

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Game *Multiplayer*

Game *multiplayer* adalah jenis permainan video yang memungkinkan beberapa pemain untuk berinteraksi dan bermain secara bersamaan dalam satu lingkungan permainan, baik melalui jaringan lokal maupun internet. Berbeda dengan game *single-player* yang hanya melibatkan satu pemain berinteraksi dengan sistem komputer, game *multiplayer* menciptakan pengalaman sosial di mana pemain dapat berkompetisi atau berkolaborasi satu sama lain secara *real-time* (Smed dkk., 2002).

2.1.1 Arsitektur Jaringan dalam Game *Multiplayer*

Secara umum ada beberapa perbedaan operasional dan teknis antara *multiplayer* yang berjalan di internet (*online*) dan yang berjalan pada jaringan lokal. Dari sisi topologi dan latensi game *online* sering melibatkan pemain yang tersebar luas sehingga cenderung mengalami latensi yang lebih besar dan variabel (*jitter*) akibat rute jaringan panjang dan kondisi internet publik. Sebaliknya, game pada jaringan lokal seperti LAN atau Wi-Fi Direct berada dalam domain jaringan yang terbatas sehingga rata-rata latensinya lebih rendah dan lebih stabil namun tetap dipengaruhi oleh kualitas sinyal nirkabel dan interferensi lokal. Hal ini memengaruhi desain mekanisme toleransi *delay* dan frekuensi update state (Pantel & Wolf, 2002).

Pada bagian dependensi infrastruktur game *online* sering bergantung pada server pusat atau layanan pihak ketiga untuk *matchmaking* dan persistensi data,

sementara game lokal dapat berjalan tanpa infrastruktur eksternal jika menggunakan model *peer-to-peer* atau host lokal misalnya salah satu perangkat sebagai server. Teknologi seperti Wi-Fi Direct juga memungkinkan komunikasi *device-to-device* tanpa *access point* tradisional (Khan dkk., 2017).

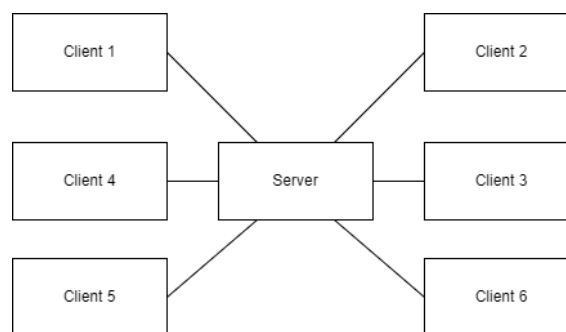
2.1.2 Metrik Performa Game *Multiplayer*

Evaluasi performa game *multiplayer* memerlukan berbagai metrik untuk mengukur kualitas pengalaman bermain. Berdasarkan penelitian (Pantel & Wolf, 2002) metrik utama yang digunakan mencakup latensi yang merepresentasikan waktu yang diperlukan untuk data berpindah dari sumber ke tujuan, di mana latensi yang rendah sangat penting untuk menjaga responsivitas dalam game. Dalam penelitian mereka tentang dampak *delay* pada game *multiplayer real-time*, ditemukan bahwa *delay* yang berbeda memiliki pengaruh yang berbeda pula terhadap performa pemain tergantung pada genre game. *Throughput* menjadi indikator penting yang menunjukkan jumlah data yang berhasil ditransmisikan per satuan waktu, mencerminkan efisiensi penggunaan *bandwidth* jaringan yang tersedia. (Beigbeder dkk., 2004) mengukur *packet loss* sebagai persentase paket data yang hilang selama transmisi, yang dapat mempengaruhi kualitas sinkronisasi game secara signifikan dalam penelitian mereka tentang efek *loss* dan latensi pada performa user. *Jitter* atau variasi dalam *delay* pengiriman paket juga perlu diperhatikan karena dapat menyebabkan ketidakstabilan dalam *gameplay* yang mengganggu pengalaman pemain. Selain metrik jaringan, penggunaan resource sistem seperti konsumsi CPU dan memori juga menjadi faktor kritis yang

mempengaruhi performa keseluruhan sistem dan kemampuan untuk menjalankan game dengan lancar tanpa hambatan (Suznjevic dkk., 2014).

2.2 Arsitektur *Client-Server*

Arsitektur *client-server* adalah model komunikasi jaringan di mana terdapat satu entitas pusat yang berperan sebagai server dan beberapa entitas pengguna yang berperan sebagai klien. Server bertugas memproses logika utama, menjaga konsistensi *state* permainan, serta menyebarkan pembaruan ke seluruh klien. Klien hanya bertugas mengirimkan *input* pemain ke server, lalu menerima hasil pembaruan *state* (Hakkun dkk., 2022).



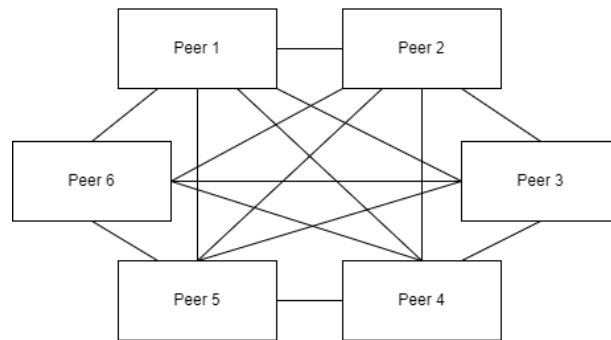
Gambar 2.1 Arsitektur *Client Server*

Model ini menawarkan kontrol terpusat dan sinkronisasi yang lebih sederhana, sehingga sering dipilih pada game *multiplayer real-time* berbasis internet. Namun, kelemahannya adalah server menjadi *bottleneck* jika server *overload* atau *down*, maka seluruh permainan akan terganggu, arsitektur *client-server* menjadi pilihan dominan bagi banyak penyedia game online karena memberikan kontrol administrasi yang lebih baik, yang memungkinkan perusahaan mengelola informasi akun dan pembayaran pemain secara terpusat. Selain keuntungan administratif, arsitektur ini secara teknis mencegah pemain melakukan

kecurangan (*cheating*) karena seluruh logika permainan dieksekusi di server pusat, sehingga pemain tidak memiliki akses untuk memanipulasi status permainan secara langsung di terminal mereka. Dibandingkan dengan arsitektur *peer-to-peer*, *client-server* juga dianggap lebih mudah diimplementasikan karena pengembang tidak perlu membangun mekanisme rumit seperti penemuan *peer* (*peer-discovery*), pengurutan *event* terdistribusi, ataupun pencegahan kecurangan yang kompleks (Khan dkk., 2012). Meskipun memiliki keunggulan dalam kontrol, arsitektur *client-server* menghadapi tantangan signifikan terkait kebutuhan *bandwidth* dan skalabilitas. Server dituntut untuk mengirimkan pesan pembaruan sesering mungkin agar dunia virtual dapat ditampilkan secara konsisten di sisi klien, yang dapat meningkatkan beban *bandwidth* secara drastis seiring bertambahnya jumlah pemain (Khan dkk., 2012).

2.3 Arsitektur *Peer-to-Peer*

Pada arsitektur *peer-to-peer* setiap *node* memiliki peran yang setara. Semua *node* dapat bertindak sebagai pengirim maupun penerima data, sehingga tidak ada entitas pusat. Model ini memiliki kelebihan berupa reduksi beban server dan pemanfaatan *bandwidth* kolektif antar *peer*. Arsitektur P2P memiliki keunggulan skalabilitas dan *fault-tolerance*, namun menghadapi tantangan pada sinkronisasi *state* serta keamanan (Eng Keong Lua dkk., 2005).



Gambar 2.2 Arsitektur Peer-to-peer

Dalam konteks aplikasi kolaboratif seperti aplikasi menggambar berbasis *peer-to-peer* komunikasi antar perangkat menjadi lebih ringan karena tidak ada beban server tunggal. Namun, sinkronisasi data antar *peer* menimbulkan kompleksitas tambahan untuk menjaga konsistensi (Sulu dkk., 2024). Meskipun P2P dapat mengurangi latensi antar *node* dalam jaringan lokal, arsitektur ini rentan terhadap inkonsistensi *state* jika *packet loss* atau *delay* terjadi, sehingga perlu mekanisme tambahan seperti *lockstep* atau *dead-reckoning* (Buyukkaya dkk., 2009).

2.4 Pengujian Aplikasi

Pengujian perangkat lunak merupakan aktivitas penting dalam pengembangan perangkat lunak yang bertujuan untuk menganalisis perangkat lunak guna menemukan perbedaan antara kondisi yang diharapkan dengan kondisi yang ada. Pengujian perangkat lunak dilakukan untuk membangun kualitas perangkat lunak yang baik dengan cara mengidentifikasi *defect*, *flaw*, atau error dalam kode aplikasi (Khan & Khan, 2012). Penelitian menunjukkan bahwa hampir 30% dari total usaha pengembangan perangkat lunak digunakan untuk melakukan

aktivitas pengujian, sehingga aktivitas pengujian memainkan peran vital dalam mengembangkan perangkat lunak yang berkualitas.

Pengujian perangkat lunak dapat dikategorikan menjadi beberapa jenis berdasarkan pendekatan dan tujuannya, salah satu kategorisasi utama adalah *white box testing* dan *black box testing*. *White box testing* merupakan teknik pengujian struktural yang merancang test case berdasarkan informasi yang diperoleh dari *source code*, dimana penguji seringkali pengembang dari kode tersebut mengetahui seperti apa kode terlihat dan menulis *test case* dengan mengeksekusi *method* dengan parameter tertentu. *Black box testing* merupakan metode pengujian fungsional yang tidak memerlukan pengetahuan tentang struktur internal perangkat lunak (Nidhra & Dondeti, 2012).

2.4.1 Black Box Testing

Black box testing digunakan sebagai metode pengujian perangkat lunak yang berfokus pada pemeriksaan fungsionalitas sistem tanpa memperhatikan struktur internal atau kode program yang digunakan. Penguji hanya mengamati masukan yang diberikan dan keluaran yang dihasilkan oleh sistem untuk menentukan apakah perilaku sistem telah sesuai dengan spesifikasi yang ditentukan, pendekatan ini menekankan pada validasi fungsi sistem dari sudut pandang pengguna akhir, sehingga sangat relevan untuk menguji aplikasi interaktif seperti game (Mustaqbal dkk., 2015).

Black box testing banyak digunakan untuk menguji aplikasi berbasis game dan sistem interaktif. *black box testing* terbukti efektif untuk memastikan bahwa setiap fungsi sistem berjalan sesuai dengan kebutuhan yang telah didefinisikan,

tanpa memerlukan pemahaman terhadap struktur internal perangkat lunak. *black box testing* juga telah terbukti mampu mengidentifikasi kesalahan fungsional pada aplikasi berbasis multimedia dan game secara efisien (Hendra & Kristianto, 2024).

Pada penelitian ini *black box testing* digunakan untuk menguji fungsionalitas utama game multiplayer lokal, seperti proses koneksi pemain, mekanisme lobby, sinkronisasi permainan, serta transisi antar babak permainan. Dengan memastikan bahwa seluruh fitur berjalan dengan benar melalui pengujian fungsional, maka hasil pengujian performa jaringan yang dilakukan pada tahap selanjutnya dapat dianggap valid dan tidak terpengaruh oleh kesalahan logika atau bug pada aplikasi. Salah satu teknik dalam *black box testing* yaitu *Boundary Value Analysis*.

2.4.2 *Boundary Value Analysis*

Salah satu teknik pengujian yang sering digunakan dalam *black box testing* adalah *Boundary Value Analysis* (BVA). Teknik ini berfokus pada pengujian nilai batas dari suatu input, karena kesalahan perangkat lunak sering terjadi pada nilai *minimum*, *maksimum*, atau nilai yang berada di sekitar batas tersebut. *Boundary Value Analysis* bertujuan untuk meningkatkan efektivitas pengujian dengan jumlah kasus uji yang relatif sedikit namun memiliki tingkat deteksi kesalahan yang tinggi (Yulistina dkk., 2020).

Boundary Value Analysis digunakan untuk menguji batas valid dan tidak valid dari suatu input sistem guna memastikan bahwa sistem mampu menangani kondisi ekstrem dengan benar. Teknik BVA sangat efektif dalam menguji sistem

yang memiliki batasan input numerik, seperti batas jumlah pengguna, waktu proses, dan nilai ambang tertentu (Ahrizal dkk., 2020). Dalam game multiplayer lokal yang dikembangkan pada penelitian ini, *Boundary Value Analysis* digunakan untuk menguji kondisi batas, seperti jumlah minimum pemain untuk memulai permainan, batas waktu menjawab soal, serta kondisi ketika hanya satu pemain yang terhubung di dalam lobby. Pengujian terhadap nilai batas ini penting untuk memastikan bahwa sistem tidak mengalami kegagalan fungsi pada kondisi ekstrem yang mungkin terjadi selama permainan berlangsung.

Penerapan *Boundary Value Analysis* dalam pengujian fungsional memberikan kontribusi penting dalam meningkatkan keandalan sistem sebelum dilakukan pengujian performa jaringan. Dengan memastikan bahwa aplikasi telah lolos pengujian pada kondisi batas, maka hasil analisis performa arsitektur *client-server* dan *peer-to-peer* dapat merepresentasikan karakteristik jaringan secara lebih akurat dan objektif.

2.5 *State of the Art*

Penelitian mengenai game multiplayer telah berkembang pesat dalam beberapa tahun terakhir, dengan berbagai pendekatan arsitektur jaringan yang diimplementasikan oleh peneliti terdahulu. Kajian terhadap literatur eksisting menunjukkan bahwa mayoritas penelitian berfokus pada implementasi arsitektur *client-server* dengan menggunakan berbagai framework dan teknologi. Beberapa peneliti telah mengembangkan game multiplayer berbasis Unity, di antaranya game tower defense dan Beat 'Em Ups menggunakan Photon Unity Networking (PUN) (Sarwodi dkk., 2020; Thuan & Raka, 2021), game first-person shooter (FPS) 3D

berbasis Unity Network (UNET) (Ramadhan dkk., 2020), serta game 2D shooting multiplayer menggunakan Unity Netcode for GameObjects (Geubrina dkk., 2024). Selain itu, game edukasi Snake and Ladder berbasis Java Socket (Syauqi A'la & Hijjah Permata Rahmatul, 2020), game FPS multiplayer menggunakan Photon Cloud (Wahyudi dkk., 2022), serta game horor co-op dengan Unity Multiplayer (Pande dkk., 2023). Meskipun penelitian-penelitian tersebut berhasil mengimplementasikan arsitektur client-server pada berbagai genre game, eksplorasi terhadap arsitektur peer-to-peer (P2P) masih sangat terbatas. Hanya ditemukan satu penelitian yang menggunakan pendekatan P2P, yaitu pengembangan aplikasi menggambar kolaboratif (Sulu dkk., 2024), namun bukan dalam konteks game multiplayer.

Dari aspek metode evaluasi, penelitian-penelitian terdahulu menunjukkan kecenderungan yang fokus pada aspek implementasi dan fungsionalitas sistem seperti sinkronisasi data, mekanisme koneksi, dan fitur matchmaking. Evaluasi performa secara komprehensif dan terukur jarang dilakukan atau tidak menjadi fokus utama dalam penelitian-penelitian tersebut. Sebagian besar peneliti lebih menekankan pada keberhasilan implementasi fitur-fitur game multiplayer daripada mengukur dan menganalisis metrik performa jaringan secara kuantitatif. Aspek performa yang seharusnya menjadi pertimbangan kritis dalam pengembangan game multiplayer belum mendapat perhatian yang memadai. Penelitian yang diusulkan berupaya mengisi kekosongan ini dengan melakukan perbandingan performa empiris menggunakan metrik terukur yang mencakup latensi dalam satuan milidetik, throughput dalam kilobit per second atau megabit per second, packet loss

dalam persentase, penggunaan CPU dalam persentase, serta penggunaan memori dalam megabyte.

Lingkup jaringan yang digunakan dalam penelitian terdahulu terbagi menjadi dua kategori utama, yaitu jaringan lokal (LAN) dan jaringan internet atau cloud. Sebagian penelitian mengimplementasikan game multiplayer pada jaringan lokal (Syauqi A'la & Hijjah Permata Rahmatul, 2020), sementara sebagian besar penelitian lainnya menggunakan infrastruktur cloud atau internet (Geubrina dkk., 2024; Pande dkk., 2023; Ramadhan dkk., 2020; Sarwodi dkk., 2020; Thuan & Raka, 2021; Wahyudi dkk., 2022). Selain itu, terdapat juga pendekatan P2P yang diterapkan pada domain non-game, yaitu aplikasi menggambar kolaboratif (Sulu dkk., 2024). Penggunaan jaringan internet dalam penelitian-penelitian tersebut menghadirkan variabel eksternal yang sulit dikontrol, seperti kualitas koneksi internet, jarak geografis antarpemain, dan variasi bandwidth, yang dapat memengaruhi hasil pengukuran performa. Penelitian ini akan menggunakan jaringan lokal (LAN) secara eksklusif untuk mengeliminasi variabel-variabel eksternal tersebut, sehingga perbandingan performa antara arsitektur client-server dan peer-to-peer dapat dilakukan secara murni tanpa pengaruh dari faktor-faktor di luar kendali arsitektur itu sendiri.

Jumlah pemain yang menjadi fokus penelitian terdahulu sangat bervariasi, dengan rentang antara dua hingga delapan pemain. Mayoritas penelitian tidak memberikan spesifikasi yang jelas mengenai jumlah pemain optimal atau tidak secara khusus fokus pada skalabilitas untuk jumlah pemain tertentu. Beberapa penelitian bahkan lebih menekankan pada kemampuan sistem untuk mendukung

jumlah pemain yang banyak secara simultan. Berbeda dengan penelitian terdahulu, penelitian ini akan membatasi fokus pada skenario dua pemain dan lima pemain. Pembatasan ini dilakukan secara deliberatif untuk memungkinkan perbandingan yang terkontrol dan representatif. Dengan fokus pada konfigurasi yang terkontrol, analisis performa dapat dilakukan dengan lebih mendalam dan hasil pengukuran akan lebih konsisten.

Framework dan library yang digunakan untuk implementasi networking dalam penelitian terdahulu sangat beragam, khususnya untuk arsitektur client-server. Penelitian-penelitian tersebut menggunakan Photon Unity Networking (PUN), Unity Network (UNET), Unity Netcode for GameObjects, Java Socket, Photon Cloud, dan Unity Multiplayer sebagai solusi networking sedangkan untuk arsitektur peer-to-peer, hanya ditemukan implementasi custom yang dikembangkan oleh untuk aplikasi non-game (Sulu dkk., 2024). Penelitian ini akan menggunakan Mirror sebagai framework untuk implementasi arsitektur client-server dan NET Framework sebagai library untuk implementasi arsitektur peer-to-peer. Pemilihan Mirror didasarkan pada fakta bahwa framework ini merupakan successor dari UNET yang telah deprecated, memiliki performa yang lebih baik, dan memiliki komunitas pengembang yang aktif. Sementara itu, implementasi TCP/IP *native* dipilih karena memberikan kontrol penuh terhadap logika jaringan dan tidak memerlukan dependensi library pihak ketiga, sehingga implementasinya lebih transparan dan mudah dikustomisasi sesuai kebutuhan penelitian. Pendekatan ini juga kompatibel secara *native* dengan Unity Engine melalui penggunaan NET Framework standar.

2.5.1 Literature Review

Implementasi *multiplayer* berbasis jaringan lokal telah dilakukan oleh beberapa penelitian terdahulu, namun kajian tersebut lebih menitikberatkan pada keberhasilan fungsional sistem dibandingkan analisis performa jaringan secara kuantitatif. Parameter penting seperti latensi jaringan, *throughput*, *packet loss*, serta konsumsi sumber daya komputasi belum menjadi fokus utama dalam evaluasi yang dilakukan, secara lebih rinci literature review disajikan pada tabel 2.1.

Tabel 2.1 Literature Review

No	Judul/Penulis (Tahun)	Metodologi / Arsitektur	Hasil / Temuan Utama
1	Penerapan Multiplayer Pada Gim Tower Defense Menggunakan Photon Unity (Sarwodi dkk., 2020)	Client-Server (Photon Unity Networking / PUN)	Penelitian ini membahas mengenai penerapan sistem multiplayer pada game tower defense berbasis Android menggunakan Photon Unity Networking (PUN). Penelitian ini menunjukkan bahwa PUN mampu mendukung sinkronisasi permainan secara real-time antar pemain melalui mekanisme Object Synchronization, Remote Procedure Call (RPC), dan Custom Properties, sehingga objek permainan seperti tower dan musuh dapat tersinkronisasi dengan baik. Dari sisi performa, hasil pengujian menunjukkan bahwa mode multiplayer tidak memberikan penurunan signifikan terhadap performa rendering, dengan rata-rata Frame Per Second (FPS) berada pada kisaran 58–60 FPS, baik pada mode singleplayer maupun multiplayer. Pengujian <i>delay</i> menggunakan fitur Photon Lag Simulation mengindikasikan bahwa pada kondisi

No	Judul/Penulis (Tahun)	Metodologi / Arsitektur	Hasil / Temuan Utama
			simulasi lag sebesar 150 ms mulai terjadi gangguan sinkronisasi berupa keterlambatan dan pergerakan objek yang tidak mulus. Selain aspek teknis, penelitian ini juga mengevaluasi pengalaman pengguna menggunakan Game Experience Questionnaire (GEQ) pada aspek kehadiran sosial, yang menunjukkan adanya perbedaan tingkat empati dan keterlibatan pemain berdasarkan gender dan pengalaman bermain. Namun demikian, penelitian ini tidak melakukan pengukuran performa jaringan secara langsung pada jaringan nyata, tidak mengevaluasi metrik jaringan seperti latensi aktual, throughput, dan packet loss, serta tidak membandingkan arsitektur jaringan dengan pendekatan lain selain client-server berbasis layanan cloud.
2	Pengembangan Teknologi Game Indonesia Untuk Permainan First Person Shooter (FPS) 3D Multiplayer "Code To Shoot" Menggunakan Unity Network (UNET) Berbasis Mobile (Ramadhan dkk., 2020)	Client-Server (Unity Network / UNET)	Penelitian ini mengembangkan game First Person Shooter (FPS) multiplayer 3D berjudul <i>Code To Shoot</i> pada platform Android dengan memanfaatkan Unity Network (UNET) sebagai framework jaringan. Implementasi multiplayer dilakukan menggunakan UNET High Level API untuk mengelola koneksi, sinkronisasi objek dinamis seperti pemain dan peluru, serta mekanisme pembuatan dan penggabungan ruang permainan secara otomatis tanpa input alamat IP manual. Arsitektur jaringan yang digunakan adalah client-server dengan pendekatan

No	Judul/Penulis (Tahun)	Metodologi / Arsitektur	Hasil / Temuan Utama
			host-as-player, di mana satu pemain berperan sebagai host sekaligus pemain aktif. Evaluasi sistem dilakukan menggunakan pengujian black box yang menunjukkan bahwa seluruh fungsi utama game, termasuk login, pengelolaan room, kontrol karakter, dan interaksi antar pemain, berjalan dengan baik dan stabil pada pengujian hingga empat pemain. Namun, penelitian ini tidak melakukan pengukuran performa jaringan secara kuantitatif seperti latensi, throughput, dan packet loss, serta tidak menganalisis konsumsi sumber daya sistem. Selain itu, penggunaan UNET yang telah tidak lagi dikembangkan dan ketergantungan pada jaringan internet publik membatasi relevansi hasil penelitian terhadap studi perbandingan performa arsitektur jaringan pada lingkungan lokal.
3	Synchronous Multiplayer Shooting in the 2D Game War Cops Using Unity Engine (Geubrina dkk., 2024)	Client-Server (Unity Netcode / Network Transform)	Penelitian ini mengembangkan game 2D shooting multiplayer synchronous berjudul <i>War Cops</i> menggunakan Unity Engine dengan dukungan Photon Unity Network sebagai framework networking. Arsitektur jaringan yang digunakan adalah client-server berbasis Photon Cloud, dengan fokus pada sinkronisasi real-time terhadap posisi pemain, aksi tembakan, dan status permainan. Mekanisme sinkronisasi dilakukan menggunakan Network Transform untuk posisi dan rotasi pemain, serta Remote Procedure Call (RPC) untuk

No	Judul/Penulis (Tahun)	Metodologi / Arsitektur	Hasil / Temuan Utama
			event tembakan dan pembaruan skor. Evaluasi performa dilakukan melalui pengukuran ping response ke server Photon dengan hasil rata-rata sebesar 201 ms, yang menunjukkan bahwa sistem masih berada dalam batas playable untuk game casual meskipun kurang optimal untuk gameplay shooting yang membutuhkan responsivitas tinggi. Pengujian fungsionalitas menggunakan black box testing menunjukkan bahwa seluruh fitur utama game berjalan dengan baik pada pengujian hingga empat pemain. Namun, penelitian ini hanya mengukur satu metrik performa jaringan, bergantung pada infrastruktur cloud dan koneksi internet publik, serta tidak menganalisis metrik jaringan lain maupun konsumsi sumber daya sistem, sehingga belum memberikan gambaran komprehensif mengenai performa arsitektur jaringan pada lingkungan yang terkontrol.
4	Gim Multiplayer Beat 'Em Ups Menggunakan Photon Unity Networking (Thuan & Raka, 2021)	Client-Server (Photon Unity Networking)	Penelitian ini mengembangkan game multiplayer bergenre Beat 'Em Ups berbasis Unity Game Engine dengan memanfaatkan Photon Unity Networking (PUN) sebagai framework jaringan yang dijalankan melalui infrastruktur cloud server milik Photon. Arsitektur jaringan yang diterapkan adalah client-server, di mana server PUN berperan sebagai perantara untuk mengelola koneksi, sinkronisasi state

No	Judul/Penulis (Tahun)	Metodologi / Arsitektur	Hasil / Temuan Utama
			permainan, serta distribusi informasi objek antarpemain melalui mekanisme networked instantiate, request ownership, dan photon view. Sistem mendukung fitur pembuatan dan penggabungan room, pemilihan karakter, serta mode co-op online untuk beberapa pemain secara bersamaan. Pengujian fungsional menggunakan black box testing menunjukkan bahwa seluruh fitur lobby, room, dan gameplay berjalan sesuai dengan yang diharapkan. Evaluasi kepuasan pengguna dilakukan melalui survei terhadap 23 responden, dengan hasil 78,26% menyatakan sangat puas terhadap gameplay, 82,61% memberikan penilaian sangat baik terhadap pengalaman bermain multiplayer, dan 91,3% menilai antarmuka pengguna sangat baik. Penelitian ini tidak melakukan pengukuran performa jaringan secara kuantitatif seperti latensi, packet loss, maupun throughput, sehingga tidak memberikan gambaran komprehensif mengenai karakteristik performa arsitektur client-server yang digunakan, dan hasilnya lebih bersifat sebagai validasi fungsional serta evaluasi pengalaman pengguna.

No	Judul/Penulis (Tahun)	Metodologi / Arsitektur	Hasil / Temuan Utama
5	Design of Networked Multiplayer Snake and Ladder Educational Game Based on Hash Map and Vector Data Structure (Syauqi A'la & Hijjah Permata Rahmatul, 2020)	Client-Server (Java Socket)	Penelitian ini mengembangkan game edukasi multiplayer <i>Snake and Ladder</i> berbasis jaringan lokal menggunakan Java Socket Programming dengan arsitektur client-server. Sistem dirancang secara custom dengan server berperan sebagai moderator yang mengelola koneksi dan pertukaran data antar pemain menggunakan protokol TCP. Struktur data HashMap digunakan untuk memetakan username dengan socket pemain, sedangkan Vector dimanfaatkan untuk mekanisme broadcasting paket data ke anggota group yang sama. Evaluasi performa difokuskan pada kemampuan server dalam menangani beberapa grup pemain secara bersamaan, dengan hasil menunjukkan server performance sangat bergantung pada spesifikasi hardware, khususnya kombinasi processor dan RAM. Server 1 (AMD E1-1200, 2GB RAM) dan Server 2 (AMD E-300, 4GB RAM) mencapai success rate 77.7% dalam mengirim paket data. Server 3 (Intel i3-4030U, 4GB RAM) mencapai 83.3%, dan Server 4 (Intel i7-7700HQ, 8GB RAM) mencapai 88.8%. Kegagalan pengiriman sering terjadi pada pemain pertama dalam group, yang diidentifikasi disebabkan oleh paket data awal yang tidak dapat diterima dan diproses dengan baik. Terdapat juga kejadian rare dimana paket data incomplete diterima client akibat

No	Judul/Penulis (Tahun)	Metodologi / Arsitektur	Hasil / Temuan Utama
			data corruption dalam proses encoding/decoding. Meskipun pengujian fungsional menunjukkan seluruh fitur berjalan dengan baik, penelitian ini hanya mengukur performa secara agregat melalui success rate pengiriman data tanpa menganalisis metrik jaringan secara rinci, serta tidak membahas konsumsi sumber daya di sisi klien, sehingga belum memberikan gambaran komprehensif mengenai performa arsitektur jaringan untuk game multiplayer real-time.
6	Action Shooter Synchronous Multiplayer Game Menggunakan Photon Cloud Network Multiplatform (Wahyudi dkk., 2022)	Client-Server / Relay (Photon Cloud)	Penelitian ini mengembangkan game action shooter multiplayer synchronous multiplatform menggunakan Unity Engine dengan dukungan Photon Cloud Network sebagai framework networking. Arsitektur jaringan yang diterapkan adalah client-server berbasis cloud, yang memungkinkan pemain dari berbagai platform, seperti Windows dan Android, untuk terhubung dalam satu sesi permainan. Sistem mendukung fitur pembuatan dan penggabungan room, pemilihan tim, pemilihan map, serta sinkronisasi aksi permainan secara real-time. Evaluasi sistem dilakukan melalui pengujian black box yang menunjukkan seluruh fungsi utama game berjalan dengan baik pada pengujian beberapa pengguna. Meskipun Photon Cloud mampu menyediakan konektivitas yang stabil pada kondisi jaringan yang baik,

No	Judul/Penulis (Tahun)	Metodologi / Arsitektur	Hasil / Temuan Utama
			penelitian ini tidak melakukan pengukuran performa jaringan secara kuantitatif dan sangat bergantung pada layanan pihak ketiga serta koneksi internet publik. Selain itu, penelitian tidak menganalisis konsumsi sumber daya sistem maupun membandingkan arsitektur jaringan dengan pendekatan lain, sehingga hasilnya terbatas pada validasi fungsional sistem.
7	Implementasi Komunikasi Peer-to-Peer dengan Mengembangkan Aplikasi Menggambar Kolaboratif (Sulu dkk., 2024)	Peer-to-Peer (P2P)	Penelitian ini mengembangkan aplikasi menggambar kolaboratif dengan menerapkan arsitektur komunikasi peer-to-peer (P2P) murni menggunakan socket programming pada bahasa pemrograman Python. Setiap node dalam sistem berperan sebagai client dan server secara bersamaan, sehingga komunikasi data dilakukan secara langsung antar pengguna tanpa server terpusat. Pendekatan ini memungkinkan sinkronisasi aktivitas menggambar secara real-time pada jumlah pengguna terbatas, dengan mekanisme broadcasting perubahan kanvas ke seluruh peer yang terhubung. Hasil penelitian menunjukkan bahwa arsitektur P2P dapat berjalan dengan baik untuk kolaborasi sederhana dengan dua hingga tiga pengguna. Namun, penelitian ini tidak melakukan pengukuran performa jaringan secara kuantitatif, memiliki keterbatasan pada mekanisme <i>discovery</i> dan <i>scalability</i>, serta belum mengkaji

No	Judul/Penulis (Tahun)	Metodologi / Arsitektur	Hasil / Temuan Utama
			penerapan arsitektur P2P pada aplikasi dengan kebutuhan sinkronisasi tinggi seperti game multiplayer real-time, sehingga ruang pengembangan dan evaluasi lebih lanjut masih terbuka.
8	Pembuatan Horor Game 3D Multiplayer Co-op "Calonarang" Berbasis Mobile (Pande dkk., 2023)	Client-Server (Unity Multiplayer)	Penelitian ini mengembangkan game horor 3D multiplayer cooperative berjudul <i>Calonarang</i> pada platform Android menggunakan Unity Engine dengan pendekatan arsitektur client-server. Game ini mengangkat cerita rakyat Bali sebagai upaya preservasi budaya, dengan dukungan fitur multiplayer hingga empat pemain dalam satu sesi permainan. Implementasi sistem mencakup mekanisme pembuatan dan penggabungan room, ruang tunggu, pemilihan karakter, serta gameplay kooperatif dalam lingkungan 3D. Evaluasi sistem dilakukan melalui pengujian black box yang menunjukkan bahwa fungsi utama game berjalan dengan baik, serta pengujian beta menggunakan kuesioner yang menghasilkan respons positif terhadap aspek visual, gameplay, dan penyampaian cerita. Namun, penelitian ini tidak melakukan pengukuran performa teknis jaringan maupun konsumsi sumber daya sistem, serta tidak memberikan analisis mendalam terhadap aspek networking, sehingga hasil penelitian lebih menitikberatkan pada konten dan pengalaman

No	Judul/Penulis (Tahun)	Metodologi / Arsitektur	Hasil / Temuan Utama
			pengguna dibandingkan evaluasi performa arsitektur jaringan.
9.	Penerapan Multiplayer Pada Aplikasi Permainan Android Menggunakan Photon PUN dengan Metode Prototyping (Lumombo & Jones, 2023)	Client-server (Photon Unity Networking)	Penelitian ini mengembangkan game 3D platform multiplayer berbasis Android menggunakan Unity 3D dengan dukungan Photon Unity Networking (PUN) dan metode prototyping. Implementasi multiplayer difokuskan pada sinkronisasi pergerakan karakter antar pemain menggunakan mekanisme Object Synchronization, yang mencakup sinkronisasi posisi, rotasi, dan animasi dasar. Evaluasi sistem dilakukan melalui pengujian black box yang menunjukkan sebagian besar fungsi permainan berjalan dengan baik, serta pengujian kualitatif terhadap sejumlah pemain yang menilai sinkronisasi antar pemain sudah cukup responsif meskipun masih ditemukan keterlambatan pada kondisi tertentu. Selain itu, pengujian performa hanya terbatas pada pengukuran Frame Per Second (FPS) pada berbagai spesifikasi perangkat. Penelitian ini tidak melakukan pengukuran performa jaringan secara kuantitatif dan sangat bergantung pada infrastruktur cloud, sehingga hasilnya belum memberikan gambaran objektif mengenai pengaruh arsitektur jaringan terhadap performa game multiplayer.

No	Judul/Penulis (Tahun)	Metodologi / Arsitektur	Hasil / Temuan Utama
10.	Game Quiz Multiplayer Menggunakan Photon Unity Networking (Adriansyah Ikhsanul Iqbal & Paramitha Nerisafitra, 2025)	Client-server (Photon Unity Networking 2)	Penelitian ini mengembangkan game kuis multiplayer edukatif berbasis Android menggunakan Unity Engine dengan dukungan Photon Unity Networking 2 (PUN 2) sebagai solusi multiplayer real-time. Sistem multiplayer memanfaatkan mekanisme pembuatan dan penggabungan room, sinkronisasi skor antar pemain, serta pengelolaan alur permainan berbasis timer. Evaluasi aplikasi dilakukan menggunakan instrumen Game User Experience Satisfaction Scale (GUESS-18) dengan Skor GUESS total adalah 33.81 dari maksimal 40, menunjukkan tingkat kepuasan pengguna yang tinggi sebesar 84.5% terhadap aspek usability, visual, dan interaksi sosial. Namun, penelitian ini tidak melakukan pengukuran performa teknis jaringan maupun konsumsi sumber daya sistem, serta sepenuhnya bergantung pada layanan cloud dan koneksi internet, sehingga hasilnya lebih berfokus pada pengalaman pengguna subjektif dan belum memberikan analisis kuantitatif mengenai performa arsitektur jaringan pada game multiplayer.

Berdasarkan tabel 2.1 dapat diketahui bahwa penelitian terdahulu didominasi oleh penggunaan arsitektur *Client-Server*, sementara arsitektur *Peer-to-Peer* (P2P) masih minim dieksplorasi dalam konteks pengembangan game.

Mayoritas penelitian mengadopsi model *Client-Server*, memanfaatkan teknologi *cloud-based* seperti Photon Unity Networking (PUN) atau Photon Cloud yang menawarkan kemudahan implementasi, skalabilitas tinggi, dan kemampuan *cross-platform* untuk mengelola koneksi, sinkronisasi, dan sistem lobby/ruangan. Sebagai contoh, studi mengenai Tower Defense dan Action Shooter secara eksplisit menggunakan Photon untuk sinkronisasi data gameobject dan manajemen server. Selain itu, Unity Network (UNET) juga menjadi solusi *Client-Server built-in* dari Unity yang relevan, khususnya untuk game *First Person Shooter* (FPS) yang membutuhkan sinkronisasi posisi yang kompleks.

Arsitektur *Peer-to-Peer* hanya teridentifikasi dalam satu penelitian dan fokusnya adalah pada aplikasi menggambar kolaboratif, bukan game. Penelitian tersebut mengimplementasikan komunikasi P2P menggunakan Python socket programming dan berhasil menunjukkan kapabilitas komunikasi real-time langsung antar peer. Namun, studi ini juga menyoroti tantangan utama P2P, yaitu penemuan peer yang masih manual dan masalah skalabilitas yang membatasi pengguna optimal hingga tiga orang. Dalam aspek teknologi, Unity Engine muncul sebagai platform paling populer, digunakan dalam sebagian besar studi karena dukungan native yang kuat untuk networking dan kompatibilitasnya dengan berbagai framework jaringan seperti Photon. Metode pengujian yang digunakan bervariasi, meliputi pengujian pengalaman bermain (*Game Experience Questionnaire*), pengujian kinerja jaringan (*Ping/Latency Test*), hingga pengujian fungsionalitas (*Blackbox* dan *Functional Testing*).

Arsitektur *peer-to-peer* (P2P) relatif jarang digunakan dalam pengembangan game *multiplayer* dibandingkan arsitektur *client-server* karena memiliki kompleksitas implementasi yang lebih tinggi. Pada arsitektur P2P, setiap node harus menangani komunikasi dua arah secara simultan, sehingga pengelolaan sinkronisasi state, konsistensi data, dan penanganan konflik menjadi lebih sulit. Selain itu, P2P menghadapi keterbatasan pada aspek *scalability*, *peer discovery*, dan manajemen trafik jaringan, terutama ketika jumlah pemain meningkat. Beberapa penelitian menunjukkan bahwa mekanisme *discovery* masih bersifat manual dan sistem *broadcast* data sederhana berpotensi menimbulkan *congestion* dan inkonsistensi data.

Pada lingkungan lokal yang terkontrol dengan jumlah pemain terbatas, beberapa kelemahan utama P2P seperti ketergantungan pada koneksi internet dan server pusat menjadi tidak relevan. Sebaliknya, P2P secara teoritis dapat menawarkan latensi yang lebih rendah karena komunikasi dilakukan langsung antar perangkat. Kajian literatur menunjukkan bahwa sebagian besar penelitian terdahulu berfokus pada arsitektur *client-server* berbasis *cloud* dan hanya mengevaluasi aspek fungsional atau pengalaman pengguna tanpa pengukuran performa jaringan yang komprehensif. Berdasarkan kajian terhadap literatur, penelitian ini didasarkan pada kebutuhan untuk membandingkan arsitektur *client-server* dan P2P secara objektif dan terukur dalam skenario game *multiplayer* yang sama dan lingkungan jaringan lokal yang identik. Perbandingan ini penting untuk mengidentifikasi *trade-off* performa masing-masing arsitektur, seperti latensi, stabilitas sinkronisasi, dan efisiensi komunikasi, yang belum banyak dibahas dalam penelitian sebelumnya.

2.5.2 Matriks Penelitian

Matriks penelitian dibuat untuk mengidentifikasi perbedaan penelitian saat ini dengan penelitian terdahulu, matriks penelitian disajikan pada tabel 2.

Tabel 2.2 Matriks Penelitian

No.	Peneliti	Arsitektur Jaringan		Framework/Teknologi				Platform		Metode Pengembangan				
		Client Server	P2P	PUN	UNet	Java Socket	Python Socket	Android	Desktop	GDLC	MDLC	Studi Literatur	Agile Extreme Programming	Prototyping
1	(Sarwodi dkk., 2020)	✓		✓				✓				✓		
2	(Ramadhan dkk., 2020)	✓			✓			✓				✓		
3	(Geubrina dkk., 2024)	✓		✓				✓			✓			
4	(Thuan & Raka, 2021)	✓		✓					✓					✓

No.	Peneliti	Arsitektur Jaringan		Framework/Teknologi				Platform		Metode Pengembangan				
		Client Server	P2P	PUN	UNet	Java Socket	Python Socket	Android	Desktop	GDLC	MDLC	Studi Literatur	Agile Extreme Programming	Prototyping
5	(Syauqi A'la & Hijjah Permata Rahmatul, 2020)	✓				✓			✓			✓		
6	(Wahyudi dkk., 2022)	✓		✓				✓					✓	
7	(Sulu dkk., 2024)		✓				✓		✓			✓		
8	(Pande dkk., 2023)	✓		✓				✓		✓				
9	(Lumombo & Jones, 2023)	✓		✓				✓						✓
10	(Adriansyah Ikhsanul Iqbal & Paramitha Nerisafitra, 2025)	✓		✓				✓		✓				

Berdasarkan tabel 2.2 teridentifikasi adanya gap penelitian yang signifikan yaitu belum terdapat perbandingan performa yang langsung dan empiris antara arsitektur *Client-Server* dan P2P dalam konteks game *multiplayer* yang sama, khususnya di lingkungan jaringan lokal (LAN) serta eksplorasi P2P untuk aplikasi game terutama yang berbasis Unity masih sangat terbatas.

2.5.3 Gap Penelitian

Berdasarkan analisis terhadap penelitian-penelitian terdahulu, teridentifikasi beberapa gap penelitian yang signifikan. Pertama, tidak ditemukan adanya penelitian yang melakukan perbandingan langsung antara arsitektur *client-server* dan *peer-to-peer* dalam konteks game *multiplayer* berbasis Unity. Kedua, evaluasi performa kuantitatif menggunakan metrik yang komprehensif masih sangat minim dalam literatur eksisting (Syauqi A'la & Hijjah Permata Rahmatul, 2020). Ketiga, arsitektur *peer-to-peer* sangat jarang dieksplorasi dalam domain game *multiplayer*, padahal arsitektur ini memiliki karakteristik yang berbeda dan potensial untuk kasus penggunaan tertentu (Sulu dkk., 2024). Keempat, tidak ada penelitian yang memberikan rekomendasi berbasis data empiris mengenai pemilihan arsitektur jaringan yang paling sesuai untuk karakteristik game tertentu, penelitian terdahulu lebih bersifat sebagai validasi fungsional serta evaluasi pengalaman pengguna (Thuan & Raka, 2021). Gap-gap ini menunjukkan perlunya penelitian yang secara sistematis membandingkan kedua arsitektur dengan metodologi yang rigorous dan metrik yang terukur.

Kontribusi dari penelitian-penelitian terdahulu terutama terletak pada keberhasilan implementasi berbagai jenis game *multiplayer* menggunakan

arsitektur *client-server* dan bukti konsep penggunaan P2P untuk aplikasi non-game (Adriansyah Ikhsanul Iqbal & Paramitha Nerisafitra, 2025; Sulu dkk., 2024). Penelitian-penelitian tersebut telah membuktikan bahwa berbagai *framework* dan teknologi *networking* dapat diintegrasikan dengan Unity Engine untuk menghasilkan game *multiplayer* yang fungsional. Namun, kontribusi yang lebih mendalam terkait dengan performa komparatif antar arsitektur masih belum tersedia. Penelitian ini berupaya memberikan kontribusi yang lebih substantif dengan mengimplementasikan kedua arsitektur *client-server* dan *peer-to-peer* pada game yang sama, kemudian mengukur performa kedua arsitektur tersebut menggunakan metrik yang komprehensif, dan melakukan analisis perbandingan yang mendalam. Hasil dari penelitian ini diharapkan dapat memberikan rekomendasi berbasis data untuk pemilihan arsitektur jaringan berdasarkan karakteristik game dan data hasil pengujian, sehingga dapat menjadi *baseline* untuk penelitian arsitektur jaringan game selanjutnya serta panduan praktis bagi pengembang game dalam menentukan arsitektur yang paling optimal untuk kebutuhan mereka.

Penelitian ini memiliki kontribusi potensial untuk mengisi kekosongan yang teridentifikasi dari penelitian-penelitian sebelumnya. Dengan menyediakan perbandingan terukur antara arsitektur *client-server* dan *peer-to-peer* di lingkungan jaringan lokal menggunakan parameter performa seperti latensi, *throughput*, *packet loss*, serta penggunaan CPU dan memori, penelitian ini dapat memberikan rekomendasi berbasis data mengenai pemilihan arsitektur jaringan yang paling tepat berdasarkan karakteristik spesifik suatu game.