

Rancang Bangun Aplikasi Perizinan Santri Berbasis Android

Hafizh Hidayat¹, Amri^{2*}, Indrawati³

^{1,2,3} Jurusan Tekniknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe
Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA

¹hafizhhidayat1234@gmail.com

^{2*}amri@pnl.ac.id

³indrawati@pnl.ac.id

Abstrak— Sistem perizinan pesantren di Aceh masih banyak dikelola secara manual. Hal ini menyebabkan pengelolaan izin yang dilakukan menjadi kurang efektif, sehingga memungkinkan untuk dimanipulasi. Hal ini sering menjadi masalah di pesantren dalam hal perizinan. Karena permasalahan tersebut dibuatlah sebuah sistem informasi perizinan santri berbasis *android*. Sistem perizinan yang dibuat agar orang tua dan ustad dapat memantau perizinan santri dengan lebih baik dan berbasis *mobile*. Metode yang digunakan pada penelitian ini adalah metode perancangan *waterfall*. Pengujian sistem menggunakan metode *load testing*. Didapatkan hasil respons *web server* sistem jika diakses 40 pengguna adalah 1,09 detik, dan jika diakses oleh 140 pengguna respons *web server* adalah 1,57 detik.
Kata kunci— Perizinan, Pesantren, Android, Santri, Load testing.

Abstract— The Islamic boarding school licensing system in Aceh is still largely managed manually. This causes the management of permits to be less effective, making it possible for them to be manipulated. This often becomes a problem in Islamic boarding schools in terms of licensing. Because of these problems, an Android-based student licensing information system was created. The licensing system was created so that parents and ustad can better monitor student permits and is mobile-based. The method used in this research is the waterfall design method. System testing uses the load testing method. It was found that the system web server response if accessed by 40 users was 1.09 seconds, and if accessed by 140 users the web server response was 1.57 seconds.

Keywords— Licensing, Islamic Boarding School, Android, Santri, Load testing.

I. PENDAHULUAN

Perkembangan teknologi telah dimanfaatkan semua lembaga untuk mempermudah kegiatan-kegiatan yang dilakukan dalam lembaga tersebut. Pesantren juga tidak luput untuk memanfaatkan perkembangan teknologi tersebut. Aceh merupakan salah satu provinsi yang mempunyai pesantren terbanyak di Indonesia. Berdasarkan data dari kementerian agama, Aceh menempati posisi pertama pesantren terbanyak sejumlah 1177 pesantren. Santri yang lokasinya jauh dari menggunakan sistem asrama yang memungkinkannya untuk menerapkan nilai-nilai dan pandangan dunia yang dianutnya dalam kehidupan keseharian santri[1].

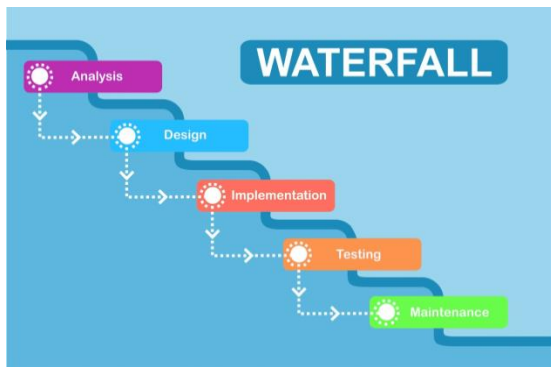
Akan tetapi santri yang melakukan mondok di pesantren, untuk dapat keluar dari pesantren tersebut harus mempunyai izin. Izin biasanya diajukan saat santri tersebut ingin meninggalkan pesantren untuk sementara waktu, misalnya santri tersebut sakit ataupun ada kegiatan lainnya. Pengelolaan data santri pesantren di Aceh masih banyak dilakukan secara manual, terutama perizinan yang masih menggunakan buku perizinan santri. Hal ini dapat menyebabkan perizinan tersebut dapat dimanipulasi oleh santri. Hal ini menjadi masalah serius tiap kali santri meminta izin terkadang orang tua dari santri tersebut tidak tau bahwa anaknya sudah meminta izin untuk pulang. Jumlah santri yang terlalu banyak pun menjadi masalah apabila tidak diiringi dengan sistem yang mumpuni. (Gatot Suherman, Zainal Arifin dan Anis Yusrotun).

Untuk menangani hal tersebut dibuatlah sebuah sistem pemantauan perizinan santri berbasis *mobile*. Pada sistem pemantauan perizinan ini data santri akan disimpan, dan dapat dipantau oleh ustad yang mengajar di kelas, dan juga orang tua dari santri tersebut melalui aplikasi dari pesantren tersebut. Izin akan diberikan apabila orang tua dan ustad dari santri tersebut telah mengkonfirmasi perizinan yang dikirimkan ke perangkat. Dalam hal ini pesantren dapat memantau perizinan dengan lebih aman, dan orang tua dari santri tersebut juga akan lebih tenang dapat mengetahui keadaan anaknya yang ada di pesantren. Pada penelitian ini juga akan menguji beban kerja sistem yang diakses oleh 40 dan 140 pengguna menggunakan metode *load testing* dan menggunakan tool *Apache Jmeter*.

II. METODOLOGI PENELITIAN

A. Metode Perancangan

Metode perancangan yang digunakan dalam penelitian ini adalah metode *waterfall*. Metode *Waterfall* merupakan salah satu model SDLC yang sering digunakan dalam pengembangan sistem informasi atau perangkat lunak. Model ini menggunakan pendekatan sistematis dan berurutan. Untuk tahapan dari metode *waterfall* dapat dilihat pada gambar 1 berikut :



Gambar 1. Tahapan metode waterfall

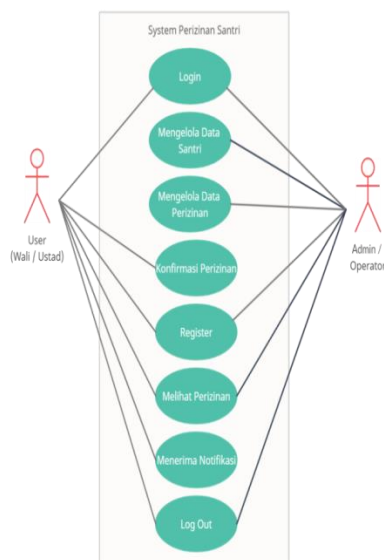
Semua tahapan dari *waterfall* dibuat berurutan dimulai dari proses analisis, kemudian desain[2]. Setelah itu proses implementasi dari desain dan program yang sudah dibuat. Setelah implementasi proses selanjutnya adalah melakukan tes terhadap aplikasi, dan terakhir adalah pemeliharaan terhadap sistem[3].

B. Metode penelitian

Untuk pengujian kehandalan jaringan pada pengujian ini menggunakan metode load testing. *Load Testing* adalah Teknik performance testing yang mana respon sistem diukur dalam berbagai *load condition*. *Load testing* memiliki cara pengujian yang menguji berdasarkan keadaan yang sebenarnya pada dunia nyata agar *website*, aplikasi ataupun *software* yang sedang dibangun dan dikembangkan dapat dimanfaatkan dan bekerja secara maksimal ketika sudah digunakan oleh user[4].

C. Perancangan Usecase Diagram

Use case diagram dirancang untuk melihat bagaimana sistem bekerja, melihat bagaimana sistem tersebut berinteraksi dengan penggunanya[5]. Adapun bentuk rancangan dari *use case* dalam penelitian ini dapat dilihat pada gambar 2.



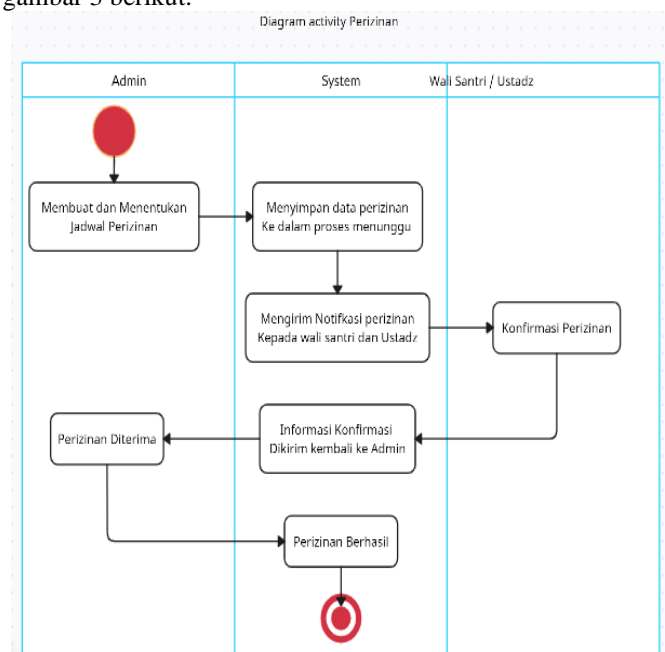
Gambar 2. Usecase Diagram

Untuk penjelasan dari gambar 2 dapat diuraikan sebagai berikut.

1. *Admin* dan *User* harus *login* terlebih dahulu untuk dapat mengakses aplikasi.
2. *Admin* dapat mengelola data santri dan pengguna.
3. *Admin* dapat mengelola data perizinan.
4. *Admin* dapat melihat perizinan.
5. *User* dapat melihat perizinan terkait dari santri
6. *User* mendapat notifikasi perizinan.
7. *User* dapat mengkonfirmasi perizinan.
8. *Admin* dan *user* dapat *logout*.

D. Perancangan Activity perizinan

Diagram activity perizinan memperlihatkan alur dari proses perizinan yang terjadi di dalam sistem. Adapun perancangan diagram *activity* perizinan dapat dilihat pada gambar 3 berikut.

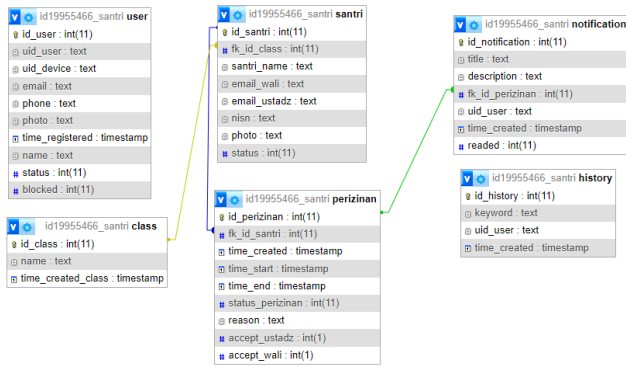


Gambar 3. Diagram Activity Perizinan

Proses perizinan yang terjadi di dalam sistem terdapat beberapa proses. Untuk memulai membuat perizinan *admin* akan membuat sebuah izin baru pada aplikasi, kemudian data izin tersebut akan disimpan dalam proses menunggu. Setelah sistem menerima data perizinan, sistem akan mengirimkan perizinan tersebut kepada wali santri dan ustad untuk di konfirmasi perizinan. Setelah konfirmasi sudah dilakukan, data konfirmasi perizinan tersebut akan diteruskan kembali ke *admin*. Dan proses terakhir *admin* menyetujui perizinan dan proses perizinan berhasil.

E. Perancangan relasi tabel database

Perancangan tabel *database* dilakukan untuk memudahkan proses dan fungsi dan hubungan dari tabel *database* pada sistem. Untuk perancangan *database* dan relasinya dapat dilihat pada gambar 4 berikut.



Gambar 4. Relasi Tabel Database

Database dibuat dalam 6 buah tabel, untuk setiap tabel ada yang mempunyai relasi terhadap tabel yang lain. Untuk tabel *user* dapat dilihat berelasi dengan tabel perizinan, fungsinya agar tabel dari perizinan tersinkron dengan tabel user. Dan juga tabel *class* memiliki relasi dengan tabel santri. Dan tabel perizinan juga terelasi dengan tabel *notification*.

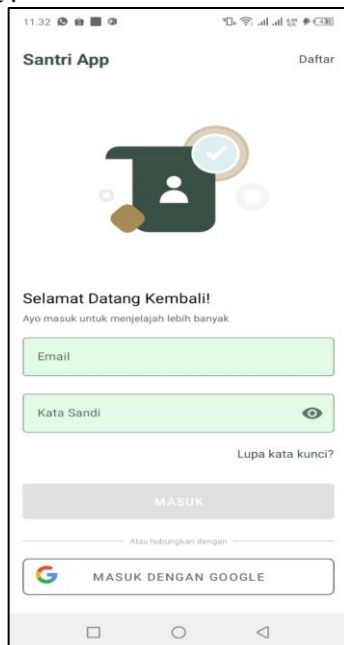
III. HASIL DAN PEMBAHASAN

A. Tampilan antarmuka

Pada bagian ini menampilkan hasil tampilan yang dibuat berupa tampilan *login*, tampilan registrasi, tampilan menu perizinan, dan tampilan menu konfirmasi perizinan.

1. Tampilan halaman login

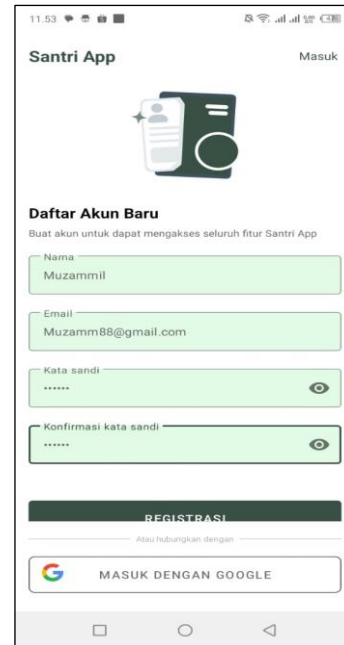
Proses *login* dibuat sebagai langkah awal dari pengguna agar bisa masuk pada aplikasi perizinan santri, pada sistem ini *login* dapat dilakukan oleh 3 pengguna yaitu *admin*, orang tua santri, dan ustad. Untuk tampilan *login* dapat dilihat pada gambar 5 berikut :



Gambar 5. Tampilan Login

2. Tampilan registrasi

Menu registrasi digunakan agar pengguna dapat mendaftar ke dalam sistem. pengguna harus memasukkan nama, email, dan kata sandi untuk dapat melakukan registrasi. Email yang akan didaftarkan harus didaftarkan terlebih dahulu di data santri. Untuk tampilan registrasi dapat dilihat pada gambar 6 berikut.



Gambar 6. Tampilan registrasi

Pengguna yang ingin didaftarkan ke sistem perizinan haruslah email yang aktif, karena setelah melakukan registrasi akan masuk pemberitahuan ke email terkait.

3. Tampilan menu perizinan

Santri yang ingin mendapatkan izin dapat menjumpai *admin*, selanjutnya *admin* dapat menambahkan data perizinan dengan memasukkan nama santri di menu perizinan. Untuk perizinannya dibuat dalam 4 tahap. Tahap yang pertama adalah menunggu, santri yang sudah mengajukan izin harus menunggu persetujuan dari wali murid dan gurunya dulu agar perizinannya dapat diproses. Tahap yang kedua adalah proses, yaitu izin yang sudah aktif dan sedang berjalan. Tahap yang ketiga adalah terlambat, santri yang sudah melebihi batas waktu izin, informasinya akan ditampilkan di tampilan terlambat. Dan tahap terakhir adalah selesai, informasi yang terlihat apabila santri kembali tepat waktu dengan batas waktu izin yang diberikan. Perizinan yang dibuat ini sudah sesuai dengan panduan dari sistem perizinan yang diterapkan di pesantren. Untuk halaman menu perizinan dapat dilihat pada gambar 7 berikut.



Gambar 7. Tampilan menu perizinan

4. Tampilan konfirmasi perizinan

Setelah izin diberikan oleh *admin*, langkah selanjutnya adalah perizinan tersebut dikonfirmasi oleh wali santri dan ustad, untuk tampilan konfirmasi perizinan dapat dilihat pada gambar 8 berikut.

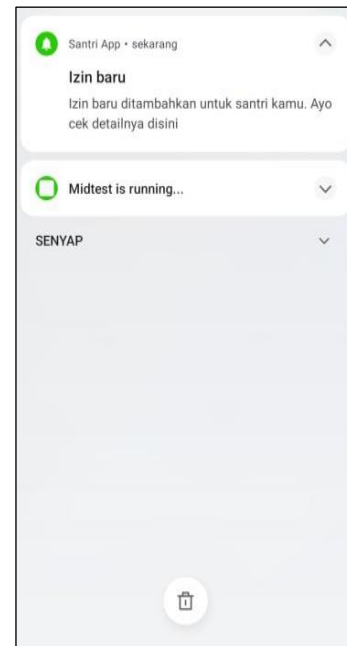


Gambar 8 Tampilan konfirmasi perizinan

Santri yang sudah diberikan izin oleh *admin* langkah selanjutnya agar izin tersebut dapat aktif adalah wali murid dan ustad harus mengkonfirmasi dulu izin yang masuk ke aplikasi pengguna. Setelah izin dikirim oleh *admin*, izin tersebut akan

muncul di aplikasi yang ter-*install* di perangkat wali murid dan ustad, dan juga akan muncul notifikasi.

Untuk notifikasinya dapat dilihat pada gambar 9.



Gambar 9 Notifikasi perizinan

B. Pengujian Load Testing

Pengujian *load testing* yang dilakukan adalah menggunakan sebuah *tool* yaitu *apache jmeter*, dengan menggunakan *tool apache jmeter* dapat membuat beberapa *virtual user* sehingga seolah sistem sedang diakses oleh banyak pengguna. Pada penelitian ini pengujian *load testing* dilakukan 2 kali percobaan, percobaan pertama dengan menjadikan beban sistem diakses oleh 40 pengguna, dan percobaan kedua beban sistem diakses oleh 140 pengguna.

1. Pengujian dengan menggunakan 40 *virtual user*

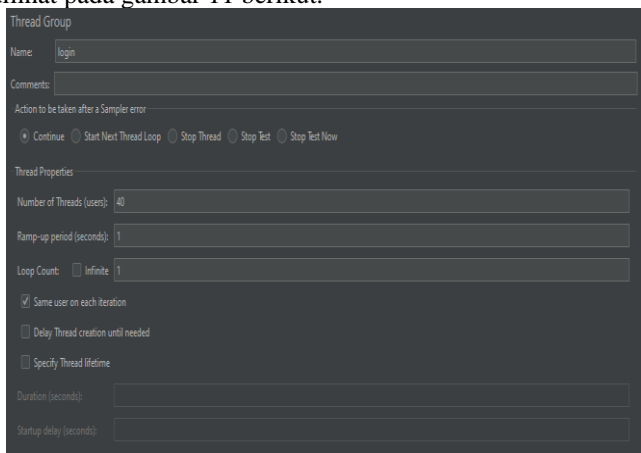
Untuk *settingan* dari pengujian, langkah pertama yang dilakukan adalah membuka menu *HTTP request* untuk mengisi *web server* sesuai dengan nama *web server* yang dibuat telah dibuat untuk aplikasi perizinan santri. Untuk tampilannya dapat dilihat pada gambar 10 berikut.



Gambar 10 Tampilan HTTP Request

Dapat dilihat pengujian kinerja sistem tersebut yaitu pada *web server* *santriapp000.webhostapp.com*. Untuk menambahkan *virtual user* pada *web server* tersebut dapat

dilakukan pada menu *thread group*. Untuk tampilannya dapat dilihat pada gambar 11 berikut.



Gambar 11. Tampilan *Thread Group* untuk 40 *virtual user*

Berdasarkan gambar 11 dapat dilihat *number of thread* di isi dengan 40 *virtual user*, yang menandakan seolah-olah ada 40 pengguna yang melakukan akses ke dalam sistem. Untuk hasil dari pengujian menggunakan 40 *virtual user* yang mengakses *web server* tersebut dapat dilihat pada gambar 12 berikut.

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received K...	Sent KB/sec
HTTP Reque...	40	1099	1033	1367	1501	1608	839	1608	0.00%	21.3/sec	158.17	2.69
TOTAL	40	1099	1033	1367	1501	1608	839	1608	0.00%	21.3/sec	158.17	2.69

Gambar 12. Tampilan hasil pengujian menggunakan 40 *virtual user*

Dari gambar 12 dapat disimpulkan waktu respon server saat diakses oleh 40 *virtual user* adalah dengan waktu minimum 0,8 detik, maksimum 1,6 detik, dan rata-rata 1,09 detik untuk setiap *user* dalam mengakses aplikasi.

2. Pengujian dengan menggunakan 140 *virtual user*

Untuk pengujian ke 2 menggunakan 140 *virtual user*. *settingan* dari pengujiannya langkah pertama sama seperti langkah pada pengujian untuk 40 *virtual user*, yaitu membuka menu *HTTP request* untuk mengisi *web server* sesuai dengan nama *web server* yang dibuat telah dibuat untuk aplikasi perizinan santri. Setelah mengisi alamat *web server*, Untuk menambahkan *virtual user* pada *web server* tersebut dapat dilakukan pada menu *thread group*. Untuk tampilannya dapat dilihat pada gambar 13 berikut.



Gambar 13. Tampilan *Thread Group* untuk 140 *virtual user*

Berdasarkan gambar 10 dapat dilihat *number of thread* di isi dengan 140 *virtual user*, yang menandakan seolah-olah ada 140 pengguna yang melakukan akses ke dalam sistem. Untuk hasil dari pengujian menggunakan 140 *virtual user* yang mengakses *web server* tersebut dapat dilihat pada gambar 14 berikut.

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received K...	Sent KB/sec
HTTP Reque...	140	1575	1555	2008	2133	2362	855	2500	0.00%	11.2/min	1.39	0.02
TOTAL	140	1575	1555	2008	2133	2362	855	2500	0.00%	11.2/min	1.39	0.02

Gambar 14. Tampilan hasil pengujian menggunakan 140 *virtual user*

Dari gambar 11 dapat disimpulkan waktu respon server saat diakses oleh 140 *virtual user* adalah dengan waktu minimum 0,8 detik, maksimum 2,5 detik, dan rata-rata 1,57 detik untuk setiap *user* dalam mengakses aplikasi.

Setelah dilakukan 2 pengujian di atas didapatkan data waktu akses *server* berbeda-beda. Untuk perbedaan akses nya dapat dilihat pada tabel 1 berikut.

TABEL I
PENGUJIAN WAKTU RESPON SERVER

Pengujian	Waktu respon server (detik)		
	Min	Max	Rata Rata
40 <i>virtual user</i>	0,8	1,6	1,09
140 <i>virtual user</i>	0,8	2,5	1,57

IV. KESIMPULAN

Berdasarkan hasil pengujian pada penelitian yang sudah dilakukan yang berjudul “Rancang Bangun Aplikasi Perizinan Santri Berbasis Android” maka dapat diambil kesimpulan bahwa aplikasi perizinan dibuat agar proses perizinan yang dilakukan di pesantren dapat dilakukan dengan lebih aman, dan juga efisien. Dan setelah melakukan pengujian terhadap aplikasi perizinan santri dengan menggunakan *apache jmeter* kinerja server saat diakses oleh 40 dan 140 pengguna,

diperoleh jika sistem tersebut diakses oleh 40 pengguna didapatkan waktu akses dengan nilai rata-rata 1.09 detik. Dan jika sistem diakses oleh 140 pengguna di dapatkan waktu akses dengan nilai rata-rata 1.57 detik.

REFERENSI

- [1] Makmun, H. A. R. (2014). Pembentukan Karakter Berbasis Pendidikan Pesantren: Studi di Pondok Pesantren Tradisional dan Modern Di Kabupaten Ponorogo. *Cendekia Vol.*, 12(2).
- [2] Wahyudi, A. (2011). Pengaruh variabel-variabel ekuitas merek terhadap keputusan pembelian telepon seluler Blackberry: studi pada Mahasiswa S-1 Universitas Brawijaya Malang. Universitas Brawijaya.
- [3] Damayanti, D., & Sumiati, S. (2018). Sistem Informasi Daya Tarik Pembelian Produk UMKM Home Industri Berbasis WEB. Konferensi Nasional Sistem Informasi (KNSI) 2018
- [4] Permatasari, D. I., Santoso, B., Ningtias, N., YR, M. H., Atika, R., Widad, N., & Maulana, I. (2019, September). Pengukuran throughput load testing menggunakan test case sampling gorilla testing. In *Seminar Nasional Sistem Informasi (SENASIF)* (Vol. 3, pp. 2008-2014).
- [5] Novianti, H., Allsela, M., & Nurul. (2016). Penerapan Konsep Customer Relationship Management (Crm) Pada Sistem Informasi Penyewaan Lapangan Futsal Di Swadaya Futsal Palembang. *Jurnal Sistem Informasi (JSI)*, 8(2), 2355–4614.

