

Implementasi *Socket Programming* Pada Aplikasi *Chat Uloen Messenger* Berbasis *Android*

Reza Riski¹, Husaini^{2*}, Muhammad Nasir³

^{1,2,3} Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe
Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA

¹rezariskitrkj@gmail.com

^{2*} husaini@pnl.ac.id

³muhnasir.tmj@pnl.ac.id

Abstrak— Dalam era perkembangan Teknologi yang pesat, Aplikasi Chatting telah menjadi salah satu cara Komunikasi yang paling populer di kalangan pengguna Internet, Salah satu Aplikasi Komunikasi yang terdapat pada layanan Internet adalah Chatting. Aplikasi ini memungkinkan dua orang pengguna atau lebih untuk berinteraksi secara online, Aplikasi Chatting dibuat dengan cara mengimplementasikan teknologi Socket Programming (Socket.io). Teknologi Socket Programming Socket.io dapat digunakan untuk membangun aplikasi chatting berbasis real-time di Android. Berdasarkan latar belakang tersebut dibuatlah Aplikasi Chatting yang Bernama Aplikasi Chat-Uloen Messenger dengan cara Meng-Implementasikan Socket Programming (Socket.io) Pada Aplikasi ini dilakukan pengujian web service untuk melihat nilai throughput dari performa server pada aplikasi dengan cara membandingkan dengan menggunakan socket.io dan tanpa menggunakan socket.io. Pengujian sistem menggunakan metode blackbox untuk mengetahui tingkat keberhasilan aplikasi dan kepuasan pengguna. Hasil dari pengujian web service yang didapatkan untuk nilai throughput dengan menggunakan socket.io mendapat hasil cukup bagus dengan nilai rata rata mencapai 447 kbps sedangkan tanpa menggunakan socket.io mencapai 47 kbps, sedangkan pengujian blackbox mendapatkan 97,5% keberhasilan aplikasi.

Kata kunci— Socket Programming, Aplikasi Chat Uloen Messenger, Real-time

Abstract - In the era of rapid technological development, chat applications have become one of the most popular ways of communication among Internet users, One of the Communication Applications found on Internet services is Chatting. This application allows two or more users to interact online, Chat Application is made by implementing Socket Programming technology (Socket.io). Socket Programming Socket.io technology can be used to build real-time based chat applications on Android. Based on this background, a Chat Application named Chat-Uloen Messenger Application was created by Implementing Socket Programming (Socket.io) In this application, web service testing was carried out to see the throughput value of server performance in the application by comparing using socket.io and without using socket.io. System testing uses the blackbox method to determine the level of application success and user satisfaction. The results of web service testing obtained for throughput values using socket.io got pretty good results with an average value of 447 kbps while without using socket.io reached 47 kbps, while blackbox testing got 97.5% application success.

Keywords - Socket Programming, Chat Uloen Messenger Application, Real-time

I. PENDAHULUAN

Dalam era perkembangan teknologi yang pesat, aplikasi chatting telah menjadi salah satu cara komunikasi yang paling populer di kalangan pengguna internet. Oleh karena itu Internet banyak diminati karena harganya yang murah dan mudah digunakan. Salah satu aplikasi komunikasi yang terdapat pada layanan internet adalah Chatting. Dengan chatting masyarakat dapat berbicara dengan siapa saja dan di mana saja dalam satu waktu. Kegiatan chatting menyenangkan karena dapat berkomunikasi dengan teman yang sudah terhubung secara langsung baik pribadi maupun dalam forum[1]. Aplikasi ini memungkinkan dua orang atau lebih untuk berinteraksi secara online, dengan kemampuan tersebut pengguna dapat mengirimkan pesan dalam berbagai format, seperti teks, gambar, multimedia lainnya. Beberapa contoh aplikasi chatting yang kini populer dan digunakan oleh masyarakat adalah, Whatsapp, Telegram, dan WeChat. Aplikasi-aplikasi ini menawarkan pengalaman berkomunikasi yang mudah dan nyaman, serta memungkinkan pengguna

untuk terhubung dan berinteraksi dengan orang lain dari berbagai tempat di seluruh dunia.

Berdasarkan latar belakang di atas dibuatlah aplikasi Chatting yang Bernama Aplikasi Chat-Uloen Messenger. Dengan adanya aplikasi Chatting Uloen Messenger ini Peneliti berharap Aplikasi ini dapat digunakan oleh Mahasiswa TRKJ Khususnya Angkatan 2019 sehingga terjalin komunikasi antar Angkatan. Dalam pelaksanaannya, Aplikasi chat Uloen messenger ini dibuat dengan cara meng-implementasikan teknologi Socket Programming (Socket.io). Socket.IO terdiri dari dua bagian, yang pertama adalah client-side yang berjalan pada browser dan yang kedua adalah server-side yang digunakan sebagai modul untuk node.js[2].

II. METODELOGI PENELITIAN

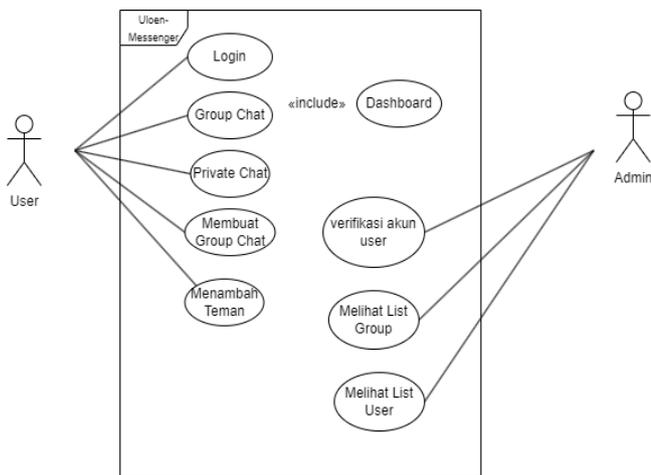
A. Perancangan Sistem

Pada perancangan sistem ini menggunakan diagram UML UML merupakan Bahasa standar yang digunakan untuk mendokumentasikan, memvisualisasikan, dan membangun sistem perangkat lunak dengan menggunakan symbol

diagram[3]. UML terdiri dari yaitu *use case diagram*, *workflow diagram*, *perancangan user interface* dan *blok diagram sistem*.

1. Rancangan Use Case Diagram

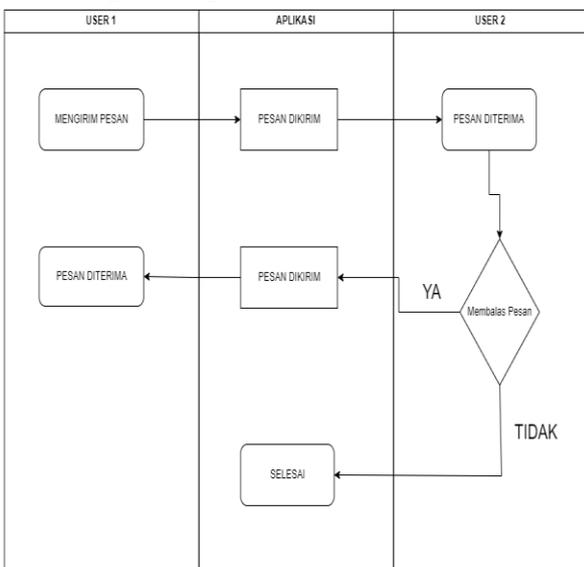
Tahapan use case diagram memiliki 2 aktor yaitu admin dan user. Adapun use case diagram aplikasi chat uloen messenger dapat dilihat pada gambar 1 berikut.



Gambar 1 Use case diagram aplikasi

2. Rancangan Workflow Diagram

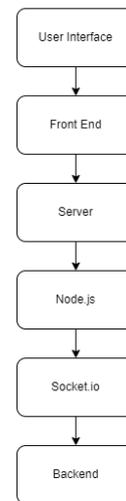
Workflow diagram dirancang untuk membantu menggambarkan alur kerja aplikasi chat uloen messenger secara visual, berikut *workflow diagram* dari aplikasi chat uloen messenger pada gambar 2 berikut.



Gambar 2 Workflow Diagram Aplikasi

3. Blok Diagram Sistem

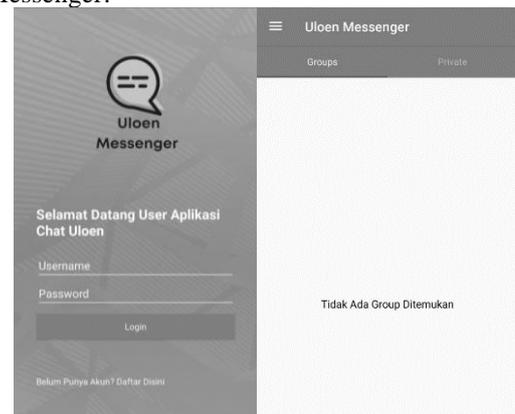
Blok diagram sistem digunakan untuk menjelaskan gambaran dari sistem yang akan dibangun[4]. Berikut merupakan perancangan blok diagram kerja sistem dapat dilihat pada gambar pada Gambar 3.



Gambar 3 Blok Diagram Aplikasi

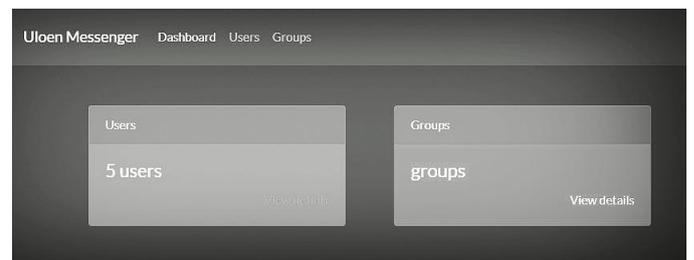
4. Perancangan User Interface

User Interface dirancang untuk sesuai kebutuhan sistem berdasarkan dari rancangan *use case diagram*. Gambar 4 merupakan tampilan rancangan *login* dan *home* aplikasi Chat Uloen Messenger.



Gambar 4 Perancangan user interface login dan home

Pada halaman web admin berfungsi untuk menambahkan daftar tempat wisata dan memasukkan informasi mengenai tempat wisata seperti pada Gambar 5 berikut.



Gambar 5 Perancangan halaman web admin

B. Teknik Pengujian

Tahapan teknik pengujian pada aplikasi Chat Uloen Messenger terdiri dari pengujian *web service*, dan pengujian *blackbox*.

1. Pengujian *Web Service*

Pada pengujian web service ini akan dilakukan Pengujian web service menggunakan Jmeter Apache. Pengujian dilakukan dengan membandingkan 2 pengujian yaitu, Pengujian web service menggunakan socket.io dan pengujian web service tanpa menggunakan socket.io. Yang kemudian hasil dari pengujian menggunakan aplikasi Jmeter Apache digunakan untuk memperoleh hasil throughput yang mana yang lebih baik antara menggunakan socket.io atau tanpa menggunakan socket.io.

Throughput merupakan jumlah permintaan yang berhasil diolah dalam satu periode waktu (detik, menit, jam) oleh server. Waktu ini dihitung mulai dari saat sampel pertama berjalan hingga sampel terakhir. Semakin tinggi nilai throughput, semakin optimal kinerja dari server tersebut[5].

Adapun standar *Throughput* menurut TIPHON adalah seperti pada Tabel 1 berikut:

TABEL I
PARAMETER *THROUGHPUT*

Kategori Throughput	Latency	Indeks
Bad	0 – 338 kbps	1
Poor	338 - 700 kbps	2
Fiar	700 – 1200 kbps	3
Good	1200 kbps – 2,1 Mbps	4
Excellent	>2,1 Mbps	5

2. Pengujian *Blackbox*

Black box testing atau dapat disebut juga *Behavioral Testing* adalah pengujian yang dilakukan untuk mengamati hasil *input* dan *output* dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. *Black-Box Testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program[6].

Pengujian yang dilakukan harus dirancang dengan baik agar dapat menemukan kesalahan secara sistematis dan dapat diperbaiki dalam waktu yang singkat. Gambar 6 menjelaskan alur kerja dari metode Blackbox Testing.



Gambar 6 Alur metode *blackbox testing*

Pengujian *blackbox* untuk menguji kelayakan aplikasi, mengecek terjadinya error dengan menggunakan teknik *Equivalence Partitioning* yang di bagi beberapa domain dengan 50 pengguna[7]. Pada table 2 merupakan salah satu contoh pengujian metode *blackbox* yang terdapat pada halaman register.

TABEL II
PENGUJIAN *BLACKBOX*

ID	Rincian Pengujian	Hasil Yang Diharapkan	Pengguna/User
			1
	Mengisi nama, <i>username</i> dan password untuk registrasi akun	Daftar akun berhasil <i>user</i> di arahkan ke halaman utama aplikasi	2 3 4 5 ...
R			50

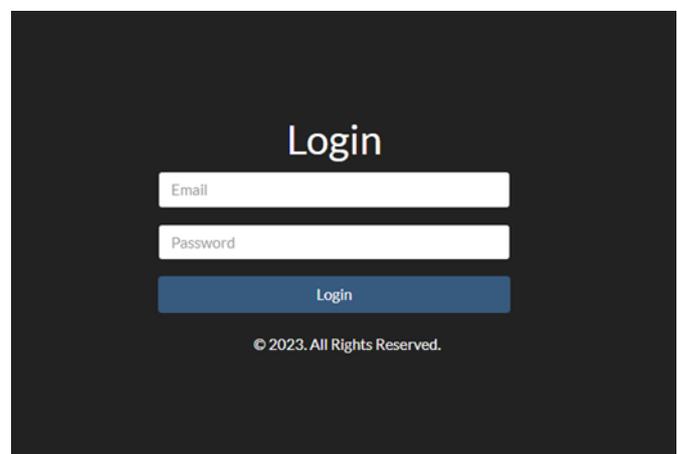
III. HASIL DAN PEMBAHASAN

A. Implementasi *User Interface*

User Interface dibuat sesuai dengan perancangan user interface yang telah didefinisikan sebelumnya. Bagian ini dijelaskan fungsi dari setiap halaman yang telah dibuat yang berkaitan dengan pengelolaan data pada aplikasi Chat Uloen Messenger. Hasil implementasi UI dari aplikasi Chat Uloen Messenger terdiri dari web dan mobile.

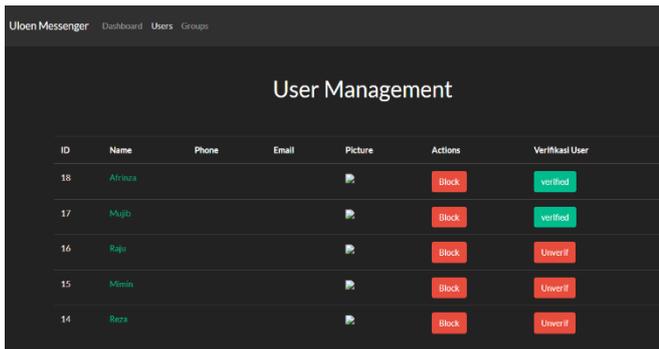
1. Tampilan Website

Halaman *login* mengharuskan pengguna untuk melakukan autentifikasi sebelum masuk pada aplikasi web ataupun mobile. Berikut pada gambar 7 merupakan tampilan halaman login pada web Chat Uloen Messenger.



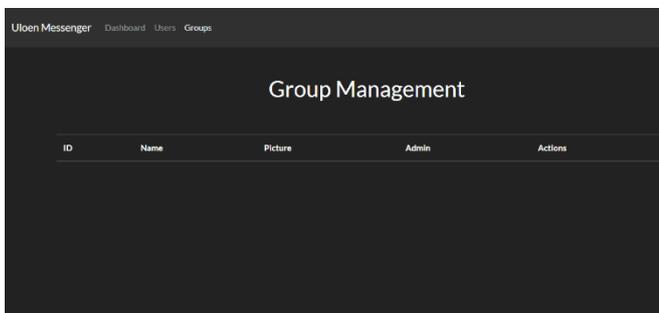
Gambar 7 Halaman login web

Admin dapat melihat daftar users dan daftar group serta dapat melakukan verifikasi akun register yang dilakukan oleh user sebelum melakukan login pada halaman user management.



Gambar 10 Halaman *user management*

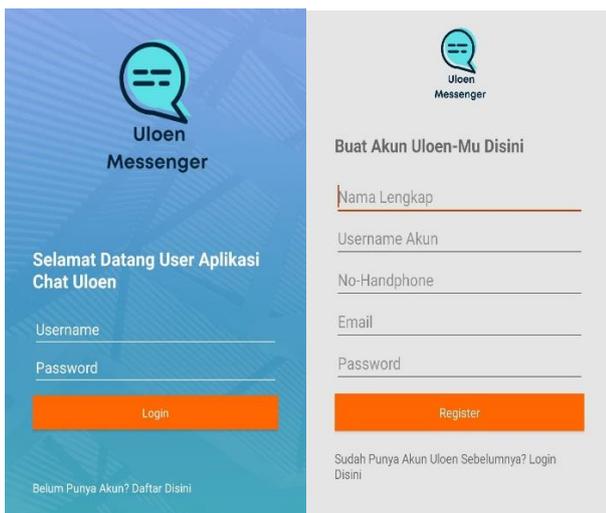
Halaman group management, admin dapat melihat dan melakukan block kepada user yang ada didalam group pada aplikasi. Berikut merupakan tampilan pada halaman transaksi pada gambar 11.



Gambar 11 Halaman *group management*

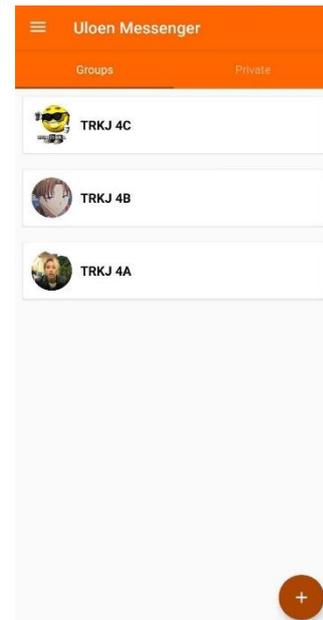
2. Tampilan Mobile

Halaman login aplikasi merupakan halaman awal yang dimana pengguna diwajibkan memasukkan email dan kata sandi yang telah mereka buat saat melakukan *register* atau pendaftaran akun melalui halaman *register*.



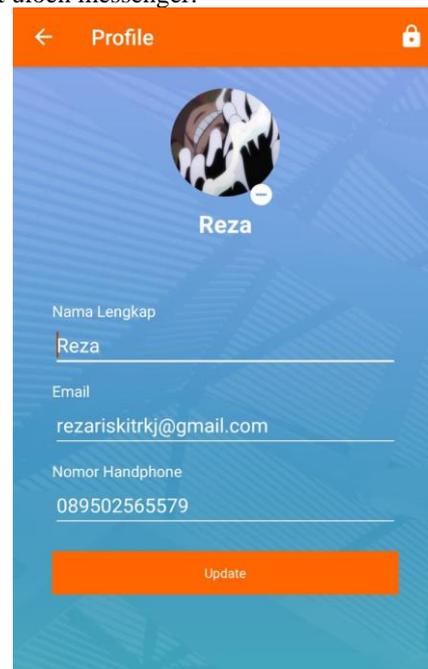
Gambar 12 Halaman *login dan register*

Halaman beranda aplikasi merupakan atau halaman *home* yang ditampilkan setelah pengguna berhasil melakukan login atau registrasi. Gambar 13 menampilkan tampilan halaman utama dari aplikasi *chat uloen messenger*.



Gambar 13 Halaman utama aplikasi

Halaman *user profile* menampilkan menu edit profile. Gambar 14 menampilkan tampilan halaman edite profile pada aplikasi chat uloen messenger.



Gambar 14 Halaman profile

B. Hasil Pengujian *Web Service*

Pengujian *web service* bertujuan untuk memperoleh hasil throughput yang mana yang lebih baik antara menggunakan socket.io atau tanpa menggunakan socket.io.

1. Pengujian Menggunakan *Apache JMeter*

Pada pengujian web service ini akan dilakukan Pengujian web service menggunakan Jmeter Apache. Pengujian dilakukan dengan membandingkan 2 pengujian yaitu, Pengujian web service menggunakan socket.io dan pengujian web service tanpa menggunakan socket.io.

a. *Traffic* dengan *looping* 1 menggunakan *socket.io*

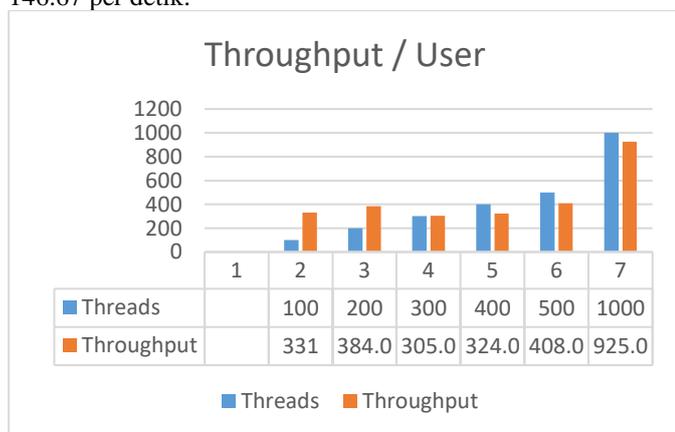
Pengujian dilakukan dengan skenario jumlah samples 100, 200, 300, 400, 500, dan 1000. *Samples* ini akan menjadi indikator jumlah pengguna virtual (*virtual users*) per *request*.

TABEL VI

HASIL PENGUJIAN TRAFFIC DENGAN LOOPING 1

Samples	Average (ms)	Max	Error %	Throughput	Received KB/sec	Sent KB/sec
100	300	488	0%	331.6	98.46	62.84
200	282	222	0%	384.6	114.09	72.87
300	192	136	0%	305.8	90.79	115.87
400	208	1058	0%	324.1	96.23	122.82
500	240	1052	0%	408.5	121.27	154.78
1000	292	460	0%	925.9	274.88	350.84
Rata-rata	252.3333	569.3333	0%	446.8	132.6	146.67

Berdasarkan Tabel 6 Berdasarkan tabel 4.1 hasil pengujian dapat diketahui bahwa dari semua thread yang dijalankan dalam waktu kurun waktu 1 detik (ramp-up), dibutuhkan waktu rata-rata eksekusi sebesar 252.3333 milliseconds atau 25,2 seconds dengan waktu minimum yang dibutuhkan untuk mengeksekusi thread yaitu 0 milliseconds dan waktu maksimumnya 569.3333 milliseconds atau 57 seconds tanpa mengalami error. Nilai rata-rata throughput menunjukkan bahwa server mampu mengeksekusi 446.8 thread per time unit dan jumlah data yang berhasil diunduh oleh server selama dilakukannya eksekusi pengujian performance adalah 132.6 per-detik dan jumlah data yang berhasil dikirim dari server selama dilakukannya eksekusi pengujian performance adalah 146.67 per detik.



Gambar 19 Grafik nilai *throughput* per *user*

Gambar menunjukkan hasil pengujian throughput pada thread yang dijalankan mengalami beberapa penurunan, pada pengujian dengan thread 200 terdapat jumlah request yang diterima sebanyak 384 data per second, sedangkan pada

pengujian dengan user 300 terdapat jumlah request yang diterima sebanyak 305 data per second, hal tersebut dapat terjadi karena lamanya pada saat proses pengiriman data akibat dari bad gateway.

b. *Traffic* dengan *looping* 1 tanpa menggunakan *socket.io*

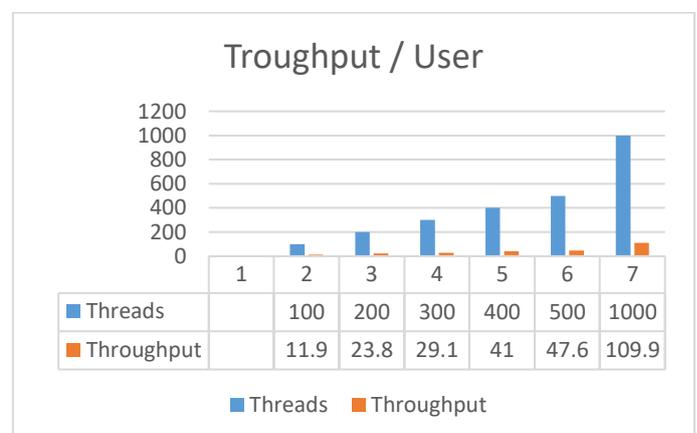
Pengujian dilakukan dengan skenario jumlah samples 100, 200, 300, 400, 500, dan 1000. *Samples* ini akan menjadi indikator jumlah pengguna virtual (*virtual users*) per *request*.

TABEL VII

HASIL PENGUJIAN TRAFFIC DENGAN THREADS 1

Samples	Average (ms)	Max	Error %	Throughput	Received KB/sec	Sent KB/sec
100	2247	8133	100%	11.9	32.47	0
200	2211	8137	100%	23.8	65.01	0
300	2275	9630	100%	29.1	79.37	0
400	2241	9636	100%	41	111.90	0
500	2270	9652	100%	47.6	129.93	0
1000	2271	8138	100%	109.9	300.30	0
Rata-rata	2252.5	8887.667	100%	43.9	119.8	0

Berdasarkan hasil pengujian pada Tabel 7 hasil pengujian dapat diketahui bahwa dari semua thread yang dijalankan dalam waktu kurun waktu 1 detik (ramp-up), dibutuhkan waktu rata-rata eksekusi sebesar 2252.5 milliseconds atau 22.5 seconds dengan waktu minimum yang dibutuhkan untuk mengeksekusi thread yaitu 0 milliseconds dan waktu maksimumnya 8887.667 milliseconds atau 88.8 seconds dengan mengalami error 100%. Nilai rata-rata throughput menunjukkan bahwa server hanya mampu mengeksekusi 43.9 thread per time unit dan jumlah data yang berhasil diunduh oleh server selama dilakukannya eksekusi pengujian performance adalah 119.8 per-detik dan jumlah data yang berhasil dikirim dari server selama dilakukannya eksekusi pengujian performance adalah 0 per detik.



Gambar 20 Grafik nilai *throughput* per *looping*

Gambar 20 menunjukkan hasil pengujian throughput pada threads yang dijalankan mengalami kenaikan, terlihat pada

pengujian awal dengan looping 100 terdapat jumlah request yang diterima sebanyak 11.9 data per seconds, sedangkan pada pengujian selanjutnya jumlah request yang diterima semakin banyak hingga pada looping 1000 hanya terdapat 109.9 data per second, hal tersebut dapat terjadi karena semakin banyak throughput yang dihasilkan maka akan semakin bertambahnya nilai throughpath yang diperoleh.

C. Hasil Pengujian *Blackbox Testing*

Pengujian *blackbox* adalah metode pengujian yang digunakan untuk menguji fungsi-fungsi sistem atau kekurangan dalam perangkat lunak yang sedang diuji. Tujuannya adalah untuk memperbaiki perangkat lunak dan meminimalkan kemungkinan adanya kekurangan dalam sistem tersebut.

TABEL XII
PENGUJIAN BLACKBOX

Test Case ID	Pengujian	Input	Hasil Yang Diharapkan
R01	Register Akun	Klik Daftar Akun	Daftar akun berhasil dan <i>user</i> diarahkan ke halaman utama
L01	LogIn Akun	Klik <i>LogIn</i>	<i>LogIn</i> berhasil dan <i>user</i> diarahkan ke halaman utama
H01	Halaman Utama	Membuat Group Chat dan Menambahkan Friendlist	Berhasil membuat group chat dan berhasil mengirim permintaan untuk berteman
P01	Halaman Profil	Klik Edit Profil	Edit profil berhasil dan diarahkan ke halaman profil
TU01	Halaman Friend Request	Menambah teman dengan cara mengkonfirmasi pada fitur friend request dan dapat melihat daftar teman	Berhasil mengkonfirmasi daftar permintaan teman dan dapat melihat daftar teman yang telah diajukan untuk berteman
BT01	Halaman Group Request	Dapat melakukan konfirmasi kepada user yang ingin memasuki group chat	Berhasil melakukan konfirmasi pada user yang ingin memasuki group chat

TQR01	Halaman Group Chat	Dapat mengirim pesan di group chat	Berhasil mengirim pesan di group chat
TQR02	Halaman Private Chat	Dapat mengirim pesan di private chat	Berhasil mengirim pesan di private chat

Pengujian *Blackbox* dilakukan dengan metode *Equivalence Partitioning* dengan jumlah user sebanyak 50 orang. Tabel 12 berikut merupakan hasil dari pengujian *Blackbox* yang telah dilakukan.

TABEL XIII
HASIL PENGUJIAN BLACKBOX

Test Case ID	Pengguna Berhasil	Pengguna Tidak Berhasil	Persentase	Hasil
R01	49	1	49/50× 100%	98%
L01	50	0	50/50× 100%	100%
H01	49	1	49/50× 100%	98%
P01	49	1	49/50× 100%	98%
TU01	48	2	48/50× 100%	96%
BT01	48	2	48/50× 100%	96%
TQR01	49	1	49/50× 100%	98%
TQR02	48	2	48/50× 100%	96%
Rata-rata				97,5%

Berdasarkan Tabel 13 disajikan hasil pengujian kelayakan aplikasi menggunakan metode *blackbox* dengan melibatkan 50 pengguna. Dalam pengujian ini, 97,5% pengguna berhasil menggunakan aplikasi, sementara sisanya mengalami kegagalan. Kegagalan tersebut dapat disebabkan oleh gangguan jaringan atau kesalahan input dari pengguna.

IV. SIMPULAN

Berdasarkan penelitian dan implementasi yang telah dilakukan, diperoleh kesimpulan pada aplikasi chat uloen messenger dengan implementasi socket programming dan pengujian web service sebagai berikut:

1. Hasil dari pengujian pada webservice menggunakan aplikasi apache jmeter menghasilkan nilai throughput yang didapatkan dalam dua kondisi yaitu menggunakan socket.io atau tanpa menggunakan socket.io. Hasil pengujian menggunakan socket.io mendapati nilai throughput sebesar 447 kbps sedangkan tanpa menggunakan socket.io 47 kbps.

2. Hasil dari pengujian aplikasi menggunakan metode blackbox testing didapati 97,5% pengguna berhasil menjalankan aplikasi, dan berhasil menjalankan berbagai fitur yang ada pada aplikasi chat uloen messenger.

REFERENSI

- [1] S. Sutikno, I. F. Astuti, and D. M. Khairina, "Membangun Aplikasi Chatting Untuk Media Perkenalan Berbasis Web," *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 13, no. 1, p. 1, 2018, doi: 10.30872/jim.v13i1.1003.
- [2] R. Maulana, "Implementasi Web Socket Pada Sistem Pelayanan Pasien Rawat Jalan Pada Puskesmas Kabupaten Gowa," *J. INSTEK (Informatika Sains dan Teknol.*, vol. 6, no. 1, p. 130, 2021, doi: 10.24252/instek.v6i1.20555.
- [3] C. B. Santoso, "Analisis dan Perancangan Sistem Informasi Produksi," *Teknois J. Ilm. Teknol. Inf. dan Sains*, vol. 6, no. 2, pp. 73–83, 2019, doi: 10.36350/jbs.v6i2.42.
- [4] Y. Zhangling and D. Mao, "A Real-Time Group Communication Architecture Based on WebSocket," *Int. J. Comput. Commun. Eng.*, vol. 1, no. 4, pp. 408–411, 2012, doi: 10.7763/ijcce.2012.v1.100.
- [5] A. wijaya. R. D. fakultas ilmu Komputer, "281612-Analisa-Kehandalan-Jaringan-Internet-Den-5Dd258Af," vol. 2, pp. 1–10, 2010.
- [6] T. Hidayat and M. Muttaqin, "Pengujian sistem informasi pendaftaran dan pembayaran wisuda online menggunakan black box testing dengan metode equivalence partitioning dan boundary value analysis," *J. Tek. Inform. UNIS*, vol. 6, no. 1, pp. 2252–5351, 2018, [Online]. Available: www.ccssenet.org/cis
- [7] D. B. Muslimin, D. Kusmanto, K. F. Amilia, M. S. Ariffin, S. Mardiana, and Y. Yulianti, "Pengujian Black Box pada Aplikasi Sistem Informasi Akademik Menggunakan Teknik Equivalence Partitioning," *J. Inform. Univ. Pamulang*, vol. 5, no. 1, p. 19, 2020, doi: 10.32493/informatika.v5i1.3778.