

SKRIPSI

**PEMANFAATAN SMARTPHONE SEBAGAI PENGENDALI
PERMAINAN BERBASIS WEB**



Priambodo Pangestu

NPM: 2013730055

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2018**

UNDERGRADUATE THESIS

**UTILIZATION OF SMARTPHONE AS WEB-BASED GAME
CONTROLLERS**



Priambodo Pangestu

NPM: 2013730055

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2018**

LEMBAR PENGESAHAN



PEMANFAATAN SMARTPHONE SEBAGAI PENGENDALI PERMAINAN BERBASIS WEB

Priambodo Pangestu

NPM: 2013730055

Bandung, 19 Desember 2018

Menyetujui,

Pembimbing

Dr. Veronica Sri Moertini

Ketua Tim Penguji

Kristopher David Harjono, M.T.

Anggota Tim Penguji

Dott. Thomas Anung Basuki

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng



PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PEMANFAATAN SMARTPHONE SEBAGAI PENGENDALI PERMAINAN BERBASIS WEB

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 19 Desember 2018



Priambodo Pangestu
NPM: 2013730055

ABSTRAK

Socket.io merupakan sebuah pustaka yang menyediakan fitur untuk melakukan komunikasi secara *real-time* dan dua arah antara *client* dan *server*. Dengan menggunakan Socket.io, *client* dapat mengirimkan pesan kepada *server* dan menerima respon tanpa harus melakukan *polling*, yang berarti proses pengecekan secara berulang terhadap *server* untuk mengetahui apakah *server* masih tersambung atau tidak. Fitur-fitur yang dimiliki oleh Socket.io dapat dimanfaatkan untuk mengembangkan aplikasi web yang membutuhkan komunikasi *real-time*. Salah satu pemanfaatan pustaka Socket.io adalah permainan berbasis web.

Permainan berbasis web yang akan dibangun dinamakan Finger For Life. Permainan ini memanfaatkan teknologi *smartphone* dan *PC*, di mana *smartphone* akan berperan sebagai pengendali di dalam permainan, dan *PC* akan berperan sebagai *console* yang akan menyediakan permainan. Untuk memainkan permainan, *smartphone* harus terkoneksi ke *PC* melalui *browser*. Oleh karena itu, fitur yang dimiliki oleh Socket.io digunakan untuk melakukan koneksi antara *smartphone* dan *PC*.

Dengan penggunaan Socket.io di dalam pengembangan aplikasi Finger For Life, diharapkan ukuran *latency* yang dihasilkan pada saat memainkan permainan akan sangat kecil. *Latency* merupakan jarak waktu yang dihasilkan pada saat suatu konten atau data dikirimkan dari *client* menuju *server*, maupun sebaliknya. Ukuran *latency* yang dihasilkan akan sangat berpengaruh pada saat tombol yang ada di *smartphone* ditekan, dengan respon yang diberikan oleh *PC* berdasarkan aksi tersebut. Semakin kecil ukuran *latency* yang dihasilkan maka akan semakin cepat respon yang diberikan.

Kata-kata kunci: Socket.io, pemanfaatan, *smartphone*, *PC*, pengendali, permainan, web, *browser*, *latency*

ABSTRACT

Socket.io is a library that enables real-time, bidirectional communication between the client and the server. With this API, client can send messages to the server and receive responses without having to poll the server, which means check continuously to the server to see whether the server still connected or not. With these features, Socket.io can be used to build a real-time communication web application such as games.

The web-based games which utilize the features of Socket.io is Finger For Life. This game use smartphone as the controller and PC as the console. To be able to play, users need to connect the smartphone to the PC through web browser. Therefore, Socket.io is used to connect a smartphone to the PC.

The use of Socket.io in the developing Finger For Life web application is expected to decrease the sum of latency of playing the web-based game. Latency is a time interval when the content or the data is being sent from client to the server and back. The sum of latency can affect how fast the response from the PC when the users click the button on the smartphone. The lower the latency, the faster the response can be sent.

Keywords: Socket.io, utilization, smartphone, PC, controller, game, web, browser, latency

*Dipersembahkan kepada Teknik Informatika UNPAR, keluarga
tercinta, teman-teman, dan diri sendiri*

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas seluruh berkat yang diberikan kepada penulis sehingga dapat menyelesaikan skripsi dengan judul **Pemanfaatan Smartphone sebagai Pengendali Permainan Berbasis Web** dengan baik dan tepat waktu. Penulis juga berterima kasih kepada pihak-pihak yang telah memberikan dukungan dan bantuan kepada penulis dalam menyelesaikan skripsi ini. Penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua penulis, Bapak Permadi dan Ibu Heni Herliana Nurhayati yang selalu memberikan dukungan selama pengerjaan skripsi ini.
2. Adik penulis, Mulyo Raharjo Pambudi yang selalu menghibur dan menemani penulis selama pengerjaan skripsi.
3. Bapak Pascal Alfadian sebagai dosen pembimbing yang telah membimbing penulis hingga dapat menyelesaikan skripsi ini.
4. Renaldi Nugroho, Gabriel Radewa, dan Reza Zacky yang telah membantu penulis dalam pengujian dan menemukan *bug* pada aplikasi Finger For Life.
5. Teman-Teman Kosan Ilham yang telah menghibur penulis dan membantu dalam pengujian aplikasi Finger For Life.
6. Teman-teman Teknik Informatika UNPAR yang telah membantu dalam pengerjaan tugas akhir ini.
7. Teman-teman Unit Kegiatan Mahasiswa POTRET UNPAR yang membantu dalam pengujian pada aplikasi Finger For Life.
8. Pihak-pihak lain yang telah membantu penulisan skripsi ini, yang terus memberikan doa dan semangat kepada penulis.

Akhir kata, penulis berharap agar skripsi ini dapat bermanfaat bagi pembaca yang hendak melakukan penelitian dan pengembangan yang terkait dengan skripsi ini.

Bandung, Desember 2018

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Metodologi	3
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 Node.js	5
2.1.1 HTTP	6
2.1.2 Path	6
2.1.3 Module	7
2.2 Express.js	7
2.2.1 express()	8
2.2.2 Application	9
2.2.3 Response	9
2.2.4 Router	10
2.3 Socket.io	10
2.3.1 Server API	11
2.3.2 Client API	13
2.4 Canvas API	14
2.4.1 Animation	14
2.4.2 canvasRenderingContext2D	16
2.5 jQuery	17
2.5.1 .submit(handler)	17
2.5.2 .val()	17
2.5.3 .html()	17
2.5.4 .preventDefault()	17
2.6 HTML Content Template (<template>)	17
3 ANALISIS	19
3.1 Analisis Arsitektur Aplikasi Finger For Life	19
3.2 Analisis Aplikasi Sejenis	20
3.3 Analisis Alur Permainan Finger For Life	26

3.4	Analisis <i>Use Case</i>	28
3.4.1	Diagram <i>Use Case</i>	28
3.4.2	Skenario <i>Use Case</i>	29
3.5	Analisis <i>State</i> aplikasi Finger For Life	30
3.6	Analisis Kebutuhan Perangkat Lunak	31
3.7	Analisis Socket.io	37
3.8	Analisis <i>Event</i>	39
4	PERANCANGAN	41
4.1	Perancangan <i>Sequence Diagram</i>	41
4.1.1	<i>Sequence</i> Permintaan Bergabung	41
4.1.2	<i>Sequence</i> Memilih Karakter	43
4.1.3	<i>Sequence</i> Memulai Permainan	44
4.1.4	<i>Sequence</i> Mengakhiri Permainan	46
4.2	Perancangan Antarmuka	47
4.3	Perancangan Struktur Direktori	52
5	IMPLEMENTASI DAN PENGUJIAN	67
5.1	Implementasi	67
5.1.1	Lingkungan Implementasi	67
5.1.2	Hasil Implementasi	68
5.2	Pengujian	73
5.2.1	Pengujian Fungsional	73
5.2.2	Pengujian Eksperimental	77
6	KESIMPULAN DAN SARAN	79
6.1	Kesimpulan	79
6.2	Saran	79
	DAFTAR REFERENSI	81
A	KODE PROGRAM CLIENT	83
A.1	Kode Program Direktori <i>public</i>	83
A.1.1	Kode Program Halaman Utama	83
A.1.2	Kode Program Halaman Permintaan Bergabung	84
A.1.3	Kode Program Halaman Memilih Karakter	88
A.1.4	Kode Program Halaman Memulai Permainan	92
A.1.5	Kode Program Halaman Mengakhiri Permainan	96
A.2	Kode Program Direktori <i>routes</i>	98
A.2.1	Kode Program <i>homeRoutes.js</i>	98
A.3	Kode Program Direktori <i>views</i>	98
A.3.1	Kode Program Halaman <i>Error</i>	98
A.3.2	Kode Program Seluruh Halaman pada Finger For Life	98
A.4	Kode Program <i>app.js</i>	100
B	KODE PROGRAM <i>Server</i>	103
B.1	Kode Program Kelas <i>users.js</i>	103
B.2	Kode Program <i>www</i>	103
C	HASIL PENGUJIAN	107
C.1	Tabel Hasil Pengujian Cross-Platform	107
C.2	Tabel Hasil Pengujian Jaringan Internet	107

DAFTAR GAMBAR

2.1	Struktur modul pada Node.js.	5
3.1	Arsitektur Finger For Life	19
3.2	Halaman awal web AirConsole pada <i>PC</i>	20
3.3	Kode yang harus dimasukan oleh pemain pada <i>smartphone</i>	21
3.4	Halaman awal Airconsole pada <i>smartphone</i>	21
3.5	Pemain diminta untuk memasukkan kode yang sudah didapatkan pada <i>PC</i>	22
3.6	Memasukan kode yang sudah didapatkan pada <i>PC</i>	22
3.7	Halaman pada <i>PC</i> yang menunjukkan berbagai permainan yang dapat dipilih.	23
3.8	Halaman pada <i>smartphone</i> yang berfungsi sebagai pengendali.	23
3.9	Halaman awal permainan The Neighborhood pada <i>PC</i>	24
3.10	Halaman awal permainan The Neighborhood pada <i>smartphone</i>	24
3.11	Halaman pada <i>PC</i> dimana permainan sedang berlangsung.	24
3.12	Halaman pada <i>smartphone</i> dimana permainan sedang berlangsung.	25
3.13	Halaman pada <i>PC</i> apabila permainan sudah selesai.	25
3.14	Halaman pada <i>smartphone</i> apabila permainan sudah selesai.	25
3.15	Halaman pada <i>PC</i> yang menunjukkan pemutusan koneksi.	26
3.16	Diagram <i>use case</i> pemain	28
3.17	Diagram <i>state</i> aplikasi Finger For Life	31
3.18	Efek <i>destination-over</i>	36
3.19	Arsitektur interaksi <i>client</i> dan <i>server</i>	38
4.1	Proses melakukan koneksi ke Socket.io dan bergabung kedalam <i>room</i>	41
4.2	Proses memilih karakter.	43
4.3	Proses memulai permainan.	44
4.4	Menampilkan para pemain yang telah selesai bermain.	46
4.5	Halaman pada <i>PC</i> yang menunjukan halaman utama saat <i>client</i> mengakses alamat web.	47
4.6	Halaman pada <i>smartphone</i> yang menunjukan halaman utama saat <i>client</i> mengakses alamat web.	48
4.7	Halaman pada <i>PC</i> yang menampilkan langkah untuk bergabung kedalam permainan.	48
4.8	Halaman pada <i>smartphone</i> yang menampilkan kolom untuk mengisi kode.	49
4.9	Halaman pada <i>PC</i> yang menampilkan karakter yang telah ditetapkan oleh pemain.	49
4.10	Halaman pada <i>smartphone</i> yang menampilkan daftar karakter yang dapat dipilih.	50
4.11	Halaman pada <i>PC</i> yang menampilkan lintasan lari dan karakter untuk dimainkan.	50
4.12	Halaman pada <i>smartphone</i> yang menampilkan telapak kaki yang berfungsi sebagai pengendali.	51
4.13	Halaman pada <i>PC</i> yang menampilkan pemenang permainan.	51
4.14	Halaman pada <i>smartphone</i> yang menampilkan teks bahwa permainan telah selesai.	52
5.1	Halaman utama pada <i>PC</i>	69
5.2	Halaman utama pada <i>smartphone</i>	69
5.3	Halaman permintaan bergabung pada <i>PC</i>	70

5.4	Halaman permintaan bergabung pada <i>smartphone</i>	70
5.5	Halaman memilih karakter pada <i>PC</i>	71
5.6	Halaman memilih karakter pada <i>smartphone</i>	71
5.7	Halaman memulai permainan pada <i>PC</i>	72
5.8	Halaman memulai permainan pada <i>smartphone</i>	72
5.9	Halaman mengakhiri permainan pada <i>PC</i>	73
5.10	Halaman mengakhiri permainan pada <i>smartphone</i>	73
5.11	Grafik pengujian <i>cross-platform</i>	78
5.12	Grafik pengujian jaringan internet.	78

DAFTAR TABEL

3.1	Tabel daftar <i>event</i> yang terdapat pada berkas <i>www</i>	40
4.1	Tabel daftar <i>method</i> pada berkas <i>users.js</i>	53
4.2	Daftar atribut yang terdapat pada berkas <i>www</i>	53
4.3	Tabel daftar <i>event</i> yang terdapat pada berkas <i>www</i>	54
4.4	Tabel daftar gambar pada direktori <i>images</i>	55
4.5	Tabel daftar atribut pada berkas <i>charDesktopScript.js</i>	56
4.6	Tabel daftar <i>event</i> yang dimiliki oleh berkas <i>charDesktopScript.js</i>	56
4.7	Tabel daftar <i>event</i> yang dimiliki oleh berkas <i>charMobileScript.js</i>	57
4.8	Daftar atribut yang dimiliki oleh berkas <i>gamePlayDesktopScript.js</i>	58
4.9	Daftar <i>method</i> yang ada pada berkas <i>gameplayDesktopScript.js</i>	59
4.10	Tabel daftar <i>event</i> yang terdapat pada berkas <i>gameplayDesktopScript.js</i>	59
4.11	Tabel daftar <i>method</i> yang dimiliki oleh berkas <i>gamePlayMobileScript.js</i>	60
4.12	Tabel daftar <i>event</i> yang dimiliki oleh berkas <i>gamePlayMobileScript.js</i>	60
4.13	Tabel daftar <i>event</i> yang dimiliki oleh berkas <i>mobileScript.js</i>	61
4.14	Tabel daftar <i>atribut</i> yang dimiliki oleh berkas <i>syncScript.js</i>	61
4.15	Tabel daftar <i>method</i> yang dimiliki oleh berkas <i>syncScript.js</i>	61
4.16	Tabel daftar <i>event</i> yang dimiliki oleh berkas <i>syncScript.js</i>	62
4.17	Tabel daftar atribut yang dimiliki oleh berkas <i>winningDesktopScript.js</i>	62
4.18	Tabel daftar <i>method</i> yang dimiliki oleh berkas <i>winningDesktopScript.js</i>	63
4.19	Tabel daftar <i>event</i> yang dimiliki oleh berkas <i>winningDesktopScript.js</i>	63
4.20	Tabel daftar <i>event</i> yang dimiliki oleh berkas <i>winningMobileScript.js</i>	63
4.21	Tabel daftar CSS yang dimiliki oleh direktori <i>stylesheets</i>	64
4.22	Tabel daftar atribut yang dimiliki oleh berkas <i>app.js</i>	65
5.1	Tabel Pengujian fungsional pada <i>PC</i>	76
5.2	Tabel Pengujian Fungsional pada <i>smartphone</i>	77
C.1	Tabel hasil pengujian <i>cross-platform</i>	107
C.2	Tabel hasil pengujian jaringan internet.	107

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Teknologi *smartphone* dan *Personal Computer (PC)* dimanfaatkan untuk mengakses berbagai macam layanan aplikasi yang tersedia. Beberapa aplikasi yang dapat diakses adalah *web browser* dan aplikasi permainan. Kedua jenis aplikasi tersebut dapat diakses melalui *smartphone* maupun *PC*. Salah satu aplikasi yang dapat memanfaatkan kedua gawai tersebut adalah permainan berbasis web yang memanfaatkan pengendali.

Permainan berbasis web adalah aplikasi permainan yang diakses melalui browser yang membutuhkan jaringan internet. Teknologi *smartphone* dapat dimanfaatkan sebagai pengendali yang memainkan permainan berbasis web melalui browser. *PC* berperan sebagai *console* yang menyediakan permainan sehingga *smartphone* dapat menjadi pengendali permainan. Untuk dapat memanfaatkan *smartphone* sebagai pengendali permainan berbasis web, *PC* dan *smartphone* harus terhubung satu sama lain.

Smartphone harus membuka *browser* untuk mengakses permainan berbasis web yang sama dengan yang ada di *browser* pada *PC*. Dengan memanfaatkan *smartphone* dan *PC* di dalam aplikasi permainan berbasis web maka dibutuhkan ukuran *latency* yang kecil. *Latency* adalah jeda waktu yang dihasilkan pada saat *client* mengirimkan data kepada *server* maupun sebaliknya. *Latency* yang dapat memberikan respon cepat pada aplikasi permainan berbasis web adalah sekitar 100 milidetik, atau sekitar 0,100 detik¹. Semakin kecil ukuran *latency* yang dihasilkan maka semakin cepat respon yang diterima.

Agar *smartphone* dan *PC* dapat terhubung satu sama lain, maka dibutuhkan suatu pustaka yang dapat memenuhi hal tersebut. Pustaka yang dapat digunakan adalah Socket.io.

Socket.io adalah pustaka yang memungkinkan *client* dan *server* untuk melakukan komunikasi dua arah secara *real-time* yang artinya jeda waktu yang sedikit antara permintaan yang dilakukan oleh *client* dengan respon yang diberikan oleh *server* maupun sebaliknya [1]. Socket.io memiliki dua bagian: *client-side library*, atau pustaka pada bagian *client* yang berjalan di dalam *web browser*, dan *server-side library*, atau pustaka pada bagian *server* yang berjalan pada bagian *server*. Socket.io memiliki fitur untuk melakukan komunikasi dari satu *server* ke beberapa *client* di dalam proses implementasinya. Teknologi ini sangat berguna untuk membantu membangun sebuah aplikasi yang membutuhkan koneksi *real-time* seperti di dalam aplikasi permainan berbasis web.

Untuk memanfaatkan teknologi Socket.io dalam membangun aplikasi permainan dibutuhkan beberapa teknologi yang dapat membantu pembangunan aplikasinya. Salah satu teknologi tersebut adalah Canvas API. Teknologi ini merupakan bagian dari elemen HTML5 yang dapat digunakan untuk mengolah objek grafis dengan menggunakan JavaScript [2]. *Canvas API* dapat juga digunakan untuk mengolah komposisi foto dan membuat animasi. Oleh karena itu, fungsi-fungsi yang ada pada *Canvas API* membantu pembangunan aplikasi permainan terutama pada bagian pengembangan grafis.

Teknologi lain yang dapat membantu membangun aplikasi permainan dalam menggunakan

¹<https://forum.unity.com/threads/question-about-acceptable-levels-of-latency-in-online-gaming.261271/>, diakses 2 Januari 2019

teknologi Socket.io adalah Node.js. Teknologi ini merupakan sebuah *platform* atau lingkungan yang didesain untuk mengembangkan aplikasi berbasis web pada bagian *server* [3]. Node.js ditulis dalam sintaks bahasa pemrograman JavaScript dan menggunakan *V8* yang merupakan *engine* JavaScript milik perusahaan *Google* untuk mengeksekusi JavaScript pada *web server*. Node.js memiliki sifat *non-blocking* yang artinya adalah Node.js tidak akan menunggu untuk mengerjakan permintaan selanjutnya. Oleh karena itu, fitur-fitur yang dimiliki oleh Node.js akan sangat membantu untuk membangun aplikasi permainan yang membutuhkan koneksi *real-time*.

Salah satu teknologi yang akan membantu dalam mengimplementasi Node.js adalah Express.js [4]. Teknologi ini menyediakan kumpulan fitur untuk mengatur penyimpanan data secara lokal dalam membangun aplikasi web maupun *mobile*. Pada proses implementasinya Express.js akan mengolah data lokal sedemikian rupa sehingga dapat dengan mudah diakses apabila diperlukan. Express.js hanya dapat digunakan untuk membangun aplikasi apabila aplikasi tersebut berjalan di dalam lingkungan Node.js. Oleh karena itu, fitur-fitur yang dimiliki oleh Express.js akan membantu di dalam pembangunan aplikasi berbasis Node.js.

Penelitian yang dilakukan merupakan aplikasi permainan berbasis web yang memanfaatkan teknologi *PC* dan *smartphone*. Oleh karena itu dibutuhkan teknologi yang dapat mengatur tampilan halaman pada layar *PC* maupun *smartphone*. Teknologi yang digunakan adalah HTML Content Template (<template>) [5]. Teknologi ini merupakan bagian dari elemen HTML5, yang berfungsi untuk menyimpan seluruh elemen-elemen HTML untuk ditampilkan ke layar *browser* pada *PC* maupun *smartphone*. Di dalam satu berkas HTML, elemen <template> dapat berjumlah lebih dari satu. Dengan begitu, beberapa halaman dapat dipilih untuk ditampilkan dalam satu waktu tertentu, sebelum menampilkan halaman lain yang disimpan oleh <template>. Proses menampilkan halaman ke layar *PC* maupun *smartphone* akan menggunakan JavaScript.

Teknologi yang akan membantu dalam penggunaan <template> adalah jQuery [6]. Teknologi ini merupakan pustaka JavaScript yang menyediakan fitur-fitur untuk mengatur berbagai elemen HTML. Pustaka ini memiliki fitur untuk memanipulasi berkas HTML. Dengan begitu, jQuery dapat mengatur untuk menampilkan <template> mana yang akan ditampilkan ke layar *PC* maupun *smartphone*.

Pada skripsi ini akan dibuat sebuah aplikasi permainan berbasis web yang memanfaatkan Socket.io. Aplikasi yang dibuat akan memanfaatkan *personal computer (PC)* dan *smartphone* untuk pengembangan aplikasinya. Para pemain akan mengkoneksikan *smartphone* pada suatu *PC* yang akan berfungsi sebagai *console* dan *smartphone* tersebut akan berfungsi sebagai *controller* untuk memainkan permainannya. Socket.io akan digunakan sebagai koneksi antara *smartphone* dan *PC* dalam aplikasi permainan yang akan dibangun. Aplikasi permainan akan dibangun berdasarkan Node.js sehingga proses eksekusi JavaScript dapat dilakukan pada *server*. Pengaturan struktur direktori di dalam pengembangan aplikasi akan menggunakan Express.js. Di dalam proses pengaturan elemen grafis yang dibutuhkan di dalam aplikasi, teknologi Canvas API akan digunakan di dalam pengembangannya. Elemen <template> akan digunakan untuk menampilkan setiap halaman-halaman web yang dibutuhkan di dalam pengembangan aplikasi permainan. Teknologi jQuery akan digunakan untuk pengaturan berbagai elemen HTML di dalam aplikasi. Aplikasi permainan akan menggunakan teknologi berbasis web sehingga untuk memainkannya, *client* harus memiliki akses internet dan mengakses alamat aplikasi permainan menggunakan *browser*.

1.2 Rumusan Masalah

1. Bagaimana membangun aplikasi permainan berbasis web dengan memanfaatkan Socket.io untuk penggunaan *smartphone* sebagai pengendali permainan berbasis web ?
2. Berapa *latency* yang dihasilkan berdasarkan penggunaan Socket.io ?

1.3 Tujuan

1. Mengetahui cara membangun aplikasi permainan berbasis web dengan memanfaatkan Socket.io untuk penggunaan *smartphone* sebagai pengendali permainan berbasis web.
2. Mengetahui jumlah *latency* yang dihasilkan berdasarkan pemanfaatan Socket.io.

1.4 Batasan Masalah

Batasan masalah yang dibuat terkait dengan pengerjaan skripsi ini adalah sebagai berikut:

- Aplikasi permainan yang dibuat merupakan permainan *multiplayer* yang hanya bisa dimainkan oleh dua orang saja.

1.5 Metodologi

Metodologi yang dilakukan dalam pengerjaan skripsi ini adalah sebagai berikut:

1. Studi literatur mengenai :
 - *Socket.io* sebagai teknologi yang akan menghubungkan *smartphone* dan *PC*.
 - *Canvas API* yang akan digunakan untuk antarmuka permainan.
 - *Node.js* sebagai *web server* dalam pembangunan aplikasi.
 - *Express.js* sebagai *Node.js framework* yang akan digunakan untuk mengatur penyimpanan data.
 - *jQuery* yang akan digunakan dalam pengaturan elemen HTML.
 - *The Content Template element* yang akan digunakan untuk menampilkan halaman-halaman HTML.
2. Menganalisis aplikasi sejenis.
3. Merancang antarmuka permainan pada *PC* dan *smartphone*. Antarmuka pada *PC* akan berbeda dengan yang ada di *smartphone*, karena *smartphone* akan bekerja sebagai *controller* dan *PC* akan bekerja sebagai *console*.
4. Menyusun cara bermain aplikasi permainan yang dibangun.
5. Mengimplementasi program aplikasi permainan berbasis web.
6. Menganalisis *latency* yang dihasilkan pada aplikasi.
7. Melakukan eksperimen dan pengujian yang melibatkan responden.

1.6 Sistematika Pembahasan

Setiap bab dalam skripsi ini memiliki sistematika penulisan yang dijelaskan ke dalam poin-poin sebagai berikut:

1. Bab 1 : Pendahuluan
Membahas mengenai gambaran umum penelitian ini. Berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metode penelitian, dan sistematika penulisan.

2. Bab 2 : Dasar Teori
Membahas mengenai teori-teori yang mendukung berjalannya penelitian ini. Berisi tentang Socket.io, Node.js, Express.js, Canvas API, jQuery, dan The Content Template element.
3. Bab 3 : Analisis
Membahas mengenai analisa masalah.
4. Bab 4 : Perancangan
Membahas mengenai perancangan yang dilakukan sebelum melakukan tahapan implementasi.
5. Bab 5 : Implementasi dan Pengujian
Membahas mengenai implementasi dan pengujian yang telah dilakukan.
6. Bab 6 : Kesimpulan dan Saran
Membahas hasil kesimpulan dari keseluruhan penelitian ini dan saran-saran yang dapat diberikan untuk penelitian berikutnya.