

**ID: 06**

## Kinerja Komunikasi Data Pada Model Taman Cerdas

### Data Communication Performance in Smart Garden Model

**Eda Mini Agnesia<sup>1</sup>, Damar Widjaja<sup>2\*</sup>**

<sup>1,2</sup>Universitas Sanata Dharma, Yogyakarta

Kampus III Paingan, Maguwoharjo, Depok, Sleman, Telp. 08112661144 ext. 52341

edamagnesia@gmail.com<sup>1</sup>, damar@usd.ac.id<sup>2\*</sup>

**Abstrak** – Penggunaan platform IoT dengan kualitas yang tidak sesuai sangat berbahaya karena dapat mengakibatkan delay pengiriman data yang tinggi dan kesalahan pengiriman data pada sistem monitoring. Banyak kendala ditemukan dalam memilih platform yang sesuai untuk sistem monitoring, seperti delay pengiriman yang tinggi, data error rate yang besar, terlalu sulit untuk membuat tampilan platform, serta kesulitan dalam mengakses platform yang digunakan yang menimbulkan kesukahan dalam menentukan platform yang akan digunakan. Oleh karena itu diperlukan suatu perbandingan kinerja komunikasi data yang dapat membantu pengguna memilih platform yang sesuai untuk proses monitoring yang dirancang.

Kinerja komunikasi data dari masing-masing platform sangat dibutuhkan untuk mengetahui kualitas dari platform App Invertor, Thingspeak dan Blynk. Kinerja platform diukur berdasarkan hasil perbandingan delay dan data error rate masing-masing platform. Hasil perbandingan kinerja ketiga platform ini dapat mempermudah pengguna untuk memilih platform yang cocok untuk dipakai sesuai dengan fungsi dan kualitas yang diinginkan. Oleh karena itu diperlukan suatu perbandingan kinerja komunikasi data untuk mengetahui kualitas dari platform App Invertor, Thingspeak dan Blynk.

Berdasarkan hasil penelitian, disimpulkan bahwa bahwa semua sistem pada model taman cerdas dapat mengirim data hasil baca sensor menuju platform IoT yang dituju. Namun hanya sistem yang menggunakan platform App Invertor dan Blynk dapat berjalan dengan baik sesuai dengan perancangan, jika dianalisis menggunakan perhitungan delay dan data error rate tiap pengiriman data. Dari hasil perbandingan kinerja platform dapat disimpulkan bahwa platform Blynk lebih unggul dibandingkan platform lainnya.

**Kata Kunci:** NodeMCU, App Inveror, Thingspeak, Blynk.

**Abstract** – The use of IoT platforms with lack of quality is very dangerous because it can lead to high data transmission delays and data transmission errors in the monitoring system. Many obstacles were found in choosing an appropriate platform for the monitoring system, such as high delivery delays, large data error rates, too difficult to create a platform display, as well as difficulties in accessing the platform used which caused difficulties in determining the platform to be used. Therefore we need a data communication performance comparison that can help users choose the appropriate platform for the monitoring process designed. The data communication performance of each platform is needed to determine the quality of the App Invertor, Thingspeak and Blynk platforms. Platform performance is measured based on the results of the comparison of the delay and error rate data for each platform. The results of the comparison of the performance of these three platforms can make it easier for users to choose a suitable platform to use according to the desired function and quality. Therefore, a comparison of data communication performance is needed to determine the quality of the App Invertor, Thingspeak and Blynk platforms. Based on the results of the study, it was concluded that all systems in the smart garden model could send sensor reading data to the intended IoT platform. However, only systems that use App Invertor and Blynk platforms can run well according to the design, if analyzed using the calculation of delay and error rate data for each data transmission. From the results of the comparison of platform performance, it can be concluded that the Blynk platform is superior to other platforms.

**Keywords:** NodeMCU, App Inveror, Thingspeak, Blynk.

## 1. Pendahuluan

Dalam dunia telekomunikasi digital, data akan dikirim menggunakan transmisi elektronik dari satu komputer ke komputer lain atau dari satu komputer ke terminal tertentu [1]. Data merupakan sinyal elektromagnetik yang dibangkitkan oleh sumber data yang dapat ditangkap dan dikirimkan ke terminal penerima. Dalam sistem *IoT* data diperoleh dari sensor, diolah dan dikirimkan ke web server *platform IoT*. Untuk mengukur kinerja komunikasi data, diperlukan dua parameter ukur yaitu *delay* dan *data error rate*.

Harry Yuliansyah (2016) meneliti “Uji Kinerja Pengiriman Data Secara Wireless Menggunakan Modul ESP8266 Berbasis *Rest Architecture*”[2]. Pengiriman data dari mikrokontroler menuju komputer server dapat dilakukan secara wireless. Pengiriman data pada penelitian ini menggunakan modul ESP8266. Penelitian ini menerapkan tiga metode yaitu dengan menggunakan AT-Command, Protokol SLIP dan NodeMCU. Pengujian dilakukan dengan cara mengirimkan data secara sistematis menuju komputer server. Kemudian data tersebut dianalisa sehingga dapat dihitung kecepatan *transfer* data dan kehandalan sistem.

S.Samsugi dan tim juga melakukan pengujian kinerja alat pengendali jarak jauh pada penerapan teknologi *internet of things* (IoT) untuk mengetahui perbedaan waktu saat sistem bekerja “Sistem Kendali Jarak Jauh Berbasis Arduino Dan Modul Wifi Esp8266”[3]. Sistem kendali pada penelitian ini dirancang menggunakan Arduino UNO dengan mikrokontroler ATmega328 sebagai pusat kendali dari sistem, serta modul wifi ESP8266 guna untuk komunikasi kontroler ke internet melalui media wifi. Interface dibuat dengan berbasis web dengan HTML5. Penelitian ini dilakukan untuk membandingkan penggunaan jaringan 3G dengan 4G. Penggunaan jaringan 3G dapat digunakan hanya saja membutuhkan waktu eksekusi yang lebih lama dibandingkan dengan jaringan 4G. Penggunaan jaringan 4G lebih baik disarankan dalam penerapan teknologi *internet of things* (IoT) karena waktu eksekusi yang lebih cepat.

Saat ini, model taman cerdas sudah tersedia, hanya saja untuk pengukuran kinerja komunikasi data masih belum diperhitungkan/diteliti. Kolam ikan cerdas berisikan sistem monitoring tingkat keasaman dan salinitas kolam ikan koi dengan tujuan mempermudah pengguna memantau salinitas air kolam dari jarak jauh menggunakan kontainer untuk menambahkan garam. Data diambil menggunakan sensor TDS yang kemudian diolah dan dikirimkan menggunakan platform *App Invertor* pada *SmartPhone* [4].

Tempat sampah cerdas berisikan sistem pembuka tutup otomatis tempat sampah dengan tujuan mempermudah masyarakat menangani masalah sampah. Sistem didesain menggunakan akuator berupa motor servo sebagai penggerak pada tutup tempat sampah. Data diambil menggunakan sensor *ultrasonik* yang kemudian diolah dan dikirimkan menggunakan platform *Thingspeak* pada Laptop [5].

Taman mini cerdas berisikan sistem penyiraman tanaman otomatis dan pemantau kondisi tanah jarak jauh menggunakan deteksi lokasi dengan tujuan mempermudah pengguna untuk memantau kondisi tanaman di lapangan. Sistem didesain menggunakan pompa air sebagai penyiram tanaman secara otomatis. Data diambil menggunakan sensor *soil moisture* yang kemudian diolah dan dikirimkan menggunakan platform *Blynk* pada *smartphone* [6].

Beberapa parameter kerja yang akan diukur/diteliti, dan kemudian dibandingkan adalah *delay* dan *data error rate*. Berdasarkan permasalahan tersebut, penulis akan meneliti tentang kinerja komunikasi data dengan cara membandingkan kinerja komunikasi data antara *App Invertor*, *Blynk* dan *Thingspeak* pada model taman cerdas. Pengiriman data dalam penelitian ini menggunakan modul *NodeMCU*. Modul *NodeMCU* memiliki beberapa keunggulan di antaranya harga murah dan *firmware* yang dapat dikembangkan [7]. Sedangkan untuk *monitoring*, peneliti menggunakan *smart phone* dan laptop.

## 2. Metodologi Penelitian

### 2.1. Studi Pustaka

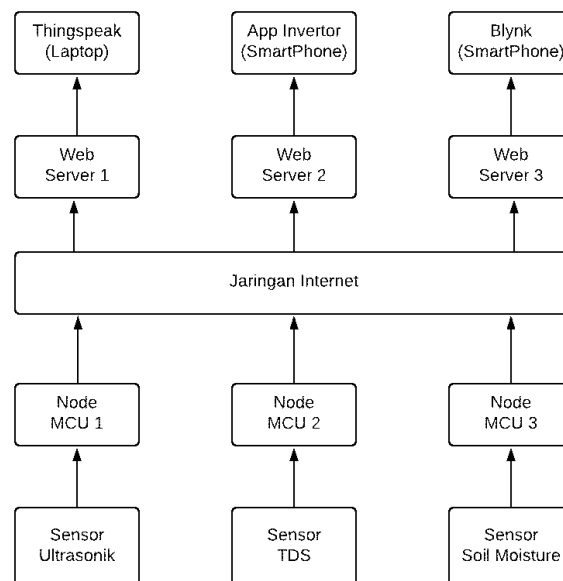
Tahap awal penelitian ini dimulai dari mencari dan mempelajari referensi dari buku, jurnal, dan artikel-artikel serta skripsi-skripsi yang berkaitan dengan *Thingspeak*, *App inventor*, *Blynk* dan mikrokontroler arduino menggunakan Modul *NodeMCU* ESP32 (Sensor *Soil Moisture*), ESP8266 (Sensor TDS), ESP8266 (Sensor *ultrasonik*). Serta mempelajari komunikasi data antara *Node Mcu* dan *IoT*.

### 2.2. Pengujian Alat dan Pengambilan Data

Tahap ini bertujuan melakukan pengujian pada alat secara keseluruhan baik perangkat keras maupun perangkat lunak agar sesuai dengan hasil yang diinginkan, serta memperhatikan komunikasi modul ESP32 (*sensor soil moisture*), ESP8266 (SensorTDS), ESP8266 (*sensor ultrasonik*). Pengambilan data dari *Thingspeak*, *App Inventor* dan *Blynk* dilakukan:

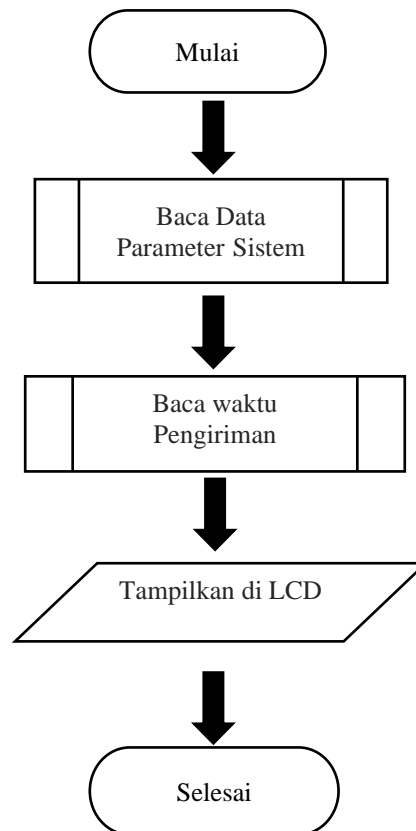
- Saat sensor TDS mendeteksi kadar garam, *NodeMCU* mampu mengirimkan data untuk ditampilkan di *App Inventor*.
- Saat sensor ultrasonik mendeteksi ketinggian sampah, *NodeMCU* mampu mengirimkan data untuk ditampilkan di *Thingspeak*.
- Saat sensor *soil moisture* mendeteksi kelembaban tanah, *NodeMCU* mampu mengirimkan data untuk ditampilkan di *Blynk*.

Diagram blok proses kerja sistem secara umum dapat dilihat pada gambar 1.



Gambar 1. Diagram blok proses kerja sistem secara umum.

Proses pengambilan data dilakukan sebanyak 5 kali pengambilan data, menggunakan interval pengiriman data tiap 2,4,10,15, dan 30 detik untuk masing-masing sistem pada Model Taman Cerdas. jumlah data yang diambil setiap interval pengiriman sebanyak 10 data. Data yang diambil digunakan sebagai sarana untuk menguji ketelitian masing-masing sensor setiap sitem dengan mengukur kecepatan deteksi sensor, menguji tingkat keberhasilan data yang dikirimkan dari masing-masing *platform* yang digunakan ketiga sistem saat bekerja menggunakan interval waktu pengiriman yang sama, serta mengamati waktu pengiriman data yang diperlukan sistem.



Gambar 2. Flowchart secara umum proses pengambilan dan penampilan data pada LCD

Dari gambar 2 dapat dilihat bahwa ketika sistem dijalankan, maka sensor akan membaca data parameter yang diambil dari masing-masing sensor pada sistem, serta membaca waktu pengiriman yang diperlukan masing-masing sistem. Untuk mempermudah pengguna memantau sistem secara langsung, maka data pada *NodeMCU* akan ditampilkan di LCD I2C 16x2.

### 3. Hasil dan Pembahasan

#### 3.1. Pengujian Kinerja Komunikasi Data Pada Masing-masing Platform

Pengujian sistem digunakan untuk mengukur tingkat keberhasilan pengiriman data dari *NodeMcu* menuju masing-masing platform. Proses pengujian dilakukan dengan mengirim data dari sisi transmitter (tx) ke sisi receiver (rx). Untuk sisi tx yaitu *NodeMCU* sebagai access point sedangkan pada sisi rx ada tiga yaitu smartphone yang sudah ter-install platform *App inverter*, *Blynk* dan terhubung *Thingspeak* yang sudah terhubung pada laptop.

Pengujian sistem dilakukan dengan beberapa bagian yaitu pengujian ketepatan pengiriman data serta perhitungan *delay* dan *data error rate* pada tiap platform *IoT* saat proses pengiriman data. Pada tahap pengujian ini jumlah data yang diambil setiap pengiriman yaitu 10 data, dengan range pengiriman data tiap 2,4,10,15, dan 30 detik. Rata-rata perhitungan *delay* dan *data error rate* pada tiap-tiap Platform akan diubah dalam bentuk grafik untuk melihat performa dan perbandingannya.

Pengujian kinerja komunikasi data secara keseluruhan dilakukan dengan cara *NodeMcu* mampu mengirimkan data sensor menuju masing-masing Platform. Adapun pengujian meliputi pengujian ketepatan pengiriman data serta perhitungan *delay* dan *data error rate* pada saat proses pengiriman data.

Perhitungan *data error rate* masing-masing sistem akan diukur berdasarkan ketepatan data

yang dikirimkan dari sensor yang tersimpan di dalam *NodeMCU* saat diterima dan ditampilkan di masing-masing *platform*. *Data error rate* diukur menggunakan rumus persamaan 1.

$$\text{Data error rate} = \left( \frac{A-B}{B} \right) \times 100 \quad (1)$$

dengan A adalah data terkirim dan B adalah data yang diterima

Perhitungan *delay* dilakukan dengan mengamati waktu data diterima pada *App Invertor* dan waktu kirim pada serial monitor. *Delay* diukur menggunakan rumus persamaan 2.

$$\text{Delay} = A - B \quad (2)$$

dengan A adalah waktu data diterima dan B adalah waktu data dikirim.

### 3.2. Data Pengujian Sistem

#### 3.2.1. Data Pengujian Komunikasi Esp 8266 – App Inventor

Pengujian komunikasi Esp 8266-App invetor dilakukan dengan cara memasukkan sensor TDS kedalam larutan air garam dan menggunakan arus tegangan 5 Volt melalui elektrida yang dimasukkan ke dalam air untuk mengukur tegangan ADC yang akan dikonversi menjadi nilai salinitas dalam ppt.

Pada pengujian pengiriman data antara *Esp 8266- Platform App Invertor* akan diambil sebanyak 50 data, dapat dilihat pada tabel 1, 2, 3, 4 dan 5. Untuk dapat mengukur kinerja *platform App Invertor* maka dilakukan perhitungan *delay* dan *data error rate* setiap interval pengiriman. Perhitungan *delay* pada *platform App invetor* menggunakan perhitungan *delay* antara *platform Thingspeak – App invetor*.

**Tabel 1.** Data *Platform App Invertor* dengan *interval* pengiriman /2 detik.

Waktu SM	Data SM (ppt)	Waktu TS	Data TS (ppt)	Waktu terima App Invertor	Data App Invertor (ppt)	Hasil perhitungan <i>delay</i> (s)	Hasil perhitungan <i>Data error rate</i>
13.07.01	2	13.07.03	2	13.07.04	2	00.00.01	0
13.07.03	2	-	-	-	-	-	1
13.07.05	2	-	-	-	-	-	1
13.07.07	5,06	-	-	-	-	-	1
13.07.08	5,77	-	-	-	-	-	1
13.07.10	7	-	-	-	-	-	1
13.07.12	7,56	-	-	-	-	-	1
13.07.14	7,56	-	-	-	-	-	1
13.07.16	4,65	-	-	-	-	-	1
13.07.18	5,06	-	-	-	-	-	1

**Tabel 2.** Data Platform App Invertor dengan interval pengiriman /4 detik.

Waktu SM	Data SM (ppt)	Waktu TS	Data TS (ppt)	Waktu terima App Invertor	Data App Invertor (ppt)	Hasil perhitungan delay (s)	Hasil perhitungan Data error rate
13.11.00	14,27	13.11.01	14,27	13.11.03	14,27	00.00.02	0
13.11.04	14,43	-	-	-	-	-	1
13.11.08	14,48	-	-	-	-	-	1
13.11.12	14,58	-	-	-	-	-	1
13.11.16	14,58	13.11.16	14,5798	13.11.18		00.00.02	0
13.11.20	14,78	-	-	-	-	-	1
13.11.24	14,89	-	-	-	-	-	1
13.11.28	14,99	13.11.32	14,987	13.11.32		00.00.00	0
13.11.33	14,99	-	-	-	-	-	1
13.11.37	15,14	-	-	-	-	-	1

**Tabel 3.** Data Platform App Invertor dengan interval pengiriman /10 detik.

Waktu SM	Data SM (ppt)	Waktu TS	Data TS (ppt)	Waktu terima App Invertor	Data App Invertor (ppt)	Hasil perhitungan delay (s)	Hasil perhitungan Data error rate
13.11.45	15,34	13.11.48	15,3433	13.11.50	15,3433	00.00.02	0
13.11.55	15,39	-	-	-	-	-	1
13.12.05	14,53	-	-	-	-	-	1
13.12.15	11,78	-	-	-	-	-	1
13.12.25	11,53	-	-	-	-	-	1
13.12.34	12,49	-	-	-	-	-	1
13.12.44	10,2	-	-	-	-	-	1
13.12.54	12,29	-	-	-	-	-	1
13.13.04	13,15	13.13.07	13,15	13.13.08	03.42.37	00.00.01	0
13.13.14	13,46	-	-	-	-	-	1

**Tabel 4.** Data Platform App Invertor dengan interval pengiriman /15 detik.

Waktu SM	Data SM (ppt)	Waktu TS	Data TS (ppt)	Waktu terima App Invertor	Data App Invertor (ppt)	Hasil perhitungan delay (s)	Hasil perhitungan Data error rate
13.07.01	2	-	-	-	-	-	1
13.07.16	4,65	13.07.19	4,65	13.07.21	4,65	00.00.02	0
13.07.32	3,28	13.07.35	3,28	13.07.37	3,28	00.00.02	0
13.07.48	2	13.07.50	2	13.07.52	2	00.00.02	0
13.08.03	8,37	13.08.06	8,37	13.08.08	8,37	00.00.02	0
13.08.19	10,46	13.08.22	10,46	13.08.23	10,46	00.00.01	0
13.08.35	11,07	13.08.38	11,07	13.08.39	11,07	00.00.01	0
13.08.51	11,53	13.08.54	11,53	13.08.55	11,53	00.00.01	0
13.09.06	12,34	13.09.09	12,34	13.09.10	12,34	00.00.01	0
13.09.22	13,56	13.09.25	13,56	13.09.27	13,56	00.00.02	0

**Tabel 5.** Data *Platform App Invertor* dengan *interval* pengiriman /30 detik

Waktu SM	Data SM (ppt)	Waktu TS	Data TS (ppt)	Waktu terima App Invertor	Data App Invertor (ppt)	Hasil perhitungan delay (s)	Hasil perhitungan Data error rate
13.07.16	4,65	13.07.19	4,65	13.07.21	4,65	00.00.02	0
13.07.46	6,23	-	-	-	-	-	1
13.08.15	10,15	-	-	-	-	-	1
13.08.45	11,22	-	-	-	-	-	1
13.09.16	13,46	-	-	-	-	-	1
13.09.46	13,92	-	-	-	-	-	1
13.10.16	12,54	-	-	-	-	-	1
13.10.45	13,82	-	-	-	-	-	1
13.11.16	14,58	-	-	-	-	-	1
13.11.45	15,34	13.11.48	15,34	13.11.50	15,34	00.00.02	0

Tabel 4.1, 4.2, 4.3, 4.4, dan 4.5. pengujian transmisi data pada *platform App invertor* dilakukan sebanyak 5 kali percobaan, dan hasil yang didapat banyak sekali data *error*, terutama pada pengiriman menggunakan interval pengiriman data /2, 4, 10 dan 30 detik, seperti pada tabel diatas. Berbeda dengan pengiriman data /15 detik hampir semua data diterima oleh *platform App Invertor*. Hal ini dipengaruhi oleh karakteristik *server Thingspeak* yang membutuhkan waktu minimal 15 detik untuk memperbaharui data (*update data*), sehingga data yang dapat diteruskan menuju *platform App invertor* sesuai dengan data yang diterima oleh *server Thingspeak*.

### 3.2.2. Data Pengujian Komunikasi Esp 8266 – Thingspeak

Pengujian komunikasi Esp 8266-*Thingspeak* dilakukan dengan meletakkan sensor *ultrasonik* pada sisi atas cover sampah, sehingga pada saat sampah dimasukkan dengan ketinggian yang bervariasi sampai batas maksimal agar dapat dibaca oleh sensor. Adapun sampah yang dimasukkan yaitu sampah kering padat berupa kertas. Nilai ketinggian sampah akan dikirimkan secara *realtime* ke *Thingspeak*.

Pada pengujian pengiriman data antara *Esp 8266- Platform Thingspeak* akan diambil sebanyak 50 data dapat dilihat pada tabel 6, 7, 8, 9 dan 10. Untuk dapat mengukur kinerja *platform Thingspeak* maka dilakukan perhitungan *delay* dan *data error rate* setiap interval pengiriman.

**Tabel 6.** Data *Platform Thingspeak* dengan *interval* pengiriman /2 detik.

Waktu SM	Data SM (Cm)	Waktu terima Thingspeak	Data Thingspeak (Cm)	Hasil perhitungan delay (s)	Hasil perhitungan Data error rate
14.10.21	26	14.10.24	26	00.00.03	0
14.10.25	26	-	-	-	1
14.10.27	26	-	-	-	1
14.10.29	26	-	-	-	1
14.10.31	6	-	-	-	1
14.10.33	21	-	-	-	1
14.10.35	16	-	-	-	1
14.10.37	17	14.10.40	17	00.00.03	0
14.10.39	18	-	-	-	1
14.10.41	17	-	-	-	1

**Tabel 7.** Data Platform Thingspeak dengan interval pengiriman /4 detik.

Waktu SM	Data SM (Cm)	Waktu terima Thingspeak	Data Thingspeak (Cm)	Hasil perhitungan delay (s)	Hasil perhitungan Data error rate
14.11.01	13	-	-	-	1
14.11.05	16	-	-	-	1
14.11.09	20	14.11.13	20	00.00.04	0
14.11.13	25	-	-	-	1
14.11.17	16	-	-	-	1
14.11.21	18	-	-	-	1
14.11.25	29	14.11.29	29	00.00.04	0
14.11.29	25	-	-	-	1
14.11.33	24	-	-	-	1
14.11.37	23	-	-	-	1

**Tabel 8.** Data Platform Thingspeak dengan interval pengiriman /10 detik.

Waktu SM	Data SM (Cm)	Waktu terima Thingspeak	Data Thingspeak (Cm)	Hasil perhitungan delay (s)	Hasil perhitungan Data error rate
14.12.14	17	14.12.17	17	00.00.03	0
14.12.24	6	-	-	-	1
14.12.34	24	-	-	-	1
14.12.44	8	-	-	-	1
14.12.54	10	-	-	-	1
14.13.04	14	-	-	-	1
14.13.14	14	-	-	-	1
14.13.24	24	-	-	-	1
14.13.34	25	14.13.38	25	00.00.04	0
14.13.44	16	-	-	-	1

**Tabel 9.** Data Platform Thingspeak dengan interval pengiriman /15 detik.

Waktu SM	Data SM (Cm)	Waktu terima Thingspeak	Data Thingspeak (Cm)	Hasil perhitungan delay (s)	Hasil perhitungan Data error rate
14.11.09	20	14.11.13	20	00.00.04	0
14.11.25	29	14.11.29	29	00.00.04	0
14.11.41	26	14.11.45	26	00.00.04	0
14.11.56	9	-	-	-	1
14.12.10	15	-	-	-	1
14.12.24	6	-	-	-	1
14.12.38	12	-	-	-	1
14.12.54	10	-	-	-	1
14.13.10	12	-	-	-	1
14.13.24	24	-	-	-	1



**Tabel 10.** Data Platform Thingspeak dengan interval pengiriman /30 detik.

Waktu SM	Data SM (Cm)	Waktu terima Thingspeak	Data Thingspeak (Cm)	Hasil perhitungan delay (s)	Hasil perhitungan Data error rate
14.12.16	11	-	-	-	1
14.12.46	14	14.12.49	14	00.00.03	0
14.13.26	6	-	-	-	1
14.13.55	17	-	-	-	1
14.14.25	55	-	-	-	1
14.14.55	17	14.14.58	17	00.00.03	0
14.15.25	19	-	-	-	1
14.15.56	19	-	-	-	1
14.16.26	17	-	-	-	1
14.16.55	13	-	-	-	1

Tabel 6, 7, 8, 9 dan 10. pengujian transmisi data pada platform Thingspeak dilakukan sebanyak 5 kali percobaan, dari hasil pengujian banyak sekali data error yang didapat, terutama pada pengiriman menggunakan interval pengiriman data /2, 4, 10 dan 30 detik, seperti pada tabel diatas. Berbeda dengan pengiriman data /15 detik data yang diterima oleh platform Thingspeak lebih banyak. Hal ini dipengaruhi oleh karakteristik server Thingspeak yang membutuhkan waktu minimal 15 detik untuk memperbaharui data (*update data*), sehingga data yang dapat diterima oleh platform Thingspeak dengan interval pengiriman data /15 detik lebih banyak.

### 3.2.3. Data Pengujian Komunikasi Esp 32 – Blynk

Pengujian komunikasi Esp 32-Blynk dilakukan dengan cara memasukkan sensor *Soil Moisture* kedalam media yang sudah berisi tanah dengan kondisi tanah dalam keadaan kering nilai range analog sensor > 3500, kondisi tanah dalam keadaan lembab nilai range analog sensor < 3500 dan > 1600 serta kondisi tanah dalam keadaan basah nilai range analog sensor < 1600. Hasil keluaran sensor *soil moisture* akan dikonversi menjadi tegangan 12 BIT dan akan dikirim secara *real time* ke *blynk*.

Pada pengujian pengiriman data antara Esp 32- Platform blynk akan diambil sebanyak 50 data dapat dilihat pada tabel 11, 12, 13, 14 dan 15. Untuk dapat mengukur kinerja platform blynk maka dilakukan perhitungan *delay* dan *data error rate* setiap interval pengiriman.

**Tabel 11.** Data Platform Blynk dengan interval pengiriman /2 detik.

Waktu SM	Data SM (V ADC)	Waktu terima Blynk	Data Blynk (V ADC)	Hasil perhitungan delay (s)	Hasil perhitungan Data error rate
13.18.10	2,99	13.18.11	2,99	00.00.01	0
13.18.12	3,01	13.18.13	3,01	00.00.01	0
13.18.14	3	13.18.15	3	00.00.01	0
13.18.16	2,93	13.18.18	2,93	00.00.02	0
13.18.18	2,7	13.18.19	2,7	00.00.01	0
13.18.20	2,66	-	-	-	1
13.18.22	2,62	13.18.23	2,62	00.00.01	0
13.18.24	2,45	13.18.25	2,45	00.00.01	0
13.18.26	2,39	13.18.27	2,39	00.00.01	0
13.18.28	2,32	13.18.29	2,32	00.00.01	0

**Tabel 12.** Data *Platform Blynk* dengan *interval* pengiriman /4 detik.

Waktu SM	Data SM (V ADC)	Waktu terima <i>Blynk</i>	Data <i>Blynk</i> (V ADC)	Hasil perhitungan <i>delay</i> (s)	Hasil perhitungan Data <i>error rate</i>
13.19.56	-	-	-	-	1
13.20.00	1,2	13.20.01	1,2	00.00.01	0
13.20.04	1,38	13.20.05	1,38	00.00.01	0
13.20.08	1,47	13.20.10	1,47	00.00.02	0
13.20.12	-	-	-	-	1
13.20.16	1,32	13.20.16	1,32	00.00.00	0
13.20.20	1,61	13.20.21	1,61	00.00.01	0
13.20.24	3,24	13.20.25	3,24	00.00.01	0
13.20.28	2,07	13.20.29	2,07	00.00.01	0
13.20.32	-	-	-	-	1

**Tabel 13.** Data *Platform Blynk* dengan *interval* pengiriman /10 detik.

Waktu SM	Data SM (V ADC)	Waktu terima <i>Blynk</i>	Data <i>Blynk</i> (V ADC)	Hasil perhitungan <i>delay</i> (s)	Hasil perhitungan Data <i>error rate</i>
13.21.00	1,81	13.21.01	1,81	00.00.01	0
13.21.10	1,81	13.21.11	1,81	00.00.01	0
13.21.20	1,82	13.21.22	1,82	00.00.02	0
13.21.30	1,89	13.21.31	1,89	00.00.01	0
13.21.40	2,23	13.21.41	2,23	00.00.01	0
13.21.50	2,25	13.21.50	2,25	00.00.00	0
13.22.00	1,92	13.22.01	1,92	00.00.01	0
13.22.10	1,96	13.22.11	1,96	00.00.01	0
13.22.20	-	-	-	-	1
13.22.30	-	-	-	-	1

**Tabel 14.** Data *Platform Blynk* dengan *interval* pengiriman /15 detik.

Waktu SM	Data SM (V ADC)	Waktu terima <i>Blynk</i>	Data <i>Blynk</i> (V ADC)	Hasil perhitungan <i>delay</i> (s)	Hasil perhitungan Data <i>error rate</i>
13.23.00	1,81	13.23.01	1,81	00.00.01	0
13.23.15	2,14	13.23.16	2,14	00.00.01	0
13.23.30	1,9	13.23.31	1,9	00.00.01	0
13.23.45	2,02	13.23.47	2,02	00.00.02	0
13.24.00	1,71	13.24.00	1,71	00.00.00	0
13.24.15	1,87	13.24.16	1,87	00.00.01	0
13.24.30	2,58	13.24.31	2,58	00.00.01	0
13.24.45	1,52	13.24.46	1,52	00.00.01	0
13.25.00	2,54	13.25.01	2,54	00.00.01	0
13.25.15	1,37	13.25.16	1,37	00.00.01	0

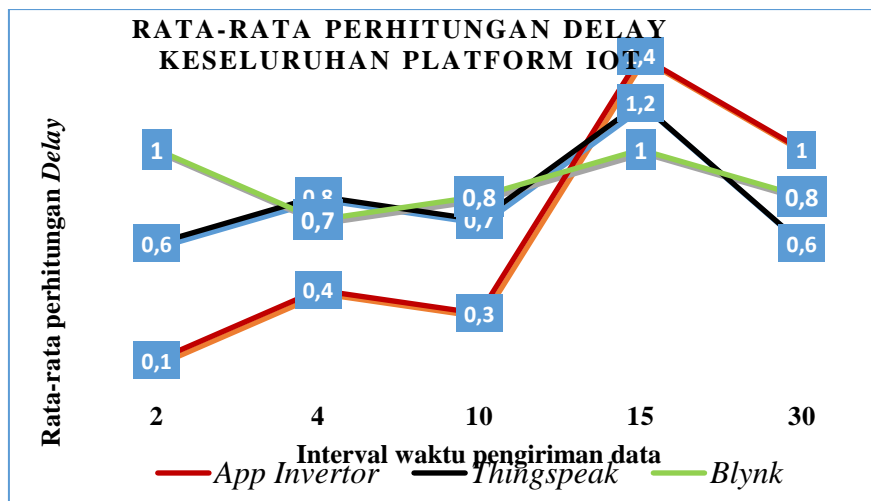
Tabel 15. Data Platform Blynk dengan interval pengiriman /30 detik.

Waktu SM	Data SM (V ADC)	Waktu terima Blynk	Data Blynk (V ADC)	Hasil perhitungan delay (s)	Hasil perhitungan Data error rate
13.23.00	1,79	13.23.01	1,79	00.00.01	0
13.23.30	1,9	13.23.31	1,9	00.00.01	0
13.24.00	1,71	13.24.01	1,71	00.00.01	0
13.24.30	2,58	13.24.30	2,58	00.00.00	0
13.25.00	2,54	13.25.01	2,54	00.00.01	0
13.25.30	2,86	13.25.31	2,86	00.00.01	0
13.26.00	1,57	13.26.01	1,57	00.00.01	0
13.26.30	-	-	-	-	1
13.27.00	2,31	13.27.01	2,31	00.00.01	0
13.27.30	1,64	13.27.31	1,64	00.00.01	0

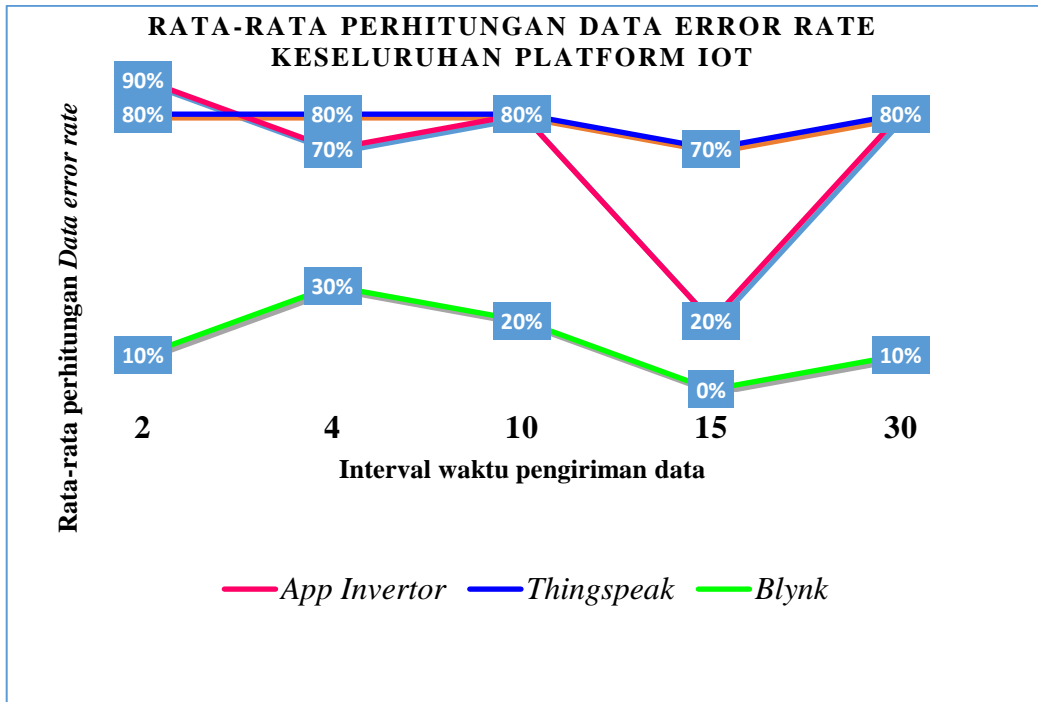
Pada tabel 11, 12, 13, 14 dan 15. pengujian transmisi data pada platform Blynk dilakukan sebanyak 5 kali percobaan, dari hasil pengujian ada beberapa data error yang didapat pada interval pengiriman tertentu, terutama pada pengiriman menggunakan interval pengiriman data /4 dan 10 detik, seperti pada tabel 13 dan 14 diatas. Berbeda dengan pengiriman data /15 detik, seperti pada tabel 14 menunjukkan bahwa tidak ada data error rate yang diterima oleh platform Blynk. Hal ini dipengaruhi oleh karakteristik server Thingspeak yang membutuhkan waktu minimal 15 detik untuk memperbaharui data (update data), sehingga data yang dapat diterima oleh platform Thingspeak lebih banyak.

### 3.3. Hasil Pengujian Keseluruhan Sistem

Hasil pengujian delay dan data error rate keseluruhan sistem digunakan sebagai pembandingan kinerja Platform IoT pada sistem taman cerdas. Hasil perbandingan kinerja platform bertujuan untuk mengetahui kualitas dari masing-masing platform. Data dari hasil pengujian akan diambil rata-ratanya dan diubah menjadi grafik untuk melihat performa dan perbandingannya. Data yang digunakan untuk membandingkan kinerja platform sesuai dengan rata-rata perhitungan delay dan data error rate masing-masing interval pengiriman data. Data hasil pengujian delay dan data error rate keseluruhan sistem dapat dilihat pada grafik 1 dan 2.



Gambar 3. Rata-rata perhitungan delay keseluruhan platform IoT



Gambar 4. Rata-rata perhitungan data *error rate* keseluruhan *platform IoT*

Berdasarkan grafik 1 dan 2 dapat kita lihat bahwa performa *delay* dan *data error rate* masing-masing *platform* bervariasi sesuai dengan interval pengiriman data yang digunakan. Rata-rata perhitungan *data error rate* sangat berpengaruh terhadap perhitungan *delay*, karena rata-rata dari perhitungan *delay* akan dihitung sesuai dengan jumlah data yang diterima masing-masing *platform IoT*. Semakin banyak data yang diterima *platform IoT* maka semakin tinggi hasil perhitungan *delay* yang didapat.

Pada *platform App Inventor* memperoleh nilai perhitungan *delay* yang sangat rendah, dapat dilihat pada grafik 1. Hal ini dikarenakan besarnya nilai *data error rate* yang didapatkan sangat besar, (dapat dilihat pada tabel 2). Besarnya nilai *data error rate* disebabkan *platform App Inventor* tidak dapat menerima semua data yang dikirimkan *NodeMCU*. Pada pengujian ini, *platform App Inventor* yang menggunakan server *Thingspeak* untuk transmisi data, sehingga data yang diterima *platform App Inventor* dipengaruhi data pada *platform Thingspeak*.

Pada *platform Thingspeak* memperoleh nilai perhitungan rata-rata *delay* yang cukup tinggi, dapat dilihat pada grafik 1. Hal ini dikarenakan nilai *data error rate* yang didapatkan cukup besar (dapat dilihat pada tabel 2). Besarnya nilai *data error rate* disebabkan *platform Thingspeak* yang membutuhkan waktu minimal 15 detik untuk memperbaharui (*update*) data, sehingga pada proses pengiriman data pada interval tertentu *platform Thingspeak* tidak dapat menerima semua data yang dikirimkan dari *Node MCU*.

Pada *platform Blynk* memperoleh nilai perhitungan *delay* yang sangat tinggi, dapat dilihat pada grafik 1. hal ini dikarenakan nilai *data error rate* yang didapatkan sangat kecil, (dapat dilihat pada tabel 2.). Nilai *data error rate* yang sangat kecil disebabkan *platform Blynk* yang dapat menerima data perdetik, sehingga hampir semua data yang dikirimkan dari *Node MCU* dapat diterima oleh *platform Blynk* tanpa pengaruh interval pengiriman data yang digunakan.

Berdasarkan hasil Analisis yang diperoleh, dengan membandingkan nilai perhitungan *delay* dan *data error rate* keseluruhan sistem. *Platform Blynk* lebih unggul dibandingkan *platform* lainnya, hal ini dikarenakan nilai perhitungan *data error rate* yang sangat kecil, begitu juga dengan nilai perhitungan *delay* untuk tiap pengiriman data yang berkisar antara 1-2 detik. sedangkan untuk *platform App inventor* bisa dikatakan cukup bagus, karena pada perhitungan *delay* juga memperoleh nilai yang rendah dan sama dengan nilai perhitungan *delay* pada *platform*

*Blynk*, berkisar antara 1-2 detik per pengiriman data. Kelemahan dari *platform App Invertor* yaitu *App Invertor* yang menggunakan server *Thingspeak* untuk transmisi data, yang menyebabkan ada banyak data yang dikirimkan *Node MCU* tidak diterima oleh *platform App Invertor*, sehingga *platform App Invertor* memperoleh nilai *data error rate* yang besar.

Dari hasil analisis juga dapat disimpulkan bahwa *platform Thingspeak* memiliki *performa* yang kurang bagus, jika dilihat dari hasil perhitungan *delay* per pengiriman data yang didapatkan cukup tinggi, berkisar antara 3-4 detik. Nilai *delay* yang tinggi pada *platform Thingspeak* dipengaruhi lokasi *server Thingspeak* yang berada di Inggris. Keunggulan dari *platform Thingspeak* yaitu riwayat data yang diterima dapat dilihat dengan mudah oleh pengguna, serta dapat didownload dalam bentuk *csv*.

#### 4. Kesimpulan

Setelah mempelajari alat yang dibuat sebelumnya, implementasi, dan pengujian Kinerja komunikasi Data pada Model Taman Cerdas, dapat diambil kesimpulan bahwa Semua sistem pada model taman cerdas dapat mengirim data hasil baca sensor menuju *platform IoT* yang dituju. Namun hanya sistem yang menggunakan *platform App Invertor* dan *Blynk* dapat berjalan dengan baik sesuai dengan perancangan, jika dianalisis menggunakan perhitungan *delay* dan *data error rate* tiap pengiriman data. Dari hasil perbandingan kinerja *platform* dapat disimpulkan bahwa *platform Blynk* lebih unggul dibandingkan *platform* lainnya.

#### Ucapan Terima Kasih

Terima kasih penulis sampaikan kepada Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Universitas Sanata Dharma yang telah mendanai penelitian ini.

#### Referensi

- [1] Ariyus, Dony dan Andri, Rum. 2008. “*Komunikasi Data*”
- [2] Yuliansah, Harry. 2016. Uji Kinerja Pengiriman Data Secara Wireless Menggunakan Modul ESP8266 Berbasis Rest Architecture. *Jurnal Rekayasa dan Teknologi Elektro*, Vol. 10, No. 2.
- [3] S, Samsugi, dkk. 2017. Sistem Kendali Jarak Jauh Berbasis Arduino dan Modul Wifi ESP8266. *Prosiding Seminar Nasional XII “Rekayasa Teknologi Industri dan Informasi 2017 Sekolah Tinggi Teknologi Nasional Nusantara”*.
- [4] Putri, Novia Indriani. 2020. *Sistem Monitoring Dan Pengendalian Jarak Jauh Tingkat Keasaman Dan Salinitas Kolam Ikan Koi*. Skripsi. Universitas Sanata Dharma.
- [5] Moris, Fridyastuti Dwi. 2020. *Pembuka Tutup Otomatis Dan Pemantau Isi Tempat Sampah Jarak Jauh Dengan Deteksi Lokasi*. Skripsi. Universitas Sanata Dharma.
- [6] Jonathan. 2020. *Penyiram Tanaman Otomatis Dan Pemantauan Kondisi Tanah Jarak Jauh Dengan Deteksi Lokasi*. Skripsi. Universitas Sanata Dharma.
- [7] Saptaji. 2020. IoT Dengan Arduino dan Thingspeak. <http://saptaji.com/2016/11/21/iot-dengan-arduino-dan-thingspeak/> diakses 15 Desember 2020.