

Intelligent Maritime Transportation System: Visualisasi Data Kapal Berbasis AIS Menggunakan Peta Daring

Shidqon Famulaqih, Christyowidiasmoro, ST., MT., dan Dr. I Ketut Eddy Purnama, ST., MT.
Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111
E-mail: shidqon11@mhs.ee.its.ac.id

Abstrak— Penelitian Intelligent Maritime Transportation System (IMTS) dilakukan untuk mengatasi masalah kemaritiman di Indonesia. Diantaranya adalah pencurian ikan oleh kapal yang tidak terdeteksi dan tidak adanya informasi publik terkait kapal yang sedang beroperasi di Indonesia. Salah satu fungsi IMTS adalah sebagai information provider yang menyediakan segala informasi yang dibutuhkan oleh pelaku kemaritiman seperti pihak pelabuhan, perusahaan perkapalan, perusahaan ekspedisi, dan Search and Rescue (SAR). Informasi yang diberikan berupa informasi kapal dari AIS yang didapatkan melalui sistem penerima data AIS dan hubungannya dengan pelabuhan yang ada. Karena itu diperlukan modul visualisasi untuk IMTS agar bisa menerima dan menyimpan data dari sistem penerima data AIS dan menampilkannya agar bisa diakses oleh semua pelaku kemaritiman. Tugas akhir ini dilakukan untuk membuat aplikasi database berbasis website untuk melakukan visualisasi data dari IMTS menggunakan peta daring.

Kata kunci—Visualisasi Data, Automatic Identification System (AIS), Geographical Information System (GIS), Google Maps API

I. PENDAHULUAN

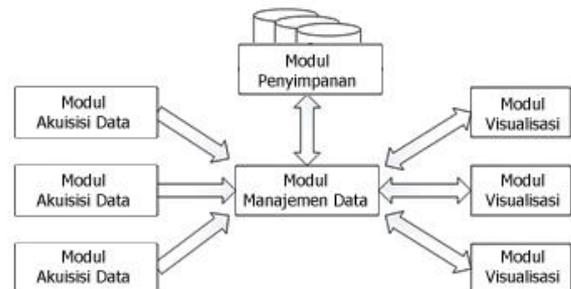
Penelitian Intelligent Maritime Transportation System (IMTS) dilakukan untuk mengatasi masalah kemaritiman di Indonesia. Diantaranya adalah pencurian ikan oleh kapal yang tidak terdeteksi dan tidak adanya informasi publik terkait kapal yang sedang beroperasi di Indonesia [1]. IMTS memiliki empat fungsi utama yaitu, monitoring, controlling, enforcement dan information provider. Monitoring adalah pemantauan gerak kapal, baik yang memiliki ijin maupun yang tidak memiliki ijin. Controlling adalah untuk mengatur jalur yang dapat dan yang tidak dapat dilalui kapal. Enforcement untuk penegakan hukum terkait kapal yang melakukan pelanggaran. Sedangkan information provider yaitu sebagai penyedia segala informasi yang dibutuhkan oleh pelaku kemaritiman seperti pihak pelabuhan, perusahaan perkapalan, perusahaan ekspedisi, dan Search and Rescue (SAR).

IMTS menggunakan sistem penerima data AIS yang disebar diseluruh kota di area pesisir di Indonesia untuk mendapatkan data kapal. Sebagai information provider untuk menampilkan informasi dari data tersebut, diperlukan modul yang dapat menerima dan menyimpan data dari semua sistem penerima dan menampilkannya agar bisa diakses oleh seluruh pelaku kemaritiman. Aplikasi yang umum untuk modul seperti ini adalah aplikasi database berbasis website.

Data IMTS terdiri dari data kapal dari AIS dan data tentang pelabuhan. Data tersebut ada banyak dan kompleks membuat informasi yang diinginkan sulit dimengerti jika ditampilkan apa adanya. Teknik penampilan data yang tidak sesuai dengan jenisnya juga mengakibatkan tingkat pemahaman pengguna kurang maksimal. Oleh karena itu bagaimana data ditampilkan sesuai jenisnya sangat mempengaruhi pengguna menyerap informasi dari data tersebut. Data kapal dari AIS berisi informasi umum kapal, rute, lokasi, dan navigasi kapal [3]. Sedangkan data pelabuhan yang ditampilkan berisi nama, negara, kode, dan lokasi pelabuhan [4]. Data-data ini berkaitan dengan lokasi bisa dikategorikan sebagai data spasial. Data spasial biasanya disimpan sebagai koordinat dan topologi, dan merupakan data yang dapat dipetakan. Data spasial sering diakses, dimanipulasi atau dianalisis menggunakan Geographic Information Systems (GIS). Salah satu contoh aplikasi dari GIS adalah peta daring Google Maps. Dengan Google Maps API data spasial bisa divisualisasi dengan jelas [2]. Data lokasi bisa ditandai dengan sebuah penanda (marker). Rute perjalanan bisa digambarkan sebagai garis vektor berdasarkan perpindahan posisi marker.

II. DESAIN DAN IMPLEMENTASI SISTEM

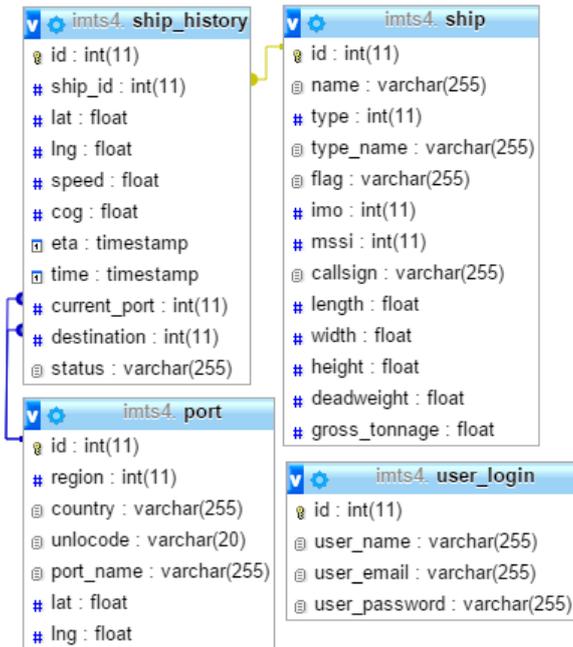
IMTS memiliki 4 komponen utama, yaitu modul akuisisi data, modul menyimpan data, modul manajemen data, dan modul visualisasi data yang ditunjukkan oleh gambar 1. Beberapa modul akuisisi data dapat menyediakan data ke modul manajemen. Semua data dapat disimpan dalam satu atau beberapa modul penyimpanan untuk mendukung skalabilitas dan redundansi. Modul visualisasi berfungsi untuk melihat dan memantau semua data yang telah dikumpulkan dan diproses. Modul visualisasi ini adalah aplikasi berbasis *website* yang dapat dibuka menggunakan *web browser*. Modul Akuisisi dan modul visualisasi terhubung dengan modul manajemen data melalui API untuk mengakses data dari modul penyimpanan.



Gambar 1. Komponen Utama IMTS

A. Modul Penyimpanan

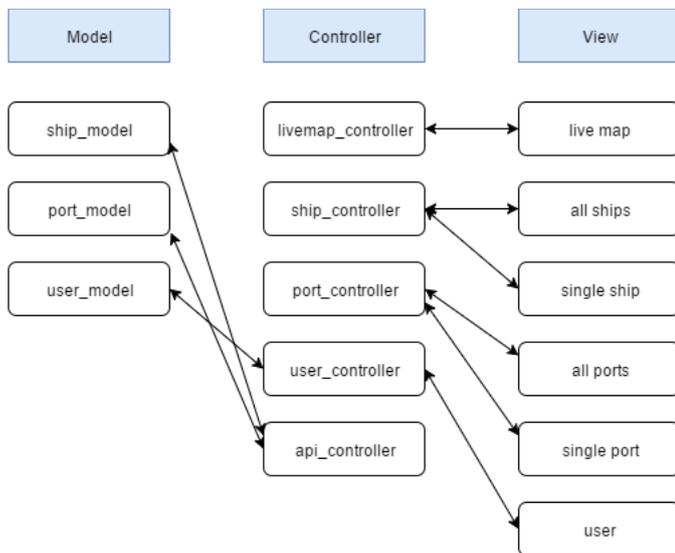
Data IMTS adalah data dari AIS yang terpasang di kapal dan data port. Mengacu dari data tersebut *database* didesain dalam bentuk relasi diagram yang ditunjukkan gambar 2. *Database* yang digunakan untuk desain ini adalah MySQL.



Gambar 2. Relasi diagram *database*

B. Modul Manajemen dan Visualisasi Data

Modul manajemen data dibuat dengan struktur *Model View Controller* (MVC) yang ditunjukkan oleh gambar 3. *Model* dibagi menjadi 3, yaitu *model* untuk kapal, port dan pengguna. *Controller* dibagi menjadi 4, yaitu *controller* untuk kapal, port, pengguna, dan API. Sedangkan *view* dibagi menjadi 7, yaitu *view* untuk halaman *live map*, *all ships*, *all ports*, *single ship*, *single port* dan *user*

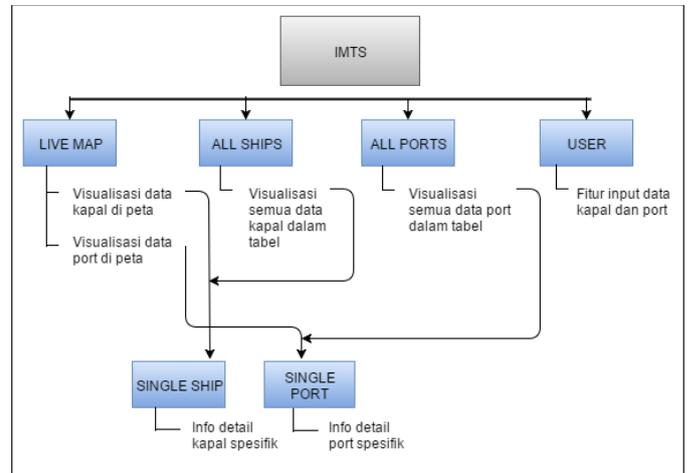


Gambar 3. Struktur MVC modul manajemen data

‘Ship_model’ bertanggung jawab melakukan *query* untuk data kapal. ‘Port_model’ bertanggung jawab melakukan *query* untuk data kapal. Sedangkan ‘user_model’ bertanggung jawab melakukan *query* data akun pengguna.

‘Livemap_controller’ bertanggung jawab mengatur kapan halaman ‘live map’ ditampilkan. ‘Ship_controller’ bertanggung jawab mengatur kapan halaman ‘all ships’ dan ‘single ship’ ditampilkan. ‘Port_controller’ bertanggung jawab mengatur kapan halaman ‘all ports’ dan ‘single port’ ditampilkan. ‘User_controller’ bertanggung jawab menentukan kapan halaman ‘user’ ditampilkan serta sebagai jembatan data dari ‘user_model’. Sedangkan ‘api_controller’ bertanggung jawab sebagai *Representational State Transfer* (REST) API untuk pertukaran data kapal dan port dengan program lain atau pihak ketiga.

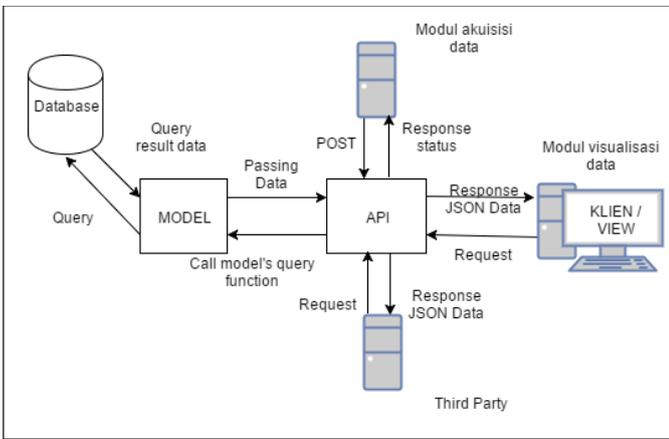
Halaman ‘live map’ bertanggung jawab menampilkan data kapal dan port dalam peta. Halaman ‘all ships’ bertanggung jawab menampilkan data semua kapal dalam bentuk tabel untuk keperluan pencarian kapal. Halaman ‘single ship’ bertanggung jawab menampilkan data lengkap dari suatu kapal tertentu. Halaman ‘all ports’ bertanggung jawab menampilkan data semua port dalam bentuk tabel untuk keperluan pencarian port. Halaman ‘single port’ bertanggung jawab menampilkan data lengkap dari suatu port tertentu dan daftar kapal yang sedang berada di, berangkat dari, baru saja tiba di, dan sedang menuju ke port tersebut. Halaman ‘user’ bertanggung jawab menampilkan halaman yang memberikan pengguna hak akses untuk melakukan penambahan atau perubahan data kapal dan port setelah melakukan autentikasi akun. Alur penampilan *view* ditunjukkan oleh gambar 4.



Gambar 4. Bagian dari *view*

C. Desain REST API

API dibagi menjadi 2 yaitu API untuk data kapal dan API untuk data port. Prinsip kerja dari API ini adalah ketika API ini dipanggil ‘api_controller’ akan menjalankan fungsi *query* data yang ada pada ‘model_ship’ atau ‘model_port’ lalu memberikan response berupa hasil *query* yang dikonversi menjadi format JSON. Masing-masing api ditentukan parameter yang diperlukan untuk mengambil data maupun menambah data.



Gambar 5. Alur request / response REST API

REST API bisa dipanggil menggunakan metode HTTP (POST, GET, PUT, DELETE) melalui URL dengan parameter yang ditentukan. Pemanggilan dengan metode POST pada API kapal dibedakan menjadi 2 yaitu untuk data spesifikasi kapal dan data riwayat perjalanan kapal. Sedangkan untuk API port hanya untuk data spesifikasi port. Pemanggilan dengan metode GET baik untuk API kapal maupun port dibagi berdasarkan parameter ID kapal /port, area yang tampil di peta (*viewport*).

```
http://server address/api/ships?
http://server address/api/ports?
```

Tabel 1. Parameter POST data spesifikasi kapal

Parameter	Nilai	Description
data	1	POST untuk data spesifikasi kapal
name	string	Nama kapal
imo	integer	IMO
mssi	integer	MSSI
callsign	string	Callsign
flag	string	Kode negara
type	string	Tipe kapal
deadweight	float	Deadweight
gross_tonnage	float	Gross Tonnage

Tabel 2. Parameter POST data riwayat kapal

Parameter	Nilai	Description
history	1	POST untuk data riwayat kapal
id	integer	ID unik kapal
lat	float	Latitude kapal
lng	float	Longitude kapal
speed	float	Kecepatan kapal
current_port	integer	ID unik port tempat kapal berada
destination	integer	ID unik port tujuan kapal

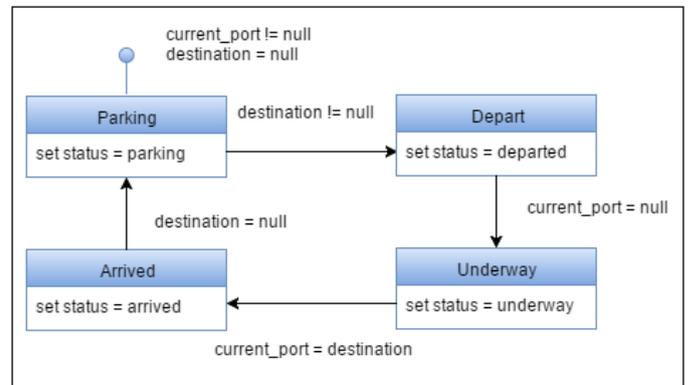
Tabel 3. Parameter GET data kapal atau port

Parameter	Nilai	Description
id	integer	ID unik kapal / port
lat1	float	Posisi <i>viewport</i> peta
lat2	float	
lng1	float	
lng2	float	
portcall (khusus port)	"ship_in", "arrivals", "departures", "expected_arrivals"	Status kapal terhadap port

Contoh *response*:

```
[{
  "id" : integer,
  "name" : string,
  "lat" : float,
  "lng" : float,
  // data kapal / port lain
}, ...]
```

Pada saat POST data kapal beberapa data diolah terlebih dahulu sebelum disimpan ke *database*. Data tersebut adalah data sudut kapal terhadap arah utara (*course*), status pergerakan kapal, dan status posisi kapal berada di port mana (*current port*). Data *course* dan status pergerakan kapal sudah ada dari data yang dikirim AIS. Pengolahan ulang ini dimaksudkan sebagai antisipasi jika sensor AIS rusak dan tidak mengirim data tersebut maka data dari hasil perhitungan yang akan digunakan. Data *course* didapatkan dari perhitungan posisi sekarang dengan posisi sebelumnya. Data status kapal didapat dari data tujuan kapal dan *current port*. Sedangkan *current port* didapat dari *query* data port dengan parameter *viewport* dengan radius tertentu dimana titik tengah adalah posisi terbaru kapal.

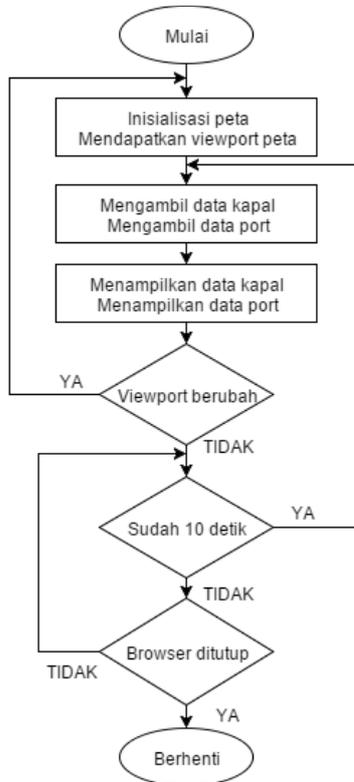


Gambar 6. State diagram status kapal

D. Desain Visualisasi Data

Ketika peta dibuka akan didapatkan *viewport* dari peta, lalu dari dilakukan *query* data berdasarkan parameter *viewport* tersebut untuk mendapatkan data kapal dan port. Kemudian data ditampilkan pada peta. Ketika peta digeser maka *viewport* peta juga berubah, maka dilakukan *query* data lagi untuk mengambil

data pada area peta yang baru. Data secara otomatis akan di muat ulang setiap 10 detik untuk mendapatkan data terbaru.



Gambar 7. Alur visualisasi data pada peta

Kapal divisualisasikan menggunakan *marker* berupa ikon kapal, sedangkan port dengan ikon jangkar. Warna *marker* kapal berbeda-beda sesuai jenis kapal. Ketika pada suatu area terdapat beberapa *marker* dengan posisi yang terlalu berdekatan maka akan dilakukan pengelompokan pada *marker* tersebut menjadi *cluster marker* dengan info jumlah *marker* yang ada dalam *cluster* tersebut. Ketika *cluster* diklik secara otomatis peta akan diperbesar sampai tingkat dimana posisi *marker* dalam *cluster* tersebut tidak terlalu berdekatan. Sedangkan jalur kapal divisualisasikan dengan garis yang menghubungkan posisi-posisi sebelumnya.



Gambar 8. Marker Kapal, Port dan Cluster



Gambar 9. Visualisasi Jalur Kapal

III. PENGUJIAN DAN ANALISIS

Pengujian terdiri dari pengujian API dan pengujian visualisasi. Data yang diujikan adalah data dari antenna AIS yang dipasang pada koordinat 7°17'05"S, 112°47'45"E yang mencakup 66 data kapal yang diambil dalam kurun waktu 2 jam dengan penambahan beberapa *dummy* data.

A. Pengujian API

Pengujian ini dilakukan untuk mengetahui apakah API yang dirancang memberikan response yang diinginkan untuk setiap request yang dikirim. Pengujian dilakukan dengan memanggil API untuk request data untuk setiap parameter yang ada dan mengetahui besar ukuran data yang dimuat setiap mengambil data untuk ditampilkan pada live map. Hasil pengujian ditunjukkan oleh tabel 4, tabel 5 dan gambar 10.

Tabel 4. Pengujian API kapal

No	Parameter	Response
1	lat1 = -5.6350 lng1 = 114.7518 lat2 = -8.3609 lng2 = 107.2481	Menghasilkan respon 6 data kapal.
	lat1 = 2.0554 lng1 = 116.3256 lat2 = 0.4353 lng2 = 112.573	Menghasilkan respon <i>not found</i> untuk peta yang menampilkan daerah daratan di tengah pulau Kalimantan
2	lat1 = null lng1 = 104.7379 lat2 = 0.8567 lng2 = 102.8620	Menghasilkan respon status <i>error</i> karena ada parameter <i>viewport</i> yang kosong (lat1).
3	lat1 = 5.0825 lng1 = 121.5848 lat2 = abcd lng2 = 114.0811	Menghasilkan respon status <i>error</i> karena ada parameter <i>viewport</i> yang diisi string (lat2).
4	lat1 = 4.3654 lng1 = 190.6540 lat2 = -2.1117 lng2 = 115.6467	Menghasilkan respon status <i>error</i> karena ada parameter <i>viewport</i> yang melebihi jangkauan nilai latitude dan longitude (lng1).
5	id = 3380	Menghasilkan respon 1 data kapal
6	id = KRI	Menghasilkan respon status <i>error</i> karena parameter id diisi string
7	id = null	Menghasilkan status error karena parameter id kosong

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Dari hasil pengujian API dan visualisasi yang sudah dilakukan dapat ditarik beberapa kesimpulan sebagai berikut :

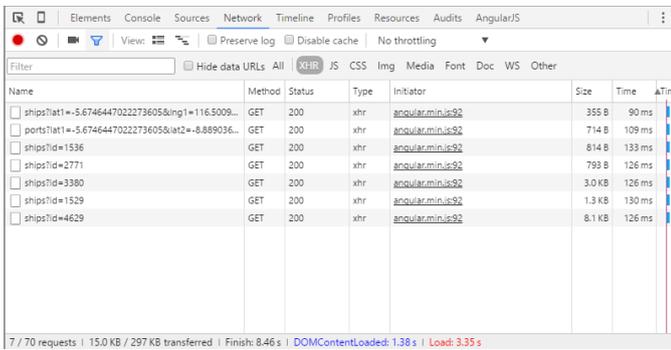
- 1) REST API memberikan *response* seperti yang didesain untuk setiap *request*. Rata-rata data yang dimuat untuk menampilkan halaman adalah 15K dimana request data pelabuhan rata-rata hanya 714B sedangkan data kapal 4,13KB.
- 2) Terdapat jeda waktu dengan rata-rata 4,92 detik dalam menampilkan data terbaru ketika ada data baru yang diterima.

B. Saran

- 1) Untuk mengurangi jeda waktu dalam menampilkan data terbaru, aplikasi bisa diatur untuk melakukan muat ulang data terbaru secara otomatis menjadi lebih cepat dari 10 detik. Namun hal ini akan memberatkan pengolahan disisi *client* sehingga diperlukan optimasi dibagian proses penampilan *view*.

DAFTAR PUSTAKA

- [1] Eddy Ketut. Pengembangan Intelligent Maritime Transportation System untuk Penegakan Kedaulatan Maritim Indonesia. Kompetitif Nasional – Pengembangan IPTEK, 2015, Indonesia.
- [2] Shunfu Hu, Ting Dai, Online Map Application Development Using Google Maps API, SQL Database, and ASP.NET, ICT Journal, 2013 International Journal of Information and Communication Technology Research on, vol. 3, no. 3, Mar. 2013.
- [3] U.S. Coast Guard Navigation Center, 2015., AIS Messages. <http://www.navcen.uscg.gov/?pageName=AISMessages>. Diakses pada tanggal 23 Januari 2016.
- [4] Pusat Data dan Informasi - Sekretariat Jenderal Kementerian Perhubungan - Republik Indonesia, 2010., Sistem Informasi Geografis Prasarana Transportasi. <http://gis.dephub.go.id/mapping/Prasarana/PelabuhanList.aspx>. Diakses pada tanggal 23 Januari 2016.

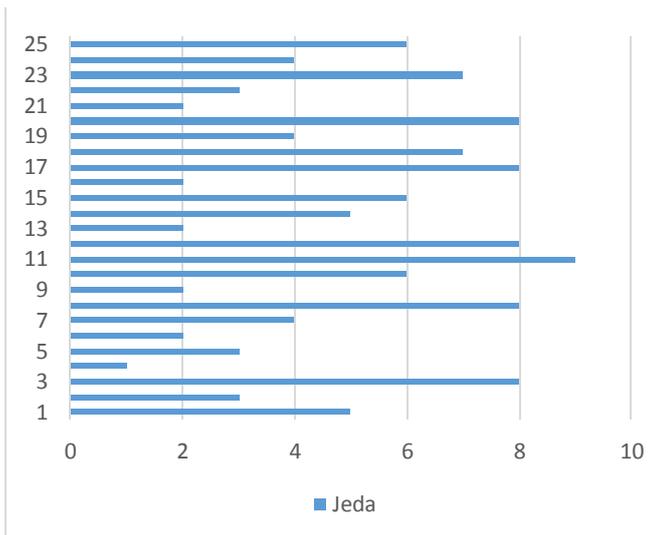


Name	Method	Status	Type	Initiator	Size	Time	Alt
ships?lat1=-5.6746447022273605&lng1=116.5009...	GET	200	xhr	angular.min.js?2	355 B	90 ms	
ports?lat1=-5.6746447022273605&lat2=-8.889036...	GET	200	xhr	angular.min.js?2	714 B	109 ms	
ships?id=1536	GET	200	xhr	angular.min.js?2	814 B	133 ms	
ships?id=2771	GET	200	xhr	angular.min.js?2	793 B	126 ms	
ships?id=3380	GET	200	xhr	angular.min.js?2	3.0 KB	126 ms	
ships?id=1529	GET	200	xhr	angular.min.js?2	1.3 KB	130 ms	
ships?id=4629	GET	200	xhr	angular.min.js?2	8.1 KB	126 ms	

Gambar 10. Besar data *response* dari API

B. Pengujian Visualisasi

Aplikasi ini didesain untuk mengambil data terbaru setiap 10 detik. Dengan begitu ada kemungkinan ketika data terbaru diterima, akan ada jeda penampilan data terbaru antara 1 – 10 detik. Pengujian ini dilakukan untuk mengetahui jeda rata-rata penampilan data terbaru pada peta daring. Pengujian dilakukan dengan melakukan POST data riwayat posisi kapal melalui API. Waktu yang dicatat adalah kapan data terbaru dikirim melalui API, kapan data terbaru diterima oleh *client* (*web browser*), dan kapan data terbaru ditampilkan pada peta. Pengujian dilakukan dengan 25 kali pengulangan.



Gambar 10. Jeda waktu penampilan data terbaru hasil pengujian