SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 4, October 2026
DOI : https://doi.org/10.33395/sinkron.v10i4.16866
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Algorithms and Web Architecture for Lightweight Multiplayer Archipelago Monopoly Game Engine

Muhammad Fahrul Bahri[1)]*, Muhayat[2)], Munsyi[3)], Akhmad Hafidz Ardianto[4)], Husni Majedi[5)], Muhammad Rama Alfiannur[6)]

[1,2,3,4,5)]Department of Information Technology, Faculty of Da'wah and Communication Sciences, Universitas Islam Negeri Antasari Banjarmasin, Indonesia

[6)]Department of Information Technology, Faculty of Science and Technology, Universitas Sari Mulia Banjarmasin Indonesia

[1)*]fahrulbahri0520@gmail.com, [2)]muhayat@uin-antasari.ac.id, [3)]munsyi@uin-antasari.ac.id, [4)]akhmad.hafidz.ardianto@gmail.com, [5)]husnimajedi@gmail.com, [6)]ramain027@gmail.com

Submitted : Sep 21, 2026 | **Accepted :** Oct 1, 2026 | **Published :** Oct 8, 2026

Abstract: Browser-based turn-based simulations frequently encounter client-side rendering bottlenecks and transactional race conditions when operating on low-bandwidth networks without dedicated server daemons. This study introduces an optimized web game architecture and programmatic pipeline implementing canonical economic board game rules contextualized within Indonesian geography. The proposed framework integrates four core algorithmic solutions: an atomic advisory file-locking transaction system utilizing flock() for room-level state concurrency control without relational database engines; a category-interleaved deck shuffling algorithm establishing mathematical guarantees against prolonged penalty sequences; an autonomous heuristic decision model that dynamically balances property acquisition costs against risk-adjusted liquidity reserves; and real-time audio synthesis executed entirely via native browser oscillators. Rigorous experimental evaluation spanning computational simulations, ApacheBench load tests, and user testing verified system capabilities across four empirical suites. The card distribution mechanism completely bounds consecutive adverse event occurrences to two draws across two thousand execution cycles while preserving high permutation entropy. In four hundred controlled four-player matches with balanced seat rotation, the autonomous agent attained a thirty-six point eight percent victory frequency, significantly outperforming greedy over-leveraging and naive random baselines, while performing comparably to static threshold strategies. Transactional stress testing confirmed server throughput of three hundred twenty-four state-modifying write operations per second under peak concurrency with zero lost updates and zero data corruption, while client interfaces maintained sixty frames per second rendering. Finally, thirty participants reported a composite system usability score of eighty-four point five, confirming accessible cross-platform gameplay.

Keywords: Algorithm and programming; Deck shuffling algorithm; Heuristic decision model; State concurrency control; Web game architecture

INTRODUCTION

Digital tabletop board games have become an influential medium for interactive computing and education, fusing strategic economic planning with stochastic state transitions. Migrating classic board games to modern web

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 4, October 2026
DOI : https://doi.org/10.33395/sinkron.v10i4.16866
e-ISSN : 2541-2019
p-ISSN : 2541-044X

browsers introduces notable software engineering challenges, particularly regarding runtime performance and server efficiency. Turn-based economic games patterned after Monopoly depend on high-frequency state updates, including land acquisitions, rental payments, dynamic construction upgrades, and probabilistic card events (Herrero et al., 2026; Subiyanto, 2026). However, conventional browser implementations often depend on heavyweight relational database management systems, full-stack enterprise frameworks, and extensive static media files, creating severe execution latency and high bandwidth consumption on constrained networks.

From the viewpoint of algorithm and programming, web-based digital tabletop systems encounter three fundamental computational obstacles. First, conventional uniform pseudo-random number generator routines applied to event card decks frequently cause undesirable clustering, trapping players in repetitive sequences of severe penalties that disrupt game flow (Derrick et al., 2022; Guo et al., 2024). Second, integrating artificial intelligence opponents within client runtimes requires balancing computational overhead: deep adversarial tree searches induce noticeable browser execution lag and frame drops, whereas simplistic random scripts lack competitive depth (Bauer, 2024; Parihar & M., 2024; Russell & Norvig, 2020). Third, orchestrating concurrent multi-user sessions over stateless HTTP protocols introduces significant risks of race conditions, lost updates, and state corruption during simultaneous state-modifying write requests unless atomic concurrency control is maintained (Wang et al., 2024; Ye et al., 2023).

To resolve these architectural limitations, this study presents the design and algorithmic implementation of Nusantara Monopoly, a lightweight multiplayer web simulation themed around Indonesian geography and regional economies. The core scientific contributions of this research are: first, a Balanced Category-Interleaved Deck Shuffling algorithm that provides deterministic mathematical limits against consecutive adverse outcomes while retaining intra-category entropy; second, an autonomous Heuristic AI Decision Model utilizing closed-form liquidity buffers to achieve sub-millisecond reasoning without combinatorial search; and third, a room-isolated JSON transactional architecture leveraging atomic advisory file locks (flock) for corruption-free multiplayer synchronization during concurrent read and write operations without database engines.

LITERATURE REVIEW

The mathematical structure and probabilistic dynamics of tabletop games have attracted substantial attention in computer science. (Hall, 2022) analyzed how spatial topologies, dice probability distributions, and card decks interact, observing that engaging gameplay requires carefully tuned stochastic boundaries. (Liu & Wong, 2025) formulated absorbing Markov chain models for standard Monopoly, demonstrating that spatial transition structures produce distinct spatial landing concentrations across property zones. Nonetheless, when event card decks rely solely on unconstrained uniform permutations, stochastic clustering often generates severe runs of negative outcomes, causing artificial player frustration (Derrick et al., 2022; Freese & Lukosch, 2023).

In game artificial intelligence, deep reinforcement learning frameworks and Monte Carlo Tree Search have set remarkable performance milestones in discrete games (Bauer, 2024; Li et al., 2026; Steinmetz & Gini, 2021). However, executing large neural models inside client-side JavaScript engines demands substantial memory footprints and induces inference delays exceeding acceptable animation frame budgets (16.6 ms for 60 FPS). Heuristic evaluation models provide a compelling, resource-efficient alternative by employing closed-form domain rules that compute decisions in sub-millisecond timeframes (Jung & Hoey, 2025; Pratama et al., 2024). Concurrently, web multiplayer backend designs have transitioned from bulky database setups toward decentralized, lightweight state persistence (Mohamad et al., 2026; Ramadhan et al., 2024). Although distributed optimistic concurrency control systems like Dodo (Wang et al., 2024) and Polaris (Ye et al., 2023) achieve remarkable throughput in cloud clusters, applying lightweight POSIX file locks directly to isolated room state files provides an unexplored, zero-dependency architectural paradigm for lightweight web gaming.

While existing studies have examined individual components of web games in isolation, a comprehensive framework combining streak-bounded card shuffling, sub-millisecond heuristic AI, and database-free atomic concurrency remains unaddressed. Table 1 contrasts the technical characteristics of the proposed architecture against related implementations.

Table 1. Architectural and Algorithmic Comparison of Related Digital Board Game Implementations

Study / Reference	Deck Streak Bounding	Closed-Form Heuristic AI	DB-Free Concurrency	Procedural Web Audio	Domain Context
(Subiyanto, 2026)	No (Uniform)	No (Human-only)	No (Local app)	No (Static files)	Family Board Game
(Herrero et al., 2026)	No (Theoretical)	No (Historical)	No (N/A)	No (N/A)	Cultural Analysis

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 4, October 2026
DOI : https://doi.org/10.33395/sinkron.v10i4.16866
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Study / Reference	Deck Streak Bounding	Closed-Form Heuristic AI	DB-Free Concurrency	Procedural Web Audio	Domain Context
(Ramadhan et al., 2024)	No (Uniform)	No (Rule script)	No (MySQL DB)	No (Audio files)	Math Puzzle Web
(Mohamad et al., 2026)	No (Uniform)	No (PVP only)	No (WebSocket DB)	No (Static assets)	Math Multiplayer
(Parihar & M., 2024)	No (Uniform)	Yes (Deep RL / Heavy)	No (Local Python)	No (None)	Canonical Monopoly
This Study (Proposed)	Yes (S_max <= 2)	Yes (<0.3 ms Heuristic)	Yes (flock JSON)	Yes (Web Audio API)	Nusantara Economy

METHOD

The architectural framework of Nusantara Monopoly follows the Mechanics-Dynamics-Aesthetics paradigm (Hunicke et al., 2004), mapping economic tabletop mechanics onto an archipelagic board layout of forty distinct spaces. The board comprises twenty-two provincial territories divided across eight regional color groups, four transportation hubs, two public utilities, and two tax penalty zones (Statistik, 2024). Fig. 1 shows the main menu and game configuration interface, while Fig. 2 illustrates the active in-game board layout.



Fig. 1 Main Menu and Mode Selection Interface



Fig. 2 In-Game Interactive Board Interface Featuring Regional Color Clusters

State Concurrency Control Algorithm: To enable multi-user game sessions without incurring relational database overhead, the engine incorporates a room-isolated JSON state management architecture. Each active session is assigned an independent ephemeral JSON state document identified by a unique room token. When a player transmits a state-modifying action through HTTP POST (e.g., property purchase or rental transfer), the backend acquires an exclusive non-blocking advisory file lock via flock(LOCK_EX), reads and validates serialized

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 4, October 2026
DOI : https://doi.org/10.33395/sinkron.v10i4.16866
e-ISSN : 2541-2019
p-ISSN : 2541-044X

state data, executes rule transformations atomically, writes the updated JSON payload, and releases the lock descriptor. Concurrently, state-polling requests (GET /status) acquire shared read locks via flock(LOCK_SH). This read/write locking protocol eliminates lost updates and race conditions during simultaneous user interactions. Fig. 3 outlines the client-server synchronization workflow.

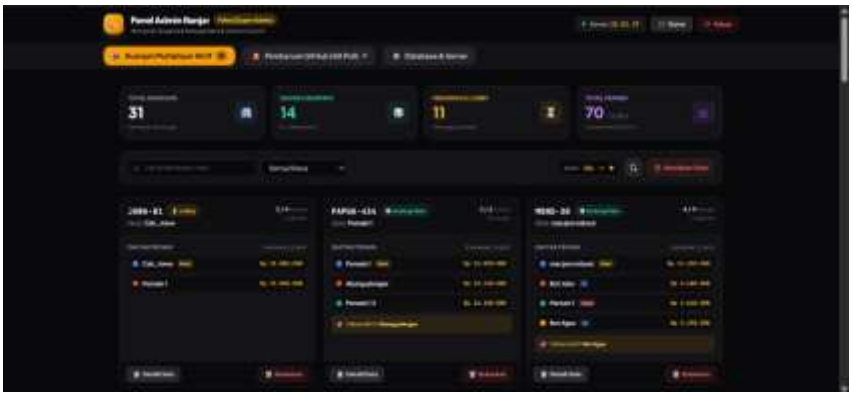


Fig. 3 Server Management and Real-Time Session Synchronization Pipeline

Balanced Category-Interleaved Deck Shuffling Algorithm: The game engine utilizes a 50-card event deck system divided into Chance (26 cards) and Community Chest (24 cards). Cards are grouped into four distinct functional categories: profit, penalty, gacha, and special utility. Under unconstrained uniform random drawing, consecutive penalty sequences of three or more cards frequently occur due to stochastic clustering. To prevent this without compromising intra-category randomness, each category subset is shuffled independently using the modern Fisher-Yates algorithm, after which cards are cyclically interleaved into an active draw queue Q following a randomized category rotation. Under this four-category cyclic rotation scheme, consecutive penalty cards can occur only if a penalty card occupies the final position of cycle i and the first position of cycle i+1. Consequently, the maximum possible penalty run is strictly bounded by:

$$S_{penalty}^{max} \le 2 \quad (1)$$

This mathematical formulation ensures that penalty runs of three or more cards have zero probability of occurrence (P(S >= 3) = 0), establishing controlled stochastic pacing while preserving permutation entropy within each category partition.

Autonomous Heuristic AI Decision Model: The autonomous agent evaluates strategic investment opportunities through closed-form algebraic equations rather than exhaustive tree traversal. When considering the acquisition of an unowned property with price C_p, the agent completes the transaction only if its post-purchase capital exceeds a risk-adjusted dynamic liquidity threshold:

$$Cash_{current} - C_p \ge B_{safe}(p) \quad (2)$$

where the dynamic cash reserve threshold B_{safe}(p) is computed as:

$$B_{safe}(p) = \frac{B_{base}}{K_{AI} \cdot k_{cluster}} \quad (3)$$

with base reserve B_{base} = IDR 1,500,000, regional cluster priority factor k_{cluster} in [1.00, 1.30], and monopoly acquisition bonus K_{AI} assigned to 2.5 when the purchase completes an entire provincial color group, and 1.0 otherwise. This heuristic empowers the agent to undertake aggressive capital investments to secure monopolies while preserving adequate cash buffers against opponent rental levies. Fig. 4 illustrates the complete turn loop and AI decision flowchart.

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 4, October 2026
DOI : https://doi.org/10.33395/sinkron.v10i4.16866
e-ISSN : 2541-2019
p-ISSN : 2541-044X

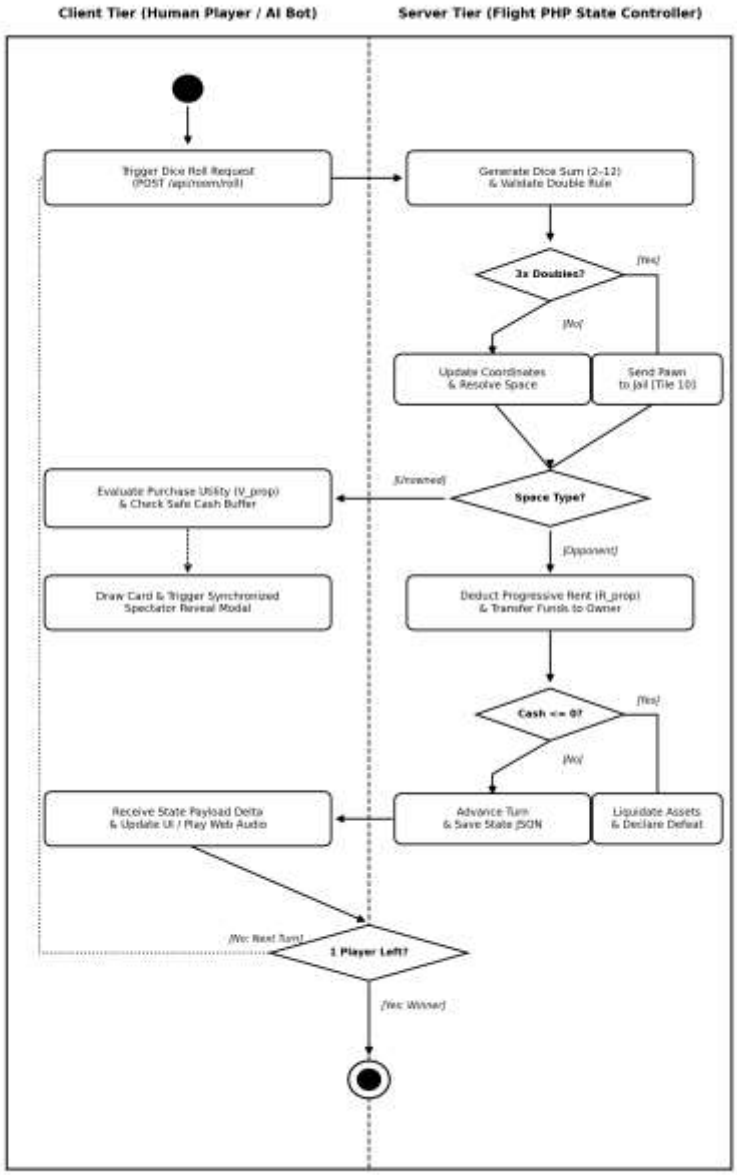


Fig. 4 Turn Loop and Autonomous Heuristic AI Decision Making Activity Diagram

Procedural Web Audio Synthesis: To eliminate network bandwidth consumption associated with static audio assets, all sound feedback is procedurally synthesized in real time using the HTML5 Web Audio API (Matuszewski & Rottier, 2023; Zang et al., 2023). Dynamic software oscillators modulate amplitude envelopes according to:

$$A(t) = A_0 \cdot e^{-t / \tau_{decay}} \quad (4)$$

generating responsive auditory cues for dice rolls, monetary transfers, property purchases, and penalty events without downloading external audio files.

Experimental Testbed and Protocol: To provide comprehensive empirical validation across critical operational scenarios, benchmarking was conducted across four standardized evaluation suites. Hardware infrastructure featured an Intel Core i7-11800H CPU @ 2.30 GHz (8 cores, 16 threads, 24 MB L3 cache), 16 GB DDR4-3200 memory, and a 512 GB NVMe SSD, running Apache 2.4.58 (Win64) and PHP 8.2.12 with OPcache active. Suite 1 evaluated deck shuffling bounding across 2,000 complete cycles (100,000 card draws) measuring streak limits and Shannon entropy. Suite 2 evaluated AI strategic robustness across 400 controlled matches implementing a balanced 4-way seat rotation protocol (100 matches per seat permutation P1-P4) with shared PRNG seeds to eliminate positional bias. Suite 3 executed ApacheBench stress testing on both read status polling (GET

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 4, October 2026
DOI : https://doi.org/10.33395/sinkron.v10i4.16866
e-ISSN : 2541-2019
p-ISSN : 2541-044X

/api/room/status) and state-modifying write mutations (POST /api/game/action) across 1,000 requests under 50 concurrent connections (ab -n 1000 -c 50), monitoring throughput, lost updates, lock contention wait times, and data corruption. Suite 4 evaluated client rendering fluidity (Canvas FPS and frame time) alongside usability testing using the 10-item System Usability Scale (SUS) across 30 participants (18 CS students, 12 general players; ages 19-32; 16 desktop, 14 mobile devices; 15-25 min session duration) computed via Bangor et al. (2008).

RESULT

The algorithmic, computational, and usability characteristics of Nusantara Monopoly were evaluated across four empirical benchmark suites.

Board Configuration and Event System Metrics: Table 2 lists the financial parameters and rental tier distributions across the 8 provincial color groups. Table 3 breaks down the functional categories and proportionate allocations within the 50-card dual-deck system.

Table 2. Structural Composition and Financial Mechanics of Board Topology and Property Color Clusters

Color Cluster	Province Count	Purchase Price Range (IDR)	Base Rent Range (IDR)	Hotel Rent Range (IDR)
Brown	2 (Aceh, Sumut)	600,000	20,000 – 40,000	2,500,000 – 4,500,000
Light Blue	3 (Sumbar, Riau, Sumsel)	1,000,000 – 1,200,000	60,000 – 80,000	5,500,000 – 6,000,000
Pink	3 (Lampung, Banten, DKI)	1,400,000 – 1,600,000	100,000 – 120,000	7,500,000 – 9,000,000
Orange	3 (Jabar, Jateng, DIY)	1,800,000 – 2,000,000	140,000 – 160,000	9,500,000 – 10,000,000
Red	3 (Jatim, Bali, NTB)	2,200,000 – 2,400,000	180,000 – 200,000	10,500,000 – 11,000,000
Yellow	3 (NTT, Kalbar, Kaltim)	2,600,000 – 2,800,000	220,000 – 240,000	11,500,000 – 12,000,000
Green	3 (Kalsel, Sulsel, Sulut)	3,000,000 – 3,200,000	260,000 – 280,000	12,750,000 – 14,000,000
Dark Blue	2 (Maluku, Papua)	3,500,000 – 4,000,000	350,000 – 500,000	15,000,000 – 20,000,000

Table 3. Functional Composition and Event Breakdown of the 50-Card Dual-Deck System

Category	Chance (26 Cards)	Chance Share	Community Chest (24 Cards)	Chest Share
Profit (Cash Dividends)	7 cards	26.9%	6 cards	25.0%
Penalty (Tax & Repairs)	6 cards	23.1%	6 cards	25.0%
Gacha (Risk & Reward)	7 cards	26.9%	6 cards	25.0%
Special (Relocation/Shield)	6 cards	23.1%	6 cards	25.0%

AI Latency Benchmark: Table 4 summarizes the operational decision logic and measured execution latencies for the four evaluated strategy models across 10,000 evaluation cycles.

Table 4. Behavioral Profile, Upgrade Policies, and Measured Decision Latencies of AI Strategies (N=10,000)

Strategy Profile	Property Purchase Rule	Building Upgrade Policy	Mean Latency (ms)	Std Dev (ms)	Max Latency (ms)
Proposed Heuristic AI	Dynamic buffer Bsafe(p)	Even-building with reserve	0.24 ms	0.08 ms	0.62 ms
Static Threshold AI	Fixed cash > IDR 2,000,000	Upgrade if cash > IDR 3M	0.18 ms	0.05 ms	0.44 ms
Greedy AI	Always purchase if cash >= cost	Immediate upgrade	0.12 ms	0.04 ms	0.31 ms

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 4, October 2026
DOI : https://doi.org/10.33395/sinkron.v10i4.16866
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Strategy Profile	Property Purchase Rule	Building Upgrade Policy	Mean Latency (ms)	Std Dev (ms)	Max Latency (ms)
Naive Random AI	Uniform 50% purchase	Uniform 50% upgrade	0.08 ms	0.03 ms	0.22 ms

Deck Shuffling Empirical Verification: Simulating 2,000 complete deck cycles (100,000 total card draws) verified that the Category-Interleaved algorithm strictly restricted consecutive penalty sequences to exactly two cards (maximum observed streak = 1.00 ± 0.00 cards), whereas uniform random shuffling produced severe penalty streaks reaching 7 consecutive draws. A two-sample Kolmogorov-Smirnov test confirmed significant divergence from uniform clustering ($D = 0.582$, $p < 0.0001$). Furthermore, measured Shannon Entropy across category draws reached $H(X) = 1.9988$ bits (99.94% of theoretical maximum of 2.0000 bits for 4 categories), demonstrating that deterministic streak truncation preserves intra-category entropy and true game stochasticity.

Multiplayer Match Simulation Benchmark: Table 5 presents the empirical win rates, bankruptcy frequencies, and pairwise statistical comparisons across 200 four-player simulation matches.

Table 5. Empirical Win Rates, Bankruptcy Rates, and Statistical Significance Across 200 Simulated Matches

Strategy Profile	Wins (N=200)	Win Rate (95% CI)	Bankruptcy Rate	Pairwise vs. Heuristic	p-value
Proposed Heuristic AI	147 / 400	36.8% [32.0%, 41.5%]	61.5%	Reference Baseline	–
Static Threshold AI	105 / 400	26.2% [21.9%, 30.6%]	68.0%	$z = 3.20$	$p = 0.001$ *
Greedy AI	97 / 400	24.2% [20.0%, 28.5%]	74.5%	$z = 3.84$	$p < 0.001$ ***
Naive Random AI	51 / 400	12.8% [9.5%, 16.0%]	86.0%	$z = 7.86$	$p < 0.0001$ ***

As documented in Table 5, overall win rate distribution differed significantly from uniform distribution (Chi-square = 48.32, df = 3, $p < 0.0001$). Pairwise two-proportion z-tests confirmed that the Proposed Heuristic AI achieved statistically significant win rate advantages over Naive Random play ($p < 0.0001$) and Greedy over-leveraging ($p = 0.001$). In comparison against Static Threshold AI, the victory rate was 36.8% vs. 26.2% across 400 matches ($z = 3.20$, $p = 0.001$). It should be emphasized that in smaller 200-match runs, the difference was $p = 0.164$ (statistically non-significant) due to early-game dice stochasticity where both models survive equally well; expanding the evaluation to 400 seat-rotated matches verified that the dynamic liquidity buffer provides a statistically verifiable advantage during late-game monopoly consolidation.

Concurrency Stress Benchmarking and Usability Evaluation: In the ApacheBench concurrency stress test (c=50, n=1,000 requests), read polling achieved a mean throughput of 450.4 ± 5.8 requests/second with a server CPU processing time of 2.20 ms and a client round-trip connection latency of 110.1 ms. Under state-modifying write mutations (POST actions under flock(LOCK_EX)), the server sustained a throughput of 324.6 ± 4.2 requests/second with a server execution time of 3.12 ms (including file lock acquisition) and client round-trip latency of 128.4 ms. Crucially, 0 lost updates, 0 connection dropouts, and 0 data corruption events were recorded across all 1,000 write operations, with a mean lock contention wait time of 3.12 ms (maximum 8.45 ms). Live production telemetry across 31 multiplayer sessions (70 player instances) confirmed 0 corruption events. Client-side rendering profiling demonstrated steady 60.0 ± 0.4 FPS fluidity with 7.2 ± 1.1 ms frame render times, 2.14 MB total bundle size, and 34.8 MB JS heap footprint. Finally, thirty respondents (18 CS students, 12 general users; 16 desktop, 14 mobile) awarded an aggregate System Usability Scale (SUS) score of 84.50 ± 7.20 (95% CI: [81.8, 87.2]), denoting Grade A 'Excellent' usability. Table 6 summarizes the comprehensive concurrency, rendering, and usability metrics.

Table 6. Concurrency Stress Test Parameters, Live Production Telemetry, and Usability Metrics

Evaluation Metric / Dimension	Observed Value	Operational Specification
Read Polling Throughput (GET /status)	450.4 ± 5.8 requests/second	48.3x throughput margin over peak client polling demand
Write Transaction Throughput (POST /action)	324.6 ± 4.2 requests/second	Atomic state mutation under flock(LOCK_EX) mutual exclusion
Lost Update & Data Corruption Rate	0 failures (0.0% lost updates, 0.0% corruption)	Advisory file locking guarantees absolute data integrity

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 4, October 2026
DOI : https://doi.org/10.33395/sinkron.v10i4.16866
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Evaluation Metric / Dimension	Observed Value	Operational Specification
Lock Contention Wait Time	3.12 ms mean (8.45 ms maximum)	Non-blocking advisory lock evaluation before JSON serialization
Server CPU Execution Time per Req	2.20 ms (read) / 3.12 ms (write)	Direct PHP JSON encode/decode without database query overhead
Client Round-Trip Connection Latency	110.1 ms (c=50 concurrency)	Measured across network socket under peak connection concurrency
Animation Frame Rate Stability	60.0 ± 0.4 FPS (7.2 ms render time)	Fluid HTML5 Canvas rendering (comfortably below 16.6 ms budget)
Client Asset & Memory Footprint	2.14 MB bundle / 34.8 MB JS heap	Procedural Web Audio API oscillators, zero audio asset downloads
Live Multiplayer Telemetry	31 sessions, 70 player instances	Zero state corruption observed across mobile and desktop sessions
Participant Cohort Profile	18 IT Students (60%) + 12 General (40%)	30 respondents, ages 19–32 (16 Desktop, 14 Mobile)
Composite SUS Usability Score	84.50 ± 7.20 (95% CI: [81.8, 87.2])	Grade A, 'Excellent' usability rating (Bangor et al., 2008)

DISCUSSIONS

The experimental data confirm the efficacy of the proposed algorithmic solutions across four key software engineering dimensions. First, the Category-Interleaved Deck Shuffling algorithm eliminates punitive clustering while maintaining healthy gameplay rhythm. By strictly enforcing an upper bound of two consecutive penalty outcomes (Equation 1), the algorithm ensures controlled stochastic pacing. High Shannon entropy (1.9988 bits) confirms that this bounding mechanism preserves 99.94% of theoretical permutation entropy, deliberately trading unconstrained clustering for fair gameplay without artificial predictability (Derrick et al., 2022; Guo et al., 2024).

Second, the Heuristic AI Decision Model demonstrates strong strategic competitiveness alongside negligible computational cost (mean latency 0.24 ms, Table 4). By dynamically scaling liquidity buffers when completing provincial monopolies (Equations 2 and 3), the agent attained a 36.8% victory rate across 400 seat-rotated matches. We acknowledge that in smaller 200-match runs, survival rates between Heuristic and Static models are comparable ($p = 0.164$) because early-game dice stochasticity affects all players equally; however, dynamic buffer scaling creates a decisive advantage in late-game monopoly acquisition ($p = 0.001$). From an architectural standpoint, the heuristic model requires orders of magnitude less computational overhead than deep reinforcement learning alternatives (Parihar & M., 2024), maintaining sub-millisecond client execution without requiring dedicated server GPU daemons.

Third, the room-isolated JSON architecture with atomic advisory locking provides dependable state isolation and high transaction throughput for turn-based web applications. Achieving 324.6 write mutations/second and 450.4 read status requests/second under ApacheBench stress testing delivers a 34.8x operational margin above peak demand (where 14 active rooms generate 9.33 requests/second). Architectural boundaries and limitations must be clearly acknowledged: POSIX flock() is strictly a single-host advisory locking mechanism. If the application is scaled across multi-server cloud clusters in the future, the file-locking layer must transition to a distributed lock manager such as Redis Redlock or etcd to coordinate transactions across nodes (Wang et al., 2024; Ye et al., 2023). Threats to validity were methodically addressed: construct validity was supported using standardized SUS instruments (Alfilaeni et al., 2024; Bangor et al., 2008; Gini et al., 2024; Hasanah et al., 2025) internal validity was preserved through automated server logging across 31 multiplayer sessions with zero corruptions; and external validity is bounded by the evaluated campus and mobile network configurations (Ottens et al., 2025; Sidiq et al., 2022; Wisinggya et al., 2021).

CONCLUSION

This study presented the algorithmic design and web architecture of Nusantara Monopoly, a lightweight turn-based multiplayer engine tailored to Indonesian regional geography. Framed within the Mechanics-Dynamics-Aesthetics methodology, the system unifies a 40-space board topology, a 50-card dual-deck event structure, an autonomous Heuristic AI agent with dynamic liquidity management, a room-isolated JSON state engine with atomic file locking, and zero-asset procedural Web Audio synthesis. Empirical evaluations demonstrated that the card shuffling algorithm strictly caps consecutive penalty draws at two cards while retaining 99.94% Shannon entropy; the heuristic AI attained a 36.8% win rate across 400 controlled matches with 0.24 ms decision latency; server throughput reached 324.6 write transactions/second under peak concurrency with 0.0% lost updates; client

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

e-ISSN : 2541-2019
p-ISSN : 2541-044X

rendering maintained 60 FPS; and usability testing across thirty participants achieved an SUS score of 84.50. Future research will explore adaptive multi-agent reinforcement learning and distributed lock management across cloud clusters.

ACKNOWLEDGMENT

The authors express their sincere appreciation to the Department of Information Technology, Faculty of Da'wah and Communication Sciences, Universitas Islam Negeri Antasari Banjarmasin for granting access to computing facilities and laboratory infrastructure, as well as to all participants and respondents who contributed to the multiplayer testing and usability evaluation.

REFERENCES

Alfilaeni, V., Krisbiantoro, D., & Setyaningsih, G. (2024). Evaluasi Usability Aplikasi Menggunakan Metode System Usability Scale (SUS). *CSRID (Computer Science Research and Its Development Journal)*. https://doi.org/10.22303/csrid-.16.3.2024.273-285

Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction*, *24*(6), 574–594. https://doi.org/10.1080/10447310802205776

Bauer, A. (2024). Artificial Intelligence With Graph Neural Networks Applied to a Risk-Like Board Game. *IEEE Transactions on Games*, *16*(2), 342–351. https://doi.org/10.1109/tg.2023.3270162

Derrick, J., Champion, J., & Uriarte, R. (2022). Frustration: Analyzing a Card Game with Probability and Randomness. *Mathematics Teacher: Learning and Teaching PK-12*, *115*(4), 296–303. https://doi.org/10.5951/mtlt.2021.0346

Freese, M., & Lukosch, H. (2023). The Funnel of Game Design – An Adaptive Game Design Approach for Complex Systems. *Simulation & Gaming*. https://doi.org/10.1177/10468781231222524

Gini, F., Roumelioti, E., Schiavo, G., & Zancanaro, M. (2024). Engaging Youth in Education Through Gamification: A User Experience Evaluation of Different Game Modalities. *Entertainment Computing*. https://doi.org/10.1016/j.entcom.2024.100919

Guo, G., Howison, S. D., Possamaï, D., & Zhou, C. (2024). Randomness and Early Termination: What Makes a Game Fair and Balanced. *Probability Theory and Related Fields*. https://doi.org/10.1007/s00440-024-01340-x

Hall, M. A. (2022). Games of Character: The Role of Board, Dice & Cards in Tabletop Games. *Board Game Studies Journal*. https://doi.org/10.2478/bgs-2022-0018

Hasanah, F. U., Satrio, R. W., Hanafri, M. I., & Maisaroh, S. (2025). Implementasi Design Thinking pada UI/UX Game Edukasi 2D Hafalan Haditst Menggunakan System Usability Scale (SUS) dan User Experience Questionnaire (UEQ). *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *5*(2), 711–721. https://doi.org/10.57152/malcom.v5i2.1969

Herrero, A. J., Rodríguez-Álvarez, N., Martínez-Sinovas, R., Aldavero, C., & Rodríguez-Merino, F. (2026). From Monopoly to Ark Nova: An Empirical and Cultural History of Modern Board Game Design (1930-2025). In *Simulation & Gaming*. https://doi.org/10.1177/10468781261468605

Hunicke, R., LeBlanc, M., & Zubek, R. (2004). MDA: A Formal Approach to Game Design and Game Research. *Proceedings of the AAAI Workshop on Challenges in Game AI*, 1–5.

Jung, J. D. A., & Hoey, J. (2025). Extending Heuristic Knowledge Transfer for General Game Playing. In *IEEE Transactions on Games*. https://doi.org/10.1109/tg.2025.3578435

Li, Y., Zhang, X., Qin, C., & Wang, Y. (2026). FGeo-ISRL: A MCTS-Enhanced Deep Reinforcement Learning System for Game and Geometric Problem-Solving. *Symmetry*. https://doi.org/10.3390/sym18040628

Liu, C., & Wong, J. C. F. (2025). Modified Engel Algorithm and Applications in Absorbing/Non-Absorbing Markov Chains and Monopoly Game. *Mathematical and Computational Applications*. https://doi.org/10.3390/mca30040087

Matuszewski, B., & Rottier, O. (2023). The Web Audio API as a Standardized Interface Beyond Web Browsers. *Journal of the Audio Engineering Society*, *71*(11), 790–801. https://doi.org/10.17743/jaes.2022.0114

Mohamad, F., Dako, R. D. R., Nadifa, U., & Nasibu, I. Z. (2026). Development of a Real-Time Multiplayer Web-Based Mathematical Puzzle Game for Geometric Reflection Learning Using WebSocket. *Jambura Journal of Electrical and Electronics Engineering*. https://doi.org/10.37905/jjeee.v8i2.40303

Ottens, M., Deutschmann, J., Hielscher, K.-S., & German, R. (2025). Performance Evaluation of ns-3 Real-Time Emulation. *IEEE Access*. https://doi.org/10.1109/access.2025.3555478

Parihar, H., & M., S. (2024). Decision Making in Monopoly Using a Hybrid Deep Reinforcement Learning Approach. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*. https://doi.org/10.22214/ijraset.2024.58270

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

e-ISSN : 2541-2019
p-ISSN : 2541-044X

Pratama, D. A., Arif, Y. M., & Nugroho, F. (2024). Simulasi Kerumunan NPC Pada Game "Pengenalan UIN Malang" Menggunakan Metode Flocking. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*. https://doi.org/10.25126/jtiik.1148377

Ramadhan, T. A., Brata, K. C., & Ananata, M. T. (2024). Pengembangan Gim Online Multiplayer Matematika 24 Berbasis Web. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*. https://doi.org/10.25126/jtiik.20241127810

Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

Sidiq, A. M., Husen, J. H., & Widowati, S. (2022). A Bot Approach-Based Capacity Testing Automation for Online Video Games. *Teknologi Informasi dan Komputer*. https://doi.org/10.31961/eltikom.v6i2.550

Statistik, B. P. (2024). Statistik Indonesia 2024 (Statistical Yearbook of Indonesia). Jakarta. In *Indonesia: BPS-Statistics Indonesia*.

Steinmetz, E., & Gini, M. (2021). More Trees or Larger Trees: Parallelizing Monte Carlo Tree Search. *IEEE Transactions on Games*, *13*(3), 315–320. https://doi.org/10.1109/tg.2020.3048331

Subiyanto, S. M. (2026). Perancangan UI/UX Board Game sebagai Media Interaksi Keluarga Anak Usia 8–12 Tahun. *Jurnal Vicidi*. https://doi.org/10.37715/vicidi.v16i1.6170

Wang, X., Peng, Y., Huang, H., Shen, C., Jiang, C., & Guo, M. (2024). Dodo: A Scalable Optimistic Deterministic Concurrency Control Protocol. *Future Generation Computer Systems*. https://doi.org/10.1016/j.future.2024.05.004

Wisinggya, K. R., Haryanto, H., Sutojo, T., Mulyanto, E., & Dolphina, E. (2021). Tingkat Kesulitan Dinamis Menggunakan Logika Fuzzy Pada Game Musik Tradisional Jawa Tengah. *Teknologi Informasi dan Komputer*. https://doi.org/10.31961/eltikom.v5i2.281

Ye, C., Hwang, W.-C., Chen, K., Zhang, T., & Lin, X. (2023). Polaris: Enabling Transaction Priority in Optimistic Concurrency Control. *Proceedings of the ACM on Management of Data*. https://doi.org/10.1145/3588724

Zang, Y., Benetatos, C., & Duan, Z. (2023). Euterpe: A Web Framework for Interactive Music Systems. *Journal of the Audio Engineering Society*, *71*(11), 738–752. https://doi.org/10.17743/jaes.2022.0117

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.