

Perancangan *Smart Locker* dengan Implementasi Sistem IoT dan Aplikasi *Mobile Android*

Fitri Elvira Ananda¹, Dandun Widhiantoro², Salsabilla Haurameuthia³,
Muhammad Iqbal Tejasumirat⁴

^{1,2,3,4} Politeknik Negeri Jakarta, Jurusan Teknik Elektro, Prodi Broadband Multimedia
Jl Prof.DR.GA Siwabessy, Kampus Baru UI Depok, Beji, Depok, Jawa Barat, 16425, Indonesia
fitri.ananda@elektro.pnj.ac.id¹

Intisari — Locker merupakan sebuah fasilitas penyimpanan yang dapat dijumpai pada sejumlah sarana umum seperti kolam renang umum, perpustakaan, dan sarana olahraga seperti gymnasium. Kebanyakan locker yang ada masih menggunakan cara penguncian konvensional yaitu kunci fisik. Hal ini berdampak negatif dari segi efektivitas, efisiensi, dan keamanan. Oleh karena itu dibuat sistem berbasis QR-Code yang terintegrasi dengan aplikasi android berfungsi untuk monitoring ketersediaan, melakukan peminjaman locker, dan membuka serta mengunci locker melalui aplikasi android. Alat smart locker dibuat menggunakan mikrokontroler ESP32 sebagai pengatur logika dan jaringan yang digunakan ketika menerima data yang dikirimkan dari perangkat android. ketika data diterima, ESP32 akan merespon dengan mengaktifkan relay yang mengatur magnetic lock yang digunakan sebagai pengganti kunci konvensional, sementara aplikasi android berfungsi untuk monitoring ketersediaan, melakukan peminjaman locker, dan membuka serta mengunci locker melalui aplikasi android. Pengujian aplikasi yang dilakukan untuk aplikasi ini merupakan pengujian yang mengacu pada standar ISO 25010 dengan aspek functional suitability, usability, dan compatibility. Hasil pengujian aspek functional suitability dan compatibility mendapatkan persentase kelayakan 100%. Kemudian untuk usability menunjukkan bahwa dari 10 poin kuesioner yang disebar kepada responden mendapatkan hasil persentase kelayakan sebesar 78.98% yang termasuk kategori acceptable, hal ini menunjukkan bahwa aplikasi android yang dirancang memiliki kualitas yang baik. Pada pengujian alat dilakukan penekanan tombol pada aplikasi android ditekan secara bersamaan melalui dua perangkat yang berbeda terdapat selisih waktu untuk membuka locker dengan rata-rata waktu 0.874 detik.

Kata kunci — Aplikasi Android, ESP32, ISO 25010, QR-Code, *Smart Locker*

Abstract — A locker is a storage facility that can be seen at several public facilities such as public swimming pools, a library, and sports facilities like a gymnasium. Most lockers at public facilities still use conventional locking techniques. This has a negative impact in terms of effectiveness, efficiency, and security. Therefore, a QR-Code-based system that is integrated with the android application has been created to function for monitoring availability, borrowing lockers, and opening and locking lockers through the android application. Smart locker system using ESP32 as microcontroller and network that used to receive data sent from android application. When data is received, ESP32 will respond by activating the relay which controls the magnetic lock that replaced the conventional lock, while the android application is created for monitoring locker availability, borrowing lockers, and opening and locking lockers through the android application. This test is carried out for this application, which refers to the ISO 25010 standard with aspects of functional suitability, usability, and compatibility. The results of testing aspects of functional suitability and compatibility get 100% feasibility percentage results. Then for usability, it shows that from the 10-point questionnaire distributed to respondents, the percentage of feasibility results is 78.98% which is included in the acceptable category, this shows that the mobile application has good quality. The system test was done by pressing the button in the android application at the same time with two different devices and having a time difference while opening the lockers with a time average of 0.874 seconds.

Keywords — Android Application, ESP32, ISO 25010, QR-Code, Smart Locker.

I. PENDAHULUAN

Pada setiap tempat umum yang terdapat locker, locker merupakan salah satu fasilitas penunjang yang digunakan banyak orang untuk menyimpan barang, dokumen, dan lain sebagainya. Locker yang masih menggunakan cara konvensional dengan kunci fisik masih sering dijumpai di tempat umum, tetapi dari segi kenyamanan dan keamanan masih minim. Dengan menggunakan kunci fisik, pengunjung perlu menjaga kunci tersebut supaya tidak hilang, terdapat banyak cara untuk menduplikat kunci fisik, dan jika kunci fisik terbuat dari logam, erosi logam dapat terjadi [1].

Telah ditemukan beberapa solusi untuk membuat smart locker diantaranya yaitu aplikasi hanya dapat membuka dan menutup locker melalui *bluetooth* berupa notifikasi yang dikirimkan melalui smartphone, menggunakan Arduino Uno (R3) sebagai mikrokontroler. Namun locker belum dilengkapi disisi aplikasi-nya untuk mengetahui locker yang sedang tersedia ataupun tidak [2]. Selain itu, juga terdapat pengaplikasiannya akses locker memanfaatkan ponsel pada pengguna locker dan menggunakan mikrokontroler Arduino/Raspberry pi [3].

Kemudian terdapat penelitian yang memanfaatkan Node MCU ESP32 dan MySQL sebagai database. Sistem ini menggunakan fitur LINE Chat Bot untuk membuka locker. Pada aplikasi tersebut sudah dilengkapi untuk membuka locker, mengetahui locker yang sedang tersedia, mengembalikan locker, dan fitur melaporkan masalah yang terjadi pada locker [4]. *Smart Locker* adalah jenis layanan pintar yang menggunakan teknologi informasi untuk mendukung layanan logistik dalam kehidupan kita sehari-hari [5]

Untuk mengatasi permasalahan tersebut, penelitian ini membuat rancangan sistem yang terintegrasi dengan aplikasi android berfungsi untuk memantau ketersediaan, melakukan peminjaman locker, dan membuka hingga mengunci locker, serta melakukan absensi kehadiran pengujung melalui aplikasi android yaitu Sistem Locker Keyless. Sistem ini menggunakan ESP32

sebagai mikrokontroller-nya. ESP32 merupakan sistem-on-chip berkemampuan WiFi dan Bluetooth yang sangat kuat dengan banyak GPIO dan papan pengembangan yang menunjukkan kekuatannya dalam merancang modul IoT yang sangat mudah diakses [6]. Sistem ini menggunakan framework Laravel sebagai API. Terdapat salah satu penelitian yang menggunakan framework Laravel yaitu penelitian terkait sistem pembelajaran cloud [7]. Kelebihan dari Laravel ialah kode yang simple dan sederhana, dokumentasi yang lengkap dan mudah digunakan [8]. Menurut Supardi, Laravel dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, serta untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas, dan menghemat waktu [9]. Pengguna dapat mengakses loker dengan memindai QR Code yang terletak pada pintu loker. QR Code bekerja dengan menampilkan data dari database yang di generate lalu menampilkan sebuah gambar QR Code [10].

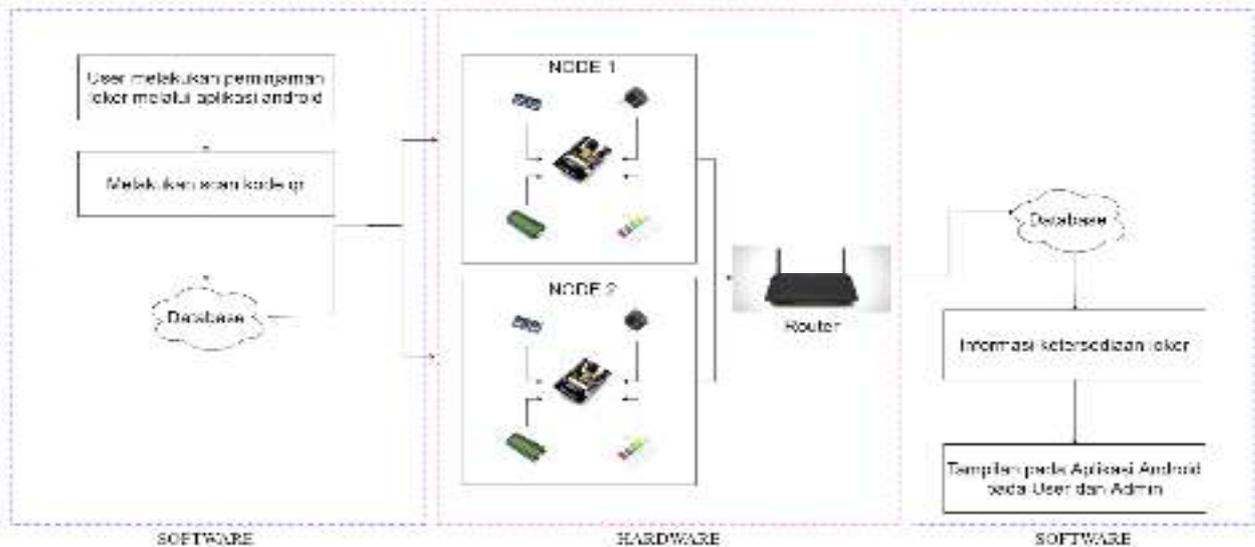
Sistem akan terhubung dengan aplikasi yang dapat diakses oleh pengunjung. Sistem ini juga dilengkapi dengan kunci yang hanya bisa diakses melalui aplikasi masing-masing pengunjung sehingga hanya pengunjung tersebutlah yang bisa mengakses buka ataupun tutup locker tersebut. Namun, sebelum melakukan peminjaman locker, pengunjung harus mengisi daftar kehadiran terlebih dahulu yang fiturnya sudah ada dan terdaftar absensi ketika pengunjung melakukan login. Dengan pembuatan locker ini, tentunya akan sangat berguna bagi pengunjung karena memudahkan pengunjung untuk efisiensi dan efektivitas dalam melakukan peminjaman locker.

II. METODE PENELITIAN

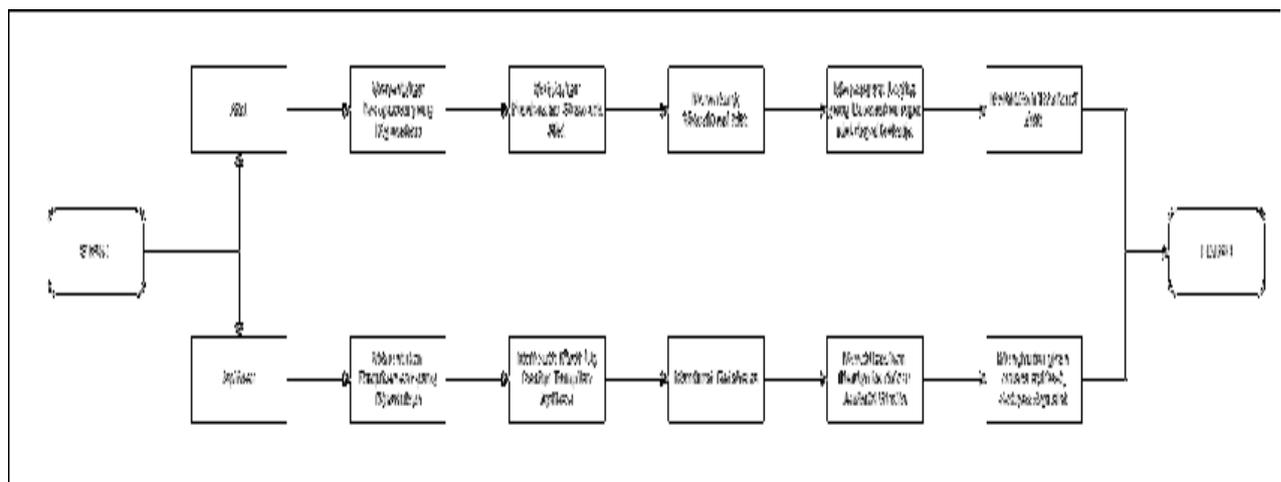
Pada Gambar 1 memperlihatkan blok diagram sistem yang terdiri dari *hardware* dan *software* ketika terintegrasi satu sama lain untuk melakukan peminjaman hingga membuka/menutup locker. Pada bagian *hardware* berupa alat pengunci loker

otomatis yang menggunakan komponen ESP32 sebagai *microcontroller*, relay, dan *magnetic lock* yang dapat mendeteksi locker sedang digunakan atau tidak. Pada bagian *software*, aplikasi dirancang dengan android studio dan database MySQL. Aplikasi dapat melakukan presensi dan daftar akun,

menampilkan informasi ketersediaan locker, meminjam locker, membuka dan menutup locker. Pada setiap locker diberikan QR-Code, pembacaan QR-Code tersebut dapat mengakses hardware, lalu data dikirim ke MySQL melalui *Framework* Laravel.



Gbr.1 Diagram Blok Sistem



Gbr. 2 Langkah Pembuatan Sistem Smart Locker

Sistem *Smart Locker* akan bekerja seperti penjelasan berikut:

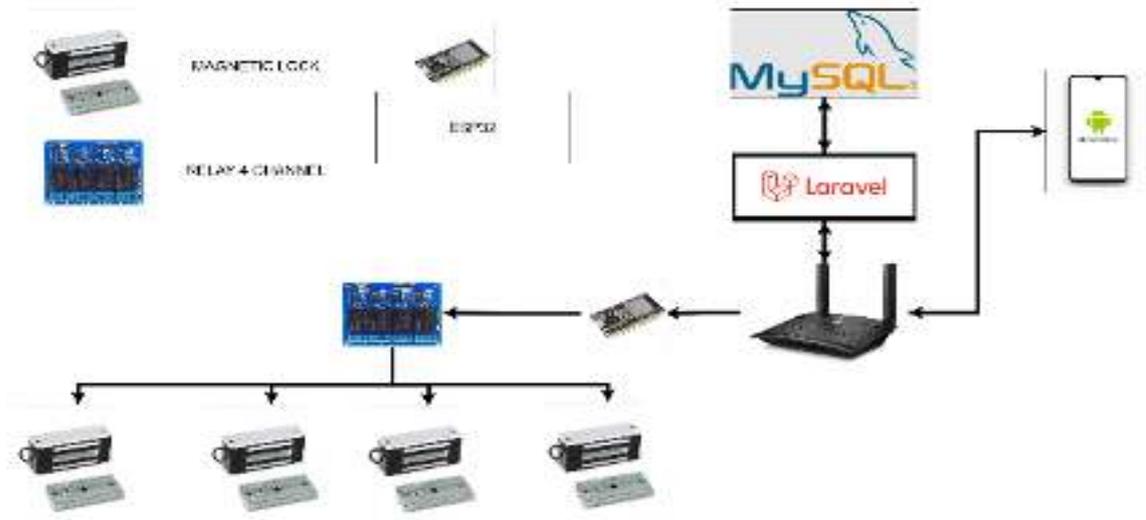
- 1) Membuka aplikasi android yang telah terhubung dengan sistem.
- 2) Setelah itu melakukan registrasi akun yang nanti akan digunakan untuk login aplikasi.
- 3) Lalu login pada aplikasi menggunakan akun yang telah dibuat.
- 4) Jika berhasil masuk. Pilih locker yang ingin digunakan dan dapat dipinjam.
- 5) Lakukan proses *scanning* kode QR yang terdapat pada locker yang dipilih.

- 6) Setelah proses *scan* berhasil, locker dapat diakses untuk membuka, menutup, dan menyelesaikan peminjaman.

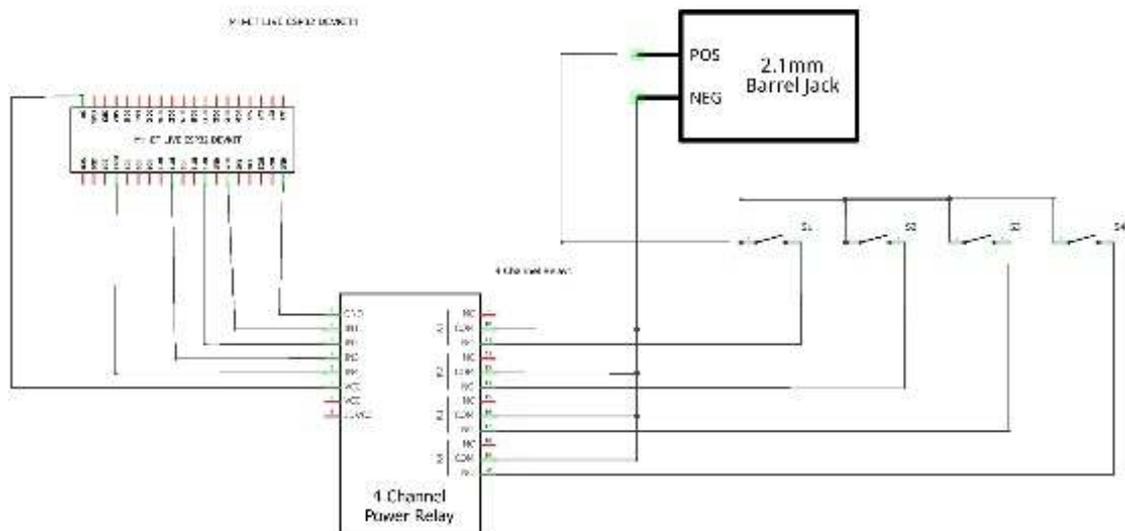
Pada Gambar 2 dapat dilihat langkah pembuatan Sistem *Smart Locker* dari sisi hardware dimulai dari penentuan komponen, membuat desain skematik dan visualisasi perangkat, melakukan pemrograman pada *hardware* serta merealisasikannya. Sedangkan pada sisi perancangan *software* aplikasi dimulai dengan menentukan *requirement* sistem, mendesain mock-up aplikasi, merancang database, merealisasikan

desain dengan tools Android Studio IDE Versi 4.2.1 serta pengkoneksikan antara *hardware* dengan *software*. Gambar 3 menunjukkan visualisasi sistem *smart locker* yang dibuat, pada penelitian ini *smart locker* yang telah dirancang akan diimplementasikan untuk 1 (satu) lemari yang terdiri dari 4 (empat) locker. Gambar 4

menunjukkan skematik hardware yang terdiri dari ESP32 terhubung dengan relay 4 channel yang menggunakan pin GPIO 21, 19, 17, dan 15, sedangkan pin input relay terhubung dengan tegangan 5 Volt pada ESP32. Lalu catu daya 12V 10A digunakan sebagai sumber tenaga dari *magnetic lock*.



Gbr. 3 Visualisasi Sistem



Gbr. 4 Skematik Hardware

III. HASIL DAN PEMBAHASAN

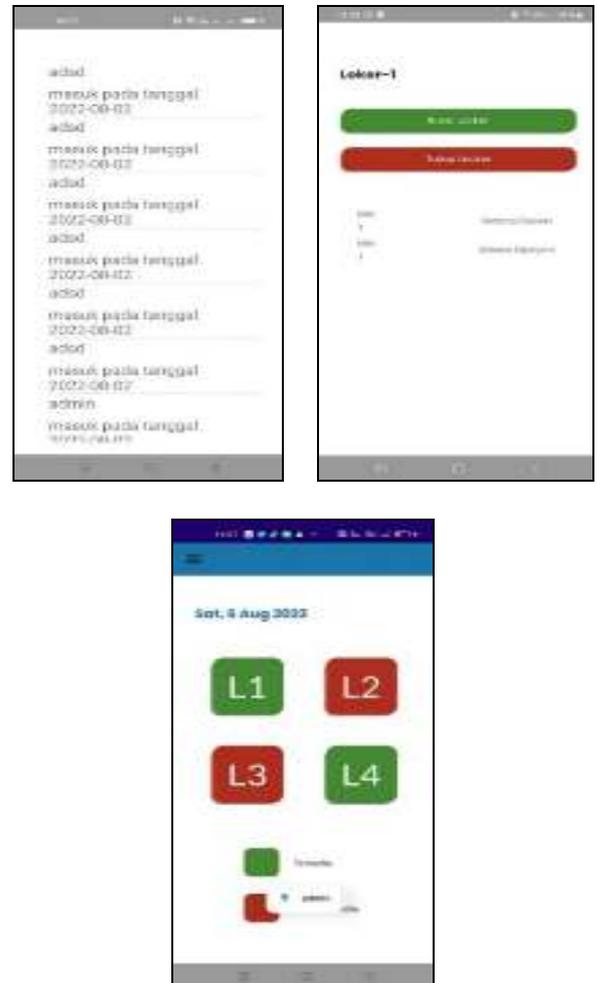
Setelah melakukan proses pengumpulan *requirement* dan perancangan aplikasi, Langkah selanjutnya ialah proses realisasi aplikasi pada Android Studio. Aplikasi android yang dibuat diberi nama “Booka”. Berikut merupakan hasil dari realisasi aplikasi pada sisi Pemustaka (*user*), dan Admin.



Gbr. 6 Aplikasi *User*

Halaman Utama pada Aplikasi Booka berfungsi untuk melihat loker mana yang sedang tersedia. Terdapat Indikator warna merah yang berarti loker sedang digunakan oleh pemustaka lain dan warna hijau yang berarti loker dapat digunakan. Halaman Pinjam Loker dapat diakses ketika pemustaka memilih loker yang tersedia. Pada halaman ini, pemustaka harus memindai QR-Code yang terdapat pada pintu loker. Jika berhasil, akan menampilkan halaman buka loker.

Perbedaan antara halaman admin dengan halaman pemustaka ialah, jika loker berwarna merah, admin tetap dapat masuk ke halaman tersebut. Pada halaman admin, tidak dikirimkan parameter nama_loker dan id.



Gbr.7 Aplikasi Admin

Gambar 8 menampilkan hasil dari realisasi alat yang telah dibuat, terdapat perbedaan penempatan komponen seperti ESP32, pada visualisasi alat diletakan dibagian atas locker namun pada realisasinya ditempatkan di sisi belakang locker.

A. Pengujian Suitability

Pengujian aspek functional suitability menggunakan metode *blackbox testing* yaitu pengujian terhadap fungsi-fungsi atau proses yang terjadi pada aplikasi Booka. Pengujian dilakukan untuk memeriksa fitur-fitur yang terdapat pada aplikasi Booka sudah dapat berjalan dengan baik atau tidak dan apakah sudah sesuai dengan kebutuhan. Peralatan yang digunakan pada pengujian functional suitability ini adalah:

- 1) *Smartphone* Android
- 2) *Booka.apk*



Gbr. 8 Realisasi Hardware

Terdapat 22 test case pada pengujian ini meliputi fungsi pada halaman untuk pemustaka dan admin. Dari hasil pengujian yang telah dilakukan, maka akan didapat data hasil pengujian. Setiap jawaban akan bernilai 1 untuk setiap kategori ketercapaian yang ada. Jika tercapai maka poin ketercapaian tercapai akan bertambah satu dan begitu pula untuk ketercapaian tidak tercapai. Hasil dari pengujian aspek functional suitability dituliskan pada Tabel 1 berikut.

Tabel 1 Hasil Ketercapaian Functional Suitability

Ketercapaian	
Ya	Tidak
22	0

Dari hasil pengujian functional suitability diatas, Langkah berikutnya adalah menghitung persentase kelayakan dengan melakukan perhitungan sesuai dengan rumus sebagai berikut:

$$\text{Persentase kelayakan (\%)} = \frac{22}{22} \times 100\% = 100\%$$

Berdasarkan hasil perhitungan persentase kelayakan didapatkan hasil 100%. Merujuk pada Tabel 1, hasil perhitungan ini mendapat kategori sangat layak. Hasil pengujian ini menunjukkan bahwa semua halaman dan fitur-fitur yang berada pada aplikasi *Booka* dapat berfungsi dengan baik.

B. Pengujian Usability

Pengujian pada karakteristik usability dilakukan dengan membuat kuesioner dengan memperhatikan aspek efektivitas, efisiensi, dan kepuasan pengguna dalam

menggunakan aplikasi *Booka*. Pembuatan kuesioner untuk pengujian usability menggunakan layanan Google Form dengan jumlah responden sebanyak 23 responden.

Langkah-langkah yang akan dilakukan untuk mencapai pengujian karakteristik usability adalah menyiapkan pertanyaan untuk kuesioner kepada target responden. Lalu menyebarkan kuesioner kepada target responden. Setelah itu lakukan pengecekan hasil pengisian kuesioner yang telah diisi oleh para responden. Pertanyaan pada kuesioner sesuai dengan metode SUS (*System Usability Scale*). Penilaian ini menggunakan skala *Likert* dengan memberikan lima pilihan jawaban dengan masing-masing nilai yang ditunjukkan pada Tabel 2.

Tabel 2 Nilai Skala Usability

Jawaban	Nilai
Sangat Tidak Setuju (STS)	1
Tidak Setuju (TS)	2
Ragu-Ragu (R)	3
Setuju (S)	4
Sangat Setuju (SS)	5

Langkah selanjutnya adalah melakukan analisis data pengujian sesuai dengan metode SUS (*System Usability Scale*).

Tabel 3 Hasil Pengujian Usability

Jumlah	1737.5
Nilai rata-rata SUS (Jumlah SUS/Jumlah Responden)	78.98

Pada Tabel 3 diatas, merupakan hasil perhitungan nilai dari responden. Sebuah sistem dapat dikatakan memiliki nilai usability yang baik jika memiliki nilai SUS diatas 71, masuk ke kategori *acceptable*. Dari tabel diatas, diperoleh nilai rata-rata SUS sebesar 78.98. Dengan kata lain, tingkat usability dari aplikasi android *Booka* dapat diterima dan memiliki aspek usability yang baik.

C. Pengujian Compatibility

Pengujian pada *compatibility* dilakukan untuk sub-karakteristik *co-existence* yang dilakukan dengan metode observasi yaitu dengan cara menjalankan Aplikasi *Booka* bersamaan dengan aplikasi lainnya. Tujuan

dilakukan nya adalah untuk mengetahui aplikasi Booka dapat berjalan dengan baik atau tidak jika dijalankan bersama dengan aplikasi lain pada waktu yang sama

Berdasarkan hasil pengujian *compatibility* yang dilakukan, terlihat bahwa pengujian pengujian untuk karakteristik *co-existence* dengan menjalankan Aplikasi Booka secara bersamaan dengan aplikasi lainnya dalam satu perangkat. Dari 10 aplikasi yang diuji didapatkan kesemua aplikasi berhasil dijalankan secara bersamaan dengan Aplikasi Booka tanpa mengalami *error*, *crash* maupun *force close*. Sehingga, persentase kelayakan bias dikatakan 100%. Hasil tersebut menunjukkan bahwa Aplikasi Booka dapat berjalan dengan baik dan lancar apabila dijalankan secara bersamaan dengan aplikasi lainnya

D. Pengujian Penekanan Tombol dari Dua Perangkat

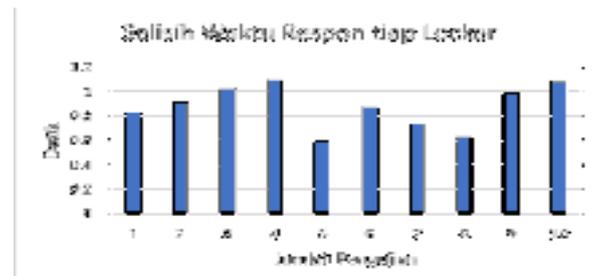
Pengujian ini dilakukan untuk mengetahui perbandingan waktu respon ESP32 dan relay ketika tombol pada aplikasi ditekan dari perangkat yang berbeda, pengujian ini menggunakan 2 perangkat android untuk mendapatkan data. Tabel 4 berikut menunjukkan hasil pengujian yang telah dilakukan.

Tabel 4 Pengujian Respon Alat Ketika Tombol Ditekan Dalam Waktu Bersamaan

No. Pengujian	Selisih Waktu Respon antar Locker (detik)
1	0.82
2	0.91
3	1.02
4	1.10
5	0.59
6	0.87
7	0.74
8	0.63
9	0.98
10	1.08
Rata-rata	0.874

Pada tabel 4 menampilkan data yang didapatkan, ketika tombol pada aplikasi ditekan terdapat *delay* untuk membuka maupun locker dengan nilai yang bervariasi. Seperti pada pengujian pertama ketika di uji untuk locker 1 dan locker 2, terdapat selisih

waktu untuk membuka kedua locker tersebut sebesar 0.8 detik. Dari keseluruhan data pengujian yang dilakukan, didapatkan nilai rata-rata selisih waktu ketika tombol ditekan secara bersamaan untuk membuka locker yang berbeda adalah 0.874 detik. Grafik dari pengujian tersebut dapat dilihat pada Gambar 9



Gbr. 9 Selisih Waktu Respon Setiap Locker Saat Tombol Ditekan Secara Bersamaan

IV PENUTUP

Dari pembahasan, hasil pengujian serta analisa data pada penelitian ini dapat disimpulkan bahwa: berdasarkan pengujian yang mengacu pada standar ISO 25010 untuk aplikasi android pada aspek *functional suitability* dan *compatibility* diperoleh persentase kelayakan sebesar 100%, sehingga dapat dikategorikan sangat layak. Hal ini menunjukkan bahwa semua halaman dan fitur-fitur yang terdapat pada aplikasi Booka dapat berjalan dan berfungsi dengan baik. Pada aspek *usability* didapatkan persentase sebesar 78.98%, sesuai dengan Standar Kelayakan SUS, aplikasi Booka dikategorikan baik. Dari sisi hardware, system berfungsi dengan baik dan sebagaimana mestinya. Pengujian respon hardware dilakukan dengan saat tombol pada aplikasi ditekan secara bersamaan melalui dua perangkat yang berbeda terdapat selisih waktu untuk membuka locker dengan rata-rata waktu 0.874 detik. Pengembangan penelitian dapat dilakukan dengan menambahkan scanner QR-Code secara otomatis.

UCAPAN TERIMA KASIH

Penelitian ini dibiayai oleh dana DIPA Politeknik Negeri Jakarta tahun 2022 dengan skema Penelitian Dosen Pemula.

REFERENSI

- [1] Churi, A., Bhat, A., Mohite, R., & Churi, P. (2016, December). E-zip: An electronic lock for secured system. In 2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT) (pp. 45-49). IEEE.
- [2] Alqahtani, H. F., Albuainain, J. A., Almutiri, B. G., Alansari, S. K., AL-awwad, G. B., Alqahtani, N. N., ... & Tabeidi, R. A. (2020, March). Automated smart locker for college. In 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS) (pp. 1-6). IEEE
- [3] Parab, P., Kulkarni, M., & Shinde, V. (2018). Smart Locker Management System Using IoT. system, 8, 10.
- [4] Sa-ngiampak, J., Hirankanokkul, C., Sunthornyotin, Y., Mingmongkolmitr, J., Thunprateep, S., Rojsrikul, N., ... & Pipattanasomporn, M. (2019, October). LockerSwarm: an IoT-based smart locker system with access sharing. In 2019 IEEE International Smart Cities Conference (ISC2) (pp. 587-592). IEEE.
- [5] Lian, J. W., Chen, C. T., Shen, L. F., & Chen, H. M. (2020). Understanding user acceptance of blockchain-based smart locker. The Electronic Library.
- [6] Arief Budijanto, S. T., & Winardi, S. (2021). Interfacing ESP 32. Scopindo Media Pustaka.
- [7] Nugroho, K., & Murdowo, S. (2019, September). Mobile cloud learning system using laravel framework and android studio web view. In 2019 International Seminar on Application for Technology of Information and Communication (iSemantic) (pp. 141-144). IEEE.
- [8] Yudhanto, Y., Prasetyo H. A. (2019). Mudah menguasai Framework Laravel. PT Elex Media Komputindo.
- [9] Supardi, Y. (2019). Semua Bisa Menjadi Programmer Laravel Basic. Elex Media Komputindo.
- [10] Kurniawan, Y. I., Nurjaman, A. L., & Afuan, L. (2021). Sistem Presensi Karyawan Menggunakan Quick Response Code di CV. Jenderal Software. Jurnal Teknologi dan Informasi, 11(2), 168-182.