

Aplikasi Android Loker Penyimpanan *Keyless* Berbayar pada Stasiun MRT (*Mass Rapid Transportation*) Kota Jakarta

Haykhal Bhatara Dimas Arief¹, Viving Frendiana²

^{1,2} Program Studi Broadband Multimedia, Jurusan Teknik Elektro, Politeknik Negeri Jakarta
Jl. Prof. DR. G. A. Siwabessy, Kampus Universitas Indonesia, Depok, Indonesia, 16425

E-mail: haykhal.bhataradimasarief.te17@mhs.w.pnj.ac.id

Abstrak

MRT atau *Mass Rapid Transportation* adalah salah satu layanan transportasi yang dibangun untuk mengatasi kemacetan di Jakarta. Tujuan utama dari dibangunnya MRT tersebut adalah memberikan kemudahan masyarakat dalam upaya meningkatkan mobilitas yang andal, nyaman dan terjangkau. Meningkatnya jumlah masyarakat yang menggunakan transportasi tersebut, tentu harus beriringan dengan fasilitas yang menunjang dalam berkegiatan sehari-hari salah satunya adalah fasilitas penyimpanan barang yang aman. Sebagai solusi dari permasalahan tersebut, maka perlu dibuat suatu rancangan sistem yang dapat menyediakan penyimpanan berbentuk loker yang efisien dan aman yaitu Loker Penyimpanan *Keyless* Berbayar pada Stasiun MRT Kota Jakarta. Sistem ini terdiri dari loker dan aplikasi yang saling terhubung. Aplikasi ini terintegrasi dengan *Firebase Realtime Database* sebagai koneksi antara aplikasi dengan loker. Fungsi dari aplikasi adalah tempat pengguna melakukan pemesanan loker berdasarkan lokasi stasiun yang diinginkan dan untuk membuka loker. Kemudian sebagai alat proses pembayaran sesuai dengan pilihan waktu pemakaian dan menentukan loker yang digunakan dengan memindai *QR code* yang muncul pada aplikasi ke ESP32-CAM yang berfungsi sebagai pemindai. Pengujian yang dilakukan untuk aplikasi ini merupakan pengujian yang mengacu pada standar ISO 25010 dengan aspek *functional suitability*, *compatibility*, *portability*, *performance efficiency*, dan *usability*. Hasil pengujian aspek *functional suitability*, *compatibility*, dan *portability* mendapatkan hasil presentase kelayakan 100%. Kemudian untuk aspek *performance efficiency* dilakukan pada dua perangkat dengan hasil kinerja CPU (*Central Processing Unit*) sebesar 9,973% dan 1,58% serta hasil kinerja memori sebesar 128,521 MB dan 61,199 MB. Pengujian terakhir yaitu aspek *usability* menunjukkan bahwa dari sepuluh poin kuesioner yang disebar kepada responden mendapatkan hasil delapan poin kuesioner sangat layak dan dua poin kuesioner layak.

Keywords: MRT, Aplikasi, Database QR code, ISO 205010

Abstract

MRT or Mass Rapid Transportation is one of the transportation services built to overcome congestion in Jakarta. The main purpose of the construction of the MRT is to provide convenience for the community in an effort to increase mobility that is reliable, comfortable and affordable. The increasing number of people who use this transportation, of course, must go hand in hand with supporting facilities in daily activities, one of which is safe storage of goods. As a solution to these problems, it is necessary to design a system that can provide storage in the form of an efficient and safe locker, namely a Paid Keyless Storage Locker at the Jakarta City MRT Station. This system consists of lockers and applications that are connected to each other. This app is integrated with the Firebase Realtime Database as a connection between the app and the locker. The function of the application is where the user places an order for a locker based on the desired station location and to open the locker. as a processing tool according to the choice of time and determine the locker used with the QR code payment that appears on the application to the ESP 32-CAM which functions as a function. The tests carried out for this application are tests that refer to the ISO 25010 standard with aspects of functional suitability, compatibility, portability, performance efficiency, and usability. The results of testing aspects of functional suitability, compatibility, and portability get 100% feasibility presentation results. Then for the performance efficiency aspect, it was carried out on two devices with CPU (Central Processing Unit) performance results of 9.973% and 1.58% and memory performance results of 128,521 MB and 61,199 MB. The last test, namely the usability aspect, showed that from the ten-point questionnaire distributed to the respondents to get the results, eight points were very feasible and two questionnaire points were feasible.

Keywords: MRT, Application, Database, QR code, ISO 205010

1. Pendahuluan

Teknologi saat ini berkembang sangat pesat. Hal ini tentunya mendukung banyak aspek kehidupan manusia. Sehingga aspek-aspek tersebut mengubah perilaku dan budaya sebagian besar masyarakat kota dari layanan-layanan yang serba konvensional beralih ke layanan yang digital, instan dan efisien. Perubahan perilaku masyarakat tersebut yang menginginkan kemudahan, salah satunya kemudahan bermobilisasi, yang tentu sangat tidak relevan dengan kemacetan di Jakarta yang tinggi. Rata-rata jumlah pertumbuhan kendaraan bermotor dalam lima tahun terakhir mencapai angka 9,5% per tahun, sedangkan pertumbuhan pada ruas jalan hanya 0,1% per tahunnya [1]. Hal ini menunjukkan bahwa dalam beberapa tahun kedepan jalan di Jakarta tidak dapat lagi untuk menampung luapan jumlah kendaraan yang terus tumbuh melebihi panjang jalan yang ada. Berdasarkan studi tersebut maka DKI Jakarta membutuhkan alat transportasi yang dapat mencakup banyak orang demi menunjang segala aktifitas yang dilakukan oleh warga Jakarta [2]. Alat transportasi masal tersebut yang sudah beroperasi dalam rangka menangani masalah tersebut salah satunya adalah MRT. Tujuan utama dari dibangunnya MRT tersebut adalah memberikan kemudahan masyarakat dalam upaya meningkatkan kuliatas mobilitas yang andal, nyaman dan terjangkau.

Meningkatnya jumlah masyarakat yang menggunakan transportasi tersebut, tentu harus beriringan dengan fasilitas-fasilitas yang menunjang dalam berkegiatan sehari-hari salah satunya belum adanya fasilitas penyimpanan barang yang aman, dengan lokasi yang strategis, efisien dan terjangkau untuk menyimpan barang-barang penting yang diperlukan ketika beraktifitas.

Sebagai solusi dari permasalahan di atas, maka perlu dibuat suatu penelitian mengenai rancangan suatu sistem yang dapat menyediakan penyimpanan berbentuk loker yang efisien dan aman yaitu Loker Penyimpanan *Keyless* Berbayar pada Stasiun MRT. Sistem ini terdiri dari loker sebagai penyimpanan yang digunakan pengguna untuk menyimpan barang yang dimiliki yang tersambung dengan aplikasi yang dapat diakses pengguna dalam mengoperasikan loker tersebut [3]. Sistem ini juga dilengkapi dengan kunci yang hanya bisa diakses melalui aplikasi masing-masing penggunaannya sehingga hanya pengguna tersebutlah yang bisa mengakses buka dan tutupnya loker tersebut. Dengan jumlah pengguna MRT yang kian meningkat, prospek pembuatan loker ini kedepannya tentu akan sangat dibutuhkan. Hal ini mendukung untuk memudahkan masyarakat lebih luas lagi untuk efisiensi dan efektivitas dalam mobilisasi sehari-hari.

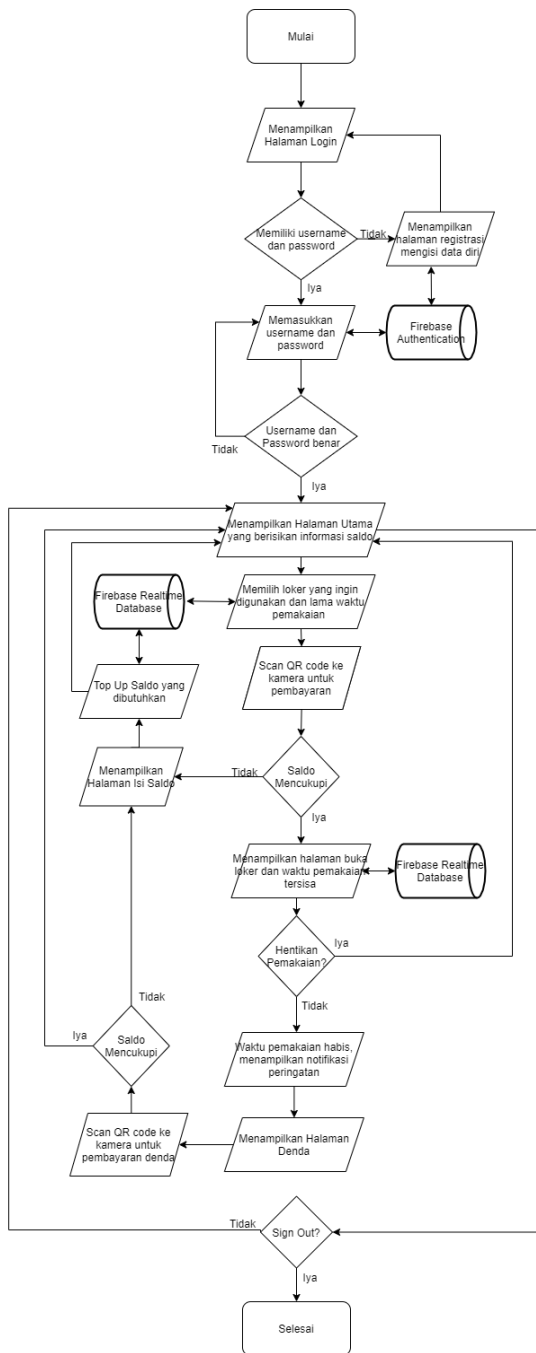
2. Metode Penelitian

Sistem Loker *Keyless* Berbayar Pada Stasiun MRT merupakan alat yang berfungsi untuk penyimpanan barang-barang, baik barang sehari-hari atau berharga lainnya di Stasiun MRT dari *user* yang tentunya penumpang dari transportasi MRT Jakarta. Alat ini menggunakan sistem *keyless* atau tanpa kunci, loker ini bisa disewa oleh *user* sesuai dengan waktu pemakaiannya. Sistem *keyless* pada loker ini adalah dengan menggunakan *interface* “*open & close*” pada aplikasi, sehingga *user* hanya perlu mengklik “*open & close*” setelah melakukan payment dengan sistem *QR code* untuk menggunakan loker tersebut [4]. Dalam pembuatannya, akan dibuat rancang bangun dua loker pada area stasiun dengan masing-masing area terdapat dua pintu loker bertumpuk, sehingga total loker yang digunakan adalah 4 pintu loker.

Program aplikasi ini berupa aplikasi mobile berbasis android bernama “*Smart Storage*”. Mobile app “*Smart Storage*” dapat digunakan oleh *user* yang telah melakukan registrasi untuk masuk dan menggunakan sistem loker berbayar. Pada aplikasi yang dibuat memiliki fungsi untuk menampilkan data terkait ketersediaan loker, *interface* untuk membuka loker serta sistem pembayaran. Aplikasi Mobile “*Smart Storage*” akan terintegrasi dengan database *real-time* dari Firebase yang akan membuat aplikasi “*Smart Storage*” akan selalu melakukan perbaruan jika alat bekerja [5]. Data yang didapatkan dari *database* akan menampilkan hasil data nilai pemantauan.

Pada Gambar 1 diketahui cara kerja dari aplikasi berdasarkan *flowchart*. Berikut merupakan penjelasan dari cara kerja aplikasi:

- 1) Pengguna mengaktifkan aplikasi *Smart Storage* yang telah terinstal pada perangkat androidnya.
- 2) Melakukan proses *login* apabila pengguna sudah memiliki *username* dan *password* yang telah teregistrasi. Apabila belum memilikinya maka pengguna harus melakukan proses registrasi dengan mengisikan data diri berupa nama, nomor telepon, email, dan *password*. Data-data yang telah pengguna masukkan pada halaman registrasi akan tersimpan di Firebase *realtime database* kemudian *username* dan *password* kemudian akan tersimpan juga pada Firebase *authentication*. Pada proses login fungsi Firebase *authentication* akan mencocokkan apakah *username* dan *password* pengguna telah terdaftar.
- 3) Setelah proses *login*, pengguna akan mendapatkan akses menuju ke halaman utama yang berisikan informasi mengenai saldo yang dimiliki pengguna serta tombol untuk menuju fitur utama dari aplikasi ini yaitu pemesanan loker, halaman pengisian saldo, halaman profil dan tombol *sign out* untuk kembali ke halaman *login*.



Gambar 1. Diagram Alir Cara Kerja Aplikasi

- 4) Apabila pengguna akan melakukan pemesanan loker, maka pengguna akan menuju halaman untuk memilih loker yang akan digunakan berdasarkan stasiun tempat pengguna ingin menggunakan loker.
- 5) Setelah memilih lokasi loker, pengguna akan diberi pilihan untuk menggunakan loker dalam rentang waktu tertentu sesuai dengan kebutuhan pengguna.
- 6) Setelah itu, maka akan terbuka halaman di mana pengguna harus melakukan pembayaran sesuai dengan pemakaian. Proses akhir dari paymentasi tersebut adalah melakukan scan QR code yang

muncul pada gawai pengguna. QR code di scan menggunakan kamera yang sudah tertera pada loker Penyimpanan tersebut.

- 7) Terakhir, setelah proses pembayaran, maka pengguna akan ditujukan ke halaman buka loker yang berisikan tombol untuk membuka pintu loker yang akan digunakan serta sisa waktu pemakaian loker tersebut. Apabila waktu pemakaian loker telah habis maka akan dikirimkan notifikasi ke gawai pengguna. Apabila sudah muncul notifikasi tetapi pengguna belum melakukan tindakan lanjutan terhadap penggunaan loker maka akan dikenakan denda.

Tabel 1. Spesifikasi Perangkat Keras

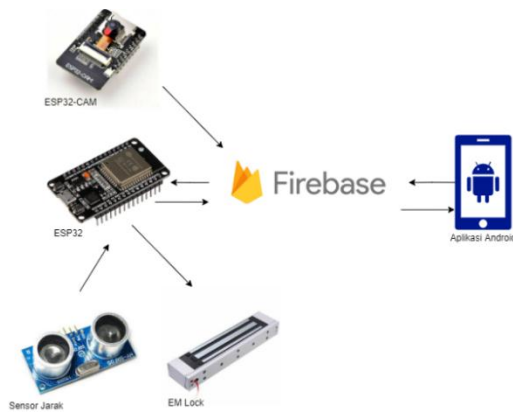
No.	Nama Hardware	Spesifikasi
1.	Laptop ASUS X455LJ	Operating System 64bit; Kapasitas RAM 8GB; OS Windows 10.
2.	Handphone Samsung Galaxy A50S	OS Android R 11; Kapasitas RAM 4GB; Ukuran layar 6.4” 1080x2340 pixels

Tabel 2. Spesifikasi Perangkat Lunak

No.	Nama Software	Spesifikasi
1.	Windows 10	Operating System 64bit; Kapasitas RAM 8GB; OS Windows 10 Enterprise.
2.	Android Studio IDE	Versi 4.2.1
3.	OS Android	OS Android R 11

Pada Tabel 1 dan Tabel 2 merupakan daftar perangkat lunak dan perangkat keras yang digunakan dalam perancangan aplikasi. Untuk pembuatan aplikasi ini menggunakan Android Studio IDE (*Integrated Development Environment*) [7],[8] sebagai *text editor* dan sebagai perangkat lunak dalam pembuatan aplikasi android dan *Google Firebase* sebagai *realtime database* dan autentikasi pengguna.

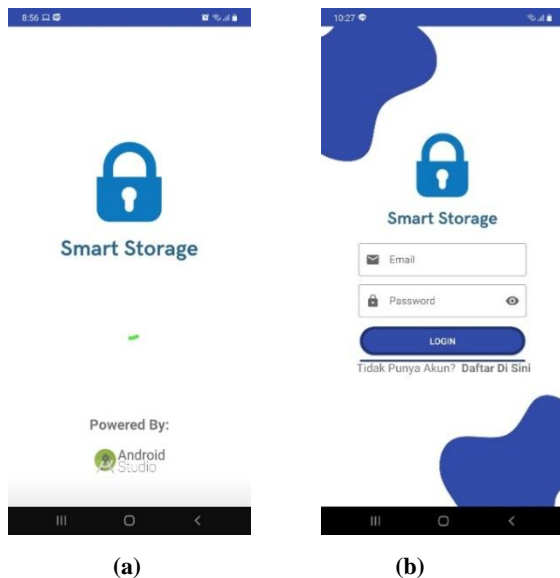
Visualisasi sistem aplikasi digambarkan pada Gambar 2. Data dari aplikasi android terdiri dari dua arah ke *database* yaitu *Google Firebase* [6]. Kemudian data yang terdapat pada *database* menjadi masukan untuk mikrokontroler ESP32. Hasil pembacaan sensor pada ESP32 kemudian akan dikirimkan ke *database* yang kemudian dapat menjadi masukan pada aplikasi android [10]. Data pembacaan QR code oleh modul kamera ESP32-CAM akan dikirimkan ke *database* dan kemudian data tersebut juga akan menjadi masukan untuk aplikasi android.



Gambar 2. Topologi Sistem Pada Aplikasi

3. Hasil dan Pembahasan

Halaman *splash screen* merupakan halaman pertama yang muncul ketika aplikasi dijalankan dengan tujuan memberikan jeda dan memberitahukan identitas aplikasi tersebut. Hasil realisasi dari halaman *splash screen* ditunjukkan pada Gambar 3a. *splash screen* berisikan logo dan komponen *progressbar*.

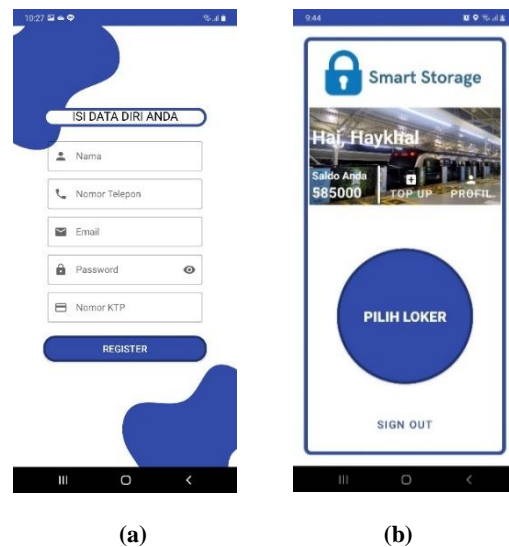


Gambar 3. Halaman (a) *Splashscreen* (b) *Login*

Halaman *login* merupakan halaman di mana pengguna akan memasukkan kredensial berupa *email* dan *password* yang telah teregistrasi oleh fitur *firebase authentication* untuk memastikan apakah pengguna yang akan melakukan proses *login* merupakan pengguna yang telah terdaftar pada *database*. Setelah kredensial pengguna dinyatakan terdaftar oleh *firebase authentication*, maka pengguna akan diarahkan menuju halaman utama. Apabila kredensial pengguna tidak terdaftar atau terjadi kesalahan maka akan muncul

notifikasi langsung dari *firebase*. Hasil realisasi dari halaman *login* digambarkan pada Gambar 3b.

Halaman *register* adalah halaman di mana pengguna memasukkan data diri berupa nama, nomor telepon, *email* dan *password* yang nantinya *email* dan *password* tersebut akan menjadi kredensial yang harus diisi pada halaman *login* untuk dapat melanjutkan ke halaman utama. Hasil realisasi pada halaman *register* digambarkan pada Gambar 4a.



Gambar 4. Halaman (a) *Register* (b) *Utama*

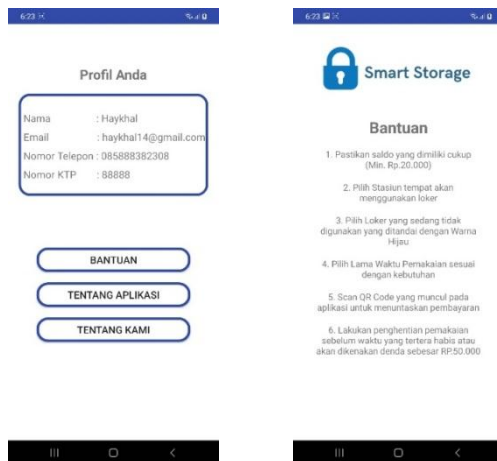
Halaman utama merupakan halaman yang muncul setelah melakukan proses *login*, pada halaman ini pengguna akan mendapatkan informasi berupa saldo yang dimiliki, serta kata sambutan yang akan menampilkan nama yang telah tercatat pada halaman registrasi serta tombol untuk *sign out* dan menuju halaman pilih loker untuk bisa menggunakan layanan dari alat ini. Hasil Realisasi dari halaman utama dapat dilihat pada Gambar 4b.

Halaman isi saldo merupakan halaman di mana pengguna mengisi saldo untuk dapat menggunakan layanan loker *keyless*. Pada halaman ini pengguna mengisi saldo sesuai yang dibutuhkan dengan minimal pengisian Rp.20.000 dan hasil dari penambahan saldo akan tercatat pada *Firebase Realtime Database*. Hasil dari realisasi halaman ini saldo digambarkan pada Gambar 5.



Gambar 5. Halaman Isi Saldo

Halaman Profil merupakan halaman di mana informasi dari pengguna yang sudah dimasukkan pada halaman registrasi akan ditampilkan. Selain berisi informasi pengguna, dari halaman profil juga dapat menuju halaman lain yang berisikan informasi-informasi mengenai aplikasi mulai dari bantuan cara pakai aplikasi, tentang aplikasi, dan tentang pengembangan aplikasi. Untuk realisasi dari halaman profil, halaman bantuan, halaman tentang aplikasi, dan halaman tentang kami digambarkan pada Gambar 6.



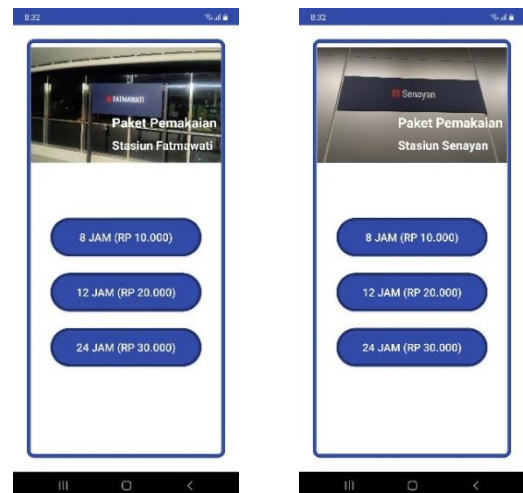
Gambar 6. Halaman Profil

Sebelum dapat menggunakan layanan dari loker *keyless*, pengguna harus melakukan proses pemesanan loker dengan memilih loker pada stasiun yang diinginkan oleh pengguna, ketika menekan tombol pilih loker pada halaman utama maka pertama akan muncul halaman pilih stasiun, setelah memilih stasiun tempat loker kemudian pengguna akan ditunjukkan ke halaman pilih loker berdasarkan stasiun yang sudah dipilih. Pada halaman pilih loker terdapat dua pilihan loker dengan

status pemakaian loker tersebut, loker dengan warna merah menandakan loker tersebut sedang digunakan sehingga hanya loker dengan warna hijau yang dapat digunakan. Hasil realisasi dari halaman pilih stasiun dan loker digambarkan pada Gambar 7.



Gambar 7. Halaman (a) Pilih Stasiun (b) Pilih Loker Stasiun Senayan



(a) Stasiun Senayan

(b) Stasiun Fatmawati

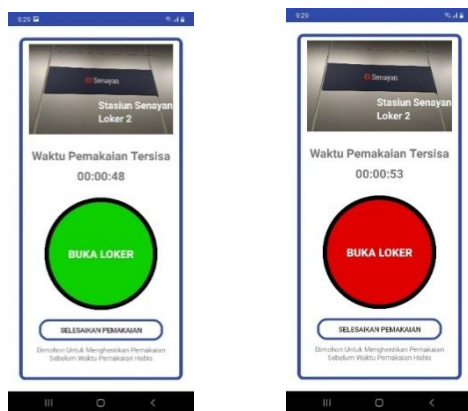
Gambar 8. Halaman Lama Waktu Pemakaian

Setelah memilih loker berdasarkan stasiun tempat loker tersebut berada, maka selanjutnya adalah memilih lama waktu pemakaian penggunaan loker. Setelah memilih lama waktu pemakaian, maka pengguna akan diarahkan ke halaman *QR code* untuk melakukan proses pembayaran. Hasil realisasi dari halaman pilih lama waktu pemakaian terbagi dua berdasarkan stasiun yang telah dipilih sebelumnya pada halaman pilih stasiun dan digambarkan pada Gambar 8.



Gambar 9. Halaman QR code Generator

Halaman *QR code Generator* ini bertujuan sebagai penyelesaian proses pembayaran untuk pelanggan sebelum dapat menggunakan layanan dari loker *keyless*. Halaman *QR code Generator* terbagi tiga berdasarkan paket waktu pemakaian yang ada. Realisasi dari halaman *QR code Generator* digambarkan pada Gambar 9.



Gambar 10. Halaman (a) Ketika Loker Terbuka (b) Ketika Loker Terkunci

Setelah selesai melakukan proses pembayaran pada halaman *QR code*, maka pengguna akan diarahkan pada halaman buka loker. Halaman buka loker dibuat berdasarkan loker yang ada yang berarti terdapat empat halaman buka loker. Pada halaman ini pengguna dapat membuka loker sesuai yang telah dipesan pada halaman pilih loker dan dapat menghentikan layanan loker *keyless* serta terdapat *timer* yang menunjukkan sisa waktu pemakaian. Hasil realisasi pada halaman buka loker dapat dilihat pada Gambar 10.

Halaman denda merupakan halaman yang muncul ketika waktu pada *timer* di halaman buka loker sudah

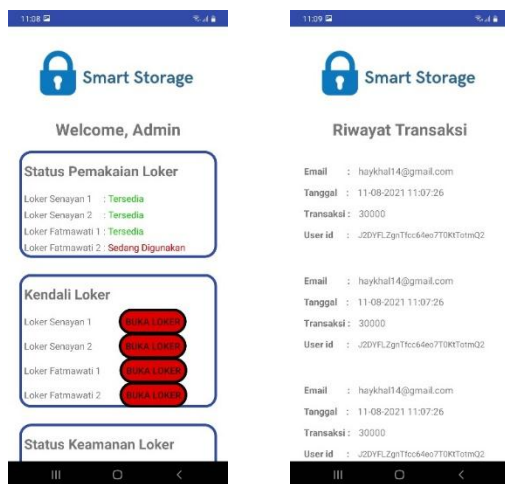
habis dan muncul notifikasi untuk pengguna membayar denda karena telah melewati batas waktu pemakaian. Disini pengguna diharuskan membayar denda dengan menunjukkan *QR code* yang muncul pada halaman ini lalu setelah proses pembayaran berhasil maka loker akan terbuka sekali dengan tujuan supaya pengguna dapat mengambil barang yang tertinggal didalam loker. Hasil realisasi dari halaman denda terdapat pada Gambar 11.



Gambar 11. Halaman Denda

Halaman utama admin merupakan halaman yang berisi kondisi ketersediaan loker, status keamanan loker sebagai bentuk pemantauan oleh admin terhadap layanan aplikasi *Smart Storage*. Selain pemantauan, admin juga dapat membuka loker secara langsung tanpa perlu melakukan pemesanan seperti *user* apabila terdapat kondisi yang membutuhkan admin untuk membuka loker. Realisasi dari halaman utama admin terdapat pada Gambar 12a.

Halaman riwayat pembayaran adalah halaman di mana admin dapat melihat data pembayaran *user* yang dicatat pada halaman *QR code Generator* yang merupakan halaman di mana *user* akan melakukan pembayaran. Data yang ditampilkan pada halman riwayat pemakaian berupa data *email*, tanggal transaksi, *user id*, dan nominal transaksi yang dilakukan. Realisasi dari halaman riwayat pembayaran terdapat pada Gambar 12b.



Gambar 12. Halaman (a) Admin (b) Riwayat Transaksi

Pengujian aspek merupakan pengujian yang mengacu pada standar ISO 25010.

Pengujian Aspek *Functional Suitability*

Pengujian aspek *functional suitability* dilakukan untuk mengetahui apakah sistem sudah mampu menyediakan fungsi yang sesuai dengan kebutuhan. Pengujian aspek *functional suitability* dilakukan dengan menggunakan metode *blackbox testing* [9] yaitu pengujian terhadap fungsi-fungsi atau proses yang terjadi pada aplikasi *Smart Storage*. Ada empat puluh lima menu/halaman/fitur mulai dari halaman login, halaman splashscreen, register, halaman utama, dll yang diuji. Hasil pengujian *functional suitability* ditunjukkan pada Tabel 3, dengan memeriksa fitur-fitur yang ada pada aplikasi apakah dapat berjalan dengan baik atau tidak.

Tabel 3. Hasil Ketercapaian Pengujian *Functional Suitability*

Ketercapaian	
Ya	Tidak
45	0

$$\text{Presentase kelayakan (\%)} = \frac{(\text{Skor yang di dapat})}{(\text{Skor maksimal})} \times 100\%$$

$$= \frac{45}{45} \times 100 \%$$

$$= 100 \%$$

Pengujian Aspek *Compatibility*

Pengujian aspek *compatibility* dilakukan untuk mengetahui apakah sistem aplikasi *Smart Storage* dapat bertukar informasi dengan sistem, produk atau komponen yang lain serta menjalankan fungsi lain yang diperlukan secara bersamaan ketika berbagi perangkat

keras dan perangkat lunak yang sama. Hasil pengujian aspek *compatibility* ditunjukkan pada tabel 4.

Tabel 4. Hasil Pengujian Aspek *Compatibility*

Aplikasi Yang Dijalankan	Test Case	Ketercapaian	
		Berhasil	Gagal
Smart Storage	Whatsapp	01	✓
	Samsung Browser	02	✓
	Telegram	03	✓
	Line	04	✓
	Youtube	05	✓
	Gojek	06	✓
	Camera	07	✓
	Twitter	08	✓
	Spotify	09	✓
	Facebook	10	✓

Pengujian Aspek *Portability*

Pengujian aspek *portability* dilakukan untuk mengetahui apakah sebuah sistem, produk atau komponen aplikasi dapat digunakan pada perangkat keras dan perangkat lunak yang berbeda-beda. Hasil pengujian aspek *portability* ditunjukkan pada Tabel 5.

Tabel 5. Hasil Ketercapaian Pengujian Aspek *Portability* pada Berbagai Versi Android

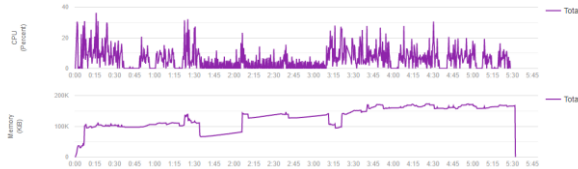
Test Case	Nama Perangkat	API Level	Ketercapaian	
			Berhasil	Gagal
01	ALE-L23	21		✓
02	Nexus 4	22		✓
03	Moto G Play	23	✓	
04	LG G6 VS988	24	✓	
05	Pixel	25	✓	
06	LG-H932	26	✓	
07	Redmi 6A	27	✓	
08	Oppo A5s	27	✓	
09	Nokia 9	28	✓	
10	SM-J701F	28	✓	
11	SM-105FN	29	✓	
12	Pixel 5e	30	✓	
13	SM-A507FN	30	✓	

Pengujian Aspek *Performance Efficiency*

Pengujian aspek *performance efficiency* dilakukan untuk mengetahui kinerja relatif terhadap sumber daya yang digunakan dalam menjalankan aplikasi *Smart Storage*. Pada pengujian aspek *performance efficiency* disini menguji karakteristik *Resource utilization* yaitu sejauh mana jumlah berdasarkan jenis sumber daya yang digunakan oleh sistem atau produk dapat

memenuhi persyaratan ketika menjalankan fungsi dari sistem atau produk tersebut.

Hasil pengujian aspek *performance efficiency* dengan menggunakan *Firestore Test Lab Robo Test* [11]. Berdasarkan pada Gambar 13 menunjukkan bahwa pengujian *performance* dengan *robo test* pada perangkat Pixel untuk menjalankan aplikasi *Smart Storage* berhasil dan dilakukan selama 5 menit 45 detik.



Gambar 13. Hasil Pengujian Performance pada Perangkat Pixel

Pengujian Aspek Usability

Pengujian aspek *usability* merupakan pengujian yang bertujuan untuk menentukan sejauh mana produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi dan kepuasan dalam konteks penggunaan tertentu. Pengujian aspek *usability* dilakukan dengan membuat kuesioner dengan memerhatikan aspek efektivitas, efisiensi, dan kepuasan pengguna dalam menggunakan aplikasi *Smart Storage*. Pada pengujian aplikasi *Smart Storage* memiliki target responden masyarakat umum dengan catatan pernah menggunakan layanan MRT. Pembuatan kuesioner untuk pengujian aspek *usability* menggunakan layanan Google Form. Hasil pengujian aspek *usability* ditunjukkan pada Tabel 6.

Tabel 6. Hasil Pengujian Aspek Usability

No. Kuesioner	Total	Presentase Kelayakan
1.	41	82 %
2.	39	78 %
3.	41	82 %
4.	44	88 %
5.	44	88 %
6.	38	76 %
7.	45	90 %
8.	43	86 %
9.	46	92 %
10.	44	88 %

4. Kesimpulan

Berdasarkan pengujian aspek *compability* menunjukkan bahwa aplikasi *Smart Storage* dapat bertukar informasi dengan sistem, produk atau komponen yang lain serta menjalankan fungsi lain yang diperlukan secara bersamaan ketika berbagai perangkat keras dan

perangkat lunak yang sama. Pengujian aspek *performance efficiency* dilakukan pada dua perangkat yaitu Pixel dan Samsung Galaxy S20 dengan menggunakan *tools Firestore Test Lab Robo Test*. Pada pengujian kinerja CPU perangkat Pixel didapatkan hasil 9,973% dan perangkat Samsung Galaxy S20 didapatkan hasil 1,58% yang di mana hasil ini masih berada pada batas aman dari batas aman yang mengacu pada *Little Eye* sebesar 15%. Pengujian aspek *usability* menunjukkan aplikasi *Smart Storage* sangat layak digunakan oleh pengguna dari sisi efektivitas, efisiensi dan kepuasan dalam penggunaan aplikasi.

Daftar Acuan

- [1] <https://www.bps.go.id> (diakses Agustus 2021)
- [2] F. Muhammad., Strategi Komunikasi Pemasaran PT. Mass Rapid Transit Jakarta, Jurnal Kommas (2020)
- [3] F. Cheng, Build Mobile Apps with Ionic 2 and Firebase Hybrid Mobile App Development, Apress, (2017)
- [4] N. Rahmalia, Praktis Berbagi Informasi dengan QR code, Sudahkah Kamu Mencobanya?, 2021.
- [5] A. Akbar, Firebase Adalah: Cara Kerja, Sejarah, Manfaat, Kelebihan dan Kekurangannya, 2021.
- [6] Firestore Test Lab (diakses Agustus 2021)
- [7] Guntoro, Apa Itu Android Studio. Badoy Studio, 2020.
- [8] N. Safaat, Pemrograman Aplikasi Mobile Berbasis Android. Informatika Bandung, 2018.
- [9] Rachmawati, A. L. Nugraha, A. Awaluddin, Desain Aplikasi Mobile Informasi Pemetaan Jalur Batik Solo Trans Berbasis Android Menggunakan Location Based Service, Jurnal Geodesi UNDIP Vol. 6, Issue 2 (2017)
- [10] Sanjaya, Ningsih, Sistem Pendukung Keputusan Penentuan Calon Penerima Beasiswa Menggunakan Metode Fuzzy Database Model Tahani, Jurnal SIMETRIS, Vol 7 No 2 November (2016)
- [11] Soleh, Lukman, Firestore test lab untuk android virtual tes lab yang disediakan firebase, 2021