

Prototype Sistem Multi-Telemetri Wireless Untuk Mengukur Suhu Udara Berbasis Mikrokontroler ESP8266 Pada Greenhouse

Hanum Shirotu Nida^{*1}, Mahar Faiqurahman², Zamah Sari³

^{1,2,3}Universitas Muhammadiyah Malang

hanumshirotunida@outlook.com^{*1}, mahar@umm.ac.id², abdzamahsari@gmail.com³

Abstrak

Telemetri wireless adalah proses pengukuran parameter suatu obyek yang hasil pengukurannya dikirimkan ke tempat lain melalui proses pengiriman data tanpa menggunakan kabel (wireless), sedangkan multi telemetri adalah gabungan dari beberapa telemetri itu sendiri. Penelitian ini merancang prototype sistem multi-telemetri wireless untuk mengukur suhu udara dan kelembaban udara pada greenhouse dengan menggunakan sensor DHT11 dan data hasil dari pembacaan sensor dikirim dengan menggunakan modul WiFi ESP8266 ke server dengan menggunakan protokol HTTP. Dalam penelitian ini diuji nilai sensor DHT11, heap memory ESP8266, jarak atau jangkauan ESP8266, uji coba data missing handling dan kestabilan jaringan. Berdasarkan hasil pengujian diketahui bahwa sensor DHT11 memiliki rata-rata kesalahan ukur suhu 0.92 °C dan kelembaban 3.1%. Modul WiFi ESP8266 mampu menyimpan dan mengirim buffer hingga 100 data dan dapat melakukan pengiriman dalam jangkauan 50 meter. Data missing handling memanfaatkan buffer untuk menyimpan data selama server sedang tidak dapat diakses oleh sensor node agar data tidak hilang. Kestabilan pengiriman data atau koneksi sensor node dengan server dipengaruhi oleh jumlah access point yang sedang berkomunikasi di sekitar access point server dengan menggunakan channel yang sama.

Kata kunci: Telemetri, Wireless, ESP8266, DHT11, Greenhouse

Abstract

Wireless telemetry is a process of measuring an object parameter, the measurement results sent to another node through the process of sending data wireless, whereas multi telemetry is a composite of several telemetry itself. In this study is designed prototype multi-telemetry systems to measure air temperature and humidity in greenhouse using DHT11 sensors and outcome data from sensor readings are sent using the WiFi module ESP8266 to the server using HTTP protocol. In this study tested the sensor value of DHT11, heap memory of ESP8266, ESP8266 distance or range, data missing handling test and network stability. Based on test results DHT11 known that the sensor has an average error of measuring temperature is 0.92 °C and humidity is 3.1%. ESP8266 WiFi module is able to store and transmit buffers up to 100 data and can make deliveries within a range of 50 meters. Also, data missing handling utilizing a buffer to store data during unavailable server observer is not accessible by the sensor nodes to reduce the risk of data loss. The stability data or sensor connection to the server node is affected by the number of access points are communicating around the access point server by using the same channel.

Keywords: Telemetri, Wireless, ESP8266, DHT11, Greenhouse

1. Pendahuluan

Greenhouse atau yang biasa disebut rumah kaca adalah sebuah bangunan yang terdiri dari bahan kaca atau plastik yang tebal dan menutup seluruh permukaan baik dinding maupun atap. Model bangunan greenhouse ini banyak digunakan untuk kegiatan pertanian yang tidak tergantung pada keadaan musim dan dapat dilakukan sepanjang tahun. Tumbuhan yang dibudidayakan pun dapat bervariasi. Sehingga menentukan karakteristik kondisi lingkungan greenhouse yang harus dipenuhi oleh petani. Beberapa parameter yang digunakan sebagai acuan menentukan kondisi lingkungan dari greenhouse, untuk tanaman tertentu diantaranya adalah kondisi suhu, kelembaban udara, pencahayaan, kadar CO₂, dan sebagainya [1].

Berbagai teknologi telah dikembangkan untuk memantau kondisi dari *greenhouse*. Proses pemantauan ini bertujuan untuk menganalisis segala perubahan kondisi lingkungan pada *greenhouse*. Salah teknologi yang digunakan adalah sistem *monitoring* dengan memanfaatkan sensor. Beberapa penelitian sebelumnya telah memanfaatkan sistem sensor untuk melakukan *monitoring*, seperti pada sistem *monitoring* listrik Prabayar [2], sistem *monitoring* air nutrisi akuaponik [3], dan sistem *monitoring healthcare* yang dipasang pada tubuh [4]. Sedangkan penerapan sistem *monitoring* pada *greenhouse* telah dilakukan pada beberapa penelitian sebelumnya [5] [6] [7]. Pada penelitian tersebut, di dalam arsitektur sistem *monitoring* yang dibangun, juga memanfaatkan sensor sebagai instrumen untuk mengambil data dari lingkungan. Data hasil *sensing* tersebut dikirim oleh sensor *node* ke *server* dengan memanfaatkan jaringan *wireless*. Hal ini dilakukan dengan pertimbangan karena lokasi dari sensor *node* cukup jauh dari *server* pengolah data, sehingga penggunaan medium *wireless* dapat meningkatkan efisiensi komunikasi. Jika jumlah sensor *node* yang dipasang pada *greenhouse* tersebut cukup banyak maka sistem tersebut dapat membentuk *wireless sensor network*.

Berkaitan dengan arsitektur *wireless sensor network* terdapat konsep telemetri. Telemetri adalah proses pengukuran parameter suatu obyek (benda ruang, kondisi alam) yang hasil pengukurannya dikirimkan ke tempat lain melalui proses pengiriman data baik dengan menggunakan kabel maupun tanpa kabel (*wireless*) [8][9]. Multi-telemetri *wireless* merupakan gabungan dari beberapa telemetri yang hasilnya dikirim ke tempat lain menggunakan jaringan nirkabel. Multi-telemetri *wireless* diangkat dari teknologi *wireless sensor network* (WSN), yaitu suatu jaringan nirkabel yang terdiri dari kumpulan sensor *node* yang tersebar di suatu area tertentu (*sensor field*).

Penelitian yang telah dilakukan tentang pengembangan sistem telemetri *wireless* dilakukan pada "Perancangan Sistem Telemetri *Wireless* untuk Mengukur Suhu dan Kelembaban Berbasis Arduino Uno R3 ATmega328P dan Xbee Pro" [10]. Pada penelitian ini proses pengiriman data menggunakan modul Xbee Pro sebagai modul komunikasi dan Arduino Uno R3 ATmega328P sebagai sensor *node* yang menghubungkan modul WiFi dengan sensor. Pada penelitian yang telah dilakukan, ini penggunaan komponen Xbee Pro dirasa cukup mahal, apalagi untuk pengembangan sistem telemetri yang skalabilitasnya sangat tinggi dan membutuhkan sensor *node* yang cukup banyak.

Berdasarkan pembahasan dari penelitian sebelumnya, penelitian ini dikembangkan *prototype* sistem multi-telemetri *wireless* untuk mengukur suhu udara pada *greenhouse* dengan menggunakan sensor suhu DHT11 dan modul WiFi ESP8266 sebagai sensor *node* yang diimplementasikan pada *greenhouse*. Modul WiFi ESP8266 dipilih dalam penelitian ini karena memiliki kelebihan yaitu biaya yang jauh lebih rendah dibandingkan Xbee Pro dan dapat digunakan langsung untuk membaca data sensor tanpa menggunakan perangkat Arduino sehingga dapat menghemat konsumsi energi. Dalam penelitian ini juga dilakukan pengukuran kinerja ESP8266 apakah modul tersebut dapat berfungsi dengan baik ketika melakukan proses *sensing* dan pengiriman data. Sensor suhu akan dipasang di dalam beberapa *greenhouse* dan data yang dibaca oleh sensor kemudian dikirimkan ke *server* dengan menggunakan modul WiFi ESP8266 yang tersambung dengan sensor. *Server* menerima data yang dikirimkan oleh modul WiFi dan menyimpannya ke dalam *database* kemudian menampilkan data dalam bentuk *website*.

2. Metode Penelitian

Pada penelitian ini, sistem multi-telemetri digunakan untuk memantau suhu udara pada *prototype greenhouse* yang dapat dilakukan dari tempat yang berbeda. Sistem ini dapat memantau beberapa *greenhouse* sekaligus sehingga dapat mempermudah proses pemantauan. Pada sistem yang dibangun ini, proses pengambilan data suhu dan kelembaban udara digunakan sensor DHT11. Sedangkan untuk proses pengiriman data hasil sensor, digunakan modul WiFi ESP8266. Data hasil pembacaan sensor dikirim dari sensor *node* ke server dengan menggunakan protokol HTTP, selanjutnya data hasil pengiriman tersebut akan ditampilkan dalam bentuk aplikasi *web*.

Dalam penelitian ini juga dilakukan pengukuran kinerja dari modul WiFi ESP8266 yang digunakan sebagai media untuk mengirimkan data secara *wireless*, sekaligus sebagai sensor *node* yang melakukan pembacaan hasil proses *sensing* oleh sensor. Pengujian dilakukan pada beberapa hal meliputi analisis *heap memory*, pengaruh jangkauan atau jarak pengiriman ESP8266 terhadap keberhasilan pengiriman data, *data missing handling* ketika *server* tidak dapat diakses oleh sensor *node* baik karena jaringan yang tidak stabil maupun ketika *server* sedang

off, serta pengujian kestabilan jaringan. Selain itu pada penelitian ini juga dilakukan kalibrasi dengan membandingkan hasil data sensor DHT11 dengan alat pengukur suhu HTC-1.

Dalam skenario arsitektur sistem yang dibuat, terdapat tiga *greenhouse* yang masing-masing dipasang sensor suhu dan kelembaban udara menggunakan sensor DHT11 yang terhubung dengan modul WiFi ESP8266. Komponen utama yang terdapat pada arsitektur sistem, yaitu:

1. *Sensor node*

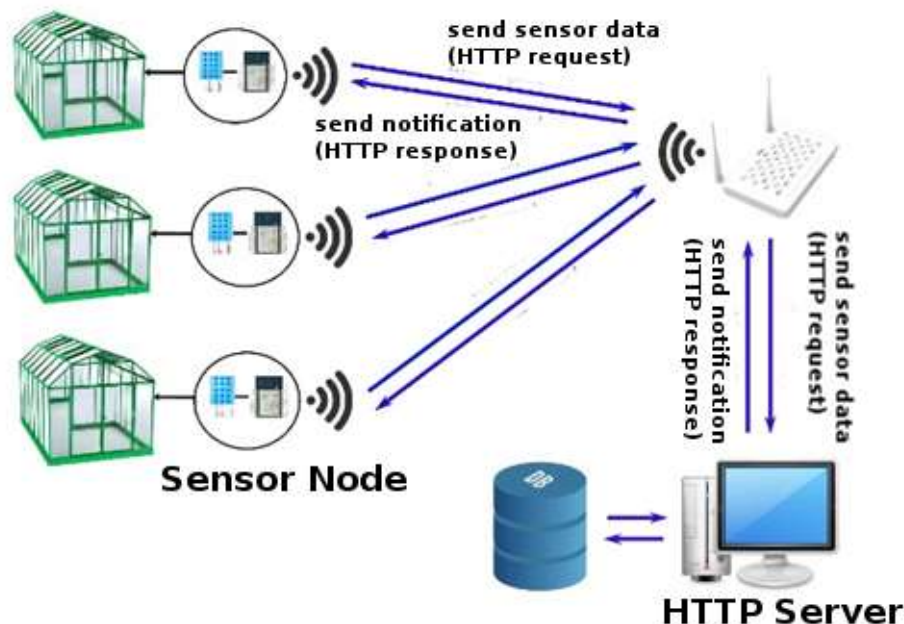
Sensor node terdiri dari sensor suhu dan kelembaban DHT11 dan modul WiFi ESP8266 sebagai media komunikasinya. *Sensor node* mengirim data hasil *sensing* dalam bentuk HTTP *request* ke *server* melalui protokol HTTP.

2. *Access point*

Access point digunakan sebagai perantara komunikasi antara *sensor node* dan *server*. Dalam skenario *system* digunakan *access point* agar *server* dapat terkoneksi ke beberapa *sensor node* sekaligus.

3. *Server*

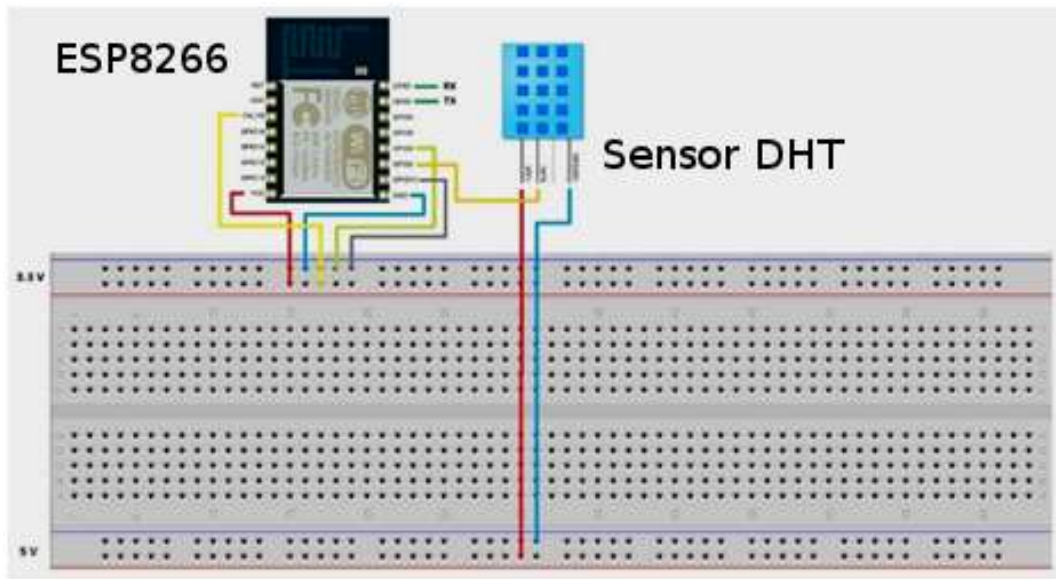
Server berfungsi untuk menerima dan mengolah data dari hasil *sensing* dari *sensor node* melalui protokol HTTP. Setelah berhasil menerima data sensor dari *sensor node*, *server* akan mengirimkan HTTP *response* ke *sensor node*, sebagai pemberitahuan bahwa proses pengiriman telah berhasil.



Gambar 1. Arsitektur Sistem

Gambar 1 menunjukkan arsitektur sistem yang diimplementasikan dalam penelitian ini. Modul WiFi ESP8266 yang juga sekaligus berperan sebagai *sensor node*, melakukan pembacaan nilai sensor DHT11, dan kemudian mengirimkan data hasil *sensing* tersebut melalui HTTP *request* ke komputer *server* dengan perantara *access point*. Setelah itu data akan dimasukkan ke dalam *database* yang terdapat pada HTTP *server*. HTTP *server* mengirimkan HTTP *response* kembali ke *sensor node* sebagai pemberitahuan bahwa HTTP *server* tersebut menerima data sensor. Ketika *sensor node* menerima HTTP *response* dari *server*, maka *sensor node* akan melakukan proses *sensing* kembali dan kemudian mengirimkan hasilnya melalui HTTP *request* ke *server*. Proses tersebut akan diulang selama proses *sensing* berlangsung.

Dalam arsitektur yang dibangun, diimplementasikan mekanisme *data missing handling* yaitu suatu mekanisme pada *sensor node* yang digunakan untuk mengatasi hilangnya data ketika *server* tidak bisa diakses oleh *sensor node*. Proses ini dilakukan dengan memanfaatkan memori di *sensor node* sebagai *buffer* dalam pengiriman data. *Buffer* adalah area *memory* di dalam ESP8266 yang digunakan untuk menyimpan data ketika data sedang dipindahkan antara dua titik, yaitu antara *sensor node* dengan *server*.

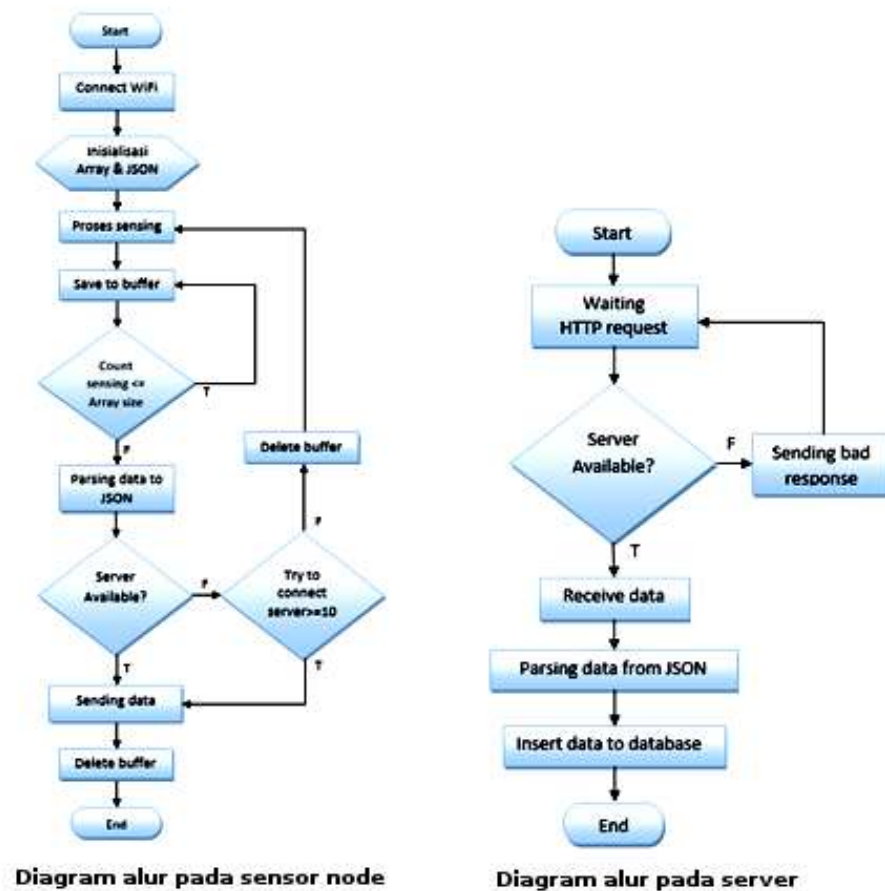


Gambar 2. Rangkaian Sensor Node

Gambar 2 merupakan rangkaian dari sensor *node* terdiri dari modul WiFi ESP8266 dan sensor suhu DHT11 yang diberi tegangan 3.3 Volt dan 5 Volt. Sensor DHT11 akan melakukan pembacaan suhu di dalam *greenhouse* secara berkala. Data dari sensor DHT11 kemudian dibaca oleh ESP8266 melalui Pin GPIO02. Setelah proses pembacaan data dari sensor DHT11 dilakukan, ESP8266 akan menyimpan data hasil *sensing* tersebut ke *array buffer* yang terdapat di dalam memori ESP8266. Secara periodik juga data hasil *sensing* yang ada di dalam *buffer* tersebut akan dikirimkan ke *server* melalui jaringan WiFi.

Gambar 3 merupakan diagram alur dari proses yang terdapat pada sensor *node* dan *server*. Setelah sensor *node* aktif maka sistem akan mencoba terhubung ke jaringan yang tersedia. Setelah itu sensor *node* melakukan proses inialisasi *array* yang digunakan untuk mekanisme *buffer* pada memori. Sensor *node* melakukan proses *sensing* yang hasilnya disimpan dalam *buffer* yang sudah diinisialisasi. Besarnya total data sensor yang disimpan ini dalam *buffer array* tersebut tergantung dari besar *buffer array* yang sudah diinisialisasi sebelumnya. Setelah isi *buffer* penuh, maka sensor *node* akan mengirimkan data hasil *sensing* ke *server*, dengan terlebih dahulu mengubah data ke dalam format JSON. Proses pengiriman data dilakukan dengan menggunakan HTTP *request*. Jika pengiriman berhasil, maka sensor *node* akan menerima HTTP *respons* dari *server*. Selanjutnya isi dari *buffer array* yang terdapat di dalam sensor *node* akan dikosongkan agar dapat digunakan untuk menyimpan kembali data hasil *sensing* berikutnya. Jika pengiriman gagal (sensor *node* tidak menerima HTTP *respons* dari *server*), maka sensor *node* akan mencoba mengirim ulang data hasil sensor sampai beberapa kali. Sementara data hasil sensor akan tetap disimpan dalam *buffer array*. Proses yang dilakukan ini merupakan mekanisme *data missing handling* yang diterapkan dalam sensor *node*. Jika selama beberapa kali proses pengiriman (dengan HTTP *request*) gagal, maka sensor *node* akan mengosongkan isi *buffer array*, dan memulai kembali proses *sensing* untuk mengambil data yang terbaru.

Ketika *server* aktif, maka *server* akan selalu berada pada kondisi *listening* untuk menunggu pengiriman data hasil *sensing* dari sensor *node* dalam bentuk HTTP *request*. Jika *server* dapat menerima HTTP *request* dari sensor *node*, maka *server* akan langsung mengirimkan HTTP *respons* ke sensor *node*, sebagai pemberitahuan bahwa *server* sudah menerima data hasil *sensing* dari sensor *node*. Untuk selanjutnya *server* melakukan parsing pada HTTP *request* untuk mengambil data hasil *sensing* dalam bentuk JSON. Kemudian data hasil parsing tersebut disimpan ke dalam *database*, untuk ditampilkan ke dalam *web monitoring*.



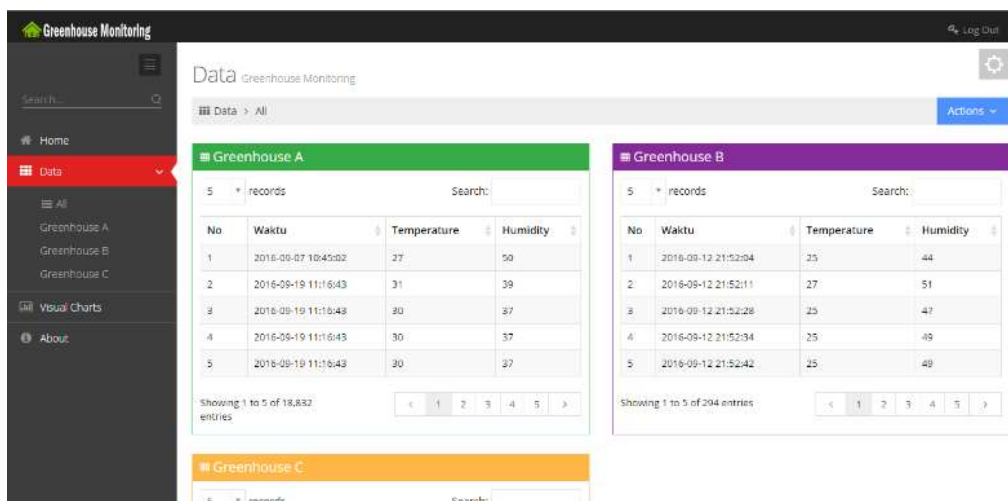
Gambar 3. Diagram Alur Pada Sensor Node dan Server

3. Hasil Penelitian dan Pembahasan

3.1 User Interface

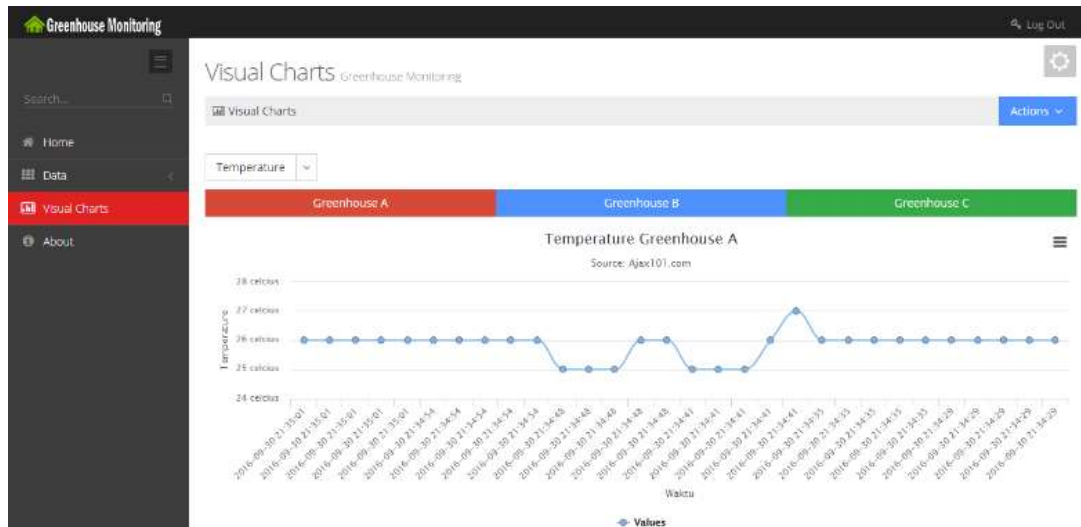
Gambar 4 dan Gambar 5 merupakan tampilan dari *website monitoring* yang menampilkan data yang dikirim oleh sensor *node*. Dalam penelitian yang dilakukan *website monitoring* ini dibangun dengan menggunakan Framework CodeIgniter, PHP dan MySQL.

1. Form tampil semua data



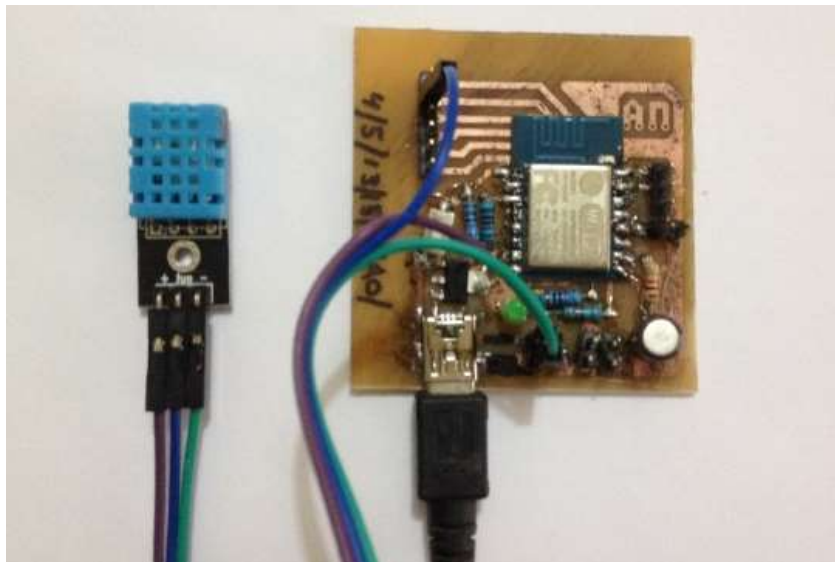
Gambar 4. Halaman Tampil Semua Data

2. Form grafik



Gambar 5. Halaman Tampil Grafik

3.2 Rangkaian Sensor Node



Gambar 6. Rangkaian Sensor Node

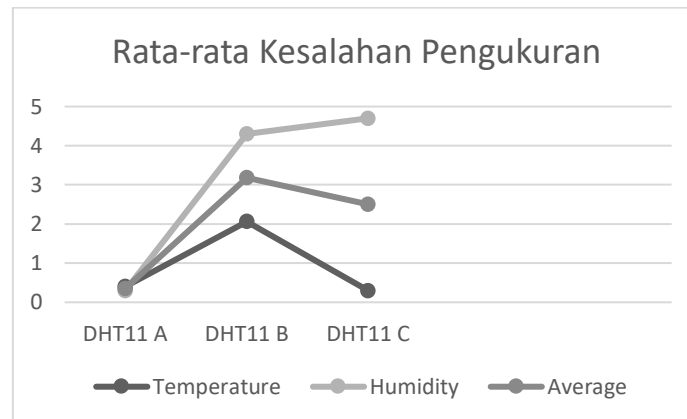
Implementasi dari rangkaian sensor *node* dapat dilihat pada Gambar 6. Rangkaian sensor *node* ini aktif ketika diberi tegangan 5 Volt, di dalam papan PCB tegangan dibagi menjadi 3.3 Volt dan 5 Volt. Sensor DHT11 memiliki 2 Pin, yaitu VCC, data dan *ground*. Pin yang digunakan untuk mengambil data sensor adalah Pin data yang disambungkan dengan Pin GPIO pada ESP8266.

3.3 Pengujian Sensor DHT11

Pengujian sensor DHT11 dilakukan untuk menguji apakah sensor DHT11 dapat membaca suhu dan kelembaban dengan baik. Pengujian sensor suhu dilakukan dengan membandingkan hasil pengukuran sensor DHT11 dengan *thermometer humidity digital* HTC-1. Pengujian ini dilakukan di dalam ruangan dengan mengambil sampel data setiap menit.

Hasil pengujian pada sensor DHT11 dapat dilihat pada Gambar 7. Data hasil pengujian tersebut merupakan perbandingan antara data sensor DHT11 dengan pengukur suhu HTC -1, yang didapatkan hasil yang berbeda tiap sensornya. Hasil yang paling rendah kesalahan ukurnya adalah DHT11 A dengan *temperature error* 0.41 °C dan *humidity error* 0.3%, sedangkan

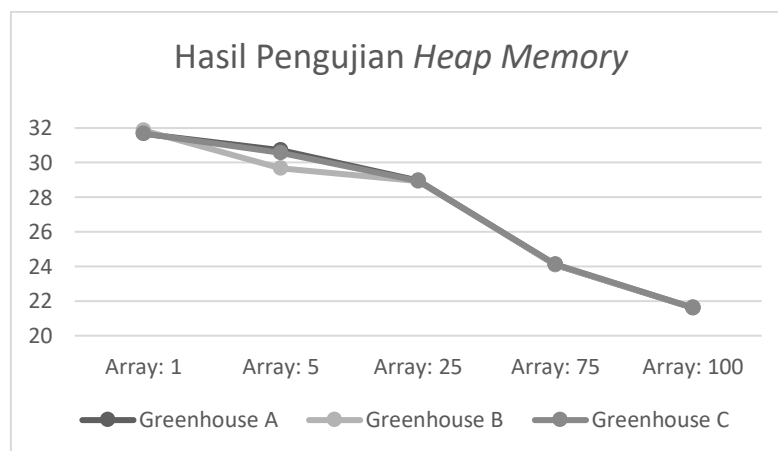
kesalahan ukur yang paling tinggi adalah DHT11 B dengan *temperature error* 2.6 °C dan *humidity error* 4.3%. Rata-rata kesalahan ukur suhu adalah 0.92 °C dan kelembaban adalah 3.1%.



Gambar 7. Rata-rata Kesalahan Ukur Sensor DHT11

3.4 Pengujian Heap Memory ESP8266

Pengujian *heap memory* dilakukan untuk mengetahui kemampuan modul WiFi ESP8266 dalam melakukan proses penyimpanan data. Modul WiFi ESP8266 mempunyai RAM size > 36kB yang digunakan untuk menjalankan program pada modul tersebut, serta digunakan untuk menyimpan data yang diproses oleh mikrokontroler. Parameter pengujian yang digunakan adalah besar *array* yang berfungsi sebagai *buffer* data pembacaan sensor. Dalam pengujian dilihat pengaruh besar *buffer* terhadap *heap memory* yang ada dalam ESP8266. Pengujian dilakukan dengan memvariasikan ukuran *array* yang digunakan untuk *buffer* data dan pengaruhnya terhadap *heap memory* yang kosong pada ESP8266.



Gambar 8. Hasil Pengujian Heap Memory ESP8266

Gambar 8 menunjukkan hasil pengujian *heap memory*. Dari hasil pengukuran *heap memory* pada ESP8266 tersebut dapat dilihat bahwa semakin besar ukuran *array* yang diinisialisasikan maka akan berpengaruh terhadap *heap memory* yang tersisa di dalam ESP8266. Dari ketiga pengujian, minimal *heap memory* yang tersisa di dalam RAM adalah 21612 Bytes yang dapat menampung hingga 100 data. Modul ESP8266 akan *error* dan *reset* otomatis ketika *heap memory* yang tersisa kurang dari nilai tersebut.

3.5 Pengujian Jarak atau Jangkauan ESP8266

Pengujian jarak atau jangkauan dilakukan untuk mengetahui seberapa jauh jangkauan modul WiFi ESP8266 dapat melakukan pengiriman data. Parameter yang digunakan adalah jarak antara *access point* dengan modul WiFi ESP8266. Hasil pengujian dilihat dari seberapa jauh jarak dan pengaruh dengan data pembacaan sensor dapat terkirim atau tidak. Pengujian dilakukan di tempat terbuka tanpa penghalang dinding.

Tabel 1. Pengujian Jarak ESP8266

Array	Jarak (m)	Greenhouse A		Greenhouse B		Greenhouse C	
		Freeheap	Ket	Freeheap	Ket	Freeheap	Ket
5	5	30296	Data terkirim	30712	Data terkirim	30328	Data terkirim
5	10	29752	Data terkirim	30808	Data terkirim	30136	Data terkirim
5	20	29192	Data terkirim	30712	Data terkirim	30512	Data terkirim
5	30	30680	Data terkirim	30712	Data terkirim	29944	Data terkirim
5	40	31560	Data terkirim	30760	Data terkirim	29944	Data terkirim
5	50	31168	Data terkirim	30520	Data terkirim	30328	Data terkirim
5	60	30920	Tidak Stabil	30808	Tidak Stabil	29944	Tidak Stabil
5	70	-	Data tidak terkirim	-	Data tidak terkirim	-	Data tidak terkirim

Hasil dari pengujian jarak dan jangkauan dapat dilihat pada Tabel 1. Pengujian jarak dari ketiga ESP8266 didapatkan hasil untuk jarak 5–50 meter data masih bisa terkirim. Sedangkan pada jarak 60 meter, beberapa data dapat terkirim, dan beberapa data tidak dapat terkirim. Sedangkan lebih dari 60 meter data tidak dapat terkirim sama sekali. Dari hasil pengujian juga dapat dilihat bahwa jarak atau jangkauan sensor *node* dan *server* tidak berpengaruh terhadap *free heap memory* pada ESP8266.

3.6 Pengujian Data Missing Handling

Mekanisme *data missing handling* dilakukan untuk mengatasi agar data tidak hilang ketika *server* sedang tidak tersedia atau tidak dapat diakses oleh sensor *node*. Skenario pengujian dilakukan dengan cara sensor *node* melakukan proses *sensing* selama 10 menit, kemudian jika *server* tidak tersedia dan data tidak dapat terkirim maka sensor *node* akan mencoba mengirim data berulang sebanyak 10 kali. Ketika percobaan pengiriman ke 10 *server* masih *unavailable* atau data masih belum terkirim, maka sensor *node* akan menghapus data sebelumnya dan akan menggantinya dengan data yang baru dengan proses *sensing* ulang.

Tabel 2 merupakan hasil pengujian *data missing handling* ketika *server* tidak tersedia dengan menunjukkan proses yang sedang berjalan dalam *server* dan sensor *node*. Adanya mekanisme *data missing handling*, data hasil pembacaan sensor tidak hilang ketika *server* sedang tidak tersedia. Data hasil pembacaan sensor tersimpan pada *buffer* dan akan terkirim ketika *server* aktif. Data otomatis akan dihapus dan diganti dengan yang baru ketika *server* masih tidak aktif dalam jangka waktu tertentu (dalam penelitian ini dibuat 10 menit) sehingga ketika *server* aktif maka data terbaru yang akan dikirim ke *server*. Hal ini dimaksudkan agar data tidak hilang ketika *server* *unavailable* atau sedang tidak dapat diakses oleh sensor *node*.

Dalam penelitian ini juga dilakukan pengujian terhadap *free heap memory* ESP8266 dan dari hasil pengujian, dapat dilihat bahwa proses sensor *node* untuk mencoba terhubung dengan jaringan secara berulang tidak berpengaruh terhadap *heap memory* ESP8266.

3.7 Pengujian Kestabilan Jaringan

Pengujian kestabilan jaringan dilakukan untuk mengetahui pengaruh jaringan terhadap pengiriman data. Parameter pengujian yang digunakan adalah jumlah *access point* yang sedang berkomunikasi di sekitar *access point server* dan *channel* yang digunakan. Dalam pengujian dilihat pengaruh banyaknya *access point* yang berkomunikasi dengan *channel* yang sama, dengan usaha sensor *node* untuk mengirim data ke *server* (percobaan terhubung ke *server* dengan jumlah 10 kali percobaan tiap pengiriman data).

Pengujian dilakukan pada jarak 10 meter, besar *array* 5 dan *delay* 1000 dengan dua kondisi. Kondisi pertama, yaitu proses pengiriman data dilakukan pada lingkungan yang terdapat kurang dari 10 *access point* yang sedang berkomunikasi sekitar *access point server* dan 2 diantaranya menggunakan *channel* yang sama dengan *access point server* (*channel* 11+7).

Kondisi kedua, yaitu proses pengiriman data dilakukan pada lingkungan yang terdapat lebih dari 10 *access point* yang sedang berkomunikasi disekitar *access point server* dan 5 diantaranya menggunakan *channel* yang sama dengan *access point server* (*channel* 11+7). Hasil pengujian kestabilan jaringan dapat dilihat pada Tabel 3.

Dari hasil pengujian kestabilan jaringan didapatkan hasil untuk kondisi satu, yaitu data dari ketiga ESP8266 dapat terkirim dengan satu kali percobaan pengiriman data ke *server*. Sedangkan pada pengujian kondisi dua pengiriman data tidak stabil dan ada yang tidak terkirim. Dari hasil pengujian dapat dilihat bahwa semakin banyak jumlah *access point* yang sedang berkomunikasi sekitar *access point server* dengan menggunakan *channel* yang sama maka akan berpengaruh terhadap kestabilan pengiriman data atau koneksi sensor *node* dengan *server*.

Tabel 2. Pengujian Data Missing Handling Ketika Server Unavailable

Menit ke	Server	Sensor node	Free heap
1	Available	Proses sensing	30448
10	Available	Mengirim data (Data terkirim)	30448
11	Available	Proses sensing	30256
20	Unavailable	Mengirim data (Data tidak terkirim)	30256
21	Unavailable	Try to connect server 1	30880
22	Unavailable	Try to connect server 2	30880
23	Unavailable	Try to connect server 3	30880
24	Unavailable	Try to connect server 4	30880
25	Unavailable	Try to connect server 5	30880
26	Unavailable	Try to connect server 6	30880
27	Unavailable	Try to connect server 7	30880
28	Unavailable	Try to connect server 8	30880
29	Unavailable	Try to connect server 9	30880
30	Unavailable	Try to connect server 10	30880
31	Unavailable	Proses sensing	31088
40	Unavailable	Mengirim data (Data tidak terkirim)	31088
40	Unavailable	Try to connect server 1	31088
41	Unavailable	Try to connect server 2	31088
42	Unavailable	Try to connect server 3	31088
43	Available	Mengirim data (Data terkirim)	31088
44	Available	Proses sensing	30280

Tabel 3. Pengujian Kestabilan Jaringan

Pengiriman Data	Jumlah Percobaan Sensor Node Terhubung ke Server					
	ESP8266 A		ESP8266 B		ESP8266 C	
	Kondisi 1	Kondisi 2	Kondisi 1	Kondisi 2	Kondisi 1	Kondisi 2
Data 1	1	1	1	10	3	5
Data 2	1	4	1	10	1	1
Data 3	1	1	4	10	1	1
Data 4	1	10	1	10	1	1
Data 5	1	10	1	10	1	5

4. Kesimpulan

Berdasarkan proses observasi, analisis dan implementasi pada tugas akhir tentang sistem multi-telemetri *wireless* untuk mengukur suhu udara, penulis membuat beberapa kesimpulan, yaitu:

1. Sistem multi-telemetri *wireless* untuk mengukur suhu udara dirancang dengan menggunakan sensor suhu DHT11, modul WiFi ESP8266 sebagai media komunikasi dan HTTP protokol untuk protokol komunikasinya.
2. Aplikasi berbasis web dirancang dengan menggunakan CodeIgniter, PHP dan MySQL untuk menampilkan data hasil pembacaan sensor yang dikirim oleh sensor *node*.

3. Dari hasil pengujian didapatkan bahwa sensor DHT11 memiliki rata-rata kesalahan ukur suhu adalah 0.92 °C dan kelembaban adalah 3.1%. Sedangkan modul WiFi ESP8266 mampu menyimpan dan mengirim *buffer* hingga 100 data dan dapat melakukan pengiriman dalam jangkauan 50 meter. Pengujian data *missing handling* memanfaatkan *buffer* untuk menyimpan data selama server sedang tidak dapat diakses oleh sensor *node* untuk mengurangi risiko hilangnya data. Kestabilan pengiriman data atau koneksi sensor *node* dengan *server* dipengaruhi oleh jumlah *access point* yang sedang berkomunikasi sekitar *access point server* dengan menggunakan *channel* yang sama. Dalam hal ini berjalannya sistem bergantung pada kestabilan jaringan, rangkaian sensor *node* dan hubungan antara sensor *node* dan server.

Referensi

- [1] H. Zhang, H. Xue, "Design of Greenhouse Environmental Parameters Prediction System," in International Conference on Computer and Computing Technologies in Agriculture. Springer, 2010, Pp. 502–507.
- [2] D. Risqiwati, "Rancang Bangun Sistem Monitoring Listrik Prabayar dengan Menggunakan Arduino Uno," KINETIK, Vol. 1, No. 2, Pp. 47–54, 2016.
- [3] A. Maarif, "System Monitoring And Controlling Water Nutrition aquaponics Using Arduino Uno Based Web Server," KINETIK, Vol. 1, No. 1, 2016.
- [4] K. . Bhoomika, K. N. Muralidhara, "Secured Smart Healthcare Monitoring System Based on IOT," International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 3, No. 7, 2015.
- [5] D. Chaudhary, S. Nayse, "Application of wireless sensor networks for greenhouse parameter control in precision agriculture," International Journal of Wireless & Mobile Networks (IJWMN), Vol. 3, No. 1, Pp. 140–149, 2011.
- [6] M. S. Munir, "Rancangan Smart Greenhouse dengan Teknologi Mobile untuk Efisiensi Tenaga, Biaya dan Waktu dalam Pengelolaan Tanaman," Universitas Pembangunan Nasional Veteran, 2010.
- [7] S. Sawidin, O. Melo, T. Marsela, "Monitoring Kontrol Greenhouse untuk Budidaya Tanaman Bunga Krisan dengan LabView," Jurnal Nasional Teknik Elektro dan Teknik Informatika, Vol. 4, No. 4, 2015.
- [8] I. Pambudi, Y. Nugraha, M. Djamal, "Sistem Telemetry Pemantau Gempa Menggunakan Jaringan GSM," Jurnal Otomasi, Kontrol, dan Instrumentasi ITB, Vol. 4, No. 1, Pp. 47, 2012.
- [9] M. Wastharini, D. Arseno, I. Hidayat, "Perancangan dan Implementasi Sistem Telemetry Suhu Ruang Berbasis Mikrokontroler," Universitas Telkom Bandung, 2010.
- [10] H. Susanto, R. Pramana, "Perancangan Sistem Telemetry Wireless untuk Mengukur Suhu dan Kelembaban Berbasis Arduino Uno R3 ATmega328p dan XBee Pro," Universitas Maritim Raja Ali Tanjung Pinang, 2013.