

Implementasi *Wireless Sensor Node* Sebagai Pendukung Pertanian Modern Berbasis Pemrograman *State Machine*

Rivaldi Wibowo¹, Sabriansyah Rizqika Akbar², Bayu Priyambadha³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹rivaldiwe@gmail.com, ²sabrian@ub.ac.id, ³bayu_priyambadha@ub.ac.id

Abstrak

Pada saat ini sistem *monitoring* yang dilakukan oleh petani masih menggunakan prosedur manual, yaitu melakukan pengecekan secara langsung ke lapangan. Pada saat pengecekan secara manual tersebut bisa terjadi kesalahan pada saat *input* data seperti kesalahan penulisan dikarenakan banyaknya data. Dari permasalahan tersebut, maka diperlukan sistem yang secara periodik melakukan pengecekan pada keadaan tanaman. Oleh sebab itu digunakanlah WSN sebagai sarana pengecekan secara nirkabel untuk mempermudah dalam melakukan *monitoring* tanaman dan metode *state machine* yang digunakan sebagai metode yang dipakai dalam penelitian. Pada penelitian ini digunakan beberapa sensor antara lain, sensor suhu, sensor cahaya, dan sensor tanah. Pada penelitian ini metode *state machine* digunakan sebagai metode yang berguna untuk melihat tingkah laku sistem yang telah dibuat peneliti. Pengujian pada sistem yang dibuat dilakukan sebanyak 6 kali yaitu pengujian pada eksekusi setiap *state*, pengujian lama eksekusi transisi, pengujian total waktu eksekusi *state* dan transisi, pengujian sensor, pengujian pengiriman data, serta pengujian *sleep mode*. Sehingga hasil dari implementasi metode *state machine* berhasil dilakukan dengan sistem dapat berjalan. Dalam penelitian ini hasil rata-rata dari pengujian *state* sebanyak 10 kali dari total transisi *state* “diam” adalah 21543,6 *microsecond*, *state* “termonitor” 921594,4 *microsecond*, dan “terkirim” sebesar 1161218,4 *microsecond*. Kemudian hasil dari sensor yang diuji sesuai dengan yang diharapkan serta data yang masuk sesuai dengan tampilan serial monitor. Untuk pengujian *sleep mode* tidak sesuai dengan pengaturan yang telah dilakukan dikarenakan adanya faktor *delay* pada program yang mempengaruhi hasil keluaran sistem menjadi lebih lama dari waktu yang telah ditentukan.

Kata kunci : WSN, *State Machine*, Pertanian

Abstract

At this time, the monitoring system conducted by farmers still using manual procedures, namely to check directly to the field. At the time of manual checking can occur error at the time of input data such as writing errors due to the number of data. From these problems, it is necessary that the system periodically checks on the state of the plant. Therefore WSN is used as a means of checking wirelessly to facilitate the monitoring of plants and state machine methods used as a method used in research. In this research used several sensors, among others, temperature sensors, light sensors, and soil sensors. In this research the state machine method is used as a useful method to see the behavior of systems that have been made by researchers. Tests on the system made done as much as 6 times that is ,times the test on the execution of each state, testing the length of transition execution, testing the total state execution time and transition, sensor testing, data delivery testing, and sleep mode test. So the result of the implementation of state machine method successfully done with the system running. In this Research the average results of state testing as much as 10 times the total state station transition is state “diam” 21543.6 microsecond, state “termonitor” 921594.4 microsecond, and “terkirim” 1161218.4 microsecond. Then the results of the tested sensors in accordance with the expected and incoming data in accordance with the display serial monitor. For sleep mode test is not in accordance with the settings that have been done due to a delay factor in the program that affects the of the system becomes longer than the time specified.

Keywords: WSN, *State Machine*, Agriculture

1. PENDAHULUAN

Sektor pertanian di Indonesia saat ini perlu adanya perkembangan ke arah yang lebih baik. Seperti adanya perubahan pada sistem *monitoring* pada tanaman agar dapat menghemat waktu pengecekan tanaman. Menurut Gerard pengambilan sampel data secara manual bisa terjadi kesalahan pengambilan data. Oleh sebab itu perlu adanya sistem untuk *monitoring* tanaman agar keadaan pertumbuhan tanaman terjamin. Dengan adanya sebuah sistem yang dapat melakukan *monitoring* tanaman akan menjadikan proses pengambilan data menjadi lebih efisien. (Mendez, et al., 2011)

Kemajuan teknologi pada saat ini memungkinkan *monitoring* tanaman menggunakan sensor. Penggunaan sensor pada pertanian menggunakan sensor seperti sensor cahaya, tanah, udara, serta kelembaban. Fungsi dari masing masing sensor tersebut sangat berperan besar untuk mendapatkan informasi keadaan tanaman secara akurat. Keadaan tanaman yang telah diperoleh oleh sensor perlu didistribusikan secara nirkabel untuk mengakses data keadaan tanaman. Oleh sebab itu diperlukan adanya penerapan *Wireless Sensor Network* (WSN) pada pertanian agar mempermudah melakukan pengawasan tanaman.

Wireless Sensor Network merupakan sistem nirkabel yang menghubungkan beberapa perangkat. Kelebihan dari WSN adalah kemampuan untuk tetap berfungsi pada keadaan lingkungan yang buruk sekalipun. Pada saat ini WSN telah banyak digunakan pada bermacam macam bidang termasuk pertanian. Pada WSN terdiri dari kumpulan kecil beberapa perangkat yang disebut node yang berfungsi mengirim dan menerima sinyal yang spesifik berdasarkan sensor yang digunakan dan berdasarkan dari pemantauan keadaan lingkungan dari sensor tersebut. (Garzón & Riveros, 2010)

Pada penelitian yang akan dilakukan, peneliti menggunakan metode *state machine*. *State Machine* adalah permodelan perangkat yang memiliki jumlah *state* yang terbatas dan dapat digunakan pada waktu tertentu untuk *input* ataupun melakukan transisi dari *state* pertama ke *state* selanjutnya. Tujuan utama dari metode *state machine* adalah untuk menggambarkan tingkah laku suatu sistem. Dengan menggunakan metode *state machine* dapat mempermudah pengaturan perangkat dan mempermudah untuk menerapkan kedalam sistem yang akan dibuat.

(Setiawan, 2006)

Berdasarkan keterangan yang telah disampaikan diatas maka peneliti mengusulkan sebuah penelitian berjudul Implementasi *Wireless Sensor Node* sebagai pendukung pertanian modern berbasis pemrograman *State Machine*. Pada sistem yang akan dibuat diterapkan metode *state machine* yang memudahkan untuk melihat tingkah laku sebuah sistem yang dibuat. Sistem tersebut menggunakan module *wireless* yang nantinya data akan dapat dikirim melalui udara tanpa perantara. Data tersebut dapat diakses user menggunakan browser pada smartphone.

2. DASAR TEORI

2.1 Arduino UNO

Arduino merupakan papan rangkaian elektronik yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan AVR dari perusahaan Atmel yang bersifat open source, fleksibel, dan mudah digunakan dalam hal perangkat keras dan perangkat lunak, dari sisi perangkat kerasnya Arduino menggunakan prosesor Atmel AVR (Atmega8 atau Atmega168) dan memiliki pin yang dapat digunakan sebagai input maupun output. Sedangkan dari sisi perangkat lunaknya, Arduino menggunakan bahasa pemrograman umum dan tersedia boot-loader yang berjalan pada modulnya

Bahasa pemrograman Arduino adalah bahasa C tetapi bahasa ini sudah dipermudah menggunakan *library-library* sehingga pemula pun bisa mempelajarinya dengan cukup mudah. Papan Arduino UNO menggunakan mikrokontroler Atmega328P. Papan ini memiliki 14 pin input/output digital(enam diantaranya untuk output PWM), enam buah input analog, 16 MHz crystal ocillator, sambungan USB, ICSP header, dan tombol reset. Dalam penggunaannya cukup dengan menghubungkan kabel USB atau dengan memberikan daya menggunakan adapter AC ke DC dengan baterai. Gambar 2.1 menunjukkan tampilan papan Arduino dari atas.

2.2 Sensor LDR

Light Dependent Resistor atau disingkat LDR adalah jenis Resistor yang nilai hambatan atau nilai resistansinya tergantung pada intensitas cahaya yang diterimanya. Nilai hambatan LDR akan menurun pada saat cahaya

terang dan nilai hambatannya akan menjadi tinggi jika dalam kondisi gelap. Dengan kata lain, fungsi LDR (*Light Dependent Resistor*) adalah untuk mengantarkan arus listrik jika menerima sejumlah intensitas cahaya (kondisi terang) dan menghambat arus listrik dalam kondisi gelap.

Naik turunnya nilai hambatan akan sebanding dengan jumlah cahaya yang diterimanya. Pada umumnya, nilai hambatan LDR akan mencapai 200 Kilo Ohm ($k\Omega$) pada kondisi cahaya terang.

2.3 Modul Wireless ESP8266

ESP8266 merupakan modul wifi yang memiliki fungsi sebagai perangkat tambahan mikrokontroler agar dapat terhubung langsung dengan wifi dan membuat koneksi TCP/IP. Modul ini membutuhkan daya sekitar 3.3V dengan memiliki tiga mode wifi yaitu *Station*, *Access Point* dan *Both* (keduanya). Modul ini juga dilengkapi dengan prosesor, memori dan GPIO dimana jumlah pin bergantung dengan jenis ESP8266 yang digunakan.

2.4 Sensor Tanah

Sensor kelembaban tanah yang digunakan merupakan kombinasi dari YL-38 (modul pengkondisi sinyal) dan YL-69 (probe sensor). Modul sensor ini memiliki 4-pin, yaitu GND (untuk *ground*), VCC (3.3-5V), AO (keluaran analog yang akan dibaca oleh Arduino), dan DO (keluaran digital) dapat diatur sensitivitasnya menggunakan knob pengatur, dan menghasilkan logika digital *HIGH/LOW* pada level kelembaban tertentu.

2.5 RTC

Real Time Clock merupakan suatu chip (IC) yang memiliki fungsi sebagai penyimpan waktu dan tanggal. Dalam penelitian ini akan menggunakan jenis DS1307. DS1307 merupakan *Real-time clock (RTC)* dengan jalur data parallel yang memiliki antarmuka serial Two-wire (I2C), sinyal luaran gelombang-kotak terprogram (*Programmable Squarewave*), deteksi otomatis kegagalan daya (*power-fail*) dan rangkaian *switch*, konsumsi daya kurang dari 500nA menggunakan mode baterai cadangan dengan operasional oscillator. Tersedia fitur industri dengan ketahanan suhu: -40°C hingga $+85^{\circ}\text{C}$. Tersedia dalam kemasan 8-pin DIP atau SOIC

2.6 Sensor DHT11

DHT11 merupakan sensor digital yang dapat mengukur suhu dan kelembaban udara di sekitarnya. Sensor ini sangat baik serta fitur kalibrasi yang akurat. Koefisien kalibrasi disimpan dalam OTP program memory, sehingga ketika internal sensor mendeteksi sesuatu, maka modul ini menyediakan koefisien tersebut dalam kalkulasinya.

DHT11 termasuk sensor yang memiliki kualitas terbaik, dinilai dari respon, pembacaan data yang cepat dan kemampuan anti-interferensi. Ukurannya yang kecil dan dengan transmisi sinyal hingga 20 meter membuat produk ini cocok digunakan untuk banyak aplikasi pengukuran suhu dan kelembaban.

2.7 Watchdog Timer

Watchdog timer merupakan timer yang terpisah dengan sistem CPU mikrokontroler sehingga tidak akan terpengaruh bagaimanapun keadaan CPU bekerja. *Watchdog timer* berperan sangat penting untuk kelangsungan hidup CPU dan menjaga CPU agar tetap ON karena dapat mengatasi kemacetan alur program pada mikrokontroler, biasanya dalam PC kemacetan ini kita sering menyebutnya "*hang up*". Kemacetan program ini biasa disebabkan oleh beberapa hal baik itu dari segi hardware maupun dari softwrenya sendiri. Prinsip dari *watchdog timer* ini sama seperti timer lain yang selalu mencacah naik (*count up*) setiap satu siklus mesin, namun bedanya dengan timer lain *watchdog timer* ini akan mencacah terus menerus tanpa henti sekalipun kondisi CPU macet karena seperti dikatakan bahwa timer ini terpisah dengan CPU. Setelah register *watchdog* maksimum maka akan terjadi *overflow* dan secara otomatis alur program pada mikrokontroler akan tereset ke awal

2.8 Library FSM

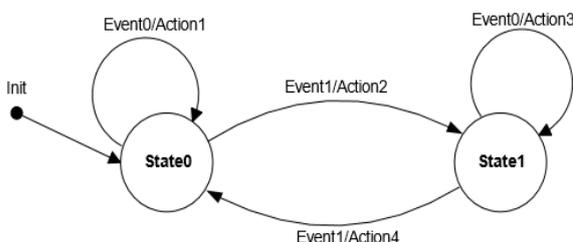
Merupakan *library* yang dipasang dalam Arduino IDE untuk membuat sebuah *state machine*. Pada *library* ini terdapat 2 class yaitu *state* dan *fsm*. *State Machine* merupakan metode perancangan sistem kontrol yang menggambarkan sebuah tingkah laku dari sistem yang dijalankan. Dalam sistem yang menggunakan *state machine* terdapat 3 prinsip kerja yaitu *State* (Keadaan), *Event* (Kejadian), dan *Action* (aksi). Untuk menggunakan *library* FSM perlu dilakukan insialisasi terlebih dahulu yaitu dengan mengetikkan "`#include`

<Fsm.h>”. Dengan melakukan inialisasi *library* kita dapat memanggil *library* tersebut untuk menerapkan metode *state machine* yang terdapat dalam *library*.

Pada *library* FSM untuk memulai pembuat sebuah *state* pertama tama kita perlu membuat Fsm dan *state* diatas kode program utama. Kemudian dalam fungsi Setup() perlu kita buat transisi yang berfungsi untuk perpindahan antar *state* . Dalam traansisi jugaa perlu dibuat sebuah trigger untuk digunakan untuk mebuat sebuah *state* berpindah ke *state* berikutnya. Setelah pembuatan transisi maka selanjutnya adalah pembuatan sebuah *state*. Pada *library* FSM pada setiap *state* terdapat *enter* dan *exit* . terakhir adalah mendefinisikan trigger pada menu loop() dan *state* dapat digunakan.

2.9 Finite State Machine

State machine di representasikan dalam bentuk diagram. Dalam diagram tersebut secara visual digambarkan tingkah laku yang dimiliki oleh sistem kontrol yang kompleks kedalam bentuk yang lebih sederhana dan relative mudah. Dalam diagram , *state-state* yang terdapat pada sebuah sistem digambarkan sebagai lingkaran yang diberi label unik , sedangkan transisi *state* yang diakibatkan oleh *event* tertentu direpresentasikan sebagai anak panah yang berasal dari *state* yang ditinggalkan menuju *state* yang aktif. Setiap transisi yang terjadi umumnya juga di ikuti oleh aksi yang dilakukan oleh sistem yang dirancang. Secara praktis setiap diagram *state* yang dirancang akan selalu memiliki sebuah transisi awal (inisial) yang menuju salah satu *state* sejak sistem kontrol tersebut dihidupkan. Gambar 1 berikut memperlihatkan contoh penggambaran diagram *state* :



Gambar 1 Contoh Diagram *State* Sederhana (Sumber : Setiawan ,2006)

Diagram pada gambar 1 tersebut memperlihatkan FSM dengan dua buah *state* dan dua buah input serta empat buah aksi output yang berbeda : seperti terlihat pada gambar,

ketika sistem mulai dihidupkan, sistem akan bertransisi menuju *state0*, pada keadaan ini sistem akan menghasilkan *Action1* jika terjadi masukan *Event0*, sedangkan jika terjadi *Event1* maka *Action2* akan dieksekusi kemudian sistem selanjutnya bertransisi ke keadaan *State1* dan seterusnya. (Setiawan, 2006)

3. PERANCANGAN DAN IMPLEMENTASI

3.1 Perancangan Perangkat Keras

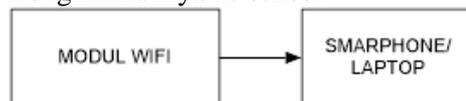
Tahap perancangan konsep sistem ini bertujuan agar perancangan sistem penelitian yang dilakukan menjadi terstruktur. Perancangan sistem pada penelitian ini seperti pada Gambar 2 berikut :



Gambar 2 Diagram Block *Node Client*

Berdasarkan Gambar 2 perancangan sistem dapat dijelaskan sebagai berikut :

1. Setiap sensor melakukan sensing dan mengirimkan pada mikrokontroler
2. Mikrokontroler mengolah data yang didapatkan dari sensor-sensor
3. Modul wifi menerima kumpulan data dan mengirimkannya ke sensor



Gambar 3 Diagram Block *Node Server*

Berdasarkan Gambar 3 perancangan sistem dapat dijelaskan sebagai berikut :

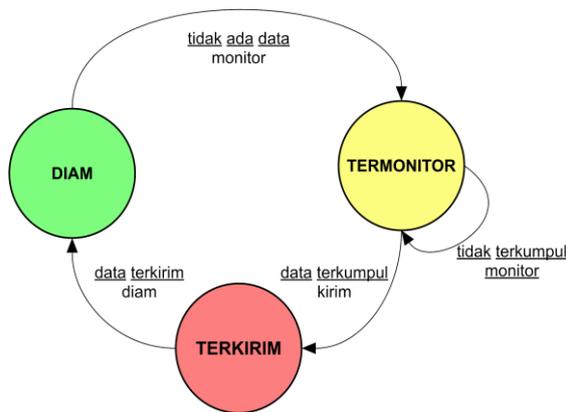
1. Modul *Wifi Server* menerima data dari *wifi* modul client
2. *Wifi* Modul *server* menyimpan data tersebut dan mengirimkan data tersebut ke laptop atau *smartphone*.

3.2 Perancangan Perangkat Lunak

3.2.1 Perancangan State Machine

Perancangan *state* berdasarkan dari penelitian yang sedang dilakukan. *State* yang dibuat terdiri dari 3 *state* dan 4 transition. Masing masing *state* memiliki tugas sendiri dalam sistem yang dibuat. DI dalam 4 transition yang dibuat masing masing memiliki trigger yang berbeda untuk melakukan perpindahan antar *state*. Berikut gambar dari *state* yang

dibuat.

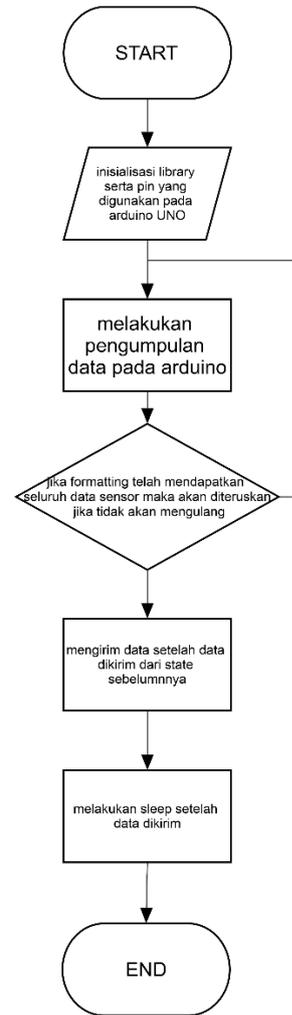


Gambar 4 Perancangan State Machine

Pada Gambar 4 dapat dilihat bahwa state terdiri dari 3 yaitu diam, termonitor, dan terkirim. Masing masing state tersebut memiliki tugas tersendiri di dalam sistem. Di dalam sistem state machine diterapkan pada Node Client.

3.2.2 Perancangan Pengiriman Data Node Client

Node Client berfungsi sebagai pengirim data yang bertugas mengakuisisi data yang didapatkan dari sensor-sensor. Pengiriman data dilakukan oleh modul wireless ESP8266. Berikut merupakan tampilan diagram alir pada Node Client.

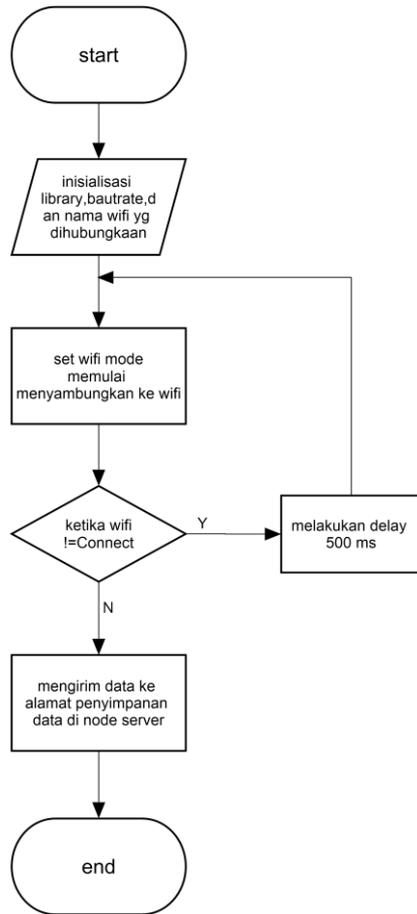


Gambar 5 Diagram Alir Pengiriman Data ESP8266

Pada gambar 5 diatas dijelaskan bahwa proses pengiriman data pada sisi client diawali dengan melakukan proses inisialisasi. Proses inisialisasi ini meliputi inisialisasi baudrate, pin sensor, dan channel. Inisialisasi baudrate berfungsi untuk menentukan mikrokontroler yang digunakan bekerja pada rate yang telah ditentukan. Inisialisasi pin sensor berfungsi untuk menentukan sensor bekerja pada pin yang telah ditetapkan. Kemudian melakukan proses menghubungkan ke alamat server. Setelah terhubung data dikirimkan ke server.

3.2.3 Perancangan penerimaan Node Server

Node Server merupakan sebuah node yang berfungsi sebagai penerima data yang dikirimkan oleh node client. Alur kerja server dapat dilihat pada Gambar 6 di bawah ini.



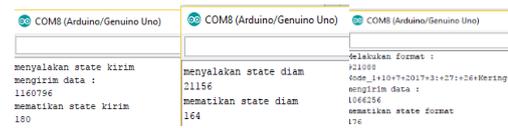
Gambar 6 Diagram Alur Penerimaan Data Node Server

Setelah Server aktif maka root akan dapat menerima data yang dikirimkan oleh client yang kemudian disimpan di dalam folder bernama simpan data yang kemudian dapat diakses oleh user menggunakan smartphone dengan web browser.

4. PENGUJIAN DAN ANALISIS

4.1 Pengujian Eksekusi State Machine

Metode state machine pada penelitian ini diterapkan pada Node Client yang berfungsi untuk mengatur sensor sensor yang terhubung pada arduino. Untuk prosedur pengujian, pertama yaitu menguji eksekusi setiap state yang ada pada Node Client. Kemudian menguji setiap transisi yang dilakukan oleh state. Hasil pengujian dapat dilihat melalui serial monitor. Pada gambar 7 dapat dilihat tampilan saat pengujian dilakukan.



Gambar 7 Tampilan Perhitungan State

Gambar 7 merupakan tampilan dari serial monitor yang menampilkan hasil dari state diam yang lama eksekusinya yaitu menambahkan waktu eksekusi diam enter dan diam exit. Lama dari state diam tersebut ditampilkan dalam mikrosecond. State yang diuji merupakan 3 buah state yang telah dibuat pada saat perancangan sistem.

Tabel 1 Tampilan Hasil Eksekusi

	STATE	P-1(ms)	P-2(ms)	P-3(ms)	P-4(ms)	P-5(ms)	P-6(ms)	P-7(ms)	P-8(ms)	P-9(ms)	P-10(ms)
Diam	enter	21160	21152	21152	21168	21160	21164	21156	21152	21148	21156
	exit	164	164	168	168	172	172	164	160	164	164
	total	21324	21316	21320	21336	21332	21336	21320	21312	21312	21320
Ternom tor	enter	921080	921068	921092	921632	921076	921080	922104	921076	921080	920612
	exit	176	176	176	176	176	176	176	180	176	180
	total	921256	921244	921268	921808	921252	921256	922280	921256	921256	920792
Terkirim	enter	1160820	1160824	1160820	1160820	1160820	1160832	1160832	1160832	1160828	1160832
	exit	180	176	180	180	176	176	180	184	176	184
	total	1161000	1161000	1161000	1161000	1160996	1161008	1161012	1161016	1161004	1161016

Pada tabel 1 merupakan kumpulan data eksekusi sebanyak 10 kali pada setiap state. Data tersebut digunakan sebagai pembandingan eksekusi pertama hingga eksekusi state ke 10. Pada setiap state ditampilkan dalam format waktu mikrosecond. Hal ini dilakukan agar tampilan waktu lebih detail untuk mempermudah melihat perbedaan waktu setiap eksekusi state. Pada tabel 1 pengujian pada Diam dilakukan tanpa adanya sleep mode yang digunakan pada saat dilapangan. Hal ini dilakukan agar mempercepat pengujian sistem.

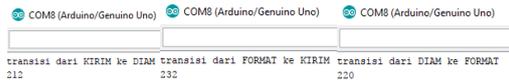
Tabel 2 Tampilan Hasil Konversi Data

	ms	1.161	1.161	1.161	1.161	1.160996	1.161008	1.161012	1.161016	1.161004	1.161016
DIAM	s	0,0021324	0,0021316	0,002132	0,0021336	0,0021332	0,0021336	0,002132	0,0021312	0,0021312	0,002132
	ms	1,182324	1,182316	1,18232	1,182336	1,182328	1,182344	1,182332	1,182328	1,182316	1,182336
TERNOM TOR	s	0,921256	0,921244	0,921268	0,921808	0,921252	0,921256	0,92228	0,921256	0,921256	0,920792
	ms	2,10358	2,10356	2,103588	2,104144	2,10358	2,1036	2,104612	2,103584	2,103572	2,103128
TERKIRIM	s	1,161	1,161	1,161	1,161	1,160996	1,161008	1,161012	1,161016	1,161004	1,161016
	ms	1,161	1,161	1,161	1,161	1,160996	1,161008	1,161012	1,161016	1,161004	1,161016

Pada tabel 2 merupakan hasil konversi dari mikrosecond ke second pada masing masing state. Data tersebut adalah gabungan dari data dari state enter dan state exit pada masing masing state yang ada pada sistem.

4.2 Pengujian Lama Eksekusi Transisi

Pada penelitian ini metode state machine diuji dengan menghitung lama transisi dari state pertama ke state kedua. Perhitungan transisi tersebut dilakukan pada masing masing state yang ada. Tampilan saat data perhitungan transisi dapat dilihat pada gambar 8 dibawah.



Gambar 8 Tampilan Perhitungan State

Tabel 3 Tampilan Perhitungan State

STATE	p1 (ms)	p2(ms)	p3(ms)	p4(ms)	p5(ms)	p6(ms)	p7(ms)	p8(ms)	p9(ms)	p10(ms)
Diam	220	220	220	220	220	220	220	228	224	216
Termonitor	232	232	224	232	224	228	224	228	224	228
Terkirim	216	212	216	216	212	212	212	212	212	212

Tabel 3 diatas merupakan hasil hasil dari eksekusi transisi masing masing *state*. Data diatas terformat dalam waktu mikrosecond. Terdapat 3 transisi yang diuji yaitu transisi dari diam ke termonitor, termonitor ke terkirim dan terkirim ke diam.

4.3 Pengujian Total Eksekusi State dan Transisi

Pada penelitian ini metode *state machine* diuji dengan menghitung lama transisi dari *state* pertama ke *state* kedua. Kemudian menguji seberapa lama transisi dari masing masing *state*.Kemudian kedua data tersebut ditambahkan hingga mendapatkan hasil total eksekusi. Tampilan saat data perhitungan dapat dilihat pada tabel dibawah ini

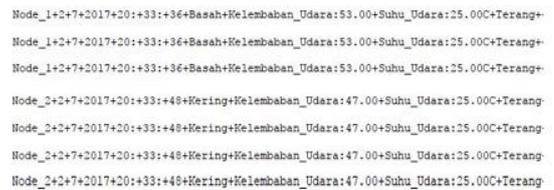
Tabel 4 Total Eksekusi State dan Transisi

STATE & TRANSISI	P-1 (ms)	P-2 (ms)	P-3 (ms)	P-4 (ms)	P-5 (ms)	P-6 (ms)	P-7 (ms)	P-8 (ms)	P-9 (ms)	P-10 (ms)
DIAM	diam	21324	21316	21320	21336	21332	21330	21312	21312	21320
	Transisi diam	220	220	220	220	220	220	228	224	216
	total	21544	21536	21540	21556	21552	21556	21540	21536	21536
TERMONITOR	Termonitor	921256	921244	921268	921808	921252	921256	922280	921256	920792
	Transisi termonitor	232	232	224	232	224	228	224	228	224
	total	921488	921476	921492	922040	921476	921484	922504	921484	921480
TERKIRIM	terkirim	1161020	1161020	1161020	1161020	1160996	1161008	1161012	1161016	1161016
	Transisi terkirim	216	212	216	216	212	212	212	212	212
	total	1161236	1161212	1161236	1161236	1161208	1161220	1161224	1161228	1161216

Tabel 4 diatas merupakan hasil hasil dari eksekusi transisi dan *state*. Data diatas terformat dalam waktu mikrosecond. Terdapat 3 transisi yang diuji yaitu transisi dari diam ke termonitor, termonitor ke terkirim dan terkirim ke diam.

4.4 Pengujian Masing Masing Sensor

Pengujian masing masing sensor dilakukan berdasarkan Batasan penelitian yang telah ditetapkan. Pengujian dilakukan dengan membuat miniatur keadaan lingkungan. Pada pengujian ini, jarak antara *Client* dan *Server* berjarak kurang lebih 80m. Pengujian tersebut dilakukan pada kedua *Node Client* yang berada pada kondisi berbeda. Pada gambar 9 merupakan tampilan sensor yang ditampilkan pada serial monitor masing masing *Node Client*.

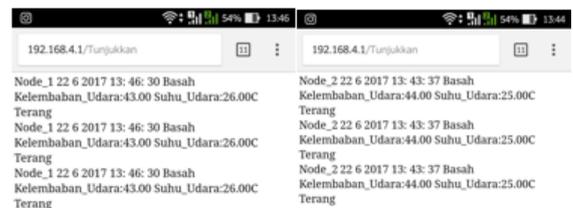


Gambar 9 Pengujian Sensor

Gambar 9 merupakan tampilan dari serial monitor yang menampilkan hasil dari sensor sensor yang terhubung pada Arduino. Pada tampilan diatas terdapat 2 Node yang diuji secara bersamaan. Tanda “+” diatas berfungsi untuk menggabungkan data yang terpisah agar terkirim secara berurutan dan bersamaan.

4.5 Pengujian Pengiriman dari Node Client ke Server

Pada pengujian ini yang diuji adalah data keluaran dari *Node Client* yang dikirim ke *Node Server* kemudian ditampilkan kepada User. Data tersebut dapat kita akses melalui browser pada alamat 192,168.4.1/Tunjukkan. Data hasil yang ditampilkan merupakan Node yang mengirim data, waktu pengiriman serta hasil dari sensor sensor yang digunakan. Pada gambar 10 dibawah merupakan tampillah hasil sensor dari *smartphone*.



Gambar 10 Pengujian Tampilan Pengiriman

4.6 Pengujian Sleep Mode pada Node Client

Pada pengujian *Sleep Mode* dilakukan terpisah dengan seluruh *state* dikarenakan untuk mempermudah pengujian. *Sleep Mode* di implementasikan menggunakan watchdog pada *state* diam. *State* diam melakukan keadaan tidur dengan waktu 8 detik. Berikut hasil dari pengujian yang dilakukan.

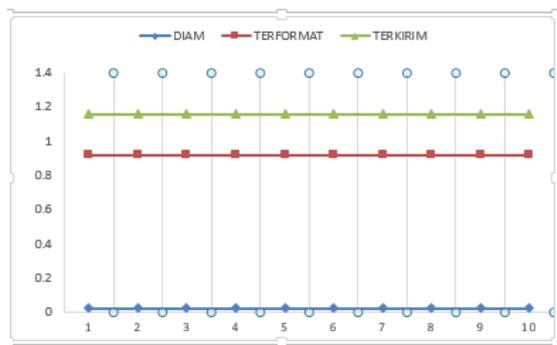
Tabel 5 Pengujian Sleep Mode

PENGUJIAN	HASIL(D2-D1)	WAKTU HASIL PENGUJIAN	
		D1/D2	NODE/TANGGAL/JAM/MENIT/DETIK
P1	38 detik	Data 1	Node_1+18+7+2017+10:+50:+19
		Data 2	Node_1+18+7+2017+10:+50:+57
P2	38 detik	Data 1	Node_1+18+7+2017+10:+51:+34
		Data 2	Node_1+18+7+2017+10:+52:+12
P3	37 detik	Data 1	Node_1+18+7+2017+10:+52:+49
		Data 2	Node_1+18+7+2017+10:+53:+26
P4	37 detik	Data 1	Node_1+18+7+2017+10:+54:+4
		Data 2	Node_1+18+7+2017+10:+54:+41
P5	38 detik	Data 1	Node_1+18+7+2017+10:+55:+18
		Data 2	Node_1+18+7+2017+10:+55:+56
P6	38 detik	Data 1	Node_1+18+7+2017+10:+56:+33
		Data 2	Node_1+18+7+2017+10:+57:+11
P7	37 detik	Data 1	Node_1+18+7+2017+10:+57:+48
		Data 2	Node_1+18+7+2017+10:+58:+25
P8	37 detik	Data 1	Node_1+18+7+2017+10:+59:+3
		Data 2	Node_1+18+7+2017+10:+59:+40
P9	38 detik	Data 1	Node_1+18+7+2017+11:+0:+17
		Data 2	Node_1+18+7+2017+11:+0:+55
P10	38 detik	Data 1	Node_1+18+7+2017+11:+1:+32
		Data 2	Node_1+18+7+2017+11:+2:+10

Pada Tabel 5 hasil pengujian dilakukan sebanyak 10 kali. Setiap pengujian dilakukan dengan mengurangi hasil data jam kedua dari data jam pertama. Hasil data rata rata adalah 38 detik untuk waktu eksekusi sleep mode. Data tersebut merupakan gabungan antara delay setiap state , transisi serta waktu watchdog mengeksekusi timer yang dimiliki.

4.7 Analisis Pengujian State

Pada gambar 11 dibawah ini merupakan kumpulan data yang telah dikumpulkan selama 10 kali pada pengujian state yang dilakukan. Pengujian dilakukan dengan menggunakan mikrosecond sebagai format hitung agar data terlihat lebih jelas dikarenakan state ter eksekusi kurang dari 2 detik.



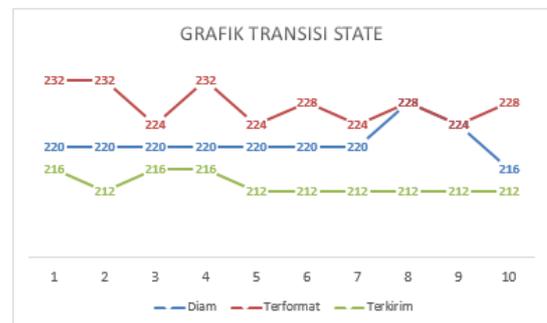
Gambar 11 Grafik Eksekusi State

Gambar 11 diatas merupakan hasil grafik dari eksekusi masing masing state yang telah dijadikan second. Untuk yang berwarna biru merupakan state diam. Kemudian yang

berwarna merah merupakan state termonitor. Terakhir yang berwarna hijau merupakan state terkirim.

4.8 Analisis Pengujian Transisi State

Pada gambar 12 dibawah ini merupakan kumpulan data yang telah dikumpulkan selama 10 kali pada pengujian transisi yang dilakukan. Pengujian dilakukan dengan menggunakan mikrosecond sebagai format hitung agar data terlihat lebih jelas dikarenakan state ter eksekusi kurang dari 2 detik.

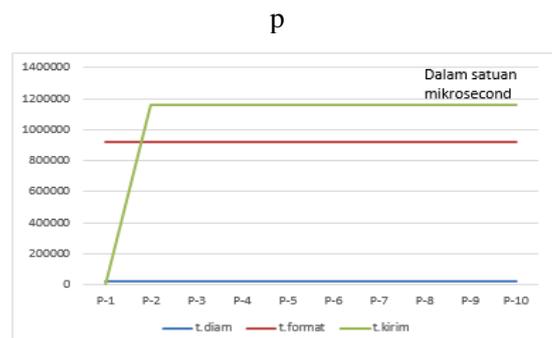


Gambar 12 Grafik Transisi State

Gambar 12 diatas merupakan hasil grafik dari eksekusi masing masing transisi. Untuk yang berwarna biru merupakan state diam. Kemudian yang berwarna merah merupakan state termonitor. Terakhir yang berwarna hijau merupakan state terkirim. Dapat kita lihat bahwa state format berubah beberapa mikrosecond. Waktu transisi paling cepat yaitu pada proses transisi terkirim.

4.9 Analisis Total Waktu Eksekusi

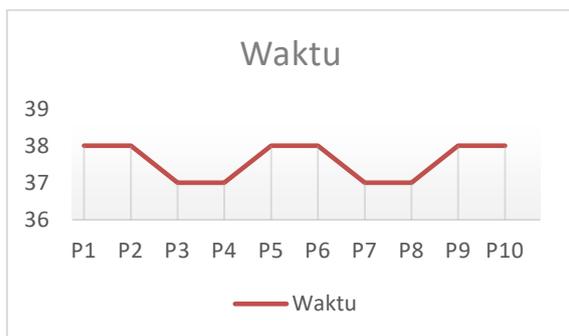
Pada gambar 13 dibawah ini merupakan kumpulan data yang telah dikumpulkan selama 10 kali pada pengujian transisi yang dilakukan. Pengujian dilakukan dengan menggunakan mikrosecond sebagai format hitung agar data terlihat lebih jelas dikarenakan state ter eksekusi kurang dari 2 detik.



Gambar 13 Grafik Eksekusi Total

Gambar 13 diatas merupakan hasil grafik dari gabungan waktu eksekusi dan eksekusi masing masing transisi. Untuk yang berwarna biru merupakan *state* diam. Kemudian yang berwarna merah merupakan *state* termonitor. Terakhir yang berwarna hijau merupakan *state* terkirim. Dapat kita lihat bahwa *state* format berubah beberapa mikrosecond

4.10 Analisis Hasil Waktu Sleep Mode



Gambar 14 Grafik Hasil Sleep Mode

Gambar 14 diatas merupakan hasil grafik hasil dari gabungan waktu eksekusi dan eksekusi masing masing transisi serta data watchdog timer. Dalam grafik diatas grafik waktu ditunjukkan dengan warna merah. Kemudian P1-P10 merupakan banyaknya pengujian yang dilakukan oleh peneliti. Dari data hasil tersebut dapat disimpulkan bahwa *sleep mode* mempengaruhi waktu total eksekusi seluruh *state* dan transisi. *Sleep Mode* yang di atur selama 8 detik diakibatkan oleh timer dan delay tambahan dari *state* serta transisi menjadikan *sleep mode* menjadi lebih panjang yaitu rata rata 38 detik.

5. KESIMPULAN

Berdasarkan hasil dari tahapan perancangan, implementasi, pengujian serta analisis hasil pengujian yang telah dilakukan, penulis dapat menarik kesimpulan bahwa :

- Sensor node dapat diimplementasikan pada sistem dengan merakit komponen-komponen yang telah ditentukan seperti Arduino Uno, sensor LDR, sensor DHT11, ESP8266, modul RTC , sensor YL-38 dan YL-69. Sensor pada kedua *node client* mampu melakukan deteksi sekaligus pengambilan data pada pertanian. Hasil dari sensor tersebut mampu dikirimkan oleh

kedua *node client* menuju *node server* menggunakan modul *wireless* ESP8266.

- Metode *state machine* sebagai metode sistem berhasil diimplementasikan pada *node*.
- Kedua node mengirimkan data dalam waktu yang berbeda sehingga menampilkan data secara bergantian. Kedua *node client* menampilkan data dari sensor-sensor yang terhubung ke mikrokontroler.
- Dari pengujian sensor dan juga *state machine* dapat dikatakan bahwa implementasi *state machine* pada pertanian modern berbasis *wireless* telah berhasil dilakukan.

DAFTAR PUSTAKA

Garzón , C. A. L. & Riveros, O. J. R., 2010. Temperature, humidity and luminescence monitoring system using Wireless Sensor Networks (WSN) in flowers growing..

Kramer, M., Bader, S. & Oelmann, B., 2013. Implementing Wireless Sensor Network Applications using Hierarchical Finite State Machines.

Liu, X., Chen, . S. & Song, . W., 2014. A Design and Implementation of Watchdog Based on Observer Pattern and Finite State Machine.

Sun, D., Lu, J., Wang, W. & Lin, Z., t.thn. DESIGN OF WSN NODES AND NETWORK PERFORMANCE ANALYSIS IN A TEA PLANTATION.

Xiao, L. & Guo, L., 2010. The Realization of Precision Agriculture Monitoring System Based on Wireless Sensor Network.

Anon., 2010. DHT11 Humidity & Temperature Sensor. [Online] [Diakses 5 April 2016].

Arduino CC, 2016. [Online] Available at: <https://www.arduino.cc/en/Main/ArduinoBoardNano> [Diakses Oktober 2016].

b., 2016. www.baselinesystems.com. [Online] [Diakses 2017 Mei 17].

D-Robotics, 2010. DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output.

e., 2016. www.elecrow.com. [Online] Available at: <https://www.elecrow.com/wiki/index.php>

- ?title=Tiny_RTC [Diakses 7 April 2017].
- F. T. D. i. L., 2016. <http://www.ftdichip.com/>. [Online] [Diakses 18 Juli 2017].
- Iwan Setiawan, S. M., 2006. PERANCANGAN SOFTWARE EMBEDDED SYSTEM BERBASIS FSM.
- Larsen, N., 2011. BASIC WATCHDOG TIMER. 2011 Juni.
- Mendez, G. . R., Yunus, M. . A. M. & Mukhopadhyay, D. S. . C., 2011. A WiFi based Smart Wireless Sensor Network for an Agricultural Environment.
- Setiawan, I., 2006. PERANCANGAN SOFTWARE EMBEDDED SYSTEM BERBASIS FSM , pp. 1-2.
- Sun, D., Lu , J., Wang, W. & Lin, Z., t.thn. DESIGN OF WSN NODES AND NETWORK PERFORMANCE ANALYSIS IN A TEA PLANTATION.