

# Pengujian Kinerja Firebase Pada Aplikasi Layanan Bimbingan Magang Berbasis Mobile

Indrawati<sup>1</sup>, Afla Nevrisa<sup>2</sup>, Radhyatammardhiyah<sup>3</sup>, Novira Dwina<sup>4</sup>, Amir. D<sup>5</sup>

<sup>1,2,3,4</sup> Tenik Rekayasa Komputer Jaringan Politeknik Negeri Lhokseumawe

<sup>5</sup>Teknik Rekayasa Jaringan Telekomunikasi Politeknik Negeri Lhokseumawe

Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA

<sup>1</sup>indrawati@pnl.ac.id

**Abstrak** Permasalahan yang terjadi pada penggunaan aplikasi magang adalah menjamin efektifitas dan kinerja aplikasi dalam melakukan proses mengelola, menyimpan, mengumpulkan, mengirim pesan atau data ke perangkat pengguna. Untuk mengakomodasi permasalahan tersebut digunakan *Firestore*. *Firestore* adalah platform yang disediakan oleh Google untuk membangun dan mengelola aplikasi web dan mobile. Rancang bangun aplikasi layanan bimbingan magang terdiri dari admin, dosen/mentor pembimbing magang, dan mahasiswa. Selanjutnya dilakukan pengujian kinerja hasil rancangan. Uji kinerja menggunakan metode *Google Cloud Platform*. Beberapa hasil pengujian antara lain; jumlah koneksi aktif tertinggi dalam satu jam mencapai 20 user, hasil ini menunjukkan jumlah perangkat *client* terhubung ke database secara *realtime*. Tindakan akses dokumen sebanyak 9 akses yang menunjukkan jumlah tindakan membaca atau mengakses data yang tersimpan pada database sebanyak 9 kali. *Request latency* per database tertinggi sebesar 0,06 detik, nilai ini lebih kecil dari nilai kategori terendah sebesar 100 ms, sehingga kinerja server dapat dinilai memberikan respon yang mulus dan cepat kepada setiap user, karena waktu tunggu sangat singkat saat merespon permintaan data. *Index write* di *Firestore* terjadi pada rentang waktu jam 7.25 PM-7.30 PM dengan jumlah *index write* sebesar 7,4/s. Hasil ini menunjukkan kegiatan penulisan dan perubahan data di database lonjakan rendah. Selanjutnya tindakan *document write* tertinggi yang dimonitoring selama 1 jam sebesar 0,365/s, terjadi pada rentang waktu 7.25 PM-7.30 PM. Lonjakan ini menunjukkan bahwa pada periode waktu ini jumlah operasi tulis meningkat. Puncak *snapshot* yang aktif tertinggi berada pada rentang waktu jam 7.25PM-7.35PM dengan nilai *snapshot* aktif sebesar 17/s. *Snapshot* tertinggi mengindikasikan adanya aktivitas user yang tinggi saat mengakses aplikasi.

**Kata kunci**— Sistem, bimbingan, firebase, kinerja, layanan, magang, user

**Abstract**— The problem that occurs in the use of internship applications is to ensure the effectiveness and performance of the application in carrying out the process of managing, storing, collecting, sending messages or data to the user's device. To accommodate these problems, Firestore is used. Firestore is a platform provided by Google to build and manage web and mobile applications. The design of the internship guidance service application consists of admins, lecturers/internship mentors, and students. Furthermore, performance testing of the design results is carried out. Performance testing uses the Google Cloud Platform method. Some of the test results include; the highest number of active connections in one hour reaches 20 users, these results indicate the number of client devices connected to the database in real time. Document access actions as many as 9 accesses which indicate the number of actions to read or access data stored in the database as many as 9 times. The highest request latency per database is 0.06 seconds, this value is smaller than the lowest category value of 100 ms, so that server performance can be assessed as providing a smooth and fast response to each user, because the waiting time is very short when responding to data requests. The write index in Firestore occurs in the time range of 7.25 PM-7.30 PM with a write index of 7.4/s. These results show that the activity of writing and changing data in the database is low spike. Furthermore, the highest document write action monitored for 1 hour is 0.365/s, occurring in the time range 7.25 PM-7.30 PM. This spike indicates that during this time period the number of write operations increases. The highest active snapshot peak is in the time range 7.25PM-7.35PM with an active snapshot value of 17/s. The highest snapshot indicates high user activity when accessing the application.

**Keywords:** System, guidance, firebase, service, performance, internship, user

## I. PENDAHULUAN

### A. Latar Belakang Masalah

Permasalahan yang terjadi pada penggunaan aplikasi magang adalah menjamin efektifitas dan kinerja aplikasi dalam melakukan proses mengelola, menyimpan, mengumpulkan, mengirim pesan atau data ke perangkat pengguna dan menganalisis interaksi pengguna dengan aplikasi. Efektifitas dan kinerja aplikasi yang baik harus dikelola secara optimal menggunakan platform yang sesuai dengan kebutuhan aplikasi. Banyak platform yang tersedia dilapangan telah diteliti oleh peneliti sebelumnya [1][2][3], salah satunya adalah *firebase*.

*Firestore* adalah sebuah platform yang disediakan oleh google untuk membantu pengembang dalam membangun dan mengelola aplikasi web dan mobile. *Firestore* menawarkan

berbagai layanan dan menyediakan solusi lengkap untuk banyak aspek pengembangan aplikasi, termasuk *backend*, autentikasi, penyimpanan data, dan analitik. Pada artikel ini akan dijelaskan penggunaan *firebase* pada aplikasi layanan bimbingan magang. Aplikasi layanan bimbingan magang dirancang dan dibangun dari beberapa bagian *user interface* yaitu; admin, dosen pembimbing/mentor, dan mahasiswa.

Aplikasi ini dapat memberikan layanan kepada user dari saat mendaftar magang sampai tahap proses evaluasi penilaian, dan laporan magang mahasiswa. Untuk mengamati kinerja *firebase* terhadap penggunaan aplikasi layanan magang mahasiswa, maka dilakukan uji kinerja.

### B. Uji kinerja

Pada aplikasi ini antara lain uji; *conection client* (koneksi klien yang aktif), *document reads* (akses dokumen), *request latency per database* (waktu respon per database), *document writes* (optimasi kinerja untuk penulisan dokumen), *index writes* dan *snapshot*. Pengamatan kinerja *Firestore* menggunakan metode *Google Cloud Platform* (GCP).

Uji kinerja aplikasi dilakukan dengan mengamati kinerja *Firestore* dengan scenario sebanyak 28 user yang melakukan aktifitas akses pada layanan aplikasi magang. Aktifitasnya dapat berupa akses mendaftar, *upload* dan *download* dokumen. Pengujian dilakukan pada Jurusan Teknik Informatika dan Komputer Politeknik Negeri Lhokseumawe.

C. Tinjauan Pustaka

1) *Firestore*; adalah *platform* pengembangan aplikasi yang disediakan oleh *google* yang menyediakan berbagai layanan untuk membantu pengembang dalam membangun, mengelola, dan menskalakan aplikasi mereka [4][5]. *Firestore* mencakup beberapa komponen utama, antara lain:

2) *Firestore Realtime*; terdiri dari beberapa bagian secara rinci dijelaskan berikut ini.

a) *Database NoSQL*, database ini menyimpan dan menyinkronkan data dalam waktu nyata. Data disimpan sebagai *JSON* dan dapat diakses oleh semua *client* secara bersamaan. Mendukung penyimpanan dan pembaruan data secara *real-time*.

b) *Cloud Firestore*, ini merupakan *database NoSQL* berbasis dokumen yang lebih kuat dan fleksibel dibandingkan *Realtime Database*. Mendukung *query* yang lebih kompleks, transaksi, dan *batch writes*. Data disimpan dalam koleksi dan dokumen.

3) *Firestore Authentication*; ini merupakan layanan untuk mengelola otentikasi *user* (pengguna) menggunakan email/sandi, *open authorization provider* seperti *Google*, *Facebook*, *Twitter*, *GitHub*, dan lainnya. Mempermudah pengelolaan otentikasi dan otorisasi pengguna.

4). *Firestore Cloud Messaging* (FCM); layanan yang memungkinkan pengiriman pesan dan notifikasi *push* ke aplikasi di berbagai *platform*. Mendukung pengiriman pesan ke perangkat individu, kelompok perangkat, atau topik.

5). *Firestore Hosting*; merupakan layanan *hosting* yang cepat dan aman untuk aplikasi web statis dan dinamis. Mendukung *SSL* (*Secure Sockets Layer*) secara otomatis dan menyediakan penyebaran satu klik.

6). *Firestore Cloud Functions*; merupakan *platform* untuk menjalankan fungsi *backend* dalam menanggapi peristiwa dari *firebase* dan *google cloud*. Fungsi dijalankan di infrastruktur *google cloud* tanpa perlu mengelola server.

7). *Firestore Analytics*; alat analitik gratis yang menyediakan wawasan tentang penggunaan aplikasi dan keterlibatan pengguna. Mengumpulkan data secara otomatis dan menyediakan laporan yang dapat disesuaikan.

8. *Firestore Crashlytics*; alat pelaporan kerusakan yang membantu pengembang melacak, memprioritaskan, dan memperbaiki masalah stabilitas aplikasi. Menyediakan laporan terperinci tentang kerusakan dan masalah performa.

9. *Firestore Performance Monitoring*; alat untuk memantau kinerja aplikasi dan memahami masalah yang mungkin mempengaruhi pengalaman pengguna. Mengumpulkan data tentang *latency* jaringan, waktu respons, dan kinerja render aplikasi.

D. Kinerja *Firestore*

Monitoring kinerja *Firestore* dilakukan untuk memantau beberapa komponen, antara lain;

1). Alat. Dilakukan untuk memantau kinerja aplikasi dan mengidentifikasi masalah yang mempengaruhi pengalaman pengguna.

2). *Fitur*. Dilakukan untuk memonitoring beberapa fitur, antara lain;

a) *Trace Metric*: Mengukur waktu yang dibutuhkan untuk operasi tertentu di aplikasi, seperti peluncuran aplikasi atau pemuatan layar.

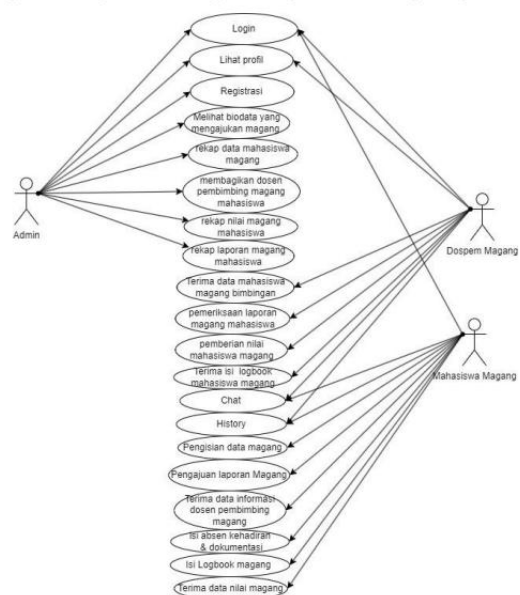
b). *HTTP/S Network Monitoring*: Memantau permintaan jaringan dan mengidentifikasi latensi, waktu respons, dan kegagalan.

c) *Custom Traces*: Pengembang dapat menambahkan khusus untuk mengukur kinerja bagian tertentu dari aplikasi.

II. METODOLOGI PENELITIAN

A. Diagram use case Layanan Magang

Aplikasi layanan magang mahasiswa dirancang seperti gambar 1[3]. Gambar ini menggambarkan interaksi antara pengguna (aktor) dengan sistem atau aplikasi, yang disebut use cases (kasus penggunaan). Pada aplikasi magang, diagram use case akan menunjukkan bagaimana pengguna utama, seperti mahasiswa, dosen/mentor pembimbing magang, dan administrator, berinteraksi dengan fitur-fitur aplikasi.



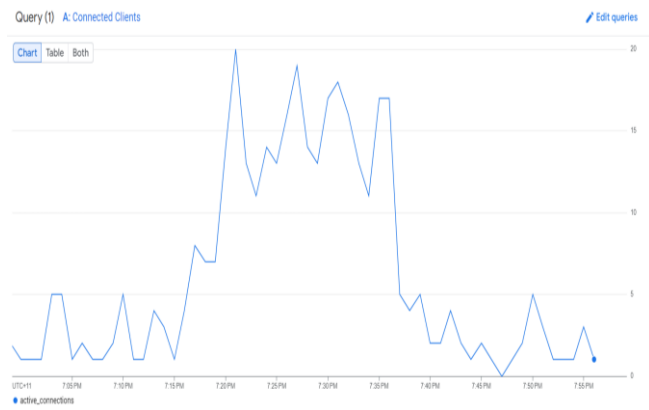
Gambar 1. Diagram use case aplikasi magang mahasiswa

Gambar 1 menunjukkan keterkaitan antara berbagai aspek *firebase* dan bagaimana mereka berkontribusi untuk menyediakan solusi komprehensif bagi pengguna [6]. Uji kinerja *firebase* dilakukan terhadap beberapa parameter, antara lain; *connection client* (koneksi klien yang aktif), *document reads* (akses dokumen), *request latency per database* (waktu respon per database), *document writes* (optimasi kinerja untuk penulisan dokumen), *index writes* dan *snapshot*.

### III. HASIL DAN PEMBAHASAN

#### A. Connection Client

Menggambarkan saat sejumlah *user* secara aktif terhubung dan menggunakan layanan aplikasi magang. Saat *user* melakukan *login*, akun mereka dianggap aktif dan sistem dapat mengidentifikasi serta memberikan akses sesuai dengan izin yang dimiliki oleh akun tersebut. Secara umum, *active client login* adalah proses autentikasi yang memungkinkan pengguna mengakses layanan atau aplikasi secara sah [7][8][9].



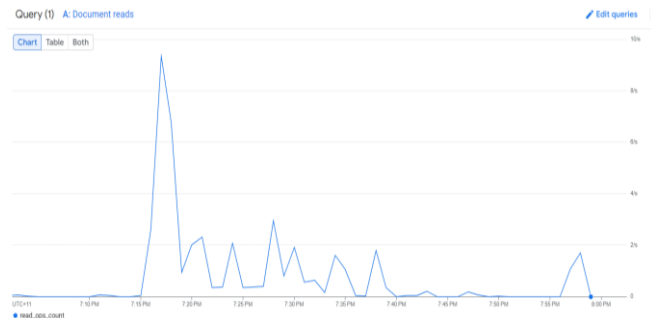
Gambar 2. Grafik hasil monitoring jumlah koneksi aktif

Berdasarkan hasil pemantauan dengan alat monitoring *Google Cloud Platform*, disebut *GCP*, seperti terlihat pada grafik gambar 2. Diketahui bahwa dalam rentang waktu satu jam dimulai dari jam 7.00 PM sampai dengan jam 8.00 PM, diketahui bahwa terjadi fluktuasi jumlah koneksi dalam periode waktu tersebut sebanyak 101 koneksi. Dengan *active client login* tertinggi terjadi pada rentang waktu 7.20.PM-7.25 PM dimana *active client login* tertinggi sebanyak 20 *client*. Ini menunjukkan bahwa jumlah perangkat atau klien yang terhubung ke *database* secara *real-time*. Setiap kali ada data baru atau perubahan pada *database*, semua *client* yang terhubung akan secara otomatis diperbarui.

#### B. Document Reads

Mengacu pada tindakan membaca atau mengakses data yang disimpan dalam sebuah dokumen database khususnya dalam *Data Base Managemen System* (DBMS) [10], monitoring terhadap aktifitas ini juga menggunakan *GCP*. Berdasarkan hasil *monitoring document read*, seperti

diperlihatkan pada gambar 3, diberikan informasi bahwa jumlah pembacaan *document reads* berfluktuasi dalam rentang pengamatan satu jam. *Document reads* menunjukkan bahwa berapa kali data di dalam sebuah dokumen dibaca atau diakses. setiap kali aplikasi magang mahasiswa mengambil data dari *firestore*, itu akan dihitung sebagai satu bacaan dokumen.



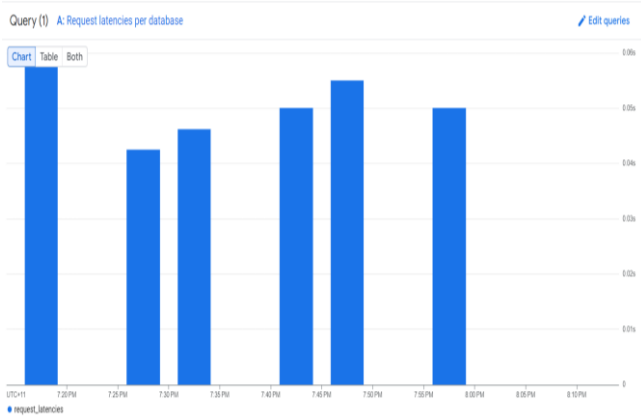
Gambar 3. Grafik hasil monitoring jumlah pembacaan dokumen

Berdasarkan grafik gambar 3 diketahui banyaknya *document reads* dalam rentang satu jam, mulai dari jam 7.00 PM– 8.00 PM. Jumlah *document reads* (pembacaan dokumen) sebanyak 23 kali diakses per jam, grafik di atas mengindikasikan waktu-waktu puncak atau sepi aktivitas pengguna. Fluktuasi penggunaan tertinggi *document reads* berada pada rentang waktu monitoring jam 7.15 PM-7.20 PM dengan jumlah pembacaan data sebanyak 9 akses selama rentang waktu tersebut.

#### C. Request Latency per Database

Adalah merupakan metrik penting untuk mengukur kinerja dan responsivitas *database* atau waktu yang dibutuhkan oleh sebuah *server database* untuk memproses dan memberikan respons terhadap suatu permintaan data dari sebuah aplikasi [11][12][13].

Berdasarkan gambar 4, disajikan grafik yang memberikan gambaran tentang *request latency per database*. *Request latency per database* dibutuhkan oleh setiap *database* dalam aplikasi sistem layanan untuk merespons permintaan data. Setiap batang pada grafik mewakili satu *database* dan tinggi batang menunjukkan seberapa lama waktu tunggu untuk *database* tersebut membutuhkan waktu untuk memproses atau merespon sebuah permintaan.

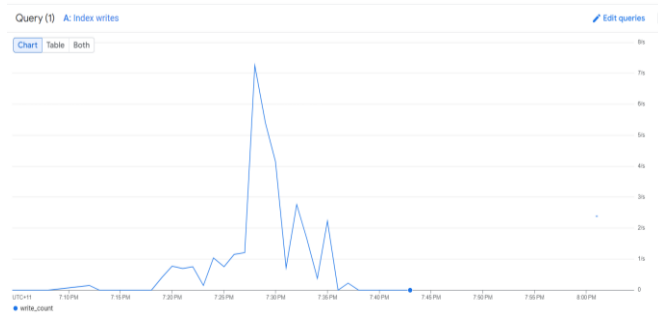


Gambar 4. Grafik hasil monitoring waktu tunggu

*Request latency per database* rata-rata server untuk merespon permintaan layanan berdasarkan grafik pada gambar 4 diketahui sebesar 0,025 detik, sedangkan *request latency per database* tertinggi selama rentang waktu pengamatan sebesar 0,06 detik. Nilai *request latency* rata-rata dan nilai *request latency* tertinggi berada pada nilai ketetapan *latency* terendah yaitu dibawah 100 milidetik. Nilai ini menunjukkan bahwa server dapat memberikan respon yang mulus dan cepat kepada setiap user, karena waktu tungguanya sangat singkat saat merespon permintaan data.

**D. Index Writes**

Merupakan sebuah metric yang merujuk pada ukuran seberapa sering data baru ditulis atau diubah dalam sebuah sistem, perubahan data diartikan sebagai proses menghapus, menambah, mengedit data *firestore database*. Metrik ini merupakan indikator penting untuk memahami tingkat aktivitas dan beban kerja pada sistem tersebut.



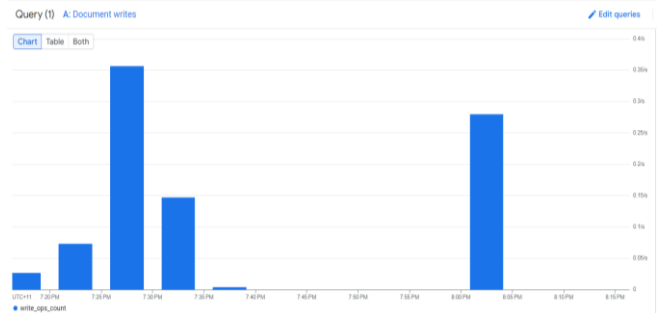
Gambar 5. Grafik hasil monitoring indeks write

Berdasarkan data hasil monitoring yang ditunjukkan gambar 5 diketahui bahwa rata-rata data ditulis dan diubah sebanyak 11 kali selama kurung waktu selama satu jam. Lonjakan tertinggi perubahan dan penulisan data di *firestore* terjadi pada rentang waktu antara jam 7.25PM - 7.30PM dengan jumlah *indeks write* sebesar 7,4/s. *Indeks write* yang tinggi dapat memengaruhi kinerja sistem, terutama pada operasi tulis. Ini bisa menyebabkan penurunan kecepatan respon server,

terutama jika sistem tidak dirancang untuk menangani beban tulis yang tinggi.

**E. Document Writes**

Grafik pada gambar 6 merupakan hasil pengujian *document writes* seperti kegiatan menambah data baru, mengubah data yang sudah ada, serta menghapus data. Pengujian ini memberikan gambaran *visual* tentang seberapa sering data baru ditulis dalam sebuah sistem dalam periode waktu satu jam. Dalam konteks ini, dokumen bisa merujuk pada berbagai jenis data, seperti catatan dalam database, *file* atau *entri log* [12].



Gambar 6. Grafik document writes

Berdasarkan grafik gambar 6, diperoleh informasi bahwa jumlah total operasi *document writes* pada *firestore* sebesar 0,895/s, sedangkan rata-rata operasi *document writes* sebesar 0,149/s. Lonjakan operasi *document writes* yang dimonitoring selama 1 jam yaitu sebesar 0,365/s. Pada grafik tersaji beberapa informasi.

- 1). Puncak aktivitas *document writes*. Pada grafik gambar 6, terdapat beberapa periode waktu di mana jumlah *document writes* meningkat secara signifikan. Ini mengindikasikan adanya aktivitas yang intensif dalam menulis data baru pada waktu-waktu tersebut.
- 2). Aktivitas *document writes* yang rendah. Di antara puncak-puncak aktivitas, terdapat periode di mana jumlah operasi *document writes* relatif rendah. Ini menunjukkan adanya periode di mana sistem dan aplikasi relatif tenang atau penggunaannya relative sepi pada aktifitas *document writes*.

**F. Snapshot Listener**

Adalah fitur dalam *firebase realtime database* yang memungkinkan terjadinya pembaruan data secara real-time dari database. Ketika terjadi perubahan data maka *firebase* akan mengirimkan *snapshot* dari data tersebut ke *listener snapshot*. Pengiriman ini adalah gambaran instan dari data pada saat perubahan terjadi.



Gambar 7. Hasil monitoring jumlah perubahan data pada database *firestore*

Grafik gambar 7, memberikan gambaran visual mengenai tingkat aktivitas dari suatu sistem yang menggunakan *snapshot listener*. *Snapshot listener* adalah mekanisme yang digunakan untuk memantau perubahan data secara real-time dalam sebuah aplikasi.

Grafik gambar 7 menggambarkan tentang fluktuasi jumlah *Snapshot Listener* dari waktu ke waktu. Setiap puncak pada garis grafik menunjukkan peningkatan jumlah *listener* yang aktif, sementara penurunan menunjukkan penurunan jumlah *listener*. Puncak *snapshot* yang aktif berada pada rentang waktu jam 7.25PM - 7.35PM dengan nilai *snapshot* aktif sebesar 17/s.

Grafik ini menunjukkan pola aktivitas pengguna atau sistem. Jika ada puncak yang terjadi pada waktu-waktu tertentu, puncak tersebut mengindikasikan adanya aktivitas user yang tinggi pada waktu tersebut.

#### IV. KESIMPULAN

Dari analisa dan pembahasan yang telah dijelaskan sebelumnya, maka dapat disimpulkan bahwa;

1. Aktif *client login* tertinggi sebanyak 20 client, jumlah ini menunjukkan jumlah perangkat *client* terhubung ke *database* secara *realtime*.
2. Proses pembacaan data sebanyak 9 akses selama rentang waktu 7.15 PM-7.20 PM. Jumlah ini menggambarkan tindakan membaca atau mengakses data yang disimpan dalam dokumen *database* pada aplikasi magang.
3. *Request latency per database* tertinggi sebesar 0,06 detik, lebih kecil dari nilai ketetapan terendah 100 ms. Nilai ini menunjukkan bahwa *server* waktu tunggunya sangat singkat saat merespon permintaan data, sehingga dapat memberikan respon yang mulus dan cepat kepada setiap user saat melakukan akses pada layanan magang.
4. Lonjakan tertinggi *indeks write* terjadi pada rentang waktu antara jam 7.25 PM-7.30 PM dengan nilai 7,4/s. *Indeks*

*write* menunjukkan perubahan dan penulisan data di *firestore*. Hasil pengujian *indeks write* diketahui memiliki waktu tunggu yang sangat singkat, saat merespon permintaan data, sehingga tidak memengaruhi kinerja sistem, terutama pada operasi tulis.

5. *Document write* tertinggi yang dimonitoring selama 1 jam yaitu nilainya sebesar 0,365/s, terjadi pada rentang waktu 7.25 PM-7.30 PM. Lonjakan ini menunjukkan bahwa periode waktu ini, jumlah operasi tulis meningkat secara signifikan, karena adanya aktivitas yang intensif dalam menulis data baru pada waktu-waktu tersebut pada aplikasi magang.
6. Puncak *snapshot* yang aktif bernilai sebesar 17/s, berada pada rentang waktu jam 7.25PM-7.35PM. Nilai ini menunjukkan pola aktivitas pengguna atau sistem. Jika ada puncak yang terjadi pada waktu-waktu tertentu, itu mengindikasikan adanya aktivitas user yang tinggi pada waktu tersebut pada sistem aplikasi magang.

#### REFERENSI

- [1] B. Hariyanto, "Perancangan Sistem Magang Berbasis *Web* Pada Dinas Tenaga Kerja Dan Transmigrasi Provinsi Lampung," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 2, no. 3, pp. 334–343, 2021, doi: 10.33365/jatika.v2i3.1233.
- [2] D. I. Mulyana, A. A. Albahy, M. Khalid, M. H. Siregar, and M. U. A. Sadid, "Implementation of An Android-Based Mobile Application for Real-Time Student Learning Reports at SMP IDN Jonggol, West Java," *Int. J. Softw. Eng. Comput. Sci.*, vol. 4, no. 1, pp. 175–183, 2024, doi: 10.35870/ijsecs.v4i1.2228.
- [3] Indrawati, Afla Nevrisa, Radhyatammardhiyyah, Novira Dwina, Amir. D., "Analisa Kinerja Aplikasi Sistem Layanan Bimbingan Magang Mahasiswa", *Jurnal Mekanova : Mekanikal, Inovasi dan Teknologi* Vol 10 No. 2, Oktober 2024 P-ISSN : 2477-5029 E-ISSN : 2502-0498II.
- [4] H. Noprison, "Perancangan Aplikasi Monitoring Magang Untuk Mendukung Peningkatan Kualitas Pengalaman Bekerja Bagi Mahasiswa," *INTECOMS J. Inf. Technol. Comput. Sci.*, vol. 5, no. 2, pp. 72–90, 2022, doi: 10.31539/intecom.v5i2.4925.
- [5] M. I. Mahali, N. H. P. S. Putro, Sumaryanto, and B. Rahmat, "Android and FIREBASE mBaaS-based information system design of students activity unit (SAU) using the rational unified process (RUP) Method," *ACM Int. Conf. Proceeding Ser.*, pp. 6–12, 2019, doi: 10.1145/3348400.3348402.
- [6] *Firebase*, <https://firebase.google.com/docs/database/security/indexing-data?hl=id>, diakses 1 Oktober 2023.
- [7] T. A. Cinderatama, P. N. Malang, and Y. Yunhasnawa, "Desain Dan Implementasi *Hybrid Cloud Computing* Sebagai Infrastruktur Untuk Analisis Big Data Menggunakan Analytic Hierarchy Process ( AHP )," no. June, 2019, doi: 10.33633/tc.v17i4.1871.
- [8] Sutarti, Alif Alfiyansyah, "Analisis dan Implementasi Sistem *Monitoring* Koneksi *Internet* Menggunakan The Dude Di STIKOM Al Khairiyah", *Jurnal Sistem Informasi* Volume.4, Agustus 2017, p-ISSN: 2406-7768 e-ISSN: 2581-2181.
- [9] Daniel Richardo, Tatit Kurniasih, "Implementasi Google Cloud Pub/Sub menggunakan Metode *Subscription Pull* dalam Pengiriman Data Promosi Toko di PT XYZ", *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, Vol 5. No 2 2024, E-ISSN2723-7079, P-ISSN 2727-8074.
- [10] *Firebase*, "Membaca dan Menulis Data di Android", <https://firebase.google.com/docs/database/android/read-and-write?hl=id>, diakses 1 Oktober 2024.
- [11] Tubagus Entus Madhadi dan Lintang Yuniar Banowosari, "Analisis Perbandingan Performansi QoS VPN *Encryption Protocol* Pada Jaringan Berbasis *Hybrid Cloud*", *Jurnal Ilmiah KOMPUTASI*, pp.

- 69-82, Volume 20 No : 1, Maret 2021, p-ISSN 1412-9434/e-ISSN 2549-72.
- [12] Ryan Ari Setyawan, " Penerapan *Firebase Realtime Database* Pada Aplikasi Catatan Harian Diabetes Melitus". FAHMA –Jurnal Informatika Komputer, Bisnis dan Manajemen Vol.22, No.1, Januari 2024, pp. 1~9.
- [13] Ida Bagus Agung Eka Mandala Putra , Made Sri Inradewi Adnyana , Lie Jasa. " Analisis *Quality of Service* Pada Jaringan Komputer", Majalah Ilmiah Teknologi Elektro, Vol. 20, No.1, Januari – Juni 2021 DOI: <https://doi.org/10.24843/MITE.2021.v20i01.P11>