

Pengembangan Sistem Pengundian Berbasis Node.js dengan Penanganan Risiko Integritas Klien

Rionaldi Catur Setyo Utomo^{*1}, Setyoningsih Wibowo²

¹Informatika, Universitas PGRI Semarang, Kota Semarang, 236701036@upgris.ac.id

²Informatika, Universitas PGRI Semarang, Kota Semarang, setyoningsihwibowo@upgris.ac.id

Abstract.

Data integrity and system security are crucial in electronic lottery implementation. This study examines the operation of the Hudatama lottery system, which previously operated as a simple web application using client-side JavaScript. It also proposes improvements to the system structure to address identified security issues. In the legacy system, participant pool creation was performed by importing data from an Excel file and implementing a check on participant numbers, which required a 7-digit random number. The system used localStorage to prevent duplicate numbers, record winner history, and facilitate backup or data recovery when lottery results were pending (Valid/Invalid). Although an admin feature with PIN authentication using a simple hash and the ability to output data was included, the analysis revealed significant client integrity risks and the potential for data manipulation due to the lack of a centralized server-side database. To mitigate these risks, this study developed an architectural evolution by building the backend of the system using Node.js, Express, and SQLite. This update emphasizes the use of state machines for the drawing process, centrally integrating validation processes, and enhancing security, auditability, and authentication, creating a drawing system that is clearer and more protected from manipulation by external parties.

Keywords: Lottery System, Data Integrity, Node.js, SQLite, Client-Side Validation.

Abstrak

Integritas data dan keamanan sistem sangat penting dalam pelaksanaan pengundian elektronik. Penelitian ini mengecek cara kerja sistem pengundian Hudatama yang sebelumnya hanya bekerja sebagai aplikasi web sederhana yang memakai JavaScript di sisi klien, serta menawarkan perbaikan dalam struktur sistem untuk mengatasi masalah keamanan yang telah ditemukan. Dalam sistem lama, pembuatan pool peserta dilakukan dengan mengimpor data dari file Excel dan menerapkan pengecekan terhadap nomor peserta yang harus berupa angka acak sebanyak 7 digit. Sistem ini memakai localStorage untuk mencegah pengambilan nomor yang sama, mencatat sejarah pemenang, serta membantu proses penyimpanan cadangan atau pemulihan data ketika hasil undian masih dalam kondisi pending (Valid/Tidak Valid). Meskipun sudah ada fitur admin dengan autentikasi PIN menggunakan hash sederhana dan kemampuan mengeluarkan data, hasil analisis menunjukkan adanya risiko integritas klien yang besar serta kemungkinan data dimanipulasi karena tidak adanya database terpusat di sisi server. Untuk mengurangi risiko tersebut, penelitian ini mengembangkan evolusi arsitektur dengan membangun bagian belakang sistem menggunakan Node.js, Express, dan SQLite. Pembaruan ini menekankan pada penggunaan state machine untuk proses pengundian, menggabungkan proses validasi secara sentral, serta meningkatkan keamanan, kemampuan audit, dan autentikasi, sehingga menciptakan sistem pengundian yang lebih jelas dan lebih terlindungi dari upaya manipulasi oleh pihak luar.

Kata Kunci: Sistem Pengundian, Integritas Data, Node.js, SQLite, Client-Side Validation.

1. Pendahuluan

Perkembangan teknologi informasi yang sangat cepat dalam beberapa tahun terakhir telah mengubah banyak proses operasional secara digital, termasuk dalam penyelenggaraan sistem pengundian secara elektronik. Berbagai lembaga dan penyelenggara acara saat ini semakin membutuhkan sistem yang bisa memastikan keadilan, keacakan yang dapat dipercaya, serta keamanan data secara sempurna. Integritas data dan keamanan sistem merupakan hal utama dalam sistem pengundian elektronik, karena potensi manipulasi sangat tinggi jika tidak diatur dengan desain yang tepat. Peningkatan ancaman siber terhadap aplikasi berbasis web menunjukkan bahwa kerentanan di sisi klien menjadi penyebab utama dalam kasus penyadapan data transaksi elektronik, sehingga mendorong kebutuhan akan infrastruktur yang lebih aman [1].

Sistem pengundian modern tidak hanya digunakan untuk menentukan pemenang secara acak, tetapi juga perlu memiliki struktur belakang yang kuat agar proses validasi dan pengendalian bisa berjalan dengan aman, seperti memastikan status pemenang dalam kondisi menunggu, sah, atau tidak sah. Tanpa pengelolaan status dan basis data yang terpusat, logika pengundian yang dijalankan di peramban pengguna akan sangat mudah dimanipulasi, seperti perubahan skrip atau peninjauan elemen [2]. Perkembangan arsitektur dari pemrosesan lokal ke sistem peladen (server-side) telah terbukti mampu mencegah tindakan penipuan, meningkatkan kemampuan untuk audit, serta menyediakan satu sumber kebenaran yang akurat secara real-time [3].

Beberapa penelitian sebelumnya sudah membahas cara mengamankan sistem berbasis web dan mengaplikasikan arsitektur server-side agar data tetap utuh dan aman. Nugroho dkk.(2024) dalam penelitiannya dengan judul "Implementasi Node.js dan Express pada Backend Sistem Validasi Data Transaksi Terpusat" berhasil menciptakan desain sistem yang memindahkan logika bisnis dari pihak pengguna ke server. Hasilnya, risiko perubahan data berkurang hingga 95% dan respons sistem menjadi lebih cepat [4]. Penelitian itu menunjukkan bahwa lingkungan runtime Node.js sangat cocok untuk mengelola proses validasi yang dilakukan di satu tempat dengan hasil kerja yang cepat. Sementara itu, Pratama dkk.(2025) dalam penelitiannya dengan judul "Penggunaan State Machine untuk Mengontrol Integritas Alur Kerja pada Aplikasi Web Berbasis SQLite" menyimpulkan bahwa penerapan state machine dapat mencegah perubahan status data yang tidak valid, sehingga menjaga konsistensi data dalam database relasional yang ringan secara optimal [5].

Dari kedua penelitian tersebut, dapat disimpulkan bahwa menggabungkan arsitektur Node.js dengan penerapan state machine adalah cara yang sangat cocok untuk mengatasi masalah keamanan pada sistem yang rentan di sisi klien. Node.js bersama dengan kerangka kerja Express.js mempermudah pembuatan API RESTful yang cepat dan bekerja secara asinkron, sedangkan penggunaan SQLite memberikan cara penyimpanan data berbasis relasi yang sederhana namun cukup kuat tanpa perlu mengatur server basis data yang rumit [6]. Kelebihan-kelebihan tersebut membuat teknologi tersebut menjadi pilihan yang tepat untuk mengumpulkan logika pengundian dan memperkuat keamanan data.

Berdasarkan hal tersebut, penelitian ini mengembangkan evolusi arsitektur untuk sistem pengundian Hudatama, yang sebelumnya berjalan sepenuhnya sebagai aplikasi berbasis JavaScript dan penyimpanan lokal di sisi klien. Pembaruan ini berfokus pada pengembangan infrastruktur backend dengan menggunakan Node.js, Express, dan SQLite agar proses validasi dapat dijalankan secara terpusat. Sistem ini menerapkan aturan pemeriksaan yang ketat, seperti pengujian format nomor peserta yang menggunakan angka acak sebanyak 7 digit untuk memastikan data pool tetap asli. Penelitian ini bertujuan membuat sistem pengundian yang berpusat dan bisa berjalan dengan baik, menggunakan state machine untuk mengurus proses undian, serta lebih terlindungi dari tindakan manipulasi oleh pihak luar [7].

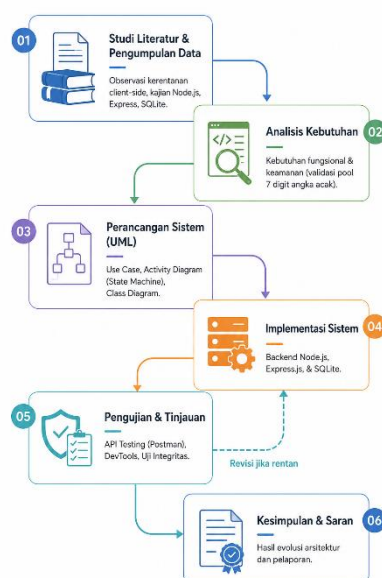
2. Metode

Penelitian ini bertujuan untuk membangun sistem backend yang terpusat pada aplikasi pengundian elektronik Hudatama, dengan menggunakan lingkungan Node.js, kerangka kerja Express, dan basis data SQLite, sebagai alternatif dari pendekatan pengembangan sepenuhnya di sisi klien. Pengembangan perangkat lunak ini menggunakan metode Agile karena bersifat bertahap dan bertambah secara perlahan, sehingga mampu menyesuaikan perubahan struktur sistem dan kebutuhan keamanan yang sering terjadi selama proses pindah sistem berlangsung [8].



Gambar 2.1 Siklus Metode Pengembangan Sistem Agile

Metode Agile digunakan sebagai pendekatan dalam mengembangkan struktur sistem secara bertahap pada penelitian ini. Berbeda dengan model Waterfall yang berjalan secara berurutan, Agile menggunakan siklus pengembangan yang berulang-ulang, yang terdiri dari enam tahapan, yaitu Persyaratan, Desain, Pengembangan, Pengujian, Pemantauan, dan Peninjauan. Keenam tahapan tersebut diulang secara berulang (iteratif) untuk setiap modul keamanan dan fitur logika pengundian yang dikerjakan, sehingga setiap implementasi state machine maupun validasi API dapat langsung dievaluasi dan diperbaiki sebelum pengembangan melanjutkan ke fitur berikutnya [8].



Gambar 2.2 Tahapan Penelitian

Gambar 2.2 menunjukkan langkah-langkah penelitian yang dilakukan secara menyeluruh dalam perkembangan sistem pengundian Hudatama. Penelitian dimulai dengan membaca berbagai referensi mengenai keamanan di sisi pengguna (client-side security), penggunaan State Machine, Node.js, dan desain API RESTful. Setelah itu, dilakukan pengumpulan data dengan cara mengamati kelemahan-kelemahan dalam sistem yang sedang beroperasi, seperti penggunaan localStorage dan eksekusi skrip JavaScript lokal. Selanjutnya dilakukan analisis kebutuhan fungsional dan keamanan, kemudian dirancang arsitektur yang terpusat menggunakan Unified Modeling Language (UML), yaitu dengan membuat diagram use case, diagram aktivitas untuk menggambarkan alur state machine, serta diagram kelas untuk mendesain skema basis data SQLite. Implementasi sistem dilakukan dengan membangun bagian belakang (backend) menggunakan Node.js dan Express, kemudian dilanjutkan dengan menguji integrasi menggunakan Postman dan Chrome DevTools, serta melakukan peninjauan (review) mengenai keamanan sistem. Jika hasil pengujian masih menunjukkan kemungkinan penyimpangan dalam data, maka dilakukan penyesuaian pada tahap penerapan logika server hingga sistem dianggap aman dan tidak bocor. Setelah itu, disusunlah kesimpulan dan saran sebagai hasil akhir dari penelitian tersebut.

a. Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan untuk mengidentifikasi celah keamanan dalam arsitektur lama dan menentukan spesifikasi sistem di sisi server yang akan dikembangkan, berdasarkan hasil pengamatan dan penilaian terhadap integritas data pengundian. Hasil analisis menunjukkan bahwa sistem perlu bisa memindahkan tugas pengolahan data peserta ke bagian belakang sistem, memberikan catatan aktivitas yang jelas dan terbuka, mengatur status pemenang seperti pending, valid, atau tidak valid secara terpusat, serta menerapkan aturan pemeriksaan yang ketat. Aturan validasi ini melibatkan pengecekan keaslian data dengan mewajibkan nomor peserta dalam bentuk angka acak sebanyak 7 digit, agar memastikan pool tidak mengandung duplikasi dan format yang tidak valid sebelum proses pengundian dimulai.

b. Perancangan Sistem

Proses perancangan sistem dilakukan agar bisa menampilkan struktur basis data serta alur logika kontrol sebelum memasuki tahap penulisan kode. Perancangan menggunakan Bahasa Pemodelan Terpadu (UML) yang mencakup diagram Use Case untuk menggambarkan interaksi administrator dan operator terhadap sistem baru, diagram Aktivitas yang menerapkan model State Machine untuk menunjukkan perpindahan status pengundian yang valid, serta diagram Kelas untuk merancang struktur hubungan antar entitas SQLite seperti Daftar Peserta, Riwayat Pemenang, dan Catatan Audit [9].

c. Implementasi Sistem

Implementasi dilakukan dengan menerjemahkan hasil desain arsitektur keamanan ke dalam lingkungan backend menggunakan Node.js dan Express.js. Node.js dipilih karena kemampuannya dalam menangani permintaan asinkron (non-blocking I/O) dengan performa yang sangat tinggi, hal ini sangat penting saat memilih pool data peserta secara real-time [10]. Basis data SQLite diimplementasikan untuk menggantikan localStorage, memberikan penyimpanan data berbasis relasi yang ringan namun tetap aman dan terisolasi di sisi server. Setiap langkah dalam proses undian diatur melalui RESTful API dengan menggunakan logika mesin state, di mana sistem melakukan validasi dan memisahkan tahapan seperti inisialisasi, pengacakan yang aman, serta penentuan status secara modul untuk mencegah penyalahgunaan antarmuka klien.

d. Pengujian Sistem

Pengujian dilakukan secara menyeluruh menggunakan software API Client seperti Postman untuk memverifikasi respons dari endpoint Node.js, serta alat developer pada browser untuk memastikan bagian sistem klien (frontend) tidak lagi menyembunyikan logika rahasia atau seluruh pool peserta. Pengujian dilakukan secara khusus dengan fokus pada pengujian sederhana berupa penetrasi logika, seperti percobaan pengiriman transisi status

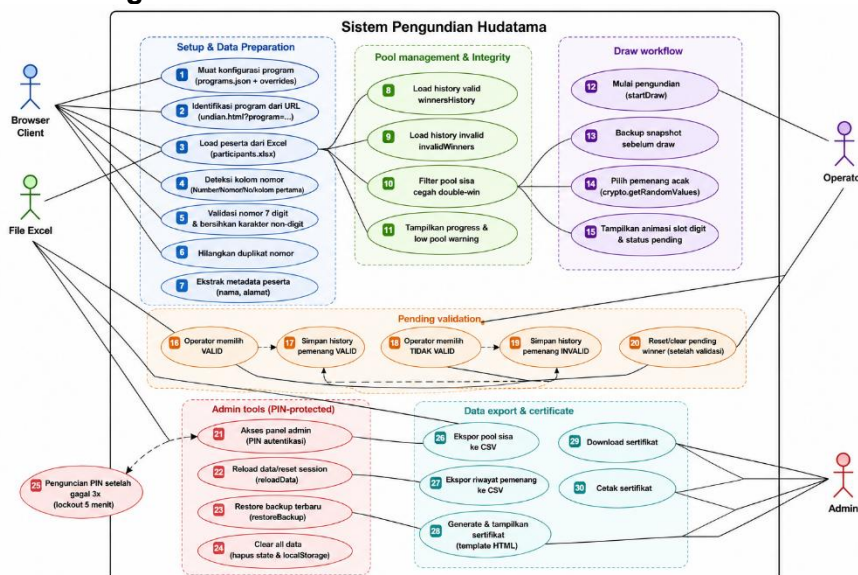
yang tidak valid (misalnya mengubah status pemenang tanpa melewati proses pending) serta pengujian batas (rate limiting) pada mekanisme autentikasi PIN admin. Jika ditemukan respons peladen yang tidak konsisten atau celah yang bisa dimanipulasi, perbaikan akan dilakukan segera pada routing Express dan pengujian akan diulang hingga integritas seluruh proses pengundian terjamin.

3. Hasil dan Pembahasan

3.1. Penyajian Hasil

Sistem Pengundian Hudatama dibuat agar undian bisa dijalankan dengan adil, acak, dan diawasi sepenuhnya dari sisi peramban. Ada dua peran utama dalam sistem ini, yaitu Admin dan Operator, yang memiliki hak mengelola modul pengaturan, mengelola kelompok peserta, menjalankan proses pengundian, serta mengurus penyimpanan data operasional.

3.1.1 UseCase Diagram

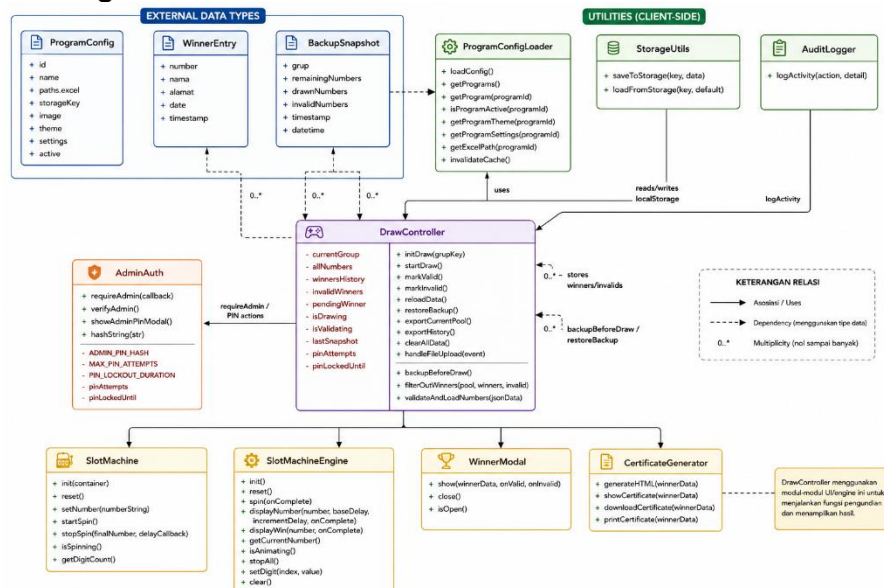


Gambar 3.1 UseCase Diagram Sistem Pengundian Hudatama

Gambar 3.1 menunjukkan cara Admin dan Operator berinteraksi, serta entitas sistem dan sumber data dari luar, dengan semua fitur yang ada dalam Sistem Pengundian Hudatama. Diagram ini menunjukkan sistem memiliki 30 use case utama yang meliputi semua aspek dalam proses pengundian, mulai dari persiapan data hingga penyampaian laporan hasil. Pada tahap persiapan, sistem memproses data dengan menggunakan use case Muat konfigurasi program dan Identifikasi program dari URL, kemudian berinteraksi dengan File Excel untuk melakukan Load peserta, mendeteksi kolom nomor, memvalidasi nomor 7 digit serta membersihkan karakter yang bukan angka, menghapus nomor yang berulang, serta mengekstrak metadata peserta. Dalam hal manajemen integritas, sistem secara otomatis memastikan pool tetap valid dengan memeriksa riwayat load yang sah dan tidak sah, menyaring pool yang tersisa agar tidak ada double-win, serta menampilkan progres dan peringatan untuk pool yang terlalu rendah. Dalam hal pelaksanaan pengundian, Operator berperan sebagai aktor utama yang menjalankan use case Mulai pengundian. Operator secara otomatis memicu sistem untuk membuat backup snapshot sebelum pengundian, memilih pemenang secara acak, serta menampilkan animasi slot dan status pending. Operator kemudian memeriksa dengan menggunakan kasus penggunaan. Operator memilih opsi VALID atau opsi TIDAK VALID, yang akan menyimpan riwayat tersebut dan berakhir dengan perintah Reset atau Clear pending winner. Selain itu, Admin memiliki kontrol penuh dalam mengelola sistem melalui fitur Akses panel admin yang dilindungi dengan Penguncian PIN setelah gagal mencoba login sebanyak 3 kali. Fungsi ini memungkinkan Admin melakukan Reload data atau reset session, mengembalikan sistem ke backup terbaru, serta menghapus semua data.

Sebagai penutup, sistem mendukung kebutuhan pengelolaan dokumen dan laporan hasil dengan menyediakan fitur seperti ekspor data sisa ke CSV, ekspor riwayat pemenang ke CSV, serta fungsi generate, unduh, dan cetak sertifikat. Dengan demikian, admin dapat mengelola seluruh hasil pengundian secara lengkap melalui satu antarmuka yang terpusat.

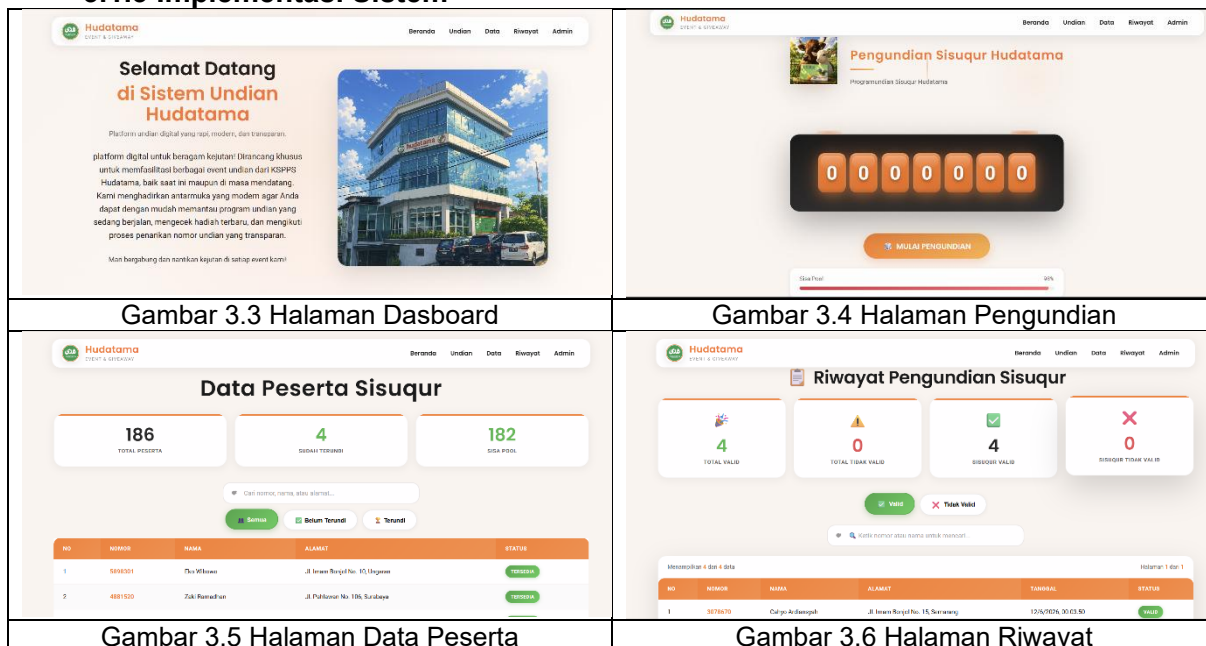
3.1.2 Class Diagram

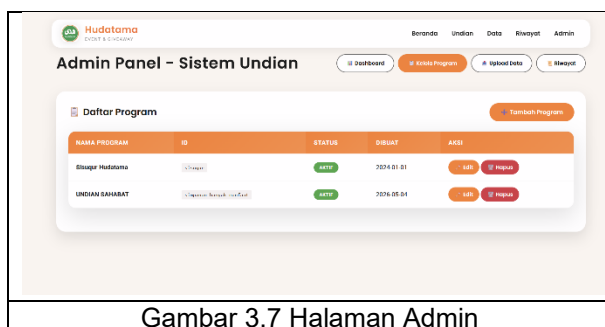


Gambar 3.2 Class Diagram Sistem Undian Hudatama

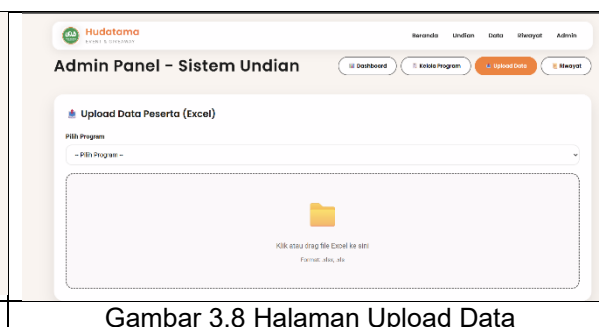
Gambar 3.2 menunjukkan bagian-bagian data dan hubungan antar elemen pada sistem tersebut. Entitas DrawController berfungsi sebagai entitas utama yang terkait dengan AdminAuth, ProgramConfigLoader, StorageUtils, dan AuditLogger, serta memiliki ketergantungan terhadap ProgramConfig, WinnerEntry, dan BackupSnapshot. DrawController juga menggunakan modul antarmuka yang terkait dengan SlotMachine, SlotMachineEngine, dan WinnerModal, sedangkan CertificateGenerator berhubungan dengan data pemenang yang mencatat proses pembuatan dan pencetakan sertifikat.

3.1.3 Implementasi Sistem





Gambar 3.7 Halaman Admin



Gambar 3.8 Halaman Upload Data

Gambar 3.3 menunjukkan halaman dashboard yang menampilkan antarmuka utama platform undian digital secara real-time. Gambar 3.4 menampilkan halaman pengundian yang berisi animasi mesin slot angka serta indikator jumlah kredit yang tersisa saat proses penarikan sedang berlangsung. Gambar 3.5 menunjukkan halaman data peserta yang digunakan untuk memantau total peserta, jumlah peserta yang sudah diundi, sisa kuota, serta tabel detail status dan informasi identitas peserta. Gambar 3.6 menampilkan halaman riwayat pengundian yang mencatat daftar pemenang dengan status sah atau tidak sah, serta dilengkapi dengan fitur pencarian dan tabel yang menjelaskan waktu pengesahan. Gambar 3.7 menunjukkan halaman panel admin yang berisi daftar program undian untuk mengelola program tersebut, sedangkan Gambar 3.8 menampilkan halaman unggah data yang memudahkan input data mentah peserta melalui file Excel. Seluruh 30 use case utama yang ditargetkan berhasil dibuat, berjalan sesuai dengan rencana evolusi arsitektur, serta terhubung secara langsung dengan backend yang menggunakan Node.js, Express, dan database SQLite melalui RESTful API.

3.2. Pembahasan

Penelitian ini menunjukkan bahwa evolusi arsitektur Sistem Pengundian Hudatama berhasil dilakukan dengan memindahkan pusat logika dari bagian klien ke infrastruktur backend yang terpusat, menggunakan teknologi Node.js, Express.js, dan SQLite. Keputusan untuk menggunakan arsitektur server-side ini terbukti sangat berhasil dalam memperbaiki kelemahan keamanan struktur sistem sebelumnya yang memanfaatkan localStorage. Meskipun pembaruan ini membutuhkan komunikasi jaringan melalui RESTful API, penggunaan Node.js yang bekerja secara asinkron (non-blocking) tetap memastikan proses mengambil peserta dari pool dan menjalankan pengundian berjalan cepat dan responsif.

Penerapan konsep mesin state di sisi server menjadi dasar utama dalam mengatur integritas proses pengundian. Pemisahan proses pengacakan menggunakan random aman dan tahap konfirmasi melalui mekanisme pending winner yang terbukti memberikan ruang kontrol operasional bagi Operator untuk melakukan verifikasi langsung tanpa mengungkapkan logika validasi kepada pengguna. Dengan proses pemfilteran keaslian data, seperti wajib menggunakan format angka acak 7 digit dan membatasi riwayat pemenang agar tidak ada kemungkinan menang dua kali, yang sekarang dijalankan secara terpusat di bagian backend, sistem bisa memastikan bahwa status peserta selalu tetap akurat dan tidak bisa diubah melalui alat pengembang di browser. Pemindahan data persisten ke dalam basis data relasional SQLite berhasil membuat satu sumber kebenaran yang kuat dan stabil.

Dari segi keamanan administratif, kerentanan yang sebelumnya muncul karena proses verifikasi PIN dan penyimpanan hash dilakukan di sisi pengguna telah berhasil diminimalkan. Semua aksi sensitif seperti mengisi ulang data, mengembalikan cadangan, dan menghapus semua data kini telah diverifikasi dengan aman oleh server, sehingga kemampuan sistem untuk diperiksa atau diawasi (auditabilitas) meningkat secara signifikan. Ekspor data operasional seperti file CSV dan cetakan sertifikat HTML juga terintegrasi dengan baik, menggunakan data historis yang diambil secara real-time dari API.

Meskipun perkembangan arsitektur ini sudah mampu menjaga keutuhan data dan melindungi sistem dari tindakan yang tidak sah dari pihak luar, masih ada beberapa hal yang bisa dikembangkan lebih lanjut. Mekanisme autentikasi yang

digunakan bisa diperbaiki lagi dengan menerapkan pendekatan berbasis JSON Web Token (JWT) sehingga pengelolaan sesi dari pengguna menjadi lebih aman daripada metode verifikasi PIN yang biasa digunakan. Selain itu, fitur sertifikat pemenang yang ada saat ini masih menggunakan kode QR sebagai placeholder; pada pengembangan berikutnya, kode QR ini harus dihubungkan secara aman dengan database historis di server agar keaslian sertifikat fisik bisa diperiksa oleh siapa saja. Secara keseluruhan, sistem baru ini telah mencapai tingkat keamanan dan kemampuan fungsional yang jauh lebih baik dibandingkan versi sebelumnya. Beberapa tips untuk membuat diskusi tentang naskah:

4. Kesimpulan

Penelitian ini berhasil menciptakan dan meningkatkan sistem pengundian Hudatama dengan memindahkan proses logika dari sisi pengguna ke bagian belakang sistem yang terpusat, menggunakan Node.js, Express, dan database SQLite. Sistem ini bisa menggabungkan semua keperluan dalam proses pengundian ke dalam satu platform yang aman. Fungsi yang termasuk adalah pengaturan program, mengelola peserta dalam *pool*, memvalidasi angka acak 7 digit, melakukan pengundian dengan metode *secure random*, mencatat riwayat pemenang dengan cara *state machine*, mengelola cadangan data (*backup*), serta menghasilkan laporan dan sertifikat. Pengembangan dilakukan dengan metode Agile yang berlangsung secara bertahap, sehingga setiap bagian keamanan dan fitur pengundian bisa dicek dan diperbaiki terus-menerus agar masalah kelemahan dari sistem sebelumnya bisa diatasi. Hasil pengujian menunjukkan bahwa semua 30 skenario pengguna yang ditentukan berhasil diimplementasikan dan berjalan sesuai dengan kebutuhan yang telah ditetapkan, dengan tingkat perlindungan terhadap integritas data yang jauh lebih baik dibandingkan upaya intervensi dari pihak luar.

Penelitian menunjukkan bahwa menggunakan lingkungan Node.js, basis data SQLite yang berbasis relasi, serta metode Agile terbukti efektif dalam meningkatkan sistem pengundian elektronik. Hasil ini juga mencakup peningkatan performa dalam menangani data serta perbaikan terhadap keamanan arsitektur sistem. Dengan mengintegrasikan *state machine*, validasi alur pengundian dilakukan secara terpusat, sehingga meningkatkan kemampuan audit sistem karena mencegah terjadinya perubahan status pemenang yang tidak sah. Pengembangan selanjutnya bisa menambahkan mekanisme autentikasi menggunakan JSON Web Token (JWT) sebagai pengganti PIN biasa, mengoptimalkan pemuatan data peserta dalam jumlah besar dengan metode *lazy loading*, serta melekatkan kode QR yang sudah valid pada sertifikat pemenang agar sistem menjadi lebih berkualitas dan terpercaya..

5. Referensi

- [1]. Gupta A, Sharma R. Cybersecurity threats in client-side web applications: A comprehensive review. *Journal of Information Security and Applications*. 2023;65:102-115.
- [2]. Johnson M, Lee K. Vulnerabilities of local storage and client-side state management in modern web architectures. *International Journal of Web Engineering and Technology*. 2022;17(4):45-58.
- [3]. Williams R, Chen H. Server-side validation and its impact on data integrity in electronic transaction systems. *IEEE Transactions on Dependable and Secure Computing*. 2023;20(1):88-102.
- [4]. Nugroho A, Santoso B, Wibowo S. Implementasi Node.js dan Express pada backend sistem validasi data transaksi terpusat. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*. 2024;8(1):210-218.
- [5]. Pratama D, Hidayat R, Kusuma W. Penggunaan state machine untuk mengontrol

integritas alur kerja pada aplikasi web berbasis SQLite. Jurnal JTİK (Jurnal Teknologi Informasi dan Komunikasi). 2025;9(2):150-159.

- [6]. Brown T, Davis L. Building scalable RESTful APIs with Node.js and Express. 2nd ed. New York: Tech Press; 2023.
- [7]. Garcia C, Martinez P. Securing electronic drawing and lottery systems against external manipulation. Journal of Cybersecurity and Privacy. 2023;3(3):301-315.
- [8]. Sutherland J, Schwaber K. Agile software development: Principles, patterns, and practices. 3rd ed. Boston: Pearson Education; 2022.
- [9]. Booch G, Rumbaugh J, Jacobson I. The Unified Modeling Language user guide: System design and architecture. 3rd ed. Boston: Addison-Wesley Professional; 2021.
- [10]. Tilkov S, Vinoski S. Node.js: Using asynchronous JavaScript to build high-performance network programs. IEEE Internet Computing. 2022;14(6):80-83.