terdistribusi menggunakan layanan Google Cloud untuk meningkatkan kinerja dan kehandalan website forum. Peneliti akan menganalisis arsitektur sistem terdistribusi, mempertimbangkan faktor-faktor seperti keamanan, skalabilitas, dan kehandalan, dan mengukur kinerja sistem dalam skenario penggunaan yang berbeda. Selain itu, kami akan membahas manfaat dan tantangan yang terkait dengan implementasi sistem terdistribusi menggunakan layanan cloud.

II. METODOLOGI PENELITIAN

Metodologi yang digunakan pada penelitian ini menggunakan metodologi deskriptif untuk menjelaskan bagaimana proses penelitian. Alur penelitian untuk implementasi sistem terdistribusi website forum menggunakan google cloud platform adalah sebagai berikut:



Gambar 1. Alur penelitian

2.1 Pembuatan MySQL Instance di GCP

Membuat instance atau MySQL pada Google Cloud Service sesuai dengan spesifikasi yang ditentukan untuk melakukan pengujian.

2.2 Pembuatan database.

Tahap ini dilakukan setelah MySQL Server sudah di buat dan konfigurasi sesuai keperluan peneliti, di tahap ini juga peneliti membuat tabel dan column.

2.3 Konfigurasi read-replica pada master server

Tahap ini dilakukan dengan melakukan konfigurasi read replika dari server utama MySQL, sehingga server MySQL menjadi 2. Dimana primary MySQL menjadi master dan read replika server menjadi slave.

2.4 Pengujian read-replica

Tahap ini dilakukan diperlukan untuk menguji apakah penerapan read replica berdampak pada beban server MySQL yang sudah dibuat sebelumnya. Peneliti membuat script sederhana dengan bahasa pemrograman python, dengan melakukan proses query *insert* dengan jumlah record 500 ribu query, dan kemudian melakukan proses read 1 juta query secara bersamaan.

a. *Insert.py*

```

import mysql.connector
from faker import Faker
import random

primary_host = '34.128.101.31'
primary_user = 'root'
primary_password = 'r4ngk1ng'
primary_database = 'test_db'
  
```

try:

```
primary_connection=mysql.connector.connect(
```

```

host=primary_host,
user=primary_user,
password=primary_password,
database=primary_database
)
  
```

```
print("Successfully connected to the MySQL server.")
```

```
fake = Faker()
```

```
num_rows = 1000000
```

```
batch_size = 10000
```

```
insert_query = ""
```

```
INSERT INTO high_workload
```

```
(name, gender, address, phone, email, date_reg,
```

```
last_login)
```

```
VALUES (%s, %s, %s, %s, %s, %s, %s)
```

```
""
```

```
cursor = primary_connection.cursor()
```

```
for i in range(0, num_rows, batch_size):
```

```
batch_data = []
```

```
for _ in range(batch_size):
```

```
name = fake.name()
```

```
gender = random.choice(['Male', 'Female'])
```

```
address = fake.address()
```

```
phone = fake.phone_number()
```

```
email = fake.email()
```

```
date_reg = fake.date()
```

```
last_login = fake.date_time_this_decade()
```

```
values = (name, gender, address, phone, email,
```

```
date_reg, last_login)
```

```
batch_data.append(values)
```

```
cursor.executemany(insert_query, batch_data)
```

```
primary_connection.commit()
```

```
print(f"Successfully inserted {i + batch_size} rows of data.")
```

```
print(f"Successfully inserted {num_rows} rows of data into the 'high_workload' table.")
```

```
except mysql.connector.Error as error:
```

```
print("Error connecting to MySQL server:", error)
```

b. *Read.py*

```
import mysql.connector
```

```
import time
```

```
import locale
```

```
primary_host = '$IP_address'
```

```
primary_user = '$Username'
```

```
primary_password = '$password'
```

```
primary_database = 'test_db'
```

```
try:
```

```
primary_connection=mysql.connector.connect(
```

```

host=primary_host,
user=primary_user,
password=primary_password,
database=primary_database
)

print("Successfully connected to the MySQL server.")

cursor = primary_connection.cursor()

start_time = time.time()
end_time = start_time + 60
total_rows = 0

while time.time() < end_time:
    query = "SELECT COUNT(*) FROM high_workload"
    cursor.execute(query)

    total_rows += cursor.fetchone()[0]

execution_time = time.time() - start_time

formatted_total_rows = locale.format_string("%d",
total_rows, grouping=True)

print(f"Execution time: {execution_time} seconds")
print(f"Total rows selected: {formatted_total_rows}")





except mysql.connector.Error as error:
    print("Error connecting to MySQL server:", error)

```

III. HASIL DAN PEMBAHASAN

3.1 Pembuatan MySQL Instances

Pada tahap ini peneliti membuat MySQL Instance pada Google Cloud SQL.

<input type="checkbox"/>	Instance ID  	Type	Public IP address
<input type="checkbox"/>	  mysql-forum	MySQL 8.0	34.101.186.109

Gambar 2. MySQL instance

Pada gambar menampilkan informasi dari instance yang dibuat adalah MySQL dengan versi 8.0, dan port yang digunakan adalah 3306 atau port bawaan dari MySQL. Dan nama instance yang dibuat yaitu, *mysql-forum*.

Summary

Region	asia-southeast2 (Jakarta)
DB Version	MySQL 8.0.33
vCPUs	1 vCPU
Memory	628.74 MB
Storage	10 GB
Network throughput (MB/s) 	125 of 125
Disk throughput (MB/s) 	Read: 4.8 of 125.0 Write: 4.8 of 37.9
IOPS 	Read: 300 of 15,000 Write: 300 of 4,500
Connections	Public IP
Backup	Automated
Availability	Multiple zones (Highly available)
Point-in-time recovery	Enabled

Gambar 3. Informasi dari instance MySQL

Gambar 3 menunjukkan informasi dari instances MySQL yang dibuat sebelumnya, di antaranya adalah menggunakan 1 vCPU, 628.74 MB RAM, Storage type SSD 10 GB, dengan network throughput 125 dan disk throughput read 125.0 dan write 37.9. Dan ditambah dengan konfigurasi Multiple zones sehingga ketika zone yang dimana server diletakan mengalami kegagalan, akan berpindah ke zone lain dalam region yang sama secara otomatis dan secepatnya.

3.2 Pembuatan database

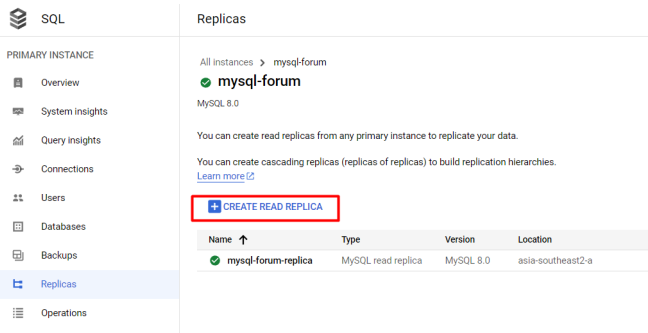
Tabel dibuat dengan nama *high_workload* kemudian diberikan 8 kolom untuk uji coba, dengan informasi kolom sebagai berikut

Table: <i>high_workload</i>	
Columns:	
<i>id</i>	int PK
<i>name</i>	char(100)
<i>gender</i>	varchar(45)
<i>address</i>	varchar(1000)
<i>phone</i>	varchar(45)
<i>email</i>	varchar(45)
<i>date_reg</i>	date
<i>last_active</i>	date

Gambar 4. Tabel *high_workload*

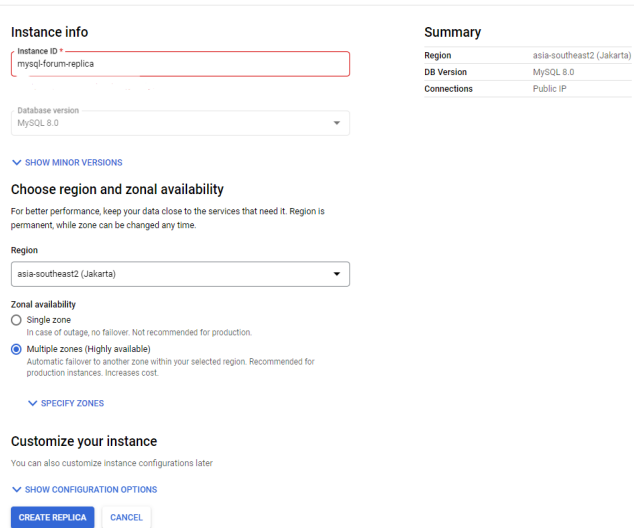
Pada tahap ini kolom pada tabel *high_workload* sudah dibuat dan disesuaikan dengan kebutuhan.

3.3 Konfigurasi read replica pada server



Gambar 5. Pembuatan Read Replica

Gambar menunjukkan lokasi untuk membuat read replica dari instance utama yang dibuat sebelumnya.



Gambar 6. Detail pembuatan read replica

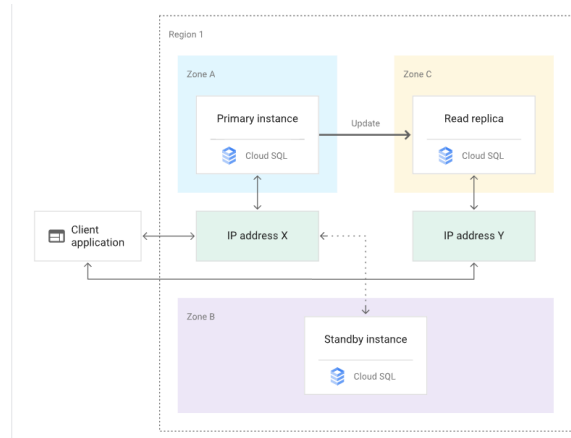
Gambar 6 menunjukkan informasi lebih lanjut yang dibutuhkan untuk pembuatan read replica. Dengan nama *mysql-forum-replica*, dan region Jakarta, dalam memilih region pastikan pilih yang terdekat dengan dimana server akan digunakan. Dan pilih *Multiple zones (Highly available)* agar instance yang dibuat fail over, sehingga apabila salah satu zone gagal, dapat berpindah ke zone lain tanpa adanya downtime yang signifikan.

Instance ID	Type	Public IP address
mysql-forum	MySQL 8.0	34.101.186.109
mysql-forum-replica	MySQL read replica	34.101.106.120

Gambar 7. Hasil pembuatan read replica

Gambar 7 menunjukkan read replica sudah di buat, read replica akan muncul di bawah server MySQL utama

3.4 Pengujian read-replica

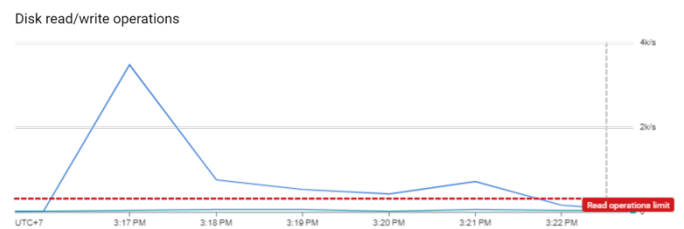


Gambar 8. Arsitektur read replica

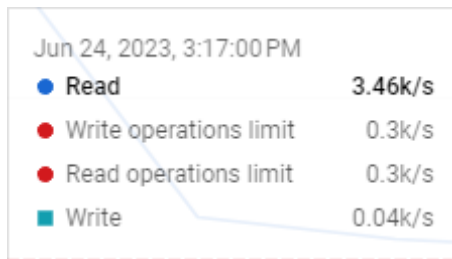
Gambar 8 menampilkan arsitektur yang digunakan dalam read replica dengan proses sebagai berikut:

1. Client mengirimkan data *write* ke server database melalui IP Address X atau IP Address public dari server.
2. Data kemudian diterima oleh server database utama.
3. Kemudian slave database akan melakukan proses replikasi read secara otomatis.
4. Dan selanjutnya ketika client melakukan request *read* dari server, maka secara otomatis akan diarahkan ke slave database yang bertugas sebagai read replica
5. Sehingga beban query dari *write* dan *read* terpisah, dan server utama tidak terbebani secara penuh.

Pengujian read replica

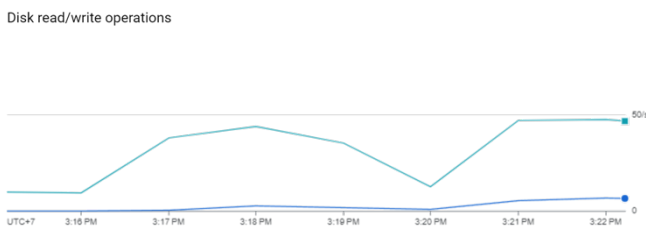


Gambar 9. Aktifitas read pada server MySQL Replica

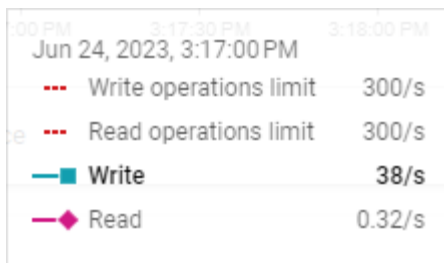


Gambar 10. Aktifitas read pada server MySQL Replica

Pada gambar 9, 10 menunjukkan dengan pengujian yang dilakukan kurang lebih 7 menit di rentang waktu antara 24 Juni 2023, jam 3.15 WIB sampai dengan 3.22 WIB menunjukkan bahwa proses *read* berjalan dengan baik dan mampu menerima request yang cukup tinggi.



Gambar 11. Aktifitas write pada server MySQL Utama



Gambar 12. Aktifitas write pada server MySQL Utama

Pada gambar 11, 12 menunjukkan dengan waktu pengujian yang sama, server MySQL bisa menerima request *write* tanpa terbebani oleh request *read*, hal ini berarti proses sistem terdistribusi database dengan teknik read replica dapat berjalan dengan baik.

IV. KESIMPULAN

Penelitian ini mengimplementasikan sistem terdistribusi pada database website forum menggunakan Google Cloud Platform. Tujuan utama penelitian ini adalah meningkatkan kinerja dan kehandalan website forum melalui distribusi beban kerja. Dalam penelitian ini, teknik read replica

berhasil diterapkan untuk mendistribusikan beban kerja pada database.

Dengan menggunakan read replica, beban kerja terbagi secara merata antara server replica, mempercepat pemrosesan data dan mengurangi waktu respon. Selain itu, distribusi beban kerja pada database juga membantu meningkatkan skalabilitas sistem, memungkinkan website forum untuk menangani jumlah pengguna yang lebih besar dan lalu lintas yang tinggi.

Secara keseluruhan, penerapan sistem terdistribusi menggunakan read replica pada Google Cloud Platform membawa manfaat signifikan bagi website forum, termasuk peningkatan kinerja, kehandalan, dan skalabilitas. Dalam konteks pengelolaan database website forum, distribusi beban kerja melalui read replica merupakan pendekatan yang efektif untuk mengoptimalkan penggunaan sumber daya dan meningkatkan responsivitas sistem.

Dalam penelitian selanjutnya, dapat dilakukan eksplorasi lebih lanjut untuk menguji dan membandingkan berbagai teknik dan strategi distribusi beban kerja pada sistem terdistribusi dalam konteks website forum.

UCAPAN TERIMA KASIH

Kami ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua individu dan pihak yang telah memberikan kontribusi dan dukungan selama penelitian ini. Tanpa kerja keras, panduan, dan bantuan mereka, penelitian ini tidak akan terwujud. Kami ingin mengucapkan terima kasih kepada:

1. Pembimbing kami, Ibu Agustina Srirahayu, M.Kom, atas arahan, saran, dan pengawasannya yang berharga sepanjang penelitian ini.
2. Keluarga dan teman-teman kami atas dukungan, motivasi, dan pengertian yang mereka berikan selama proses penelitian ini.
3. Institusi dan lembaga yang telah memberikan akses dan izin untuk mengumpulkan data yang diperlukan.

Terima kasih atas semua dukungan dan kontribusi yang telah diberikan. Semua bantuan ini sangat berarti bagi kelancaran dan kesuksesan penelitian kami.

REFERENSI

- [1] E. M. Onyema, E. C. Deborah, A. O. Alsayed, Q. N. Naveed and S. Sanober, "Online Discussion Forum as a Tool for Interactive," *International Journal of Recent Technology and Engineering*, p. 4853, 2019.
- [2] D. Irwan, H. Sukoco and S. Wahjuni, "Service High Availability Pada Native Server dan Virtual Server Menggunakan Proxmox VE," *Jurnal Kajian Ilmiah*, p. 138, 2020.
- [3] F. Zhang, "Research on the Internet Plus Visual Communication Design - the Application of Visual Design in Internet," *Journal of Physics: Conference Series*, p. 1, 2021.
- [4] A. Gupta, P. Goswami, N. Chaudhary and R. Bansal, "Deploying an Application using Google Cloud," *IEEE Xplore*, p. 236, 2020.
- [5] A. Lance and D. A. Chuvakin, "Designing and deploying a data security strategy with Google Cloud," *Google Cloud Whitepaper*, p. 3, 2021.
- [6] V. Kumar, A. A. Laghari, S. Karim, M. Shakir and A. A. Brohi, "Comparison of Fog Computing & Cloud Computing," *Mathematical Sciences and Computing*, p. 33, 2019.
- [7] A. A. Laghari, H. He, I. A. Halepoto, M. S. Memon and S. Parveen, "Analysis of Quality of Experience Frameworks for Cloud Computing," *International Journal of Computer Science and Network Security*, p. 228, 2017.
- [8] A. A.-S. Ahmad and P. Andras, "Scalability analysis comparisons of cloud-based software services," *Journal of Cloud Computing: Advances, Systems and Applications*, p. 1, 2019.
- [9] R. A. Campêlo, M. A. Casanova, D. O. Guedes and A. H. F. Laender, "A brief survey on replica consistency in," *Journal of Internet Services*, p. 2, 2020.
- [10] Google, "Google Cloud Resource: Google Cloud," 22 Juni 2023. [Online]. Available: <https://cloud.google.com/docs/overview>.
- [11] E. Setyawati, S. H. Wijoyo and N. Soeharmoko, "Pengenalan Database, DBMS dan RDBMS," in *RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)*, Banyumas, CV. PENA PERSADA, 2020, p. 1.
- [12] P. Edastama, A. S. Bist and A. Prambudi, "Implementation Of Data Mining On Glasses Sales Using The Apriori Algorithm," *International Journal of Cyber and IT Service Management*, p. 159–172, 2021.
- [13] B. Rawat, S. Purnama and M. , "MySQL Database Management System (DBMS) On FTPSite LAPAN Bandung," *International Journal of Cyber and IT Service Management*, p. 174, 2021.
- [14] T.V. Database-Driven Web Development, 2021: Apress, Manitoba. w