

IMPLEMENTASI WEB SOCKET PADA SISTEM PELAYANAN PASIEN RAWAT JALAN PADA PUSKESMAS KABUPATEN GOWA

REZA MAULANA¹

Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Islam
Negeri Alauddin Makassar
E-mail: saya@rezamaulana.com

ABSTRAK

Pelayanan Pasien rawat jalan Puskesmas Pallangga pada umumnya memiliki alur yang panjang. Pasien yang datang berobat mengambil nomor antrian pasien kemudian menuju ke kamar buku untuk memperlihatkan kartu berobat. Pada tahap ini pasien yang seharusnya hanya menunggu antrian panggilan untuk diperiksa tetapi masih harus berjalan menuju kamar buku. Pasien akan kesulitan mengikuti alur pelayanan yang panjang sebelum memasuki ruangan dokter, begitu pula saat pengambilan obat, pasien masih harus menunggu obat disiapkan oleh apoteker. Sistem ini akan menggunakan teknologi *websocket*. *Websocket* standar teknologi web untuk melakukan komunikasi secara realtime pada web ataupun aplikasi mobile. Dengan mengimplementasikan Web Socket pada sistem informasi rawat jalan akan mengakibatkan alur proses data bisa menjadi lebih cepat dari sisi transfer dan *User Experience* dibanding dengan teknologi web native tanpa web socket dapat memudahkan pelayanan pasien rawat jalan di Puskesmas Pallangga.

Kata Kunci: *Websocket*, *web base*, Sistem, Pelayanan, Pasien, Rawat Jalan, Puskesmas.

I. PENDAHULUAN

Puskesmas merupakan salah satu instansi pemerintahan di bidang kesehatan. Puskesmas Pallangga mempunyai peranan sangat penting di Kecamatan Palangga dalam meningkatkan kesehatan masyarakat. Dalam melakukan pelayanan rawat jalan kepada pasien Puskesmas Pallangga memiliki 3 poli, yaitu poli umum, poli gigi, dan poli kesehatan ibu dan anak (KIA). Pada poli umum pemeriksaan pasien terbagi menjadi 3, yaitu pemeriksaan anak, dewasa dan lansia.

Berdasarkan alur proses bisnis pada Puskesmas Kabupaten Gowa, pelayanan pasien rawat jalan memiliki alur yang panjang. Setelah mengambil nomor antrian pasien kemudian menuju ke kamar buku untuk memperlihatkan kartu berobat. Pada tahap ini pasien yang seharusnya hanya menunggu antrian panggilan untuk diperiksa tetapi masih harus berjalan menuju kamar buku. Pasien akan kesulitan mengikuti alur pelayanan yang panjang sebelum memasuki ruangan dokter, begitu pula saat pengambilan obat, pasien masih harus menunggu obat disiapkan oleh apoteker.

Pada Sistem Pelayanan Pasien Rawat Jalan di Puskesmas Pallangga, *websocket* akan digunakan pada tahap pendaftaran antrian pasien. Pada saat data antrian pasien telah disimpan maka secara otomatis daftar antrian pemeriksaan awal yang berada di meja perawat akan terisi tanpa melakukan proses *reload page* pada sistem. Begitu pula saat pemeriksaan awal selesai, maka daftar antrian pemeriksaan lanjutan di meja dokter akan terisi secara otomatis. Daftar antrian obat pada apotik akan terisi secara otomatis saat dokter selesai memeriksa dan menuliskan resep obat pasien.

Websocket merupakan standar baru untuk melakukan komunikasi secara *realtime* pada *web* ataupun aplikasi *mobile*. Pada dasarnya dengan *websocket*, *client* dapat menerima data dari *server* tanpa melakukan *request* data terlebih dahulu. Berbeda dengan *Ajax* yang hanya dapat menerima data saat *client* melakukan *request* terlebih dahulu kepada *server*.

Dengan tersedianya sistem informasi rawat jalan berbasis *online* dengan menggunakan *websocket* dapat mempercepat proses pelayanan pasien rawat jalan di Puskesmas Pallangga.

Dalam sebuah penelitian oleh Saputro (2017) dengan judul penelitian “Sistem Rekam Medis Rawat Jalan pada Puskesmas Pembantu Sukorejo di Kabupaten Nganjuk”. Tujuannya adalah untuk membantu proses rekam medis dengan sistem informasi pendaftaran pasien rawat jalan berupa sebuah aplikasi yang dapat mempermudah dalam pengolahan data pasien. Pembeda dari sistem diatas adalah sistem tersebut menggunakan aplikasi berbasis desktop, sedangkan sistem yang digunakan oleh peneliti saat ini adalah sistem berbasis web secara *real time* dengan menggunakan *websocket*.

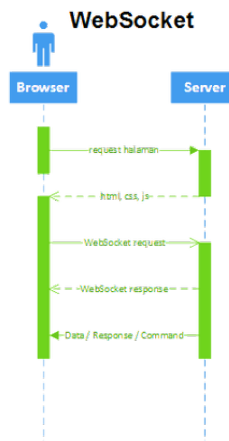
Kemudian penelitian dari Deviana (2018) dengan judul penelitian “Sistem Informasi Rawat Jalan Pada Puskesmas”. Tujuannya adalah memudahkan petugas dalam memperbarui data-data yang masuk serta memudahkan dalam melakukan pencarian data. Pembeda dari kedua penelitian ini adalah bahasa pemrograman yang digunakan. Peneliti diatas menggunakan bahasa pemrograman PHP murni dalam membangun sistemnya, sedangkan peneliti saat ini menggunakan bahasa pemrograman PHP dan *Framework CodeIgniter*.

Sedangkan penelitian dari Basiroh dan Auliya Burhanuddin (2016) dengan judul penelitian “Sistem Informasi Pengelolaan Data Pasien Rawat Jalan Pada Klinik Akupuntur dan Home Care Sehat Migoenani”. Tujuannya adalah untuk mempermudah sistem yang lama, yang masih manual menjadi sistem informasi yang berbasis komputerisasi, mempercepat pengolahan data dan transaksi, serta mempermudah petugas dalam melaksanakan tugasnya. Pembeda dari kedua penelitian ini adalah peneliti diatas menggunakan Microsoft Visual Foxpro 9.0 dalam mengembangkan sistem ini yang berarti sistem tersebut berbasis *desktop*, sedangkan peneliti saat ini menggunakan bahasa pemrograman PHP dan

Framework CodeIgniter dikarenakan peneliti akan mengembangkan sistem berbasis web.

II. METODOLOGI PENELITIAN

Websocket digunakan untuk sistem yang membutuhkan data secepat mungkin dan digunakan agar sistem dapat berjalan secara *real time*.



Gambar 1 Alur proses *Websocket*

Pada gambar 1 terlihat *Websocket client* hanya melakukan *request* pada koneksi pertama, kedua dan untuk selanjutnya *server* dapat merespon tanpa *client* harus melakukan *request* terlebih dahulu. Sehingga *Websocket* mampu menghemat *bandwidth* dan *latency* jaringan.

a. Pendekatan Penelitian

Penelitian ini menggunakan pendekatan penelitian saintifik yaitu pendekatan berdasarkan ilmu pengetahuan dan teknologi.

b. Sumber Data

Sumber data dari penelitian ini berdasarkan dari *Library Research* yang merupakan cara mengumpulkan data dari beberapa buku, jurnal, skripsi, tesis maupun literatur lainnya yang dapat dijadikan acuan pembahasan dalam masalah ini. Peneliti hanya mengambil beberapa data terkait pada sumber data *online* atau internet ataupun hasil dari penelitian sebelumnya sebagai bahan referensi bagi peneliti selanjutnya.

c. Metode Pengumpulan Data

1) Observasi

Studi lapangan (observasi) merupakan teknik pengumpulan data dengan langsung terjun ke lapangan untuk mengamati permasalahan yang terjadi secara langsung di tempat kejadian secara sistematis kejadian-kejadian, perilaku, objek-objek yang dilihat dan hal-hal lain yang diperlukan dalam mendukung penelitian yang sedang berlangsung. Dalam penelitian ini, peneliti melakukan pengamatan

secara langsung di Puskesmas Pallangga tentang proses pelayanan pasien rawat jalan di puskesmas tersebut.

2) Wawancara

Wawancara adalah proses tanya jawab yang terjadi antara orang yang mencari informasi (pewawancara) dengan orang yang memberi informasi (narasumber) dengan tujuan untuk mengumpulkan data atau informasi. Dalam penelitian ini peneliti mewawancarai pihak dari Puskesmas Pallangga.

3) Studi Literatur

Studi literatur adalah usaha untuk mempelajari produk-produk temuan ilmiah yang didokumentasikan dalam bentuk tulisan, untuk mendukung dan memperkuat argument dari penelitian baru atau penelitian lanjutan yang sedang dilakukan.

III. HASIL DAN PEMBAHASAN

Pada Sistem Pelayanan Pasien Rawat Jalan di Puskesmas Pallangga, *websocket* akan digunakan pada tahap pendaftaran antrian pasien, pemeriksaan pasien oleh perawat dan dokter, serta saat proses pengambilan obat, agar proses pelayanan dapat dilakukan dengan cepat dan lancar. Komponen *websocket* yang digunakan pada sistem ini adalah *library javascript Socket.IO*. Dengan *library javascript Socket.IO*, *client* dan *server* dapat melakukan pengiriman dan penerimaan data secara *realtime*. *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*.

1. Implementasi Aplikasi

Tahap implementasi sistem merupakan tahap menerjemahkan perancangan berdasarkan hasil analisis dalam bahasa yang dapat dimengerti oleh mesin serta penerapan perangkat lunak pada keadaan yang sesungguhnya.

a. Implementasi *Script* di sisi *server*

Dalam menjalankan server *Socket.IO* ini hal yang pertama harus dilakukan adalah mengaktifkannya melalui CLI (*Command Line Interface*) agar dapat mengirim pesan dan menerima pesan dengan perintah *node server.js* pada terminal.

```
all@IRSYAD MINGW64 /e/Data Aplikasi/xampp/htdocs/Pelayanan
$ node server.js
Server listening at port 3000
```

Gambar 2 *Script* menjalankan server *node.js*

Adapun isi file dari *server.js* dapat dilihat pada gambar 3. Pada baris 17, terdapat perintah ‘*reload_data*’ yang akan digunakan untuk memuat data saat melakukan pemanggilan ke *server*.

```

1 var socket = require( 'socket.io' );
2 var express = require('express');
3 var app = express();
4 var server = require('http').createServer(app);
5 var io = socket.listen( server );
6 var port = process.env.PORT || 3000;
7 server.listen(port, function () {
8   console.log('Server listening at port %d', port);
9 });
10 io.on('connection', function (socket) {
11   socket.on( 'new_message', function( data ) {
12     // io.sockets.emit( 'new_message', {
13       // message: data.message,
14       // date: data.date,
15       // msgcount: data.msgcount
16     // });
17     io.sockets.emit('reload_Data')]
18   });
19 });

```

Gambar 3 Script server Socket.IO

b. Implementasi Script di sisi client

Pada gambar 4 baris 143 penggunaan *emit()* berfungsi untuk mengirimkan sebuah pesan dari *server* ke *client* dan dari *client* ke *server*.

```

140 $('#simpan').click(function(argument) {
141   // $('#demo-form2').submit();
142   console.log()
143   socket.emit('new_message')
144 })

```

Gambar 4 Script untuk mengirim data

Pada gambar 5 baris 69 penggunaan *method on()* berfungsi untuk menerima data yang kemudian akan diproses oleh fungsi *handler*. Fungsi *handler* pada gambar 5 baris 72 yaitu untuk memuat ulang halaman *client*.

```

69 socket.on('reload_Data', function() {
70   // window.location.reload();
71
72   setTimeout(function(){
73     window.location.reload();
74   },2000);
75 });

```

Gambar 5 Script untuk menerima data

2. Pengujian Aplikasi

a. Pengujian Fungsional Aplikasi

Pada sistem dilakukan pengujian berdasarkan fungsinya. Sistem diakses oleh staf yang akan memasukkan data antrian pasien dan secara langsung akan masuk ke daftar antrian.

Gambar 6 Tampilan Input daftar antrian

Pada gambar 7 daftar antrian yang telah dimasukkan oleh staf akan tampil secara *real time*.

No	No. Rekam Medis	Nama Pasien	Tanggal Lahir	Jenis Kelamin	Status Periksa	Aksi
1	61312	Muhammad Saad	1976-04-12	Laki-Laki	Belum	Periksa
2	51630	Andi Muhammad Agung	2011-07-30	Laki-Laki	Belum	Periksa
3	9023	Nurul Ilimi	2015-10-10	Perempuan	Belum	Periksa
4	71520	Emawati	1996-01-01	Perempuan	Belum	Periksa
5	44002	Syamsuddin Lili	1976-03-18	Laki-Laki	Belum	Periksa

Gambar 7 Tampilan daftar antrian

b. Pengujian Performa Aplikasi

Pada sistem dilakukan pengujian performa dengan melihat kecepatan saat halaman mengirim dan menerima data. *Tools* yang digunakan adalah bawaan dari *browser* google chrome yaitu Chrome DevTools. Chrome DevTools ini dapat digunakan untuk mengecek kecepatan *website* dan mendapatkan laporannya dengan menekan Ctrl+Shift+I pada keyboard.

Pada gambar 8, terlihat hasil tes saat halaman staf mengirim data pada Chrome DevTools. Sedangkan pada gambar 9 adalah hasil tes saat halaman perawat menerima data dari halaman staf.

Name	Status	Type
moment.min.js	200	script
daterangepicker.js	200	script
custom.min.js	200	script
select2.min.js	200	script
?EIO=3&transport=polling&t=NHsy4gh	200	xhr
fontawesome-webfont.woff2?v=4.6.3	200	font
?EIO=3&transport=polling&t=NHsy4gh&sid=2y3MmkOXDYbjFYAAAC	200	xhr
?EIO=3&transport=websocket&sid=2y3MmkOXDYbjFYAAAC	101	websocket

46 requests | 26.0 kB transferred | 1.4 MB resources | Finish: 475 ms | DOMContentLoaded: 351 ms | Load: 369 ms

Gambar 8 Hasil tes halaman staf mengirim data

Name	Status	Type
moment.min.js	200	script
daterangepicker.js	200	script
custom.min.js	200	script
select2.min.js	200	script
?EIO=3&transport=polling&t=NHsy56f	200	xhr
fontawesome-webfont.woff2?v=4.6.3	200	font
?EIO=3&transport=polling&t=NHsy59g&sid=FrDPntdcG6UAv592AAAD	200	xhr
?EIO=3&transport=websocket&sid=FrDPntdcG6UAv592AAAD	101	websocket

45 requests | 15.2 kB transferred | 1.4 MB resources | Finish: 486 ms | DOMContentLoaded: 367 ms | Load: 380 ms

Gambar 9 Hasil tes halaman perawat menerima data

Lebih jelasnya terdapat pada tabel 1 yang menunjukkan hasil pengujian kecepatan halaman staf mengirim data ke halaman perawat. Sedangkan pada tabel 2 adalah hasil pengujian kecepatan halaman perawat saat menerima data dari halaman staf.

Tabel 1 Halaman Staf mengirim data

Objek	Page size	Load Time	Reques t
Halaman Staf	1.4 MB	369 ms	46

Tabel 2 Halaman Perawat menerima data

Objek	Page size	Load Time	Reques t
Halaman Perawat	1.4 MB	380 ms	45

Pada gambar 10, terlihat hasil tes saat halaman perawat mengirim data pada Chrome DevTools. Sedangkan pada gambar 11 adalah hasil tes saat halaman dokter menerima data dari halaman perawat.

Name	Status	Type
daterangepicker.js	200	script
custom.min.js	200	script
select2.min.js	200	script
select2.min.css	200	stylesheet
?EIO=3&transport=polling&t=NHs-ftf	200	xhr
fontawesome-webfont.woff2?v=4.6.3	200	font
?EIO=3&transport=polling&t=NHs-f_k&sid=c8Pu5zxbQ4LmEmbJAAG	200	xhr
?EIO=3&transport=websocket&sid=c8Pu5zxbQ4LmEmbJAAG	101	websocket

55 requests | 15.6 kB transferred | 1.6 MB resources | Finish: 426 ms | DOMContentLoaded: 308 ms | Load: 322 ms

Gambar 10 Hasil tes halaman perawat mengirim data

Name	Status	Type
jqvmap.min.css	200	stylesheet
daterangepicker.css	200	stylesheet
custom.min.css	200	stylesheet
select2.min.css	200	stylesheet
?EIO=3&transport=polling&t=NHs-gQ8	200	xhr
fontawesome-webfont.woff2?v=4.6.3	200	font
?EIO=3&transport=polling&t=NHs-gTD&sid=A3AJh9doHcUd8_yzAAAH	200	xhr
?EIO=3&transport=websocket&sid=A3AJh9doHcUd8_yzAAAH	101	websocket

54 requests | 14.9 kB transferred | 1.6 MB resources | Finish: 512 ms | DOMContentLoaded: 391 ms | Load: 408 ms

Gambar 11 Hasil tes halaman dokter menerima data

Lebih jelasnya terdapat pada tabel 3 yang menunjukkan hasil pengujian kecepatan halaman perawat mengirim data ke halaman dokter. Sedangkan pada tabel 4 adalah hasil pengujian kecepatan halaman dokter saat menerima data dari halaman perawat.

Tabel 3 Halaman Perawat mengirim data

Objek	Page size	Load Time	Request
Halaman Perawat	1.6 MB	322 ms	55

Tabel 4 Halaman Dokter menerima data

Objek	Page size	Load Time	Request
Halaman Dokter	1.6 MB	408 ms	54

Pada gambar 12, terlihat hasil tes saat halaman dokter mengirim data pada Chrome DevTools. Sedangkan pada gambar 13 adalah hasil tes saat halaman apoteker menerima data dari halaman dokter.

Name	Status	Type
moment.min.js	200	script
daterangepicker.js	200	script
custom.min.js	200	script
select2.min.js	200	script
select2.min.css	200	stylesheet
?EIO=3&transport=polling&t=NH0HAF	200	xhr
fontawesome-webfont.woff?v=4.6.3	200	font
?EIO=3&transport=polling&t=NH0HDH&sid=e44ypGcBPNSU3_yAAAK	200	xhr

55 requests | 15.3 kB transferred | 1.6 MB resources | Finish: 467 ms | DOMContentLoaded: 346 ms | Load: 361 ms

Gambar 12 Hasil tes halaman dokter mengirim data

Name	Status	Type
daterangepicker.js	200	script
custom.min.js	200	script
select2.min.js	200	script
select2.min.css	200	stylesheet
?EIO=3&transport=polling&t=NH0Hb5	200	xhr
fontawesome-webfont.woff?v=4.6.3	200	font
?EIO=3&transport=polling&t=NH0Hel&sid=x95MAid9kaySdWhgEAAAL	200	xhr
?EIO=3&transport=websocket&sid=x95MAid9kaySdWhgEAAAL	101	websocket

54 requests | 14.2 kB transferred | 1.6 MB resources | Finish: 501 ms | DOMContentLoaded: 384 ms | Load: 396 ms

Gambar 13 Hasil tes halaman apoteker menerima data

Lebih jelasnya terdapat pada tabel 5 yang menunjukkan hasil pengujian kecepatan halaman dokter mengirim data ke halaman apoteker. Sedangkan pada tabel 6 adalah hasil pengujian kecepatan halaman apoteker saat menerima data dari halaman dokter.

Tabel 5 Halaman Dokter mengirim data

Objek	Page size	Load Time	Request
Halaman Dokter	1.6 MB	361 ms	55

Tabel 6 Halaman Apoteker menerima data

Objek	Page size	Load Time	Request
Halaman Apoteker	1.6 MB	396 ms	54

Berdasarkan pengujian performa dengan melihat kecepatan sistem saat mengirim dan menerima data menggunakan Chrome DevTools. *Load time* dari keseluruhan sistem yang diuji membutuhkan waktu memuat halaman dibawah 1 detik. Dengan hasil tersebut sistem dapat digunakan dengan cepat.

IV. KESIMPULAN

Adapun kesimpulan dari penelitian ini yang dapat diambil adalah dengan adanya sistem berbasis *website* serta teknologi *websocket*, penggunaan sistem secara *real time* dapat mempermudah Staf Puskesmas Pallangga dalam pelayanan Pasien. Dikarenakan data yang telah diinput oleh staf akan secara langsung tampil pada daftar antrian tanpa melakukan *request data* terlebih dahulu. Pasien yang datang berobat lebih cepat dilayani dikarenakan staf tidak perlu lagi mencari buku rekam medis pasien pada kamar buku.

DAFTAR PUSTAKA

- Basiroh dan Auliya Burhanuddin. (2016). *Sistem Informasi Pengelolaan Data Pasien Rawat Jalan Pada Klinik Akupuntur dan Home Care Sehat Migoenani*.
- Deviana, Prista Avinda. (2018). *Sistem Informasi Rawat Jalan Pada Puskesmas*.
- Saputro, Yayan Bagus. (2017). *Sistem Rekam Medis Rawat Jalan Pada Puskesmas Pembantu Sukorejo Di Kabupaten Nganjuk*.