

Perekayasa Teknologi XML Dan XMPP Dalam Membangun Aplikasi Jaringan Multiplayer Online Game Ular Tangga Berbasis Web Dengan Mengimplementasikan Teknologi Bi-Directional Stream Over Synchronous Http

Oleh:

Ir. Zefri Yenni, MM ; Haddad Shamir*)

Universitas Putra Indonesia "YPTK" Padang

Intisari

Penelitian ini merekayasa teknologi Extensible Markup Language (XML) dan Extensible Messaging and Presence Protocol (XMPP) untuk menghadirkan komunikasi realtime pada aplikasi web. Dengan tujuan mengaplikasi jaringan permainan ular tangga multiplayer berbasis web, dengan memanfaatkan software library jQuery sebagai alat bantu perekayasa XML dan Strophe.js yang mengimplementasikan protokol XMPP dalam JavaScript yang keduanya dibangun menggunakan JavaScript.

Penggunaan teknik *Bi-directional Stream Over Synchronous* HTTP meningkatkan respon dan kinerja jaringan dimana aplikasi *web* bekerja tanpa melakukan *refreshing* halaman. Dengan menggunakan utilitas *Strophe Builder* yang tersedia pada library Strophe.js. *Stanza* XML direkayasa dengan mengacu kepada protokol permainan yang telah didefinisikan. Pendekatan dengan penggunaan permainan ular tangga dapat membantu anak-anak menghafal hari-hari besar nasional.

Kata kunci : Jaringan Multiplayer Online, XML, XMPP dan Java Script

I. PENDAHULUAN

Salah satu teknologi yang memungkinkan komputer dapat saling berkomunikasi adalah XML (*Extensible Markup Language*). Sesuai dengan sifatnya yang "*Extensible*", XML dapat diperluas untuk memenuhi kebutuhan programmer dalam menciptakan protocol komunikasi.

Protokol XMPP (*Extensible Messaging and Presence Protocol*) adalah salah satu protokol yang dibangun di atas teknologi XML. Protokol ini memungkinkan komunikasi data berbasis XML secara *real time*. Sesuai dengan sifatnya yang "*Extensible*", protokol ini dapat diperluas sesuai kebutuhan. Untuk itu penulis melakukan penelitian untuk merekayasa teknologi XML dan XMPP dalam membangun aplikasi jaringan *Multiplayer Online Game* "Ular Tangga" berbasis Web. Dimana dalam penelitian ini akan memanfaatkan teknologi *Opensource* dan *Openstandard*, termasuk didalamnya teknologi HTTP (*Hypertext Transfer Protocol*), BOSH (*Bi-directional Stream Over Synchronous HTTP*) serta JavaScript Framework jQuery dan Strophe.js.

Implementasi teknologi *Bi-directional Stream Over Synchronous HTTP* (BOSH) yang digunakan dalam penelitian ini akan

memberikan perbandingan metoda BOSH dengan metoda *polling* yang umum digunakan. Berbeda dengan teknologi *polling* biasa yang melakukan *sampling* untuk mendapatkan status ketersediaan data melalui interval waktu tertentu, BOSH membuka soket (koneksi) selama mungkin (dikenal dengan sebutan *long polling*) sehingga seketika data tersedia, data dapat langsung diproses tanpa harus menunggu penjemputan data sesuai dengan interval waktu *polling*.

2. TINJAUAN PUSTAKA

a. Unified Modelling Language (UML)

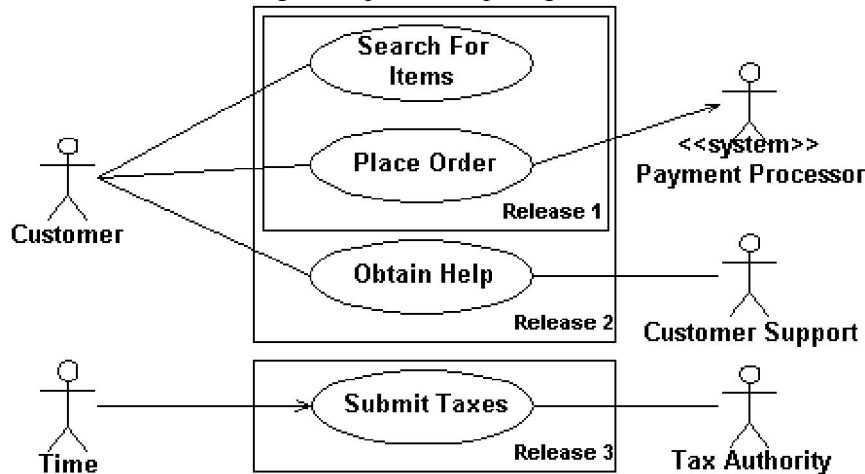
Unified Modeling Language (UML) adalah bahasa pemodelan *general-purpose* terstandar dalam bidang perekayasa perangkat lunak berorientasi objek.

UML digunakan untuk mencirikan, menampilkan, mengubah, membangun dan mendokumentasikan artefak dari sistem perangkat lunak berorientasi objek dalam pengembangannya

b. Use Case Diagram

Use case diagram adalah diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Diagram ini secara umum merupakan pola perilaku sistem dan urutan transaksi yang berhubungan yang dilakukan oleh satu aktor. *Association* digunakan untuk

menggambarakan bagaimana *actor* terlibat dalam *use case*.
Contoh *use case* diagram dapat dilihat pada gambar 1 berikut.

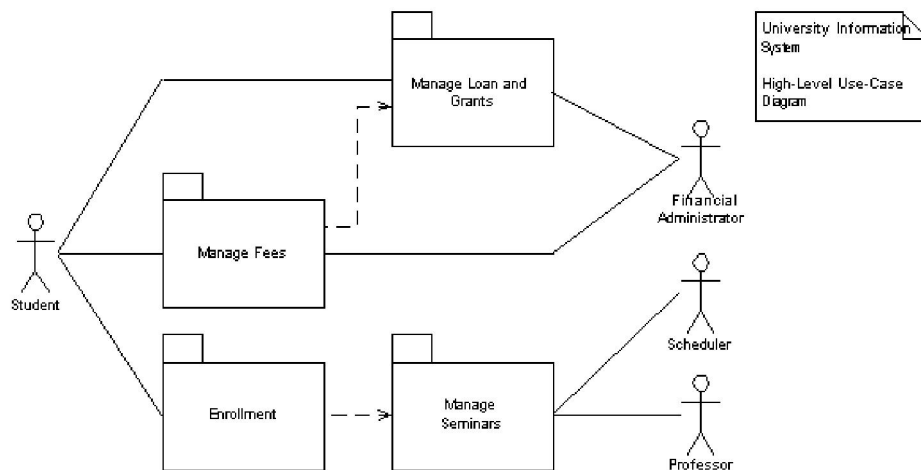


Gambar 1. Contoh Use Case Diagram

c. Package Diagram

Package diagram merupakan diagram yang memperlihatkan bagaimana elemen model diorganisasikan/dikelompokkan kedalam *packages*. *Packages* digunakan untuk

mengorganisasikan sebuah diagram yang besar kedalam menjadi beberapa diagram kecil. *Packages* digambarkan sebagai direktori / file folders yang berisi model-model elemen.



Gambar 2. Contoh Package Diagram

d. HTTP

HTTP berfungsi sebagai protokol *request-response* dalam komputasi *client-server*. Sebuah *web browser* akan berperan sebagai *client* sedangkan aplikasi yang melayani *request* akan berfungsi sebagai *server*. *Client* akan mengirimkan HTTP *request message* kepada *server*, sedangkan *server* yang menyimpan konten atau menyediakan *resource* (misalnya *file HTML*)

atau mengerjakan fungsi lain, akan mengembalikan *response message* kepada *client*.

Protokol HTTP didesain untuk mengizinkan mediator untuk meningkatkan kinerja atau menyanggupkan komunikasi antara *client* dan *server*. *Website* dengan trafik tinggi mendapatkan keuntungan dengan server web cache yang menyampaikan konten atas nama server asli, yang disebut *origin server* untuk meningkatkan waktu respon. HTTP

Proxy memfasilitasi komunikasi saat *client* berada pada jaringan privat dengan menjadi mediator antara *client* dan *server*.

Resource HTTP diidentifikasi dan ditemukan pada jaringan dengan menggunakan *Uniform Resource Identifiers* (URI) dan *Uniform Resource Locator* (URL). URL akan digunakan oleh *client* untuk melakukan permintaan (*request*) terhadap sebuah *resource* dengan berbasis kepada skema URI *http* atau *https*.

e. HTTP Request Method

Protokol HTTP mendefinisikan sembilan metode yang mengindikasikan aksi yang akan dikerjakan pada sebuah *resource*.

f. Web Server

Web server adalah sekumpulan *hardware* maupun *software* yang berfungsi mengantarkan konten *web* yang dapat diakses melalui internet. *Web server* secara umum digunakan untuk *meng-hosting website*, *application server*, atau dalam penelitian ini digunakan sebagai *proxy* yang memediasi komunikasi antara aplikasi jaringan (*javascript*) dengan XMPP *server*.

g. Extensible Markup Language (XML)

XML (*eXtensible Markup Language*) adalah sebuah bahasa markah untuk mendeskripsikan data. XML merupakan turunan (*subset*) atau versi ringkas dari SGML (*Standard Generalized Markup Language*), sedangkan SGML merupakan sebuah standar ISO untuk format dokumen. SGML tidak berisi berupa *tag-tag* siap pakai seperti halnya bahasa HTML, melainkan berupa aturan-aturan standar dalam pembuatan *tag-tag* format dokumen. SGML banyak dipakai untuk mengelola dokumen dalam jumlah besar, frekuensi revisi tinggi dan dibutuhkan dalam beragam format tampilan. SGML jarang dipakai karena sangat rumit dan kompleks. XML dibuat dengan konsep yang lebih sederhana dan ringkas, tujuannya agar bisa dipakai sebagai aplikasi *desktop* dan jaringan internet

h. Java Script

JavaScript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML, bahasa ini adalah bahasa skrip pertama untuk *web*. Bahasa pemrograman ini memberikan kemampuan tambahan terhadap bahasa HTML dengan mengizinkan pengeksekusian perintah-perintah di sisi *client*, yang artinya di sisi *browser* bukan di sisi *server web*. JavaScript mampu mengakses *Document Object Model* serta melakukan manipulasi terhadapnya.

i. Extensible Messaging and Presence Protocol (XMPP)

Extensible Messaging and Presence Protocol (XMPP) adalah protokol komunikasi berstandar terbuka untuk *message oriented middleware* berbasis XML.

j. BOSH

BOSH membutuhkan mediator antara *client* XMPP dan *server* XMPP yang disebut *connection manager*. *Connection manager* pada dasarnya adalah sebuah HTTP *server* yang berfungsi menjembatani koneksi antara *client* dengan *server* XMPP.

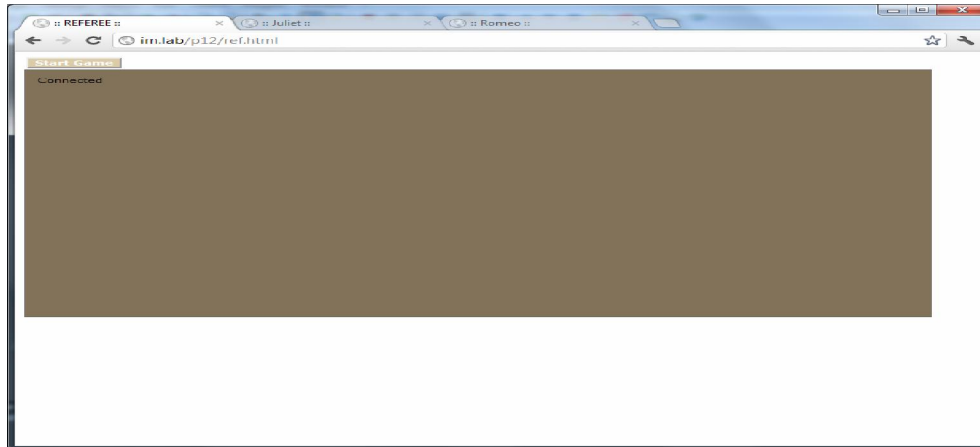
3. PENGUJIAN SISTEM

1. Pengujian Aplikasi Referee

Aplikasi *referee* adalah aplikasi yang mengontrol jalannya permainan ular tangga. Aplikasi ini juga berfungsi untuk memediasi komunikasi antar aplikasi *player*.

a. Eksekusi Aplikasi Referee

Pada keadaan ini, aplikasi *referee* menunggu permintaan *register* dari aplikasi *player*. Tombol “*Start Game*” dalam keadaan *disabled*. Tombol tersebut akan *enable* jika sudah ada 2 aplikasi *player* yang melakukan *register*

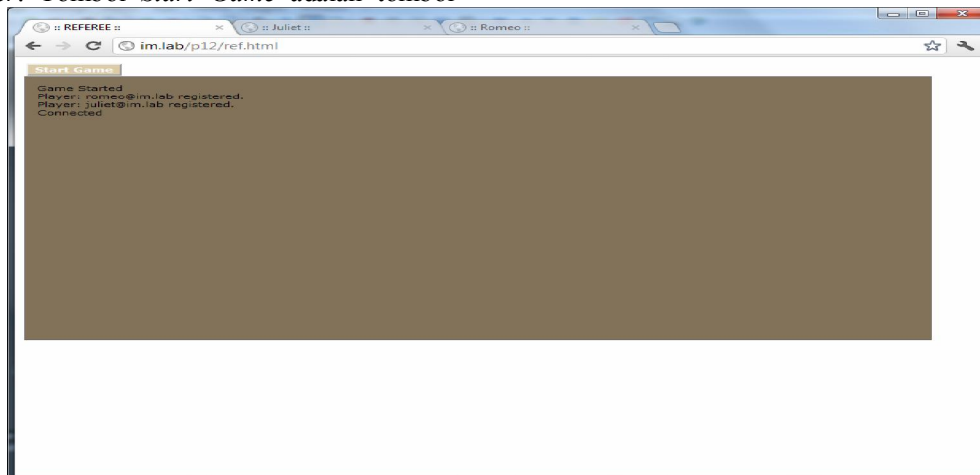


Gambar 3. Eksekusi Aplikasi Referee

b.. Eksekusi Aplikasi *Referee* Memulai Permainan

Tombol *Start Game* akan aktif jika aplikasi romeo dan aplikasi juliet melakukan *register*. Tombol *Start Game* adalah tombol

yang berfungsi untuk menginisialisasi permainan yang jika diklik akan memulai rutin permainan seperti yang terlihat pada gambar 4 di bawah

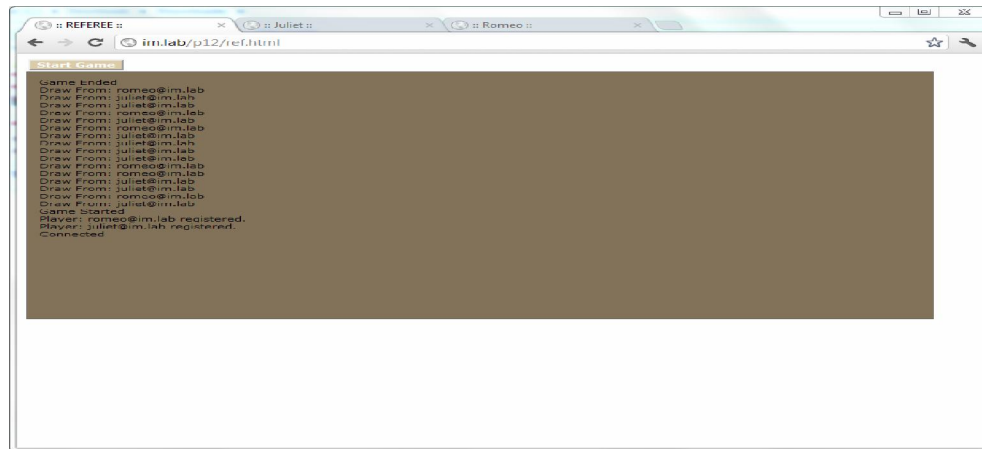


Gambar 4. Eksekusi Aplikasi *Referee* Memulai Permainan

c. Eksekusi Aplikasi *Referee* Pada Saat Permainan Berakhir

Aplikasi *referee* akan menampilkan *log* selama permainan. *Log* ini member laporan

sinyal *draw* sekaligus menampilkan status permainan dari permainan dimulai hingga selesai seperti yang terlihat pada gambar 5 di bawah.



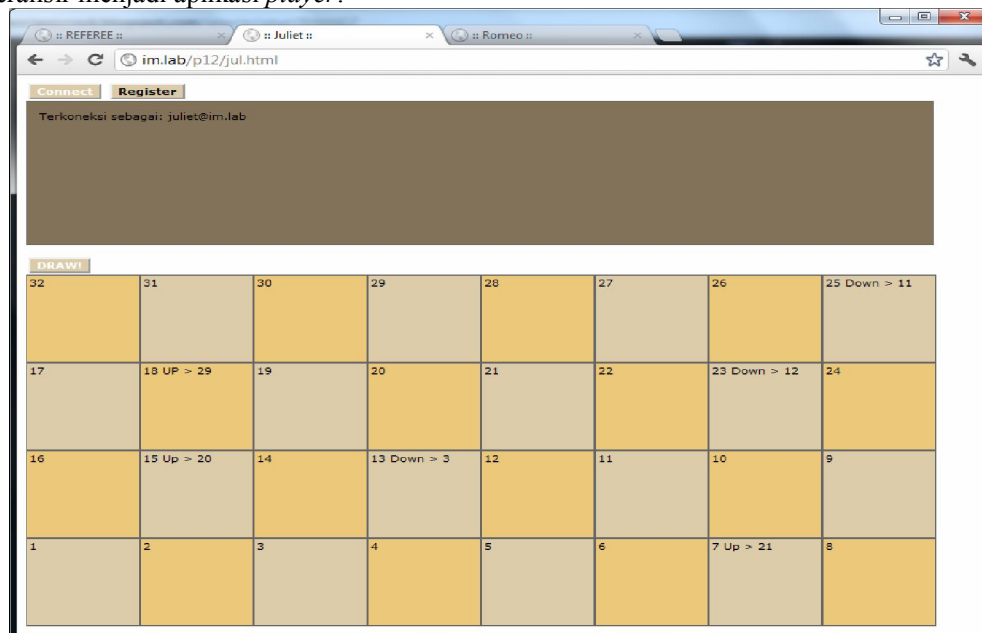
Gambar 5. Eksekusi Aplikasi *Referee* Pada Saat Permainan Berakhir

2 Pengujian Aplikasi *Player*

Aplikasi *Player* adalah aplikasi antar muka pemain ular tangga. Aplikasi ini berjumlah dua buah yaitu diberi nama “Juliet” dan “Romeo”. Juliet dan Romeo dapat digeneralisir menjadi aplikasi *player*.

a. Eksekusi Aplikasi *Player Juliet*

Aplikasi Juliet adalah aplikasi yang menyediakan antar muka permainan ular tangga. Aplikasi ini memiliki tombol operasi, penampil *log* permainan dan papan permainan seperti yang terlihat pada gambar 6.

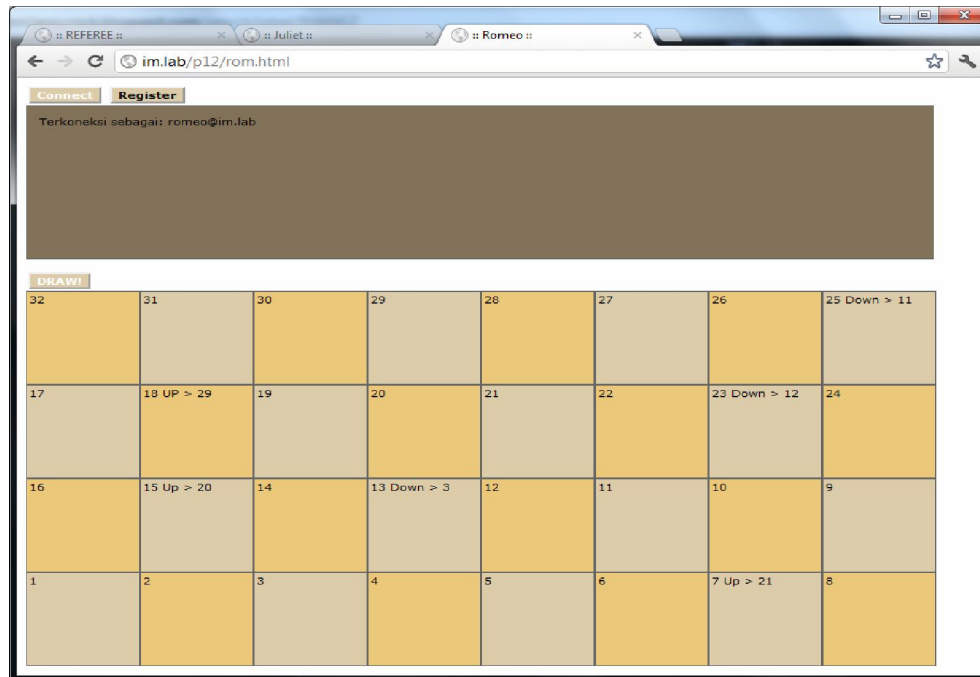


Gambar 6. Eksekusi Aplikasi *Player Juliet*

b. Eksekusi Aplikasi *Player Romeo*

Aplikasi Romeo adalah aplikasi yang menyediakan antar muka permainan ular

tangga. Aplikasi ini memiliki tombol operasi, penampil *log* permainan dan papan permainan seperti yang terlihat pada gambar 7.

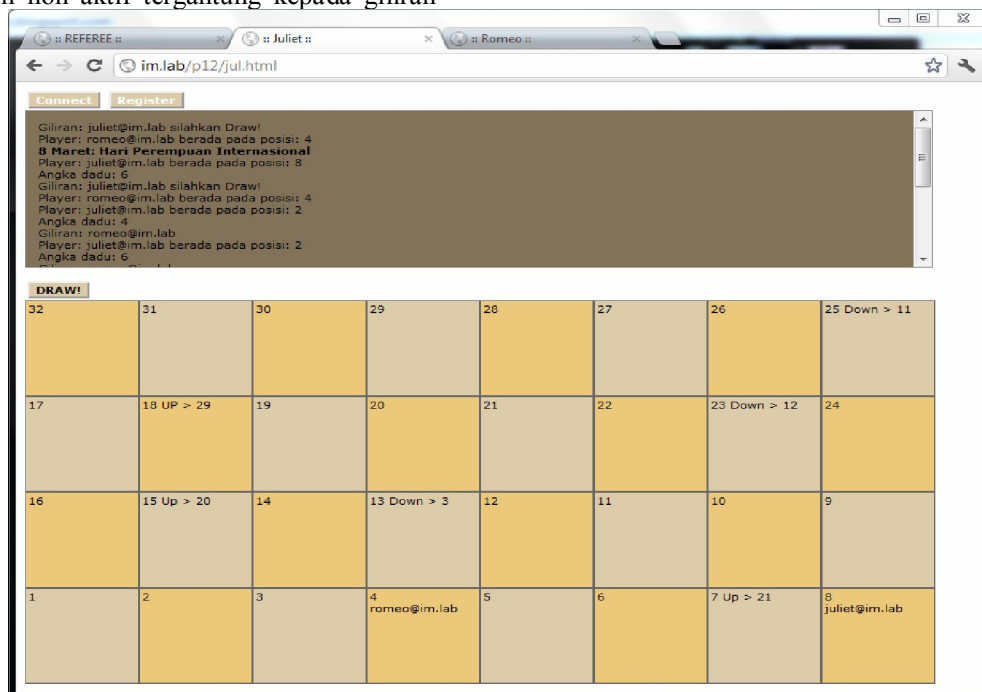


Gambar 7. Eksekusi Aplikasi *Player* Romeo

c. Eksekusi Aplikasi *Player* Saat Permainan Berjalan

Tombol “DRAW!” akan bergantian aktif dan non aktif tergantung kepada giliran

pelemparan dadu. Aplikasi juga menampilkan informasi angka dadu, giliran pelemparan dadu dan informasi hari-hari penting jika bidak berada pada kotak tertentu.

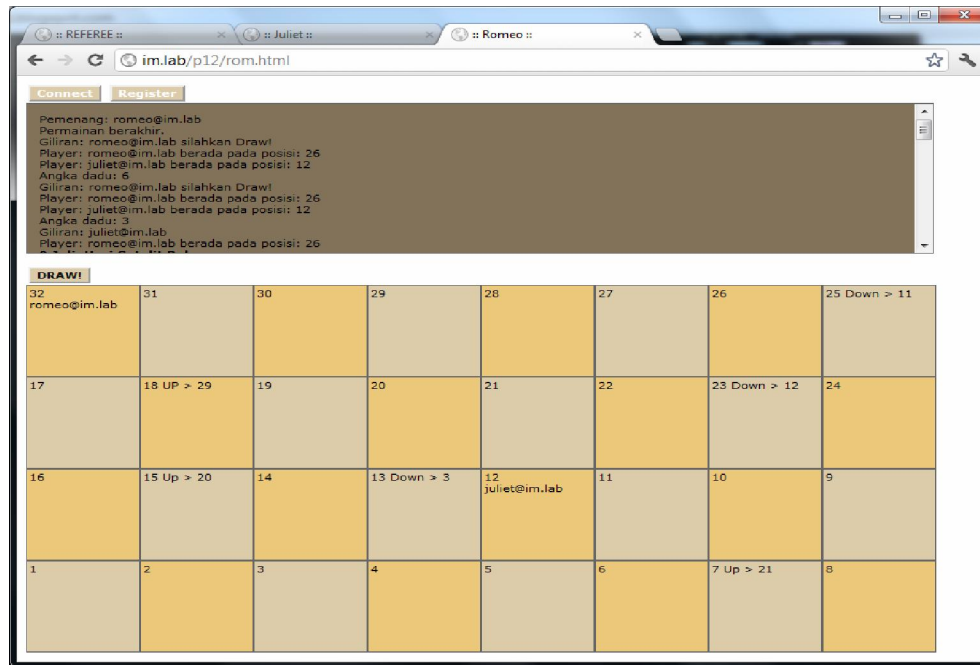


Gambar 8. Eksekusi Aplikasi *Player* Saat Permainan Berjalan

d. Eksekusi Aplikasi *Player* Pada Saat Permainan Berakhir

pemenang pada unit penampil log seperti yang terlihat pada gambar 9 Di bawah.

Aplikasi player akan menampilkan status permainan berakhir dan player



Gambar 9. Eksekusi Aplikasi *Player* Pada Saat Permainan Berakhir

3 Pengujian Jaringan

Pengujian jaringan dilakukan untuk membuktikan koneksi BOSH yang dibangun. Pengujian ini membuktikan bahwa koneksi BOSH berjalan dengan baik yang memungkinkan komunikasi *real time* dapat berlangsung pada aplikasi *web*.

a. Pengujian Jaringan Aplikasi *Referee*

Gambar 10 di bawah menunjukkan bahwa Strophe.js melakukan koneksi ke `http-bind` selama 55,42 detik yang menandakan mekanisme BOSH sedang bekerja. Mekanisme ini membuka koneksi dalam waktu yang lama kepada *server* sehingga data dapat dikirimkan kepada *client* seketika data tersebut tersedia tanpa harus melakukan *polling* secara berkala.

Name Path	Met...	Status Text	Type	Initiator	Size Conte	Time Laten	Timeline
ref.html	GET	304 Not Mod	text/...	Other	290B	14ms	
jquery.js	GET	304 Not Mod	appli...	ref.html:7	291B	14ms	
strophe.min.js	GET	304 Not Mod	appli...	ref.html:7	289B	68ms	
http-bind	POST	200 OK	text/...	strophe.min.js	865B	37ms	
http-bind	POST	200 OK	text/...	strophe.min.js	458B	18ms	
http-bind	POST	200 OK	text/...	strophe.min.js	366B	20ms	
http-bind	POST	200 OK	text/...	strophe.min.js	527B	17ms	
http-bind	POST	200 OK	text/...	strophe.min.js	400B	30ms	
http-bind	POST	200 OK	text/...	strophe.min.js	377B	17ms	
http-bind	POST	200 OK	text/...	strophe.min.js	320B	16ms	
http-bind	POST	200 OK	text/...	strophe.min.js	339B	55.42s	
http-bind	POST	200 OK	text/...	strophe.min.js	13B	0B	

12 requests | 4.43KB transferred | 56.33s (onload: 149ms, DOMContentLoaded: 148ms)

Gambar 10. Pengujian Jaringan Aplikasi Referee

b. Pengujian Jaringan Aplikasi Player Juliet

Gambar 11 di bawah menunjukkan bahwa Strophe.js melakukan koneksi ke http-bind selama 32,4 detik yang menandakan

mekanisme BOSH sedang bekerja. Mekanisme ini membuka koneksi dalam waktu yang lama kepada *server* sehingga data dapat dikirimkan kepada *client* seketika data tersebut tersedia tanpa harus melakukan *polling* secara berkala.

Name Path	Met...	Status Text	Type	Initiator	Size Conte	Time Laten	Timeline
jquery.fancybox-th	GET	304 Not Mod	appli...	ju.html:10	287B	94ms	
jquery.fancybox-m	GET	304 Not Mod	appli...	ju.html:10	288B	120ms	
http-bind	POST	200 OK	text/...	strophe.min.js	867B	21ms	
http-bind	POST	200 OK	text/...	strophe.min.js	458B	17ms	
http-bind	POST	200 OK	text/...	strophe.min.js	366B	19ms	
http-bind	POST	200 OK	text/...	strophe.min.js	527B	16ms	
http-bind	POST	200 OK	text/...	strophe.min.js	334B	7ms	
http-bind	POST	200 OK	text/...	strophe.min.js	399B	17ms	
http-bind	POST	200 OK	text/...	strophe.min.js	376B	15ms	
http-bind	POST	200 OK	text/...	strophe.min.js	319B	19ms	
http-bind	POST	200 OK	text/...	strophe.min.js	126B	9ms	
http-bind	POST	200 OK	text/...	strophe.min.js	738B	17ms	
http-bind	POST	200 OK	text/...	strophe.min.js	545B	7ms	
http-bind	POST	200 OK	text/...	strophe.min.js	330B	32.40s	
http-bind	POST	200 OK	text/...	strophe.min.js	145B	32.39s	
http-bind	POST	200 OK	text/...	strophe.min.js	13B	0B	

21 requests | 7.40KB transferred | 46.24s (onload: 197ms, DOMContentLoaded: 197ms)

Gambar 11. Pengujian Jaringan Aplikasi Juliet

c. Pengujian Jaringan Aplikasi *Player Romeo*

Gambar 12 di bawah menunjukkan bahwa Strophe.js melakukan koneksi ke http-bind selama 1 menit yang menandakan

mekanisme BOSH sedang bekerja. Mekanisme ini membuka koneksi dalam waktu yang lama kepada *server* sehingga data dapat dikirimkan kepada *client* seketika data tersebut tersedia tanpa harus melakukan *polling* secara berkala.

Name	Path	Method	Status	Type	Initiator	Size	Time	Timeline
jquery.fancybox-th	/js/fancybox/source/	GET	200 OK	text/...	rom.html:25	(fro...	Pen...	
jquery.fancybox-th	/js/fancybox/source/	GET	200 OK	appli...	rom.html:26	(fro...	Pen...	
jquery.fancybox-m	/js/fancybox/source/	GET	200 OK	appli...	rom.html:29	(fro...	Pen...	
http-bind		POST	200 OK	text/...	strophe.min.js	867B	21ms	
http-bind		POST	200 OK	text/...	strophe.min.js	673B	11ms	
http-bind		POST	200 OK	text/...	strophe.min.js	458B	18ms	
http-bind		POST	200 OK	text/...	strophe.min.js	265B	8ms	
http-bind		POST	200 OK	text/...	strophe.min.js	366B	25ms	
http-bind		POST	200 OK	text/...	strophe.min.js	173B	16ms	
http-bind		POST	200 OK	text/...	strophe.min.js	527B	17ms	
http-bind		POST	200 OK	text/...	strophe.min.js	334B	7ms	
http-bind		POST	200 OK	text/...	strophe.min.js	398B	18ms	
http-bind		POST	200 OK	text/...	strophe.min.js	205B	8ms	
http-bind		POST	200 OK	text/...	strophe.min.js	375B	15ms	
http-bind		POST	200 OK	text/...	strophe.min.js	182B	5ms	
http-bind		POST	200 OK	text/...	strophe.min.js	741B	27ms	
http-bind		POST	200 OK	text/...	strophe.min.js	548B	17ms	
http-bind		POST	200 OK	text/...	strophe.min.js	249B	1.0min	
http-bind		POST	200 OK	text/...	strophe.min.js	57B	1.0min	
http-bind		POST	(pen...	Pen...	strophe.min.js	13B	0B	

20 requests | 3.90KB transferred | 4.5min (onload: 97ms, DOMContentLoaded: 97ms)

Gambar 12. Pengujian Jaringan Aplikasi Romeo

4 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan maka dapat diambil beberapa kesimpulan diantaranya sebagai berikut:

1. Penggunaan teknik *Bi-directional Stream Over Synchronous* HTTP meningkatkan respon dan kinerja jaringan dimana aplikasi *web* bekerja tanpa melakukan *refreshing* halaman.
2. Kinerja aplikasi jaringan berbasis *web* dapat ditingkatkan menggunakan BOSH dengan memanfaatkan Strophe.js untuk berhubungan dengan XMPP *server* yang melayani koneksi antar aplikasi.
3. *Stanza XML* direkayasa dengan menggunakan utilitas *Strophe Builder* yang tersedia pada *library* Strophe.js. *Stanza XML* direkayasa dengan mengacu kepada protokol permainan yang telah didefinisikan.
4. Pendekatan dengan penggunaan permainan ular tangga dapat membantu

anak-anak menghafal hari-hari besar nasional.

DAFTAR PUSTAKA

Alamsyah, Andry (2003). *Pengantar JavaScript*. From <http://ike.dinus.ac.id/umum/andry-javascript.php>, 5 Maret 2012

Fielding, Roy, dkk (1999). *Hypertext Transfer Protocol – HTTP/1.1*. From <http://www.ietf.org/rfc/rfc2616.txt>, 20 Maret 2011

Flanagan, David (2001). *JavaScript: The Definitive Guide, 4th Edition*. United States of America: O'Reilly

Flanagan, David (2011). *jQuery Pocket Reference*. United States of America: O'Reilly

Harold, Elliotte Rusty (2004). *XML in a Nutshell*. United States of America: O'Reilly

Loreto, S., dkk (2011). *Known Issues and Best Practices for the Use of Long Polling and Streaming in Bidirectional HTTP*. From <http://www.rfc-editor.org/rfc/rfc6202.txt>, 20 Maret 2012

McFarland, David Sawyer (2011). *JavaScript & jQuery: The Missing Manual*. United States of America: Pogue Press

Moffitt, Jack (2010). *Professional XMPP Programming with Javascript and jQuery*. Indianapolis: Wiley

Paterson, Ian, dkk (2010). *XEP-0124: Bidirectional-streams Over Synchronous HTTP (BOSH)*. From <http://xmpp.org/extensions/xep-0124.html>, 20 Maret 2012

Ray, Erik T (2003). *Learning XML, Second edition*. United States of America: O'Reilly

Saint-Andre, Peter, dkk (2009). *XMPP: The Definitive Guide*, United States of America: Oreilly

Saint-Andre, Peter (2011). *Extensible Messaging and Presence Protocol (XMPP): Core*. From <http://xmpp.org/rfcs/rfc6120.html>, 20 Maret 2012

Saint-Andre, Peter, Ed (2004). *Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence*. From <http://xmpp.org/rfcs/rfc6121.html>, 20 Maret 2011

Saint-Andre, Peter (2011). *Extensible Messaging and Presence Protocol (XMPP): Address Format*. From <http://xmpp.org/rfcs/rfc6122.html>, 20 Maret 2012

Sikos, Leslie (2011). *Web Standards: Mastering HTML5, CSS3, and XML*. United States of America: Apress

Saint-Andre, Peter (2007). *A Uniform Resource Name (URN) Namespace for Extension to the Extensible Messaging and Presence Protocol (XMPP)*. From <http://www.rfc-editor.org/rfc/rfc4854.txt>, 23 Maret 2012

Thomas, Stephen A (2001). *HTTP Essentials: Protocols for Secure, Scalable Web Sites*. United States of America: Wiley

Walmsley, Priscilla (2001). *Definitive XML Schema*. United States of America: Prentice Hall

Wright, William (2003). *Jabber Developer's Handbook*, United States of America: Sams

York, Richard (2009). *Beginning JavaScript and CSS Development with jQuery*. United States of America: Wrox