

Analisa Penggunaan WebRTC dan Websocket pada Real Time Multiplayer Online Game Tradisional Ceki

Gede Humaswara Prathama¹, Ni Made Ary Esta Dewi Wirastuti², Yoga Divayana³

[Submission: 23-11-2018, Accepted: 31-03-2019]

Abstract—Ceki is one of the traditional card games that has been quite popular in Malaysia, Singapore and Indonesia. In Bali the game of ceki has become a tradition, where the game is played when there is community activity. Multiplayer online games are one way to introduce and develop traditional games to be more interesting to play. The problems that are often faced to develop multiplayer online games include the latency problems that are different from each player so that it can interfere with games from other players who have better latency. In this study, a real time multiplayer online game will be built to play traditional games. discussing how the rules of traditional games obtained from the rules of the Balinese game will be applied to the game engine from online multiplayer games, how to implement online game multiplayer architecture on client-server and peer to peer web-browsers. The focus of this research is on the performance of the use of *websocket* and *webRTC* on latency for data synchronization that supports real time multiplayer online games based on peer to peer connection. From the average latency needed for data synchronization between web socket compared to *webRTC*, it can be concluded that the use of *webRTC* has a very significant effect on latency. The number of players has no effect on latency both on *websocket* and *webRTC*

Intisari—Ceki adalah salah satu permainan kartu tradisional yang pernah cukup populer di Malaysia, Singapura dan Indonesia. Di Bali permainan ceki sudah menjadi tradisi, dimana permainan ceki dimainkan saat adanya kegiatan masyarakat. *Multiplayer online games* adalah salah satu cara untuk mengenalkan dan mengembangkan *games* tradisional menjadi lebih menarik untuk dimainkan. Permasalahan yang sering dihadapi untuk mengembangkan *multiplayer online game* diantaranya adalah permasalahan *latency* yang berbeda dari masing-masing pemain sehingga bisa mengganggu permainan dari pemain lain yang memiliki *latency* lebih baik. Pada penelitian ini akan dibangun *real time multiplayer online game* untuk memainkan permainan tradisional ceki, menganalisa tentang bagaimana aturan-aturan dari permainan tradisional ceki yang didapat dari aturan permainan masyarakat Bali akan diterapkan pada *game engine* dari *multiplayer online game*, bagaimana implementasi arsitektur *multiplayer online game* pada *web-browser* yang berbasis *client-server* dan *peer to peer*. Fokus dari penelitian ini adalah pada performansi dari penggunaan *websocket* dan *webRTC* terhadap *latency* pada sinkronisasi data yang mendukung *real time multiplayer online game* berbasis koneksi *peer to peer*. Dilihat dari rata-rata *latency* yang dibutuhkan untuk sinkronisasi data antara *websocket*

dibandingkan dengan *webRTC*, dapat disimpulkan bahwa penggunaan *webRTC* memberikan pengaruh yang sangat signifikan terhadap *latency*. Jumlah pemain tidak berpengaruh pada *latency* baik pada *websocket* maupun *webRTC*.

Kata Kunci - Ceki, Multiplayer Online Game, websocket, webRTC, latency, web protocol

I. PENDAHULUAN

Games atau permainan telah menjadi bagian hidup bagi masyarakat sejak dahulu, baik hanya sebagai hiburan hingga menjadi mata pencaharian. *Games* atau permainan yang ada saat ini juga banyak dipengaruhi oleh budaya (permainan tradisional) dan perkembangan teknologi. Saat ini permainan yang berbasis teknologi mendominasi di masyarakat dan permainan tradisional mulai ditinggalkan oleh masyarakat, dari permainan yang dimainkan sendiri (*Singleplayer*) hingga permainan yang bisa dimainkan lebih dari 1 orang (*Multiplayer*) dan permainan yang dimainkan secara *offline* hingga permainan yang dimainkan secara *online*. Tidak sedikit pengembang *games* saat ini yang mengangkat kembali permainan tradisional yang berbasis teknologi dan terkadang ditambahkan sedikit modifikasi agar *games* lebih menarik untuk dimainkan. *Games* tradisional umumnya adalah permainan yang dimainkan oleh lebih dari 1 orang, maka pengembang *games* diharuskan mengembangkan *games* yang bisa dimainkan lebih dari 1 orang, berdasarkan *rating* di layanan penjualan aplikasi, *game* yang bisa dimainkan dengan teman atau orang lain selain *bot* lebih diminati oleh masyarakat, hal ini disebabkan karena adanya interaksi sosial yang terjadi jika memainkannya dengan teman atau orang lain dibandingkan dengan *bot*, maka *multiplayer online games* adalah salah satu cara untuk mengembangkan *games* tradisional menjadi lebih menarik untuk dimainkan dengan lebih dari 1 pemain.

Multiplayer online games adalah permainan yang bisa dimainkan oleh lebih dari satu pemain secara *online* melalui jaringan internet dalam waktu yang sama. *Multiplayer online game* menjadi menarik karena adanya interaksi sosial yang terjadi dalam permainan yaitu berupa kerja sama, kompetisi hingga persaingan dalam meraih *reward* pada permainan dengan teman atau orang lain. Secara umum *Multiplayer online game* memiliki beberapa kategori di antaranya adalah *Massively Multiplayer Online game* (MMO), *Massively Multiplayer Online First-person shooter games* (MMOFPS), *Massively Multiplayer online Real-time strategy games* (MMORTS), *Massively Multiplayer Online Role-playing games* (MMORPG), *Multiplayer Online Battle arena* (MOBA). Salah satu jenis *multiplayer online games* yang cukup populer dimainkan adalah *card games*, permainan ini menjadi populer karena cara bermain yang sederhana dan tidak sedikit *card games* yang ada saat ini berasal dari

¹Mahasiswa, Jurusan Teknik Elektro dan Komputer Fakultas Teknik Universitas Udayana, Jln. Gatot Subroto II Denpasar 80231 INDONESIA (telp: 081805635084; fax: - ; e-mail: huma.elektro@gmail.com)

^{2,3}Dosen, Jurusan Teknik Elektro dan Komputer Fakultas Teknik Universitas Udayana, Jln. Jalan Kampus Bukit Jimbaran 80361 INDONESIA (telp: 0361-703315; fax: 0361-4321; e-mail: dewi.wirastuti@ee.unud.ac.id) yogadivayana@gmail.com)



permainan tradisional di mana permainan ini sudah ada sebelum adanya teknologi seperti saat ini.

Ceki adalah salah satu permainan kartu tradisional yang pernah cukup populer di Malaysia, Singapura dan Indonesia, namun saat ini permainan *ceki* sudah mulai jarang dimainkan di Malaysia dan Singapura. *Ceki* disebutkan berasal dari permainan di negara Tiongkok (Cina kuno) dan disebut juga menurunkan cara permainan tradisional *mahjong* yang memang sangat populer di Tiongkok hingga saat ini. *Ceki* dimainkan dengan aturan dan cara memainkan yang berbeda-beda di setiap daerah di Indonesia tergantung dari budaya dan tradisi yang ada di daerah tersebut. Di Bali permainan *ceki* sudah menjadi tradisi, di mana permainan *ceki* dimainkan saat adanya kegiatan masyarakat atau upacara adat, bahkan saat ini sudah dibuatkan kompetisi permainan *ceki* oleh pemerintah daerah Bali. Permainan *ceki* bisa dimainkan oleh 4 – 5 pemain namun idealnya dimainkan oleh 5 pemain agar peredaran kartu berputar dengan baik. 1 set kartu *Ceki* atau yang sering juga disebut dengan *Sampian* oleh masyarakat bali terdiri dari 60 kartu dan dimainkan dengan 2 set kartu.

Arsitektur *multiplayer online games* umumnya berbasis *client-server*, di mana *server* digunakan sebagai *data center*, *game lobby*, *game engine* dan *game room*. *Protocol* yang digunakan sebagai sarana komunikasi antara *client* dan *server* adalah *TCP* atau *UDP* dengan kelebihan dan kekurangan masing-masing disesuaikan dengan *games* yang akan dikembangkan, atau bisa juga kombinasi dari *TCP* dan *UDP*. Untuk mendapatkan kesan *real-time* pada *multiplayer online games*, pengembang umumnya melakukan optimalisasi paket menggunakan serialisasi data paket yang dikirim melalui *protocol* untuk mengurangi *latency* pada pergerakan pemain yang dilihat oleh pemain lain dalam permainan. Arsitektur *Multiplayer online games* juga bergantung pada *platform* yang digunakan untuk mengembangkan *games* dan tidak jarang *games* yang dikembangkan bisa dimainkan melalui banyak *platform* atau yang sering disebut dengan *cross-platform*. *Web-browser* adalah salah satu *platform* atau *virtual machine* yang digunakan untuk memainkan *online games*.

Permasalahan yang sering dihadapi untuk mengembangkan *multiplayer online games* di antaranya adalah permasalahan *latency* yang berbeda dari masing-masing pemain tergantung dari koneksi internet yang digunakan masing-masing pemain sehingga bisa mengganggu permainan dari pemain lain yang memiliki *latency* lebih baik. Permasalahan lainnya adalah protokol apa yang sebaiknya digunakan untuk mendapatkan kesan *real time* dari permainan, bagaimana mengoptimalkan paket data dari *client* ke *server* begitu juga sebaliknya, bagaimana jika salah satu atau beberapa pemain kehilangan koneksi sementara dan yang sering menjadi permasalahan adalah penggunaan *bandwidth* untuk komunikasi data yang tentu saja diharuskan penggunaan *bandwidth* sekecil mungkin.

Pada penelitian ini akan dibangun *real time multiplayer online games* untuk memainkan permainan tradisional *ceki*, menganalisa tentang bagaimana aturan-aturan dari permainan tradisional *ceki* yang didapat dari aturan permainan masyarakat Bali akan diterapkan pada *game engine* dari *multiplayer online game* yang akan dibangun. Bagaimana implementasi arsitektur *multiplayer online games* pada *web-browser* yang berbasis *client-server* dan *peer to peer*. Fokus dari penelitian ini adalah

pada uji performansi dari penggunaan *websocket* dan *webRTC* terhadap *latency* dan penggunaan *bandwidth* untuk sinkronisasi data yang mendukung *real time multiplayer online games* berbasis *client-server* dan *peer to peer*.

II. TINJAUAN PUSTAKA

A. *Ceki*

Ceki merupakan sebuah permainan kartu yang mirip dengan permainan di cina. Setiap satu *pack* terdiri dari 60 kartu (30 jenis bentuk). Jumlah kartu yang dibutuhkan dalam permainan *ceki* adalah 2 *pack* kartu dengan masing-masing jumlah bentuk kartu terdiri dari 4 buah pasangan dengan jumlah total 120. Pada permainan *ceki* jumlah kartu yang dipegang pemain berjumlah 11 lembar. Permainan *ceki* memiliki beberapa istilah atau aturan dalam memainkannya. Adapun istilah-istilah atau aturan permainan *ceki* adalah sebagai berikut :

- *Mukak* merupakan istilah untuk mengambil kartu hasil dari kocokan. Daerah lain di Bali sering menggunakan istilah ngupak.
- *Ngejuk* merupakan istilah untuk mengambil kartu buangan pemain sebelumnya.
- *Ngutang* merupakan istilah saat pemain membuang sebuah kartu dari tangan.
- *Medua* atau *lawang* merupakan istilah untuk kartu yang dipegang pemain memiliki gambar dan tepi yang sama sebanyak dua lembar.
- *Soca* merupakan istilah untuk tiga lembar kartu di tangan pemain yang memiliki bentuk tepi dan gambar yang sama.
- *Serigat* merupakan tiga lembar kartu di tangan yang memiliki bentuk tepi yang sama. *Serigat* terdiri dari dua jenis, yaitu *serigat dua* dan *serigat telu*.
- *Mecari* merupakan kondisi pemain menunggu kartu terakhir saat pemain sudah mengumpulkan satu *soca*, dua *serigat*, dan 1 *medua* atau *lawang*.
- *Nyaga* merupakan kondisi pemain menunggu kartu terakhir setelah mengumpulkan dua *soca*, satu *serigat* dan dua *medua* atau *lawang*. Kondisi *nyaga* memiliki aturan setiap pemain yang melakukan “*mukak*”, kartu yang dibuka harus diperlihatkan ke pemain dalam kondisi *nyaga*.
- *Mbet* merupakan istilah saat kartu terakhir yang dicari atau dijaga pada proses *mecari* atau *nyaga* sudah keluar atau terbang, sehingga pemain tidak bisa menang.
- *Minde* merupakan kondisi saat pemain pertama membuang kartu yang sama sebanyak dua kali dan kartu yang dibuang tersebut diambil oleh pemain berikutnya.

B. Permainan Online (Online Games)

Permainan *online* atau dialih bahasakan menjadi permainan daring (*Online Games*) adalah jenis permainan komputer yang memanfaatkan jaringan internet, sebagai medianya. Permainan *online* disediakan sebagai tambahan layanan dari perusahaan penyedia jasa *online*, atau dapat diakses langsung melalui sistem yang disediakan dari perusahaan yang menyediakan permainan tersebut [1]. Terdapat banyak pola pada *online*

games yang bisa digunakan pengembang *games*, untuk permainan kartu secara *online* ada 2 pola yang bisa digunakan yaitu *Shadowplayer pattern* dan *Serverfocused pattern*, dimana setiap kali beban *server* harus rendah dan komputasi pada klien cukup kuat dan para pengguna *games* terus bertambah, pola desain *Shadowplayer* harus diterapkan untuk implementasi namun jika beban pada klien harus rendah, kemudian pola desain *Serverfocused* yang harus diterapkan [2].

C. Client-server

Client-server adalah model arsitektur perangkat lunak yang terdiri dari dua bagian, sistem klien dan sistem *server*, baik berkomunikasi melalui jaringan komputer atau pada komputer yang sama. Aplikasi *client-server* adalah sistem terdistribusi yang terdiri dari perangkat lunak klien dan *server*. Aplikasi *server* klien menyediakan cara yang lebih baik untuk berbagi beban kerja. Pada awal ditemukan arsitektur *Client-server* proses klien selalu memulai koneksi ke *server*, sementara proses *server* selalu menunggu permintaan dari klien mana pun. Dengan kecepatan internet saat ini aplikasi dengan arsitektur *Client-server* mulai banyak dikembangkan dengan jumlah klien yang sangat banyak sehingga beban *server* menjadi sangat besar, ini disebabkan oleh semakin banyak klien yang terhubung pada *server*, maka rata-rata *bytes*/detik yang dikirimkan akan semakin besar, begitu juga dengan rata-rata paket/detik dan semakin besar ukuran *file* yang ditransfer maka akan semakin besar pula *throughput*-nya sampai mencapai kondisi saturasi sedangkan *latency* akan semakin cepat jika ukuran paket semakin kecil [3][4].

D. WebSocket

WebSocket adalah protokol komunikasi komputer, menyediakan saluran komunikasi *full duplex* melalui koneksi *TCP* [5][6]. Protokol *websocket* distandarisasi oleh IETF sebagai RFC 6455 pada tahun 2011. *WebSocket* adalah protokol *TCP* yang berbeda dari *HTTP*. Kedua protokol terletak di layer 7 dalam model OSI, dengan demikian *websocket* bergantung pada *TCP* pada layer 4. Protokol *websocket* memungkinkan interaksi antara klien web (seperti *browser*) dan *web server* dengan *overhead* yang lebih rendah, memfasilitasi transfer data *real time* dari dan ke *server*. Hal ini dimungkinkan dengan menyediakan cara standar bagi *server* untuk mengirim konten ke klien tanpa diminta terlebih dahulu oleh klien, dan memungkinkan pesan diteruskan bolak-balik sambil menjaga koneksi tetap terbuka. Protokol *websocket* saat ini didukung di sebagian besar *browser* termasuk Google Chrome, Microsoft Edge, Internet Explorer, Firefox, Safari dan Opera [6][7]. *WebSocket* juga membutuhkan aplikasi web di *server* untuk mendukungnya. Untuk membuat sambungan *websocket*, klien harus mengirim permintaan *handshake websocket*, lalu *server* akan mengembalikan respons *handshake websocket*, seperti yang ditunjukkan pada contoh di bawah ini [5]:

```
GET /chat HTTP/1.1
```

```
Host: server.example.com
```

```
Upgrade: websocket
```

```
Connection: Upgrade
```

```
Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==
```

Gede Humaswara Prathama: Analisa Penggunaan WebRTC dan...

```
Sec-WebSocket-Protocol: chat, superchat
```

```
Sec-WebSocket-Version: 13
```

```
Origin: http://example.com
```

Handshake pada *websocket* menyerupai *HTTP* dalam berkomunikasi dengan *server* untuk menangani koneksi *HTTP* serta koneksi *WebSocket* pada *port* yang sama. Setelah koneksi terbentuk, komunikasi beralih ke protokol biner dua arah yang tidak sesuai dengan protokol *HTTP* [8]. Selain *upgrade header*, klien mengirimkan *header sec-websocket-key* yang berisi *byte* acak *base64-encoded*, dan *server* membalas dengan *hash key* di *header sec-websocket-accept*. Ini dimaksudkan untuk mencegah *proxy caching* mengirim ulang percakapan *websocket* sebelumnya dan tidak memberikan otentikasi, privasi atau integritas apa pun. Fungsi *hashing* menambahkan *string* tetap 258EAF5E914-47DA-95CA-C5AB0DC85B11 (sebuah *GUID*) ke nilai dari *header sec-websocket-key* (yang tidak didekodekan dari *base64*), menerapkan fungsi *hashing SHA-1*, dan mengkodekan hasilnya menggunakan *base64*. Setelah koneksi terbentuk, klien dan *server* dapat mengirim data *websocket* atau teks bolak-balik dalam mode *full-duplex*.

E. WebRTC

WebRTC (*Web Real Time Communication*) adalah proyek sumber terbuka gratis yang menyediakan layanan bagi *browser web* dan aplikasi seluler dengan *real time communication* (*RTC*) melalui antarmuka *Application Programming Interface* (*API*). Hal ini memungkinkan komunikasi audio dan video bekerja di dalam halaman web yang memungkinkan komunikasi *peer-to-peer*, menghilangkan kebutuhan untuk *install plugin* atau mengunduh aplikasi lain. *WebRTC* juga dapat mengurangi konsumsi daya dan penggunaan *bandwidth* dengan demikian *webRTC* dapat meningkatkan efisiensi sistem [9]. Misinya adalah untuk mengaktifkan aplikasi *RTC* berkualitas tinggi yang dikembangkan untuk peramban, *platform* seluler, dan perangkat *IoT*, dan memungkinkan mereka semua berkomunikasi melalui seperangkat protokol umum. Pembentukan dengan cepat koneksi *peer RTC* adalah persyaratan penting untuk aplikasi *web* masa depan. Juga, kinerja transfer media dan data yang baik khususnya pada perangkat seluler, sangat penting untuk mendapatkan pengalaman pengguna yang baik [10]. Referensi implementasi dirilis sebagai perangkat lunak bebas di bawah ketentuan lisensi *BSD*. Komponen utama dari *WebRTC* termasuk beberapa *API JavaScript* adalah:

- *getUserMedia*, yaitu komponen yang mengakuisisi media audio dan video (misal, dengan mengakses kamera dan mikrofon perangkat) [11].
- *RTCPeerConnection* memungkinkan komunikasi audio dan video antar sesama. Ini melakukan pemrosesan sinyal, penanganan *codec*, komunikasi *peer-to-peer*, keamanan, dan manajemen *bandwidth* [11].
- *RTCDataChannel* memungkinkan komunikasi dua arah dari data acak antar sesama yang menggunakan *API* yang sama dengan *websocket* dan memiliki *latency* yang sangat rendah [11].

p-ISSN:1693 – 2951; e-ISSN: 2503-2372



9 772503 237078

API WebRTC juga mencakup fungsi statistik:

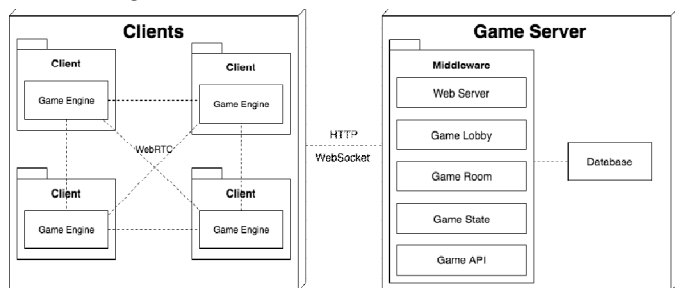
- *getStats* memungkinkan aplikasi web untuk mengambil sekumpulan statistik tentang sesi *WebRTC*. Data statistik ini sedang dijelaskan dalam dokumen W3C yang terpisah [11].

API *PeerConnection*, kanal data dan *media capture browser* dijelaskan secara rinci di W3C. W3C mengembangkan *ORTC (Object Real-Time Communications)* untuk *WebRTC*. Ini biasanya disebut sebagai *WebRTC 1.1*.

III. METODE PENELITIAN

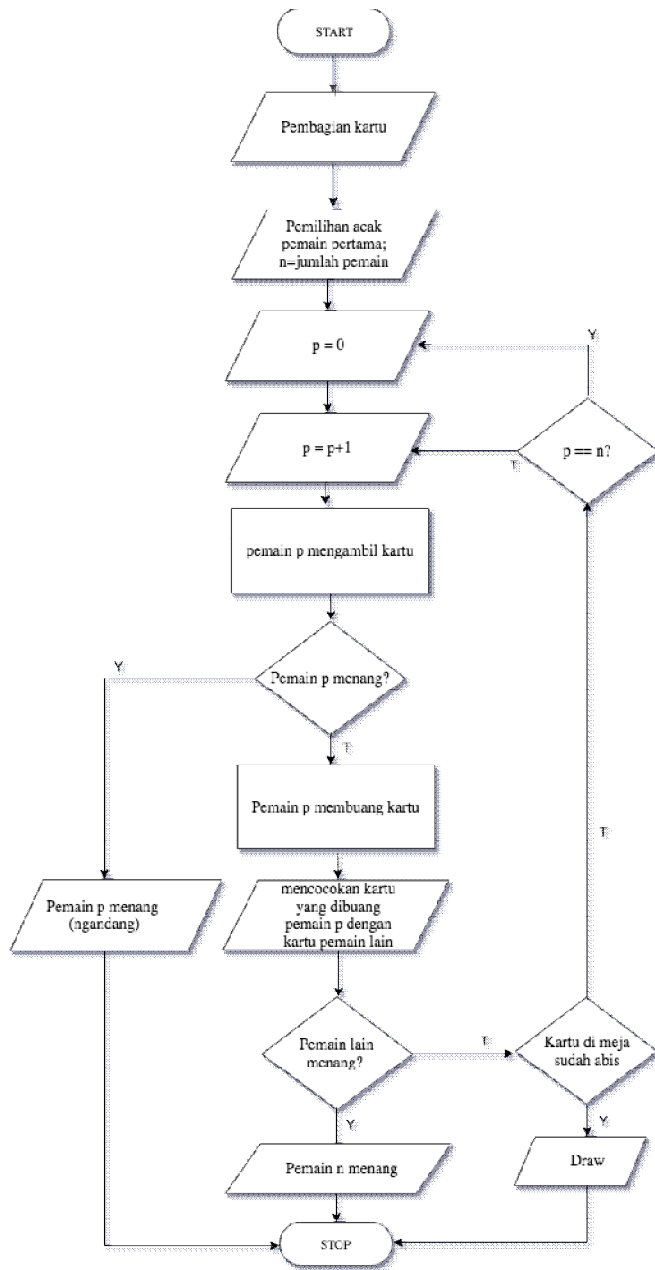
Metode penelitian menguraikan beberapa hal antara lain, rancangan penelitian yang terdiri dari gambaran umum, *Game server*, *Gameplay Engine*, alur komunikasi data, serialisasi data, rancangan *database* dan rancangan *user interface*.

A. Rancangan Penelitian.



Gambar 1: Gambaran umum sistem

Pada gambaran umum penelitian ini dijelaskan bahwa pola yang digunakan pada sistem ini adalah pola *shadow player* dimana *game engine* terletak pada *client*. Sistem dimulai dari *game server* dimana *game server* bertugas hanya untuk mendaftarkan pemain baru, mempersiapkan permainan pada *game room*, mendistribusikan kartu pada *game room* dan mencatat histori permainan pada saat permainan berlangsung. Ketika permainan berlangsung proses diserahkan kepada *gameplay engine* yang ada pada masing-masing *client*, *gameplay engine* disini bertugas untuk mengatur permainan mulai dari mengatur giliran pemain, komunikasi data permainan antar *client* melalui *gameplay engine*, komunikasi data ke *server* untuk melaporkan histori permainan dan menentukan hasil permainan jika ada yang menang atau terjadi *draw* pada permainan sesuai dengan *rule* yang ada pada *gameplay engine*. Pada komunikasi data antar *client* yang dilakukan oleh *gameplay engine* dilakukan tanpa *server (peer to peer)*. Alur permainan ceki pada penelitian ini disesuaikan dengan permainan yang berlaku pada daerah kabupaten Tabanan dan kabupaten Negara dikarenakan pada beberapa daerah aturan permainan terdapat sedikit perbedaan, berikut adalah *flowchart* dari *gameplay* :



Gambar 2: Diagram alur Permainan Kartu Ceki

Pada penelitian ini menggunakan 2 protokol pada komunikasi data baik pada *client to server* dan *peer to peer*. Pada komunikasi *client to server* digunakan *websocket* dan *HTTP* sebagai protokol. *WebSocket* digunakan untuk komunikasi *game state* pada *game room* secara *real time* tanpa perlu melakukan *long polling* dan *HTTP* digunakan pada *game lobby* dimana *client* melakukan pendaftaran akun, *login* dan masuk ke dalam *game room*. Pada komunikasi *peer to peer* digunakan *webRTC* sebagai protokol untuk komunikasi *peer to peer* antar *client*, diharapkan dengan menggunakan *webRTC* latency yang didapatkan oleh *client* sangat kecil. Proses serialisasi data juga dilakukan untuk meminimalkan data paket yang dikirim melalui protokol baik pada *client-server* dan *peer to peer*. Pada penelitian ini digunakan jenis serialisasi data yaitu *JSON* yang dikonversi menjadi *arraybuffer* ketika dikirim. Rancangan *database* dibuat dengan tujuan untuk mengidentifikasi kebutuhan *file-file database* (basis data) yang dibutuhkan

dalam pembuatan program. Rancangan struktur tabel pada penelitian ini dapat dilihat seperti berikut:

TABEL I
STRUKTUR TABEL PEMAIN

Nama field	Type	Key	Keterangan
id_pemain	int (9)	Pri	Id kode pemain
Username	varchar (20)		Nama unik pemain
email	varchar (100)		Email pemain
Password	varchar (100)		Password pemain
Menang	Int (9)		Jumlah kemenangan pemain
draw	Int (9)		Jumlah draw pemain
kalah	Int (9)		Jumlah kekalahan pemain

TABEL II
STRUKTUR TABEL GAME_ROOM

Nama field	Type	Key	Keterangan
id_room	int (9)	Pri	uid game room
mulai	datetime ()		Waktu saat permainan dimulai
selesai	datetime ()		Waktu saat permainan selesai
game_state	longtext		Histori permainan berupa JSON yang di convert menjadi string

TABEL III
STRUKTUR TABEL GAME_ROOM_PLAYER

Nama field	Type	Key	Keterangan
id_room	int (9)	For	Foreign key dari table game_room
Id_pemain	int (9)	For	Foreign key dari tabel pemain

status	smallint(3)	Status hasil permainan terhadap pemain (0 untuk dalam permainan, 1 untuk menang, 2 untuk draw dan 3 untuk kalah)
--------	-------------	--

B. Proses Pengujian

Proses pengujian sistem dilakukan untuk melihat hasil yang dihasilkan sistem. Proses pengujian yang dilakukan pada penelitian ini dilakukan pada dua hal antara lain.

1) *Latency*: yaitu mengukur dan membandingkan *latency* yang didapat dengan skenario sebagai berikut:

- membandingkan 2 pemain menggunakan *websocket* dengan *websocket* dan *webRTC*.
- membandingkan 3 pemain menggunakan *websocket* dengan *websocket* dan *webRTC*.
- membandingkan 4 pemain menggunakan *websocket* dengan *websocket* dan *webRTC*.
- membandingkan 5 pemain menggunakan *websocket* dengan *websocket* dan *webRTC*.

C. Jadwal Penelitian

Jadwal pelaksanaan penelitian dilakukan dari bulan Maret 2018 sampai dengan Agustus 2018 yang diawali dari proses studi literatur sampai dengan penyusunan laporan.

IV. HASIL DAN PEMBAHASAN

Multiplayer online games ceki berbasis *client-server* sehingga terdapat 2 bagian pada arsitektur yaitu arsitektur pada *server* dan arsitektur pada *client*. Untuk dapat menguji performansi dari *websocket* dengan *websocket + webRTC* maka pada penelitian ini dibangun 2 arsitektur yang menjalankan protokol *websocket* dengan *websocket + webRTC*.

A. Arsitektur multiplayer online games pada permainan tradisional ceki

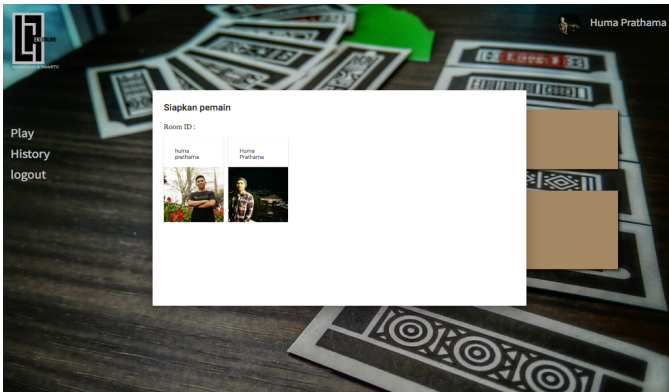
Server pada *multiplayer online games ceki* secara umum digunakan untuk mempertemukan antara pemain, mencatat *game state* dan mencatat histori permainan dari masing-masing pemain. Terdapat 4 komponen utama yang ada pada *server* yaitu *Database*, *middleware*, *Game room* dan *game state*. *Database* pada *multiplayer online games ceki* memiliki hanya 3 tabel seperti pada perancangan yaitu tabel pemain, tabel room dan tabel room pemain. *Middleware* pada arsitektur ini adalah untuk mengautentifikasi pengguna yang memanggil *server* dan menentukan data yang boleh diakses oleh pengguna. *Game room* pada arsitektur ini adalah untuk mempertemukan para pemain pada permainan yang sama dan bertugas untuk menginisialisasikan permainan dengan mengacak kartu serta membagikannya kepada pemain. *Game room* hanya menggunakan protokol *websocket* pada 2 arsitektur yang dibangun pada penelitian ini. Pada arsitektur yang hanya menggunakan *websocket*, *game room* bertugas juga untuk sinkronisasi data setiap langkah dari pemain. *Game*



state pada arsitektur ini adalah untuk mencatat setiap langkah dari pemain yang ada pada 1 *room*, jadi ketika ada pemain yang berkoneksi ulang ke dalam *room*, *game state* yang bertugas memberikan *state* terakhir dari permainan. *Game state* hanya menerima koneksi melalui *protocol http* pada 2 arsitektur yang dibangun pada penelitian ini. *Client* pada arsitektur ini merupakan aplikasi yang akan diakses oleh pemain melalui *web browser*.

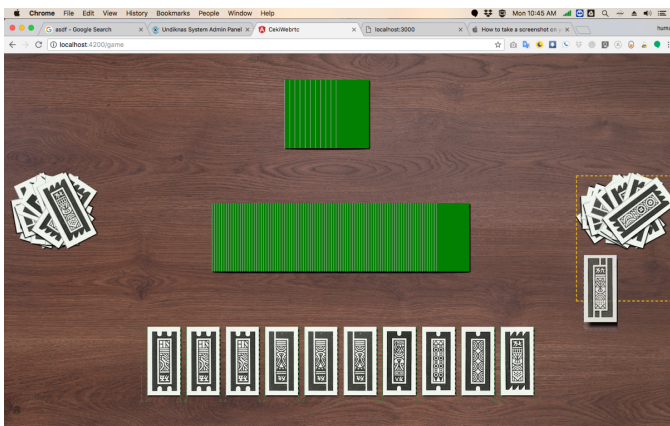
B. Gameplay permainan tradisional ceki pada multiplayer online games

Permainan dimulai dengan menginisialisasi permainan dengan memilih menu *create room* atau *join room*, lalu bergiliran masing-masing pemain boleh mengambil kartu dan membuang kartu, hingga mendapatkan hasil dari permainan. Terdapat 2 kemungkinan hasil dari permainan yaitu salah satu pemain menang atau permainan berakhir imbang.



Gambar 3: Tampilan *join room*

Saat giliran dari 1 pemain, pemain tersebut hanya boleh membuang 1 kartu dan mengambil 1 kartu dari kartu tengah atau dari kartu buangan pemain sebelumnya, kartu yang boleh diambil dari kartu buangan pemain sebelumnya hanya kartu buangan terakhir dari pemain sebelumnya, begitu seterusnya hingga menemukan pemenang atau *draw*. Dalam kenyataannya lama pemain dalam memilih kartu yang diambil dan membuang kartu tidak ditentukan, namun pada *gameplay* dari penelitian ini ditambahkan waktu gilir, fitur itu ditambahkan untuk mengantisipasi pemain yang tidak melanjutkan permainan atau meninggalkan permainan terlalu lama, sehingga ketika waktu habis kartu yang diambil dan dibuang akan otomatis dipilih oleh *game engine*.



Gambar 4: Tampilan pemain membuang kartu

C. Aturan Permainan Ceki Online

Aturan permainan ceki online secara umum masih sama dengan aturan permainan ceki tradisional yang ada namun terdapat aturan baru yang disesuaikan dengan kondisi dimana pemain bisa melakukan kecurangan atau tidak *fair play*. Beberapa kondisi yang dimaksud adalah saat salah satu pemain tidak melakukan gerakan dalam waktu yang lama atau bahkan meninggalkan permainan saat permainan masih berlangsung sehingga merugikan pemain lain, dengan kondisi seperti ini pada permainan ceki online ini ditambahkan fitur *timer* untuk membatasi waktu yang dihabiskan oleh 1 pemain pada setiap gerakannya. Lama waktu yang dipilih adalah 1 menit, pemilihan waktu 1 menit yang boleh digunakan pemain untuk melakukan gerakan berdasarkan pada permainan lain seperti permainan MOBA mobile legend yang ketika pemain tidak aktif dalam waktu 2 menit maka pemain akan digerakan oleh *game engine*, permainan yang lain seperti domino online yang merupakan permainan kartu juga menggunakan *timer*, rata-rata dari beberapa permainan domino pemain diberikan waktu antara 15-20 detik untuk melakukan gerakan, jika lebih dari itu maka kartu akan digerakan oleh *game engine*, permainan ceki pada penelitian menggunakan 1 menit adalah dari pertimbangan jumlah kartu yang dimainkan lebih banyak dari kartu domino, model permainan juga lebih kompleks dari domino namun tidak lebih kompleks dari game moba. Terdapat juga kondisi dimana pemain kehilangan koneksi atau koneksi sangat lambat, kondisi seperti ini sangat sulit dihindari dan solusi yang bisa digunakan seringkali merugikan semua pemain dan juga pengembang *game*, kesulitannya dikarenakan hal seperti ini bergantung kepada koneksi internet yang digunakan pemain, *hardware* yang digunakan pemain dan hal lainnya yang bisa mengganggu permainan diluar dari arsitektur permainan itu sendiri, maka hal yang bisa dilakukan untuk menjaga permainan tetap berlangsung adalah memanfaatkan *timer* untuk *game engine* menggerakkan pemain tertentu jika melewati *timer* yang telah ditentukan pada permainan yaitu 1 menit.

D. Hasil uji performansi websocket dengan websocket dan webRTC

Berikut adalah hasil uji performansi yang didapatkan dengan membanding *RTT* setiap paket dari arsitektur yang menggunakan *websocket* dengan *websocket* dan *webRTC*. pengujian dilakukan dalam 4 tahap, yaitu pengujian terhadap 2 pemain, 3 pemain, 4 pemain dan 5 pemain.

TABEL IV
RANGKUMAN HASIL UJI PERFORMANSI

No.	Jumlah pemain	latency			
		websocket		websocket + webRTC	
		range (ms)	average (ms)	range (ms)	average (ms)
1.	2 pemain	36-98	41	1-3	2
2.	3 pemain	36-52	40	1-11	3
3.	4 pemain	37-138	56	1-23	7
4.	5 pemain	37-51	40	1-13	7

V. KESIMPULAN

Simpulan yang dapat diambil dari hasil pengujian adalah Arsitektur permainan pada penggunaan *websocket* dengan *websocket* + *webRTC* menjadi berbeda. Perbedaan ada pada sinkronisasi data, pada *websocket* sinkronisasi data dilakukan oleh server sedangkan *webRTC* sinkronisasi dilakukan oleh *client* yang melakukan gerakan dengan mengirim ke *peer* yang lain dan ke *server*. *Gameplay* dari permainan *ceki* pada *online multiplayer games* dengan kenyataan tidak jauh berbeda, hanya ditambahkan aturan-aturan baru untuk menjaga kelancaran permainan, seperti waktu yang diberikan kepada setiap pemain ketika mendapat giliran. Pada rangkuman hasil uji performansi dapat dilihat rata-rata *latency* yang dibutuhkan untuk sinkronisasi data antara *websocket* dibandingkan dengan *websocket+webRTC* sangat signifikan. Pada *webRTC latency* terendah adalah 1 ms yang bisa disebut dengan *realtime*, pada *websocket latency* terendah adalah 36 ms dimana 36 ms adalah waktu yang dibutuhkan untuk melakukan ping ke *server*. dari hasil rangkuman juga dapat dilihat bahwa jumlah pemain tidak berpengaruh pada *latency* baik pada *websocket* maupun *webRTC*. *Latency* hanya dipengaruhi oleh kecepatan internet yang digunakan. Penggunaan *webRTC* masih terlalu dini untuk diterapkan ke tahap produksi untuk *games*, hal ini dikarenakan belum semua platform mendukung dan menggunakan standar *webRTC*, dimana pada penelitian ini ada beberapa *browser* yang tidak mendukung penggunaan *webRTC* dan beberapa *browser* yang tidak menggunakan standar W3C untuk *webRTC*.

REFERENSI

- [1] A. Rollings and E. Adams, *Andrew Rollings and Ernest Adams on Game Design*, New Riders, 2003.
- [2] D.Varun Ranganathan, *Design Patterns for Multiplayer Card Games. Computer, Communication and Signal Processing (ICCCSP)*. Chennai: IEEE, 2017.
- [3] H. C. Kusuma, *Rancang Bangun Aplikasi Massive Multiplayer Online Role Playing Game Berbasis Client-Server dengan Optimalisasi Paket*. Electronic computers. Computer science, Surabaya, jul. 2011.
- [4] M. Somantri, A. Darmariyadi, *Penerapan Pengolahan Paralel Model Cluster Sebagai Web Server*. Majalah Ilmiah Teknologi Elektro, [S.l.], v. 6, n. 1, jan. 2007. ISSN 2503-2372. Available : <https://ojs.unud.ac.id/index.php/JTE/article/view/233>.
- [5] P. L Greco, (2017). *HTML5 WebSocket: A Quantum Leap in Scalability for the Web*. Retrieved from [websocket.org: http://www.websocket.org/quantum.html](http://www.websocket.org/quantum.html).
- [6] Mozilla Developer Network. (2015). Websockets. Retrieved from MDN web docs: <https://developer.mozilla.org/en-US/docs/Glossary/WebSockets>.
- [7] Yuzo Fujishima, F. U. (2009, Desember 9). *Web Sockets Now Available In Google Chrome*. Retrieved from [blog.chromium.org: https://blog.chromium.org/2009/12/web-sockets-now-available-in-google.html](https://blog.chromium.org/2009/12/web-sockets-now-available-in-google.html).
- [8] I. Fette, A. Melnikov, *Relationship to TCP and HTTP*, RFC 6455 The WebSocket Protocol. IETF. sec. 1.7, Dec. 2011.
- [9] K. K. Dev, *WebRTC Implementation Analysis and Impact of Bundle Feature*, International Conference on Communication Systems and Network Technologies. 5. Gwalior, India: IEEE, 2015.
- [10] L. A. Sajjad Taheri, *WebRTCBench: A Benchmark for Performance Assessment of WebRTC Implementations*, Embedded Systems For Real-time Multimedia (ESTIMedia). 13, Amsterdam: IEEE, 2015.
- [11] W3C, *WebRTC 1.0: Real-time Communication Between Browsers*. Available: <https://www.w3.org/TR/2008/WD-html5-20080610/comms.html#tcp-connections>, 2018.



[Halaman ini Sengaja di Kosongkan]

