

IMPLEMENTASI CLOUD COMPUTING TERHADAP APLIKASI PEMESANAN PARKIR DI KOTA JAKARTA

¹Muhamad Ramadhani Alfarizi

²Hanum Putri Permatasari

¹Universitas Gunadarma, ramadhani.alfarizi@gmail.com

²Universitas Gunadarma, hanum@staff.gunadarma.ac.id

ABSTRAK

Parkiran merupakan salah satu aspek penting dalam kehidupan bertransportasi di kota Jakarta karena transportasi yang terus berkembang dari waktu ke waktu. Namun, perkembangan transportasi ini tidak diimbangi dengan ketersediaan lahan parkir yang memadai. Oleh karena itu, dibuatlah aplikasi pemesanan parkir berbasis android “G-Parking” yang bertujuan agar memenuhi kebutuhan masyarakat dalam mencari lahan parkir di kota Jakarta. Aplikasi ini dibuat dengan menggunakan android studio dengan bahasa pemrograman kotlin. Aplikasi ini juga diimplementasikan dengan Google Cloud Platform sebagai layanan cloud computing yang digunakan. Tujuan dari diimplementasikannya cloud computing ini adalah agar data pada aplikasi dapat berjalan secara realtime tanpa menggunakan sebuah perangkat keras. Hal ini dapat menghemat biaya pembuatan dan operasional dari aplikasi. keduanya dibuat dan diintegrasikan menggunakan metode Software Development Life Cycle (SDLC). Berikutnya dilakukan pengujian dengan metode black-box testing untuk menguji segala fungsi yang telah dibuat. Selanjutnya dilakukan pengujian User Acceptance Tested (UAT) untuk mengetahui apakah aplikasi sudah memenuhi kebutuhan dan telah layak digunakan. Hasil yang didapat adalah aplikasi ini dapat berjalan dengan baik.

Kata kunci: Aplikasi, Parkiran, Cloud Computing, Android, Jakarta

PENDAHULUAN

Pada era saat ini, perkembangan teknologi semakin berkembang dengan pesat dan cepat terutama pada bidang komputerisasi. Hal ini membuat perusahaan yang bergerak pada bidang komputerisasi mulai melebarkan sayap untuk menyediakan layanan komputer awan atau yang sering dikenal dengan istilah *cloud computing*.

Cloud computing adalah sebuah teknologi yang menggabungkan sejumlah komputer menjadi sebuah sistem yang menyatu dan memungkinkan untuk mengelola sumber daya secara bersamaan melalui jaringan internet. Banyak pengembang maupun perusahaan yang membangun aplikasinya dengan menggunakan *cloud computing*. *Cloud computing* memiliki tiga jenis model layanan menurut *National Institute of Standards and*

Technology (NIST) oleh Mell dan Grance, 2011, yaitu *Platform as a Service* (PaaS), *Infrastructure as a Service* (IaaS) dan *Software as a Service* (SaaS).

Parkir merupakan salah satu komponen atau aspek yang tidak dapat terpisahkan dalam kebutuhan transportasi karena transportasi terus berkembang dari waktu ke waktu. Berdasarkan informasi yang dikeluarkan oleh Badan Pusat Statistik (BPS) provinsi DKI Jakarta, jumlah transportasi di kota Jakarta telah mencapai 11.839.921 unit pada tahun 2019. Jumlah transportasi meningkat sebesar 0,7 persen dari tahun sebelumnya yang mencapai 11.762.763 unit. Peningkatan jumlah transportasi ini tidak diimbangi dengan ketersediaan lahan parkir yang memadai. Hal ini menyebabkan sulitnya masyarakat

dalam mencari lahan parkir untuk kendaraannya.

Berdasarkan survei yang dilakukan oleh perusahaan Uber, waktu yang dihabiskan masyarakat dalam mencari lahan parkir rata-rata mencapai 21 menit per hari. Angka ini menjadi peringatan penting untuk membenahi tata kelola lahan parkir di kota Jakarta. Permasalahan tersebut dapat dipecahkan dengan menciptakan sebuah aplikasi yang dapat memberikan kemudahan kepada masyarakat dalam mencari lokasi lahan parkir untuk kendaraannya. Implementasi *cloud computing* dapat juga digunakan pada jasa, dagang, manufaktur dan industri (Fikri, dkk., 2015), sektor pendidikan (Abidah, Hamdani, Amrozi, 2020) dan bisnis (Rumetna, 2018).

Dengan dibuatnya aplikasi tersebut, maka diharapkan dapat menghemat waktu masyarakat dalam mencari lahan parkir pada saat mengunjungi tempat-tempat tertentu di kota Jakarta. Agar aplikasi dapat berjalan secara optimal, dibutuhkan perangkat keras (*server*) agar data pada aplikasi tersebut dapat berjalan secara *realtime*.

Penggunaan perangkat keras tersebut memerlukan biaya yang tidak sedikit. Hal ini dikarenakan selain membutuhkan biaya untuk melakukan pemasangan, diperlukan juga biaya perawatan agar perangkat keras tersebut tetap bisa berjalan normal.

Dengan perkembangan teknologi saat ini, penggunaan *cloud computing* merupakan salah satu alternatif yang dapat digunakan sebagai pengganti dari kebutuhan perangkat keras dalam menjalankan aplikasi secara *realtime*. Implementasi *cloud computing* dilakukan dengan tujuan untuk mengurangi biaya dari pembuatan aplikasi (Afdhal, 2013) (Iksan, & Arfriandi, 2015), sehingga diharapkan aplikasi yang dibuat dapat

berjalan secara optimal tanpa memerlukan biaya yang mahal.

METODE PENELITIAN

Pengimplementasian *cloud computing* menggunakan *Google Cloud Platform* sebagai layanan *cloud computing* (Krishnan & Gonzalez, 2015) yang akan digunakan dengan model layanan *Infrastructure as a Service* (IaaS). Model penyebaran *cloud computing* yang digunakan adalah *public cloud*. Tujuan dari penerapan layanan *cloud computing* ini adalah untuk menggantikan peran dari perangkat keras (*server*) pada sisi *back-end* aplikasi sehingga nantinya data pada aplikasi tersebut dapat berjalan secara *realtime* tanpa memerlukan sebuah perangkat fisik.

Dalam melakukan implementasi *cloud computing* terhadap aplikasi, dibutuhkan beberapa perangkat keras (*hardware*) dan perangkat lunak (*software*) dengan spesifikasi sebagai berikut, perangkat keras (*hardware*) yakni dengan Processor Intel(R) Celeron(R) CPU 3215U @ 1.70GHz (2 CPUs), ~1.7GHz., Display Intel(R) HD Graphics, DirectX Version: DirectX 12. Adapun untuk perangkat lunak (*software*) terdiri dari Sistem Operasi Windows 10 Pro 64-bit, Browser Google Chrome dan Google Cloud Platform.

Aplikasi pemesanan parkir berjalan di *platform* android. Aplikasi dibuat dengan menggunakan bahasa pemrograman kotlin dan menggunakan perangkat lunak android studio. Di dalam aplikasi ini, terdapat fitur layanan pemesanan parkir, layanan aktivitas pengguna, layanan pencarian lokasi parkir, dan layanan rekomendasi lokasi lahan parkir.

Alur penelitian secara rinci yang digunakan adalah dengan metode *Software Development Life Cycle* (SDLC). Adapun tahapan yang dilakukan yakni, pertama tahap

perencanaan dimana dilakukan pendefinisian tujuan implementasi *cloud computing* dan pemilihan model layanan *cloud computing*. Tahap berikutnya adalah analisis dimana dilakukan analisis kebutuhan perangkat lunak dan pemilihan layanan dari jasa *cloud computing*. Berikutnya tahap perancangan dimana dilakukan perancangan diagram dengan *Unified Modeling Language* sebagai pengembangan perangkat lunak berbasis *object-oriented* (Booch, Jacobson, & Rumbaugh, 1966). untuk mengimplementasikan *cloud computing* ke aplikasi. Selanjutnya, tahap implementasi yakni dilakukan pengimplementasian rancangan diagram *cloud computing* ke aplikasi dengan menggunakan *Google Cloud Platform*. Tahap terakhir adalah

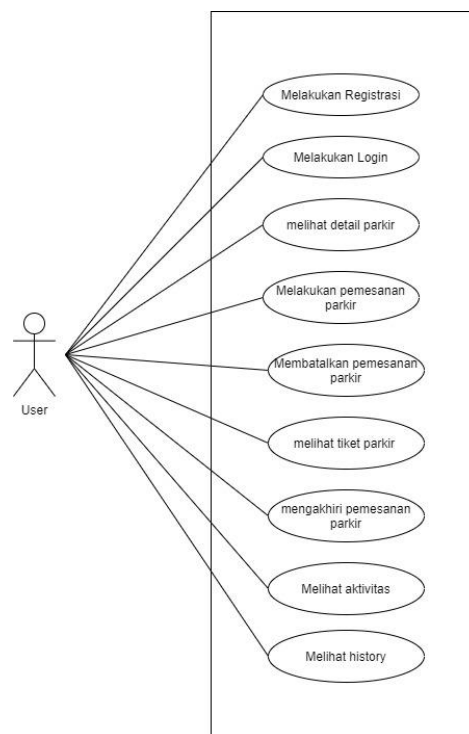
pengujian, yakni dilakukan uji coba terhadap rancangan diagram yang telah diimplementasikan.

HASIL DAN PEMBAHASAN

Pada bagian ini, akan diperlihatkan gambaran diagram UML dari aplikasi pemesanan parkir berbasis android. Diagram UML terdiri dari *use case diagram*, *activity diagram*, dan *class diagram*.

Use Case Diagram

Pada bagian ini, akan diperlihatkan gambaran *use case diagram* dari aplikasi pemesanan parkir berbasis android. Adapun *use case diagram* yang dilakukan oleh pengguna dapat dilihat pada gambar 1.



Gambar 1. Use Case Diagram Aplikasi

Use case diagram berfungsi untuk menggambarkan pengguna aplikasi dan perilaku pengguna terhadap aplikasi. Aksi yang dapat dilakukan oleh pengguna antara lain

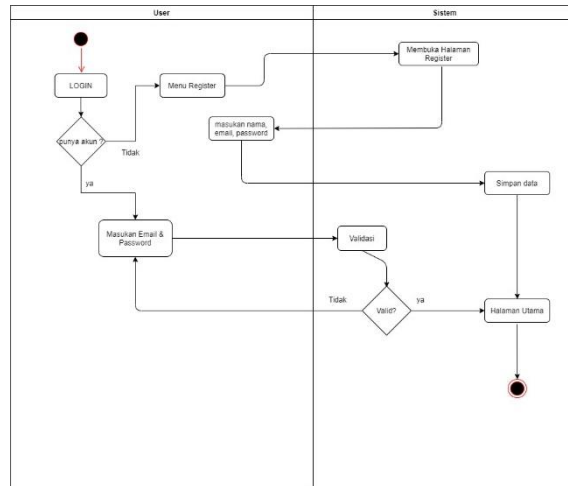
adalah melakukan registrasi akun, melakukan login, melakukan pemesanan parkir, mendapatkan rincian parkir, melihat tiket parkir, mengakhiri pesanan, melihat aktivitas, melihat

histori pengguna, dan melakukan pengaturan akun.

Activity Diagram

Pada bagian ini akan diperlihatkan gambaran *activity*

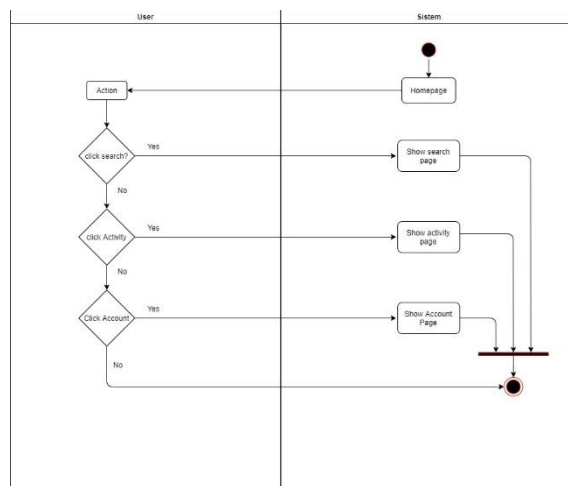
diagram dari aplikasi pemesanan parkir berbasis android. *activity diagram* terdiri dari *activity login* dan *register*, *homepage activity*, dan *booking activity*.



Gambar 2. Activity Login dan Register

Gambar 2. menunjukkan bahwa pengguna akan diarahkan ke halaman *login* terlebih dahulu dan melakukan *login* untuk masuk ke aplikasi. jika pengguna belum memiliki akun, maka pengguna akan diarahkan ke halaman *register* dan melakukan *register* dengan memasukan data nama, *e-mail*, dan *password*. Jika sudah, maka pengguna dapat melanjutkan proses *login*. Pengguna akan diberikan dua opsi pilihan *login* yaitu *login* menggunakan *Gmail* dan *login* menggunakan *e-mail*

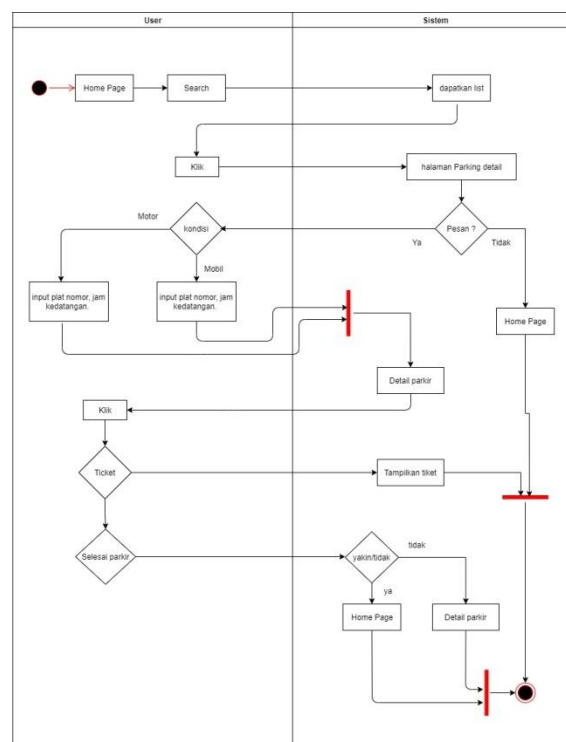
selain *Gmail* dengan memasukkannya secara manual. Jika pengguna ingin menggunakan akun *Gmail*, maka akan muncul *pop up* tampilan dari *Gmail*, lalu pengguna memilih akun yang akan digunakan dalam aplikasi dengan syarat akun tersebut harus sudah terdaftar di dalam sistem. Jika pengguna ingin menggunakan *e-mail* pribadi secara manual, maka pengguna hanya memasukan *e-mail* dan *password* yang telah terdaftar.



Gambar 3. Activity Homepage

Gambar 3. menunjukkan bahwa jika pada saat pengguna berhasil melakukan *login* pada aplikasi, maka halaman yang akan muncul pertama kali adalah halaman utama aplikasi. Terdapat empat menu dalam tampilan halaman utama yaitu menu *search*, *activity*, dan *account*. Ketika menu *search* diklik oleh pengguna, maka pengguna akan diarahkan ke halaman pencarian untuk melakukan pencarian dan pemesanan lahan parkir. Ketika

menu *activity* diklik oleh pengguna, maka pengguna akan diarahkan ke halaman *activity* aplikasi, dimana menu ini berfungsi untuk melihat aktivitas pengguna dalam menggunakan aplikasi. Ketika menu *account* diklik, maka pengguna akan diarahkan ke halaman akun, dimana pengguna dapat melakukan pengaturan terhadap akun yang dimiliki.



Gambar 4. Activity Booking

Gambar 4. menunjukkan bahwa pengguna dapat mencari lokasi parkir dengan menggunakan fitur *search* pada aplikasi. Ketika fitur *search* ditekan, maka pengguna akan diberikan beberapa rekomendasi *list* parkir oleh sistem. Setelah pengguna menekan *list* parkir, maka akan muncul halaman detail parkir yang akan tampil di layar *smartphone* pengguna. Pengguna diberikan dua opsi jenis kendaraan yaitu motor dan mobil. Pengguna dapat memilih antara kedua opsi tersebut berdasarkan kebutuhan yang

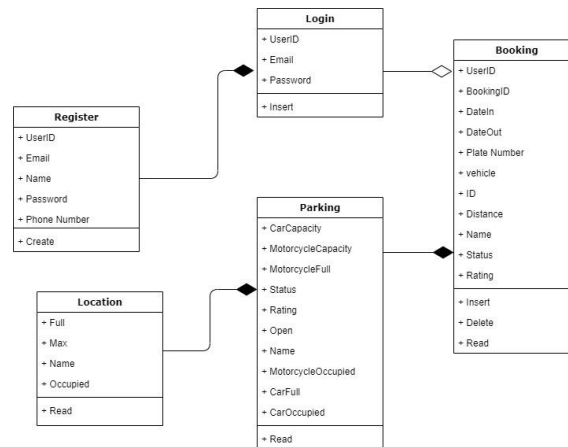
diperlukan. Ketika pengguna telah memilih jenis kendaraan, pengguna wajib mengisi plat nomor dari kendaraan yang dimiliki. Selanjutnya pengguna wajib mengisi waktu pemesanan. Setelah mengisi plat nomor dan waktu pemesanan, maka sistem akan menampilkan tiket parkir yang akan digunakan oleh pengguna untuk mengakses lokasi yang dipesan. Jika pengguna telah selesai menggunakan layanan, maka pengguna wajib menekan tombol *finish booking* lalu

pengguna akan diarahkan ke halaman utama.

Class Diagram

Pada bagian ini akan diperlihatkan rancangan *class diagram* dari aplikasi pemesanan parkir berbasis android. *Class diagram* memiliki tiga area pokok yaitu nama, atribut, dan

operasi. Nama untuk identitas sebuah *class*, atribut untuk memberikan karakter pada data, dan operasi berfungsi untuk memberikan aksi yang akan dilakukan. *Class diagram* pada aplikasi pemesanan parkir berbasis android dapat dilihat pada Gambar 5.

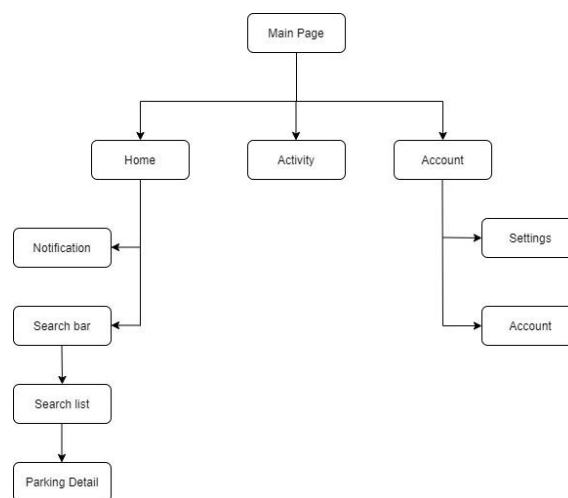


Gambar 5. Class Diagram

Gambar 5. menunjukkan tabel yang digunakan pada *database*. Terdapat lima buah tabel diantaranya, *register*, *login*, *location*, *parking* dan *booking*. Masing-masing tabel terdapat atribut dan aksi yang dapat dilakukan.

Pada aplikasi pemesanan parkir berbasis android ini menggunakan struktur navigasi hirarki. Struktur ini menggunakan percabangan untuk menampilkan data berdasarkan kriteria tertentu. Pada tampilan utama disebut sebagai *master page*, sedangkan untuk tampilan cabang disebut *slavepage*.

Struktur Navigasi

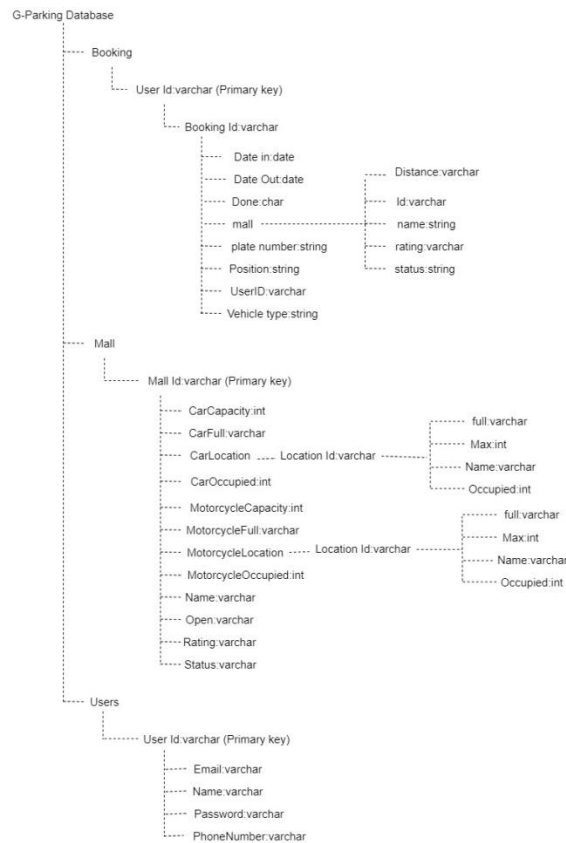


Gambar 6. Struktur Navigasi

Pada struktur navigasi ini beberapa halaman memiliki tujuan yang berbeda bergantung kepada aksi yang dilakukan oleh pengguna. Terdapat empat menu yang ada pada halaman utama, menu tersebut di antaranya adalah menu *home*, menu *activity*, dan menu *account* yang ditunjukkan pada Gambar 6.

Model *database* yang digunakan pada aplikasi pemesanan parkir berbasis android adalah model *database non relational* dengan menggunakan *Google Cloud Firebase realtime-database*. Ja'fahrudin & Tanone, 2017 menggunakan *Firebase Cloud Messaging* pada pembuatan aplikasi objek wisata.

Database



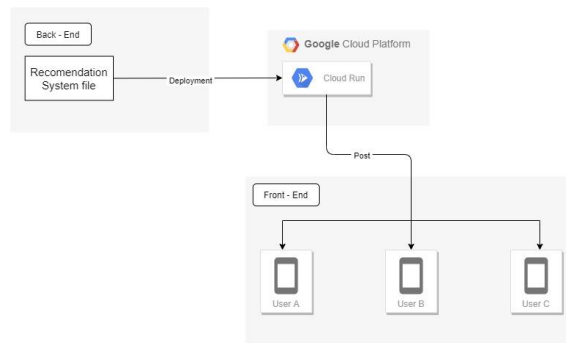
Gambar 7. Rancangan Database

Firebase Realtime Database menggunakan skema model *tree* di mana *database* akan menggunakan *key value* sehingga dapat diakses secara *realtime*. Adapun rancangan *database* yang dilakukan seperti pada Gambar 7.

Perancangan Diagram Cloud Computing

Pada bagian ini, akan diperlihatkan gambaran rancangan diagram dari *cloud computing* yang akan diimplementasikan pada aplikasi

pemesanan parkir berbasis android. Beberapa bagian dari aplikasi yang akan diimplementasikan di antaranya adalah sistem rekomendasi yang akan digunakan untuk merekomendasikan lahan parkir kepada pengguna berdasarkan *rating* lahan parkir (Gambar 8) serta *database* yang akan digunakan sebagai wadah untuk mengumpulkan data pada aplikasi (Gambar 9).



Gambar 8. Rancangan Diagram *Cloud Computing* Sistem Rekomendasi

File dari sistem rekomendasi akan di-*deploy* pada layanan *Cloud Run* yang terdapat pada *Google Cloud Platform* kemudian *file* tersebut akan

diintegrasikan pada bagian *front-end* dengan menggunakan metode *POST*.



Gambar 9. Rancangan Diagram *Cloud Computing Database*

Data pada aplikasi akan disimpan ke dalam sebuah *database* kemudian *database* tersebut akan diintegrasikan dengan *Google Cloud firebase Realtime Database*. Diagram *cloud computing* tersebut terdiri dari beberapa bagian, yaitu *back-end*, *cloud computing*, dan *front-end*. Adapun gambaran dari rancangan diagram dapat dilihat pada gambar di atas.

Implementasi dan Uji Coba

Pada tahap ini, semua perancangan yang telah dibuat akan dikonversi ke dalam bahasa pemrograman serta akan dilakukan

implementasi dari rancangan diagram *cloud computing* yang telah dibuat ke *Google Cloud Platform*. Selanjutnya akan dilakukan pengujian menggunakan metode *black-box testing*.

Implementasi Tampilan Awal

Gambar 10. merupakan tampilan awal saat aplikasi pertama kali dibuka. Terdapat dua tombol yang dapat diklik oleh pengguna. Kedua tombol tersebut yaitu tombol *register* dan tombol *login*.



Gambar 10. Halaman Awal

Tampilan halaman login berfungsi agar pengguna dapat mengakses aplikasi. Pada tampilan ini, pengguna diminta mengisi *e-mail* dan *password* yang telah didaftarkan pada halaman *register*. Jika pengguna klik Ayo Daftar, maka muncul tampilan halaman registrasi untuk melakukan input data serta pembuatan akun pengguna.

Implementasi Tampilan Login

Gambar 11. merupakan tampilan halaman *login* agar dapat mengakses aplikasi. Pada tampilan ini, pengguna diminta mengisi *e-mail* dan *password* yang telah didaftarkan pada halaman *register*.

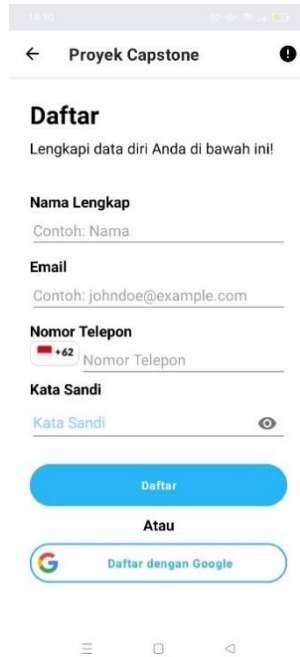


Gambar 11. Halaman Login

Implementasi Tampilan Register

Gambar 12. merupakan tampilan halaman *register* untuk

melakukan *input* data serta pembuatan akun pengguna.



Gambar 12. Halaman Register

Implementasi Tampilan Utama

Gambar 13. merupakan tampilan halaman utama yang merupakan halaman pertama yang akan muncul setelah pengguna melakukan

login. Pada halaman ini terdapat tiga menu utama yaitu menu *home*, *activity*, dan *account*.

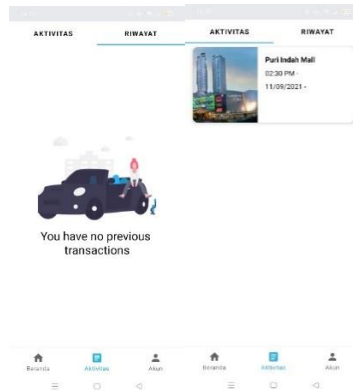


Gambar 13. Halaman Utama

Implementasi Tampilan Menu Aktivitas

Gambar 14. merupakan tampilan halaman aktivitas yang akan diakses oleh pengguna melalui menu pada halaman utama. Terdapat dua menu pada halaman aktivitas ini. Kedua menu tersebut yaitu menu *activity* dan

menu *history*. Apabila pengguna melakukan pemesanan parkir, maka *list* pesanan tersebut akan muncul pada menu *activity*. Apabila pemesanan parkir telah selesai, maka *list* pesanan parkir tersebut akan masuk ke menu *history*.



Gambar 14. Halaman Menu Aktivitas

Implementasi Rancangan Diagram Cloud Computing

Pada tahap ini akan dilakukan langkah-langkah pengimplementasian dari rancangan *cloud computing* yang telah dibuat. Pengimplementasian ini dibagi menjadi dua tahap. Tahap pertama yaitu melakukan implementasi sistem rekomendasi menggunakan *Google Cloud Run*. Selanjutnya tahap kedua yaitu melakukan implementasi *database* menggunakan *Google Cloud Firebase Realtime-Database*.

Pengujian *Black-box Testing*

Black-box testing adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak. Pada tahap ini dilakukan pengujian untuk menentukan kendala yang tersedia pada aplikasi saat diintegrasikan dengan *cloud computing*. Metode pengujian yang digunakan untuk menguji aplikasi berdasarkan masukan yang diberikan oleh pengguna untuk mendapatkan

hasil yang diharapkan dengan metode *black-box testing*.

User Acceptance Test (UAT)

Setelah melakukan pengujian aplikasi, tahap selanjutnya adalah pengujian terhadap penerimaan pengguna. Pengujian ini dilakukan kepada tiga belas (13) responden yang bekerja pada lembaga JAKI untuk mengetahui apakah aplikasi dapat dijalankan dengan baik dan fitur yang ada telah sesuai dengan kebutuhan pengguna. Pengujian dilakukan dengan mekanisme sebagai berikut:

1. Meminta responden untuk membuka aplikasi.
2. Meminta responden untuk melihat tampilan aplikasi dan menggunakan fitur-fitur yang terdapat pada aplikasi.
3. Meminta responden untuk menjawab pertanyaan yang telah diberikan melalui google form dan mencatat jawaban yang diberikan.

Berikut adalah hasil dari kuesioner.

Tabel 1.
Hasil Kuesioner Responden

Pertanyaan	Sangat Baik (SB)	Baik (B)	Kurang Baik (KB)
Apakah aplikasi pemesanan parkir ini sangat mudah digunakan dan di operasikan?	53,8%	46,2%	0%

Tabel 1 lanjutan

Apakah perpaduan warna pada aplikasi terlihat menarik?	46,2%	46,2%	7,7%
Apakah setiap fitur pada aplikasi dapat berjalan dengan baik	30,8%	61,5%	7,7%
Bagaimana menurut Anda tampilan pada aplikasi secara keseluruhan?	38,5%	61,5%	0%
Bagaimana pendapat Anda tentang tampilan tiap-tiap halaman dari aplikasi ini?	30,8%	69,2%	0%
Apakah penempatan komponen pada aplikasi terlihat sesuai?	46,2%	53,8%	0%

KESIMPULAN DAN SARAN

Pengimplementasian *cloud computing* terhadap aplikasi pemesanan parkir berbasis android telah berhasil dilakukan. Pengimplementasian ini menggunakan *Google Cloud Platform* sebagai layanan *cloud computing* dengan model layanan *Infrastructure as a Service* (IaaS). Adapun yang diimplementasikan yaitu sistem rekomendasi dan database dari aplikasi. Dengan diimplementasikannya *cloud computing* terhadap aplikasi, maka kebutuhan akan perangkat fisik (*server*) sudah tidak diperlukan lagi sehingga dapat menghemat biaya pembuatan dan operasional dari aplikasi. Aplikasi pemesanan parkir ini juga telah berhasil dibuat menggunakan android studio dan bahasa pemrograman kotlin. Keduanya telah diujikan dengan menggunakan metode black-box testing dan telah layak untuk digunakan. Terakhir, dilakukan *User Acceptance Test* (UAT) untuk pengujian terhadap penerimaan pengguna dengan memberikan kuesioner kepada responden. Dengan

adanya aplikasi ini, maka dapat menjadi solusi untuk mengatasi permasalahan lahan parkir yang ada di kota Jakarta.

Aplikasi ini dapat diimplementasikan dengan *Google Maps API* yang terdapat pada *Google Cloud Platform*, sehingga pengguna dapat melakukan pemesanan parkir berdasarkan lokasi dari lahan parkir melalui fitur google maps. Selain itu aplikasi ini juga dapat diimplementasikan *Google Cloud Firebase Authentication* sehingga user dapat login menggunakan akun google yang dimiliki. aplikasi ini masih dapat dikembangkan dengan mengintegrasikan *e-wallet*, sehingga tidak perlu lagi dilakukan pembayaran fisik saat selesai menggunakan lahan parkir. Selain itu, fitur pengaturan pada menu akun aplikasi dapat diaktifkan sehingga pengguna dapat melakukan setingan sesuai keinginannya.

DAFTAR PUSTAKA

Abidah, I. N., Hamdani, M. A. & Amrozi, Y. (2020). Implementasi sistem basis data cloud computing pada sektor pendidikan. Vol.1 (2), 77-84, Agustus 2020. Diakses

- tanggal 16 Juni 2021 dari <https://journal.ubaya.ac.id/index.php/saintek/article/view/2868>.
- Afdhal. (2013). Studi perbandingan layanan cloud computing. *Jurnal Rekayasa Elektroika*, vol.10, no.4, pp.193-201.
- Booch, G., Jacobson, I. & Rumbaugh, J. (1966). The unified modeling language for object-oriented development, <https://www2.ccs.neu.edu/research/demeter/course/f96/readings/uml9.pdf>. Diakses tanggal 17 Juni 2021.
- Fikri, Abdillah, L. A., & Apriyani, E. (2015). Perancangan teknologi cloud untuk penjualan online kain songket Palembang. *Seminar Nasional Sistem Informasi Indonesia*, pp.387-392.
- Iksan, N. & Arfriandi, A. (2015). Pengendali listrik rumah berbasis cloud computing. *Jurnal Infotel*, vol.7, no.1, pp.53-60.
- Ja'fahrudin & Tanone, R. (2017). Penggunaan teknologi mobile backend as a service (mbaaS) backendless dan firebase cloud messaging pada perancangan aplikasi travel Marketplace Berbasis Android. Salatiga: Universitas Kristen Satya Wacana.
- Krishnan, S. P. T. & Gonzalez, J. L. U. (2015). *The google cloud platform difference*. ISBN 9781484210048. Berkeley: Apress.
- Mell, P. & Grance, T. (2011). The nist definition of cloud computing, <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>. Diakses tanggal 17 Juni 2021.
- Rumetna, M. S. (2018). Pemanfaatan cloud computing pada dunia bisnis: studi literatur. Vol.5, No. 3, Agustus 2018. Diakses tanggal 16 Juni 2021 dari <https://jtiik.ub.ac.id/index.php/jtiik/article/view/595>.