

© Jurnal Nasional Teknik Elektro dan Teknologi Informasi
Karya ini berada di bawah Lisensi Creative Commons Atribusi-BerbagiSerupa 4.0 Internasional
Terjemahan dari 10.22146/jnteti.v13i1.10095

Sistem Berbasis Komputasi Kabut Untuk Sistem Parkir Pintar Terdesentralisasi Menggunakan Firebase

Haposan Yoga Pradika Napitupulu¹, I Gde Dharma Nugraha¹

¹ Departemen Teknik Elektro, Fakultas Teknik, Universitas Indonesia, Depok, Indonesia

[Diserahkan: 5 Oktober 2023, Direvisi: 29 November 2023, Diterima: 8 Desember 2023]

Penulis Korespondensi: Haposan Yoga Pradika Napitupulu (email: haposan.yoga@ui.ac.id)

INTISARI — Pertumbuhan jumlah kendaraan tidak dapat dihindari sementara ketersediaan lahan parkir tidak berbanding lurus dengan kondisi ini. Saat ini, banyak pusat perbelanjaan yang tidak memiliki lahan parkir yang memadai, sehingga pelanggan mengalami kesulitan untuk menemukan tempat parkir kosong. Penelitian telah dilakukan untuk mengatasi masalah ketersediaan lahan parkir ini. Sebagian besar penelitian ini mengusulkan *narrowband-internet of things* (NB-IoT) sebagai *fog node*. Untuk tujuan komunikasi, *fog node* berbasis NB-IoT ini memiliki beberapa kekurangan, di antaranya adalah masalah keamanan dan privasi, rendahnya kecepatan data, tingginya biaya pengembangan, ketergantungannya dengan sistem nirkabel, dan cakupannya yang hanya satu area. Pada penelitian ini, komputasi kabut diusulkan untuk mendesentralisasikan sistem parkir pintar dengan menggunakan Firebase, sehingga dapat mencakup beberapa area atau mal dalam satu sistem dan antarmuka. Alih-alih menggunakan NB-IoT, penelitian ini menggunakan server lokal yang terdesentralisasi sebagai *fog node* untuk pertukaran data yang cepat. Basis data (pangkalan data) Firestore (Firebase) juga digunakan untuk mengamankan, mengelola, dan menganalisis data di *cloud*. Kemudian, aplikasi Android dibuat sebagai antarmuka pengguna untuk memesan dan menemukan tempat parkir kosong. Aplikasi Android dibuat menggunakan Android Studio dan autentikasi diimplementasikan agar akses data tetap aman dan privat. Skenario pengujian dilakukan dengan mengikuti desain *unified modeling language* (UML). Hasil penelitian menegaskan bahwa sistem komputasi kabut berhasil mendukung sistem parkir pintar yang terdesentralisasi dan dapat diimplementasikan dengan mencakup beberapa area atau mal dalam satu sistem.

KATA KUNCI — Parkir Pintar, *Fog Node*, Firebase, MySQL.

I. PENDAHULUAN

Pertumbuhan jumlah kendaraan saat ini tidak dapat dihindari. Sayangnya, peningkatan jumlah kendaraan ini tidak sejalan dengan ketersediaan lahan parkir. Setiap hari, kendaraan dibuat dan diproduksi untuk memenuhi permintaan. Di sisi lain, lahan untuk tempat parkir terbatas dan tidak dapat diproduksi. Sebagai contoh, banyak kompleks perbelanjaan dan mal di kota-kota besar tidak memiliki lahan parkir yang memadai. Kondisi ini dapat menyebabkan ketidaknyamanan bagi pelanggan, terutama pelanggan yang baru pertama kali mengunjungi mal-mal tersebut. Hal ini dapat membuat pelanggan frustrasi karena harus menghabiskan banyak waktu dan menelusuri area yang luas hanya untuk menemukan tempat parkir yang kosong. Kondisi tersebut tentu saja tidak menguntungkan pelanggan karena harus mengorbankan waktu untuk melakukan pekerjaan yang tidak dikehendaki dan merugikan mal karena dapat mendapatkan penilaian buruk dari pelanggan. Selain pelanggan, petugas parkir yang bekerja di mal tersebut juga akan berdampak oleh masalah tempat parkir.

Dengan memanfaatkan teknologi saat ini, orang dapat dengan mudah menjelajahi mal dan menemukan tempat parkir yang sesuai. Pemanfaatan teknologi ini akan memungkinkan pelanggan untuk mengambil keputusan serta menghemat waktu berharga dengan cara memanfaatkan informasi. Salah satu jenis teknologi informasi yang paling umum digunakan saat ini adalah komputasi awan. Teknologi ini memungkinkan orang untuk memproses dan menyimpan data di satu tempat di *cloud* tanpa memerlukan server yang terpisah, sehingga memperlancar pekerjaan. Tujuannya adalah memudahkan orang untuk menemukan tempat parkir yang kosong. Teknologi ini juga dapat mengurangi kemacetan dan meningkatkan

pengalaman pelanggan. Tidak hanya menampilkan ketersediaan tempat parkir, teknologi ini juga memberikan informasi kepada pelanggan mengenai tarif fasilitas parkir. Informasi ini akan ditampilkan di situs web gedung dan di aplikasi seluler.

Selain komputasi awan, terdapat *node* komputasi kabut, yang merupakan kombinasi dan pengaturan yang bagus untuk memenuhi kebutuhan koneksi dalam jumlah besar dan latensi yang rendah. *Internet of vehicle* (IoV) muncul sebagai sistem manajemen sumber daya baru yang dapat meningkatkan pemanfaatan sumber daya sistem dan kualitas layanan kendaraan [1], [2]. Penelitian lain telah menunjukkan fungsi parkir pintar yang menjanjikan [3]. Hasil penelitian tersebut menunjukkan bahwa komputasi awan kendaraan mampu memberikan latensi sangat rendah bagi klien dan kendaraan [3].

Terdapat beberapa aplikasi potensial untuk *node* komputasi kabut, salah satunya adalah pengawasan rumah pintar, yaitu sistem yang sudah ada menggunakan *home gateway* untuk aplikasi parkir pintar, layanan pengiriman, dan ritel pintar. Sistem tersebut dapat digantikan dengan *fog node* dan sistem *narrowband-internet of things* (NB-IoT) [4]. Sistem yang diusulkan pada [4] menggunakan sistem NB-IoT, yaitu setiap tempat parkir memiliki sensor NB-IoT dengan *ID* unik. Pada setiap mobil juga harus dipasang sensor pelacakan NB-IoT. NB-IoT di tempat parkir akan berada pada mode tidur kecuali jika terdapat mobil dengan NB-IoT datang ke tempat parkir tersebut. Sebuah penelitian telah menunjukkan bahwa penggunaan manajemen terpusat dalam infrastruktur komputasi awan berbasis kendaraan memungkinkan proses pengambilan keputusan yang cepat [5]. Dari penelitian-penelitian sebelumnya, dapat disimpulkan bahwa komputasi

awan sangatlah penting dan memiliki potensi besar untuk mendukung sistem parkir pintar yang terdesentralisasi yang mencakup beberapa mal atau area.

Penelitian sebelumnya telah menggunakan NB-IoT untuk sistem parkir pintar [6]–[10]. Hasil penelitian tersebut menunjukkan bahwa penggunaan NB-IoT untuk komputasi kabut adalah cara yang menjanjikan untuk menyelesaikan masalah parkir karena NB-IoT mirip dengan komputasi kabut dalam hal kemampuan mengurangi tunda komunikasi. Selain itu, NB-IoT memiliki jangkauan yang luas dan dapat menjangkau secara vertikal, sehingga dapat mencapai setiap sudut tempat parkir. NB-IoT juga memiliki konsumsi daya rendah, efisiensi tinggi, dan mudah dioperasikan. Sistem parkir pintar meningkatkan kenyamanan orang karena dapat menyelesaikan permasalahan menemukan tempat parkir kosong.

Menggunakan komputasi kabut, penelitian lain telah membuat parkir pintar yang terdesentralisasi [11], [12]. Ide yang diusulkan serupa dengan temuan penelitian lain yang menggunakan IoT, yaitu dengan menempatkan sensor di tempat parkir dan memasang sensor di dalam mobil. Pembangunan *gateway* sebagai sistem desentralisasi untuk komunikasi setiap sensor di area parkir dimaksudkan untuk bertukar data dan mengumpulkan data sensor yang secara fisik terkumpul di area parkir. Selanjutnya, data-data tersebut disimpan di *cloud*. Namun, penggunaan NB-IoT memiliki kelemahan dalam hal keamanan dan privasi serta kecepatan data yang lebih rendah. Pengembangan parkir pintar juga mahal karena harus melengkapi setiap tempat parkir di setiap mal dengan modul NB-IoT. Walaupun biayanya mahal, NB-IoT masih bergantung pada konektivitas, sehingga operasi akan mati ketika konektivitas nirkabel mati. Sebuah penelitian juga menunjukkan bahwa kerahasiaan data penting harus dilindungi [13]. Jika melihat penelitian sebelumnya, sebagian besar penelitian hanya berfokus pada pengembangan parkir pintar untuk satu area cakupan saja.

Penelitian ini mengusulkan sebuah sistem yang berbasis komputasi kabut untuk sistem parkir pintar yang terdesentralisasi dengan menggunakan sistem data Firebase, sehingga mencakup beberapa area atau mal dalam satu sistem dan antarmuka. Sistem ini dirancang untuk digunakan pada tempat parkir yang berada di mal. Saat ini, mal-mal telah dilengkapi dengan sensor yang dapat mengidentifikasi ketersediaan tempat parkir. Sensor-sensor tersebut mengumpulkan data-data tempat parkir yang tersedia, yang kemudian dikirimkan ke server lokal sebagai *fog node* untuk mempercepat pembaruan. Dengan demikian, setiap mal akan mengirimkan data tempat parkir ke server lokalnya sebagai sistem terdesentralisasi yang menjaga keamanan data karena disimpan di MySQL di server lokal. Sistem yang diusulkan tidak hanya melayani satu mal, tetapi beberapa mal yang terdaftar dalam aplikasi. Selanjutnya, *fog node* akan dihubungkan ke *cloud*. Pada penelitian ini, Firebase digunakan untuk mengumpulkan semua data dan melakukan komputasi awan. *Cloud* juga digunakan untuk mengelola data. Firebase menawarkan autentikasi untuk membaca dan menulis data yang dapat mengamankan data. Dengan terhubung ke Firebase, reservasi dilakukan menggunakan aplikasi Android sebagai antarmuka pengguna. Cara ini dapat mengamankan data karena terdapat dua lapisan autentikasi untuk mengakses data. Autentikasi pertama terdapat di *fog node*, sementara yang kedua di *cloud*. Selain itu, pembaruan akan lebih cepat karena

fog node dan sensor berada dalam satu jaringan. Struktur makalah ini adalah sebagai berikut: Bagian II membahas komputasi awan dan parkir pintar, Bagian III menyajikan desain dan implementasi, Bagian IV menyajikan hasil dan analisis, lalu Bagian V menyajikan kesimpulan.

II. KOMPUTASI KABUT DAN PARKIR PINTAR

A. PARKIR PINTAR

Parkir pintar adalah jenis sistem parkir yang menggunakan berbagai teknologi untuk mengelola tempat parkir secara efisien. Banyak penelitian terkait parkir pintar telah dilakukan dan banyak kota sudah memulai serta mengembangkan proyek parkir pintar ini. Konsep ini bertujuan untuk membantu pengemudi memanfaatkan teknologi informasi dan komunikasi untuk menemukan tempat parkir yang memuaskan secara efisien.

Sistem ini juga diartikan sebagai sistem pintar yang membantu pengemudi untuk mengetahui ada atau tidaknya kendaraan di tempat parkir. Sistem akan menemukan tempat parkir yang kosong dengan menggunakan sensor, kemudian mengarahkan pengemudi ke tempat parkir tersebut. Selain itu, parkir pintar juga melibatkan penggunaan sistem untuk menemukan tempat parkir kosong dari jarak jauh. IoT dan komputasi awan bertujuan untuk mengurangi waktu yang dibutuhkan untuk menemukan tempat parkir kosong dengan menggunakan sebuah platform. Tidak hanya waktu yang akan berkurang, konsumsi bahan bakar dan emisi karbon dioksida (CO₂) sebagai akibat dari kendaraan di perkotaan dan tempat parkir yang tidak dikelola juga dapat berkurang [14].

Perkembangan parkir pintar saat ini sangat bergantung pada komputasi awan. Hal ini juga dapat disebut sebagai *vehicular fog computing* (VFC). Temuan ini mulai berkembang dan menjanjikan. Penggunaan komputasi awan untuk transportasi dapat memberikan komputasi yang ringan dan ada di mana pun pada tepi jaringan kendaraan [15].

B. KOMPUTASI KABUT

Saat ini, tren revolusi informasi sedang melesat. Teknologi seperti *artificial intelligence* (AI), IoT, komputasi awan, dan *big data* merambah semua bidang ekonomi sosial dan secara signifikan berdampak pada kehidupan manusia. Dengan teknologi komputasi awan, data dalam jumlah besar dapat disimpan, dianalisis, dan dikelola di *cloud*. Tekanan pada pusat data *cloud* juga meningkat karena adanya lonjakan aplikasi data. Oleh karena itu, pencarian model komputasi baru merupakan topik penelitian yang esensial.

Di antara lapisan *cloud* dan *end device* terdapat sebuah lapisan, yaitu lapisan *fog*, yang telah diperkenalkan dalam komputasi awan. Model komputasi terdistribusi digunakan oleh komputasi kabut. *Fog node* didistribusikan ke tepi *end device* untuk melakukan pemrosesan data, komputasi, dan komunikasi, sehingga secara efektif mengurangi tekanan komputasi pusat data *cloud*. Komputasi kabut adalah layanan canggih komputasi awan untuk mengatasi dan memecahkan masalah, seperti kurangnya mobilitas dan latensi [14], [16]. Terdapat tiga lapisan yang khususnya digunakan dalam komputasi kabut, yaitu lapisan *cloud*, lapisan *fog*, dan lapisan pengguna. Server *fog* adalah bagian dari lapisan *fog* yang ditempatkan di sekitar bangunan.

Kerangka kerja komputasi kabut memungkinkan penyimpanan *end-to-end* data penting dan layanan komputasi di *cloud server* dengan cara membuat lapisan *fog* dengan penyimpanan yang tepat dan kemampuan komputasi di antara

cloud server, kemudian menambahkannya ke terminal aplikasi. Lalu, lapisan fog dipindahkan ke *fog server* di dekat perangkat. Untuk menyalurkan dan memproses data yang diunggah oleh *user layer*, komputasi kabut diperkuat dengan berbagai perangkat komunikasi, penyimpanan data, dan komputasi. Dengan demikian, cara ini secara efektif mengurangi beban komputasi dan tekanan penyimpanan *cloud server*.

C. BASIS DATA FIRESTORE

Firebase adalah platform pengembangan untuk aplikasi web dan seluler. Platform ini menawarkan berbagai layanan, yaitu manajemen data, analisis data, proses autentikasi, *web hosting*, sistem penyimpanan, dan basis data (pangkalan data) yang disebut basis data Firestore (Firebase). Basis data Firestore memungkinkan pengguna untuk menyimpan dan melakukan sinkronisasi antaraplikasi secara *real-time* karena sistem basis data ini didasarkan pada NoSQL yang disediakan di *cloud*. Firestore menyediakan dokumentasi yang detail dan *software development kits* (SDK) lintas platform untuk membantu membuat dan mengirimkan aplikasi di Android. Firebase dapat melakukan sinkronisasi dengan server lokal dan menggunakan data JSON untuk melakukan sinkronisasi secara *real-time*.

Basis data Firestore lebih fleksibel karena data disimpan dalam bentuk dokumen dengan pengaturan di dalam koleksi dan subkoleksi. Hal ini sangat berbeda dengan MySQL. Pada MySQL, struktur kolom atau tabel harus didefinisikan. Sementara itu, pada basis data Firestore, setiap dokumen di dalam *cloud* memiliki strukturnya sendiri. Dengan demikian, basis data Firestore menawarkan fleksibilitas yang lebih besar. Namun, disarankan agar struktur data di setiap dokumen didefinisikan serupa dengan yang lain, sehingga kueri dapat dibuat lebih mudah [17].

Tidak hanya fleksibel, basis data Firestore juga memiliki *expressive query* yang memungkinkan data disaring dan diurutkan dalam sebuah grup menggunakan antarmuka kueri yang ekspresif dan sederhana. Pengguna dapat dengan mudah memperbarui data secara *real-time* karena basis data Firestore dapat melakukannya dengan menggunakan SDK yang kuat. Sebaliknya, basis data Firestore dapat mendukung secara *offline*. Data kueri yang aktif digunakan akan disimpan pada *cache* dalam perangkat dan dapat ditulis, dibaca, dan dideteksi. Setelah perangkat terhubung ke internet, data akan langsung disinkronkan [17]. Selain itu, basis data Firestore memberikan keamanan dan privasi yang tangguh karena kemampuannya mengimplementasikan aturan akses autentikasi dan otorisasi dengan mudah, dengan cara mengintegrasikan *identity and access management* (IAM), Web Client SDK, dan Firebase Authentication ke dalam *cloud* [17].

Sistem autentikasi di Firebase mudah digunakan dan hanya memerlukan sedikit modifikasi. Firebase Authentication terintegrasi langsung ke dalam Firebase Database. Selain itu, komputasi awan memungkinkan dilakukannya pengelolaan dan pemrosesan data dalam jangka waktu yang lama serta dilakukannya analisis data yang komprehensif [18], [19]. Dengan demikian, Firebase Database memfasilitasi akses kontrol ke data. Firebase juga memiliki keamanan bawaan di tingkat *data node*.

D. MySQL

MySQL adalah sistem manajemen basis data relasional *open source* yang paling banyak digunakan [20]. Sistem ini

digunakan sebagai *software as a service* (SaaS) melalui *cloud* dan disebut sebagai *relational database management system* (RDBMS). Sistem manajemen basis data ini dirilis pada tahun 1995 dan saat ini didukung oleh Oracle Corporation. Sebagai sistem manajemen basis data yang paling terkenal, sistem ini dapat mendukung penyedia layanan *hosting*, seperti WebHost Manager (WHM), Bluehost, GoDaddy, dan Rackspot.

Selain itu, terdapat beberapa platform media sosial terkenal yang memanfaatkan MySQL, di antaranya adalah YouTube dan Facebook. Begitu pula dengan Amazon yang menggunakan basis data MySQL di *cloud* melalui *hosting* untuk memberikan layanan kepada pelanggannya. Di samping memiliki kelebihan, MySQL memiliki keterbatasan dalam hal keandalan, keamanan, kegunaan, dan kinerja.

Beberapa organisasi swasta memiliki sistem manajemen basis data untuk tujuan khusus, meskipun di pasaran terdapat banyak jenis RDBMS yang mendukung *cloud*. Namun, studi riset pasar menunjukkan bahwa dengan lebih dari sepuluh juta instalasi, MySQL merupakan RDBMS yang paling populer di *cloud*.

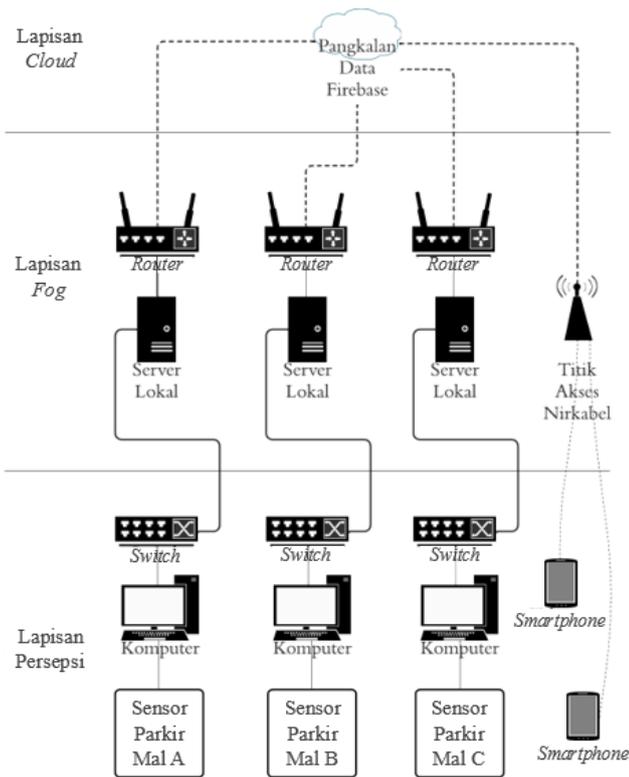
MySQL juga menggunakan koneksi antara klien dan server yang tidak terenkripsi secara *default* [21]. Fitur ini sangat penting untuk pengelolaan parkir yang aman dan terlindungi. Fitur ini menggunakan nama klien atau alamat IP, nama pengguna, dan kata sandi. MySQL versi 5.0 dan versi yang lebih tinggi bahkan memiliki kemampuan untuk memeriksa *string* kompleks yang digunakan sebagai kata sandi. Namun, MySQL memiliki kerentanan dalam hal keamanan, termasuk mode operasi, cara *padding*, dan *key derivation*, yang merupakan bagian dari prinsip *inner encryption* [22]. MySQL juga telah diimplementasikan pada sistem kontrol yang terintegrasi karena MySQL merupakan sistem manajemen basis data yang dapat memudahkan dokumentasi dan pelaporan [23].

Oleh karena itu, tidak hanya mengimplementasikan MySQL, penelitian ini juga mengimplementasikan basis data Firestore untuk memberikan lapisan keamanan lebih. Selain itu, sinkronisasi *real-time* dari MySQL ke Firebase dihubungkan dengan menggunakan *JsonNode*, dengan tujuan untuk mengamankan dan melindungi data. Keamanan basis data MySQL dan Firestore pada lapisan *cloud* dapat mengelola basis data parkir dengan lebih aman dan terjamin dibandingkan dengan metode lain yang hanya menggunakan NB-IoT.

III. DESAIN DAN IMPLEMENTASI

A. ARSITEKTUR

Arsitektur sistem yang dibangun dalam penelitian ini ditunjukkan pada Gambar 1. Sistem dibangun berdasarkan komputasi kabut yang terdesentralisasi, yang terdiri atas sistem parkir yang dalam penelitian ini diasumsikan sudah ada. Data-data tersebut disimpan ke dalam setiap server lokal setiap mal dengan MySQL, yang artinya setiap mal memiliki server lokal sebagai sistem penyimpanan berbasis MySQL. Pada penelitian ini, server lokal yang digunakan adalah komputer lokal dengan beberapa basis data sebagai representasi dari server lokal di beberapa mal. *Fog node* dapat mengurangi latensi ketersediaan lahan parkir yang akan diperbarui di *cloud*. Setiap server lokal dihubungkan ke dalam *cloud* dengan menggunakan basis data Firestore. Pada lapisan *cloud* dari basis data Firestore, semua data diamankan dengan menggunakan autentikasi untuk mencegah akses yang tidak sah. Semua data juga dapat dikelola dan dianalisis di lapisan *cloud*, sehingga data *insight* dapat dengan mudah diperoleh untuk keperluan selanjutnya.



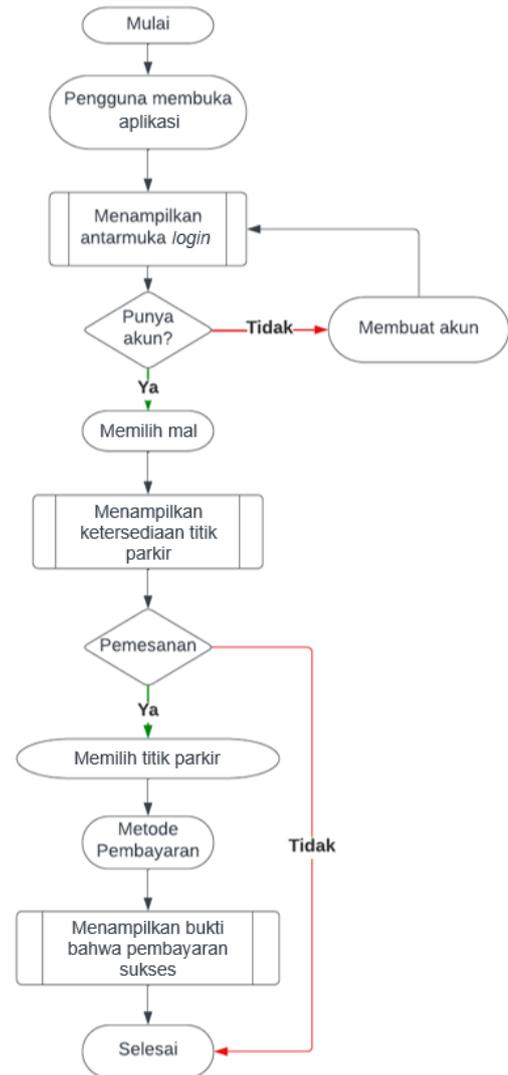
Gambar 1. Arsitektur komputasi kabut untuk parkir pintar yang dibuat.

Koneksi dari server lokal ke basis data Firestore dilakukan dengan menggunakan Node.js. Aplikasi Android yang dibangun, yaitu aplikasi parkir pintar, kemudian dihubungkan ke cloud sebagai antarmuka pengguna yang memungkinkan pengguna untuk berinteraksi dengan sistem dan memperoleh beberapa informasi, seperti ketersediaan tempat parkir, pemilihan mal, pemilihan tempat parkir, dan pemesanan tempat parkir yang diinginkan. Kemudian, koneksi dari aplikasi parkir pintar ke lapisan cloud dari basis data Firestore diamankan dengan cara mengimplementasikan autentikasi untuk aturan membaca dan menulis. Selain autentikasi untuk membaca dan menulis, autentikasi juga diimplementasikan pada registrasi pengguna dan percobaan login.

B. DESAIN SISTEM

Desain sistem dibangun dan dipisahkan menjadi tiga universal multi language (UML) yang berbeda, yang terdiri atas diagram alir pengguna, pseudocode sistem, dan diagram kelas. Semua UML diperlihatkan pada Gambar 2.

Gambar 2 menunjukkan diagram alir dari sisi pengguna. Pada tahap pertama, pengguna membuka aplikasi parkir pintar yang telah dibuat pada penelitian ini dengan menggunakan Android Studio. Setelah membuka aplikasi tersebut, laman baru yang menunjukkan antarmuka login pengguna akan muncul. Pada tahap ini, pengguna harus memasukkan nama yang sudah terdaftar beserta kata sandinya. Jika pengguna belum memiliki nama pengguna, pengguna harus melakukan registrasi terlebih dahulu. Untuk registrasi, pengguna akan dibawa ke laman baru, yaitu laman registrasi. Setelah menyelesaikan registrasi di antarmuka registrasi, aplikasi parkir pintar akan membawa pengguna ke antarmuka login, sehingga pengguna dapat memasukkan dan menggunakan nama pengguna dan kata sandi yang telah terdaftar. Setelah berhasil masuk, laman baru akan menampilkan daftar beberapa mal. Di sini, pengguna harus memilih mal yang akan mereka kunjungi. Setelah memilih mal, aplikasi parkir pintar akan

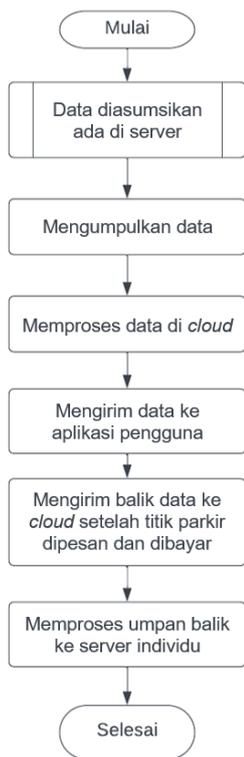


Gambar 2. Diagram alir pengguna.

memberikan daftar tempat parkir di mal yang dipilih beserta ketersediaan tempat parkirnya. Pengguna dapat melanjutkan dengan memilih tempat parkir atau hanya melihat-lihat saja. Setelah memilih tempat parkir dan pengguna ingin memesan tempat parkir tersebut, aplikasi parkir pintar akan mengarahkan pengguna ke laman pembayaran. Di sini, disediakan beberapa metode pembayaran yang dapat digunakan pengguna. Setelah pengguna melakukan pembayaran, laman baru akan menampilkan notifikasi “pembayaran berhasil” yang memberitahukan bahwa pemesanan tempat parkir yang dipilih telah dibayar.

Gambar 3 menunjukkan diagram aktivitas proses dalam sistem parkir pintar ini. Data yang dibutuhkan diasumsikan sudah ada di server lokal. Data tersebut diperoleh dari sensor parkir yang terpasang di lokasi. Data dari sensor parkir diterima dan disimpan ke dalam MySQL di server lokal. Selanjutnya, data dari server lokal diproses ke lapisan cloud yang merupakan basis data Firestore. Data tersebut dikirim ke aplikasi parkir pintar pengguna. Jika terdapat perubahan data dari pengguna lain yang melakukan pemesanan dan pembayaran, data tersebut dikirim kembali ke cloud, sehingga ketersediaan tempat parkir yang telah dipesan oleh pengguna dapat diperbarui. Selanjutnya, cloud akan mengirimkan instruksi ke server lokal masing-masing dan memberi tahu administrator.

Gambar 4 menunjukkan hubungan aktivitas di antara berbagai hal, yang disebut juga sebagai diagram kelas.



Gambar 3. Diagram alir sistem.



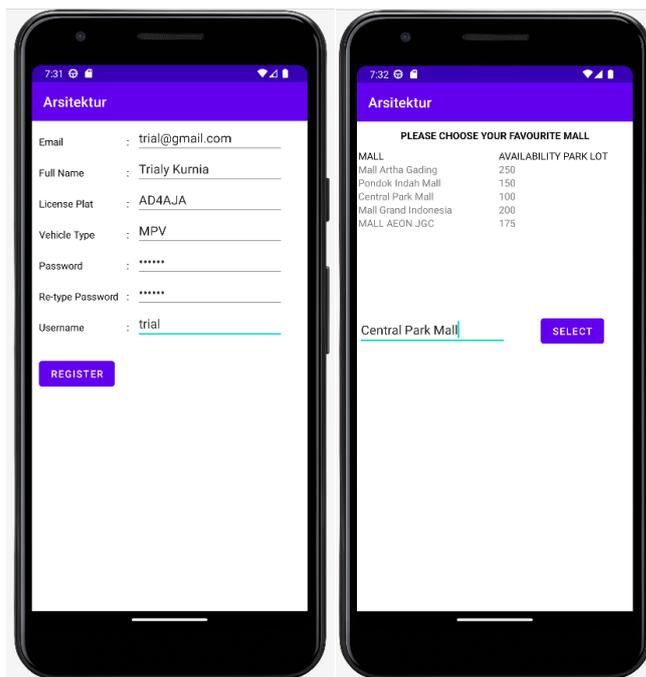
Gambar 4. Diagram kelas.

Konsumen, sebagai pengguna aplikasi parkir pintar, memiliki koneksi dengan mal. Konsumen berhak memilih mal tertentu untuk melihat ketersediaan tempat parkir mal tersebut, kemudian mal menawarkan data tempat parkir yang sesuai. Konsumen juga memiliki koneksi dengan Event Booking. Event Booking ini akan memiliki koneksi dengan *item* yang dibutuhkan, dalam hal ini adalah ketersediaan tempat parkir yang berasal dari server lokal masing-masing mal untuk diimplementasikan pada saat proses pemesanan dan pembayaran.

C. GRAPHICAL USER INTERFACE (GUI)

Penelitian ini juga mengembangkan dan membangun sebuah aplikasi Android, yaitu aplikasi parkir pintar. Aplikasi yang merupakan bagian dari penelitian ini berfungsi untuk sarana interaksi antara pengguna dengan sistem. *Graphical user interface* (GUI) pada penelitian ini dibangun dengan menggunakan Android Studio, sehingga aplikasi parkir pintar dapat berjalan pada *smartphone* dengan sistem operasi Android.

Gambar 5(a) menunjukkan laman registrasi aplikasi parkir pintar. Pengguna yang belum terdaftar harus mendaftar dari laman ini. Pada laman ini, pengguna harus memasukkan beberapa data yang diperlukan, seperti alamat email, kata sandi,



(a) (b)

Gambar 5. *Graphical user interface* (GUI) sistem parkir pintar, (a) laman registrasi, (b) laman daftar mal.

nama lengkap pengguna, plat nomor, jenis kendaraan, dan konfirmasi kata sandi. Konfirmasi kata sandi dilakukan dengan mengetik ulang kata sandi tersebut. Hal ini penting untuk berjaga-jaga jika pengguna tidak sadar telah salah mengetik kata sandi.

Gambar 5(b) menunjukkan daftar mal. Pada laman ini, aplikasi parkir pintar menampilkan daftar mal beserta jumlah tempat parkir yang tersedia. Hal ini dimaksudkan agar pengguna dapat mengetahui ketersediaan tempat parkir di setiap mal, sehingga dapat menentukan tujuan dan menjadikan ketersediaan tempat parkir sebagai bahan pertimbangan. Di sini, pengguna dapat memilih mal yang diinginkan dan menekan tombol "Select" untuk proses selanjutnya. Proses selanjutnya adalah memilih tempat parkir. Selain dua GUI yang disajikan pada Gambar 5, terdapat beberapa laman GUI untuk mendukung sistem parkir pintar yang telah dibuat pada aplikasi parkir pintar Android ini menggunakan Android Studio.

1) TAMPILAN BERANDA APLIKASI

Tampilan beranda mencakup kolom untuk nama pengguna, kata sandi, tombol *login*, dan tombol registrasi. Laman ini didesain untuk *login*. Setelah memiliki akun, pengguna dapat langsung memasukkan nama pengguna dan kata sandi, kemudian menekan tombol "Login." Jika tidak memiliki akun, pengguna dapat menekan tombol registrasi, yang akan mengarahkan pengguna ke laman pendaftaran baru.

2) LAMAN DAFTAR MAL

Laman ini terdiri atas mal-mal yang telah terdaftar dalam sistem parkir pintar. Oleh karena itu, laman ini menampilkan beberapa mal beserta jumlah tempat parkir yang tersedia di setiap mal yang terdaftar.

3) LAMAN PEMILIHAN TEMPAT PARKIR

Pada laman ini, pengguna dapat memilih tempat parkir yang tersedia. Setiap tempat parkir dilengkapi dengan informasi mengenai jenis kendaraan yang dapat diparkir di sana, mengingat beberapa mal memiliki tempat parkir yang dirancang khusus untuk mobil sedan karena adanya plafon

yang rendah. Oleh karena itu, untuk memilih tempat parkir yang diinginkan, pengguna harus melihat informasi tempat parkir tersebut.

4) LAMAN KONFIRMASI PEMESANAN

Pada laman ini, aplikasi parkir pintar menampilkan proses pembayaran, sehingga pengguna dapat menentukan metode pembayaran yang hendak digunakan. Pada laman ini terdapat tombol “Confirm” dan tombol “Cancel.” Tombol “Confirm” berfungsi untuk mengonfirmasi pemesanan, sedangkan tombol “Cancel” berfungsi untuk membatalkan pemesanan yang kemudian mengarahkan pengguna ke laman sebelumnya, yaitu laman pemilihan tempat parkir.

5) LAMAN SUKSES

Laman ini menunjukkan bahwa pemesanan berhasil. Setelah pemesanan berhasil, aplikasi akan mengirimkan umpan balik berupa data yang telah diperbarui ke dalam cloud dari basis data Firestore. Selanjutnya, server lokal MySQL akan diperbarui.

D. RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)

RDBMS didasarkan pada model relasional, sehingga disebut sistem manajemen basis data relasional. Basis data yang paling umum digunakan adalah sistem ini. Sistem ini terdiri atas banyak tabel. Setiap tabel dalam RDBMS harus mengimplementasikan *primary key*. Data dapat diakses dengan mudah dalam RDBMS karena terdiri atas kumpulan tabel yang terorganisasi. Semua yang ada dalam RDBMS disimpan dalam bentuk relasi.

RDBMS diimplementasikan pada MySQL dan disimpan ke dalam server lokal masing-masing mal untuk ketersediaan data tempat parkir. Di sisi lain, data pengguna dan riwayat parkirnya disimpan ke dalam basis data yang berbeda dan berada di penyedia aplikasi parkir pintar. Tabel digunakan oleh RDBMS untuk menyimpan data. Tabel adalah kumpulan elemen data yang terhubung dan menggunakan baris dan kolom untuk menyimpan informasi. Setiap tabel mewakili entitas dunia nyata, seperti orang, lokasi, atau peristiwa, yang menjadi tempat pengumpulan data. Perspektif logis dari basis data adalah kumpulan data yang teratur ke dalam tabel relasional.

Gambar 6 menunjukkan RDBMS penelitian parkir pintar ini. RDBMS ini menunjukkan hubungan antar basis data. Salah satu perhatian utama yang dibahas dalam penelitian ini adalah pengembangan sistem parkir pintar yang dapat diimplementasikan di banyak area cakupan. Sistem parkir pintar ini dirancang untuk dapat terhubung ke berbagai server lokal di berbagai mal. Secara keseluruhan, penelitian ini menggunakan enam basis data.

1) BASIS DATA PENGGUNA (DB1 MEMBUAT NAMA PENGGUNA)

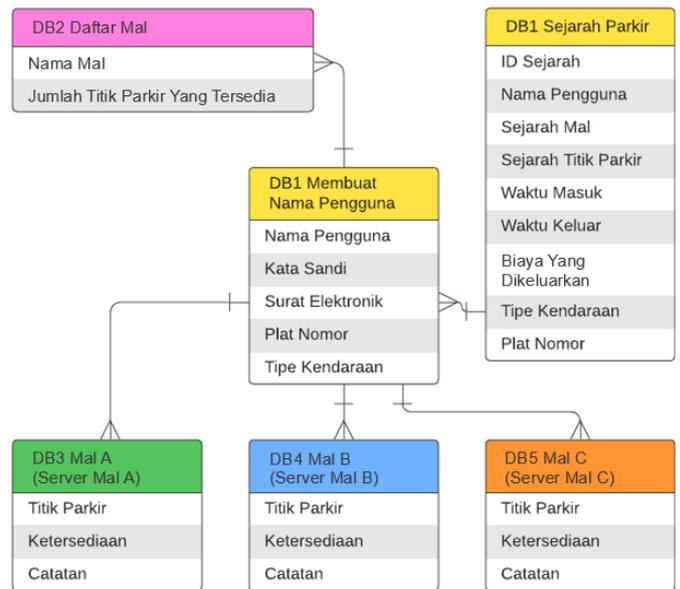
Basis data ini berisi data pengguna yang disimpan pada saat pendaftaran. Basis data ini terdiri atas nama pengguna, kata sandi, alamat email, plat nomor, dan jenis kendaraan. Basis data ini disimpan ke dalam basis data Firestore.

2) BASIS DATA MAL (DB2 DAFTAR MAL)

Basis data ini berisi mal-mal yang terdaftar beserta jumlah tempat parkir yang tersedia. Basis data mal juga disimpan ke dalam basis data Firestore. Jumlah tempat parkir yang tersedia diperbarui dari server lokal MySQL mal tersebut.

3) BASIS DATA PARKIR MAL A (DB3 MAL A)

Basis data ini berisi daftar identitas tempat parkir beserta ketersediaan dan keterangannya di mal A. Keterangan tersebut



Gambar 6. RDBMS pada penelitian ini.

menunjukkan jenis kendaraan yang dapat diparkir di tempat parkir tersebut. Basis data ini disimpan di server lokal mal. Kemudian, data tersebut diteruskan ke basis data Firestore.

4) BASIS DATA PARKIR MAL B (DB4 MAL B)

Basis data ini berisi daftar identitas tempat parkir beserta ketersediaan dan keterangannya di mal B. Keterangan tersebut menunjukkan jenis kendaraan yang dapat diparkir di tempat parkir tersebut. Basis data ini disimpan di server lokal mal. Kemudian, data tersebut diteruskan ke basis data Firestore.

5) BASIS DATA PARKIR MAL C (DB5 MAL C)

Basis data ini berisi daftar identitas tempat parkir beserta ketersediaan dan keterangannya di mal C. Keterangan tersebut menunjukkan jenis kendaraan yang dapat diparkir di tempat parkir tersebut. Basis data ini disimpan di server lokal mal. Kemudian, data tersebut diteruskan ke basis data Firestore.

6) BASIS DATA RIWAYAT PARKIR (DB1 RIWAYAT PARKIR)

Basis data ini berisi riwayat parkir pengguna dan terdiri atas nama pengguna, riwayat mal, riwayat tempat parkir, jam masuk, jam keluar, biaya yang dihabiskan, jenis kendaraan, dan plat nomor. Riwayat parkir digunakan untuk tujuan analisis bisnis. Basis data ini disimpan di basis data Firestore karena memiliki fungsi analitik yang dapat mendukung tujuan ini.

E. MENGMPLEMENTASIKAN AUTENTIKASI DI FIREBASE

Salah satu isu yang dibahas dalam penelitian ini adalah masalah keamanan dan privasi. Sistem parkir pintar terdesentralisasi berbasis komputasi awan dengan menggunakan basis data Firestore ini mengimplementasikan sistem autentikasi berdasarkan fungsi pemblokiran yang mengizinkan dan memblokir pengguna ketika terdaftar ke dalam aplikasi. Sistem ini juga mengizinkan dan memblokir pengguna ketika masuk ke dalam aplikasi. Berikut adalah kode autentikasi tersebut.

```
const functions = require('firebase-functions');

exports.beforeCreate =
functions.auth.user().beforeCreate((user,
context) => {
```

```
});
exports.beforeSignIn =
functions.auth.user().beforeSignIn((user,
context) => {
});
```

Selain menerapkan autentikasi untuk registrasi dan *login*, penelitian ini juga menggabungkan penggunaan pelacakan alamat IP untuk memantau aktivitas yang mencurigakan di dalam sistem parkir pintar. Pendekatan ini memantau alamat IP pengguna selama proses *login* dan membandingkannya dengan alamat IP dari permintaan berikutnya. Kode yang diimplementasikan ditunjukkan sebagai berikut.

```
exports.beforeSignIn =
functions.auth.user().beforeSignIn((user,
context) => {
  return {
    sessionClaims: {
      signInIpAddress: context.ipAddress,
    },
  };
});
```

Ketika pengguna mencoba mengakses sumber daya yang memerlukan autentikasi dengan Firebase Authentication, sistem parkir pintar akan membandingkan alamat IP pengguna dengan IP yang digunakan untuk mencoba *login*. Berikut kode dari metode ini.

```
app.post('/getRestrictedData', (req, res) => {
  // Obtain the past ID token.
  const idToken = req.body.idToken;

  // ID token verification, if revoked then
  check and its payload will be decoded.
  admin.auth().verifyIdToken(idToken,
  true).then((claims) => {

    // IP address during requesting
    const requestIpAddress =
    req.connection.remoteAddress;

    // IP address when signing-in
    const signInIpAddress =
    claims.signInIpAddress;

    // Determine whether the request IP
    address origin is suspicious in comparison to
    the session IP addresses. The current request
    timestamp and the auth_time of the ID token
    can provide additional indications of abuse,
    particularly if the IP address changes
    abruptly. If there is a sudden geographical
    shift in a short period of time, it will give
    stronger indications of suspected abuse.
    if
    (!isSuspiciousIpAddressChange(signInIpAddress,
    requestIpAddress)) {

      // Unusual IP address change. Re-
      authentication is required. You can also call
      admin.auth().revokeRefreshTokens(claims.sub) to
      revoke all user sessions.
      res.status(401).send({error:
      'Unauthorized access. Please login again!'});
    } else {

      // Invalid Access. Attempt to return
      data.
```

TABEL I
BASIS DATA TEMPAT PARKIR MAL A

Titik Parkir	Ketersediaan	Keterangan
LotA1	TRUE	Sedan
LotA2	TRUE	None
LotA3	FALSE	None
LotA4	TRUE	None
LotA5	FALSE	Sedan

Gambar 7. Semua basis data yang terdaftar di Firebase dan basis data tempat parkir mal.

```
getData(claims).then(data => {
  res.end(JSON.stringify(data));
}, error => {
  res.status(500).send({ error: 'Server
error!' });
});
```

IV. HASIL DAN DISKUSI

A. BASIS DATA DI MySQL

Penelitian ini menggunakan MySQL sebagai platform untuk membuat basis data. Basis data dibagi menjadi tiga, yaitu *parking_mall_a*, *parking_mall_b*, dan *parking_mall_c*. Pada ketiga basis data tersebut, hanya terdapat satu tabel yang berisi titik parkir, ketersediaan, dan keterangan seperti yang ditunjukkan pada Tabel I. Pada kolom Titik Parkir, terdapat identifikasi untuk setiap tempat parkir yang ada di mal. Sementara itu, kolom ketersediaan menunjukkan ketersediaan tempat parkir. Terakhir, kolom keterangan menunjukkan informasi tempat parkir, yaitu mengenai jenis kendaraan yang dapat diparkir di tempat tersebut. Keterangan “none” berarti tempat parkir dapat digunakan untuk semua jenis kendaraan. Pada penelitian ini, diasumsikan bahwa data telah dimasukkan ke dalam server.

B. FIREBASE

Firebase adalah sebuah *backend as a service* (BaaS) yang ditawarkan oleh Google untuk memudahkan para pengembang aplikasi mengembangkan sebuah aplikasi (web dan seluler). Dalam proyek penelitian parkir pintar ini, Firebase digunakan sebagai *cloud*. Data yang diperlukan *fog node* disimpan ke dalam *cloud* dan *cloud* juga memiliki kemampuan untuk melakukan autentikasi aktivitas. Di dalam basis data Firestore ini, data dapat dikelola, diamankan, dan dianalisis.

Gambar 7 menyajikan seluruh basis data yang ada di Firebase, seperti *db_customer*. Basis data *db_customer* berisikan nama pengguna, kata sandi, dan hal-hal lain yang telah didaftarkan sebelumnya ke dalam aplikasi. Detail isi *db_customer* adalah nama, nomor polisi, kata sandi, nama pengguna, dan jenis kendaraan.

Gambar 7 juga menunjukkan basis data tempat parkir di mal A. Diasumsikan bahwa tempat parkir tersebut sudah ada di

TABEL II
DOKUMENTASI PENGUJIAN

Skenario Uji Coba	Kasus Uji Coba	Langkah Uji Coba	Hasil	Status
Pengguna mendaftarkan sebagai pengguna baru.	Mendaftarkan pengguna baru.	Masukkan semua data, tekan tombol register, pastikan data terdaftar ke dalam Firebase.	Data tersimpan ke dalam Firebase.	BERHASIL
Cek login.	Mengecek validitas login.	Memasukkan data yang benar dan salah.	Jika hasil benar, selanjutnya akan menuju ke GUI selanjutnya. Jika salah, akan diberitahukan bahwa nama pengguna atau sandi tidak terdaftar.	BERHASIL
Pengguna membuka Aplikasi.	Pengguna membuka aplikasi dan menekan tombol yang tersedia.	1. Pengguna membuka aplikasi.	Akan muncul aplikasi.	BERHASIL
Pengguna membuka pilihan mal.	Pengguna membuka pilihan mal dengan menekan "choose mall".	1. Pengguna memilih mal.	Pengguna berada pada mal yang dipilih.	BERHASIL
Pengguna membuka pilihan titik parkir.	Pengguna membuka pilihan parkir mal pada mal yang dipilih.	1. Pengguna memilih mal. 2. Pengguna memilih titik parkir pada mal yang dipilih.	Pengguna dapat memilih titik parkir.	BERHASIL
Memuat data dari basis data.	Memuat basis data dari Firebase sehingga ketersediaan parkir dapat terlihat.	Pengguna membuka titik parkir dan akan memunculkan ketersediaan parkir.	Pengguna dapat memesan titik parkir.	BERHASIL
Pengguna membuka metode pembayaran.	Pengguna membuka metode pembayaran.	Pengguna membuka metode pembayaran dengan memilih titik parkir.	Pengguna dapat memproses metode pembayaran dan akan muncul "payment success".	BERHASIL
Menghubungkan MySQL lokal ke Google Cloud.	Uji coba pembacaan basis data, tabel, dan data via Google Cloud.	Membuka terminal pada Google Cloud.	Google Cloud dapat membaca data dari MySQL lokal.	BERHASIL
Menghubungkan Firebase Cloud ke Aplikasi.	Uji coba pembacaan dan penulisan pangkalan data ke Firebase Cloud.	Uji coba pada Aplikasi dan Firebase Cloud.	Aplikasi dapat membaca dan menulis dari firebase.	BERHASIL

server. Terdapat lima tempat parkir di mal A. Setiap tempat parkir berisi informasi ketersediaan, nama, dan informasi jenis kendaraan yang dapat diparkir. Misalnya, keterangan di LotA3 menunjukkan "none," yang mengindikasikan bahwa tempat parkir tersebut cocok untuk semua jenis kendaraan. Sebaliknya, tulisan di tempat parkir LotA5 bertuliskan "sedan," menandakan bahwa tempat tersebut hanya diperuntukkan untuk sedan dan bukan kendaraan lain. Penataan tersebut juga diimplementasikan ke dalam basis data tempat parkir di mal B dan mal C.

C. PENGUJIAN

Setelah semuanya dibangun dan dikompilasi menjadi satu, dimulai dari pembuatan basis data pada server lokal, menghubungkan basis data pada basis data Firestore, dan menghubungkan aplikasi Android, selanjutnya dilakukan pengujian sesuai dengan perancangan UML. Hasil pengujian ditunjukkan pada Tabel II.

Terdapat beberapa skenario pengujian yang mewakili cara kerja sistem berdasarkan UML, dimulai dengan skenario pengujian pengguna yang terdaftar sebagai pengguna baru dan diakhiri dengan menghubungkan cloud Firebase ke aplikasi. Hasil seluruh skenario menunjukkan "BERHASIL", yang berarti sistem dapat berjalan dengan baik.

Penelitian ini mengusulkan komputasi kabut pada sistem parkir pintar terdesentralisasi dengan menggunakan Firebase untuk mencakup beberapa area atau mal dalam satu sistem dan antarmuka. Artinya, sistem dapat mengakomodasi beberapa tempat parkir dari beberapa mal beserta keamanan dan privasinya menjadi satu sistem dengan menerapkan autentikasi

di Firebase. Sistem ini juga menawarkan kecepatan data yang lebih tinggi karena terkoneksi dengan infrastruktur jaringan internet, sehingga masyarakat dapat melihat dan memesan terlebih dahulu tempat parkir yang tersedia di mal yang diinginkan. Fitur ini bermanfaat bagi masyarakat karena masyarakat tidak perlu mencari tempat parkir saat tiba. Namun, terdapat kelemahan dari sistem parkir ini, terutama ketika koneksi internet terputus dan sistem parkir tidak dapat digunakan. Untuk mengatasi masalah ini, sistem manual dapat menjadi alternatif.

V. KESIMPULAN

Sistem parkir pintar biasanya menggunakan NB-IoT dan telah diusulkan oleh beberapa peneliti. Namun, sistem tersebut memiliki beberapa kelemahan, seperti kurangnya keamanan dan privasi, rendahnya kecepatan data, tingginya biaya pengembangan, ketergantungannya terhadap sistem nirkabel, dan cakupannya yang hanya meliputi satu area. Penelitian ini mengusulkan komputasi kabut berdasarkan sistem parkir pintar yang terdesentralisasi menggunakan Firebase untuk mencakup beberapa area atau mal yang berbeda. Penelitian parkir pintar ini menggunakan server lokal terdesentralisasi sebagai fog node untuk memfasilitasi pertukaran data yang cepat. Selain itu, penelitian parkir pintar ini menggunakan basis data Firestore untuk melakukan autentikasi, mengamankan, mengelola, dan menganalisis data di cloud. Sementara itu, aplikasi Android parkir pintar digunakan sebagai GUI untuk berinteraksi dengan pengguna, menampilkan informasi, mencari ketersediaan tempat parkir, dan memesan tempat parkir. Aplikasi parkir pintar ini dibuat dengan menggunakan Android Studio. Hasil

penelitian menunjukkan bahwa sistem yang dibuat dan diusulkan bekerja dengan baik dan dapat diimplementasikan untuk mencakup beberapa area atau mal dalam satu sistem. Penelitian selanjutnya disarankan untuk mengimplementasikan dan menghubungkan sistem secara langsung ke server mal, menguji waktu eksekusi, dan melakukan analisis perbandingan. Selain itu, perlu dilakukan studi kelayakan ekonomi untuk implementasi yang sesungguhnya, mengingat dalam penelitian ini server lokal dibangun di komputer lokal dengan menggunakan beberapa basis data.

KONFLIK KEPENTINGAN

Selama melakukan, merancang, menguji, dan menyajikan hasil penelitian ini, penulis menyatakan tidak ada konflik kepentingan.

KONTRIBUSI PENULIS

Konseptualisasi, Haposan Yoga Pradika Napitupulu; metodologi, Haposan Yoga Pradika Napitupulu; perangkat lunak, Haposan Yoga Pradika Napitupulu; validasi, Haposan Yoga Pradika Napitupulu; analisis, Haposan Yoga Pradika Napitupulu; penulisan, Haposan Yoga Pradika Napitupulu; visualisasi, Haposan Yoga Pradika Napitupulu; pengawasan, I Gde Dharma Nugraha.

REFERENSI

- [1] A.J. Kadhim dan J.I. Naser, "Proactive load balancing mechanism for fog computing supported by parked vehicles in IoV-SDN," *China Commun.*, vol. 18, no. 2, hal. 271–289, Feb. 2021, doi: 10.23919/JCC.2021.02.019.
- [2] M. Kong, J. Zhao, X. Sun, dan Y. Nie, "Secure and efficient computing resource management in blockchain-based vehicular fog computing," *China Commun.*, vol. 18, no. 4, hal. 115–125, Apr. 2021, doi: 10.23919/JCC.2021.04.009.
- [3] Y. Zhang, C.-Y. Wang, dan H.-Y. Wei, "Parking reservation auction for parked vehicle assistance in vehicular fog computing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, hal. 3126–3139, Apr. 2019, doi: 10.1109/TVT.2019.2899887.
- [4] C. Tang dkk., "Towards smart parking based on fog computing," *IEEE Access*, vol. 6, hal. 70172–70185, Nov. 2018, doi: 10.1109/ACCESS.2018.2880972.
- [5] F.H. Rahman dkk., "Street parked vehicles based vehicular fog computing: TCP throughput evaluation and future research direction," *2019 21st Int. Conf. Adv. Commun. Technol. (ICACT)*, 2019, hal. 26–31, doi: 10.23919/ICACT.2019.8701912.
- [6] X. Lin dkk., "Application research of NB-IoT technology based on fog computing in intelligent parking system," *2019 IEEE 3rd Adv. Inf. Manag. Commun. Electron. Automat. Control Conf. (IMCEC)*, 2019, hal. 1496–1503, doi: 10.1109/IMCEC46724.2019.8984053.
- [7] Y.-C.P. Chang, S. Chen, T.-J. Wang, dan Y. Lee, "Fog computing node system software architecture and potential applications for NB-IoT industry," *2016 Int. Comput. Symp. (ICS)*, 2016, hal. 727–730, doi: 10.1109/ICS.2016.0150.
- [8] M.A. Hoque dan R. Hasan, "Towards an analysis of the architecture, security, and privacy issues in vehicular fog computing," *2019 SoutheastCon*, 2019, hal. 1–8, doi: 10.1109/SoutheastCon42311.2019.9020476.
- [9] S. Nguyen, Z. Salcic, dan X. Zhang, "Big data processing in fog - smart parking case study," *2018 IEEE Int. Conf. Parallel Distrib. Process. Appl. Ubiquitous Comput. Commun. Big Data Cloud Comput. Soc. Comput. Netw. Sustain. Comput. Commun. (ISPA/TUCC/BDCloud/SocialCom/SustainCom)*, 2018, hal. 127–134, doi: 10.1109/BDCloud.2018.00031.
- [10] B. Cheng, J. Fuerst, G. Solmaz, dan T. Sanada, "Fog Function: Serverless fog computing for data intensive IoT services," *2019 IEEE Int. Conf. Serv. Comput. (SCC)*, 2019, hal. 28–35, doi: 10.1109/SCC.2019.00018.
- [11] E.C. Anderson, K.C. Okafor, O. Nkwachukwu, dan D.O. Dike, "Real time car parking system: A novel taxonomy for integrated vehicular computing," *2017 Int. Conf. Comput. Netw. Inform. (ICCNI)*, 2017, hal. 1–9, doi: 10.1109/ICCNI.2017.8123788.
- [12] M. Celaya-Echarri dkk., "Building decentralized fog computing-based smart parking systems: From deterministic propagation modeling to practical deployment," *IEEE Access*, vol. 8, hal. 117666–117688, Jun. 2020, doi: 10.1109/ACCESS.2020.3004745.
- [13] J.-E. Park dan Y.-H. Park, "Fog-based file sharing for secure and efficient file management in personal area network with heterogeneous wearable devices," *J. Commun. Netw.*, vol. 20, no. 3, hal. 279–290, Jun. 2018, doi: 10.1109/JCN.2018.000040.
- [14] A.M.S. Maharjan dan A. Elchouemi, "Smart parking utilizing IoT embedding fog computing based on smart parking architecture," *2020 5th Int. Conf. Innov. Technol. Intell. Syst. Ind. Appl. (CITISIA)*, 2020, hal. 1–9, doi: 10.1109/CITISIA50690.2020.9371848.
- [15] X. Huang, D. Ye, R. Yu, dan L. Shu, "Securing parked vehicle assisted fog computing with blockchain and optimal smart contract design," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 2, hal. 426–441, Mar. 2020, doi: 10.1109/JAS.2020.1003039.
- [16] A.M. Farooqi, M.A. Alam, S.I. Hassan, dan L. Ansari, "Approaches of shared smart parking model in fog and roadside cloud environment: A detailed survey," *2021 6th Int. Conf. Commun. Electron. Syst. (ICES)*, 2021, hal. 914–924, doi: 10.1109/ICES51350.2021.9489178.
- [17] Y. Sukmana dan Y. Rosmansyah, "The use of Cloud Firestore for handling real-time data updates: An empirical study of gamified online quiz," *2021 2nd Int. Conf. Electron. Commun. Inf. Technol. (CECIT)*, 2021, hal. 1239–1244, doi: 10.1109/CECIT53797.2021.00220.
- [18] K.S. Awaisi dkk., "Towards a fog enabled efficient car parking architecture," *IEEE Access*, vol. 7, hal. 159100–159111, Nov. 2019, doi: 10.1109/ACCESS.2019.2950950.
- [19] J. Tuvakov dan K. Park, "On the fog node model for multi-purpose fog computing systems," *2018 IEEE 9th Annu. Inf. Technol. Electron. Mob. Commun. Conf. (IEMCON)*, 2018, hal. 1211–1214, doi: 10.1109/IEMCON.2018.8614845.
- [20] H.-L. Shieh, W.-S. Chang, S.-F. Lin, dan S.-B. Jhang, "A motorcycle parking lot management system based on RFID," *2013 Int. Conf. Fuzzy Theory Appl. (iFUZZY)*, 2013, hal. 268–272, doi: 10.1109/iFuzzy.2013.6825448.
- [21] I. Zoratti, "MySQL security best practices," *2006 IET Conf. Crime Secur.*, 2006, hal. 183–198.
- [22] L. Zhang, J. Fan, dan Y. Zhou, "The security analysis of MySQL's encryption functions," *2015 Int. Conf. Comput. Sci. Mech. Automat. (CSMA)*, 2015, hal. 5–8, doi: 10.1109/CSMA.2015.8.
- [23] H.Y.P. Napitupulu, "Design and realization of integrated control system based on Microsoft Visual Basic .Net and Mitsubishi's programmable logic controller (PLC) through ethernet cable," *2023 Int. Conf. Comput. Sci. Inf. Technol. Eng. (ICCSITE)*, 2023, hal. 290–295, doi: 10.1109/ICCSITE57641.2023.10127819.