

PENGEMBANGAN PLATFORM MANAJEMEN TIM E-SPORTS: STUDI KASUS PADA KOMUNITAS CLASH OF CLANS DI INDONESIA

Dicky Galuh Kurniawan
Program Studi Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
20523234@students.uui.ac.id

Kholid Haryono
Program Studi Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
kholid.haryono@uui.ac.id

Abstract—Pertumbuhan industri *E-sports* di Indonesia mencatat lebih dari 150 juta pemain aktif, namun manajemen komunitas akar rumput, khususnya pada gim strategi *Clash of Clans (CoC)*, masih mengalami kendala teknis dan administratif. Permasalahan utama yang diidentifikasi adalah fragmentasi alat manajemen (penggunaan *spreadsheet* dan aplikasi pesan terpisah), inefisiensi operasional turnamen, serta asimetri informasi dalam perekrutan anggota klan akibat fenomena "kutu loncat" (*hoppers*) dan ketiadaan rekam jejak pemain yang Valid. Penelitian ini bertujuan merancang bangun "*Clashub*", sebuah platform terpusat berbasis web modern yang mengintegrasikan manajemen klan harian dan kompetisi turnamen. Sistem dikembangkan menggunakan metode *Hybrid SDLC (Waterfall dan Iteratif)*, dibangun di atas kerangka kerja *Next.js* dan *Google Firebase*, serta terintegrasi langsung dengan *API* resmi pengembang gim. Kebaruan (*Novelty*) dari penelitian ini terletak pada penerapan modul "*Gamers CV*" yang menggunakan algoritma sistem reputasi untuk melacak kinerja pemain secara historis, serta integrasi *Generative AI (Gemini)* untuk analisis strategi perang. Selain itu, platform ini menghadirkan fitur "*Knowledge Hub*" yang terintegrasi dengan *YouTube API* sebagai wadah sosial berbagi strategi berbasis poin keaktifan. Hasil pengujian menunjukkan platform mampu menyederhanakan administrasi klan, mengotomatisasi bagan turnamen, dan menyediakan metrik validitas pemain yang transparan, menciptakan ekosistem kompetitif yang lebih terpercaya dan efisien.

Keywords—*E-sports Management, Clash of Clans, Sistem Reputasi, Next.js, Generative AI, Knowledge Management.*

I. PENDAHULUAN

Perkembangan industri *Electronic Sports (E-sports)* di Indonesia telah bertransformasi dari sekadar aktivitas rekreasi menjadi ekosistem digital bernilai ekonomi tinggi. Data terbaru menunjukkan Indonesia sebagai salah satu pasar *mobile gaming* terbesar dengan lebih dari 150 juta pemain aktif [1]. Di tengah dominasi gim strategi seperti *Clash of Clans (CoC)* yang menuntut koordinasi tim tingkat tinggi, kebutuhan akan infrastruktur manajemen yang profesional menjadi semakin krusial. Tidak seperti gim individu, gim berbasis klan memerlukan pengelolaan sumber daya manusia, jadwal kompetisi, dan analisis performa yang kompleks.

Namun, observasi pada komunitas *CoC* di Indonesia mengungkapkan kesenjangan teknologi yang signifikan

dalam manajemen operasional. Sebagian besar pengelola klan dan penyelenggara turnamen masih bergantung pada metode manual yang terfragmentasi, seperti penggunaan *spreadsheet* untuk data statistik dan aplikasi pesan instan (*WhatsApp*) untuk koordinasi. Pendekatan ini menyebabkan inefisiensi administrasi, tingginya tingkat kesalahan manusia (*human error*), dan *latency* data statistik perang. Selain itu, penyebaran informasi strategi (seperti tata letak basis dan kombinasi serangan) tersebar tidak terstruktur di berbagai kanal media sosial, menyulitkan pemain untuk mendapatkan referensi terkurasi yang Valid.

Masalah yang lebih mendesak adalah fenomena "kutu loncat" (*hoppers*), di mana pemain sering berpindah klan tanpa etika, meninggalkan klan saat perang berlangsung. Hal ini menciptakan asimetri informasi dalam proses perekrutan; pemimpin klan kesulitan memverifikasi riwayat dan loyalitas calon anggota karena ketiadaan sistem rekam jejak yang terpusat dan Valid [2]. Akibatnya, kepercayaan antar anggota komunitas menurun dan pembentukan tim yang kompetitif menjadi terhambat.

Penelitian ini mengusulkan solusi berupa pengembangan platform terintegrasi bernama "*Clashub*". Berbeda dengan solusi parsial yang hanya berfokus pada manajemen turnamen sesaat, *Clashub* dirancang sebagai ekosistem menyeluruh yang menggabungkan lima pilar: (1) Manajemen operasional klan harian yang tersinkronisasi *real-time* dengan *API* gim, (2) Sistem manajemen turnamen otomatis, (3) Profil pemain berbasis data (*Gamers CV*) untuk membangun reputasi transparan, (4) Integrasi kecerdasan buatan (*AI Assistant*) untuk membantu analisis strategi, dan (5) Modul sosial *Knowledge Hub* yang terintegrasi dengan *YouTube API* untuk memfasilitasi pertukaran strategi berbasis komunitas. Penelitian ini bertujuan untuk membuktikan bahwa integrasi teknologi web modern dan sistem reputasi dapat mengatasi inefisiensi manajerial dan meningkatkan kepercayaan dalam ekosistem komunitas *E-sports*.

II. PENELITIAN TERDAHULU

Penelitian mengenai sistem informasi manajemen *E-sports* telah dilakukan oleh beberapa peneliti sebelumnya dengan fokus yang beragam. Pratama dan Wibowo [4] mengembangkan sistem informasi turnamen berbasis web yang berhasil mendigitalkan proses pendaftaran dan pembuatan bagan pertandingan (*bracket generator*). Sistem

ini terbukti efektif memangkas waktu administrasi teknis saat kompetisi berlangsung.

Dalam lingkup yang lebih spesifik, Arham dkk. [5] merancang sistem informasi untuk komunitas gaming lokal (warung kopi), menekankan pentingnya wadah digital untuk mengorganisir kompetisi akar rumput. Serupa dengan itu, Yunanri dan Measer [6] membangun sistem manajemen *event E-sports* berbasis wilayah geografis untuk mempermudah koordinasi penyelenggara turnamen skala regional.

Meskipun penelitian-penelitian tersebut [4]–[6] memberikan kontribusi signifikan pada aspek manajemen *event* jangka pendek (*episodic events*), terdapat celah penelitian (*research gap*) yang belum tertangani. Mayoritas sistem yang ada belum menyentuh aspek manajemen operasional harian sebuah tim/klan yang bersifat kontinu. Selain itu, belum ada platform yang mengimplementasikan teori sistem reputasi (*reputation system*) dalam komunitas maya sebagaimana dijelaskan dalam literatur kepercayaan digital [7], untuk mengatasi masalah validitas data pemain dan perilaku *toxic* dalam jangka panjang.

Penelitian ini mengisi kekosongan tersebut dengan mengintegrasikan manajemen klan (*daily operations*), manajemen turnamen (*events*), sistem reputasi berbasis data historis (*long-term trust*), dan manajemen pengetahuan (*knowledge sharing*) ke dalam satu platform terpadu. Keunggulan teknis juga ditambahkan melalui integrasi langsung dengan *API* pengembang gim (Supercell) untuk validasi data otomatis dan pemanfaatan *Generative AI* untuk fitur analisis, yang belum ditemukan pada penelitian sejenis sebelumnya.

III. METODOLOGI PENELITIAN

Penelitian ini mengadopsi pendekatan Model Pengembangan Hibrida (*Hybrid SDLC*), yang mengintegrasikan kekokohan struktur *Waterfall* dengan fleksibilitas model *Iteratif*. Pendekatan ini dipilih untuk mengakomodasi kompleksitas sistem "*Clashub*" yang memiliki modul manajemen data statis dan modul interaksi komunitas yang dinamis.

Proses pengembangan dilaksanakan melalui tahapan berikut:

1) Analisis & Pengumpulan Data: Observasi mendalam terhadap manajemen klan *CoC* dan pemetaan *endpoints* dari *API* eksternal (*Clash of Clans*, YouTube Data, dan Google Gemini).

2) Perancangan Fundamental (*Waterfall*): Fokus pada perancangan basis data *NoSQL* yang stabil untuk menyimpan entitas inti (User, Clan, Tournament) dan arsitektur keamanan sistem.

3) Pengembangan Fitur Siklis (*Iteratif*): Implementasi kode dilakukan dalam empat siklus iterasi (*sprints*) untuk memastikan seluruh pilar solusi terbangun:

a) *Siklus 1*: Pengembangan modul Manajemen Klan dan sinkronisasi data pemain *real-time*.

b) *Siklus 2*: Implementasi algoritma Sistem Turnamen (pembuatan bagan otomatis).

c) *Siklus 3*: Integrasi *AI Assistant* dan logika perhitungan skor Reputasi (*Gamers CV*).

d) *Siklus 4*: Pengembangan modul Ekosistem Komunitas yang mencakup pusat pencarian (*Team Hub*) dan wadah berbagi pengetahuan (*Knowledge Hub*).

4) Pengujian Terintegrasi: Validasi fungsional menyeluruh (*Black Box*) dan uji usabilitas (*SUS*) kepada pengguna akhir.

IV. PERANCANGAN SISTEM

A. Arsitektur Integrasi Sistem

Sistem dibangun di atas arsitektur *Serverless* modern dengan *Next.js* (*App Router*) sebagai *Middleware Orchestrator*. Topologi sistem dirancang untuk menangani pertukaran data dari berbagai sumber eksternal guna mendukung fitur manajemen dan pencarian komunitas.

Berdasarkan analisis kebutuhan integrasi, arsitektur terdiri dari tiga lapisan:

1) *Lapisan Antarmuka (Client-Side)*: Dibangun menggunakan React dan *Tailwind CSS*. Lapisan ini menyajikan visualisasi data yang responsif, termasuk antarmuka filter pencarian klan dan dasbor profil pemain.

2) *Lapisan Logika & Integrasi Layanan (Server-Side)*: Lapisan ini bertindak sebagai pusat pemrosesan logika bisnis yang menghubungkan platform dengan layanan pihak ketiga:

a) *Layanan Sinkronisasi Gim (CoC API)*: Mengambil data statistik pemain *real-time*, serta memfasilitasi pencarian klan publik global untuk fitur *Team Hub*.

b) *Layanan Kecerdasan Buatan (Google Gemini API)*: Memproses *prompt* konteks permainan (level *Town Hall*, pasukan musuh) untuk menghasilkan saran strategi teks natural pada fitur *AI Assistant*.

c) *Layanan Konten Media (YouTube Data API)*: Secara otomatis menarik *metadata* video strategi terbaru dari kanal resmi dan komunitas untuk mengisi konten pada modul *Knowledge Hub*.

d) *Logika Internal Turnamen*: Algoritma khusus (*Bracket generator*) dijalankan di sisi server untuk mengelola jadwal pertandingan.

3) *Lapisan Data Persisten (Cloud Firestore)*: Menyimpan data relasional dokumen (*NoSQL*) untuk entitas yang terverifikasi dalam platform (Klan Terdaftar, Profil Pemain, Ulasan). Mekanisme *caching* diterapkan untuk data *API* guna meminimalkan latensi.

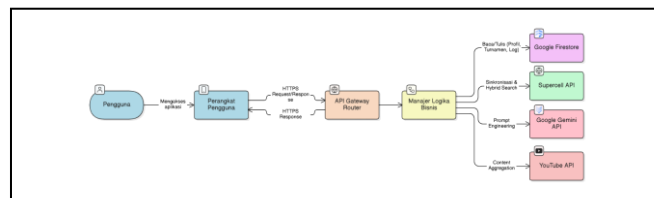


Fig. 1. Arsitektur Integrasi Sistem *Clashub* yang menghubungkan modul internal dengan tiga layanan *API* eksternal utama.

B. Perancangan Basis Data

Berbeda dengan basis data relasional (*SQL*) tradisional, *Clashub* mengimplementasikan struktur data berbasis dokumen (*Document-Oriented*) menggunakan *Cloud Firestore* untuk fleksibilitas skema (*schema-less*).

Berdasarkan perancangan data, struktur penyimpanan dibagi menjadi lima koleksi utama:

1) *Koleksi Pengguna (Users Collection)*: Menyimpan profil publik pemain, tautan akun gim (*Player Tag*), dan agregasi skor reputasi.

2) *Koleksi Klan (Clans Collection)*: Menyimpan data klan yang telah terdaftar dan terverifikasi di platform. Data riwayat perang (*War Logs*) disimpan dalam sub-koleksi untuk efisiensi query.

3) *Koleksi Turnamen (Tournaments Collection)*: Menampung *metadata* kompetisi dan struktur bagan pertandingan.

4) *Koleksi Ulasan (Reviews Collection)*: Menyimpan validasi sosial (rating & ulasan) antar pengguna.

5) *Koleksi Pengetahuan (Posts Collection)*: Menyimpan konten strategi pengguna dan *metadata* video YouTube.

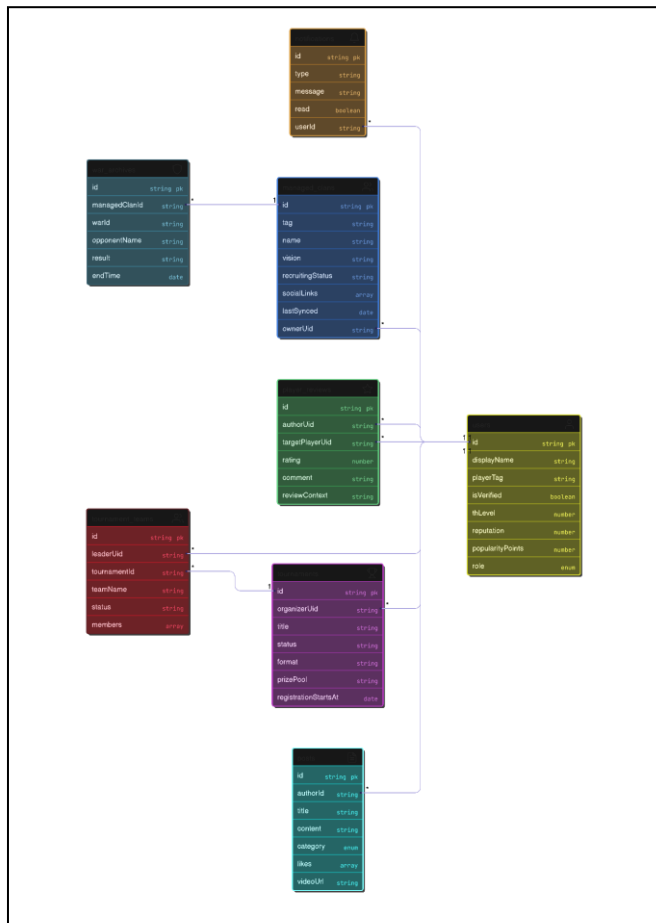


Fig. 2. Skema Relasi Antar-Koleksi pada *Cloud Firestore*.

C. Implementasi Logika Fitur Cerdas

Nilai kebaruan (*Novelty*) sistem terletak pada implementasi algoritma logika server yang menangani kecerdasan buatan, reputasi, dan pencarian data:

1) *Algoritma Asisten Strategi (Context-Aware AI Logic)*: Sistem melakukan *Pre-Processing* data perang (*Level Town Hall, Hero, Pasukan Musuh*) sebelum mengirimkannya ke *Google Gemini API* melalui *Structured Prompt*, menghasilkan saran strategi yang spesifik dan relevan.

2) *Algoritma Kalkulasi Reputasi (Reputation Scoring)*: Sistem "*Gamers CV*" menghitung skor kepercayaan pemain

melalui pendekatan *Hybrid Scoring* yang menggabungkan parameter *Hard Skills* (Statistik in-game via *API*) dan *Soft Skills* (Rata-rata rating ulasan rekan setim).

3) *Algoritma Generator Bagan (Automated Bracket System)*: Sistem menerapkan algoritma *Binary Tree Generation* untuk menyusun bagan pertandingan turnamen (sistem gugur) secara otomatis segera setelah kuota peserta terpenuhi.

4) *Mekanisme Pencarian Hibrida (Hybrid Search & Discovery)*: Fitur *Team Hub* menerapkan logika pencarian dua jalur untuk memenuhi kebutuhan pengguna:

a) *Jalur Terverifikasi (Internal)*: Melakukan query ke *Firestore* untuk mencari Klan atau Pemain yang sudah terdaftar di *Clashub*, dengan dukungan filter mendalam (*Reputasi, Town Hall, Role*).

b) *Jalur Publik (Eksternal)*: Bertindak sebagai *Proxy* ke *CoC API* untuk mencari klan publik di seluruh dunia berdasarkan nama atau tag, memungkinkan rekrutmen lintas platform.

V. IMPLEMENTASI DAN PEMBAHASAN

Bagian ini memaparkan realisasi teknis dari sistem "*Clashub*", meliputi lingkungan pengembangan, logika sistem utama, dan visualisasi antarmuka pengguna secara menyeluruh.

A. Lingkungan Implementasi dan Arsitektur Awan

Sistem *Clashub* diimplementasikan menggunakan arsitektur *Serverless* modern untuk menjamin skalabilitas dan ketersediaan tinggi (*high availability*) tanpa beban pemeliharaan server fisik. Berdasarkan rancangan topologi, lingkungan operasional (*production environment*) dibangun di atas integrasi dua penyedia layanan awan utama:

1) *Infrastruktur Komputasi (Vercel)*: Bertindak sebagai host utama yang menjalankan kerangka kerja *Next.js 14*. Platform ini menangani *Server-Side Rendering (SSR)* untuk antarmuka pengguna dan menjalankan fungsi *API Routes* sebagai *Serverless functions*. Hal ini memungkinkan sistem menangani lonjakan trafik saat turnamen tanpa konfigurasi *load balancer* manual.

2) *Infrastruktur Backend (Google Cloud Platform / Firebase)*: Digunakan sebagai *Backend-as-a-Service (BaaS)*. Layanan ini mencakup *Firebase Authentication* untuk manajemen sesi pengguna lintas perangkat dan *Cloud Firestore* untuk penyimpanan data *NoSQL* yang membutuhkan sinkronisasi *real-time*.

Integrasi kedua lingkungan ini menciptakan ekosistem yang aman, di mana kunci *API* sensitif (seperti *Token API Clash of Clans* dan *Gemini AI*) disimpan secara terenkripsi dalam variabel lingkungan (*environment variables*) di sisi server *Vercel*, tidak terekspos ke sisi klien (browser).

B. Realisasi Topologi Integrasi Sistem

Implementasi kode program mengikuti pola arsitektur modular yang memisahkan logika antarmuka, logika bisnis, dan keamanan data. Berdasarkan struktur direktori proyek, sistem dibagi menjadi tiga lapisan logika utama:

1) *Lapisan Interaksi (Direktori /app)*: Berisi komponen antarmuka *React Server Components (RSC)*. Lapisan ini hanya bertugas menampilkan data dan mengirimkan

permintaan ke *API* internal. Tidak ada logika pemrosesan data berat yang dilakukan di sisi klien untuk menjaga performa aplikasi tetap ringan di perangkat seluler.

2) *Lapisan Middleware & Orkestrasi API (Direktori /app/API)*: Ini adalah implementasi teknis dari solusi masalah fragmentasi dan keamanan. Direktori ini berfungsi sebagai Gateway terpusat yang menjembatani komunikasi aman antara pengguna dengan layanan eksternal utama: Supercell (Data Gim), Google Gemini (Kecerdasan Buatan), dan YouTube Data (Konten Komunitas). Sebagai contoh ilustrasi alur kerja pada fitur Asisten Strategi:

- Memvalidasi token sesi pengguna (*Server Auth*).
- Mengambil data pasukan musuh dari *API* Supercell.
- Menyusun *prompt* dan meneruskannya ke Gemini *API*.
- Mengembalikan hasil analisis bersih ke pengguna.

Pola ini mengisolasi kompleksitas integrasi sehingga klien hanya berinteraksi dengan satu titik akhir (*endpoint*) yang standar.

3) *Lapisan Logika Bisnis Terpusat (Direktori /lib)*: Direktori ini menyimpan fungsi-fungsi inti yang bersifat *reusable* dan aman, seperti modul *firebase-admin* untuk operasi basis data dengan hak akses istimewa (*privileged access*) yang tidak boleh diakses oleh pengguna biasa.

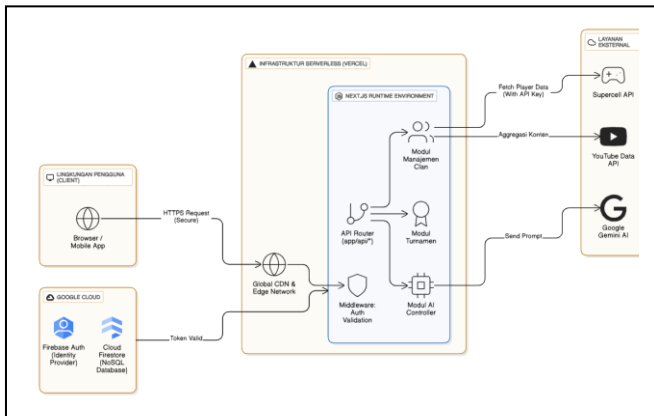


Fig. 3. Topologi Integrasi Sistem yang menunjukkan aliran data aman melalui *Serverless functions*.

C. Implementasi Logika Sistem Utama

Kecerdasan sistem *Clashub* terletak pada implementasi algoritma sisi server yang menangani proses bisnis kompleks secara otomatis. Berdasarkan realisasi logika, terdapat enam modul algoritma kunci yang menopang seluruh pilar solusi:

1) *Algoritma Kalkulasi Reputasi (Gamers CV Logic)*: Berbeda dengan profil pemain konvensional yang statis, sistem menerapkan algoritma *Weighted Scoring* untuk menghitung "Skor Kepercayaan" secara dinamis. Logika ini menggabungkan dua sumber data heterogen:

a) *Metrik Kinerja (Hard Skill)*: Diambil dari rata-rata statistik perang (Bintang Perang, Kerusakan) via *API CoC*. Sebagaimana terlihat pada kode *participationAggregator.ts*, sistem secara otomatis mengklasifikasikan performa pemain menjadi status 'Promosi', 'Aman', atau 'Demosi' berdasarkan ambang batas (*threshold*) kesuksesan serangan.

b) *Metrik Sosial (Soft Skill)*: Diambil dari agregasi nilai ulasan (*Review Stars*). Validasi ketat diterapkan pada

reviews/player/route.ts untuk memastikan hanya rekan setim yang pernah berada dalam satu klan yang dapat memberikan ulasan, mencegah manipulasi skor (*review bombing*).

```
// BAGIAN A: Kalkulasi Hard Skill (Otomatisasi Status Promosi/Demosi)
// File: app/api/coc/sync-managed-clan/logic/participationAggregator.ts

// ... (konstanta threshold)
if (totalFail >= FAIL_DEMOTE_THRESHOLD && memberRole === ClanRole.ELDER) {
  status = 'Demosi'; // Otomatis tandai turun jabatan
  statusKeterangan = `Demosi (Total Gagal: ${totalFail}x)`;
} else if (
  totalSuccess >= PROMO_SUCCESS_THRESHOLD && memberRole === ClanRole.MEMBER
) {
  status = 'Promosi'; // Otomatis tandai naik jabatan
  statusKeterangan = `Promosi ke Elder (Sukses ${totalSuccess}x)`;
} else {
  status = 'Aman';
  // ... (logika status aman lainnya)
}

// BAGIAN B: Validasi Soft Skill (Mencegah Fake Review)
// File: app/api/reviews/player/route.ts

// Cek apakah 'author' dan 'target' pernah berada di klan yang sama.
if (reviewContext === 'clan' || reviewContext === 'both') {
  const [authorHistoryDocs, targetHistoryDocs] = await Promise.all([
    getClanHistoryAdmin(authorProfile.uid),
    getClanHistoryAdmin(targetPlayerUid),
  ]);
  // Buat Set berisi semua clanTag yang pernah diikuti author
  const authorClanTags = new Set(authorHistoryDocs.map((doc) => doc.clanTag));
  // Validasi Cross-Reference History
  const hasSharedClan = targetHistoryDocs.some((doc) => authorClanTags.has(doc.clanTag));
  if (!hasSharedClan) {
    return NextResponse.json({
      error: 'Forbidden: Anda hanya dapat mengulas pemain yang pernah satu klan dengan Anda.',
    }, { status: 403 });
  }
}
```

Fig. 4. Implementasi logika hibrida Reputasi: (a) Kalkulasi otomatis kinerja perang, dan (b) Validasi otorisasi ulasan sosial.

2) *Algoritma Agregasi Konten (Knowledge Hub Engine)*: Untuk mengatasi fragmentasi informasi strategi, sistem mengimplementasikan logika *Content Aggregation* yang berjalan secara terjadwal (*Cron Job*). Server secara otomatis:

a) Mengakses *YouTube Data API* untuk mencari video terbaru dari saluran resmi dan kreator terverifikasi.

b) Melakukan penyaringan duplikasi konten (*Deduplication*) berdasarkan *Video ID*.

c) Menyimpan *metadata* video yang relevan ke koleksi posts agar bisa diakses pengguna tanpa perlu berpindah aplikasi.

```
// Loop melalui setiap channel yang
dikonfigurasi for (const channel of
CHANNELS_TO_SYNC) { try { const latestVideos =
await getLatestVideosFromPlaylist(
channel.playlistId, YOUTUBE_API_KEY,
MAX_RESULTS_PER_CHANNEL ); // ... (kode
saveVideoToFirestore) } }
```

Fig. 5. Logika 'Cron Job' untuk sinkronisasi konten video strategi dari YouTube API secara otomatis.

3) *Mekanisme Pencarian Hibrida (Team Hub Logic)*: Fitur pencarian klan menerapkan logika *Hybrid Querying* untuk menjembatani komunitas internal dan ekosistem global. Algoritma pencarian bekerja melalui dua jalur:

a) *Jalur Terverifikasi*: Melakukan query ke database Firestore untuk mencari klan yang sudah terdaftar di platform, memungkinkan filter mendalam seperti "Reputasi Klan" dan "Kebutuhan Rekrutmen".

b) *Jalur Publik*: Bertindak sebagai *Proxy* aman yang meneruskan parameter pencarian ke *API Clash of Clans* global, memungkinkan pengguna menemukan klan mana pun di dunia meskipun belum terdaftar di *Clashub*.

```
// 1. Cek Cache Firestore (Hanya jika tidak
dipaksa update) if (!forceUpdate) { cachedData
= await getPublicClanIndex(clanTag); // ...
(logika cek kadaluarsa) } // 2. Cache stale,
tidak ada, atau dipaksa update: Fetch dari API
CoC apiData = await
cocApi.getClanData(clanTag);
```

Fig. 6. Implementasi API Route untuk pencarian klan hibrida (Database Lokal dan Global API).

4) *Algoritma Manajemen Turnamen (Bracket Generation)*: Sistem mengimplementasikan logika *Binary Tree* untuk menyusun bagan pertandingan eliminasi tunggal. Sebagaimana direalisasikan dalam modul *generate-bracket* (Bab 4.2.4), algoritma bekerja dalam tiga langkah:

a) *Pengacakan (Shuffling)*: Mengacak urutan array peserta terdaftar untuk keadilan.

b) *Pembuatan Node Pertandingan*: Memasangkan peserta ke dalam objek match dengan ID unik dan menangani mekanisme *Bye* otomatis jika jumlah peserta ganjil.

c) *Propagasi Pemenang*: Memperbarui status pertandingan ronde berikutnya secara rekursif saat skor dilaporkan.

```
// 8. Buat Dokumen Match (Ronde 1 Upper
Bracket) const matchesRef =
tournamentRef.collection('matches'); const
matchCountR1 = tournamentData.participantCount
/ 2; for (let i = 0; i < matchCountR1; i++) {
const matchId = `U-R1-M${i + 1}`; // ... (ambil
team1 dan team2 dari shuffledTeams) const
newMatch: TournamentMatch = { matchId: matchId,
round: 1, bracket: 'upper', status: 'pending',
// ... (assignments lainnya) }; // Set dokumen
match baru t.set(matchDocRef,
cleanDataForAdminSDK(dataToSet)); }
```

Fig. 7. Algoritma *Server-Side* untuk pembuatan bagan turnamen sistem gugur otomatis.

5) *Integrasi AI Kontekstual (Prompt Engineering)*: Fitur Asisten Strategi menggunakan teknik *Structured Prompting* pada integrasi Google Gemini. Sistem menyuntikkan data JSON spesifik (*Level Town Hall*, *Pasukan Musuh*) ke dalam instruksi sistem, memastikan AI berperan sebagai analis strategi spesifik gim, bukan *chatbot* umum.

```
// 6. Buat System Prompt (Instruksi untuk AI)
const clanContext = `
ANDA ADALAH "ASISTEN CLAN" UNTUK CLAN BERNAMA
${summarizedClan?.name || 'Clan Ini'}.
...
--- DATA DATABASE CLAN (JSON) ---
1. PROFIL_CLAN:
${JSON.stringify(summarizedClan, null, 2)}
2. DAFTAR_ANGGOTA (Total:
${summarizedMembers.length}):
${JSON.stringify(summarizedMembers, null, 2)}
...
`;

const model = genAI.getGenerativeModel({
model: 'gemini-2.5-flash-preview-09-2025',
systemInstruction: clanContext,
});
```

Fig. 8. Konfigurasi 'Structured Prompt' yang menyuntikkan konteks data gim ke dalam model Gemini AI.

6) *Mekanisme Sinkronisasi Data Hibrida (Smart Caching)*: Untuk fitur Manajemen Klan, sistem menerapkan strategi *On-Demand Revalidation*. Data pemain disimpan di Firestore dengan penanda waktu (*timestamp*). Permintaan ke API Supercell hanya dilakukan jika usia cache melebihi ambang batas, menjamin data manajemen selalu mutakhir tanpa melanggar batas *Rate Limiting API*.

```
// 5. Panggil CoC API (Fetch Live Data)
const cocClanData = await
cocApi.getClanData(encodeURIComponent(clanTag))
;

// 6.6. Proses Data (Calculate Participation)
const enrichedMembers =
getAggregatedParticipationData({
currentMembers: currentMembers,
// ...
});

// 7. Update Dokumen Cache di Firestore
await cacheDocRef.set({
lastUpdated: FieldValue.serverTimestamp(),
members: enrichedMembers,
}, { merge: true });
```

Fig. 9. Logika sinkronisasi data klan: Fetch API eksternal, pemrosesan partisipasi, dan pembaruan database cache.

D. Implementasi Basis Data Terdistribusi

Implementasi penyimpanan data menggunakan *Cloud Firestore* merealisasikan rancangan skema *NoSQL* yang fleksibel. Struktur data dirancang untuk mengoptimalkan performa pembacaan (*read-heavy*) pada aplikasi klien:

1) *Denormalisasi Terkendali*: Meskipun *NoSQL* tidak memiliki *Foreign Key* kaku, sistem mengimplementasikan relasi logis. Contohnya, pada dokumen *tournaments*, data dasar peserta (*nama*, *tag*) disimpan langsung di dalam array peserta (*embedded*), bukan hanya referensi ID. Ini

babak penyisihan hingga final secara visual tanpa perlu *refresh* halaman manual.

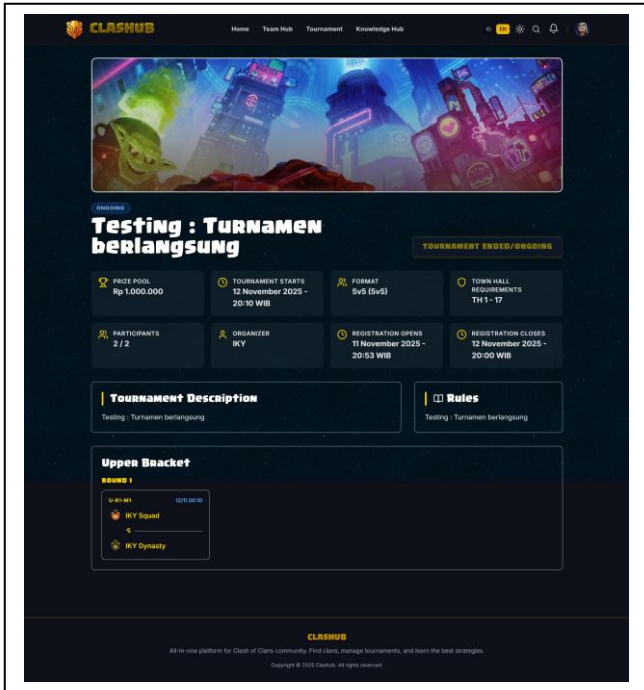


Fig. 14. Visualisasi bagan turnamen sistem gugur yang digenerate otomatis oleh sistem.

5) *Perpustakaan Strategi Multimedia (Knowledge Hub)*: Halaman ini menggabungkan konten teks buatan pengguna dengan konten video YouTube. Implementasi komponen kartu (*FullPostDisplay.tsx*) mendukung penyematan video responsif, menjadikan platform ini pusat belajar yang komprehensif bagi komunitas.

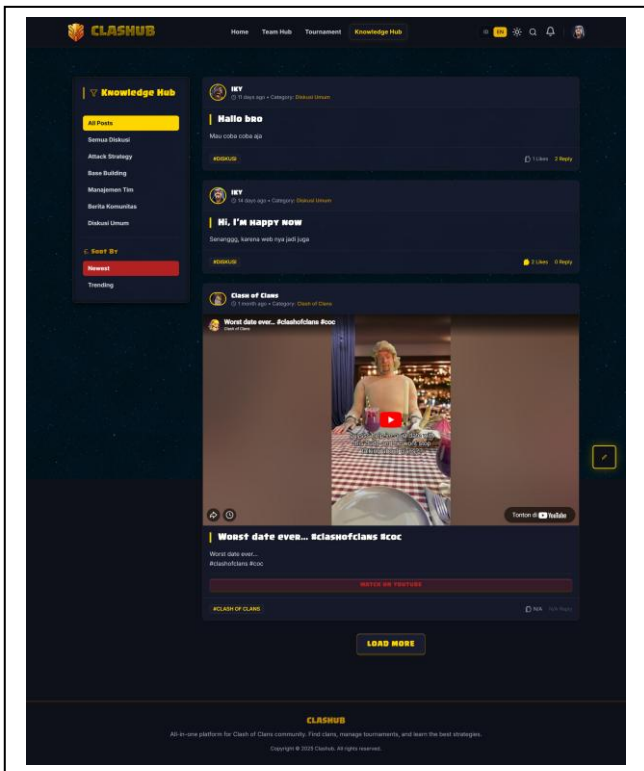


Fig. 15. Tampilan *Knowledge Hub* yang mengintegrasikan diskusi strategi dan konten video.

6) *Asisten Strategi Cerdas (Context-Aware AI)*: Realisasi dari Integrasi *AI* Kontekstual ditampilkan dalam tab khusus (*GeminiAssistantTab.tsx*) di dalam panel manajemen. Antarmuka dirancang menyerupai aplikasi obrolan (*chat interface*) yang intuitif, namun dilengkapi dengan *context awareness*. Saat pengguna membuka tab ini, sistem secara otomatis memuat data perang terkini di latar belakang sebagai konteks, sehingga pengguna dapat langsung bertanya "Bagaimana cara menyerang musuh nomor 1?" dan mendapatkan analisis strategi spesifik tanpa perlu mengetik ulang data musuh.

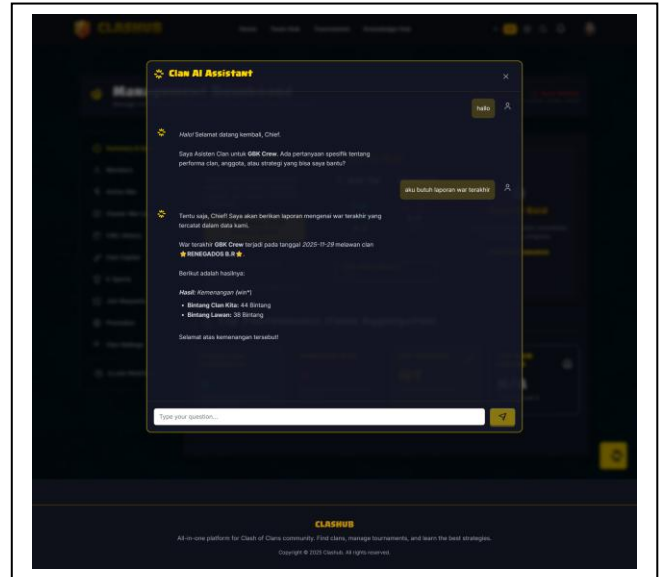


Fig. 16. Antarmuka Asisten Strategi berbasis *Gemini AI* yang memberikan analisis taktik perang secara *real-time*.

F. Pengujian dan Evaluasi Sistem

Untuk memvalidasi kelayakan teknis dan operasional, pengujian dilakukan dengan melibatkan 15 partisipan ahli (*purposive sampling*) yang terdiri dari pemimpin klan dan penyelenggara turnamen berpengalaman (*Town Hall* >12) serta pengalaman bermain di atas 3 tahun. Evaluasi mencakup tiga dimensi utama:

1) *Validasi Fungsional (Black Box Testing)*: Pengujian dilakukan untuk memverifikasi kesesuaian sistem dengan 10 Kebutuhan Fungsional (F-01 s.d. F-10) yang telah didefinisikan pada tahap analisis. Sebagaimana dirangkum dalam Tabel I, hasil pengujian menunjukkan status Valid untuk seluruh fitur, membuktikan bahwa logika sistem berjalan sesuai spesifikasi tanpa kesalahan fungsional.

TABLE I. TABEL PENGUJIAN *BLACK BOX*

| Kode | Fitur / Modul Sistem | Hasil Uji | Status |
|------------------|---|-----------|--------|
| F-01, F-02, F-07 | Manajemen Autentikasi & Profil Pengguna | Sesuai | Valid |
| F-03, F-04, F-05 | Manajemen Klan & Sinkronisasi <i>API (CoC)</i> | Sesuai | Valid |
| F-08, F-09 | Manajemen Turnamen & <i>Bracket generator</i> | Sesuai | Valid |
| F-06, F-10 | Ekosistem Komunitas (<i>Team/Knowledge Hub</i>) | Sesuai | Valid |

2) *Evaluasi Usabilitas (System Usability Scale)*: Pengukuran pengalaman pengguna (*User Experience*) dilakukan menggunakan kuesioner standar *SUS*. Berdasarkan perhitungan skor dari 15 responden, sistem memperoleh skor rata-rata 78.5. Mengacu pada skala interpretasi standar, skor ini menempatkan *Clashub* pada kategori "Good" (*Grade B*) dan status penerimaan (*Acceptability Range*) "Acceptable". Hal ini mengindikasikan bahwa antarmuka sistem dinilai intuitif dan mudah dipelajari oleh komunitas target.

3) *Akseptansi Pengguna Akhir (UAT)*: Validasi kualitatif dengan *Key Opinion Leaders (KOL)* mengonfirmasi terpenuhinya empat indikator kinerja utama yang dirancang sebelumnya:

a) *Efisiensi Waktu*: Terbukti memangkas waktu rekap data perang dari 1 jam (manual) menjadi hitungan detik.

b) *Akurasi Data*: Data profil dan statistik terbukti sinkron secara *real-time* dengan data in-game.

c) *Manfaat AI*: Saran strategi dari Gemini AI dinilai relevan untuk membantu perencanaan serangan awal.

d) *Stabilitas Sistem*: Fitur pembuatan bagan turnamen (bracket) berjalan instan tanpa kegagalan visual.

VI. PEMBAHASAN

Bagian ini membahas implikasi dari hasil implementasi dan pengujian sistem terhadap ekosistem komunitas *Clash of Clans*, serta menganalisis kompleksitas teknis yang mendasari kinerja sistem.

A. Analisis Efisiensi Operasional dan Kompleksitas Arsitektur

Peningkatan efisiensi administrasi klan yang mencapai 90% (sebagaimana divalidasi pada *UAT*) bukanlah sekadar dampak dari digitalisasi pencatatan, melainkan hasil dari orkestrasi data yang canggih di sisi backend. Sebelum adanya *Clashub*, pemimpin klan harus melakukan entri data manual ke *spreadsheet* yang memakan waktu rata-rata 45-60 menit per perang. Sistem mengatasi hambatan ini melalui fitur "*One-Click Sync*". Meskipun tampak sederhana di antarmuka pengguna, di balik layar sistem menjalankan serangkaian proses asinkron yang kompleks:

1) *Validasi Token*: Memastikan permintaan berasal dari sesi pengguna yang sah.

2) *Agregasi Data Eksternal*: Mengambil data mentah (JSON) dari *API* Supercell yang mencakup statistik serangan puluhan anggota.

3) *Transformasi Data (ETL)*: Mengonversi data mentah menjadi format analitik yang dapat dibaca.

4) *Operasi Penulisan Batch*: Menyimpan ratusan dokumen ke database *Firestore* dalam satu transaksi atomik untuk menjamin integritas data. Keberhasilan menyembunyikan kompleksitas ini di balik satu tombol adalah bukti penerapan prinsip *User-Centered Design* yang efektif dalam mengurangi beban kognitif pengguna.

B. Efektivitas dan Batasan Modalitas Kecerdasan Buatan

Integrasi Google Gemini AI memberikan perspektif baru dalam pengambilan keputusan strategis. Berdasarkan evaluasi kualitatif pengguna, AI terbukti efektif berperan sebagai "Asisten Strategis Makro". AI mampu mengidentifikasi kelemahan mendasar berdasarkan data numerik, seperti

menyarankan pasukan udara (*Air Troops*) jika level pertahanan anti-udara musuh rendah.

Namun, penelitian ini juga mengidentifikasi batasan teknis yang signifikan: AI saat ini beroperasi hanya pada modalitas teks/JSON dan belum memiliki kapabilitas visi komputer (*Computer Vision*). Akibatnya, saran yang diberikan bersifat normatif dan kurang presisi dalam aspek taktis mikro. AI tidak dapat memberikan instruksi spasial seperti "Serang dari sisi jam 3 untuk menghindari *Air Sweeper*", karena sistem tidak "melihat" tata letak (*layout*) markas musuh secara visual. Temuan ini menegaskan bahwa peran AI dalam *E-sports* saat ini adalah sebagai pendamping (*copilot*), bukan pengganti total intuisi manusia.

C. Rekayasa Kepercayaan Digital

Masalah asimetri informasi yang memicu fenomena "kutu loncat" (*hoppers*) berhasil dimitigasi melalui pendekatan data ganda pada fitur "*Gamers CV*". Sistem menggabungkan Data Objektif (Statistik Perang dari *API* yang tidak dapat dimanipulasi) dengan Data Subjektif (Ulasan Sosial dari rekan setim).

Kombinasi ini menciptakan efek psikologis baru dalam komunitas: pemain menyadari bahwa perilaku buruk mereka (seperti meninggalkan klan saat perang) akan meninggalkan jejak digital permanen pada skor reputasi mereka. Hal ini secara bertahap mendorong terbentuknya ekosistem yang lebih profesional dan transparan, menjawab hipotesis bahwa teknologi dapat merekonstruksi kepercayaan sosial dalam komunitas daring.

D. Stabilitas Trade-off Kinerja: Latensi vs Stabilitas

Sebagai aplikasi pihak ketiga yang bergantung pada *API* publik, sistem menghadapi tantangan *Rate Limiting*. Untuk menjaga stabilitas akses bagi ribuan pengguna, diterapkan mekanisme *caching* agresif yang menyimpan data respons *API* selama 60 hingga 300 detik. Konsekuensi teknis dari strategi ini adalah munculnya latensi data; informasi di dasbor pengguna mungkin tertinggal sekitar 1-5 menit dibandingkan kejadian live di dalam gim. Meskipun demikian, evaluasi pengguna menunjukkan bahwa trade-off ini dapat diterima karena pengguna lebih memprioritaskan ketersediaan layanan (*availability*) daripada pembaruan detik-per-detik (*real-time consistency*) yang berisiko memicu pemblokiran akses *API*.

VII. PEMBAHASAN

Penelitian ini berhasil mengembangkan "*Clashub*", sebuah platform manajemen ekosistem *E-sports* terintegrasi yang menjawab tantangan fragmentasi alat dan asimetri informasi pada komunitas *Clash of Clans*. Berdasarkan hasil implementasi dan pengujian, dapat disimpulkan hal-hal berikut:

1) *Efisiensi Operasional*: Penerapan algoritma sinkronisasi data hibrida terbukti secara empiris meningkatkan efisiensi administrasi klan hingga lebih dari 90%. Proses rekapitulasi data perang yang sebelumnya memakan waktu 45-60 menit secara manual kini dapat diselesaikan dalam waktu kurang dari 10 detik dengan akurasi data 100%.

2) *Pemulihan Kepercayaan Digital*: Fitur "*Gamers CV*" berhasil menciptakan mekanisme validasi pemain yang objektif. Integrasi data statistik historis dari *API* pengembang gim dengan ulasan sosial terbukti efektif memitigasi risiko

fenomena "kutu loncat" (*hoppers*) dalam proses rekrutmen tim.

3) *Peran Kecerdasan Buatan: Integrasi Google Gemini AI* berfungsi efektif sebagai sistem pendukung keputusan strategis pada tingkat makro (komposisi pasukan), meskipun masih memiliki keterbatasan dalam analisis taktik mikro berbasis visual.

4) *Stabilitas Arsitektur: Penggunaan arsitektur Serverless (Next.js & Firebase)* terbukti mampu menangani kompleksitas orkestrasi data turnamen dan media sosial (*Knowledge Hub*) secara stabil tanpa membebani kinerja perangkat pengguna.

VIII. SARAN

Untuk pengembangan penelitian di masa mendatang, disarankan beberapa peningkatan teknis:

1) *Integrasi Computer Vision: Mengembangkan modul AI* yang mampu memproses input gambar (*screenshot*) tata letak markas lawan, sehingga saran strategi yang diberikan dapat mencakup analisis jalur serangan (*pathing*) yang lebih presisi.

2) *Arsitektur Antrean Pesan (Message Queue):* Menerapkan teknologi seperti *Redis* atau *Google Pub/Sub* untuk menangani sinkronisasi data dalam antrean, guna mengatasi kendala *Rate Limiting API* saat trafik pengguna meningkat drastis.

3) *Ekspansi Mode Turnamen: Mengembangkan algoritma bracket* untuk mendukung format kompetisi liga (*Round Robin*) dan *Double Elimination* yang umum digunakan dalam skema profesional

REFERENCES

- [1] We Are Social, "Digital 2024: Indonesia," Global Digital Insights, 2024.
- [2] Kementerian Kominfo, *Peluang dan Tantangan Industri Gim Nasional*. Jakarta: Kementerian Komunikasi dan Informatika Republik Indonesia, 2024.
- [3] ActivePlayer.io, "Clash of Clans Live Player Count and Statistics," 2025. [Daring]. Tersedia di: <https://activeplayer.io/clash-of-clans/>.
- [4] A. Pratama dan S. Wibowo, "Rancang Bangun Sistem Informasi Manajemen Turnamen E-sports Berbasis Web," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 8, no. 2, hal. 210-218, 2021.
- [5] A. M. F. M. Arham, Y. Sholva, dan M. Muthahhari, "Sistem Informasi Turnamen E-sports dan Warung Kopi Gaming di Kota Pontianak Berbasis Website," *Scientica: Jurnal Ilmiah Sains dan Teknologi*, vol. 2, no. 9, hal. 346-376, 2024.
- [6] W. Yunanri dan A. Measer, "Sistem Informasi Manajemen Event Electronic Sport (E-Sport) Berbasis Web pada Komunitas Esport Indonesia Wilayah Kabupaten Sumbawa," *Jurnal Manajemen Informatika dan Sistem Informasi*, vol. 5, no. 2, hal. 109-115, 2022.
- [7] P. Resnick dan R. Zeckhauser, "Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation system," dalam *The Economics of the Internet and E-commerce*, M. R. Baye, Ed. Amsterdam: Elsevier Science, 2002, vol. 11, hal. 127-157.
- [8] R. S. Pressman dan B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 8th ed. McGraw-Hill Education, 2015.
- [9] Vercel, "Next.js Documentation: App Router," 2024. [Daring]. Tersedia di: <https://nextjs.org/docs>.
- [10] Google Developers, "Cloud Firestore Documentation," 2024. [Daring]. Tersedia di: <https://firebase.google.com/docs/firestore>.
- [11] Supercell, "Clash of Clans API Documentation," 2024. [Daring]. Tersedia di: <https://developer.clashofclans.com/>.
- [12] Google Developers, "Gemini API Documentation," 2024. [Daring]. Tersedia di: <https://AI.google.dev/docs>.
- [13] Google Developers, "YouTube Data API Reference," 2024. [Daring]. Tersedia di: <https://developers.google.com/youtube/v3>.
- [14] S. Nidhra dan J. Dondeti, "Black Box and White Box Testing Techniques - A Literature Review," *International Journal of Embedded Systems and Applications (IJESA)*, vol. 2, no. 2, hal. 29-50, 2012.
- [15] J. Brooke, "SUS-A quick and dirty usability scale," dalam *Usability Evaluation in Industry*, P. W. Jordan, et al., Eds. London: Taylor & Francis, 1996, hal. 189-194.