



Mitigasi Double Booking pada Penjadwalan Dermaga Melalui Sistem Visualisasi Spasial Real-Time

Dimas Rangga Marshandika, Gede Humaswara Prathama, Adie Wahyudi Oktavia Gama, Md. Wira Putra Dananjaya

Teknologi Informasi, Teknik dan Informatika, Universitas Pendidikan Nasional
dimasrangga620@gmail.com, huma@undiknas.ac.id

Abstrak

Pelabuhan Benoa sebagai salah satu simpul penting logistik maritim menghadapi tantangan operasional berupa latensi informasi dan potensi tumpang tindih jadwal (double booking) pada proses pra-penjadwalan sandar kapal yang masih dikelola secara konvensional. Penelitian ini bertujuan untuk merancang dan membangun sebuah sistem Pra-Booking dan monitoring dermaga berbasis website yang dilengkapi dengan visualisasi Berthing Plan secara real-time. Pembangunan perangkat lunak ini menggunakan metode Prototyping dengan memanfaatkan kerangka kerja React.js berarsitektur Virtual DOM pada sisi antarmuka untuk merender tata letak grafik dua dimensi secara responsif. Pada sisi server, sistem menggunakan Node.js dan pustaka Socket.io untuk mengelola transmisi data berlatensi rendah. Basis data PostgreSQL digunakan untuk mengamankan integritas transaksional melalui penerapan mekanisme penguncian baris data (row-level locking). Logika pencegahan benturan jadwal dikelola melalui algoritma Bounding Box Collision yang mengevaluasi irisan spasial dan temporal secara simultan, serta didukung oleh fitur mitigasi keterlambatan operasional (Extend Time). Hasil pengujian fungsional menggunakan metode Black Box Testing yang menerapkan teknik Equivalence Partitioning dan Boundary Value Analysis menunjukkan tingkat keberhasilan 100%. Hal ini membuktikan bahwa algoritma validasi kapasitas fisik dermaga dan deteksi konflik beroperasi secara akurat. Selanjutnya, pengujian penerimaan pengguna (User Acceptance Testing) yang melibatkan agen pelayaran menghasilkan persentase rata-rata sebesar 90,63%, menempatkan sistem pada kategori "Sangat Layak". Sistem ini terbukti efektif mereduksi miskomunikasi, menjaga transparansi data, dan siap diimplementasikan untuk mendukung tata kelola operasional di Dermaga Timur Pelabuhan Benoa.

Kata kunci: Pra-Booking, Berthing Plan, Real-Time, WebSocket, PostgreSQL, User Acceptance Testing.

1. Pendahuluan

Indonesia sebagai negara kepulauan menempati posisi strategis dalam jalur perdagangan laut *global*, sehingga keberadaan pelabuhan menjadi elemen krusial dalam menunjang kelancaran sistem logistik nasional. Kinerja infrastruktur maritim berkontribusi secara langsung terhadap peningkatan volume perdagangan dan daya saing ekonomi suatu negara. Oleh karena itu, penerapan transformasi digital melalui konsep *smart port* dinilai cukup mendesak dan dapat memberikan dampak signifikan dalam meningkatkan efisiensi operasional, menekan waktu tunggu kapal, serta memperbaiki kualitas layanan administrasi logistik di kawasan pelabuhan (Azisah et al., 2024). Transformasi digital pada lingkungan pelabuhan juga memerlukan integrasi solusi digital dengan proses operasional agar penerapannya dapat mendukung peningkatan efisiensi dan pengelolaan aktivitas kepelabuhanan (Yang & Hsieh, 2024).

Meskipun digitalisasi menjanjikan berbagai efisiensi, implementasi teknologi pada layanan kepelabuhanan di Indonesia seringkali masih berjalan secara parsial dan menghadapi kendala koordinasi yang serius (Belmoukari et al., 2026). Pada tahapan kritis seperti *Pra-Booking* jadwal sandar kapal, entitas agen pelayaran dan pihak operasional pelabuhan kerap kali masih bergantung pada perputaran dokumen statis dan komunikasi asinkron, seperti panggilan telepon atau aplikasi pesan instan. Penerapan kebijakan *Pre-booking Berth Allocation* pada pengelolaan kedatangan kapal dapat mendukung pengaturan alokasi dermaga dan koordinasi proses *berthing* (Mubder, 2024). Ketergantungan pada koordinasi manual ini secara empiris memicu fenomena keterlambatan pembaruan informasi (*information latency*) yang berdampak pada terjadinya kebutaan spasial dan temporal bagi para pemangku kepentingan (Lee & Lee, 2026).

Proses verifikasi yang memakan waktu lama ini sering kali menyebabkan agen kapal harus menunggu konfirmasi berlapis melalui saluran asinkron hanya untuk mendapatkan kepastian atas satu slot sandar. Ketidadaan *platform* terpusat yang bertindak sebagai sumber kebenaran tunggal (*Single Source of Truth*) membuat pihak agen kesulitan memastikan ketersediaan ruang sandar secara aktual. Ketersediaan informasi yang terintegrasi menjadi salah satu aspek penting dalam penerapan transformasi digital pada operasional pelabuhan karena dapat mendukung pengambilan keputusan dan pengelolaan aktivitas operasional (Lu & Xu, 2024). Ketidakpastian data operasional ini secara langsung melipatgandakan risiko inefisiensi alokasi serta memicu terjadinya tumpang tindih jadwal (*double booking*) di area dermaga. Penentuan posisi dan waktu sandar merupakan bagian penting dalam proses alokasi dermaga karena berkaitan dengan pengaturan penggunaan ruang dan waktu operasional kapal (Rodrigues, 2024). Apabila dibiarkan, penumpukan jadwal tersebut akan bermuara pada peningkatan waktu tunggu kapal (*waiting time*) yang berpotensi merugikan kelancaran rantai pasok logistik maritim secara keseluruhan.

Berbagai penelitian terdahulu telah berupaya merumuskan solusi untuk mengatasi inefisiensi administrasi pelabuhan melalui digitalisasi. Penelitian oleh (Emilia et al., 2025) berhasil menstrukturkan data penjadwalan berbasis *website*, namun luaran antarmuka dari sistem tersebut masih terbatas pada penyajian format tabular konvensional tanpa adanya representasi visual spasial (tata letak dermaga). Sementara itu, (Purwaningtias et al., 2025) membuktikan bahwa sistem informasi penjadwalan digital dapat secara efektif mengurangi ketidakpastian informasi, meskipun sistem tersebut belum memfasilitasi antarmuka negosiasi letak meteran ruang sandar secara dinamis dan interaktif. Di sisi lain, meskipun wacana transformasi digital seringkali diarahkan pada pemasangan sensor perangkat keras murni (*Internet of Things*), pendekatan fisik semacam ini terbukti tidak memiliki kapabilitas komputasional prediktif (*predictive scheduling*). Sensor fisik hanya dapat mendeteksi kapal yang sudah bersandar saat ini, namun tidak dapat memvalidasi proyeksi pengajuan jadwal sandar di masa depan yang objeknya belum berwujud fisik (Zhou et al., 2024).

Menjembatani kesenjangan fungsional dan teknis tersebut, penelitian ini menawarkan kebaruan (*novelty*) melalui pendekatan rekayasa perangkat lunak (*Software Engineering*) yang berfokus pada implementasi visualisasi spasial *Berthing Plan* yang bersifat interaktif dan *real-time*. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun arsitektur komunikasi data berlatensi rendah guna mewujudkan representasi visual tata letak dermaga yang akurat tanpa mewajibkan pemuatan ulang halaman browser. Lebih lanjut, penelitian ini difokuskan untuk merancang algoritma deteksi konflik (*overlap*) pada mesin *backend* guna membantu sistem agar dapat memvalidasi kapasitas fisik dermaga secara presisi demi mencegah anomali pemesanan ganda.

2. Metode Penelitian

Penelitian ini menggunakan metode pendekatan *Prototyping* sebagai kerangka kerja pengembangan sistem. Pemilihan metode ini didasarkan pada kemampuannya untuk mendemonstrasikan purwarupa (*prototype*) visual secara iteratif, sehingga memfasilitasi interaksi langsung antara pengembang dan pihak dermaga guna memastikan keselarasan arsitektur antarmuka dengan model mental petugas operasional di lapangan. Menurut (Pratiwi & Kholil, 2024), penerapan metode *Prototyping* cukup relevan dalam pengembangan sistem monitoring berbasis *website* karena memungkinkan identifikasi dan perbaikan alur kerja secara dinamis melalui umpan balik pengguna sebelum tahapan penulisan kode program (*coding*) dieksekusi secara penuh. Dalam konteks penelitian ini, fokus utama implementasi teknis tidak hanya berpusat pada akurasi representasi visual tata letak dermaga (*Berthing Plan*), melainkan juga mencakup penguatan arsitektur *backend* untuk transmisi data *Realtime* dan penanganan konkurensi jadwal yang kompleks.

2.1. Arsitektur Komunikasi *Real-time* dan Visualisasi

Guna memecahkan kendala latensi informasi di lingkungan dermaga, sistem harus meninggalkan metode penarikan data konvensional (*HTTP Short Polling*) yang terbukti memicu overhead pada kinerja *server* jaringan akibat permintaan siklik berulang. Kecepatan transmisi berlatensi rendah ini cukup vital untuk menggantikan proses koordinasi asinkron konvensional antar pemangku kepentingan (Dominguez et al., 2026). Sebagai solusi mutakhir, penelitian ini mengimplementasikan jalur komunikasi *full-duplex* melalui protokol *WebSockets* dengan memanfaatkan pustaka *Socket.io* yang terintegrasi pada lingkungan *server Node.js*. Alur transmisi tingkat lanjut ini memungkinkan *server* untuk secara aktif mendorong (*broadcast*) pembaruan data dalam format *JSON* kepada klien tepat seketika setelah terjadinya mutasi status penjadwalan (Dhawan, 2026; Irsyad & Bhakti, 2025).

Pada sisi representasi visual klien, antarmuka interaktif dibangun secara komprehensif menggunakan pustaka *React.js*. Strategi *Client-Side Rendering* yang didukung oleh arsitektur memori bayangan (*Virtual DOM*) dipilih agar sistem tidak memuat ulang (*refresh*) halaman *website* secara keseluruhan saat menerima aliran data yang tinggi. Berbekal algoritma komparasi (*diffing algorithm*) bawaan kerangka kerja ini yang secara efisien memutakhirkan elemen antarmuka (Hasnain & Ullah, 2023), sistem hanya akan merender serta menggeser blok grafis kapal yang posisi meterannya berubah saja. Pemilihan tumpukan teknologi ini secara empiris efektif dalam mencegah terjadinya anomali *layout thrashing* yang seringkali membekukan (*freeze*) browser *website* pengguna saat memuat elemen visual spasial yang kompleks (Keshari et al., 2023). Alur komunikasi dan transmisi data secara *Realtime* antar komponen sistem ini direpresentasikan melalui *Sequence Diagram* pada Gambar 1.



Gambar 1. Alur Transmisi Data *Realtime* (Sequence Diagram)

Berdasarkan arsitektur yang direpresentasikan pada Gambar 1, proses transmisi komunikasi sistem berjalan secara persisten dan sinkron. Alur dimulai ketika agen mengirimkan formulir pengajuan (*Pra-Booking*) menuju *backend Node.js*. Setelah sistem memvalidasi dan berhasil menyisipkan (*INSERT*) data ke dalam basis data *PostgreSQL*, mesin validasi akan secara seketika menembakkan instruksi pemicu siaran (*Trigger Event socket.emit*). Sinyal pembaruan berformat *JSON* ini akan diterima oleh pendengar asinkron (*socket.on*) di sisi klien *React.js*, yang pada akhirnya memicu perenderan ulang grafik secara instan di layar seluruh pengguna.

2.2. Pemodelan Logika Spasial Dermaga

Penyusunan algoritma validasi *Berthing Plan* didasarkan pada ekstraksi parameter operasional fisik di Dermaga Timur Pelabuhan Benoa. Parameter fisik ini, yang meliputi panjang standar kapal dan jarak aman wajib antar kapal di dermaga, dikonversi menjadi konstrain logika matematis. Pemodelan ini merupakan fondasi utama sistem guna mencegah entitas agen kapal mengajukan jadwal di luar kapasitas lapangan yang riil.

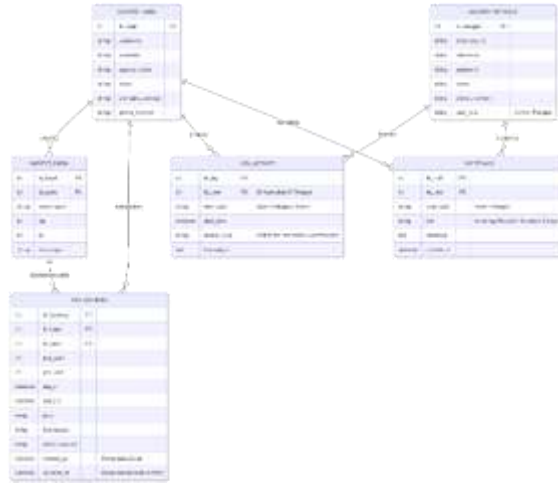
Dalam implementasinya, sistem menghitung total kebutuhan ruang sandar kapal secara komputasional dan dinamis. Titik meteran akhir dari sebuah kapal diperoleh melalui rumusan matematis pada persamaan 1.

$$POS_{END} = POS_{START} + LOA + CLEARANCE \quad (1)$$

dengan *POS_START* adalah titik meteran awal kapal yang diajukan oleh agen, *LOA* adalah panjang fisik kapal secara keseluruhan (*Length Overall*), dan *CLEARANCE* adalah jarak aman antar kapal yang diwajibkan oleh pelabuhan. Sistem kemudian memvalidasi nilai hasil perhitungan ini terhadap konstanta panjang maksimal operasional dermaga sebesar 500 meter.

2.3. Pemodelan Skema Basis Data

Perancangan struktur penyimpanan data operasional direpresentasikan melalui *Entity Relationship Diagram* (ERD) guna memetakan relasi entitas dan memastikan integritas data. Arsitektur basis data relasional ini dibangun menggunakan sistem manajemen *PostgreSQL*. Model data dirancang berpusat pada entitas transaksional utama, yaitu tabel *trx_booking*, yang memiliki relasi ketergantungan (*Foreign Key*) terhadap tabel master_kapal dan master_agen. Diagram pemodelan skema relasi ini diilustrasikan secara komprehensif pada Gambar 2.



Gambar 2 Pemodelan Skema Basis Data (*Entity Relationship Diagram*)

Berdasarkan Gambar 2, skema basis data dibagi menjadi beberapa kelompok fungsional. Model data dirancang berpusat pada entitas transaksional utama, yaitu tabel *trx_booking*, yang secara khusus dikonfigurasi untuk merekam parameter spasial (*pos_start*, *pos_end*) dan parameter temporal (*eta_in*, *ETD_OUT*) penjadwalan. Tabel inti ini memiliki relasi ketergantungan (*Foreign Key*) secara langsung terhadap kelompok data master yang bersifat statis, yakni tabel *master_kapal* (spesifikasi fisik) dan *master_agen* (identitas perusahaan). Selain itu, untuk mendukung fungsi transparansi dan keamanan otorisasi sistem, *ERD* juga mengintegrasikan tabel *master_petugas*, serta tabel *log_activity* dan notifikasi yang berfungsi vital untuk merekam jejak audit (*audit trail*) dan mendistribusikan peringatan keterlambatan operasional di lapangan.

2.4. Integritas Transaksional Basis Data

Kinerja dan ketahanan sebuah sistem layanan *Pra-Booking* yang terpusat akan menghadapi tantangan terbesarnya ketika diakses oleh puluhan agen pelayaran secara bersamaan. Tantangan konkurensi (pemrosesan data serentak) tersebut dapat memicu terjadinya anomali persaingan data (*race condition*). Kondisi fatal ini terjadi apabila dua agen kapal yang berbeda secara bersamaan menekan tombol pengajuan formulir (*submit*) pada hitungan milidetik yang identik untuk merebut titik meteran dermaga yang saling beririsan.

Sebagai garis pertahanan logis dan mitigasi terakhir untuk mengatasi celah tersebut, arsitektur *backend* mengimplementasikan mekanisme *row-level locking* pada sistem manajemen basis data relasional *PostgreSQL*. Fitur penguncian transaksi tingkat tinggi ini dieksekusi melalui penyematan klausa kueri *FOR UPDATE* di dalam kode program tepat pada saat fase deteksi irisan spasial sedang berjalan di dalam *server*.

Penerapan pendekatan penguncian baris komputasional ini memastikan bahwa antrian pemrosesan di dalam basis data tetap terisolasi dan patuh pada standar kepatuhan isolasi *ACID* (Kalay, 2025). Ketika klausa *FOR UPDATE* dipanggil, mesin *PostgreSQL* mengunci baris data jadwal terkait sehingga tertutup dari intervensi proses lain yang masuk pada waktu bersamaan. Melalui mekanisme ini, basis data memprioritaskan transaksi yang tercatat lebih awal. Instruksi pengajuan lain yang masuk setelahnya akan tertahan (mengantri) pada antrian memori, hingga ditolak (*rejected*) secara otomatis apabila menyebabkan tumpang tindih jadwal. Mekanisme preventif ini membantu sistem dalam menangani permintaan (*request*) dari agen pelayaran secara konkuren.

2.5. Metode Evaluasi Sistem

Evaluasi ketahanan fungsional sistem diuji menggunakan metode *Black Box Testing* yang berfokus pada kesesuaian antara input pengguna dan *OUT*put algoritma (Darex Susanto & Muhammad Agung Pramono, 2025). Tingkat keberhasilan dari skenario pengujian fungsional tersebut dikalkulasi menggunakan rumusan matematis pada persamaan 2.

$$P = \frac{S_{pass}}{S_{total}} \times 100\% \quad (2)$$

dengan P adalah persentase tingkat kesesuaian sistem, Spass merupakan jumlah skenario pengujian yang berhasil dieksekusi sesuai ekspektasi, dan Stotal adalah total keseluruhan skenario pengujian fungsional yang dilakukan.

Selanjutnya, pengukuran tingkat penerimaan operasional di lapangan dievaluasi melalui metode *User Acceptance Testing* (UAT) menggunakan instrumen kuesioner berskala Likert 5 poin (Risprawati et al., 2024). Rata-rata persentase kelayakan dari pengujian pengguna tersebut dihitung melalui rumusan matematis pada persamaan 3.

$$P_{UAT} = \frac{SkorAktual}{SkorMaksimal} \times 100\% \quad (3)$$

dengan P_UAT adalah persentase tingkat penerimaan pengguna, Skor Aktual merupakan total nilai yang diberikan oleh seluruh responden pada setiap pernyataan, dan Skor Maksimal merupakan batas nilai tertinggi yang dapat dicapai dalam kuesioner tersebut.

3. Hasil dan Pembahasan

Implementasi sistem *Pra-Booking* dan monitoring dermaga ini direalisasikan dengan mengintegrasikan dua pilar utama arsitektur perangkat lunak yaitu keandalan mesin validasi logika di sisi *server* (*backend*) dan responsivitas representasi spasial pada antarmuka pengguna (*frontend*). Transformasi dari pencatatan konvensional menuju *platform* digital interaktif ini terbukti secara efektif membentuk sebuah sumber kebenaran tunggal (*Single Source of Truth*), yang secara langsung mereduksi latensi informasi dan meningkatkan transparansi negosiasi antar entitas operasional di pelabuhan. Rangkaian rincian penjabaran mengenai hasil implementasi algoritma validasi, visualisasi tata letak spasial, integrasi fitur proaktif, hingga evaluasi ketahanan sistem dan tingkat penerimaan pengguna (*User Acceptance Testing*) diuraikan secara komprehensif sebagai berikut.

3.1. Implementasi Logika Validasi Spasial dan Temporal

Pencegahan tumpang tindih jadwal dieksekusi melalui algoritma pendeteksi tabrakan ruang secara logis (*Bounding Box Collision*). Konflik pada *Berthing Plan* akan dideklarasikan oleh sistem apabila terdapat irisan secara spasial dan temporal secara bersamaan pada data pengajuan yang baru dengan data jadwal yang sudah berstatus disetujui (*Approved*). Irisan spasial dievaluasi dengan membandingkan rentang titik awal (*POS_START*) hingga titik akhir (*POS_END*) dari kapal yang diajukan terhadap batas meteran kapal eksisting. Bersamaan dengan itu, irisan temporal divalidasi pada parameter waktu kedatangan (*ETA*) dan waktu keberangkatan (*ETD*). Apabila kedua irisan ini terpenuhi secara simultan, algoritma mengidentifikasi pengajuan tersebut sebagai konflik dan menolak masukan untuk memitigasi tumpang tindih pada satu titik koordinat meteran dermaga.

Secara teknis pada arsitektur basis data, mekanisme penyaringan awal dieksekusi secara ketat melalui klausa kueri berbasis parameter waktu (*eta_in* dan *ETD_OUT*). Apabila sistem mendeteksi adanya pelanggaran pada parameter tersebut, mesin validasi di *backend* akan secara otomatis memblokir transaksi pengajuan sebelum menyentuh basis data utama.



Gambar 3 Respons Sistem saat Memblokir Indikasi *Double booking*

Sebagaimana direpresentasikan pada Gambar 3, efektivitas logika algoritma ini terbukti saat sistem merespons masukan yang saling beririsan. Antarmuka langsung menampilkan jendela peringatan interaktif di layar agen kapal, menolak pengajuan tersebut secara tegas. Penolakan ini memuat rincian identitas operasional kapal eksisting yang memicu konflik (misal: MV Scorpio 68), sehingga agen dapat segera merumuskan langkah mitigasi dengan merevisi target waktu maupun koordinat meteran mereka.

Keberhasilan logika validasi yang didukung oleh mekanisme penguncian baris (*row-level locking*) pada basis data *PostgreSQL* ini terbukti dapat mencegah tumpang tindih jadwal. Hasil ini mendukung temuan Kalay (2025) yang menyatakan bahwa penerapan standar *ACID* dan *Multiversion Concurrency Control (MVCC)* merupakan pendekatan arsitektur yang memadai untuk menangani persaingan data (*race condition*) dalam pemrosesan transaksi operasional yang bersamaan.

3.2. Implementasi Visualisasi *Berthing Plan* Interaktif

Pada sisi antarmuka pengguna, komponen utama yang direalisasikan adalah kanvas pemantauan *Berthing Plan*. Kanvas visual ini menampilkan tata letak dua dimensi yang merender alokasi posisi sandar kapal dalam bentuk blok grafis dinamis. Blok-blok tersebut diproyeksikan secara presisi di atas representasi skala meteran dermaga (0–500 meter) beserta indikator fasilitas fisik pelabuhan seperti letak *fender* dan *fender*.

Tata letak ini didesain responsif dengan mekanisme horizontal *scroll* agar visualisasi tetap presisi di berbagai resolusi layar. Selain itu, sinkronisasi pembaruan warna dan posisi grafik pada antarmuka ini sepenuhnya digerakkan oleh transmisi data berlatensi rendah.



Gambar 4 Antarmuka Utama Pra Booking dan Monitoring Dermaga

Berdasarkan hasil implementasi pada Gambar 4 dan pengujian sinkronisasi antar sesi pengguna, arsitektur komunikasi data *Socket.io* terbukti efisien mereduksi latensi. Sesaat setelah petugas menekan tombol persetujuan (*Approve*) pada panel tabel, *server* langsung mengeksekusi penyiaran pesan. Sinyal pembaruan ini ditangkap oleh *React.js* pada layar *browser* agen, yang kemudian warna blok grafis kapal diubah menjadi hijau secara instan, tanpa mengharuskan pihak agen untuk memuat ulang halamannya secara manual.

Implementasi transmisi *full-duplex* yang memungkinkan pembaruan antarmuka tanpa pemuatan ulang (*refresh*) ini sejalan dengan penelitian Irsyad & Bhakti (2025) serta Dhawan (2026). Hasil pengujian mengonfirmasi bahwa penggunaan protokol *WebSocket* melalui pustaka *Socket.io* pada lingkungan *Node.js* dapat menekan latensi transmisi, sehingga alur informasi tata letak dermaga dapat disinkronkan secara *real-time*.

Optimalisasi perenderan grafik pada sisi klien (*frontend*) juga didukung oleh efisiensi manajemen state (kondisi komponen) di dalam *React.js*. Ketika protokol *WebSocket* menerima pembaruan dari *server*, data yang ditransmisikan hanya berupa paket *JSON (JavaScript Object Notation)* berukuran kecil yang berisi parameter spesifik perubahan, seperti *ID* kapal dan status persetujuan terbaru. Kerangka kerja *React.js* kemudian memproses muatan (*payload*) *JSON* tersebut menggunakan arsitektur *Virtual DOM* untuk mengidentifikasi komponen visual mana yang memerlukan pembaruan. Proses transmisi asinkron terarah ini secara signifikan mereduksi beban pertukaran data secara keseluruhan, sehingga akses pemantauan *Berthing Plan* tetap dapat berjalan lancar meskipun diakses menggunakan koneksi jaringan seluler dengan *bandwidth* yang terbatas di area lapangan dermaga.

3.3. Implementasi Fitur Mitigasi Keterlambatan (Extend Time)

Siklus operasional yang bergulir di area pelabuhan seringkali sangat dinamis dan berpotensi memicu berbagai gangguan jadwal secara tidak terduga, seperti keterlambatan manuver tongkang maupun kendala teknis pada proses bongkar muat kargo yang berisiko menyebabkan kapal melampaui rentang jadwal keberangkatan awal yang telah ditetapkan (Purwaningtias et al., 2025). Guna memecahkan ancaman ketidakpastian administratif tersebut, perangkat lunak *Pra-Booking* ini didesain dengan menyematkan sebuah fitur pemecahan masalah yang disebut mitigasi keterlambatan proaktif (*Extend Time*).

Struktur arsitektur sistem dikonfigurasi dengan menanamkan mesin fungsi penjadwalan interval latar belakang (*Scheduler*) yang bekerja secara kontinyu untuk mendeteksi sisa durasi waktu sandar seluruh kapal yang berstatus aktif. Ketika kalkulasi mesin mendeteksi bahwa waktu operasional sebuah kapal akan habis dalam kurun waktu tepat satu jam ke depan, sistem secara proaktif memancarkan notifikasi peringatan (*alert*) otomatis kepada agen yang bersangkutan. Agar eksekusi *scheduler* di latar belakang ini tidak membebani kapasitas komputasi *server*, fungsi pemantauan diatur untuk hanya melakukan eksekusi *query* pada baris data jadwal yang memiliki atribut status aktif di hari berjalan. Pendekatan penyaringan berlapis pada *level database* ini mengeliminasi pemrosesan data historis yang tidak relevan, sehingga alokasi memori pada *server* tetap terjaga dalam batas optimal.

Melalui kehadiran modul fungsional inovatif ini, kebingungan koordinasi perpanjangan durasi yang sebelumnya memakan waktu secara administratif dapat diformalkan dengan seketika. Apabila agen terbukti mengajukan perpanjangan, sistem *backend* akan menjalankan ulang algoritma pendeteksi konflik untuk membantu mengurangi anomali *delay cascade* (keterlambatan berantai) yang merugikan jadwal sandar kapal-kapal di antrean selanjutnya pada titik meteran tersebut.



Gambar 5 Antarmuka Notifikasi Peringatan dan Formulir *Extend Time*

Sebagaimana direpresentasikan secara komprehensif pada antarmuka Gambar 5, komponen notifikasi peringatan dini tersebut menyertakan tombol aksi interaktif (*call to action*) yang secara otomatis akan membentangkan formulir perpanjangan (*Extend Time*). Melalui komponen *pop-up* antarmuka ini, agen pelayaran difasilitasi untuk menyesuaikan target waktu jadwal keberangkatan (*ETD/OUT*) baru dengan praktis, tanpa mengharuskan mereka membuang waktu untuk mengisi ulang dan mendaftarkan kembali seluruh parameter data atribut identitas operasional kapal dari tahap awal. Integrasi pembaruan durasi melalui modul ini akan seketika memicu mesin kalkulasi ulang di *database backend* dan kembali mengeksekusi siaran sinyal (*broadcast*) guna merender pemanjangan durasi spasial kapal pada kanvas *Berthing Plan* di layar petugas secara *real-time*.

3.4. Hasil Pengujian Fungsional dan Penerimaan Pengguna

Evaluasi ketahanan fungsional sistem difokuskan secara eksklusif pada modul-modul inti yang menunjang operasional *Berthing Plan*, dengan menggunakan metode *Black Box Testing* yang mencakup teknik *Equivalence Partitioning* dan *Boundary Value Analysis*. Pengujian dilakukan terhadap puluhan skenario uji untuk mensimulasikan percobaan input ekstrem maupun benturan jadwal operasional. Sebagai contoh, penerapan teknik *Boundary Value Analysis (BVA)* dieksekusi dengan menguji batas nilai komputasional dari kapasitas Dermaga Timur. Algoritma diuji menggunakan skenario masukan posisi meteran tepat pada nilai batas bawah (0 meter) dan batas atas (500 meter), serta simulasi nilai di luar toleransi (seperti nilai negatif atau 501 meter). Sistem berhasil memvalidasi seluruh batasan tersebut dengan menolak secara otomatis setiap parameter masukan yang melewati

ambang batas ruang fisik, sehingga meminimalisir potensi galat (error) akibat kesalahan agen kapal. Hasil evaluasi skenario pada modul spasial dan temporal tersebut dirangkum pada Tabel 1.

Tabel 1 Rekapitulasi Hasil Pengujian Fungsional Modul *Berthing Plan*

Modul Fungsional Utama	Fokus Skenario Pengujian (BVA & EP)	Persentase Keberhasilan
Pengajuan <i>Pra-Booking</i>	Batas toleransi nilai minus, batasan kapasitas fisik maksimal (500 meter), dan validasi logika waktu sandar (ETA/ETD).	100% (Pass)
Mesin Deteksi Konflik (<i>Overlap</i>)	Perebutan slot waktu yang sama dan irisan rentang posisi tata letak spasial antar kapal.	100% (Pass)
Transmisi <i>Real-time</i> & Visualisasi	Kecepatan sinkronisasi grafik kapal pada kanvas <i>Berthing Plan</i> saat terjadi perubahan status tanpa muat ulang (refresh).	100% (Pass)
Peringatan Keterlambatan (<i>Extend Time</i>)	Otomatisasi pengiriman notifikasi tenggat waktu sandar dan pengajuan perpanjangan durasi operasional.	100% (Pass)

Mengacu pada Tabel 1, hasil pengujian fungsional secara keseluruhan menghasilkan persentase kesuksesan sebesar 100%. Tingkat kesesuaian 100% yang diperoleh dari pengujian validasi kapasitas dan deteksi konflik jadwal ini mengonfirmasi penelitian Susanto & Pramono (2025). Hasil ini memperkuat bahwa penerapan metode *Black Box Testing* menggunakan teknik *Boundary Value Analysis* dan *Equivalence Partitioning* relevan untuk memvalidasi logika batasan input dan memitigasi kesalahan pada sistem pemesanan berbasis slot.

Selanjutnya, pengujian penerimaan pengguna (*User Acceptance Testing*) dilaksanakan dengan melibatkan pihak eksternal (agen kapal) untuk memvalidasi aspek kemudahan pengoperasian dan utilitas fungsional sistem di lapangan. Pengukuran kuantitatif tingkat kepuasan sistem dilakukan menggunakan instrumen kuesioner Skala Likert pada 19 butir pernyataan. Hasil evaluasi tersebut disajikan secara rinci pada Tabel 2.

Tabel 2 Rekapitulasi Tingkat Penerimaan Pengguna per Aspek

Aspek Penilaian	Rata-Rata Persentase (%)	Kategori Kelayakan
<i>Functional Suitability</i>	90,00%	Sangat Layak
<i>Usability</i>	89,20%	Sangat Layak
<i>Performance & Reliability</i>	91,50%	Sangat Layak
<i>Perceived Usefulness</i>	92,00%	Sangat Layak

Berdasarkan Tabel 2, hasil keseluruhan memperoleh rata-rata persentase penerimaan pengguna yang tinggi, yakni sebesar 90,63%. Capaian tertinggi berada pada aspek utilitas operasional (*Perceived Usefulness*) sebesar 92,00%, yang menunjukkan bahwa responden menilai visualisasi interaktif ini lebih efisien dibandingkan prosedur negosiasi konvensional. Selanjutnya, pada aspek *Performance & Reliability* yang mencatatkan perolehan 91,50%, pengujian menunjukkan bahwa fitur transmisi data berlatensi rendah dapat mendukung pemantauan antarmuka secara aktual. Pada aspek *Functional Suitability* sebesar 90,00% dan *Usability* sebesar 89,20%, hal ini mengindikasikan bahwa fitur yang disediakan dapat dioperasikan dengan baik oleh pengguna. Penilaian ini menunjukkan bahwa penerapan arsitektur *Virtual DOM* pada *React.js* dapat merender pembaruan antarmuka tanpa memicu kendala *layout thrashing* saat memproses pembaruan data spasial.

Penerimaan pengguna yang positif terhadap stabilitas visualisasi tata letak kapal ini turut mendukung temuan Hasnain & Ullah (2023) dan Keshari et al. (2023). Penilaian dari pengguna akhir ini memvalidasi bahwa penerapan

arsitektur *Virtual DOM* pada *React.js* mampu merender pembaruan antarmuka secara efisien tanpa memicu kendala *layout thrashing* saat memproses pergeseran data spasial berfrekuensi tinggi.

4. Kesimpulan

Rangkaian penelitian rekayasa perangkat lunak ini berhasil melahirkan sebuah inovasi perancangan Sistem *Pra-Booking dan Monitoring* Dermaga yang pondasi arsitektural utamanya bertumpu pada kapabilitas pemetaan visual tata letak *Berthing Plan* secara spasial dan interaktif. Sintesis arsitektur teknologi berbasis transmisi komunikasi *full-duplex* dan berlatensi rendah, yang mendayagunakan kombinasi antara protokol *WebSocket (Socket.io)* dengan mesin memori *Virtual DOM* dari kerangka kerja *React.js*, terbukti dapat merender setiap pergeseran grafis kapal dan memperbarui status operasional antarmuka (*frontend*) pengguna dengan mulus (seketika). Inovasi ekosistem transmisi ini meningkatkan efektivitas koordinasi dengan mengeliminasi fenomena latensi informasi dan komunikasi asinkron (manual) yang selama bertahun-tahun merugikan kinerja manajemen operasional pelabuhan di lapangan. Pada sisi pertahanan integritas operasional pangkalan data (*backend*), integrasi solid antara mesin algoritma deteksi irisan logika (spasial dan temporal) dengan penerapan algoritma perlindungan konkurensi *row-level locking* tingkat mutakhir di dalam sistem *PostgreSQL*, secara teknis telah tervalidasi 100% sempurna melalui skenario evaluasi *Black Box Testing*. Validasi ini membuktikan infrastruktur sistem benar-benar kebal dalam mengisolasi dan memusnahkan ancaman masuknya transaksi anomali penjadwalan ganda (*double booking*). Mengacu secara konklusif pada respons positif pengujian partisipan lapangan (*User Acceptance Testing*) yang sukses mendulang persentase rata-rata penerimaan pada skor "Sangat Layak" sebesar 90,63%, luaran produk perangkat lunak ini dinyatakan sepenuhnya memenuhi standar fungsional tinggi yang dibutuhkan entitas pelayaran. Secara holistik, arsitektur pemantauan digital ini diakui siap dan direkomendasikan untuk segera diimplementasikan sebagai sebuah "Sumber Kebenaran Tunggal" dalam menjembatani dan mengakselerasi transparansi koordinasi, sekaligus meningkatkan tata kelola efisiensi logistik di seluruh zona area sandar pelabuhan komersial masa depan.

Reference

1. Azisah, N., Paotonan, C., & Asdar, M. (2024). Literature Review: Konsep Smart Port di Indonesia dengan Metode Analisis Bibliometrik. *Zona Laut Jurnal Inovasi Sains Dan Teknologi Kelautan*, 127–132. <https://doi.org/10.62012/zl.v5i2.31708>
2. Belmoukari, B., Audy, J.-F., Forget, P., & Adam, V. (2026). Smart Port and Digital Transition: A Theory- and Experience-Based Roadmap. *Logistics*, 10(2), 26. <https://doi.org/10.3390/logistics10020026>
3. Darex Susanto & Muhammad Agung Pramono. (2025). Pengujian Black Box Aplikasi Reservasi Lapangan Badminton Menggunakan Teknik Equivalence Partitioning. *JISCO: Journal of Information System and Computing*, 3(1), 35–43. <https://doi.org/10.30631/jisco.v3i1.120>
4. Dhawan, M. (2026). Decoupling Deployment Velocity From Platform Governance: An Empirical Case Study of WebSocket-Enabled Gateway-Microservice Architecture. *IEEE Access*, 14, 9348–9358. <https://doi.org/10.1109/ACCESS.2026.3654150>
5. Dominguez, A., Da Costa Paulo, B., Burguera, I., Tamayo, I., Elosegi, A., Cabrero Barros, S., & Zorrilla, M. (2026). A Web-Ready and 5G-Ready Volumetric Video Streaming Platform: A Platform Prototype and Empirical Study. *IEEE Access*, 14, 34655–34675. <https://doi.org/10.1109/ACCESS.2026.3669714>
6. Emilia, Melisa, S., Ariska, D., & Purwani, F. (2025). Sistem Informasi Penjadwalan Keberangkatan Speedboat Di Pasar 16 Palembang. *Menulis: Jurnal Penelitian Nusantara*, 1(12), 295–304. <https://doi.org/10.59435/menulis.v1i12.780>
7. Hasnain, M., & Ullah, S. (2023). Learning and programming challenges of React.js open-source framework. *Library Hi Tech News*. <https://doi.org/10.1108/LHTN-05-2023-0088>
8. Irsyad, H., & Bhakti, H. D. (2025). PENERAPAN PROTOKOL WEBSOCKET UNTUK SISTEM NOTIFIKASI PERUBAHAN DATA (LIVE UPDATE) PADA APLIKASI WEB E-COMMERCE. *JUSTI (Jurnal Sistem Dan Teknik Industri)*, 6(1), 101–107. (Artikel ini membahas penerapan protokol WebSocket untuk sistem notifikasi perubahan data secara real-time (live update) pada aplikasi web e-commerce. Fokus utama adalah pada peningkatan efisiensi komunikasi antara server dan klien guna mendukung pengalaman).
9. Kalay, M. U. (2025). Concurrency Challenges in Database Systems: A Focus on PostgreSQL. *Bilgisayar Bilimleri ve Teknolojileri Dergisi*, 6(1), 1–16. <https://doi.org/10.54047/bibtet.1574178>
10. Keshari, P., Maurya, P., Kumar, P., & Katiyar, A. (2023). Web Development Using ReactJS. *2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, 1571–1575. <https://doi.org/10.1109/ICAC3N60023.2023.10541743>
11. Lee, C., & Lee, S. (2026). Interrelations Among Data Quality Characteristics in Safety-Relevant Maritime Services: An Expert-Based Study. *Applied Sciences*, 16(4), 2020. <https://doi.org/10.3390/app16042020>
12. Lu, B., & Xu, X. (2024). Digital transformation and port operations: Optimal investment under incomplete information. *Transport Policy*, 151, 134–146. <https://doi.org/10.1016/j.tranpol.2024.04.008>
13. Mubder, A. A. A. M. (2024). The implementation of berth allocation policies that enable Just-in-Time arrival in port calls. *International Journal of Physical Distribution & Logistics Management*, 54(6), 610–630. <https://doi.org/10.1108/IJPDLM-11-2023-0442>
14. Pratiwi, R., & Kholil, I. (2024). Implementasi Model Prototype untuk Perancangan Sistem Informasi Project Monitoring Berbasis Web. *TIN: Terapan Informatika Nusantara*, 5(2), 135–143. <https://doi.org/10.47065/tin.v5i2.5307>
15. Purwaningtiyas, D., Linawati, S., Risdiansyah, D., Latifah, & Lutfiandi, R. (2025). Sistem Informasi Penjadwalan Kedatangan Dan Keberangkatan Kapal (SIJAKA) Berbasis Web. *Reputasi: Jurnal Rekayasa Perangkat Lunak*, 6(1), 74–79. <https://doi.org/10.31294/reputasi.v6i1.3179>

DOI: <https://doi.org/10.69693/ijmst.v4i3.13111>

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

16. Rispawati, D., Arista Pradnyani, I. G. A., Rahayu, N., & Doraq, M. Y. (2024). Analisis Penerapan E-Ticketing Sebagai Upaya Peningkatan Kepuasan Pengguna Jasa PT. ASDP Indonesia Ferry (Persero) Cabang Lembar. *Jurnal Administrasi Dan Manajemen*, 14(3), 319–326. <https://doi.org/10.52643/jam.v14i3.4781>
17. Rodrigues, F. (2024). Improved sequential insertion heuristics for berth allocation problems. *International Transactions in Operational Research*, 31(3), 1585–1608. <https://doi.org/10.1111/itor.13369>
18. Yang, Y.-C., & Hsieh, Y.-H. (2024). The critical success factors of smart port digitalization development in the post-COVID-19 era. *Case Studies on Transport Policy*, 17, 101231. <https://doi.org/10.1016/j.cstp.2024.101231>
19. Zhou, F., Yu, K., Xie, W., Lyu, J., Zheng, Z., & Zhou, S. (2024). Digital Twin-Enabled Smart Maritime Logistics Management in the Context of Industry 5.0. *IEEE Access*, 12, 10920–10931. <https://doi.org/10.1109/ACCESS.2024.3354838>