

Rancang Bangun Prototipe Sistem Kontrol Terdistribusi untuk Pemantauan dan Pengendali Ketinggian Permukaan Air pada 5 Pintu Air Berbasis *IoT (Internet of Things)* menggunakan *Socket.IO*

Zendi Iklima

Teknik Elektro, Universitas Mercu Buana, Jakarta

*zendi.iklima@mercubuana.ac.id

Abstrak—Pemantauan tinggi permukaan air pada sungai merupakan salah satu tindakan untuk mengurangi terjadinya banjir akibat kelebihan debit air sungai dan ketinggiannya. Pengembangan sistem pemantauan permukaan air sungai telah banyak dilakukan. Namun, sistem tersebut tidak saling terhubung antar pintu air yang saling berhubungan. Maka dari itu penelitian ini bertujuan untuk mengembangkan Sistem tersebut dalam pengembangan teknologi *IoT (Internet of Things)* yaitu menggunakan *workflow Socket* sebagai sistem kontrol terdistribusi. Dengan adanya pengembangan pada sistem tersebut diharapkan memberikan informasi yang akurat dan tepat waktu (*real-time*) mengenai status dari ketinggian permukaan air yaitu Siaga I, Siaga II, Siaga III dan Normal. Dengan sistem yang terpusat (*centralized Sistem*) maka sistem ini dapat melakukan pemantauan dan pengontrolan pintu air yang terintegrasi dalam suatu wilayah desa, kota, provinsi dan negara tertentu. Maka hasil pemantauan ketinggian permukaan air sungai pada 5 kanal diketahui memiliki respons 7.05ms dengan data berkapasitas 117B. Data yang diolah tersebut didistribusikan oleh server kesemua client yang terkoneksi didalam sistem dan mengirim notifikasi dengan *Google SMTP Server*. Sistem ini merekam semua data yang diambil dari sensor HC-SR04 secara *real-time* serta men-tigger untuk membuka pintu air (*gate servo MG90S*) dalam status tertentu sehingga disediakan sebuah *dashboard* untuk melakukan pemantauan dan pengendalian sistem terintegrasi pada pintu air.

Kata Kunci— *IoT, Sistem Kontrol Terdistribusi, Kanal, SMTP Server, HR-SC04, MG90S.*

DOI: 10.22441/jte.2020.v11i3.008

I. PENDAHULUAN

Perkembangan teknologi industri 4.0 memberikan dukungan yang signifikan terhadap sistem kendali jarak jauh dari segi kehandalan, ketepatan dan performa sistem. Pendekatan Sistem terintegrasi *IoT (Internet of Things)* menunjang kebutuhan user dalam melakukan pemantauan (*monitor*) serta pengendalian (*control*). *IoT* telah banyak diaplikasikan dalam beberapa sektor terutama industri [1][2] seperti Sistem otomasi, robotik, sistem keamanan, dan lain-lain [3][4][9]. DKI Jakarta merupakan ibu kota negara Indonesia yang kerap dilanda banjir pada musim penghujan tiap tahunnya. Pada tahun 2015 sampai 2017 tercatat 3540 Hektar wilayah ibu kota kerap tergenang banjir dalam rata-rata genangan 2 - 4 hari [10]. Hal ini menyebabkan gangguan

pada aktivitas masyarakat baik untuk mobilisasi, ekonomi dan lainnya. Hal ini disebabkan oleh sistem drainase tata kota yang belum berjalan dengan baik serta sistem pemantauan ketinggian permukaan air sungai yang belum terintegrasi sehingga pembukaan pintu air tidak diatur sedemikian hingga yang menyebabkan volume air naik kepermukaan sampai kepemukiman warga [5][6][7]. Berdasarkan permasalahan tersebut diperlukan suatu tindakan preventif untuk mengurangi permasalahan banjir tersebut yaitu suatu Sistem terdistribusi pemantauan dan pengendalian jarak jauh terhadap pintu air yang berada pada cakupan wilayah tertentu. Penelitian ini menggunakan 5 sampel pintu air yaitu A, B, C, D dan E perhatikan Tabel 1.

Tabel 1. Data Kanal [11][12]

Kanal	Water Distance (cm)			
	Siaga 1	Siaga 2	Siaga 3	Normal
1. Sunter Hulu	> 250	< 250	< 200	< 150
2. Pluit	> 45	< 45	< 0	< -50
3. Pasar Ikan	> 250	< 250	< 200	< 170
4. Tanjungan	> 300	< 300	< 250	< 225
5. Kali Duri	> 320	< 320	< 270	< 230

Tiap kanal memiliki ketinggian permukaan air dan status siaga yang berbeda. Adanya sistem terdistribusi pemantauan dan pengendalian jarak jauh terhadap 5 pintu air di wilayah ibu kota diharapkan dapat mengintegrasikan semua Sistem pemantauan dan pengendalian pintu air jarak jauh dalam waktu yang bersamaan. *Socket.IO* merupakan suatu jaringan komunikasi dibawah TCP layer yang memungkinkan server melakukan komunikasi *bidirectional*. Sehingga penelitian ini dibuat berdasarkan *workflow Socket.IO*. Selanjutnya dibuat dashboard untuk dapat melakukan pemantauan dan pengendalian secara *real-time* untuk mempermudah admin dalam mengambil tindakan terhadap status siaga masing-masing pintu air / kanal.

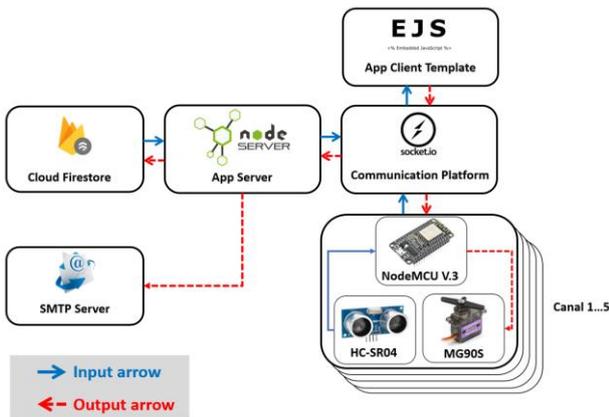
II. PENELITIAN TERKAIT

Prototipe Sistem monitoring ketinggian air dan pengendalian pintu air pada jaringan irigari telah berhasil dengan menggunakan *Short Message Service (SMS)*. Utomo dkk [5], menggunakan potensiometer sebagai parameter untuk mensimulasikan nilai pembacaan sensor jarak. Sehingga

perubahan pelampung pada pintu air dianggap dapat merubah resistensi potensiometer. Perubahan nilai potensiometer ini dan selanjutnya dikirim sebagai notifikasi melalui SMS ke HP user. Kemudian, Fikri [6] melakukan monitoring ketinggian permukaan air menggunakan mikrokontroler ATMEGA328P yang diintegrasikan kedalam Sistem dengan mengembangkan teknologi *client-server* dan *webservice*. Secara perangkat penelitian ini berhasil membuktikan keakuratan perangkatnya sebesar 96,48% yang dapat diakses secara *online* dan *real-time*. sistem irigasi terintegrasi kemudian dikembangkan dengan melakukan sistem buka tutup pada sebuah bendungan dengan melihat simulasi aliran sungai yang bercabang berhasil dilakukan oleh Rosyid [7].

III. METODOLOGI PENELITIAN

Adapun secara menyeluruh *input-ouput* yang diproses pada Sistem seperti Gambar 1 dimana sistem terdistribusi dapat melakukan pemantauan dan pengendalian 5 pintu air / kanal secara *real-time*. Alat yang dipasang pada masing-masing kanal yaitu NodeMCU v.3, sensor jarak HC-SR04, dan actuator gate servo MG90S.

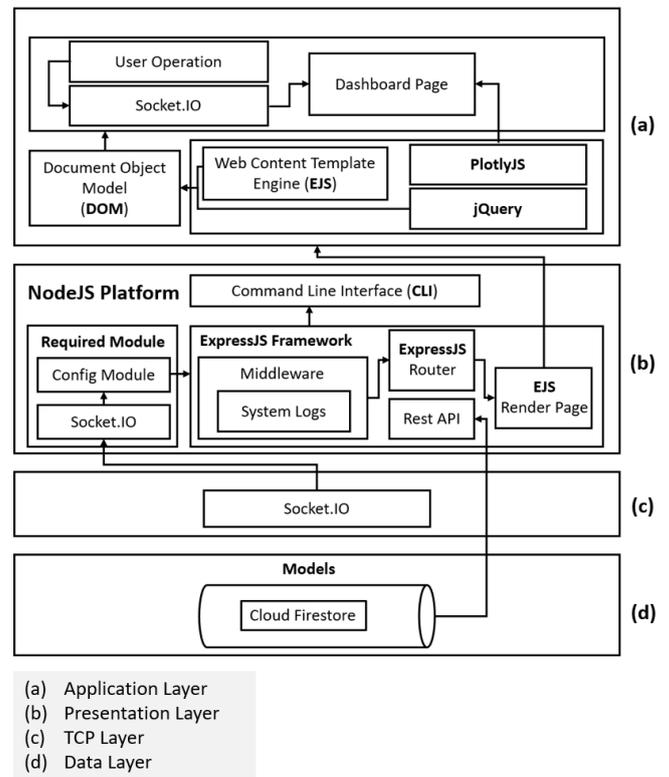


Gambar 1. Diagram Blok Sistem [12][13]

Perangkat pada kanal akan mengirimkan data jika terdapat perubahan data ke *socket server* kemudian mendistribusikan data keseluruhan client yang terkoneksi didalam Sistem. Selanjutnya perubahan status masing-masing kanal akan dikirim melalui email setiap data sensor berubah dan akan mengirimkan data rekap setiap 30 menit.

A. Arsitektur Sistem

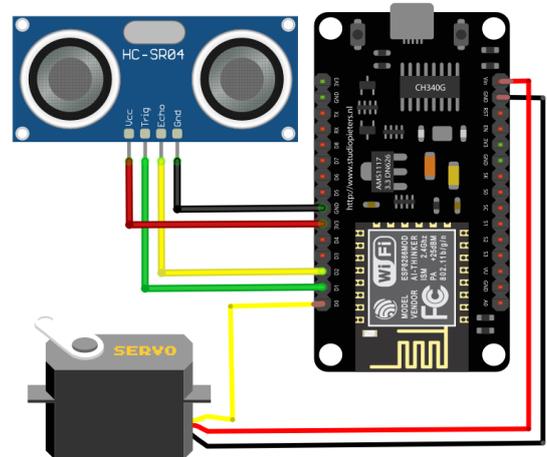
Sistem terdistribusi untuk pemantuan dan pengendalian jarak jauh kanal ini dibangun dalam *layer* yang utama yaitu application layer, presentation layer, *TCP layer* dan data layer. *ExpressJS* sebagai server dari aplikasi *dashboard* mengatur semua *event* yang diatur untuk kerangka kerja dari *Socket.IO* sehingga setiap *event* ada harus diatur pada tiap perangkat seperti *client app*, client NodeMCU dan server. Selanjutnya data yang diterima oleh server akan disimpan kedalam *cloud firestore*. Dengan dukungan *service cloud firestore snapshot* memungkinkan server melakukan integrasi data secara *real-time* berdasarkan *event* yang ada pada *Socket.IO*. perhatikanlah arsitektur sistem terdistribusi untuk pemantauan dan pengendalian 5 pintu air / kanal seperti Gambar 2.



Gambar 2. Arsitektur Sistem Terdistribusi [14]

B. Wiring Diagram

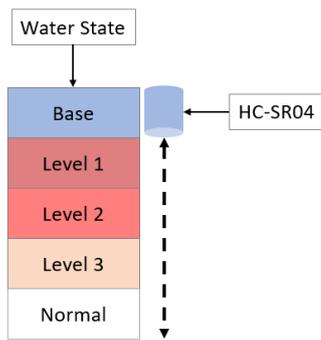
Perangkat yang dipasang pada tiap kanal sangatlah sederhana yang terdiri dari NodeMCU V.3, sensor jarak HC-SR04 dan motor servo MG-90S seperti Gambar 3 dibawah ini [6].



Gambar 3 Wiring Diagram Perangkat pada Kanal

C. Instalasi Kanal

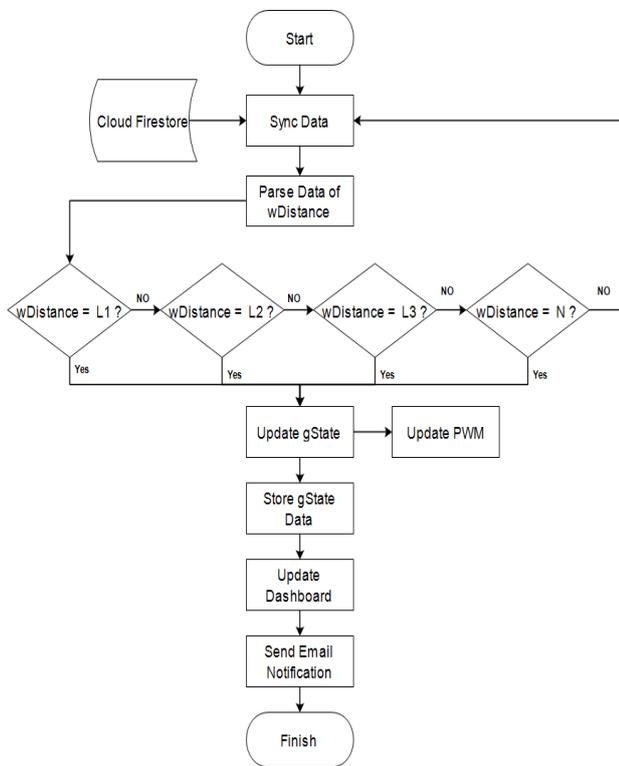
Berdasarkan Tabel 1 maka instalasi perangkat pada tiap kanal dipasang pada base dimana sensor HC-SR04 dapat membaca ketinggian air sungai seperti Gambar 4.



Gambar 4. Instalasi Perangkat pada Kanal

D. Diagram Alir Penelitian

Sistem terdistribusi untuk pemantauan dan pengendalian berbasis IoT pada kanal saling terintegrasi. Semua data pembacaan ketinggian air dipusatkan pada cloud firestore sehingga Sistem akan menampilkan data yang actual untuk memicu tindakan yang dilakukan oleh admin untuk mengatur aliran sungai. Sistem ini memiliki 4 status kanal yaitu Siaga 1 (L1), Siaga 2 (L2), Siaga 3 (L3) dan Normal. Perhatikan Gambar 5.

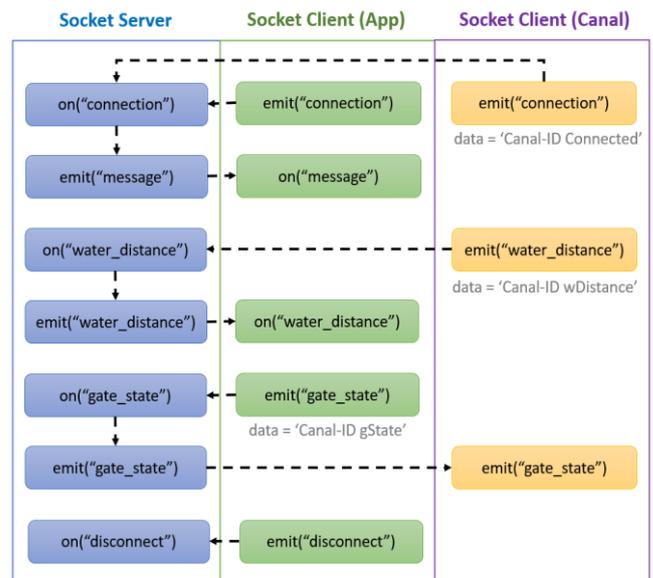


Gambar 5. Diagram Alir Sistem

Sistem ini akan menampilkan data pembacaan tinggi permukaan air sungai per milidetik. Sementara itu, sistem akan mengirimkan report pembacaan ketinggian air sungai tiap 30 menit via email. Admin akan selalu mendapatkan notifikasi ketika tinggi permukaan air sungai pada status Siaga tiap kanal.

E. Workflow Socket.IO

Socket.IO menjadi actor utama agar Sistem terdistribusi dapat berjalan dengan baik. Terdapat 2 event didalam Socket.IO yaitu onListening dan onEmit [11]. Event onListening merupakan suatu event dimana menunggu data yang didistribusikan melalui socket event. Event ini akan menerima setiap aktivitas event onEmit. Event onEmit merupakan event dimana data dikirim melalui socket event. Maka dari itu pada penelitian ini Socket.IO diinstal pada 3 sisi yaitu sisi server, client app dan client canal. Dimana untuk dapat saling terintegrasi setiap event harus dibuat secara berpasangan onListening dan onEmit. Event yang dibuat pada Sistem ini terdiri dari 2 event yaitu event ketika sensor HC-SR04 mengirimkan data (onEmit) yang diberi label water_distance. Event yang kedua dibuat ketika admin melakukan kendali terhadap gerbang kanal baik menutup atau membukanya. Event ini diberi label gate_state dimana bergantung pada status siaga pada tiap kanal. Perhatikan Gambar 6 workflow Socket.IO.

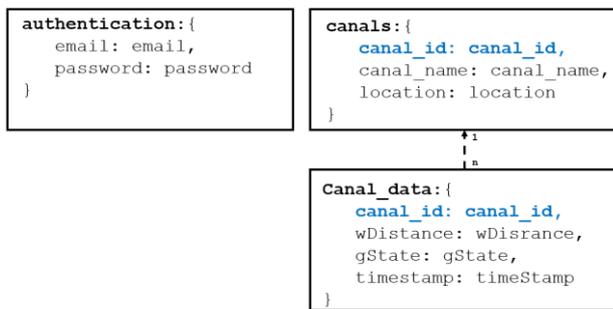


Gambar 6. Workflow Socket.IO

Sistem ini memiliki 5 kanal yang berbeda sehingga setiap data yang dikirim kedalam event socket harus memiliki identitas kanal guna membedakan kanal dengan identitas terkait pada status siaga tertentu.

F. Cloud Firestore Collection

Gambar 2 menampilkan dimana data layer menjadi bagian pada arsitektur sistem terdistribusi ini. Maka dari itu untuk menunjang workflow pada Gambar 3 diperlukan integrasi antar collection pada cloud firestore sehingga data pada tiap kanal disimpan dalam bentuk unstructured database dengan relasi canal_id seperti Gambar 7.



Gambar 7. Cloud Firestore Collection

Identitas tiap kanal akan disimpan dengan pengaturan awal yaitu 5 kanal dengan entitas *canal_id*, *canal_name* dan *location*.

IV. HASIL DAN ANALISA

Pengujian sisystem terdistribusi untuk pemantauan dan pengendalian jarak jauh pada kanal dilakukan perkanal sehingga dapat diketahui respon dari sensor HC-SR04, kapasitas data dan respon dari *socket event*. Selanjutnya tindakan yang dilakukan pada masing-masing status adalah merubah *output* pada servo MG90S untuk dapat membuka pintu air pada PWM (0 - 255) seperti tabel 2.

Tabel 2. Trigger Status Kanal terhadap nilai PWM

Canal State	Output PWM (%)	Actual Value (angle)
Siaga 1	Open 100%	180°
Siaga 2	Open 75%	135°
Siaga 3	Open 35%	63°
Normal	Open 5%	9°

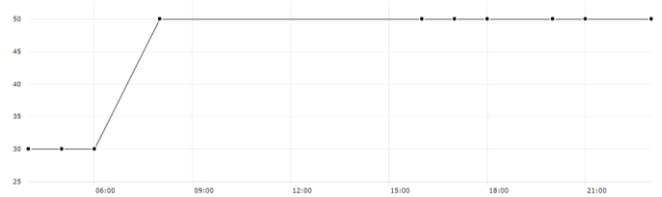
A. Pengujian Sistem Terdistribusi Tiap Kanal

Pengujian Sistem pada kanal 1 (kanal Sunter Hulu) dilakukan secara variabel acak dimana akan didapatkan status siaga pada kanal tersebut. Perhatikan tabel hasil pengujian Sistem seperti pada tabel 3.

Tabel 3. Pengujian Sistem pada Kanal 1 (Sunter Hulu)

No	Water Level (cm)	Canal State	Data Size (B)	Time Response (ms)
1	324	Siaga 1	124	7
2	281	Siaga 2	117	6
3	242	Siaga 3	117	6
4	178	Normal	112	8

Rata-rata respons data yaitu kapasitas data dan respon waktu yang masing-masing memiliki nilai 117.5 B dalam 6.75 ms. Nilai pembacaan sensor HC-SR04 dilakukan dalam rentang waktu tertentu dan data tersebut disimpan kedalam *database* sesuai Gambar 7 dimana *relation* antar *collection* sehingga di-generate kedalam sebuah grafik menggunakan plotlyJS seperti Gambar 8.



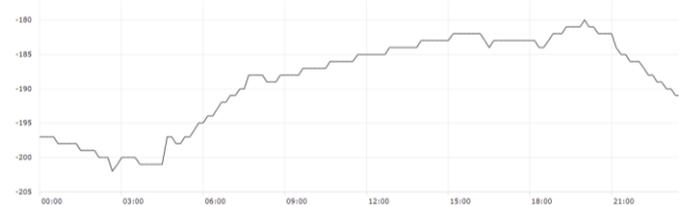
Gambar 8. Dashboard pada Kanal 1 (Sunter Hulu)

Pengujian sistem pada kanal 2 (kanal Sunter Hulu) diperoleh rata-rata respons data dengan kapasitas data 117.25 B dalam 7.75 ms seperti pada tabel 4.

Tabel 4. Pengujian Sistem pada Kanal 2 (Pluit)

No	Water Level (cm)	Canal State	Data Size (B)	Time Response (ms)
1	48	Siaga 1	123	8
2	33	Siaga 2	117	7
3	-8	Siaga 3	115	7
4	-51	Normal	114	9

Nilai pembacaan sensor HC-SR04 pada Kanal 2 yang dilakukan dalam rentang waktu tertentu seperti Gambar 9.



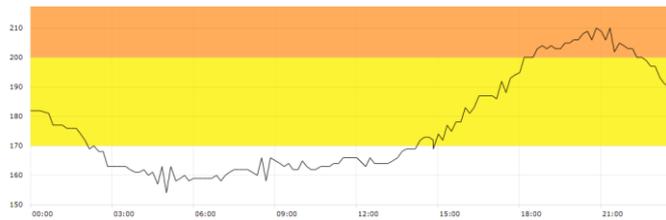
Gambar 9. Dashboard pada Kanal 2

Pengujian sistem pada kanal 3 (kanal Sunter Hulu) diperoleh rata-rata respons data dengan kapasitas data 116.75 B dalam 6.75 ms seperti pada tabel 5.

Tabel 5. Pengujian Sistem pada Kanal 3 (Pasar Ikan)

No	Water Level (cm)	Canal State	Data Size (B)	Time Response (ms)
1	257	Siaga 1	115	7
2	231	Siaga 2	116	6
3	173	Siaga 3	121	7
4	154	Normal	115	7

Nilai pembacaan sensor HC-SR04 pada Kanal 3 yang dilakukan dalam rentang waktu tertentu seperti Gambar 10.



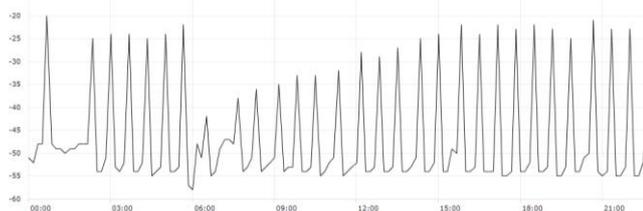
Gambar 10. Dashboard pada Kanal 3

Pengujian Sistem pada kanal 3 (kanal Sunter Hulu) diperoleh rata-rata respons data dengan kapasitas data 117.5 B dalam 6.75 ms seperti pada tabel 6.

Tabel 6. Pengujian Sistem pada Kanal 4 (Tanjungan)

No	Water Level (cm)	Canal State	Data Size (B)	Time Response (ms)
1	324	Siaga 1	124	7
2	281	Siaga 2	117	6
3	242	Siaga 3	117	6
4	178	Normal	112	8

Nilai pembacaan sensor HC-SR04 pada Kanal 4 yang dilakukan dalam rentang waktu tertentu seperti Gambar 11.



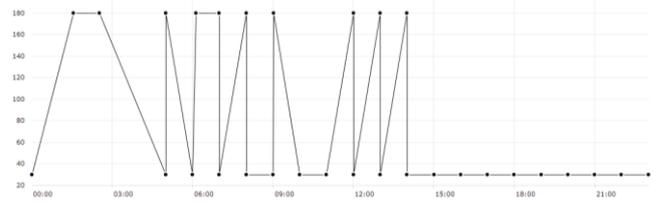
Gambar 11. Dashboard pada Kanal 4

Pengujian Sistem pada kanal 3 (kanal Sunter Hulu) diperoleh rata-rata respons data dengan kapasitas data 116 B dalam 7.25 ms seperti pada tabel 7 dibawah ini.

Tabel 7. Pengujian Sistem pada Kanal 5 (Kali Duri)

No	Water Level (cm)	Canal State	Data Size (B)	Time Response (ms)
1	333	Siaga 1	112	7
2	281	Siaga 2	117	7
3	254	Siaga 3	119	8
4	212	Normal	116	7

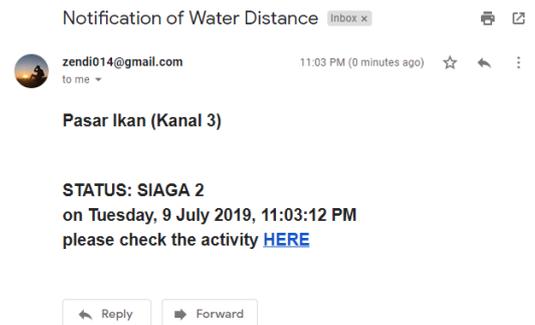
Nilai pembacaan sensor HC-SR04 pada Kanal 5 yang dilakukan dalam rentang waktu tertentu seperti Gambar 12.



Gambar 12. Dashboard pada Kanal 5

B. Pengujian Sistem Notifikasi

Notifikasi diperoleh ketika status pada kanal berada pada kondisi Siaga 3, Siaga 2 dan Siaga 3. Notifikasi ini akan dikirimkan ketika perubahan status terjadi serta dalam interval waktu 30 menit.

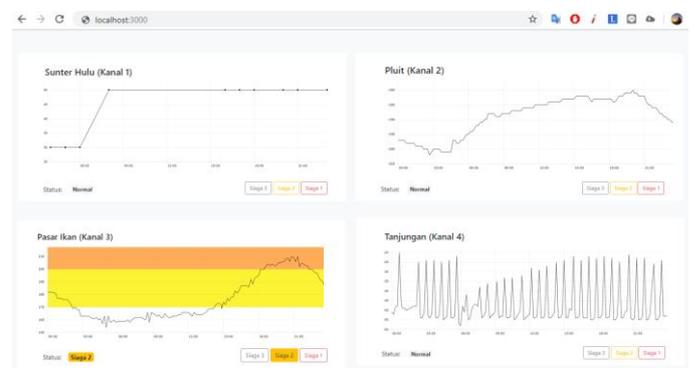


Gambar 13. Pengujian Notifikasi Sistem menggunakan SMTP Server

Data yang dikirimkan berupa identitas kanal, status kanal, catatan waktu serta link untuk melihat dashboard semua aktivitas kanal.

C. Dashboard Sistem

Berdasarkan Gambar 8 sampai Gambar 12 maka tampilan dashboard dapat diakses pada server localhost dimana menampilkan seluruh aktivitas kanal dalam sebuah halaman web seperti Gambar 14.



Gambar 14. Dashboard Sistem Terdistribusi

V. KESIMPULAN

Prototype Sistem Terdistribusi untuk pemantauan dan pengendalian jarak jauh telah berhasil diimplementasikan dengan menggunakan skema event pada Socket.IO untuk selanjutnya disimpan kedalam cloud firestore. Respons Sistem

terhadap pembacaan sensor HC-SR04 sebesar 117B dalam 7.05ms. selanjutnya nilai pembacaan dapat dilihat dalam sebuah halaman web yang menampilkan semua aktivitas kanal seperti pada Gambar 14. Pengembangan sistem dapat dilakukan mengimplementasikan Artificial Intelligence seperti ANN sehingga pintu air akan bekerja sesuai ketinggian air sungai.

DAFTAR PUSTAKA

- [1] Y. Gunardi, A. Adriansyah and T. Anindhito. "Small Smart Community: An Application Of Internet Of Things". 2015. ISSN 1819-6608.
- [2] N. Witthayawiroj and P. Nilaphruek, "The Development of Smart Home Sistem for Controlling and Monitoring Energy Consumption using WebSocket Protocol", 2017.
- [3] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego and J. Alonso-Zarate, "A Survey on Application Layer Protocols for the Internet of Things Transaction on IoT and Cloud Computing", 2015.
- [4] S. Dey, A. Roy and S. Das, "Home automation using Internet of Thing", 2017
- [5] B. T. W. Utomo and H. Hasan. "Prototiping Sistem Monitoring Ketinggian Air Dan Pengendalian Pintu Air Pada Jaringan Irigasi Berbasis Mikrokontroler Atmega16 Dengan Menggunakan Short Message Service (Sms)", V. 8, N. 1, P. 59-68, 2014. Issn 2580-8397
- [6] F. Rausan, B. P. Lapanporo and M. I. Jumarang, "Rancang Bangun Sistem Monitoring Ketinggian Permukaan Air Menggunakan Mikrokontroler ATMEGA328P Berbasis Web Service", vol.5, no.2., 2015
- [7] M. R. Alfatah and B. Hari, "Prototype Sistem Buka Tutup Otomatis Pada Pintu Air Bendungan Untuk Mengatur Ketinggian Air Berbasis Arduino", 2016 .
- [8] M .M. Khoir, "Rancang bangun alat monitoring pasang surut air laut berbasis internet of thing (IoT)", 2018.
- [9] A. Adriansyah, S. Budiyanto, J. Andika, A. Romadlan, and N. Nurdin, "Public street lighting control and monitoring system using the internet of things," *AIP Conf. Proc.*, vol. 2217, no. April, 2020, doi: 10.1063/5.0000594
- [10] <https://data.go.id>
- [11] <https://bpbd.jakarta.go.id>
- [12] <http://poskobanjirdsda.jakarta.go.id/>
- [13] <https://socket.io>
- [14] <https://nodejs.org/>
- [15] <https://blogs.cisco.com>