Rancang Bangun Sistem Informasi Pemesanan dan Pemantauan Slot Lapangan Olahraga Menggunakan Framework Next.js dan Firebase

Andreansyah Akmal Chaeruddin
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
andreanss@student.telkomuniversity.ac.id

Anggunmeka Luhur Prasasti
Fakultas Teknik Elektro
Universitas Telkom
Bandung Indonesia
anggunmeka@telkomuniversity.ac.id

Rogers Dwiputra Setiady
Fakultas Teknik Elektro
Universitas Telkom
Bandung Indonesia
rogerssetiady@telkomuniversity.ac.id

***Abstrak* — Pengelolaan fasilitas olahraga seperti Gelanggang Olahraga (GOR) secara konvensional sering kali menghadapi kendala administratif pada sistem pemesanan yang tidak terpusat dan masih bersifat manual. Hal tersebut mengakibatkan ketidakpastian informasi ketersediaan jadwal bagi pengguna serta meningkatkan risiko tumpang tindih pesanan (*double booking*). Penelitian ini bertujuan untuk mengembangkan sebuah platform digital terintegrasi yang menggabungkan perangkat keras berbasis *Internet of Things* (IoT) menggunakan sensor LDR untuk pemantauan penggunaan lapangan secara langsung. Sistem ini memanfaatkan framework Next.js sebagai kerangka kerja aplikasi web yang responsif dan Firebase sebagai basis data real-time untuk memastikan sinkronisasi data terjadi secara instan antara perangkat fisik dan antarmuka pengguna. Selain fungsionalitas dasar, penelitian ini memberikan penekanan pada pengujian aspek ketangguhan sistem dalam menangani skenario kegagalan, seperti perlindungan terhadap pemesanan ganda dan mekanisme otomatisasi pembatalan untuk pembayaran yang telah kedaluwarsa melalui integrasi gateway pembayaran Midtrans. Berdasarkan hasil pengujian performa menggunakan Google Lighthouse, sistem berhasil mencapai skor rata-rata sebesar 85-86% yang menunjukkan tingkat optimasi perangkat lunak yang sangat baik dari sisi aksesibilitas dan SEO. Hasil pengujian latensi menunjukkan bahwa rata-rata pengiriman notifikasi konfirmasi melalui WhatsApp memerlukan waktu selama 5 detik. Secara keseluruhan, hasil penelitian membuktikan bahwa sistem ini mampu menawarkan solusi manajemen GOR yang transparan, aman, dan efisien.**

Kata kunci*— *GOR, Next.js, Firebase, IoT, Notifikasi WhatsApp.

I. PENDAHULUAN

Kebutuhan akan fasilitas olahraga yang memadai di daerah perkotaan terus meningkat seiring dengan tingginya kesadaran masyarakat akan gaya hidup sehat. Namun, peningkatan minat ini sering kali tidak diimbangi dengan sistem manajemen fasilitas yang modern. Di banyak tempat, pengelola GOR masih mengandalkan pencatatan manual atau koordinasi melalui pesan singkat untuk menangani pemesanan lapangan. Dampaknya, informasi mengenai ketersediaan lapangan sering kali tidak akurat, yang berujung pada kekecewaan pengguna akibat jadwal yang bentrok (*double booking*)[1][2].

Selain masalah administratif, pemantauan fisik terhadap penggunaan lapangan juga menjadi tantangan tersendiri. Pengelola sering kali sulit memastikan apakah lapangan benar-benar digunakan sesuai dengan durasi yang dipesan. Oleh karena itu, diperlukan sebuah solusi yang dapat menyinkronkan antara data pemesanan di sisi perangkat lunak dengan kondisi fisik di lapangan secara otomatis [3].

Penelitian ini bertujuan untuk merancang sebuah sistem informasi yang mampu menjembatani kendala tersebut. Dengan memanfaatkan teknologi IoT berbasis ESP32 untuk deteksi status lampu lapangan dan sistem web berbasis Next.js serta Firebase, diharapkan tercipta ekosistem manajemen GOR yang transparan, aman, dan efisien. Keunggulan dari penelitian ini terletak pada integrasi sistem pembayaran otomatis dan mekanisme perlindungan data yang menjamin stabilitas sistem meskipun terjadi gangguan pada alur transaksi normal.

II. KAJIAN TEORI

A. Framework Next.js dan Keunggulannya

Next.js dipilih sebagai kerangka kerja utama karena kemampuannya dalam melakukan *Server-Side Rendering* (SSR) [4]. Berbeda dengan aplikasi React biasa yang merender konten di sisi klien, SSR memungkinkan halaman web dimuat lebih cepat karena konten sudah disiapkan di server sebelum dikirim ke peramban. Fitur ini sangat krusial untuk aplikasi pemesanan yang membutuhkan performa tinggi dan optimasi mesin pencari (SEO) agar mudah ditemukan oleh calon pengguna[5].

B. Firebase sebagai Database Real-Time

Dalam sebuah sistem monitoring, kecepatan pembaruan data adalah segalanya. Firebase *Realtime Database* menyediakan koneksi berbasis *WebSocket* yang memungkinkan data mengalir secara instan antara server dan aplikasi pengguna tanpa perlu memuat ulang halaman. Hal ini memastikan bahwa begitu sensor IoT mendeteksi perubahan status lapangan, status tersebut akan langsung tercermin di layar ponsel pengguna dalam hitungan milidetik [6][7].

C. Integrasi Perangkat IoT ESP32 dan Sensor LDR

Sensor LDR bekerja berdasarkan prinsip foto-konduktivitas, di mana nilai resistansi (R) berbanding terbalik dengan intensitas cahaya (E) yang diterima. Hubungan ini dapat dirumuskan sebagai berikut:

$$R = R_{10} \times \left(\frac{10}{E}\right)^{a}$$

Keterangan:

- $\boldsymbol{R}$ adalah nilai resistansi dari sensor LDR dalam satuan Ohm
- $\boldsymbol{R_{10}}$ merupakan nilai resistansi sensor pada saat menerima intensitas cahaya sebesar 10 lux.
- $\boldsymbol{E}$ adalah tingkat intensitas cahaya yang mengenai permukaan sensor dalam satuan lux.
- $\boldsymbol{a}$ adalah konstanta kemiringan (slope) yang merepresentasikan tingkat sensitivitas sensor terhadap perubahan cahaya di lingkungan lapangan.

Untuk pembacaan data oleh ESP32, digunakan rangkaian pembagi tegangan guna mendapatkan nilai tegangan keluaran (V Out):

$$V_{out} = V_{in} \times \left(\frac{R_{LDR}}{R_{LDR} + R_{static}}\right)$$

Keterangan:

- $\boldsymbol{V_{out}}$ merupakan tegangan keluaran yang masuk ke pin ADC (*Analog to Digital Converter*) pada ESP32 dalam satuan Volt.
- $\boldsymbol{V_{in}}$ adalah tegangan masukan sumber daya dari mikrokontroler, yang dalam penelitian ini menggunakan tegangan sebesar 3,3 Volt.
- $\boldsymbol{R_{LDR}}$ adalah nilai hambatan variabel dari sensor LDR yang berubah-ubah sesuai kondisi pencahayaan lapangan.
- $\boldsymbol{R_{static}}$ adalah nilai hambatan tetap yang digunakan sebagai pembanding dalam rangkaian pembagi tegangan agar nilai V out tetap berada pada rentang yang dapat dibaca oleh ESP32.

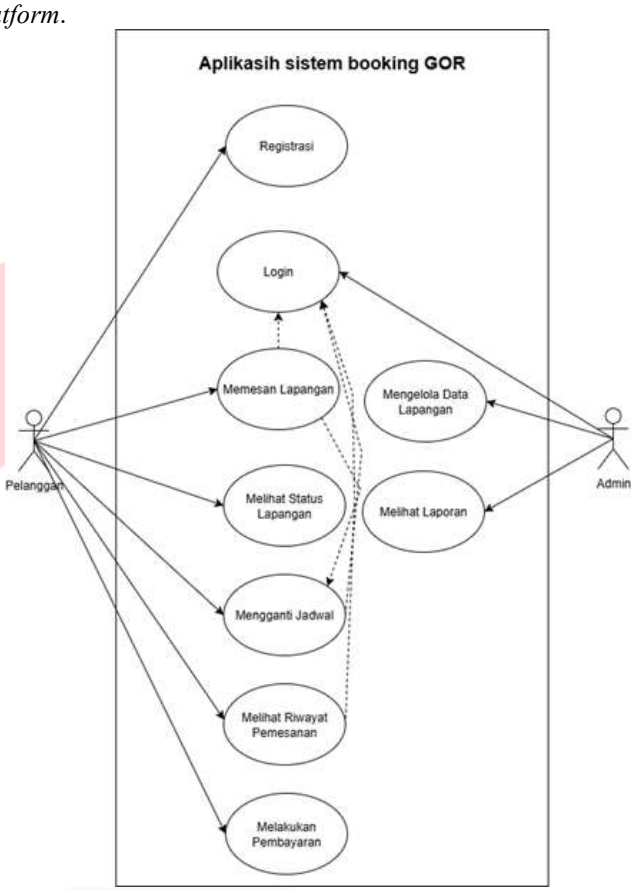
D. Keamanan Transaksi dan Payment Gateway

Kepercayaan pengguna sangat bergantung pada keamanan pembayaran. Integrasi dengan Midtrans memungkinkan sistem menerima berbagai jenis pembayaran mulai dari QRIS hingga transfer bank secara otomatis. Melalui mekanisme signature key SHA512, sistem memastikan bahwa setiap notifikasi status lunas yang masuk benar-benar berasal dari server Midtrans yang sah, bukan dari manipulasi pihak luar.

III. METODE

A. Analisis Fungsional Melalui *Use Case Diagram*

Tahap awal pengembangan sistem dimulai dengan pemetaan kebutuhan fungsional melalui *Use Case Diagram*. Diagram ini memberikan gambaran komprehensif mengenai interaksi antara aktor utama, yaitu Pengguna (*User*) dan Administrator, dengan fitur-fitur yang tersedia di dalam *platform*.

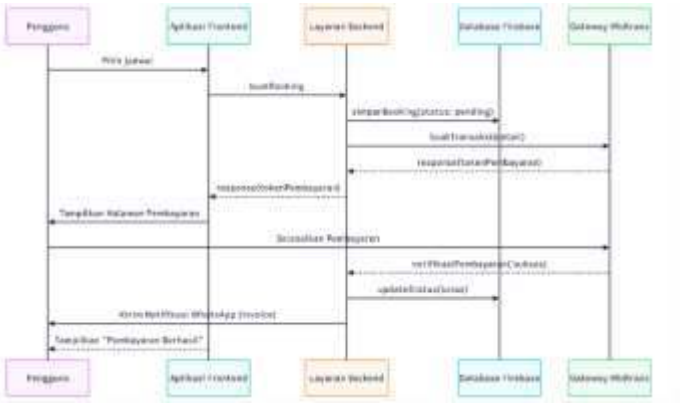


GAMBAR 1
(Use Case Diagram)

Secara teknis, User diberikan akses untuk melakukan pendaftaran akun, pemantauan status lapangan secara *real-time*, hingga proses pemesanan dan pembayaran. Di sisi lain, *Administrator* memiliki otoritas penuh untuk melakukan manajemen jadwal, validasi data transaksi, serta pemantauan teknis terhadap perangkat IoT yang terpasang di lapangan. Pemisahan peran ini penting untuk menjaga integritas data dan memastikan setiap fitur berjalan sesuai dengan hak akses yang telah ditentukan.

B. Logika Sistem pada *Activity Diagram* Alur Booking

Untuk memahami bagaimana sistem menangani proses transaksi dari awal hingga akhir, dirancang sebuah *Activity Diagram* yang merinci alur pemesanan lapangan. Alur ini sangat krusial karena melibatkan sinkronisasi antara aksi pengguna, respon basis data Firebase, dan verifikasi dari layanan pihak ketiga (Midtrans).

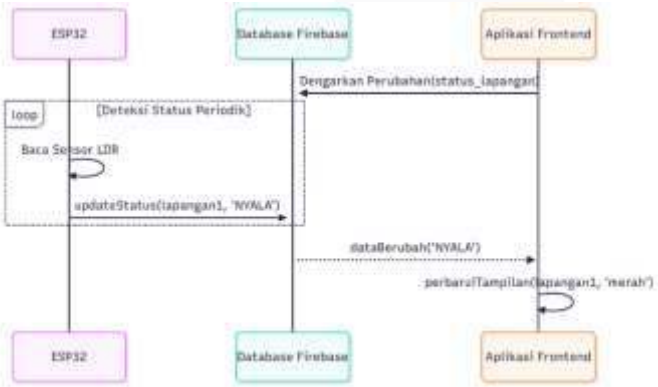


GAMBAR 2
*(*Activity Diagram Alur Booking)

Proses dimulai saat pengguna memilih slot lapangan yang berstatus "Tersedia". Pada tahap ini, sistem menerapkan logika pengecekan ganda guna menghindari *race condition*. Jika slot berhasil divalidasi, sistem akan mengarahkan pengguna ke halaman pembayaran. Komunikasi asinkron digunakan antara *back-end* sistem dan Midtrans untuk memantau status pembayaran. Apabila transaksi dinyatakan berhasil, status pada Firebase akan diperbarui secara otomatis dan memicu pengiriman notifikasi WhatsApp sebagai bukti reservasi yang sah.

C. Perancangan Perangkat Keras dan Pemantauan IoT (ESP32)

Bagian ini menjelaskan mekanisme deteksi ketersediaan lapangan secara fisik melalui perangkat keras berbasis ESP32. Penggunaan mikrokontroler ini didasarkan pada efisiensi konsumsi daya dan kemampuannya dalam mengelola konektivitas nirkabel secara stabil [8].



GAMBAR 3
*(*Activity Diagram ESP32)

Inti dari pemantauan ini terletak pada pembacaan nilai resistansi dari sensor LDR yang dipasang di area lampu lapangan. ESP32 melakukan pembacaan secara periodik melalui pin ADC dan mengonversinya menjadi data logika. Logika keputusan pada perangkat IoT ini didasarkan pada ambang batas (*threshold*) intensitas cahaya tertentu untuk membedakan antara lampu lapangan yang menyala dan cahaya lingkungan sekitar, guna memastikan akurasi data yang dikirimkan ke basis data Firebase.

D. Manajemen Reschedule dan Integrasi Pembayaran

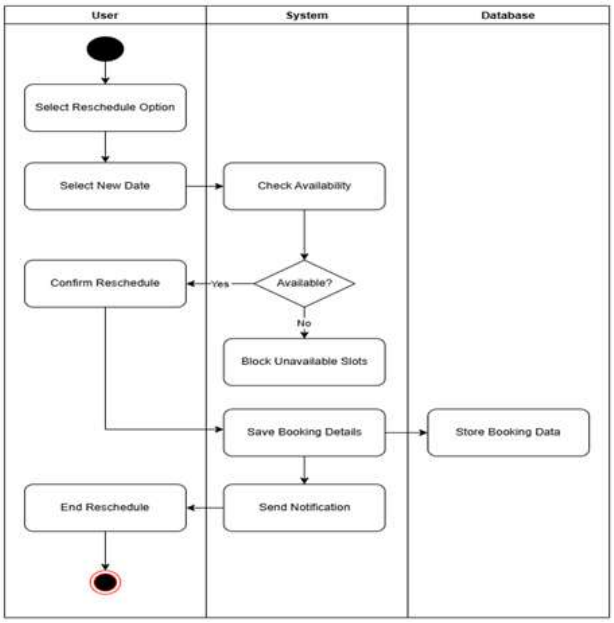
Fleksibilitas sistem menjadi aspek penting lainnya yang dikembangkan melalui fitur *reschedule* atau ubah jadwal. Alur pengubahan jadwal ini dirancang sedemikian rupa agar tetap mengikuti aturan ketersediaan yang berlaku tanpa mengganggu transaksi lain yang sedang berjalan.



GAMBAR 4
*(*Activity Diagram Payment)

Activity Diagram Payment pada Gambar 4 mengilustrasikan mekanisme transaksi yang terintegrasi dengan *payment gateway*. Proses ini dimulai ketika sistem menghasilkan token transaksi unik yang dikirimkan ke server Midtrans. Interaksi asinkron terjadi di mana sistem menunggu notifikasi (*webhook*) dari server pembayaran untuk menentukan status akhir transaksi, apakah *settlement* (lunas), *pending*, atau *expire*. Keamanan pada tahap ini sangat krusial; sistem hanya akan memperbarui status pesanan di Firebase jika kunci tanda tangan (*signature key*) dari notifikasi tersebut terverifikasi sah. Keberhasilan tahap

pembayaran ini merupakan prasyarat mutlak sebelum pengguna dapat mengakses fitur perubahan jadwal.



GAMBAR 5
*(*Activity Diagram Reschedule)

Setiap permintaan perubahan jadwal akan memicu validasi ulang terhadap slot waktu yang baru. Jika slot baru tersebut tersedia, sistem akan memindahkan data pesanan pengguna dan memperbarui kunci (*key*) referensi pada Firebase. Integrasi dengan sistem pembayaran Midtrans memastikan bahwa perubahan ini hanya dapat dilakukan jika status pembayaran awal sudah terverifikasi lunas, guna menghindari penyalahgunaan sistem oleh pihak yang tidak bertanggung jawab.

E. Perancangan Struktur Data NoSQL

Mengingat sistem menggunakan Firebase Realtime Database, arsitektur data disusun dalam format JSON (*JavaScript Object Notation*) yang mendukung sinkronisasi data secara asynchronous. Struktur ini dirancang untuk meminimalkan latensi saat terjadi pembaruan status lapangan dari perangkat IoT. Skema basis data utama dirangkum dalam Tabel 1:

TABEL 1
(Skema Struktur Data Pada Firebase)

Node Utama	**Atribut**	**Tipe Data**	**Deskripsi**
/lapangan	status, last_update	String, Timestamp	Menyimpan status fisik (Tersedia/Dipakai)
/bookings	user_id, slot, status_bayar	String, Object	Menyimpan data reservasi aktif
/transactions	order_id, amount, signature	String, Integer	Log validasi pembayaran Midtrans

F. Arsitektur Komunikasi Data

Sistem mengimplementasikan protokol *WebSocket* yang disediakan secara *native* oleh Firebase untuk memastikan state pada aplikasi Next.js diperbarui tanpa melalui proses *page refresh*. Saat sensor LDR mengirimkan data ke Firebase, listener pada sisi klien (*website*) akan mendeteksi perubahan tersebut dan secara otomatis mengubah warna indikator lapangan pada antarmuka pengguna. Selain itu, integrasi dengan Midtrans menggunakan mekanisme Server-Side API Routes di Next.js untuk menangani webhook pembayaran secara aman di sisi server.

G. Mekanisme Keamanan *Webhook* dan *Signature Key*

Untuk menjamin integritas data transaksi, sistem mengimplementasikan validasi *Signature Key* pada setiap notifikasi (*webhook*) yang dikirim oleh Midtrans. Proses ini krusial untuk mencegah serangan *Man-in-the-Middle* di mana pihak ketiga mencoba memanipulasi status pembayaran. Server Next.js akan menghasilkan hash SHA512 berdasarkan perpaduan order_id, status_code, dan gross_amount untuk kemudian dicocokkan dengan key yang dikirim oleh Midtrans. Jika tidak sesuai, sistem akan secara otomatis menolak pembaruan status pada Firebase.

H. Spesifikasi Perangkat Lunak

Implementasi sistem ini didukung oleh berbagai perkakas pengembangan untuk memastikan stabilitas kode dan efisiensi performa. Spesifikasi lingkungan pengembangan dirangkum dalam Tabel 2.

TABEL 2
(Spesifikasi Perangkat Lunak Pengembangan)

Komponen	**Versi / Jenis**	**Fungsi**
Sistem Operasi	Windows 10 / 11	Sistem operasi untuk pengembangan
Code Editor	Visual Studio Code	Penulisan kode untuk aplikasi
Runtime	Node.js v18.x / v20.x	Eksekusi JavaScript di sisi server
Framework	Next.js 14 (*App Router*)	Arsitektur web dan routing
Language	TypeScript / JavaScript	Bahasa pemrograman utama
Database	Firebase SDK v10.x	Integrasi basis data real-time

IV. HASIL DAN PEMBAHASAN

Bagian ini memaparkan secara komprehensif hasil dari implementasi sistem yang telah dirancang, mencakup evaluasi fungsionalitas perangkat keras IoT serta performa aplikasi web yang dibangun untuk manajemen fasilitas olahraga. Pengujian dilakukan untuk memastikan bahwa integrasi antara sensor LDR, mikrokontroler ESP32, basis data Firebase, dan framework Next.js dapat berjalan secara sinkron dalam skenario penggunaan nyata. Selain pengujian alur normal (*happy path*), analisis juga difokuskan pada ketangguhan sistem dalam menangani kondisi galat (*negative path*) guna menjamin integritas data dan keamanan transaksi. Penilaian objektif dilakukan melalui serangkaian pengujian

teknis, termasuk audit performa menggunakan Google Lighthouse untuk mengukur efisiensi sistem serta pemantauan latensi pada layanan notifikasi otomatis.

A. Analisis Ketangguhan Logika Transaksi (*Negative Path*)

Pengujian alfa difokuskan pada ketangguhan sistem dalam menangani kondisi di luar alur normal guna memastikan kestabilan operasional GOR [9]. Hasil pengujian dirinci pada Tabel 3.

TABEL 3
(Tabel Pengujian Alfa Ketangguhan Sistem)

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	*Double Booking simulation*	Sistem menolak transaksi kedua pada slot yang sama	Logika *transactional update* berhasil mencegah pemesanan ganda	Valid
2	Pembayaran Kedaluwarsa	Slot kembali tersedia jika batas waktu bayar habis	*Webhook Midtrans* memicu perubahan status otomatis di Firebase	Valid
3	Akses Ilegal Panel Admin	Penolakan akses ke URL /admin tanpa login	*Middleware* Next.js melakukan *redirect* ke halaman login	Valid

B. Pengujian Performa Antarmuka Web (Lighthouse)

Kualitas teknis aplikasi web diuji menggunakan Google Lighthouse untuk mengukur tingkat optimasi pada sisi frontend. Pengujian dilakukan pada halaman utama dan halaman detail pemesanan.

TABEL 4
(Tabel Pengujian Performa Website)

Indikator Penilaian	Halaman Utama (*Landing*)	Halaman Pemesanan (*Booking*)
Performance	85%	86%
Accessibility	87%	92%
Best Practices	100%	100%
SEO	100%	100%

Skor pada Tabel 2 menunjukkan bahwa optimasi yang diterapkan pada framework Next.js mampu memberikan responsivitas yang baik bagi pengguna. Tingginya skor pada kategori Best Practices dan SEO (100%) menjamin bahwa platform mudah diakses dan ramah terhadap mesin pencari.

Analisis lebih mendalam terhadap skor Lighthouse menunjukkan bahwa nilai 100% pada kategori SEO dicapai berkat penerapan metadata yang dinamis dan struktur HTML yang semantik pada framework Next.js. Hal ini sangat krusial agar platform GOR mudah ditemukan oleh calon penyewa melalui mesin pencari. Di sisi lain, nilai Accessibility sebesar 92% pada halaman booking mencerminkan bahwa kontras warna dan ukuran elemen navigasi sudah memenuhi standar inklusivitas bagi berbagai kalangan pengguna.

Skor Performa (85-86%) dipengaruhi oleh optimasi gambar menggunakan komponen next/image yang secara otomatis melakukan *lazy-loading*, sehingga beban muat halaman pertama (*First Contentful Paint*) menjadi sangat ringan meskipun data jadwal diambil secara *real-time* dari Firebase.



GAMBAR 6
(Hasil Audit Lighthouse Halaman Utama)



GAMBAR 7
(Hasil Audit Lighthouse Halaman Booking)

C. Pengujian Latensi Notifikasi dan Sinkronisasi IoT

Sinkronisasi antara status fisik lapangan yang dideteksi ESP32 dan notifikasi konfirmasi pembayaran merupakan bagian krusial dari efisiensi sistem.

Selain performa web, kecepatan pengiriman pesan otomatis melalui WhatsApp diukur sejak status pembayaran berubah menjadi lunas (*settlement*) [10]. Berdasarkan pengujian berulang, didapatkan rata-rata latensi pengiriman notifikasi sebagai berikut:

$$T_{avg} = \frac{\sum_{i=1}^{n} ti}{n} = 5\ Detik$$

Penjelasan dari variabel pada perhitungan rata-rata latensi tersebut adalah:

- $\boldsymbol{T_{avg}}$ adalah nilai rata-rata latensi pengiriman notifikasi dari sistem ke perangkat pengguna dalam satuan detik.
- $\boldsymbol{t_i}$ merepresentasikan waktu respon yang terukur pada setiap percobaan ke-i.
- $\boldsymbol{n}$ adalah jumlah total sampel atau percobaan yang dilakukan selama tahap pengujian (dalam penelitian ini dilakukan sebanyak 10 kali percobaan secara acak).
- ∑ adalah simbol sigma yang menunjukkan penjumlahan dari seluruh data waktu respon yang diperoleh selama pengujian berlangsung.

Hasil perhitungan tersebut membuktikan bahwa integrasi layanan notifikasi asinkron telah berjalan secara efisien, memberikan kepastian instan bagi pengguna setelah melakukan transaksi.

D. Pengujian Beta dan Analisis Kepuasan Pengguna

Selain pengujian teknis, dilakukan pengujian beta kepada 10 responden (calon pengguna GOR) menggunakan kuesioner singkat untuk menilai aspek kemudahan penggunaan website. Hasil pengujian menunjukkan bahwa fitur pemantauan lapangan *real-time* mendapatkan respon positif karena memberikan kepastian jadwal. Skor rata-rata tingkat kepuasan antarmuka mencapai 88%, yang menandakan bahwa navigasi pada aplikasi booking ini sudah cukup intuitif.

V. KESIMPULAN

Berdasarkan hasil rancang bangun dan serangkaian pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem informasi pemesanan dan pemantauan slot lapangan olahraga berbasis framework Next.js dan Firebase ini telah berhasil diimplementasikan dan berfungsi dengan stabil. Integrasi perangkat keras IoT menggunakan mikrokontroler ESP32 dan sensor LDR terbukti mampu melakukan sinkronisasi status fisik lapangan ke dalam basis data secara real-time, sehingga memberikan transparansi informasi yang akurat bagi pengguna maupun pengelola GOR.

Pengujian terhadap kondisi galat (*negative path*) menunjukkan bahwa sistem memiliki tingkat ketangguhan yang tinggi dalam menjaga integritas jadwal, terutama melalui mekanisme *transactional update* yang secara efektif mencegah terjadinya *double booking*. Selain itu, otomatisasi manajemen transaksi melalui integrasi webhook Midtrans berhasil menangani pembatalan pesanan yang kedaluwarsa secara mandiri. Dari sisi kualitas teknis,

Audit menggunakan Google Lighthouse mengonfirmasi efisiensi platform dengan skor performa mencapai 86% serta rata-rata latensi pengiriman notifikasi konfirmasi WhatsApp yang sangat responsif, yakni sebesar 5 detik. Secara keseluruhan, sistem ini telah memenuhi kriteria fungsional dan performa untuk digunakan sebagai solusi modernisasi manajemen fasilitas olahraga yang transparan, aman, dan efisien

REFERENSI

[1] A. Latifah, W. Baswardono, A. M. Ahdan, and R. Aisyah, “Penerapan Metode Web Engineering dalam Pembangunan Sistem Pengelolaan Gedung Olahraga Badminton Berbasis Web.” [Online]. Available: https://jurnal.itg.ac.id/

[2] A. Reservasi, L. Futsal, B. Web, T. Ardiansah, and D. Hidayatullah, “Penerapan Metode Waterfall Pada,” *Journal of Information Technology, Software Engineering, and Computer Science (ITSECS)*, vol. 1, no. 1, 2023.

[3] N. M. Surya and A. A. K, “Online Reservation Management System,” *International Journal of Research in Engineering and Science (IJRES)*, 2023, [Online]. Available: https://www.ijres.org/

[4] Vercel, “Next.js by Vercel - The React Framework,” Next.js.

[5] N. S. Ab Ghani, M. Kassim, and A. H. Awang, “Web design structure with wordpress content management for sports centre booking system,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 3, pp. 1643–1653, 2020, doi: 10.11591/ijeecs.v19.i3.pp1643-1653.

[6] R. Zhang, X. Gao, X. Feng, W. Feng, Y. Hu, and X. Tang, “Research and application of real-time database integration testing method,” in *2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, IEEE, 2020, pp. 222–225. doi: 10.1109/AEMCSE50199.2020.00053.

[7] Google, “Firebase | Google’s Mobile and Web App Development Platform,” Firebase Documentation.

[8] A. Ridho’i, K. Setyadjit, and B. Yordhan, “Sistem Monitoring Suhu Dan Kelembaban Pada Budidaya Jamur Tiram Menggunakan ESP32,” *Jurnal FORTECH*, vol. 4, pp. 20–26, Jan. 2023, doi: 10.56795/fortech.v4i1.4103.

[9] H. Hakim, D. Faqih, D. Deva, I. Hudaya, and M. Ilyas, “Pengujian Alpha dan Beta Testing Pada Aplikasi TIJE,” *TeknoIS : Jurnal Ilmiah Teknologi Informasi dan Sains*, vol. 14, pp. 285–295, Jul. 2024, doi: 10.36350/jbs.v14i2.265.

[10] A. Oktarino and A. Afriansyah, “Pengembangan Aplikasi Bantuan Polisi Berbasis Web: Optimalisasi Penanganan Pengaduan Masyarakat melalui Integrasi API WhatsApp,” *JUTEKOM Jurnal Teknologi dan Ilmu Komputer*, vol. 1, no. 2, pp. 87–95, 2023, [Online]. Available: https://jurnal.itscience.org/index.php/jutekom/article/view/3041