

---

## IMPLEMENTASI KOMUNIKASI FULL DUPLEX MENGGUNAKAN WEB SOCKET PADA SISTEM INFORMASI PENGELOLAAN ANGGARAN UNIVERSITAS ABC

**Husen**

Fakultas Teknik, Program Studi Teknik Informatika  
Universitas Siliwangi  
Email: husen@student.unsil.ac.id

**Alam Rahmatulloh**

Fakultas Teknik, Program Studi Teknik Informatika  
Universitas Siliwangi  
Email: alam@unsil.ac.id

**Heni Sulastri**

Fakultas Teknik, Program Studi Teknik Informatika  
Universitas Siliwangi  
Email: heni.sulastri@unsil.ac.id

### ABSTRAK

Penelitian ini membahas mengenai sistem pengelolaan anggaran di Universitas ABC yang didalamnya memuat manajemen pengelolaan anggaran biaya dari setiap bagian mulai dari pengajuan rencana anggaran biaya sampai dengan realisasi anggaran. Data dari hasil pengelolaan anggaran setiap bagian dapat dimonitoring oleh pihak pimpinan, namun dalam penyajian datanya belum *real-time* sehingga interaksi *user* dalam monitoring dan analisis data pada sistem kurang interaktif. Saat ini teknologi yang sedang tren pada aplikasi berbasis *web* yang *real-time* yaitu *web socket* (WS). WS dapat menyajikan komunikasi dua arah secara bersamaan (*full-duplex*). Selain itu menurut beberapa penelitian teknologi WS juga dapat meningkatkan performa sistem menjadi lebih baik. Maka pada penelitian ini akan di terapkan teknologi *web socket* sehingga proses monitoring data anggaran dapat dilakukan secara *real-time* dengan menyajikan data tersebut ke dalam bentuk grafik dan persentase. Hasil penelitian yang dilakukan menunjukkan bahwa dengan menggunakan WS nilai perbandingan *Round Trip Time* (RTT) yaitu 1:135 ms jika dibandingkan *ajax*, sehingga penggunaan teknologi WS lebih cepat dan performanya jauh lebih baik.

**Kata Kunci:** anggaran, *full-duplex*, monitoring, *real-time*, *web socket*.

### ABSTRACT

*This study discusses the budget management system at ABC University which includes the management of budget cost management from each unit starting from the submission of a budget plan to budget until the realization of the budget. Data from the budget management results of each section can be monitored by the leadership, but in the presentation of the data has not been real-time so that user interaction in monitoring and data analysis on the system is less interactive. Currently trending technology in real-time web-based applications is web socket (WS). WS can present two-way communication simultaneously (full-duplex). In addition, according to some research WS technology can also improve system performance for the better. So in this study will be applied Web Socket technology so that the process of monitored budget data can be done in real-time by presenting the data in the form of graphs and percentages. The result of the research shows that by using WS Round Trip Time (RTT) value is 1: 135 ms if compared ajax, so the use of WS technology is faster and its performance is much better.*

**Keywords:** budget, *full-duplex*, monitored, *real-time*, *web socket*.

### 1. PENDAHULUAN

Pengelolaan anggaran merupakan suatu permasalahan yang umum terjadi di berbagai bidang ilmu, terjadi akibat tidak adanya manajemen anggaran yang baik, terencana, terukur dan realisasi dapat dipantau [1]. Untuk itu dibutuhkan sebuah sistem dalam mengatasi permasalahan tersebut, seperti halnya di Universitas ABC yang sudah menggunakan sistem pengelolaan anggaran yang didalamnya mengelola data

Rencana Anggaran Biaya (RAB) yang dapat di monitoring oleh pihak pimpinan. Namun masih terdapat kendala dalam segi penyajian data yang masih menggunakan sistem satu arah (*simplex*) [2] sehingga belum *real-time* dan performa sistem yang masih lambat. Maka dibutuhkan penyajian data yang *real-time* untuk memonitoring hasil anggaran yang di serap atau yang direalisasikan sehingga meningkatkan performa sistem dan mendukung pihak pemimpin dalam memonitoring serta menganalisis hasil dari pengelolaan anggaran. Dengan menerapkan *transmisi full-duplex*, dua station secara simultan mengirim dan menerima satu sama lain. Sehingga model ini disebut juga *two-way simultaneous* dan bisa diibaratkan sebagai dua-jalur, dua-jembatan [3]. Teknologi pada aplikasi berbasis web yaitu *Asynchronous JavaScript and XML* (AJAX) dan *websocket* merupakan teknologi yang dapat menyajikan data secara *real-time*, namun AJAX memiliki kekurangan dalam meningkatkan performa sistem dibandingkan *websocket*. Sedangkan *websocket* merupakan teknologi yang dianggap akan banyak digunakan dalam penyajian data *real-time* menggantikan ajax [4]. *Websocket* merupakan protokol komunikasi *full-duplex* (dua arah secara bersamaan) melalui koneksi TCP, ditujukan untuk melakukan pertukaran pesan antar *browser* dan *server* web secara *real-time* [5]. *Websocket* dapat meningkatkan performa sistem, hal ini mengurangi beban jaringan bila dibandingkan dengan metode konvensional. Teknologi *websocket* sering digunakan saat komunikasi data dua arah. Secara khusus, efek positif dari penggunaan teknologi *websocket* lebih terasa saat menggunakan aplikasi web berbasis komunikasi data dua arah *real-time* [6]. Sehingga dalam penelitian ini akan menerapkan teknologi *websocket* dalam penyajian data pengelolaan RAB secara *real-time*. Tujuan dari penelitian ini adalah menyajikan data kedalam bentuk grafik dan persentase secara *real-time* agar memudahkan pihak pimpinan dalam memonitoring data anggaran.

## 2. METODOLOGI PENELITIAN

### 2.1 Pengumpulan Data

#### 2.1.1 Wawancara

Wawancara (Interview), wawancara merupakan teknik pengumpulan data dengan cara bertatap muka langsung dengan narasumber untuk saling bertukar informasi melalui proses tanya jawab, sehingga dapat menarik sebuah kesimpulan tentang permasalahan yang ada.

#### 2.1.2 Observasi

Observasi merupakan teknik pengumpulan data, dengan cara melakukan pengamatan langsung terhadap objek penelitian agar dapat melihat kegiatan yang dilakukan.

#### 2.1.3 Pengembangan Perangkat Lunak

Metodologi pengembangan perangkat lunak pada penelitian ini menggunakan RUP (*Rational Unified Process*). Metode RUP memiliki empat fase, yaitu *inception*, *elaboration*, *construction* dan *transition*.

#### 2.1.4 Inception

Pada tahap pertama ini dilakukan proses *business modelling* yaitu memodelkan suatu proses bisnis yang dibutuhkan kemudian mendefinisikan kebutuhan suatu sistem (*requirements*).

#### 2.1.5 Elaboration

Pada tahapan kedua lebih difokuskan terhadap perencanaan arsitektur sistem seperti melakukan analisis sistem dan desain sistem.

#### 2.1.6 Construction

Pada tahap ketiga dilakukan pengembangan komponen dan fitur-fitur sistem serta melakukan implementasi dan pengujian sistem yang fokus terhadap implementasi perangkat lunak pada kode program.

### 2.1.7 Transition

Tahap yang terakhir melakukan *deployment* atau instalasi sistem yang bertujuan untuk penerapan sistem dan pelatihan kepada pengguna sistem serta pemeliharaan sistem.

## 3. HASIL DAN PEMBAHASAN

### 3.1 Pengumpulan Data

#### 3.1.1 Wawancara

Hasil dari melakukan wawancara yang dilakukan dengan proses tanya jawab kepada narasumber di tempat penelitian, maka ada beberapa masalah yang sedang terjadi di dalam proses pengelolaan perencanaan anggaran diantaranya:

- Dalam penyajian data hasil pengelolaan anggaran dari setiap unit masih disajikan secara manual yaitu ketika admin ingin mendapatkan data yang terbaru dari hasil pengelolaan anggaran maka harus dilakukan *reload page* terlebih dahulu sehingga monitoring yang dilakukan kurang interaktif.
- Sistem masih menggunakan *ajax* dalam menangani permasalahan *real-time* data sehingga performa sistem menjadi menurun.

#### 3.1.2 Observasi

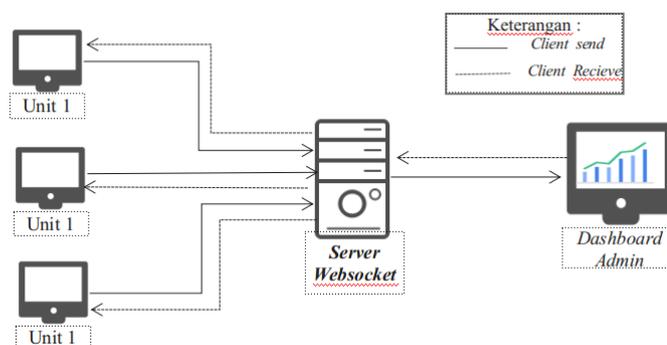
Hasil dari pengumpulan data dengan melakukan observasi dapat menyimpulkan beberapa kegiatan diantaranya sebagai berikut :

- Diperlukannya penyajian data secara *real-time* pada *dashboard* admin, sehingga mendukung dalam proses monitoring dan analisis anggaran yang diajukan dan direalisasikan dari setiap unit.
- Diperlukannya komunikasi *full-duplex* untuk mengatasi permasalahan penyajian *real-time* data dan peningkatan performa sistem.

### 3.2 Pengembangan Perangkat Lunak

#### 3.2.1 Inception

- Identifikasi Penyebab Masalah  
Pada sistem pengelolaan anggaran yang sedang berjalan di Universitas ABC ini terdapat beberapa *user* yang berperan di dalam sistem tersebut. *User* admin selaku pihak pimpinan dapat melakukan monitoring hasil dari pengelolaan data anggaran seperti serapan biaya anggaran, pengajuan anggaran serta realisasi anggaran yang dilakukan oleh setiap *user* unit. Namun dalam penyajian data di *dashboard* admin masih belum *real-time* sehingga ketika ada informasi yang terbaru admin harus *reload* halaman *dashboard* tersebut dan ini merupakan suatu kendala dalam memonitoring data anggaran yang sedang berjalan.
- Sistem Usulan  
Sistem yang diusulkan dalam permasalahan penyajian data secara *real-time* ini yaitu ketika unit melakukan pengajuan yang ada maka dalam waktu yang bersamaan informasi penyerapan yang dilakukan oleh unit akan langsung di ketahui oleh admin dengan bentuk tampilan persentase atau grafik secara *real-time*, begitupun saat unit melakukan realisasi.



Gambar 1. Sistem Usulan Dengan Menerapkan Komunikasi Full-Duplex

Pada gambar 1 merupakan gambaran sistem usulan dengan menerapkan teknologi *websocket* untuk merubah komunikasi secara *full-duplex* pada saat melakukan proses penyerapan biaya atau realisasi yang dilakukan unit dan langsung dapat dimonitoring oleh admin secara *real-time*.

### 3.2.2 Elaboration

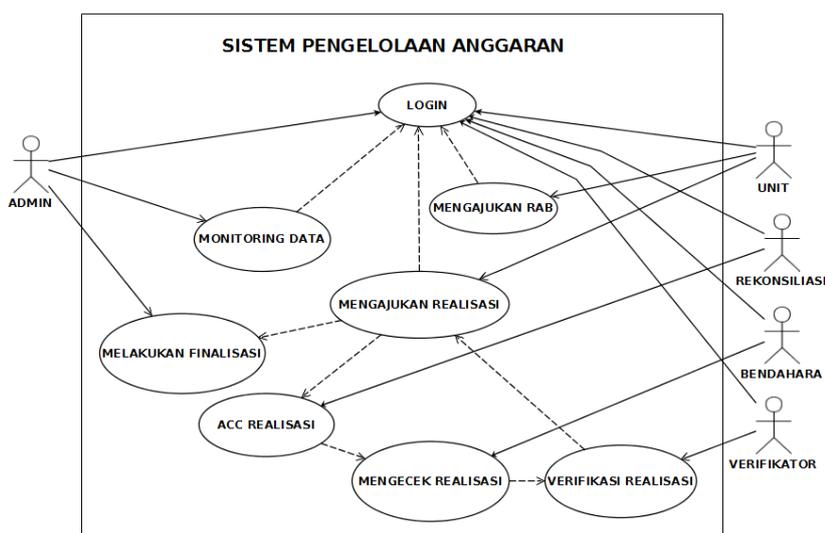
#### a. Identifikasi Use Case

Pada tabel 1 merupakan penjelasan dari beberapa *use case* dari sistem pengelolaan anggaran.

**Tabel 1. Daftar use case**

Aktor	Aktivitas
1. Admin	Monitoring Data Melakukan Finalisasi
2. Unit	Mengajukan RAB Mengajukan Realisasi
3. Verifikator	Verifikasi Realisasi
4. Bendahara	Mengecek Realisasi
5. Rekonsiliasi	Acc Realisasi

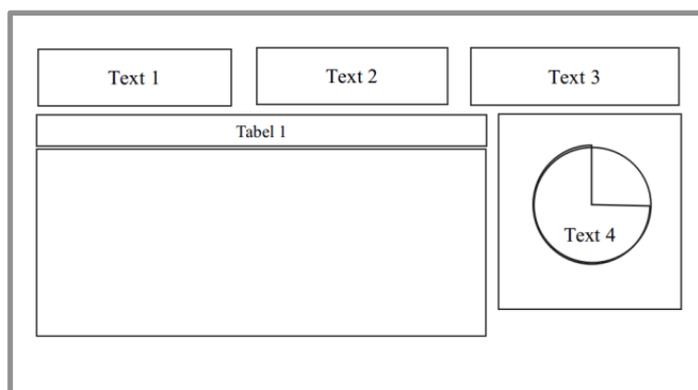
Pada Gambar 2 merupakan gambar hubungan dari masing - masing *use case* yang ada pada sistem pengelolaan anggaran dengan memiliki 5 aktor yang saling berinteraksi pada sistem.



**Gambar 2. Use Case Diagram**

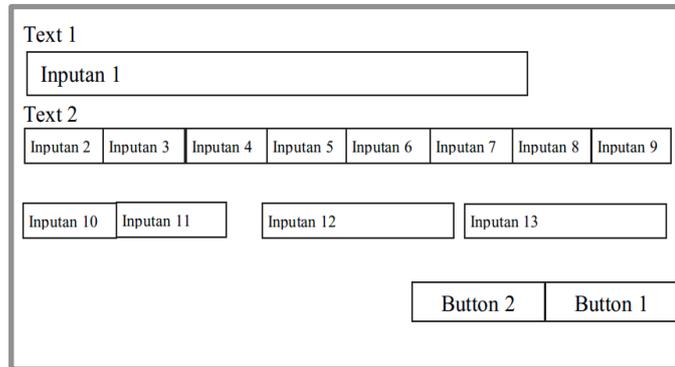
#### b. Perancangan Antarmuka

Rancangan antarmuka pada sistem pengelolaan anggaran dengan komunikasi *full-duplex* ini adalah sebagai berikut :



**Gambar 3. Antarmuka Dashboard Admin**

Pada gambar 3 merupakan rancangan antarmuka *dashboard* admin yaitu *text1* merupakan info dari sisa pagu, *text 2* adalah info serapan, *text 3* info realisasi, *text 4* info grafik pengajuan dan *table 1* adalah rekap data RAB per output.



**Gambar 4. Antarmuka Inputan Rincian Biaya**

Pada gambar 4 merupakan rancangan antarmuka pengisian rincian biaya yaitu inputan 1 adalah keterangan rincian biaya, inputan 2,4,6,8 adalah volume rincian biaya, inputan 3,5,7,9 dan 11 adalah satuan kegiatan, inputan 10 total keseluruhan volume, inputan 12 isian biaya, inputan 13 total dari jumlah volume di kalikan dengan biaya, *button 1* tombol untuk menyimpan dan *button 2* tombol untuk membatalkan.

### 3.2.3 Construction

a. Implementasi *script* di sisi *server*

Dalam menjalankan *server websocket* ini hal yang pertama harus dilakukan adalah mengaktifkannya melalui PHP CLI (*Command Line Interface*) agar dapat mengirim pesan dan menerima pesan dengan perintah `php server.php` pada terminal. Adapun isi file dari *server.php* dapat dilihat pada gambar 5.

```
1 <?php
2 require __DIR__ . "/vendor/autoload.php";
3
4 use Ratchet \Server\IoServer;
5 use MyApp\Anggaran;
6 use Ratchet\Http\HttpServer;
7 use Ratchet\WebSocket\WsServer;
8
9 $server = IoServer::factory(
10     new HttpServer(
11         new WsServer(
12             new Anggaran
13         )
14     ),
15     8080
16 );
17
18 $server->run();
19
20 ?>
```

**Gambar 5. Script Server Websocket**

Pada *server websocket* ini *port* yang digunakan yaitu *port 8080* seperti yang terlihat di gambar 6 pada baris ke 15. Perintah `$server->run()` pada baris ke 18 berfungsi untuk menjalankan *library* Ratchet dan *class* Anggaran yang merupakan konfigurasi utama dari aplikasi *websocket server*. Adapun *script* dari *class* Anggaran dapat dilihat pada gambar 6.

```
9 class Anggaran implements MessageComponentInterface
10 {
11     protected $clients;
12
13     function __construct()
14     {
15         $this->clients = new \SplObjectStorage;
16     }
17
18     public function onOpen(ConnectionInterface $conn)
19     {
20         $this->clients->attach($conn);
21         echo "New Connection ({$conn->resourceId})\n";
22     }
23
24     public function onMessage(ConnectionInterface $from, $msg)
25     {
26         foreach ($this->clients as $client) {
27             if($client !== $from)
28             {
29                 $client->send($msg);
30                 echo $msg;
31             }
32         }
33     }
34
35     public function onClose(ConnectionInterface $conn)
36     {
37         $this->clients->detach($conn);
38         echo "Connection from {$conn->resourceId} has disconnected\n";
39     }
40
41     public function onError(ConnectionInterface $conn, \Exception $e)
42     {
43         echo "An Error has occured {$e->getMessage()}";
44         $conn->close();
45     }
46 }
47 }
```

Gambar 6. Script Websocket Class Anggaran

Pada gambar 6 terdapat 4 fungsi yaitu `onOpen()`, `onMessage()`, `onClose()` dan `onError()`. Pada fungsi `onOpen()` ini akan menampilkan *resourceID* dari setiap *client* yang terhubung pada *server websocket*, sedangkan fungsi `onMessage()` menangani ketika terjadinya pengiriman atau penerimaan pesan dari *client* kepada *client* yang lain berdasarkan id *client* tersebut secara *real-time*. pada fungsi `onClose()`, fungsi ini akan menampilkan pemberitahuan dari *client* yang telah keluar dari koneksi terhadap *server websocket*. Sedangkan fungsi yang terakhir yaitu fungsi `onError()` akan menampilkan pesan kesalahan ketika terjadi *error*.

b. Implementasi script di sisi *client*

Pada penelitian ini *client* menggunakan bahasa pemrograman *javascript* dalam melakukan koneksi terhadap *server websocket*. Pada *javascript* terdapat *object* khusus untuk terkoneksi kepada *server websocket* yaitu dengan menggunakan perintah `new WebSocket('ws://url:port')` adapun script yang di implementasikan pada aplikasi pengelolaan anggaran ini bisa dilihat pada gambar 7.

```
5
6     var conn = new WebSocket('ws://localhost:8080');
7
8
9     conn.onopen = function(e){
10         console.log("Websokcet Berjalan dengan lancar!");
11     };
12
13     conn.onmessage = function(e){
14         var data = e.data;
15         if(data==1) //data 1 == serapan
16         {
17             reload_output();
18             get_realisasi_analisis_eis();
19         }else if(data==2){ //data 2 == realisasi
20             load_realisasi_eis(data);
21         }
22     }
```

Gambar 7. Implementasi Script Websocket Di Sisi Client

Pada gambar 7 di baris 6 merupakan perintah untuk melakukan koneksi ke *server websocket*. Sedangkan pada baris 8 - 11 merupakan perintah untuk menampilkan pesan ketika *client* berhasil terkoneksi dengan *server websocket*.

c. Implementasi Perangkat Lunak

Implementasi perangkat lunak pada sistem pengelolaan anggaran dengan komunikasi *full-duplex* ini adalah sebagai berikut :

d. Implementasi *dashboard* admin

Tampilan *dashboard* admin berfungsi menampilkan informasi yang berhubungan dengan proses penyusunan anggaran seperti presentase serapan biaya, realisasi dan pagu yang tersisa seperti pada gambar 8.



Gambar 8. *Dashboard* Admin

e. Implementasi pengajuan RAB

Tampilan saat unit mengisi rincian biaya untuk melakukan pengajuan RAB bisa di lihat pada gambar 9.

volume	Satkeg	Rp	Biaya	Rp	Total Biaya
0	Satuan	0	Satuan	0	Satuan
0	Satuan	0	Satuan	0	Satuan
0	Satuan	0	Satuan	0	Satuan
0	Satuan	0	Satuan	0	Satuan
0	Satuan	0	Satuan	0	Satuan

Gambar 9. Proses Pengisian Rincian Biaya

Pada gambar 9 unit mengisi formulir isian rincian biaya untuk melakukan pengajuan RAB. Ketika unit menekan tombol simpan, maka rincian biaya akan tersimpan dan jumlah biaya pengajuan RAB otomatis akan terupdate di *dashboard* admin secara *real-time* dalam bentuk presentase atau grafik.

### 3.2.4 Transition

Pada tahap ini dilakukan pengujian nilai *Round Trip Time* (RTT) terhadap teknologi *websocket* dan dibandingkan dengan *ajax*. RTT adalah waktu yang dibutuhkan sebuah paket untuk melakukan perjalanan dari sumber ke tujuan dan kembali lagi. RTT merupakan ukuran penting dalam menentukan penuhnya suatu koneksi [7]. Nilai RTT sangat tergantung dari kondisi trafik. Semakin padat trafik maka semakin besar RTT yang dihasilkan [8].

**Tabel 2. Nilai round trip time websocket**

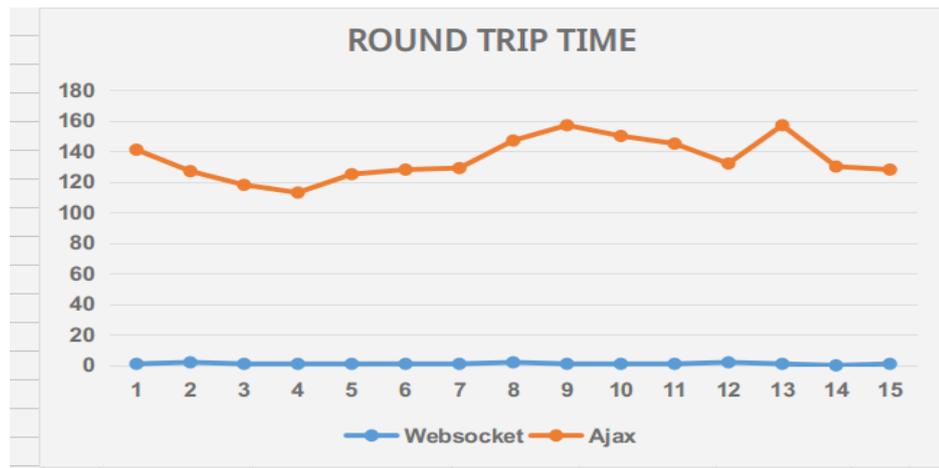
<b>Round Trip Time Websocket</b>			
<b>No</b>	<b>Send (ms)</b>	<b>Receive (ms)</b>	<b>RTT (ms)</b>
1	894	895	1
2	132	134	2
3	99	100	1
4	359	360	1
5	889	890	1
6	826	827	1
7	103	104	1
8	574	576	2
9	524	525	1
10	841	842	1
11	772	773	1
12	353	355	2
13	849	850	1
14	853	853	0
15	149	150	1
<b>AVG</b>	<b>547.8</b>	<b>548.9</b>	<b>1.13</b>

Pada tabel 2 terlihat bahwa pengujian terhadap *websocket* dengan waktu rata-rata pengiriman (*send*) 547.8 ms dan penerimaan (*receive*) adalah 548.9 ms sedangkan waktu RTT sangat rendah dengan nilai rata rata 1.13 ms , karena itu *websocket* sangat mendukung dalam peningkatan performa sistem.

**Tabel 3. Nilai round trip time ajax**

<b>Round Trip Time Websocket</b>			
<b>No</b>	<b>Send (ms)</b>	<b>Receive (ms)</b>	<b>RTT (ms)</b>
1	626	767	141
2	321	448	127
3	637	755	118
4	661	774	113
5	804	929	125
6	206	334	128
7	326	455	129
8	126	273	147
9	608	765	157
10	747	897	150
11	155	300	145
12	368	500	132
13	302	459	157
14	327	457	130
15	147	275	128
<b>AVG</b>	<b>424.07</b>	<b>559.2</b>	<b>135.13</b>

Terlihat ada tabel 3 bahwa pengujian dengan teknologi sebelumnya yang dalam hal ini dicoba menggunakan *ajax* didapatkan nilai rata-rata *request* adalah 424.07 ms dan *response* adalah 559.2 ms, sedangkan nilai rata-rata dari RTT sangat besar dibandingkan dengan nilai RTT pada *websocket* yaitu 135.13 ms.



Gambar 10. Perbandingan RTT Websocket dan Ajax

Pada gambar10 merupakan gambar grafik perbandingan dari nilai RTT antar *websocket* dan *ajax*. Pada pengujian ini terlihat bahwa nilai RTT dari *websocket* sangat rendah dibandingkan dengan *ajax* dan rata-rata waktu RTT nya pun stabil dibandingkan dengan *ajax*.

### 3.3 Kelebihan Dan Kekurangan

Kelebihan dari sistem yang sudah dibangun:

- Tampilan sistem atau aplikasi *responsive* saat di buka di perangkat *mobile*
- Sistem monitoring menjadi lebih interaktif dengan kinerja sistem yang lebih baik sehingga mudah digunakan dan dipahami oleh *user*

Adapun kekurangannya saat mengaktifkan *server websocket* masih manual yaitu dengan mengetikkan terlebih dahulu perintah pada *php cli*.

## 4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan di Universitas ABC dapat diambil kesimpulan bahwa telah berhasil dibuat sebuah sistem yang dapat menyajikan data secara *real-time* menggunakan komunikasi *full-duplex* menggunakan teknologi *websocket* dengan nilai *latency* lebih cepat 1:135 ms dibandingkan dengan *ajax*. Sistem ini dapat membantu pihak pimpinan atau admin dalam melakukan monitoring dan menganalisis data anggaran yang disajikan dalam bentuk grafik dan persentase secara *real-time*. Dengan menerapkan teknologi *websocket* juga dapat meningkatkan performa sistem yang lebih baik.

## DAFTAR PUSTAKA

- [1] A. Rahmatulloh and Husen, "Sistem Informasi Manajemen Anggaran (Simangga) Perguruan Tinggi Berbasis Web (Studi Kasus: Universitas Siliwangi)," *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 3, no. 2, pp. 14-20, 12 2017.
- [2] D. Sugiono, *Komunikasi Data dan Interface*, Malang: Kementerian Pendidikan & Kebudayaan, 2013.
- [3] S. Sukaridhoto, *Komunikasi Data dan Komputer*, Surabaya: Politeknik Elektronika Negeri Surabaya, 2016.
- [4] Q. Liu and X. Sun, "Research of Web Real-Time Communication Based on Web Socket," *Int. J. Communications, Network and System Sciences*, vol. 5, pp. 797-801, 12 2012.
- [5] S. Chernyi, "The implementation of technology of multi-user client-server applications for systems of decision making support," *Metallurgical and Mining Industry*, no. 3, 2015.
- [6] J.-t. Park, H.-s. Hwang, J.-s. Yun and I.-y. Moon, "Research and Performance Analysis of HTML5 WebSocket for a Real-time Multimedia Data Communication Environment," *Advanced Science and Technology Letters*, vol. 46, pp. 307-312, 2014.
- [7] E. m. A. reyouchi, K. Ghoumid, K. Ameziane and O. E. Mrabet, "Performance Analysis of Round Trip Time in Narrowband RF Networks For Remote Wireless," *International Journal of Computer Science & Information Technology (IJCSIT)*, vol. 5, no. 5, October 2013.

- [8] R. Tulloh, R. M. Negara and A. N. Hidayat, "Simulasi Virtual Local Area Network (VLAN) Berbasis Software Defined Network (SDN) Menggunakan POX Controller," *Jurnal Informatika, Telekomunikasi dan Elektronika*, vol. 10, no. 1, November 2015.