

Penerapan *Firestore Realtime Database* Pada Aplikasi Catatan Harian Diabetes Melitus

Ryan Ari Setyawan

Program Studi Informatika, Fakultas Teknik, Universitas Janabadra
email: ryan@janabadra.ac.id

Abstrak

Penelitian ini bertujuan mengetahui unjuk kerja atau performance dari teknologi *Firestore Realtime Database* melalui API (*application programming interface*) untuk digunakan pada aplikasi catatan harian diabetes melitus. Keunggulan *Firestore* merupakan platform yang dikembangkan oleh Google yang mempunyai library yang lengkap untuk pengembangan aplikasi cross platform maupun hybrid.

Metode API *Firestore Realtime Database* memproses sinkronisasi secara realtime melalui *JavaScript Object Notation (JSON)* yang disediakan. Metode sinkronisasi teknologi tersebut mempunyai tiga skema yakni fitur realtime, fitur offline, dan *Accessible from client devices*. Fitur realtime secara efisien menghubungkan dengan seluruh client yang terhubung secara otomatis. Pada fitur offline, *Firestore* dapat tetap secara responsif berjalan dalam keadaan luring, karena memiliki *Firestore SDK (software development kit)*. *Accessible from client devices*, layanan ini menawarkan kemudahan akses database secara realtime tanpa membutuhkan server application. Beberapa data yang kompleks sesuai dengan catatan harian diabetes melitus ini dapat melihat performa dari kinerja platform tersebut.

Hasil dari pengujian menggunakan *QoS* memperoleh hasil yang sangat baik. Hal tersebut sesuai dengan standar *TIPHON (Telecommunications and Internet Protocol Harmonization Over Network)* yang merupakan standar penilaian *QoS* yang dikeluarkan oleh *ETSI (European Telecommunications Standards Institute)* yang menghasilkan dari aplikasi diary diabetes melitus sangat baik dengan throughput 78.12 Kbps, Jitter sebesar 38.72 ms, packet loss sebesar 0 %, dan delay 22 ms.

Kata kunci—Android, Diabetes Melitus, *Firestore*, Quality of Service.

1. PENDAHULUAN

Firestore merupakan salah satu platform yang dikembangkan oleh Google yang telah diklaim lebih mudah dan efektif [1]. Pada prinsipnya *Firestore* dikembangkan untuk menjembatani beberapa *multiplatform* tanpa harus mengembangkan sendiri ketika membutuhkan mengenai kebutuhan server [2].

Firestore sendiri memiliki keunggulan dengan memiliki fitur lengkap seperti *authentication, firestore database, realtime database, google cloud messaging* bahkan sampai dengan fitur machine learning [2][3]. *Firestore* sendiri secara keseluruhan menyediakan kemudahan dengan cara membuat database server dengan no sql [4][5].

Oleh sebab itu, pemanfaatan *Firestore* dirasa perlu untuk mengembangkan aplikasi berbasis mobile, dengan memiliki keunggulan tanpa harus membuat server untuk penyimpanan database [6]. Pada penelitian ini yang akan digunakan pada fitur *Firestore* yakni *realtime database* untuk aplikasi catatan harian diabetes melitus.

Realtime database yang digunakan tentunya pada aplikasi catatan harian diabetes melitus yang ditujukan untuk mendukung aplikasi secara realtime. Proses kerjanya sama persis dengan aplikasi server yang dibuat secara manual, aplikasi yang terhubung dengan *Firestore* secara otomatis akan melakukan sesuai dengan perangkat yang telah diintegrasikan [6].

Proses integrasi inilah yang dapat menjadi kemudahan dan efisiensi dalam pengembangan, namun juga dapat menjadi faktor pengaruh kelancaran berjalannya aplikasi. Pada penelitian ini sesuai yang telah dijelaskan sebelumnya memiliki tujuan

juga untuk mengetahui performa kinerja firebase sebab pada proses integrasi dibutuhkan *time* yang sesuai serta antisipasi *loss connection* yang terjadi, sebab hal tersebut bekerja secara *realtime*. Parameter yang digunakan untuk mengetahui performa dari kinerja firebase yakni *delay*, *jitter*, *throughput* dan *packetloss* yang terjadi. Parameter yang digunakan tersebut merupakan parameter *Quality of Service (QoS)*, dimana parameter yang digunakan merupakan salah satu bentuk dari standar TIPHON (*Telecommunications and Internet Protocol Harmonization Over Network*) yang telah dikeluarkan oleh badan standar ETSI (*European Telecommunication Standard Institute*)[7]. Tujuan dari proses pengujian performa ini adalah aplikasi catatan harian diabetes melitus ini dapat digunakan tanpa memiliki kendala.

2. METODE PENELITIAN

Sesuai dengan tujuan penelitian yang telah dibahas pada bab sebelumnya untuk mengetahui performa kinerja firebase tentunya mengikuti beberapa langkah mekanisme mengukur *Quality of Service (QoS)*, merujuk hal tersebut maka penelitian ini tentunya mengacu pada penelitian yang pernah dilakukan sebelumnya agar secara terperinci dapat mengetahui langkah-langkah pengukuran kinerja dari firebase seperti yang dilakukan pada penelitian yang melakukan analisis pada aplikasi EatTrain menggunakan realtime database[7]. Proses yang digunakan adalah melakukan capture data transaksi pada lalu lintas request dan respon (transaksi) antara aplikasi dan server *firebase* [8].

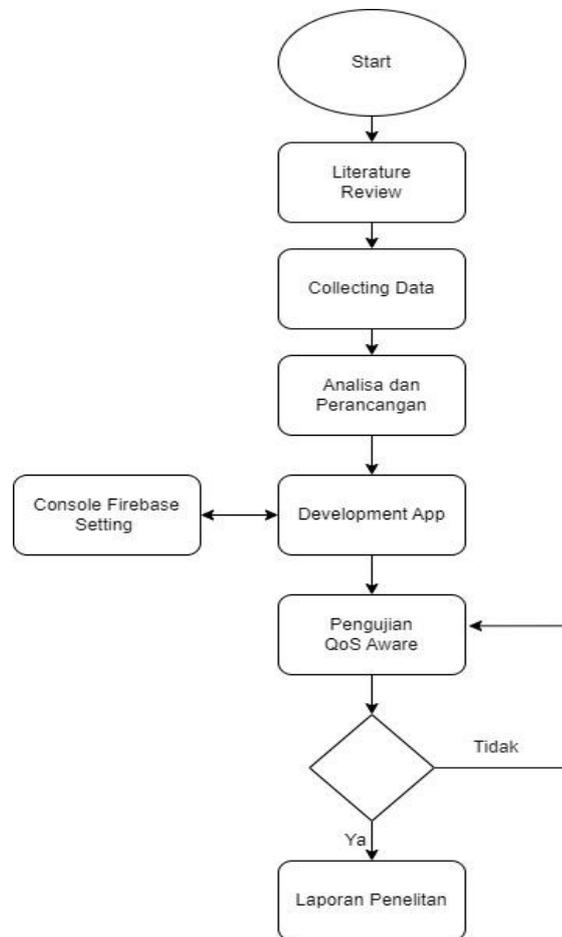
Pengujian dengan menggunakan parameter QoS pada firebase juga dilakukan performa web service yang ada pada *firebase* dan *google cloud messaging* dengan proses yang dilakukan dengan cara melakukan trial database dengan model web service yang kemudian dilakukan perekaman data lalu lintas (traffic) yang terjadi pada internet setelah itu dilakukan proses perhitungan *delay*, *jitter*, *packet loss* dan *throughput* [9].

Teknik pengukuran *Quality of Service (QoS)* dari firebase yang lain juga dilakukan mengenai studi komparatif mengenai WebXR dan AR (*Augmented Reality*) menggunakan firebase. Untuk mengetahui point of interest (PoI) dapat bekerja dengan baik pada AR maka perlu diketahui time respons (*delay*) yang terjadi [10].

Referensi mengenai beberapa langkah perhitungan mengenai performa firebase yang diukur dengan parameter QoS ini menjadi acuan pada penelitian ini untuk mengetahui performa dari penerapan firebase pada aplikasi catatan diabetes melitus, dengan teknologi yang diusulkan adalah firebase realtime database yang dapat memudahkan dan lebih efisien ketika aplikasi tersebut berjalan diharapkan tidak terjadi kendala tentunya juga harus diketahui performa *firebase realtime database*.

Untuk metode penelitian yang digunakan memiliki beberapa tahapan. Gambar 1 menjelaskan mengenai tahapan penelitian yakni literatur review, merupakan tahapan yang bertujuan untuk memahami masalah-masalah yang ada untuk selanjutnya dibuat menjadi sistem informasi dengan menggunakan wawancara dan review beberapa jurnal atau prosiding. *Collecting Data*, tahap ini adalah tahapan yang bertujuan untuk menganalisa kebutuhan sistem yang dibangun dengan melihat dari masalah yang telah diuraikan dengan mengumpulkan data yang dibutuhkan, dalam hal ini adalah data mengenai *diabetes melitus*. Analisa dan Perancangan adalah tahap dimana perancangan perangkat lunak sesuai dengan kebutuhan untuk penderita diabetes melitus serta penerapan *firebase*. Development app adalah tahap pembuatan desain mock-up user interface serta melakukan proses pengkodean sesuai dengan analisa perancangan yang telah dilakukan. *Console firebase setting*, tahapan ini adalah tahapan untuk integrasi antara aplikasi yang telah dibuat dengan *firebase realtime database* sesuai object yang akan dijadikan parameter API [11]. Pengujian QoS, tahapan ini adalah tahapan

pengujian sistem untuk mengetahui kesesuaian sistem dengan tahapan perencanaan yang telah dilakukan dengan metode QoS sesuai yang telah dijelaskan pada landasan teori mengenai parameter yang digunakan.



Gambar 1. Tahapan Penelitian

Untuk langkah pengujian atau pengukuran kinerja firebase yakni dengan menggunakan *Quality of Service* (QoS) langkah yang dilakukan adalah aplikasi catatan diabetes melitus digunakan oleh pengguna, kemudian pengguna melakukan proses penginputan data yang terdapat pada aplikasi tersebut, dengan menggunakan access point yang sama maka proses tersebut atau dapat dikatakan traffic yang terjadi tersebut dilakukan capturing (perekaman) dengan menggunakan perangkat lunak wireshark. Setelah dilakukan proses perekaman menggunakan wireshark, hasil capturing lalu lintas data yang dihasilkan langkah berikutnya yakni dilakukan perhitungan sesuai dengan parameter *delay*, *jitter*, *throughput* dan *packet loss*. Berikut merupakan penilaian parameter QoS sesuai dengan kategori sesuai dengan standar TIPHON (*Telecommunications and Internet Protocol Harmonization Over Network*)[12]. Jumlah *user* yang diujikan adalah sebesar 41 *user* (pengguna aplikasi).

Throughput adalah kecepatan (rate) transfer data efektif, yang diukur dalam bps (bit per second). *Packet Loss* adalah parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang dapat terjadi karena *collision* dan *congestion*. Tabel 1 menunjukkan pengelompokan dari *Throughput*, sedangkan tabel 2 menunjukkan pengelompokan dari *Packet Loss*.

Tabel 1. Kategori Throughput

| Kategori | Throughput (Bps) |
|--------------|------------------|
| Sangat Bagus | 100 bps |
| Bagus | 75 bps |
| Sedang | 50 bps |
| Jelek | < 25 bps |

Tabel 2. Kategori Packet Loss

| Kategori | Packet Loss (%) |
|--------------|-----------------|
| Sangat Bagus | 0% |
| Bagus | 3% |
| Sedang | 15% |
| Jelek | 25% |

Delay (Latency) merupakan waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. Tabel 3 menunjukkan beberapa pengelompokan dari Delay.

Tabel 3. Kategori Delay

| Kategori | Delay (ms) |
|--------------|-------------------|
| Sangat Bagus | <150 ms |
| Bagus | 150 ms s.d 300 ms |
| Sedang | 300 ms s.d 450 ms |
| Jelek | >450 ms |

Jitter diakibatkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket-paket diakhir perjalanan *jitter*. Tabel 4 berikut ini menunjukkan beberapa pengelompokan dari *Jitter*.

Tabel 4. Kategori Jitter

| Kategori | Jitter (ms) |
|--------------|-------------------|
| Sangat Bagus | 0 ms |
| Bagus | 0 ms s.d 75 ms |
| Sedang | 75 ms s.d 125 ms |
| Jelek | 125 ms s.d 225 ms |

3. HASIL DAN PEMBAHASAN

Penerapan *firebase* pada aplikasi *diary diabetes melitus* tentunya sesuai tahapan penelitian yakni *development* app terlebih dahulu. Untuk tampilan aplikasi menu diary diabetes melitus dapat diperlihatkan pada beberapa gambar berikut. Gambar 2 menjelaskan *splash screen* merupakan tampilan yang akan muncul setiap buka aplikasi harian buku *diabetes melitus*. Gambar 3 merupakan tampilan login sistem dimana pengguna dapat masuk ke aplikasi melalui akun google. Login ini dapat bekerja secara otomatis karena melalui google mail yang sudah terdapat di *smartphone* pengguna. Gambar 4 menjelaskan pada menu utama akan berisi menu-menu sesuai kebutuhan catatan harian pasien diabetes melitus seperti, catatan konsultasi saya, obat saya, kadar gula dan tekanan darah, riwayat kesehatan, identifikasi resiko diabetes dan yang lainnya. Gambar 5 menu jelajahi merupakan menu bantuan untuk informasi tim layanan kesehatan, pemeriksaan layanan kesehatan, info dukungan psikososial, dan indentifikasi kaki diabetes.



Gambar 2. *Splash Screen.*



Gambar 3. Login Sistem



Gambar 4. Menu Utama



Gambar 5. Menu Jelajahi

Setelah aplikasi dibuat, maka secara paralel mengintegrasikan dengan *firebase console* untuk database realtime. Platform database yang digunakan pada aplikasi realtime. Ketika terjadi perubahan data, maka aplikasi yang terhubung dengan firebase akan memperbaharui secara otomatis melalui setiap *device* (perangkat) baik website ataupun mobile [13]. Firebase mempunyai *library* (pustaka) yang lengkap untuk sebagian besar *platform web dan mobile*. Firebase dapat digabungkan dengan framework lain seperti node, java, javascript, dan lain-lain [14]. Terdapat beberapa fitur yang disediakan oleh firebase adalah sebagai berikut : 1. Analytics, fitur ini digunakan untuk mengamati tingkah laku pengguna dalam penggunaan aplikasi dan ditampilkan dalam satu dashboard. 2. *Develop*, fitur ini berupa *cloud messaging*,

authentication, *realtime database*, storage, hosting, testlab dan crash reporting. 3. Grow, fitur ini digunakan untuk mempublikasikan sebuah produk aplikasi [15]. Berikut adalah beberapa fitur *firebase*.

Selanjutnya yakni melakukan pengujian performa *firebase* pada aplikasi dengan parameter *Quality of Service (QoS)* sesuai dengan penjelasan mengenai langkah pengukuran pada gambar 1 tahapan penelitian. Pada proses pengujian ini aplikasi diary diabetes melitus melakukan proses transaksi pada *database firebase*. Kemudian dengan parameter QoS seperti *throughput*, *jitter*, *delay*, dan *packet loss* dihitung. Pengujian dilakukan dengan cara melakukan capture lalu lintas data melalui bantuan software *wireshark* terlebih dahulu, setelah itu dilakukan perhitungan sesuai parameter QoS [16].

Throughput adalah parameter untuk mengetahui kecepatan (*rate*) transfer data efektif, yang diukur dalam bps (bit per second). Untuk *Throughput* yang dihasilkan dari aplikasi yakni sebesar 78Kpbs. Untuk perhitungan *throughput*. $Throughput = \text{Paket Data Terima} : \text{Lama Pengamatan}$

$$\begin{aligned} &= 404.222 \text{ Bytes} : 41.395 \\ &= 9.765 \text{ Bytes} \\ &= 9.765 \text{ Bytes} \times 8 \\ &= 78.12 \text{ Kbps.} \end{aligned}$$

Bila membandingkan dengan kategori *throughput* yang dihasilkan sesuai dengan standar TIPHON pada Tabel 1 mengenai kategori *throughput* maka hasil pengujian yang dilakukan ini masuk kedalam kategori yang sangat bagus karena menghasilkan 78 Kbps artinya > 100 bps.

Parameter *Packet loss*, parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang dapat terjadi karena *collision* dan *congestion*. Untuk *packet loss* yang dihasilkan yakni sebesar 0 %. Untuk rumus perhitungan *packet loss*. $Packet\ loss = \frac{(\text{Paket Data Dikirim} - \text{Paket Data Diterima})}{\text{Paket Data Yang Dikirim}} \times 100\%$

$$\begin{aligned} &= \frac{(1.055 - 1.055)}{1.055} \times 100 \\ &= \frac{(0)}{1.055} \times 100 \\ &= 0\% \end{aligned}$$

Bila merujuk pada kategori *packet loss* pada Tabel 2 mengenai kategori *packet loss*, maka hasil pengujian yang diperoleh yakni 0% *packet loss*, masuk dalam kategori sangat bagus.

Untuk *jitter* yang dihasilkan yakni sebesar 39 ms. Perhitungan dari *jitter* dilakukan dengan cara berikut ini:

$$\begin{aligned} \text{Jitter} &= \text{Total Variasi Delay} : \text{Total Paket yang di terima} \\ &= 40.849.600 \text{ s} : 1.055 \\ &= 38.720 \text{ s} \times 1.000 \\ &= 38,72 \text{ ms} \end{aligned}$$

Jitter yang dihasilkan sebesar 38.72 ms, bila merujuk pada tabel 4 kategori standarisasi *jitter* maka dikatakan masuk ke dalam kategori bagus.

Untuk *delay* yang dihasilkan yakni sebesar 22 ms. Perhitungan *delay* dilakukan dengan cara berikut ini.

$$\begin{aligned} \text{Rata-Rata Delay} &= \text{Total Delay} : \text{Total Paket yang diterima} \\ &= 7.136.265 \text{ s} : 318 \\ &= 22.441 \text{ s} \times 1.000 \\ &= 22 \text{ ms} \end{aligned}$$

Tabel 5. Hasil Pengujian dengan kategori QoS

| Parameter QoS | Hasil Pengujian | Kategori QoS | Nilai Standar TIPHON |
|--------------------|-----------------|--------------|----------------------|
| <i>Throughput</i> | 78.12 Kbps | Sangat Bagus | 100 bps |
| <i>Packet loss</i> | 0 % | Sangat Bagus | 0% |
| <i>Jitter</i> | 38,72 ms | Bagus | 0 ms s.d 75 ms |
| <i>Delay</i> | 22 ms | Sangat Bagus | < 150 ms |

Sedangkan untuk delay yang dihasilkan yakni sebesar 22 ms, melihat pada kategori tabel 3 mengenai kategori jitter maka hasil delay pada penelitian ini masuk dalam kategori sangat bagus.

Bila melihat dari hasil dari pengujian secara keseluruhan menggunakan QoS memperoleh hasil yang rata-rata sangat bagus. Hal tersebut sesuai dengan standar TIPHON (*Telecommunications and Internet Protocol Harmonization Over Network*) yang merupakan standar penilaian QoS yang dikeluarkan oleh ETSI (*European Telecommunications Standards Institute*) [16].

4. KESIMPULAN

Hasil pengujian yang dilakukan dengan *user* (pengguna aplikasi) sebesar 41 *user* dan dilakukan pengujian dengan cara melakukan *capturing* pada lalu lintas data pada aplikasi serta dilakukan perhitungan maka hasil pengujian diperoleh *throughput* sebesar 78.12 Kbps, *Jitter* sebesar 38.72 ms, *packet loss* sebesar 0 %, dan *delay* 22 ms. Hasil parameter QoS tersebut menunjukkan bahwa rata-rata masuk ke dalam kategori yang sangat bagus. Parameter QoS tersebut juga menunjukkan bahwa aplikasi catatan harian diabetes melitus digunakan dapat melakukan penyimpanan dan pengaksesan data dengan sangat baik. Penerapan pada aplikasi ini setelah dilakukan pengujian dengan performa yang masuk dalam kategori sangat bagus yakni dapat membantu para pasien untuk dapat melakukan monitoring mandiri penyakit diabetes melitus dan berkala untuk menghindari resiko kompleksitas lain.

5. SARAN

Penelitian lanjut dapat dikembangkan dengan beberapa fitur layanan yang telah tersedia di dalam console firebase seperti *firebase analytics*, *google cloud messaging*, serta fitur *machine learning* yang dapat membuat aplikasi semakin pintar untuk digunakan.

6. UCAPAN TERIMA KASIH

Terimakasih kepada lab mobile apps, Prodi Informatika, Universitas Janabadra yang telah mendukung penelitian ini, dari proses perancangan sampai tahap pengujian.

DAFTAR PUSTAKA

- [1] I. Ulumudin, N. Faizah, and W. Nurcahyo, "Aplikasi Sistem Presensi Pegawai PT. Berkah Pena Ilmu dengan Metode Location Based Service (LBS) Berbasis Android Menggunakan Firebase," *Des. J.*, vol. 1, no. 1, pp. 89–98, 2023, doi: 10.58477/dj.v1i1.61.
- [2] M. Ohyver, J. V. Moniaga, I. Sungkawa, B. E. Subagyo, and I. A. Chandra, "The comparison firebase realtime database and MySQL database performance using wilcoxon signed-rank test," *Procedia Comput. Sci.*, vol. 157, pp. 396–405, 2019, doi: 10.1016/j.procs.2019.08.231.
- [3] A. Tedyyana and O. Ghazali, "Real-time Hypertext Transfer Protocol Intrusion Detection System on Web Server using Firebase Cloud Messaging," pp. 385–392, 2023, doi: 10.5220/0010946300003260.
- [4] J. Romadoni, B. R. Hakim, and M. Rasyid, "Sistem Informasi Wisata Religi di Kota Banjarmasin Berbasis Flutter dan Firebase," vol. 2, no. 1, pp. 41–53, 2023.
- [5] Sugiyatno, "Pengiriman Informasi Real Time Menggunakan Teknologi Database Firebase pada Aplikasi Mobile Android," *J. Inform. Komputer, Bisnis dan Manaj.*, vol. 21, no. 2, pp. 46–55, 2023, doi: 10.61805/fahma.v21i2.17.
- [6] E. A. W. Sanad, "Pemanfaatan Realtime Database di Platform Firebase Pada Aplikasi E-

- Tourism Kabupaten Nabire,” *J. Penelit. Enj.*, vol. 22, no. 1, pp. 20–26, 2019, doi: 10.25042/jpe.052018.04.
- [7] D. S. Mukti, N. Hidayati, and A. Rasyid, “Quality of Service Analysis on EaTrain Application at A Restaurant on A Train,” *J. Ecotipe (Electronic, Control. Telecommun. Information, Power Eng.*, vol. 10, no. 2, pp. 225–236, 2023, doi: 10.33019/jurnalecotipe.v10i2.4477.
- [8] Sugiyatno and I. M., “Analisis Perbandingan Performasi Respon Waktu Web Server dan Failover Antara Kubernetes Dan Docker Swarm pada Container Orchestration,” *J. Inform. Komputer, Bisnis dan Manaj.*, vol. 21, no. 3, pp. 43–53, 2023, doi: 10.61805/fahma.v21i3.9.
- [9] G. Albertengo, F. G. Debele, W. Hassan, and D. Stramandino, “On the performance of web services, google cloud messaging and firebase cloud messaging,” *Digit. Commun. Networks*, vol. 6, no. 1, pp. 31–37, 2020, doi: 10.1016/j.dcan.2019.02.002.
- [10] A. Bakr Rahman Shaik Venkata Sai Yakkahit Reddy Asodi, A. Bakr Rahman Shaik, V. Sai Yakkahit Reddy Asodi, and V. Garro, “Comparative Analysis of the Performance of ARCore and WebXR APIs for AR Applications,” no. June, 2023, [Online]. Available: www.bth.se.
- [11] M. Haghi Kashani, A. M. Rahmani, and N. Jafari Navimipour, “Quality of service-aware approaches in fog computing,” *Int. J. Commun. Syst.*, vol. 33, no. 8, pp. 1–34, 2020, doi: 10.1002/dac.4340.
- [12] R. A. Setyawan, “Analisis Unjuk Kerja Aplikasi VoIP Call Android di Jaringan MANET,” *Bul. Pos dan Telekomun.*, vol. 13, no. 1, p. 79, 2015, doi: 10.17933/bpostel.2015.130106.
- [13] H. Esmael and H. R. Esmael, “Apply Android Studio (SDK) ,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 5, no. 5, 2015, [Online]. Available: <https://www.researchgate.net/publication/331673953>.
- [14] A. Mazuera-Rozo *et al.*, “Taxonomy of security weaknesses in Java and Kotlin Android apps,” *J. Syst. Softw.*, vol. 187, p. 111233, 2022, doi: 10.1016/j.jss.2022.111233.
- [15] R. C. Buwono, “Web Services Menggunakan Format JSON,” *Respati*, vol. 14, no. 2, pp. 1–10, 2019, doi: 10.35842/jtir.v14i2.282.
- [16] P. R. Utami, “Analisis Perbandingan Quality of Service Jaringan Internet Berbasis Wireless Pada Layanan Internet Service Provider (Isp) Indihome Dan First Media,” *J. Ilm. Teknol. dan Rekayasa*, vol. 25, no. 2, pp. 125–137, 2020, doi: 10.35760/tr.2020.v25i2.2723.