BAB 2 LANDASAN KEPUSTAKAAN

Penelitian yang akan dilakukan adalah membandingkan kinerja antara protokol Websocket dengan protokol *server*-sent event pada teknologi *push notification* dengan menggunakan *delay* sebagai parameter untuk pengujian dan analisa dalam penelitian ini. Untuk mendukung hasil perbandingan yang tepat dibutuhkan landasan pustaka pada penelitian ini. Landasan pustaka tersebut dikhususkan pada teknologi *push notification,* protokol Websocket dan protokol *server*-sent event. Pustaka pendukung lainnya adalah mengenai teknologi *push notification*, bahasa pemrograman Python, Javascript.

2.1 Kajian Pustaka

Kajian pustaka dalam penelitian ini membahas beberapa perbandingan penelitian mengenai protokol Websocket dan SSE yang sebelumnya sudah dilakukan, penulis menggunakan pembahasan penelitian sebelumnya sebagai rujukan untuk mendukung penulisan ini. Rangkuman dari penelitian yang dijadikan rujukan terdapat pada tabel 2.1.

Pada penelitian sebelumnya yang berjudul "Pengembangan Push Notification Menggunakan Websocket" oleh Andrias Yudianto Putra. Penelitian ini menerapkan protokol Websocket untuk pengiriman notifikasi. Ruang lingkup yang digunakan adalah jaringan lokal. Perangkat yang digunakan adalah smartphone dengan sistem operasi dan laptop dengan sistem operasi windows dan linux. Pengujian yang dilakukan pada penelitian ini bertujuan untuk mendapatkan kinerja pengiriman Websocket dengan menggunakan parameter delay. Kesimpulan dari penelitian ini adalah terdapat delay yang signifikan antara perangkat smartphone dan laptop pada pengiriman notifikasi dengan menggunakan protokol Websocket (Putra, 2016).

Pada penelitian sebelumnya yang berjudul "Mobile HTML5: Efficiency and Performance of Websockets and Server-Sent Events" oleh Eliot Estep. Penelitian ini menerapkan protokol Websocket dan SSE. Dari kedua protokol yang diteliti, penulis mengacu pada penelitian protokol SSE yang diterapkan pada browser perangkat smartphone. Tujuan dari penelitian ini untuk mengetahui performa EventSource dan memperhitungkan dampak dari performa EventSource pada beberapa perangkat smartphone dengan menggunakan beberapa browser yang berbeda. Kesimpulan dari penelitian ini adalah kinerja yang tidak seragam dan sangat bervariasi tergantung pada browser dan konfigurasi pada jaringan (Estep, 2013).

Pada penelitian sebelumnya yang berjudul "Internet Based Communication using Server Push" oleh Pragya. Penelitian ini menggunakan tiga teknologi *push* yaitu long-polling, server-sent event dan Websocket. Dari tiga teknologi yang

digunakan, penulis mengacu pada penelitian protokol SSE dan Websocket sebagai pembanding dari penelitian yang akan dilakukan. Tujuan dari penelitian ini adalah untuk membandingkan kinerja pengiriman notifikasi dari tiga teknologi *push* yang sudah disebut sebelumnya. Hasil dari penelitian yang dilakukan menyebutkan bahwa pengiriman yang dilakukan oleh SSE dapat juga dilakukan oleh Websocket dengan performa yang berbeda. Pada beberapa aspek terdapat perbedaan antara protokol SSE dan Websocket seperti tingkat kesulitan penerapan, *delay* dan *server-loading* . Kesimpulan yang didapat adalah protokol Websocket lebih baik dari long-polling dan SSE karena protokol Websocket dapat melakukan komunikasi dua arah dan memiliki *delay* yang lebih kecil dibandingkan long-polling dan SSE(Pragya, 2016).

Tabel 2.1 Tabel Kajian Pustaka

No.	Judul	Object (Input)	Metode	Hasil	Perbedaan Penelitian
1.	Pengembangan Push Notification Menggunakan Websocket (Andrias Yudianto Putra, 2016)	Object : Smartphone android dan laptop yang memiliki sistem operasi windows dan linux *Input* : Judul berita, *link* berita dan target pengiriman	1. Menentukan target pengiriman notifikasi (semua client dan grup client) 2. Judul dan *link* berita serta target pengiriman dimasukan ke database 3. Judul dan *link* berita dikirimkan ke target pengiriman (client) 4. Judul dan *link* berita diterima oleh client dan ditampilkan pada fitur notifikasi	Judul dan *link* berita dapat dilihat pada fitur notifikasi dan *log* pada aplikasi client	1. Tidak melakukan penelitian dengan menggunakan protokol SSE 2. Tidak menggunakan parameter penggunaan CPU
2.	Mobile HTML5: Efficiency and Performance of Websockets and Server-Sent Events (Elliot Estep, 2013)	Object: Smartphone dan Browser *Input*: Nomor update dan waktu update	1. Menjalankan aplikasi web browser pada perangkat smartphone 2. Perangkat smartphone	Short event interval dan long event interval	Pengujian dengan menggunakan parameter penggunaan CPU dilakukan pada sisi *client*. Sedangkan pada penelitian ini dilakukan

			dibiarkan dalam keadaan idle 3. Setelah beberapa waktu perangkat smartphone kembali dinyalakan 4. Update number yang telah dikirim akan diperiksa termasuk ke dalam kategori berhasil atau gagal		pada sisi *server*.
3.	Internet Based Communication using Server Push (Pragya, 2016)	Object: Browser *Input*: Notifikasi berupa teks	• SSE 1. Pesan notifikasi diinputkan ke dalam field 2. Server akan mengecek kondisi terakhir (sebelum pesan notifikasi terbaru dimasukkan) 3. Server akan memberikan respon	Informasi penambahan notifikasi baru sebelum dikirim ke client	Notifikasi yang dikirimkan diberi interval waktu tiap 4 detik. Sedangkan pada penelitian ini tidak diberi interval waktu.

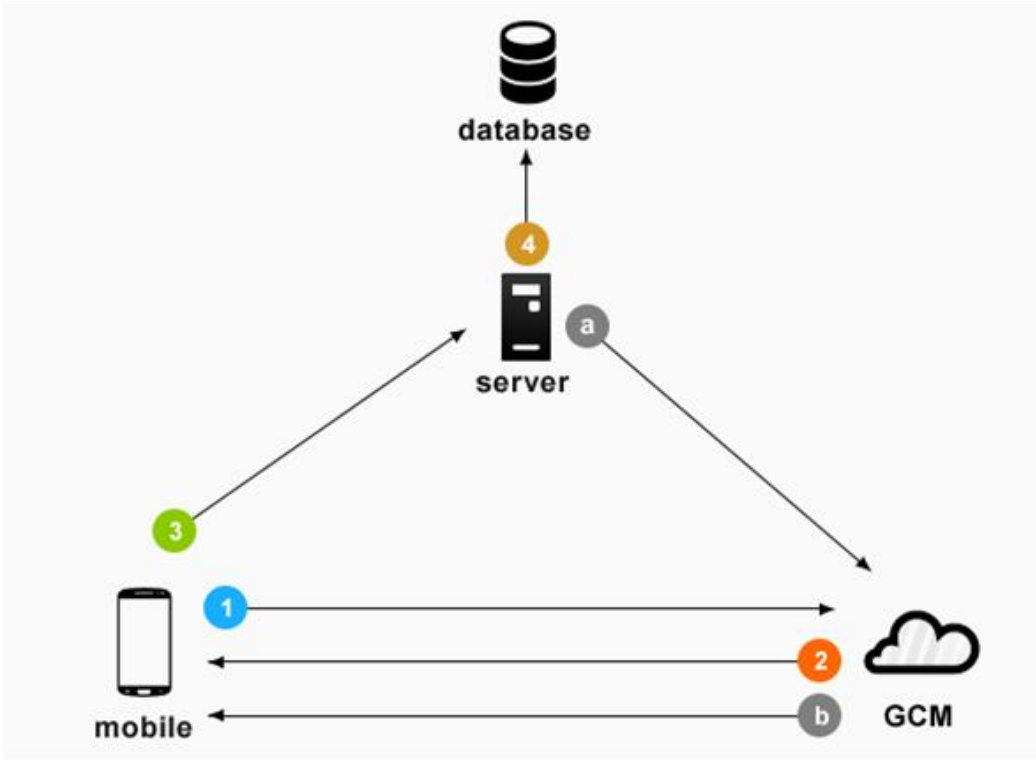
			• Websocket 1. *Client* membuka koneksi Websocket 2. Notifikasi yang dikirim berisi *recipient user* dan *action id* 3. Notifikasi dikirim secara *broadcast* pada semua koneksi yang terbuka 4. Notifikasi dikirimkan pada *client* yang aktif pada saat itu	Pesan notifikasi berupa teks	

Sumber: (Putra, 2016) (Estep, 2013) (Pragya, 2016)

2.2 Push Notification

Push notification adalah teknologi berbasis standar teknis yang berfungsi untuk mengurangi kelebihan informasi dengan mengirimkan informasi user melalui internet secara teratur (Guo & Liu, 2013). Pada perangkat *smartphone*, *push notification* adalah pesan dari *server* sebagai penyedia informasi yang muncul pada perangkat yang mana *client* tidak perlu berada pada aplikasi yang mengirimkan notifikasi atau sedang dalam menggunakan perangkat tersebut (Urbanairship.com, 2017).

Tiap smartphone memiliki sistem operasi yang berbeda dan masing – masing sistem operasi memiliki teknologi *push notification* yang berbeda. Pada penelitian ini sistem operasi pada smartphone yang digunakan adalah operasi sistem berbasis android karena sistem operasi Android lebih sederhana dan memiliki *hardware* yang dapat bersaing. Teknologi *push notification* pada *smartphone* yang berbasis android diatur oleh layanan Google Cloud Messaging (GCM) (Brüstel, 2012). GCM berfungsi sebagai *broker*. Untuk mendapatkan layanan *push notification* dari GCM, *client* harus terlebih dahulu terdaftar ke *server* untuk mendapatkan tanda pengenal (Ding, J, et al., 2014). Tanda pengenal (*token*) diperlukan agar notifikasi yang dikirim oleh *server* tepat terkirim pada *client* yang mempunyai tanda pengenal (token) tersebut.



Gambar 2.1 Mekanisme kerja*push notification* menggunakan layanan GCM

Sumber: (Webgeometrics.com)

Gambar 2.1 menjelaskan teknologi *push notification* dengan menggunakan layanan GCM sebagai *broker* pada smartphone berbasis android. Sistem kerja dari *push notification* adalah sebagai berikut:

1. *Client* mendaftar pada GCM sebagai *broker*
2. Setelah terdaftar *broker* memberikan tanda pengenal (*token*) kepada *client*. *Token* berfungsi agar notifikasi yang dikirim tepat kepada pemilik *token*
3. Setelah *token* didapatkan dari *broker*, *client* akan mengirimkan token ke *server* aplikasi
4. *Token* yang didapat dari *client* akan disimpan ke *database*

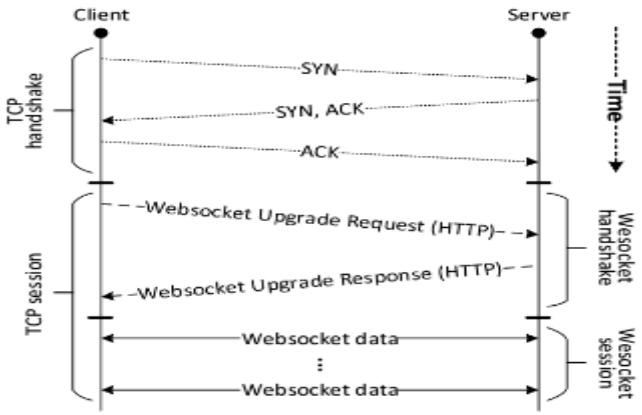
Mekanisme pengiriman notifikasi dari *server* ke *client* adalah:

1. *Server* mengirimkan notifikasi kepada *broker*
2. *Broker* menerima notifikasi dari *server* dan dikirimkan ke *client* sesuai dengan token yang disimpan di database *server* aplikasi.

2.3 Protokol Websocket

Websocket adalah protokol yang dikembangkan oleh HTML5 yang menerapkan komunikasi *full – duplex* antara *server* dengan *client* (Zhang & Shen, 2013). Protokol Websocket dirancang untuk diterapkan di *web browser* dan *webserver*, tetapi dapat digunakan oleh aplikasi *client* atau *server*. Protokol Websocket sendiri berbasis pada *Transfer Control Protocol* (TCP).

Pada dasarnya Websocket dirancang untuk menyerupai TCP kecuali pada segi fungsionalitas. Hal ini memungkinkan *server* Websocket untuk berbagi port dengan *server* HTTP, sehingga membuat Websocket mudah disebarkan dalam beberapa skala (Estep, 2013).



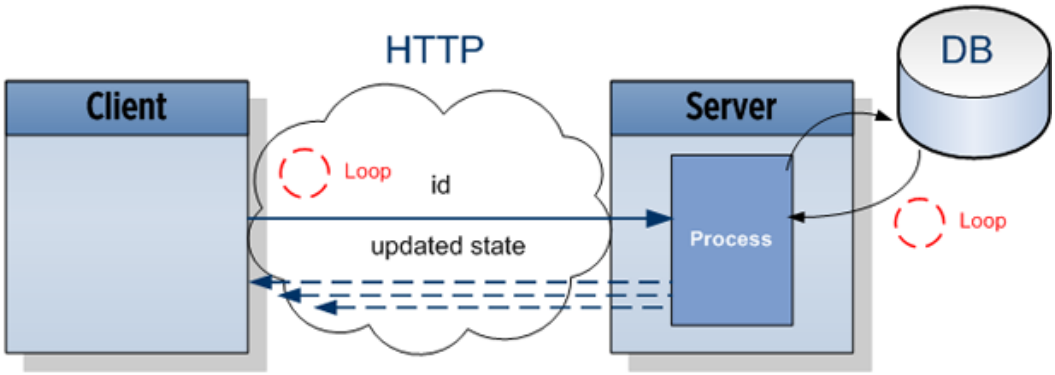
Gambar 2.2 Mekanisme kerja protokol Websocket

Sumber : (Skvorc, Horvat, & Srbljic, 2014)

Gambar 2.2 menunjukan proses *handshake* antara *client* dengan *server* pada protokol Websocket. Pertama *client* akan meminta koneksi Websocket pada *server*. Kemudian *server* akan merespon permintaan koneksi dari *client*. Setelah proses *handshake* terbentuk protokol Websocket dapat mengirim dan menerima data dalam model dual channel atau dalam jalur berbeda pada waktu yang sama (Zhang & Shen, 2013). *Server* akan secara aktif mengirim data berupa notifikasi kepada *client*. Proses pengiriman notifikasi akan terus berjalan hingga diberikan perintah untuk memutuskan koneksi pada sisi *server*. Pada penelitian ini, *push server* Websocket sekaligus menjadi *broker* sehingga pengiriman langsung dari *server* kepada *client*. Ketika koneksi *client* dengan *server* terputus, terdapat pemberitahuan pada sisi *server*.

2.4 Protokol Server-Sent Event (SSE)

Protokol server-sent event adalah protokol yang juga dikembangkan oleh HTML5. Berbeda dengan Websocket yang menerapkan komunikasi dua arah atau *full – duplex*, protokol SSE lebih memfokuskan pada komunikasi satu arah atau *half – duplex* dari server ke client. Protokol SSE lebih menyerupai HTTP *Streaming*, yang mana membuka satu koneksi selama mungkin untuk pengiriman notifikasi secara akurat atau *real time*. SSE lebih mudah diterapkan dan tidak memerlukan konfigurasi lebih untuk mendapatkan komunikasi satu arah yang lebih stabil (Estep, 2013).



Gambar 2.3 Mekanisme kerja protokol SSE

Sumber: (Real-time Web Apps, 2017)

Gambar 2.3 menunjukan mekanisme kerja protokol SSE antara client dengan server. Awalnya client melakukan permintaan koneksi dan membentuk proses yang merujuk kondisi terbaru pada database. Saat koneksi telah terbentuk,server akan mengirimkan informasi dari waktu ke waktu. Pada sisi client akan mendapatkan pembaruan informasi dari waktu ke waktu selama koneksi masih terbentuk (Pragya, 2016). SSE memiliki fitur*automatic reconnection*. Jadi apabila koneksi *client* atau *browser* terputus maka akan langsung mencoba untuk

terhubung kembali secara otomatis ke server. Tetapi pada sisi *server* tidak terdapat pemberitahuan apabila *client* terputus dari *server*.

2.5 Python

Bahasa pemrograman python merupakan bahasa pemrograman tingkat tinggi yang berorientasi objek. Struktur data tingkat tinggi yang dikombinasikan dengan *dynamic typing* dan *dynamic binding* membuat bahasa python banyak digunakan untuk *Rapid Application Development*. Bahasa python juga digunakan sebagai *scripting* untuk menghubungkan komponen – komponen yang ada (Python, 2017).

Bahasa pemrograman python memiliki struktur dan desain *syntax* yang singkat dan cukup sedikit dibandingkan dengan bahasa pemrograman beriorientasi objek lainnya. Hal ini dianggap lebih efektif dan fleksibel untuk meningkatkan kemampuan *programmer*.

Tabel 2.2 Contoh Source Code Python

1	`print ("Hello, World!")`

Tabel 2.2 adalah contoh dari penulisan dalam bahasa python. Pada *source code* diatas terdapat dua bagian yaitu *command* dan *statement*. Perintah print pada kolom kedua adalah perintah untuk mencetak *statement* Hello, World! untuk ditampilkan pada layar. Bahasa python juga sering digunakan untuk mengembangkan aplikasi yang bersifat *cross-platform*. Jadi bahasa pemrograman python dapat digunakan pada beberapa sistem operasi dan juga bahasa ini bersifat *open source* sehingga para pengembang aplikasi dapat menggunakannya dengan gratis.

2.6 Javascript

Javascript adalah bahasa pemrograman yang sering digunakan untuk membuat halaman web lebih interaktif pada pengguna (Chapman, 2017). Contoh penggunaannya adalah dapat digunakan untuk memodifikasi bentuk konten, mengubah gambar, membuka jendela baru dan dengan bantuan CSS dapat diaplikasikan untuk membuat *Dynamic HyperText Markup Language* (DHTML). Javascript sendiri merupakan bahasa pemrograman tingkat tinggi yang berorientasi objek yang diinterpretasikan pada sisi client. Sementara java adalah bahasa pemrograman tingkat tinggi berorientasi objek yang disusun pada sisi client.

2.7 Psutil

Psutil (Python System and Process Utilities) adalah *library cross-platform* yang berfungsi untuk mendapatkan informasi dari proses yang sedang bekerja dan penggunaan sistem (SSE, *memory*, *disk*, jaringan, sensor) pada python (Psutil

Documentation, 2017). Psutil sangat berguna terutama pada *monitoring* sistem, *profiling*, pembatasan *resource* proses dan manajemen pada proses yang sedang bekerja. Psutil memiliki banyak fungsi tergantung dari perintah yang digunakan. Perintah yang dapat digunakan pada psutil adalah: *ps, top, lsof, netstat, ifconfig, who, df, kill, free, nice, ionice, iostat, iotop, uptime, pidof, tty, taskset, pmap*. Psutil dapat digunakan pada sistem operasi Linux, Windows, OSX, Sun Solaris, FreeBSD, OpenBSD dan NetBSD pada arsitektur 32-bit dan 64-bit.Tabel 2.3 adalah perintah untuk melakukan instalasi *psutil* pada ubuntu.

Tabel 2.3 Tabel Instalasi Psutil

1	`$ pip install psutil`