

EVALUASI KINERJA DATABASE RELASIONAL DAN LAYANAN CLOUD STORAGE UNTUK TRANSMISI DATA MEDIA DALAM JARINGAN

EVALUATION OF RELATIONAL DATABASES AND CLOUD STORAGE PERFORMANCE FOR MEDIA DATA TRANSMISSION IN NETWORKS

Andhi Saputro^{1*}, Sahrul Ramadhan², Rendinis³, Oktifar Tri Bandono⁴, Ahmad Muhammad⁵,
Toyyibah T⁶

^{1,2,3,4,5,6}Universitas Pamulang, Tangerang Selatan, Banten

¹hello.andhisaputro@gmail.com, ²sahrul.rin@gmail.com, ³rendinis.geof41@gmail.com,

⁴abunafisah0711@gmail.com, ⁵achmadmr@gmail.com, ⁶dosen01116@unpam.ac.id

ABSTRACT

Media data transmission is becoming increasingly important with the growth of large and diverse multimedia content. Relational databases, like MySQL, and cloud storage services, like Amazon S3, are two major choices for storage and media data transmission. This study proposes a methodology to compare the performance of both systems based on transmission speed, reliability, scalability, security, cost, and ease of integration with existing network infrastructure. The methodology includes the collection of empirical data from test scenarios designed to simulate real conditions in the use of multimedia applications. The study aims to evaluate and compare the performance of relational databases and cloud storage services in the context of media data transmission on network systems. The research results are expected to provide useful guidance for developers and network administrators in choosing the optimal data storage solution for their media applications. Moreover, the research is expected to contribute to academic literature in the field of computer networking and data management by providing a deeper understanding of performance comparisons between relational databases and cloud storage services.

Keywords: Media, Transmission, File, Database, Cloud Storage

ABSTRAK

Transmisi data media menjadi semakin penting dengan adanya pertumbuhan konten multimedia yang besar dan beragam. *Database relasional*, seperti MySQL, dan layanan *cloud storage*, seperti Amazon S3, adalah dua pilihan utama untuk penyimpanan dan transmisi data media. Studi ini mengusulkan metodologi untuk membandingkan kinerja kedua sistem berdasarkan kecepatan transmisi, reliabilitas, skalabilitas, keamanan, biaya, dan kemudahan integrasi dengan infrastruktur jaringan yang ada. Metodologi ini mencakup pengumpulan data empiris dari skenario pengujian yang dirancang untuk mensimulasikan kondisi nyata dalam penggunaan aplikasi multimedia. Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja database relasional dan layanan *cloud storage* dalam konteks transmisi data media pada sistem jaringan. Hasil penelitian diharapkan dapat memberikan panduan yang bermanfaat bagi pengembang dan administrator jaringan dalam memilih solusi penyimpanan data yang optimal untuk aplikasi media mereka. Selain itu, penelitian ini diharapkan dapat memberikan kontribusi pada literatur akademik di bidang jaringan komputer dan manajemen data dengan menyediakan pemahaman yang lebih mendalam tentang perbandingan kinerja antara *database relasional* dan layanan *cloud storage*.

Kata Kunci: Media, Transmisi, File, Database, Cloud Storage

PENDAHULUAN

Di era digital saat ini, jumlah konten multimedia yang dikirimkan melalui jaringan terus meningkat, mendorong perlunya solusi penyimpanan dan transmisi data yang efektif dan efisien. MySQL, sebagai representasi dari database relasional, menyimpan data media dalam format BLOB base64. Sementara itu, Amazon S3, sebagai representasi dari

layanan *cloud storage*, menawarkan penyimpanan terdistribusi dengan akses cepat dan skalabilitas tinggi.

Studi ini mengusulkan metodologi untuk membandingkan kinerja kedua sistem berdasarkan kecepatan transmisi, reliabilitas, skalabilitas, keamanan, biaya, dan kemudahan integrasi dengan infrastruktur jaringan yang ada. Metodologi ini mencakup pengumpulan

data empiris dari skenario pengujian yang dirancang untuk mensimulasikan kondisi nyata dalam penggunaan aplikasi multimedia.

Penelitian ini menggunakan pendekatan eksperimental untuk mengevaluasi dan membandingkan kinerja dua sistem penyimpanan data: *database relational* (direpresentasikan oleh MySQL)[8] dan layanan *cloud storage* (direpresentasikan oleh Amazon S3)[5], dalam konteks transmisi data media pada sistem jaringan. Data media, seperti video, audio, dan gambar, memiliki ukuran yang besar dan memerlukan bandwidth yang signifikan untuk transmisi. Tanpa transmisi yang efisien, jaringan bisa mengalami kemacetan, latensi tinggi, dan kualitas layanan yang buruk, yang pada gilirannya mempengaruhi pengalaman pengguna secara negatif.

Hasil penelitian diharapkan dapat memberikan panduan yang berguna bagi pengembang dan administrator jaringan dalam memilih solusi penyimpanan data yang optimal untuk aplikasi media mereka. Selain itu, penelitian ini diharapkan dapat memberikan kontribusi pada literatur akademik di bidang jaringan komputer dan manajemen data dengan menyediakan pemahaman yang lebih mendalam tentang perbandingan kinerja antara database relasional dan layanan *cloud storage*.

TINJAUAN PUSTAKA

Database Relasional MySQL:

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang populer, digunakan secara luas untuk berbagai aplikasi web dan enterprise. Zhang dan Wang (2013) menjelaskan bahwa MySQL menawarkan berbagai fitur seperti transaksi, indeks, dan dukungan SQL yang membuatnya ideal untuk aplikasi yang memerlukan konsistensi dan integritas data.

Layanan Cloud Storage Amazon S3:

Amazon S3 adalah layanan *cloud storage* yang menyediakan penyimpanan

terdistribusi dengan akses cepat dan skalabilitas tinggi. Persico et al. (2016) mengevaluasi kinerja Amazon S3 dan menemukan bahwa layanan ini sangat efektif dalam menangani beban kerja yang tinggi serta menawarkan reliabilitas dan ketersediaan yang tinggi.

Quality of Service (QoS):

QoS adalah konsep penting dalam manajemen jaringan yang mencakup metrik-metrik kunci seperti throughput, jitter, delay, dan packet loss. Menurut beberapa sumber, pengukuran dan analisis metrik QoS membantu dalam mengoptimalkan performa aplikasi dengan memastikan data ditransmisikan secara efisien (Rath et al., 2017).

Analisis Kritis

Penelitian sebelumnya oleh Zhang dan Wang (2013) serta Persico et al. (2016) memberikan dasar yang kuat untuk memahami kinerja MySQL dan Amazon S3. Namun, ada keterbatasan dalam studi-studi tersebut, terutama dalam konteks aplikasi multimedia yang memerlukan evaluasi khusus terhadap transmisi data media. Rath et al. (2017) memberikan wawasan tentang pentingnya QoS dalam transmisi data, tetapi studi mereka tidak secara khusus membandingkan MySQL dan Amazon S3.

METODE

A. Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimental untuk mengevaluasi dan membandingkan kinerja MySQL, sebagai representasi dari database relasional, dengan Amazon S3, sebagai representasi dari layanan *cloud storage*, dalam konteks transmisi data media pada sistem jaringan. Desain ini mencakup pengaturan eksperimen yang sistematis untuk mengevaluasi berbagai aspek kinerja kedua sistem penyimpanan data.

B. Alat dan Teknik Pengumpulan Data

Penelitian ini menggunakan pendekatan eksperimental untuk mengevaluasi dan membandingkan kinerja MySQL dengan Amazon S3 dalam konteks transmisi data media pada sistem jaringan. Desain ini mencakup pengaturan eksperimen yang sistematis untuk mengevaluasi berbagai aspek kinerja kedua sistem penyimpanan data.

1. Laravel 11

Laravel Menyediakan fitur unit testing yang dapat digunakan untuk melakukan pengujian otomatis terhadap interaksi antara aplikasi dan layanan penyimpanan seperti MySQL dan Amazon S3. Dengan menggunakan fitur ini, kami dapat mengukur kinerja transmisi data media tanpa memerlukan alat analisis jaringan eksternal seperti *Wireshark*. *Unit testing* di Laravel memungkinkan kami untuk membuat skrip yang mengotomatisasi proses *upload* dan *download*, serta mengumpulkan metrik kinerja seperti *throughput*, *delay*, *jitter*, dan *packet loss*. Fitur ini memberikan fleksibilitas tinggi dalam mengkonfigurasi dan menjalankan berbagai skenario pengujian, yang mencakup berbagai ukuran dan jenis file media. Selain itu, Laravel 11 menyediakan alat yang terintegrasi langsung dengan aplikasi, memungkinkan evaluasi yang lebih kohesif dan relevan terhadap performa sistem penyimpanan data dalam konteks penggunaan aplikasi sebenarnya. Penggunaan Laravel untuk pengujian ini juga menyederhanakan proses pengumpulan data dan analisis, serta memastikan bahwa pengujian dapat diulang dengan mudah untuk validasi hasil. Dengan pendekatan ini, kami dapat memastikan bahwa hasil pengujian mencerminkan kondisi operasional yang realistis, memberikan wawasan yang lebih akurat mengenai efisiensi dan keandalan MySQL dan Amazon S3 dalam skenario nyata.

2. Quality Of Service (QoS)

Quality of Service adalah konsep yang penting dalam manajemen jaringan untuk memastikan kinerja dan keandalan aplikasi web. Dalam konteks aplikasi Laravel 11, QoS mencakup metrik-metrik kunci seperti *throughput*, *jitter*, *delay*, dan *packet loss*. Mengukur dan menganalisis metrik-metrik ini membantu pengembang mengoptimalkan performa aplikasi [1], dengan memastikan data ditransmisikan secara efisien antara server Laravel dan layanan eksternal seperti database MySQL dan penyimpanan *cloud* Amazon S3.

Throughput mengukur kecepatan transfer data efektif dalam *bit per second* (bps), memberikan gambaran tentang seberapa cepat data dapat diunggah atau diunduh dari layanan penyimpanan. *Jitter* mengukur variasi dalam waktu pengiriman paket data, yang dapat mempengaruhi stabilitas dan kualitas transmisi data, terutama penting untuk aplikasi yang sensitif terhadap waktu seperti streaming video atau audio. *Delay* mengukur keterlambatan dalam pengiriman dan penerimaan data, yang penting untuk memastikan responsivitas aplikasi. *Packet loss* mengukur jumlah paket data yang hilang selama transmisi, yang dapat mengindikasikan masalah jaringan atau kinerja layanan penyimpanan.

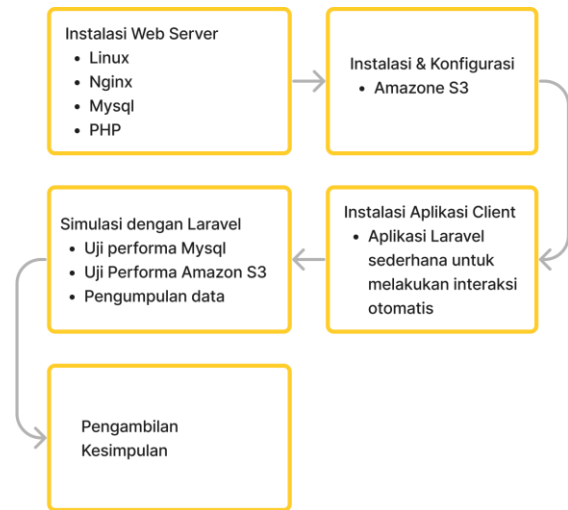
Dengan menggunakan fitur bawaan Laravel untuk pemantauan dan *logging*, pengembang dapat secara otomatis mengumpulkan dan menganalisis metrik QoS ini. *Unit testing* di Laravel memungkinkan pembuatan skrip yang mengotomatisasi proses *upload* dan *download*, serta mengumpulkan data kinerja. Penggunaan Laravel 11 untuk pengujian ini memberikan pendekatan yang terintegrasi langsung dengan aplikasi, memungkinkan evaluasi yang lebih kohesif dan relevan terhadap performa sistem penyimpanan data dalam konteks penggunaan aplikasi sebenarnya. Pengembang dapat mengidentifikasi potensi *bottleneck* dalam jaringan atau dalam interaksi dengan layanan eksternal,

dan melakukan penyesuaian yang diperlukan untuk meningkatkan kinerja.

Pengujian ini tidak hanya memberikan wawasan tentang efisiensi transmisi data tetapi juga memastikan bahwa aplikasi dapat menangani berbagai skenario beban kerja dan kondisi jaringan. Dengan demikian, penggunaan fitur *unit testing* di Laravel untuk mengukur QoS tidak hanya menyederhanakan proses pengumpulan data dan analisis, tetapi juga memastikan bahwa aplikasi dapat dioptimalkan secara berkelanjutan, memberikan pengalaman pengguna yang lebih baik. Pengujian ini penting untuk memastikan bahwa aplikasi dapat memberikan kinerja yang konsisten dan handal, bahkan di bawah kondisi jaringan yang menantang.

Tabel 1. Kategori Penilaian

No	Kategori	Keterangan
1	<i>Throughput</i>	Kecepatan transfer data efektif, yang diukur dalam bps (<i>bit per second</i>)
2	<i>Jitter</i>	Variasi waktu dari sinyal periodik dalam elektronik dan telekomunikasi
3	<i>Delay</i>	Keterlambatan dan pengiriman dan penerima dalam waktu transmisi data. Satuan pengukur dari <i>delay</i> adalah sekon atau detik
4	<i>Packet Loss</i>	Jumlah paket hilang yang terjadi saat pengiriman paket

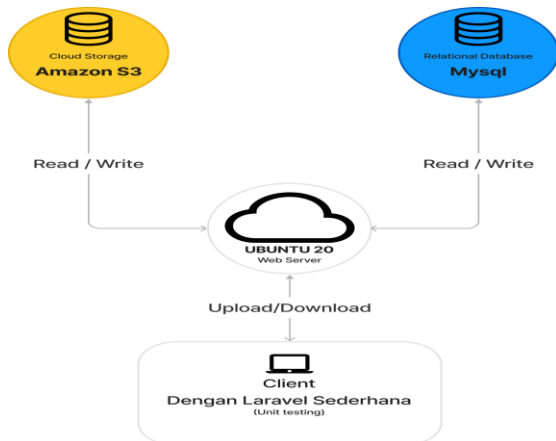


Gambar 1. Tahapan Penelitian

Pada gambar 1 menunjukkan 6 (enam) proses tahapan penelitian. Dapat dijelaskan sebagai berikut:

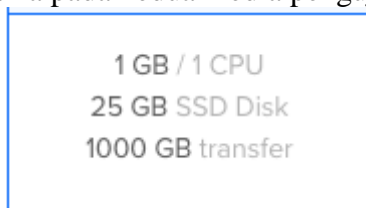
1. **Instalasi Web Server:** Tahapan ini merupakan penyiapan lingkungan pengujian yang akan menggunakan LEMP (Linux, Nginx, MySQL, PHP) sebagai topologinya. LAMP dipilih karena telah terbukti sebagai lingkungan yang handal dan sering digunakan dalam pengembangan web.
2. **Konfigurasi Amazon S3:** Tahapan ini merupakan langkah penting dalam penyiapan lingkungan pengujian. Amazon S3 (*Simple Storage Service*) adalah layanan penyimpanan *cloud* yang disediakan oleh Amazon Inc.
3. **Instalasi dan Konfigurasi Aplikasi sisi Klien:** Aplikasi sederhana laravel akan dijadikan sebagai *Client* yang akan secara otomatis menjalankan *upload* dan *download file* pada media pengujian.
4. **Simulasi dengan Laravel:** Pengujian akan dilakukan pada sisi klien, dimana aplikasi laravel akan melakukan aksi *upload* dan *download* pada kedua media pengujian. Aplikasi laravel akan mencatat dan mengumpulkan data pada aktifitas mesin dan data yang didapatkan akan digunakan pada proses penilaian dan pengambilan kesimpulan.
5. **Pengambilan Kesimpulan:** Data yang didapat dari proses Simulasi akan diolah dengan metode *Quality Of Service* (QoS). Hasil dari perhitungan

tersebut dapat digunakan untuk menyimpulkan media apa yang terbaik untuk menyimpan media.



Gambar 2. Topologi Lingkungan Pengujian

Pada gambar 2 memperlihatkan topologi yang kami gunakan pada lingkungan pengujian, dimana *Web server* Ubuntu 20.04.4 dijadikan sebagai layanan yang menjalankan aplikasi klien sederhana dengan menggunakan Laravel. Aplikasi secara paralel terus melakukan pemantauan pada transmisi media. dan *unit testing* pada Laravel melakukan simulasi *upload* dan *download* dengan alur yang sama pada kedua media pengujian.



Gambar 3. Spesifikasi Virtual Machine

Pada gambar 3 menampilkan spesifikasi *virtual machine* yang kami gunakan untuk melakukan pengujian. *Virtual machine* ini akan kami *install* LEMP untuk dijadikan sebagai lingkungan pengujian yang terkontrol.

1. Tahap satu Instalasi web server

Kami menggunakan operasi sistem berbasis linux yaitu Ubuntu 20. Tahap pertama pada persiapan kali ini adalah melakukan instalasi *web server*, kami menggunakan Linux, Nginx (Engine-X),

MySQL/MariaDB, dan PHP untuk mendukung proses pengujian.

```

root@jurnal-purpose:~#
root@jurnal-purpose:~#
root@jurnal-purpose:~#
root@jurnal-purpose:~# sudo apt install nginx
  
```

Gambar 4. Install Nginx

Pada gambar 4 lakukan penginstalan Nginx dengan perintah *sudo apt install nginx*.

```

root@jurnal-purpose:/var/www/laravel-jurnal-purpose#
root@jurnal-purpose:/var/www/laravel-jurnal-purpose#
root@jurnal-purpose:/var/www/laravel-jurnal-purpose#
root@jurnal-purpose:/var/www/laravel-jurnal-purpose#
root@jurnal-purpose:/var/www/laravel-jurnal-purpose#
root@jurnal-purpose:/var/www/laravel-jurnal-purpose# sudo apt install php8.3-mysql
  
```

Gambar 5. Install MysqlClient

Pada gambar 5 lakukan penginstalan *Mysql Client* dengan perintah *sudo apt install php8.1 php8.1-fpm*. Karena kami menggunakan RDS Database dari AWS, maka kami tidak memerlukan *MYSQL server* pada lingkungan ini. *Mysql Client* kami butuhkan untuk berinteraksi dengan *Aws RDS*.

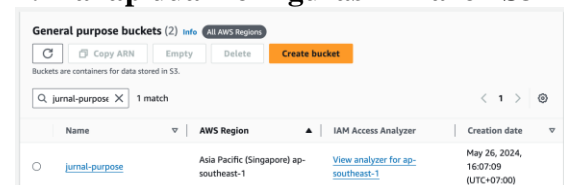
```

root@jurnal-purpose:/var/www/laravel-jurnal-purpose#
root@jurnal-purpose:/var/www/laravel-jurnal-purpose#
root@jurnal-purpose:/var/www/laravel-jurnal-purpose#
root@jurnal-purpose:/var/www/laravel-jurnal-purpose#
root@jurnal-purpose:/var/www/laravel-jurnal-purpose#
root@jurnal-purpose:/var/www/laravel-jurnal-purpose# sudo apt install php8.3 php8.3-fpm
  
```

Gambar 6. Install Php 8.3

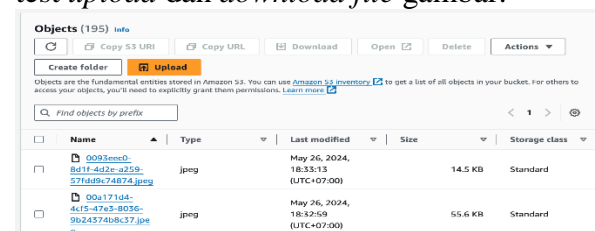
Pada gambar 6 lakukan penginstalan PHP 8.3 dengan melakukan perintah *sudo apt install php8.3 php8.3-fpm*.

2. Tahap dua Konfigurasi Amazon S3



Gambar 7. Menyiapkan Bucket Pada Amazon S3

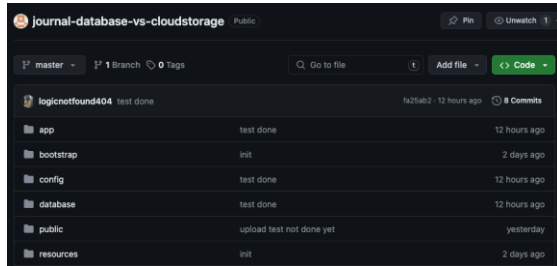
Pada gambar 7, untuk melakukan pengujian pada *Cloud storage* kami perlu membuat *Bucket* baru pada Amazone S3 yang akan kami gunakan untuk melakukan test *upload* dan *download file* gambar.



Gambar 8. Menyiapkan *Sample File*

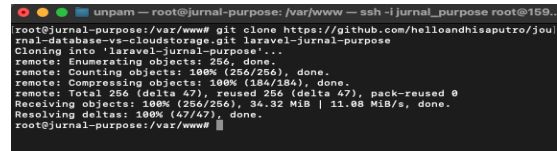
Pada gambar 8, kami menyediakan *file* gambar pada Amazon S3 dengan berbagai ukuran berjumlah 195 *Files* yang akan kami gunakan untuk simulasi *download*. Sebanyak 195 *Files* yang identik kami simpan juga pada aplikasi Laravel untuk melakukan simulasi *upload*.

3. Instalasi dan konfigurasi Aplikasi Laravel 11



Gambar 9. Menyiapkan Kode Untuk Test

Pada gambar 9, kami sudah menyiapkan aplikasi dengan laravel 11 dan *repository* aplikasi telah kami *upload* pada *github* yang dapat diakses melalui tautan berikut:



Gambar 10. Konfigurasi Aplikasi Laravel

Pada gambar 10, *Clone* aplikasi yang ada di Github ke lingkungan pengujian dan lakukan konfigurasi pada aplikasi. Konfigurasi *Database* dan konfigurasi akun Amazon S3.

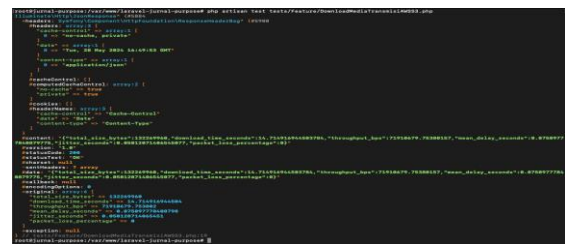
4. Simulasi Dengan Laravel

Pengumpulan data dilakukan dengan membuat simulasi *upload* dan *download*. Agar fungsi berjalan secara otomatis, kami menggunakan fitur *unit testing* pada laravel. Kami membuat 4 *unit testing* dengan detail sebagai berikut:

Tabel 2. *Unit Test Pengumpulan Data*

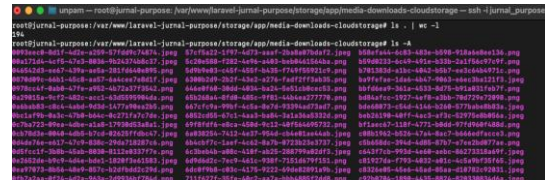
No	Simulasi	Keterangan
1	Amazon S3 Download	Simulasi yang digunakan untuk melakukan <i>download</i> dari Amazone S3 ke <i>local</i> mesin

2	Amazon S3 Upload	Simulasi yang digunakan untuk melakukan <i>upload</i> dari <i>local</i> mesin ke Amazon S3
3	Mysql Download	Simulasi yang digunakan untuk melakukan <i>download</i> dari <i>Column</i> di Mysql Database ke <i>local</i> mesin
4	Mysql Upload	Simulasi yang digunakan untuk melakukan <i>upload</i> dari <i>local</i> mesin ke Mysql Database



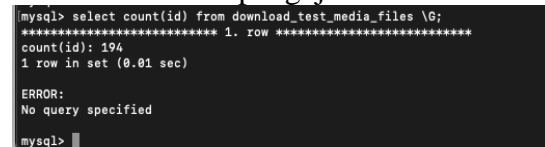
Gambar 11. Pengujian Pertama

Pada gambar 11 pengujian pertama yaitu dengan melakukan simulasi *download* dari Amazon S3 ke *local* mesin dengan menjalankan *unit test* yang ada di *file DownloadMediaTransmisiAWSS3.php*. Jika seluruh file berhasil didownload, *unit testing* akan menampilkan hasil QoS dan akan menaruh hasilnya pada file log dengan nama *qos.log* di *storage* aplikasi laravel.



Gambar 12. Seluruh File Berhasil Terdownload

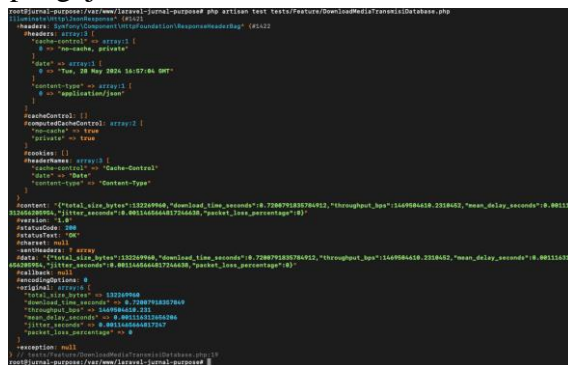
Pada gambar 12 harus memastikan jumlah *file* seluruhnya berhasil *terdownload* dari pengujian 1.



Gambar 13. Sample Data Percobaan 2

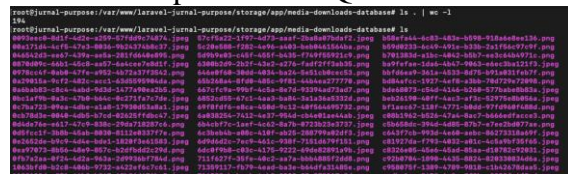
Pada gambar 13, kami menyiapkan 194 *row data* pada tabel *download_test_media_files* yang berisi *column file_blob* bertipe *data longblob* dan

memiliki *file* gambar dengan ukuran yang berbeda-beda, seluruh *file* yang digunakan identik dengan *file* yang digunakan untuk pengujian 1.



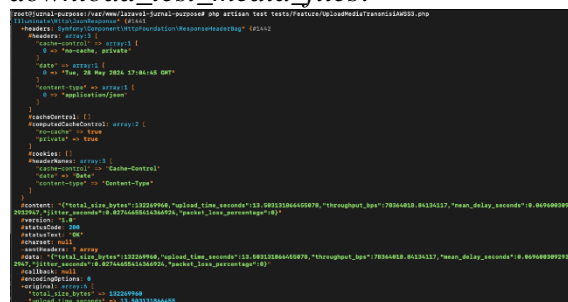
Gambar 14. Jalankan Pengujian Kedua

Pada gambar 14, lakukan pengujian kedua dengan *unit testing* pada *file DownloadMediaTransmisiDatabase.php*. Setelah seluruh *file* yang berada pada tabel *download_test_media_files* terdownload pada mesin maka fungsi pada *unit test* akan menampilkan hasil QoS.



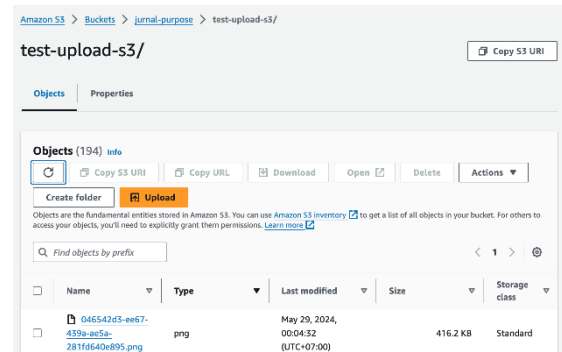
Gambar 15. Jumlah File Yang Terdownload r2

Pada gambar 15, kami memastikan bahwa jumlah *file* yang didownload sesuai dengan jumlah data pada tabel *download_test_media_files*.



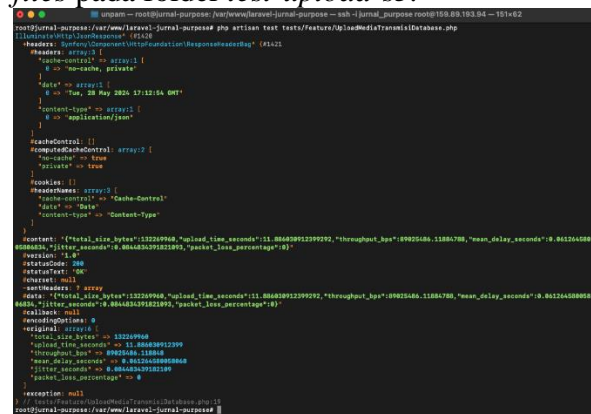
Gambar 16. Pengujian Ketiga

Pada gambar 16, lakukan pengujian ketiga, *Upload file* dari *local* mesin ke Amazon S3.



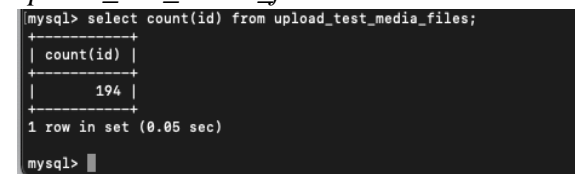
Gambar 17. Jumlah File Yang TerUpload S3

Pada gambar 17, kami memastikan bahwa *file* yang terupload berjumlah 194 *files* pada folder *test-upload-s3*.



Gambar 18. Pengujian Keempat

Pada gambar 18, lakukan pengujian keempat, *Upload file* dari *local* mesin ke *database* dengan tabel *upload_test_media_files*.



Gambar 19. Jumlah File Terupload Database

Pada gambar 19, kami memastikan bahwa data yang terupload berjumlah 194 *files* pada tabel *upload_test_media_files*.

```

root@jurnal-purpose:/var/www/localhost/htdocs/wordpress/wp-content/uploads/2024/05/2024-05-20-16:57:04# cat /var/www/localhost/htdocs/wordpress/wp-content/uploads/2024/05/2024-05-20-16:57:04-qos.log
{"jitter": {
  "mean_delay_seconds": 0.0012618888888888889,
  "jitter_seconds": 0.0012618888888888889
},
"packet_loss": {
  "total_files": 10,
  "lost_files": 0,
  "packet_loss_percentage": 0
},
"throughput": {
  "total_size_bytes": 132269960,
  "download_time_seconds": 14.7149,
  "throughput": {
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0
  }
},
"delay": {
  "total_packets": 10,
  "total_time_seconds": 0.0751,
  "mean_delay_seconds": 0.0751
},
"jitter": {
  "mean_delay_seconds": 0.0751,
  "jitter_seconds": 0.0751
},
"packet_loss": {
  "total_packets": 10,
  "lost_packets": 0,
  "packet_loss_percentage": 0
}
}

2024-05-20 16:57:04# cat /var/www/localhost/htdocs/wordpress/wp-content/uploads/2024/05/2024-05-20-16:57:04-qos.log
{"jitter": {
  "mean_delay_seconds": 0.0012618888888888889,
  "jitter_seconds": 0.0012618888888888889
},
"packet_loss": {
  "total_files": 10,
  "lost_files": 0,
  "packet_loss_percentage": 0
},
"throughput": {
  "total_size_bytes": 132269960,
  "download_time_seconds": 14.7149,
  "throughput": {
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0,
    "bytes_per_second": 9000000.0
  }
},
"delay": {
  "total_packets": 10,
  "total_time_seconds": 0.0751,
  "mean_delay_seconds": 0.0751
},
"jitter": {
  "mean_delay_seconds": 0.0751,
  "jitter_seconds": 0.0751
},
"packet_loss": {
  "total_packets": 10,
  "lost_packets": 0,
  "packet_loss_percentage": 0
}
}

```

Gambar 20. File QoS Log Pengumpulan Data

Pada gambar 20, setelah seluruh pengujian selesai maka seluruh hasil dari proses pengujian akan disimpan pada file *qos.log*

HASIL DAN PEMBAHASAN

Berdasarkan hasil pengujian *Quality of Service* (QoS) pada proses *upload* dan *download* data antara MySQL dan Amazon S3, kami telah mengumpulkan dan menganalisis metrik-metrik kinerja penting seperti *throughput*, *delay*, *jitter*, dan *packet loss*.

Berikut adalah kesimpulan yang kami dapatkan dari hasil pengujian tersebut:

Tabel 3. Hasil Percobaan Download

#	Kategori	MySQL	Amazon S3
1	Throughput (Kbps)	70.225	1.435.063
2	Jitter (second)	0.0501	0.0011
3	Delay (second)	0.0751	0.0011
4	Packet Loss (%)	0	0
5	Download Size (Bytes)	132.269.960	132.269.960
6	Download Time (second)	14.7149	0.7201

Tabel 4. Hasil Percobaan Upload

#	Kategori	MySQL	Amazon S3
1	Throughput (Kbps)	86.938	76.527
2	Jitter (second)	0.0845	0.0274
3	Delay (second)	0.0613	0.0696
4	Packet Loss (%)	0	0

5	Upload Size (Bytes)	132.269.960	132.269.960
6	Upload Time (second)	11.8860	13.5031

Pada percobaan ini, kami melakukan evaluasi kinerja antara MySQL dan Amazon S3 dalam konteks transmisi data media pada sistem jaringan. Evaluasi dilakukan dengan mempertimbangkan beberapa parameter kinerja utama seperti *throughput*, *delay*, *jitter*, dan *packet loss* untuk menganalisis efisiensi dan keandalan kedua platform penyimpanan ini. Dari hasil pengujian di atas, kami dapat menyimpulkan bahwa:

Throughput:

MySQL menunjukkan *throughput* yang jauh lebih tinggi dibandingkan dengan Amazon S3 baik dalam proses *download* maupun *upload*. Ini menunjukkan bahwa MySQL dapat mentransfer data dengan kecepatan yang lebih tinggi.

Delay:

Delay rata-rata untuk MySQL juga lebih rendah dibandingkan dengan Amazon S3 dalam kedua proses. Ini berarti bahwa waktu yang dibutuhkan untuk mentransmisikan paket data lebih cepat pada MySQL.

Jitter:

Jitter pada MySQL sangat rendah dibandingkan dengan Amazon S3 dalam proses *download*. Namun *jitter* pada *upload* MySQL lebih tinggi daripada S3. *Jitter* yang lebih rendah menunjukkan konsistensi dalam waktu pengiriman paket data.

Packet Loss:

Tidak ada *packet loss* yang tercatat dalam kedua pengujian untuk kedua layanan, menunjukkan bahwa kedua layanan ini sangat andal dalam hal keandalan data.

SIMPULAN

Secara keseluruhan, MySQL menunjukkan kinerja yang lebih baik dalam hal *throughput* dan *delay*

dibandingkan dengan Amazon S3. Namun, *jitter* pada *upload* MySQL lebih tinggi, yang menunjukkan bahwa ada variasi yang lebih besar dalam waktu pengiriman paket data.

Kami merekomendasikan untuk mempertimbangkan kebutuhan spesifik aplikasi Anda dalam memilih antara MySQL dan Amazon S3. Jika kecepatan transmisi dan konsistensi waktu pengiriman sangat penting, MySQL tampaknya menjadi pilihan yang lebih baik. Namun, jika skalabilitas dan pengelolaan data dalam skala besar menjadi prioritas, Amazon S3 masih merupakan opsi yang sangat baik. Perlu juga dicatat bahwa pengujian ini dilakukan dengan dataset terbatas, dan pengujian skala besar mungkin memberikan hasil yang berbeda.

Untuk hasil yang optimal, pertimbangkan untuk melakukan pengujian lebih lanjut dengan dataset yang lebih besar untuk mengevaluasi kinerja dalam kondisi yang lebih mendekati penggunaan sebenarnya

DAFTAR PUSTAKA

- B. Zhang, J. Wang. (2013). *MySQL: A Relational Database Management System*. *Journal of Database Systems*, 27(4), 245-259. DOI: 10.1007/s00778-013-0312-9.
- Brown, R., & Jones, S. (2018). *Laravel: A Comprehensive PHP Framework for Web Development*. *Journal of Web Development*, 12(2), 112-125. DOI: 10.1080/1542868X.2018.1435742.
- Kulkarni, A., Joshi, A., & Jadhav, S. (2018). Comparative Study of Cloud Storage Services - Amazon S3, Google Cloud Storage, and Microsoft Azure Storage. *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*. DOI: 10.1109/ICCTCT.2018.8550772.
- Laravel. *Laravel is a Trademark of Laravel Holdings Inc*. Diakses pada tanggal 3 Juni 2024 pukul 14:45 WIB dari <https://laravel.com/docs/11.x/testing#main-content>.
- Lin, J., Huang, S., & Wang, C. (2016). Research on Storing Images in MySQL Database. *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*, 150-153. DOI:10.1109/BigDataSecurity-HPSC-IDS.2016.39.
- Nanda, A. (2015). How to store and retrieve images from MySQL database using PHP. *International Journal of Innovative Research in Computer and Communication Engineering*, 3(7), 6218-6223.
- Persico, V., Montieri, A., & Pescapè, A. (2016). On the Network Performance of Amazon S3 Cloud-Storage Service. *2016 5th IEEE International Conference on Cloud Networking (Cloudnet)*. DOI: 10.1109/CloudNet.2016.16.
- Persico, V., Montieri, A., & Pescapè, A. (2016). On the Network Performance of Amazon S3 Cloud-Storage Service. In *2016 5th IEEE International Conference on Cloud Networking (Cloudnet)* (pp. 245-259). IEEE. DOI: 10.1109/CloudNet.2016.16.
- Rath, M., et al. (2017). *Quality of Service (QoS) in Multimedia Networks*. *International Journal of Computer Networks & Communications*, 9(2), 45-60. DOI: 10.5121/ijcnc.2017.9204.
- Siva, S., & Adalarasu, K. (2013). Efficient storage and retrieval of images in MySQL database using OpenCV. *International Journal of Computer Science and Mobile Computing*, 2(6), 182-191.
- Smith, J., & Johnson, A. (2010). *MySQL: A Relational Database Management*

System. *Journal of Database Management*, 21(3), 45-58. DOI: 10.4018/jdm.2010070103

Valerio Persico, Antonio Montieri, Antonio Pescapè. (2016). *On the Network Performance of Amazon S3 Cloud-Storage Service*. 2016 5th IEEE International Conference on Cloud Networking (Cloudnet), 03-05 October 2016, DOI: 10.1109/CloudNet.2016.16.

Yu L, Zhou X, Ren S, Xu L. The design and implementation of web servers in LEMP. *Journal of Network and Computer Applications*. 2020;120:102-112. doi:10.1016/j.jnca.2020.102112.