

BAB IV. ANALISIS DAN PERANCANGAN SISTEM

Pada bab analisis dan perancangan sistem berisi pembahasan analisis dan perancangan prototype dan sistem *website monitoring*. Pembahasan ditujukan untuk menguraikan kebutuhan – kebutuhan dalam pengembangan aplikasi.

4.1 Gambaran Umum Aplikasi

Sistem aplikasi *monitoring* inkubator telur puyuh berbasis *website* ini menerapkan teknologi *Internet of Things*, menggunakan sistem kontrol pintar yang mana sumber panas (lampu) dan kecepatan kipas dapat dikontrol berdasarkan suhu dan kelembapan pada ruang inkubator menggunakan logika *fuzzy*. Sistem ini juga dapat memutar telur secara otomatis dan terjadwal menggunakan sistem rak geser juga dapat mendeteksi telur yang telah menetas serta dapat dimonitoring secara jarak jauh. Sistem aplikasi yang dibuat bertujuan untuk menerapkan inovasi terbaru mengikuti perkembangan teknologi dan juga mengubah sistem kontrol lampu on dan off yang telah diterapkan pada mesin tetas yang sudah ada menjadi sistem kontrol pintar dimana cahaya lampu dan kecepatan kipas dapat disesuaikan berdasarkan suhu dan kelembapan pada ruang inkubator menggunakan logika *fuzzy*. Menggunakan MQTT (*Message Queue Telemetry Transport*) sebagai transmisi data antara *embedded system* dan *website*.

4.2 Analisis Pengguna

Selain peternak sistem aplikasi ini dapat digunakan oleh orang umum yang ingin menetas telur. *Monitoring* kondisi *real time* ruang inkubator dilakukan melalui platform *website monitoring* meliputi suhu, kelembapan, kekuatan lampu, kecepatan kipas dan status telur menetas pada ruang inkubator secara *real time*. Adapun fitur lainnya yaitu visualisasi suhu dan kelembapan dalam bentuk chart *real time*, prediksi tanggal telur akan menetas, perhitungan hari ke – n dan *history* data telur yang telah masuk dalam inkubator. Sistem aplikasi dapat diakses pengguna menggunakan *web browser*.

4.3 Analisis Kebutuhan Fungsional

Kebutuhan fungsional adalah jenis kebutuhan yang berisi proses atau fitur yang dapat dilakukan dalam aplikasi. Proses atau fitur yang dimaksud adalah sebagai berikut.

1. Sistem aplikasi dapat menampilkan kondisi ruang inkubator secara *real time* pada *website*.
2. Sistem aplikasi dapat melakukan perhitungan data menggunakan metode *fuzzy*. *Input* perhitungan didapat dari sensor suhu dan kelembapan, *output* dari perhitungan digunakan untuk menggerakkan aktuator.
3. Sistem aplikasi dapat melakukan pemutaran telur secara otomatis dan terjadwal.
4. Sistem aplikasi dapat mendeteksi telur yang telah menetas dan menampilkannya pada *dashboard monitoring*.
5. Sistem aplikasi telah terhubung dengan database sehingga data dapat tersimpan dan *history* data lampau dapat dilihat kembali.
6. Sistem aplikasi terdapat fitur manual kontrol meliputi kontrol manual on dan off dinamo, intensitas cahaya lampu, dan kecepatan kipas.

4.4 Analisis Kebutuhan Non Fungsional

Terdapat beberapa analisa kebutuhan non fungsional meliputi kebutuhan perangkat lunak (*software*) dan perangkat keras (*hardware*). Berikut adalah rincian kebutuhan sistem yang akan dibuat :

4.4.1 Perangkat lunak (*software*)

Spesifikasi umum perangkat lunak yang dibutuhkan dalam pembuatan aplikasi ini tercantum pada Tabel 4.1.

Tabel 4. 1 Spesifikasi Kebutuhan Perangkat Lunak (*Software*)

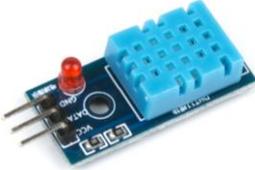
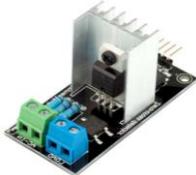
No.	Nama Perangkat Lunak	Keterangan
1.	Sistem Operasi Windows 10	Sistem operasi yang digunakan untuk membuat sistem aplikasi.
2.	Teks Editor VSCode	Text editor yang digunakan untuk membuat aplikasi <i>website</i> .

3.	Arduino IDE	Text editor yang digunakan untuk menulis kode program node mcu.
4.	XAMPP	Web server lokal untuk menjalankan <i>website</i> .
5.	Mozilla Firefox	Browser yang digunakan untuk menjalankan aplikasi <i>website</i> .
6.	Broker Emqx	Sebuah layanan server cloud yang menyediakan broker.

4.4.2 Perangkat keras (*hardware*)

Spesifikasi umum perangkat keras yang dibutuhkan dalam pembuatan aplikasi ini tercantum pada Tabel 4.2.

Tabel 4. 2 Spesifikasi Kebutuhan Perangkat Keras (*Hardware*)

No.	Nama Perangkat Keras	Gambar
1.	Node MCU ESP8266	
2.	Sensor dht11	
3.	Sensor pir	
4.	AC Light Dimmer	

5.	Modul L298N	
6.	Relay 1 channel	 <p>relay 1 channel</p>
7.	LCD I2C 16x2	
8.	Modul <i>Step-down</i>	
9.	Saklar	
10.	PCB dot Matrik	
11.	Lampu Pijar	
12.	Kipas DC 12V	
13.	<i>Dynamo Synchronous</i>	

14.	Kabel Engkel	
15.	Adaptor 12V	
16.	Box ukuran 40cm x 30cm x 50cm	

4.5 Perancangan Sistem

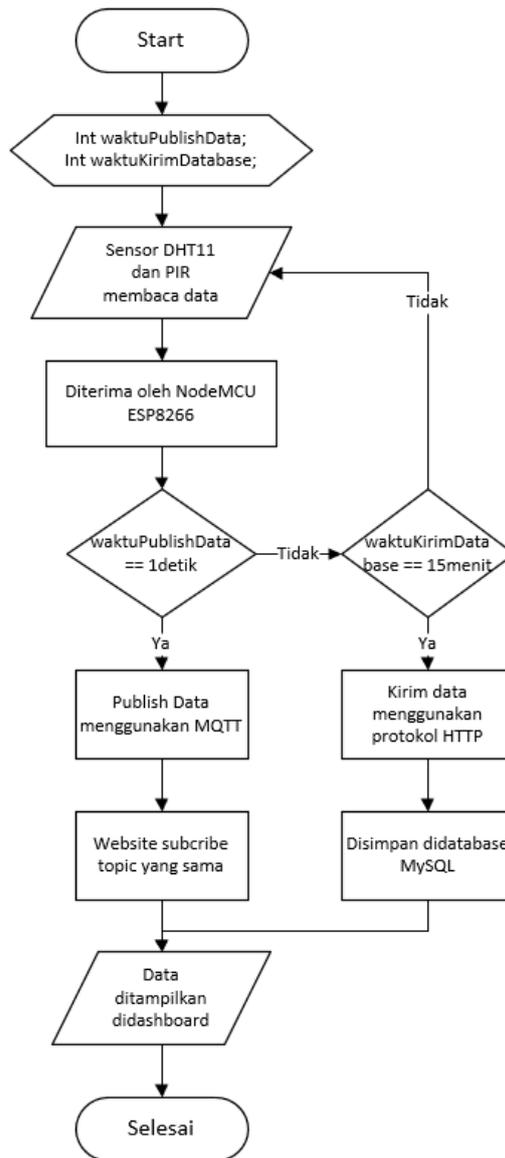
Perancangan sistem merupakan tahap untuk menggambarkan desain dan alur berjalannya sebuah sistem, yang terdiri dari langkah – langkah operasi dalam sebuah sistem. Desain yang sudah dirancang akan menggambarkan semua aktifitas user, proses sistem mulai awal sampai akhir, serta menjelaskan desain arsitektur sistem.

4.5.1 *Flowchart* Sistem *Monitoring*

Flowchart merupakan urutan langkah - langkah dari sebuah proses yang memiliki hubungan antar proses satu dengan yang lainnya dalam sebuah program atau sistem. Berikut merupakan *flowchart* sistem untuk *monitoring* ruang inkubator penetas telur.

Sensor dht11 membaca suhu dan kelembapan, sensor pir membaca pancaran sinar infrared dan gerakan pada ruang inkubator. Kemudian diterima oleh NodeMCU dan dilakukan pengolahan data jika interval waktu publish data telah siap yaitu setiap 1 detik maka data hasil dari pembacaan sensor dht11 dan pir akan di-*publish* menggunakan protokol MQTT melewati *cloud broker* emqx, disamping

itu jika interval waktu kirim database telah siap yaitu setiap 15 menit maka data akan disimpan ke database menggunakan protokol http. *Website* akan mensubscribe topic yang sama dengan topic yang telah di-*publish* dan *dashboard* akan menampilkan data sesuai dengan data yang telah di-*publish* oleh NodeMCU, *Flowchart* sistem *monitoring* ditunjukkan pada Gambar 4.1.

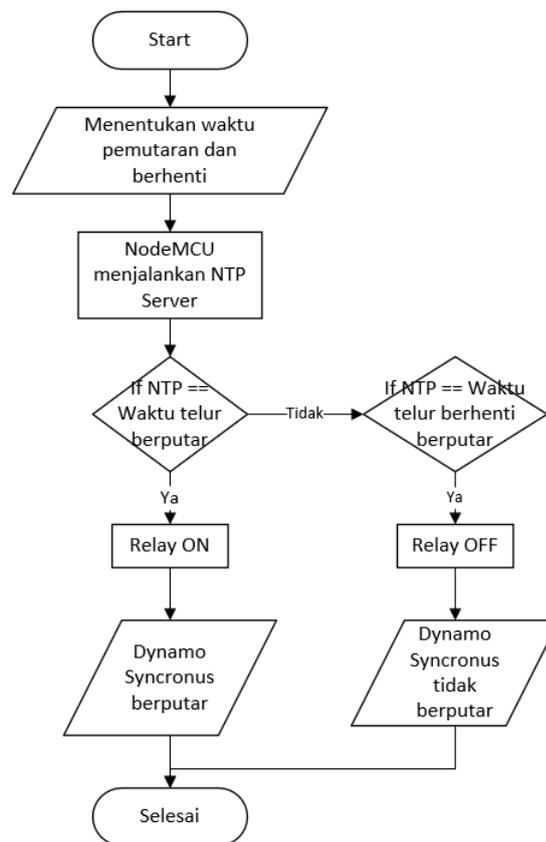


Gambar 4. 1 *Flowchart* Sistem *Monitoring*

4.5.2 *Flowchart* Sistem Otomatisasi Pemutaran Telur

Pada fungsi sistem pemutaran telur secara otomatis pertama menentukan waktu pemutaran dimulai dan waktu pemutaran berhenti setelah itu NodeMCU

menjalankan NTP Server yaitu pengsinkronan waktu didalam sebuah jaringan jika waktu ntp sama dengan waktu pemutaran dimulai maka relay akan menyala dan *dynamo synchronus* akan berputar untuk menggeser rak telur agar telur berputar, jika waktu ntp sama dengan waktu pemutaran berhenti maka relay akan mati. Selisih waktu antara waktu pemutaran mulai dan berhenti adalah 16 detik sehingga relay akan menyala selama 16 detik. *Flowchart* sistem pemutaran telur otomatisasi pemutaran telur ditunjukkan pada Gambar 4.2.

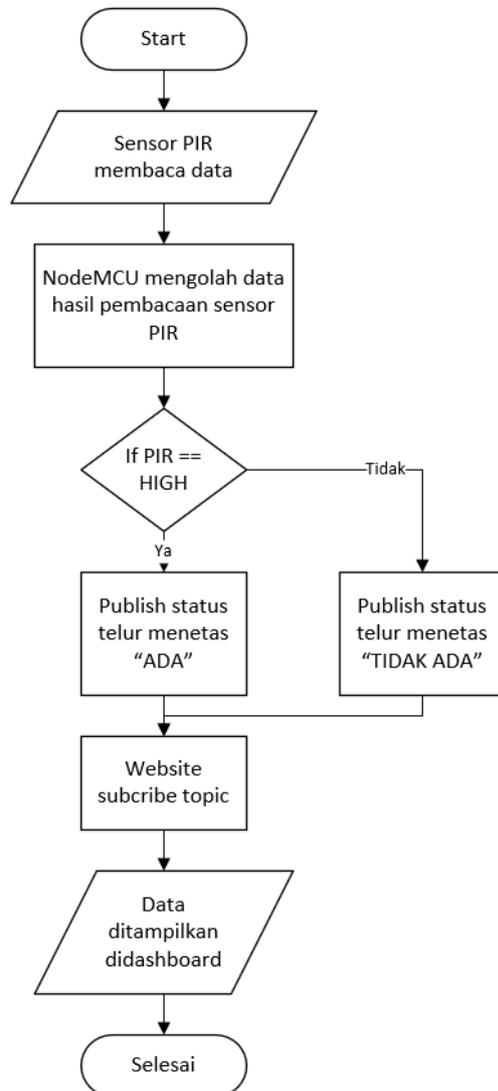


Gambar 4. 2 *Flowchart* Sistem Otomatisasi Pemutaran Telur

4.5.3 *Flowchart* Sistem Otomatisasi Deteksi Telur Menetas

Pada fungsi sistem otomatisasi deteksi telur menetas pertama sensor pir akan mendeteksi pancaran infrared dan gerakan dalam ruang inkubator kemudian hasil pembacaan data diolah oleh NodeMCU jika pembacaan sensor sama dengan HIGH maka NodeMCU akan publish status telur menetas “ADA” tetapi jika pembacaan sensor bukan HIGH maka NodeMCU publish status telur menetas “TIDAK ADA” menggunakan protokol MQTT. *Website* akan mensubscribe topic yang sama

dengan topic yang di-*publish* NodeMCU sehingga data dapat ditampilkan dihalaman *dashboard*. *Flowchart* sistem otomatisasi deteksi telur menetas ditunjukkan pada Gambar 4.3.

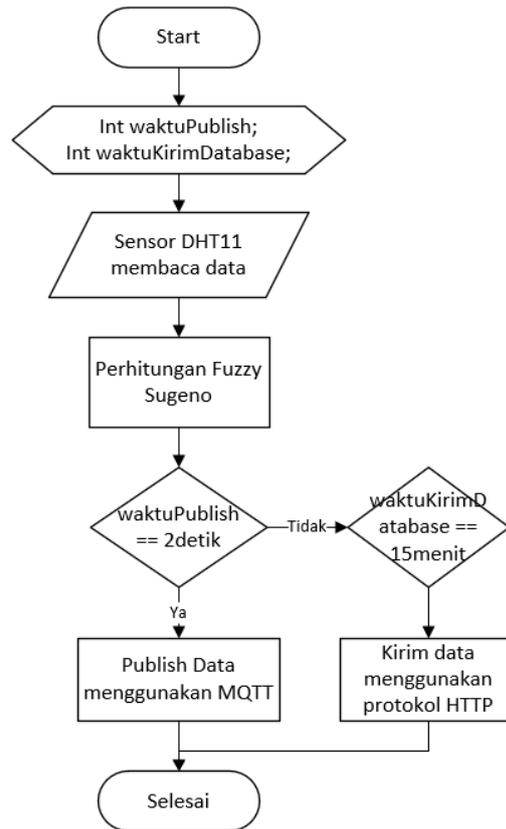


Gambar 4. 3 *Flowchart* Sistem Otomatisasi Deteksi Telur Menetas

4.5.4 *Flowchart* Sistem Otomatisasi *Fuzzy*

Data suhu dan kelembapan yang dibaca oleh sensor dht11 akan diproses oleh NodeMCU dengan metode *fuzzy* sugeno. Kemudian data *output* dari perhitungan *fuzzy* akan disesuaikan dengan rule *fuzzy* yang sudah ditentukan. Jika parameter suhu dan kelembapan ruang inkubator belum sesuai maka aktuator sebagai *output* akan aktif sesuai rules yang sudah ditentukan, lalu sistem akan secara terus menerus memantau kondisi ruang inkubator secara *realtime* dan melakukan *monitoring*

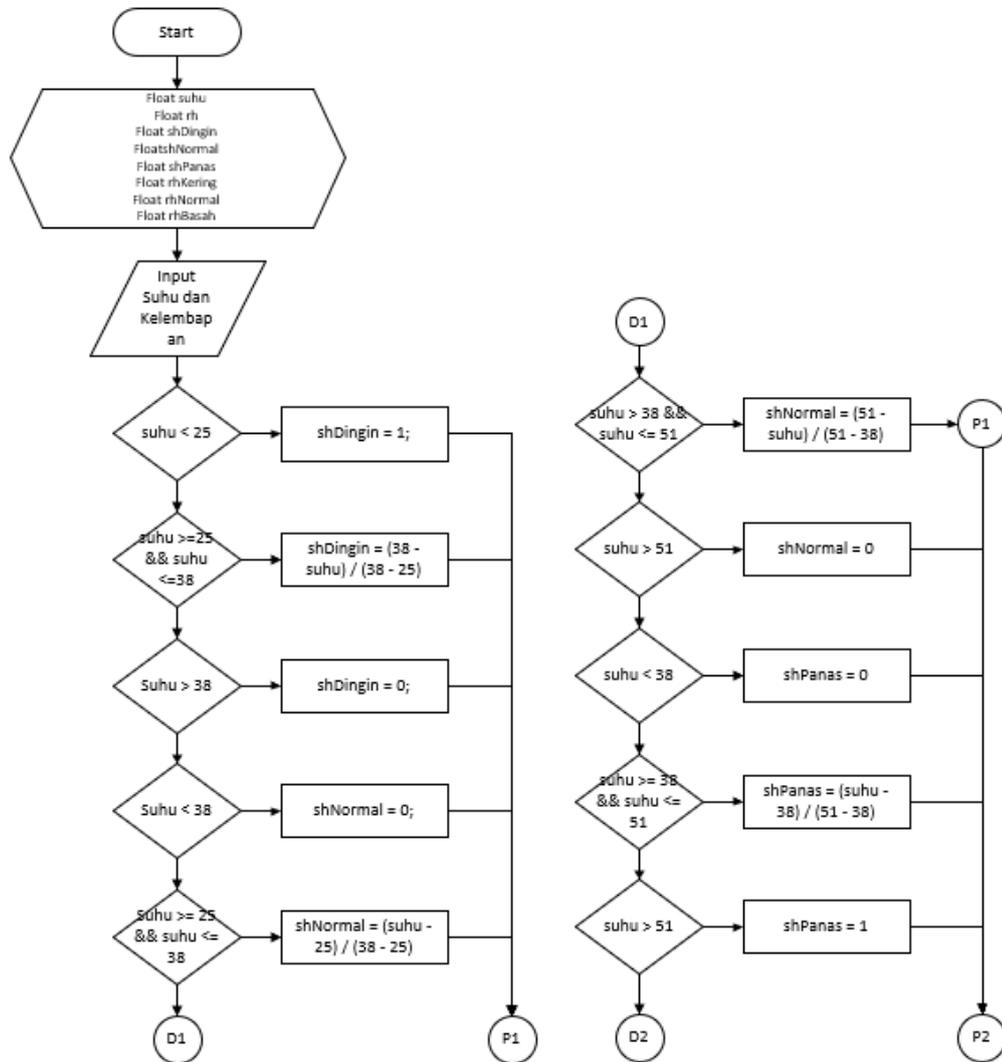
ruang inkubator. Terdapat Interval waktu berbeda yang digunakan untuk pengiriman data yaitu 2 detik untuk publish data hasil *fuzzy* dan 15 menit untuk menyimpan data kedalam database. Flowchart sistem otomatisasi *fuzzy* ditunjukkan pada gambar 4.4.



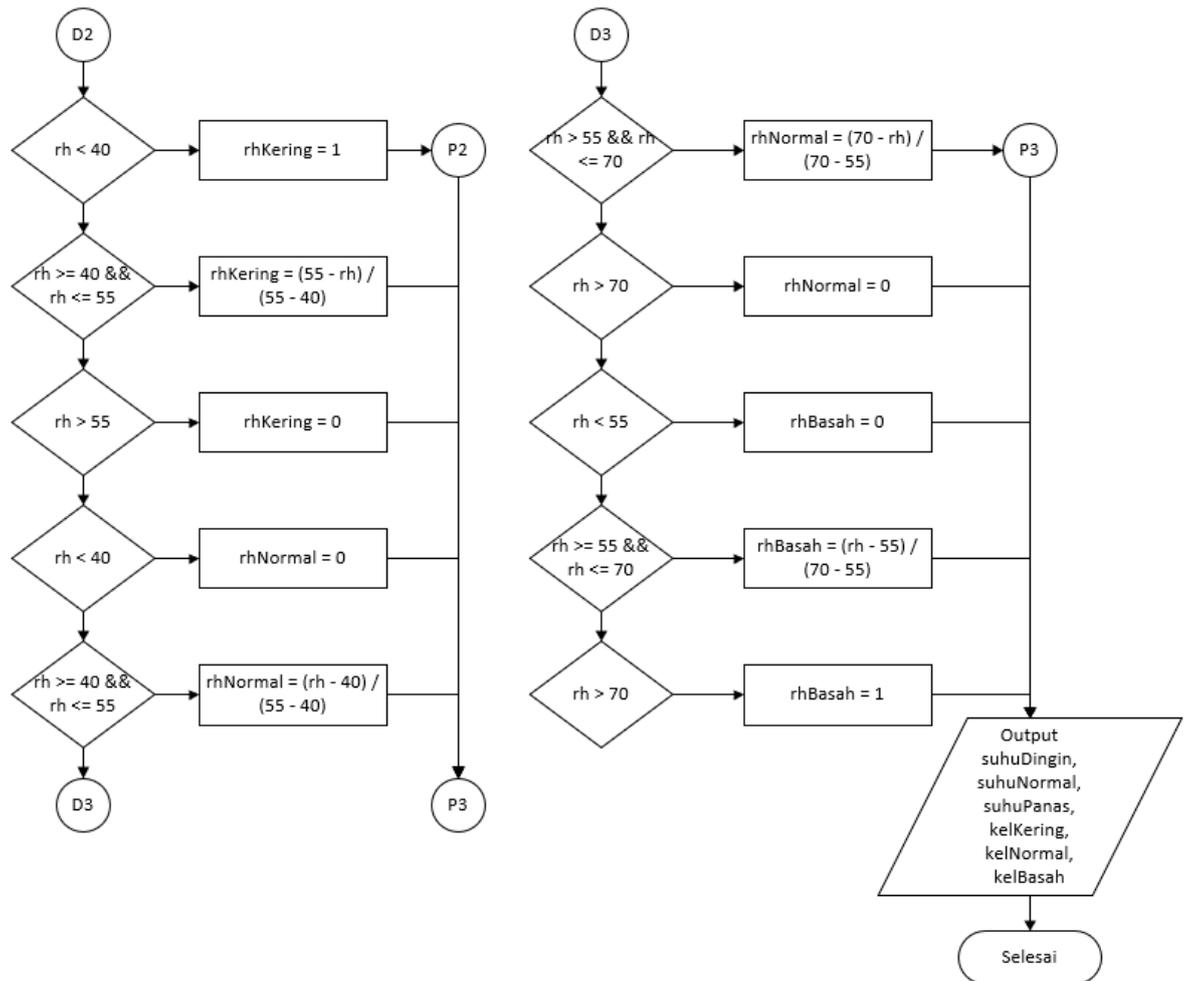
Gambar 4. 4 *Flowchart* Sistem Otomatisasi *Fuzzy*

Pada *flowchart* sistem automasi *fuzzy* dimulai dari *input* dari data sensor suhu dan kelembapan kemudian dilakukan perhitungan menggunakan metode *fuzzy* sugeno yang dimana akan dijelaskan dalam *flowchart* predefined process (perhitungan metode *fuzzy* sugeno) dan hasilnya akan di-*publish* menggunakan protokol *mqtt* dan di-*subscribe* oleh *website* lalu data dari hasil perhitungan atau status dari automasi aktuator ditampilkan dalam *website dashboard monitoring*. Proses perhitungan *fuzzy* sugeno terdapat tahapan-tahapan proses, pertama menentukan *membership function* (fungsi keanggotaan) dari nilai sensor yang diterima dan dihitung dengan representasi kurva yang sesuai dan tiap hasil

perhitungan akan disimpan dalam variabel kondisi. Proses Fuzzyfikasi himpunan suhu dan kelembapan ditunjukkan pada Gambar 4.5 dan Gambar 4.6.

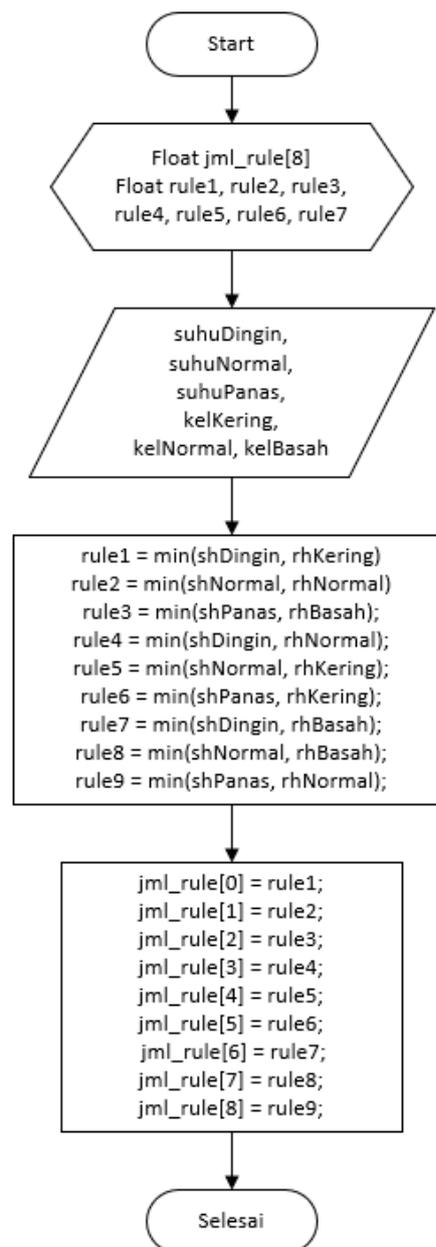


Gambar 4. 5 Flowchart Proses Fuzzyfikasi Suhu



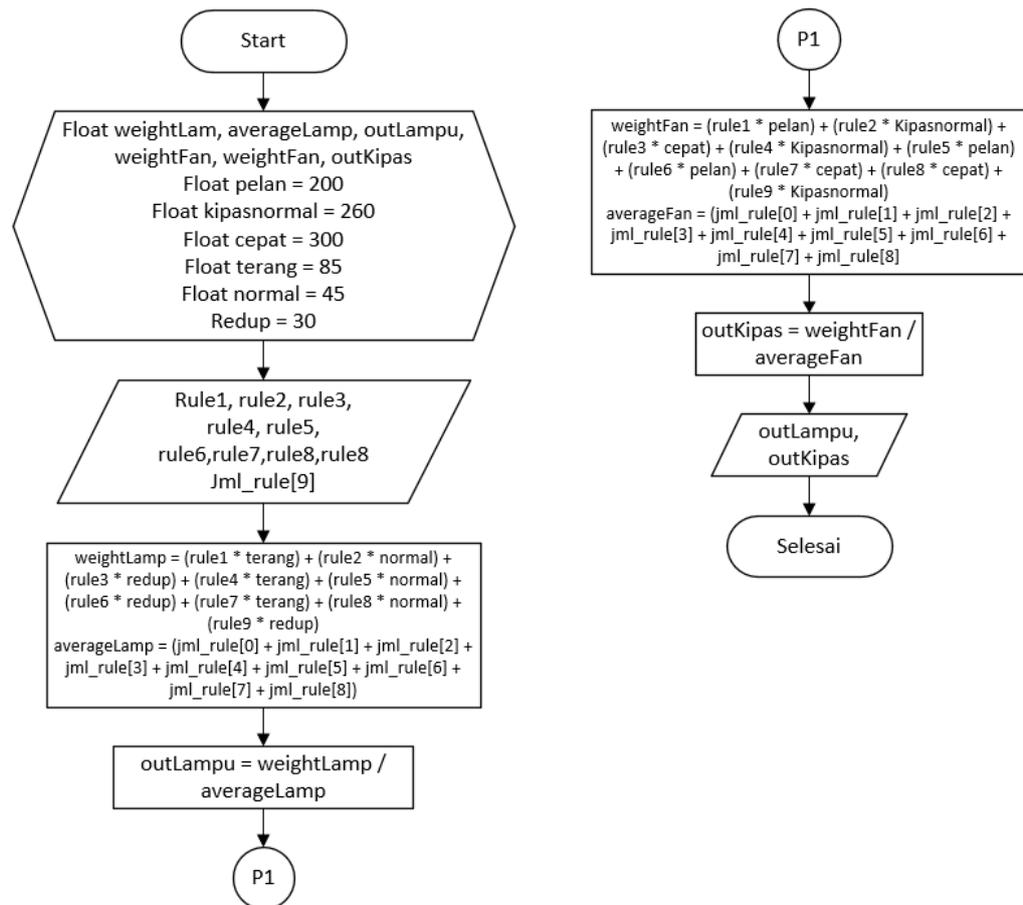
Gambar 4. 6 *Flowchart* Proses *Fuzzyfikasi* Kelembapan

Kedua, mencari nilai MIN dari dua hasil perhitungan fuzzifikasi. Dari dua variabel kondisi yang telah didapatkan sebelumnya, dalam proses fungsi implikasi akan dihitung nilai MIN dalam setiap kondisi dimana hasil perbandingan nilai akan disimpan kedalam variabel untuk masing – masing dari setiap kondisi. Kemudian variabel yang menyimpan hasil perbandingan akan dimasukkan kedalam array untuk selanjutnya digunakan sebagai pembagi pada tahap defuzzifikasi. *Flowchart* Proses Implikasi ditunjukkan pada Gambar 4.7.



Gambar 4. 7 *Flowchart* Proses Implikasi

Tahap terakhir yaitu defuzzifikasi, dalam defuzzifikasi dilakukan proses mencari nilai rata – rata terbobot (*weight average*) dengan cara menjumlahkan setiap rule dengan bobot *output* masing – masing rules kemudian dibagi dengan jumlah nilai dari rule yang mana hasil perhitungan disimpan dalam variabel. Terdapat 2 proses defuzzifikasi yaitu defuzzifikasi lampu dan kipas yang nantinya nilai *output* akan digunakan untuk menggerakkan aktuator. *Flowchart* Proses Defuzzifikasi ditunjukkan pada Gambar 4.8.



Gambar 4. 8 *Flowchart* Proses Defuzzifikasi

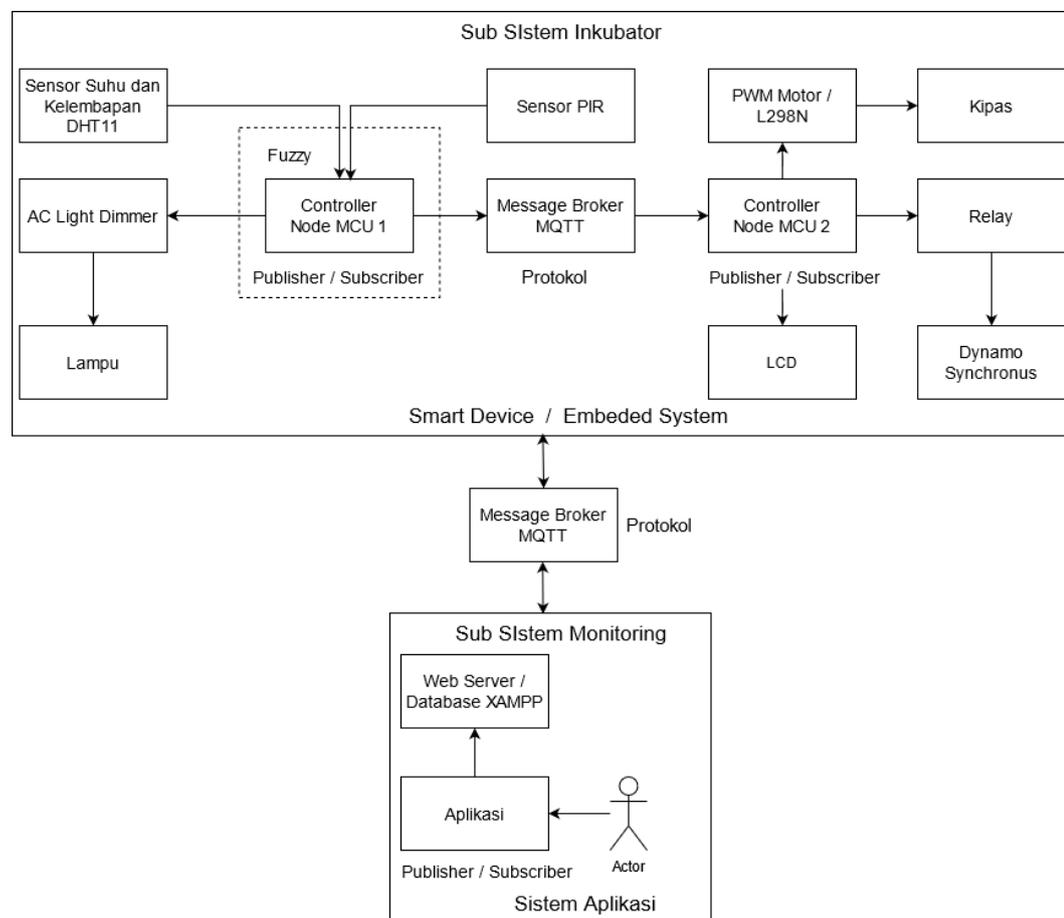
4.5.5 *Block* Diagram

Block diagram adalah sebuah alur pada diagram yang memetakan proses kerja dalam sistem dengan tujuan untuk memudahkan dalam mengenali komponen – komponen dan alur kerja yang ada di dalam sebuah sistem. Berikut adalah gambaran alur *block* diagram dari sistem.

Pada *block* diagram terdapat 2 sub sistem, yang pertama yaitu sub sistem inkubator atau *smart device* dimana sensor dht11 membaca data suhu dan kelembapan, sensor pir membaca pancaran infra merah dan gerakan. Data tersebut merupakan data *input* yang akan diolah menggunakan perhitungan metode *fuzzy* sugeno oleh NodeMCU 1. Terdapat 2 *output fuzzy* yaitu lampu dan kipas, setelah hasil perhitungan didapat *output* digunakan sebagai inputan AC *light dimmer* untuk menghidupkan lampu, sedangkan *output* kipas akan dikirim ke NodeMCU 2 menggunakan protokol mqtt. Peran NodeMCU 1 disini merupakan publisher dan

subscriber begitu juga dengan NodeMCU 2. Setelah *output* kipas diterima *output* tersebut akan digunakan sebagai *input* dari modul l298n untuk menggerakkan kipas. Proses lain yang ada dalam NodeMCU 2 yaitu menampilkan data pada LCD juga menghidupkan dan mematikan *dynamo synchronus* menggunakan relay.

Kedua yaitu sub sistem *monitoring*, pada sub sistem *monitoring* sistem aplikasi atau *website* berperan sebagai publisher dan subscriber yang mana nantinya *website* akan mensubscribe topic yang sama dengan apa yang telah di-*publish* oleh NodeMCU 1 dan 2 kemudian data akan ditampilkan pada *website* secara *realtime* dan dimasukkan dalam database. *Block diagram* sistem ditunjukkan pada Gambar 4.9.



Gambar 4. 9 *Block Diagram* Sistem

4.5.6 Database

4.5.2.1 Struktur Tabel Database

Struktur tabel database merupakan pengaturan tabel pada database yang akan digunakan untuk menyimpan data dan struktur tabel mempermudah dalam memasukkan data sesuai dengan tipe data yang telah ditentukan. Struktur tabel dari database aplikasi *monitoring* ruang inkubator telur puyuh pada MySQL adalah sebagai berikut:

Tabel 4. 3 Tabel manage

Nama Atribut	Tipe Data	Keterangan
Id	INT(5)	PRIMARY KEY, AUTO INCREMENT
jenis_telur	VARCHAR(11)	-
banyak_telur	INT(11)	-
tanggal_masuk	DATE()	-
tanggal_keluar	DATE()	-

Tabel 4. 4 Tabel sensor

Nama Atribut	Tipe Data	Keterangan
Id	INT(11)	PRIMARY KEY, AUTO INCREMENT
suhu	FLOAT()	-
kelembapan	FLOAT()	-
fuzzy_lampu	FLOAT()	-
fuzzy_kipas	FLOAT()	-
waktu	TIME()	-
tanggal	DATE()	-

Tabel 4. 5 Tabel device

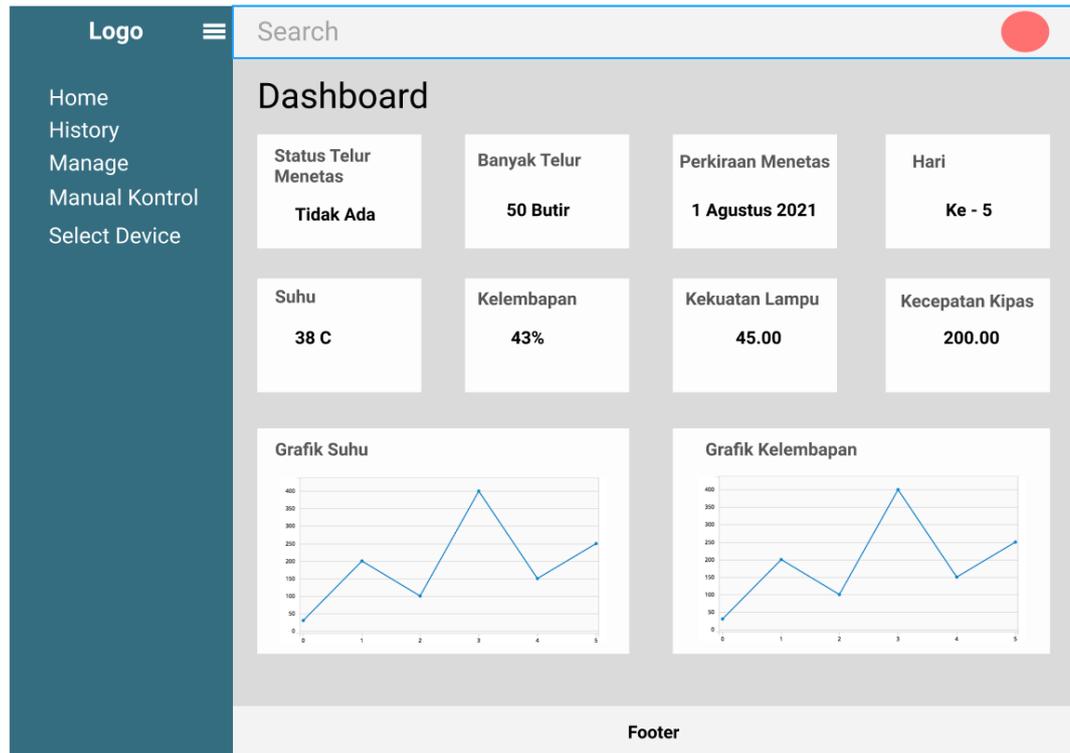
Nama Atribut	Tipe Data	Keterangan
Id	INT(11)	PRIMARY KEY, AUTO INCREMENT
mac_address	VARCHAR(25)	-
nama_device	VARCHAR(11)	-
topicColl	VARCHAR(25)	-

4.5.7 Design Interface

Desain Interface merupakan desain antarmuka yang digunakan dalam komputer, perangkat lunak, *website* dan mobile yang berfokus pada penampilan atau gaya. Tujuannya adalah membuat interaksi antara pengguna dan perangkat agar lebih mudah digunakan dan seefisien mungkin. Berikut adalah rancangan desain interface dari *website* auto assessment MySQL:

1. Halaman *dashboard website*

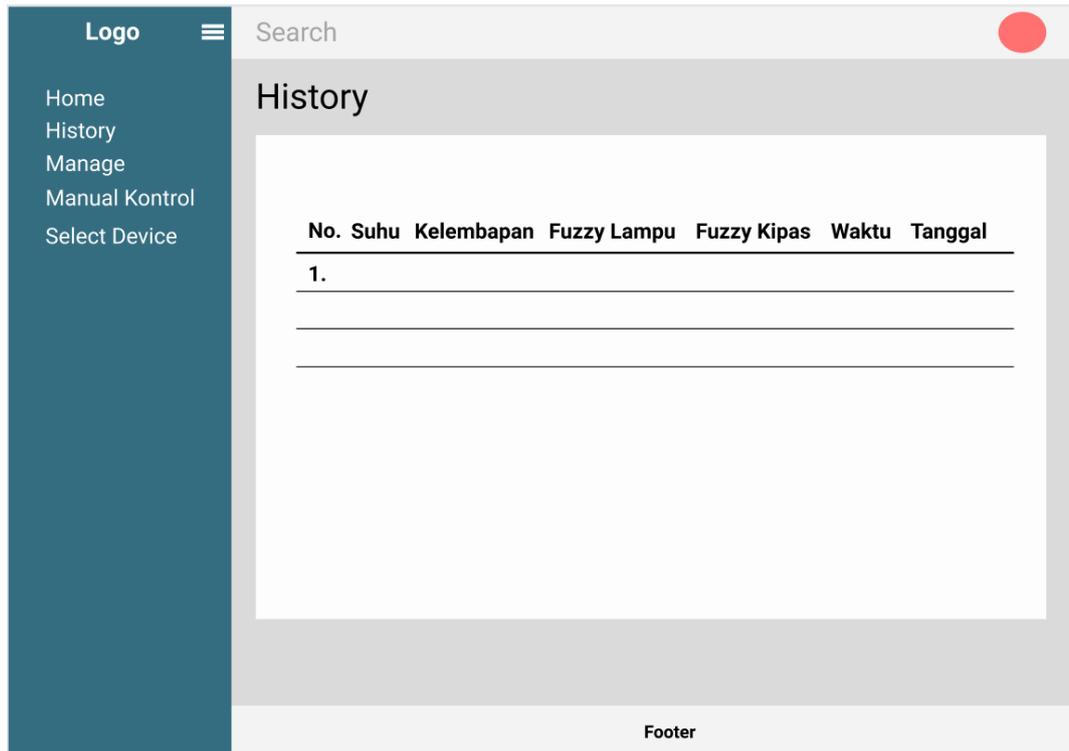
Halaman *dashboard* pada sebuah *website* merupakan bagian penting dalam user interface (antarmuka) yang letaknya berada diantara data dan desain. *Dashboard website* menampilkan berbagai informasi, matriks, angka, dan visualisasi data. Pada Gambar 4.10 merupakan mockup *dashboard website* dari sistem aplikasi menampilkan beberapa informasi meliputi status telur menetas, banyak telur, tanggal perkiraan menetas, hari ke- i, suhu *realtime*, kelembapan *realtime*, kekuatan lampu (hasil *fuzzy*), kecepatan kipas (hasil *fuzzy*), grafik suhu dan kelembapan *realtime*.



Gambar 4. 10 Halaman *Dashboard Website*

2. Halaman *History*

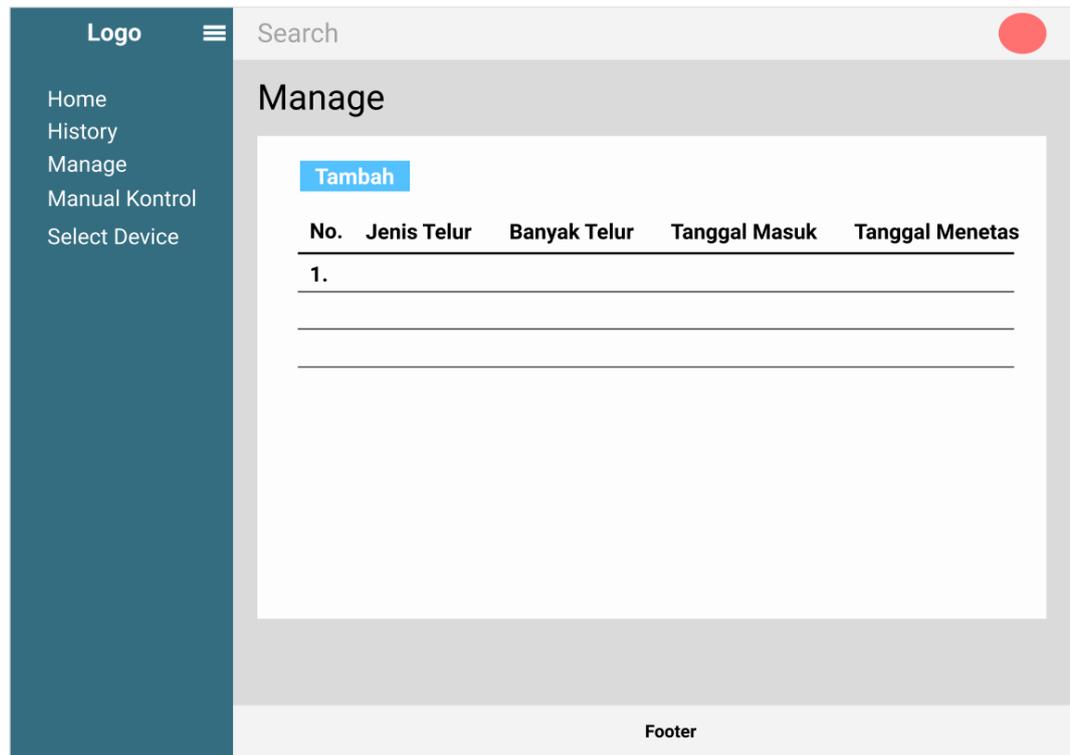
Pada Gambar 4.11 merupakan halaman *history* yang mana menampilkan data *log* yang telah tersimpan dalam database. Data tersebut meliputi suhu, kelembapan, hasil *fuzzy* lampu, hasil *fuzzy* kipas, waktu dan tanggal. Data *log* berfungsi agar pengguna dapat melihat riwayat suhu dan kelembapan pada waktu lampau dan melakukan evaluasi dari data yang ditampilkan.



Gambar 4. 11 Halaman *History*

3. Halaman Manage

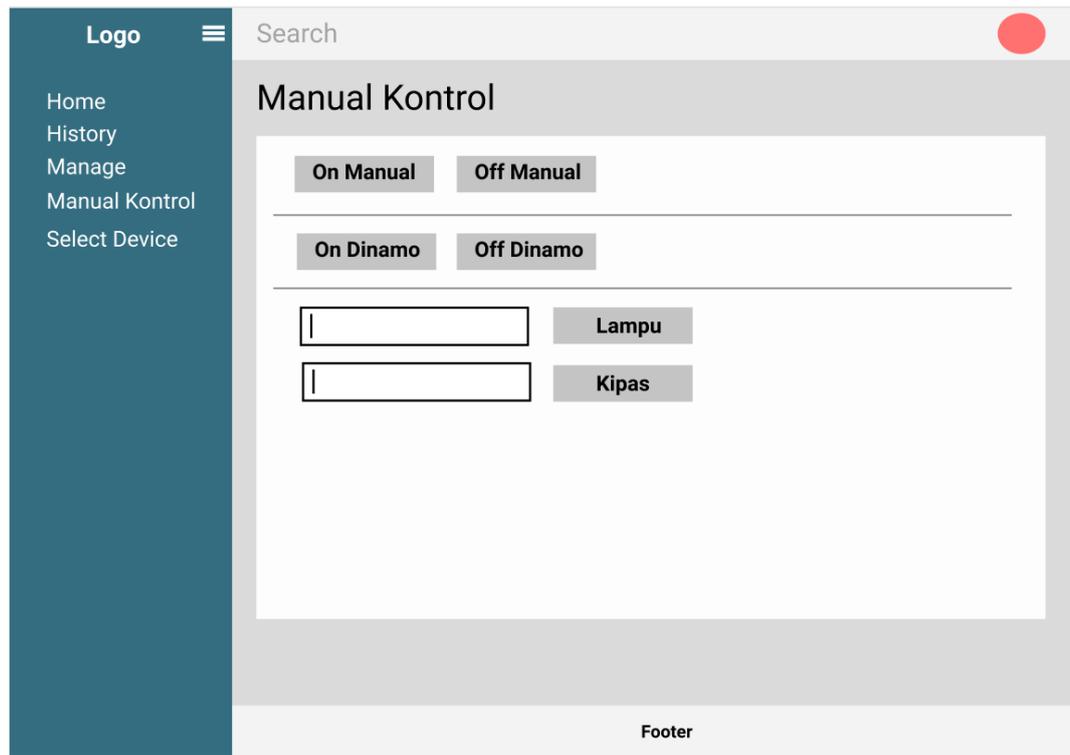
Pada Gambar 4.12 merupakan halaman manage yang mana menampilkan data telur yang akan dimasukkan dalam ruang inkubator, data tersebut meliputi jenis telur, banyak telur, tanggal masuk, dan tanggal keluar dimana tanggal keluar merupakan hasil penjumlahan dari tanggal masuk yang dimasukkan pengguna dijumlah 17 yang akan secara otomatis ditambahkan ke database. Pengguna dapat menambah data ketika akan memasukkan telur, data tersebut akan masuk dalam database dan dapat digunakan untuk arsip dan *history*.



Gambar 4. 12 Halaman Manage *Website*

4. Halaman Manual Control

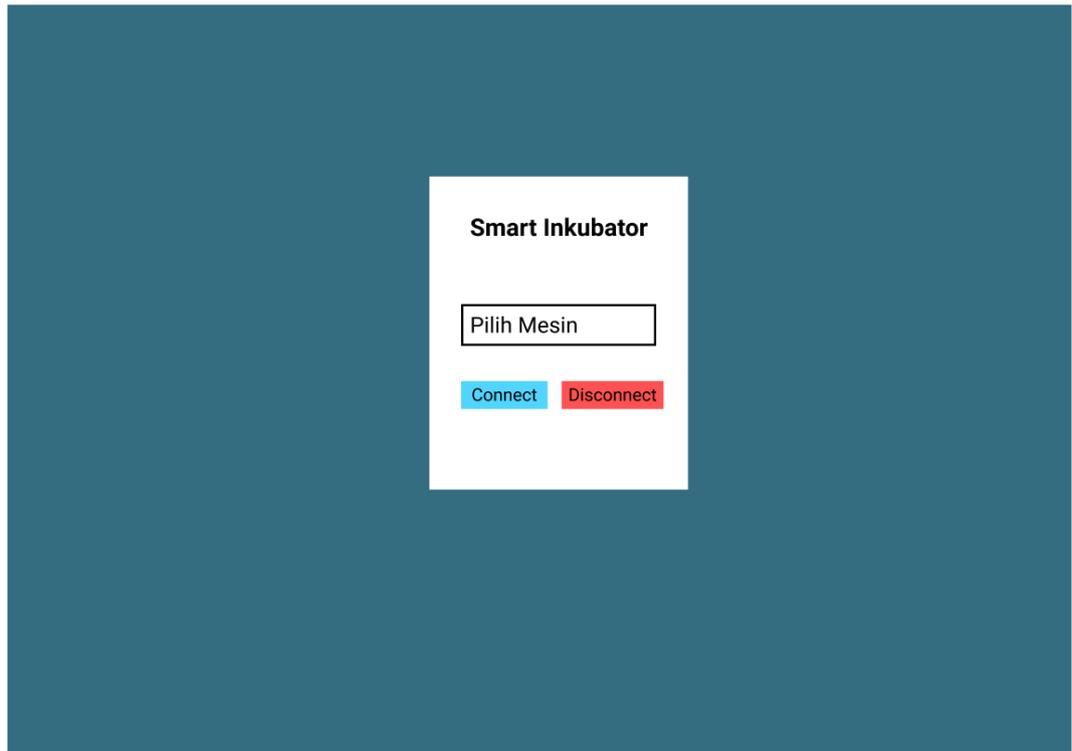
Pada Gambar 4.13 merupakan halaman manual kontrol dari sistem aplikasi, kontrol manual berarti mematikan kontrol otomatis dimana pada halaman ini pengguna dapat mengontrol sistem alat ruang inkubator secara manual. Kontrol manual antara lain kontrol on / off dinamo, mengatur intensitas cahaya lampu dengan memasukkan angka 1 – 100, dan mengatur kecepatan kipas dengan memasukkan angka 1 – 1000.



Gambar 4. 13 Halaman Manual Kontrol *Website*

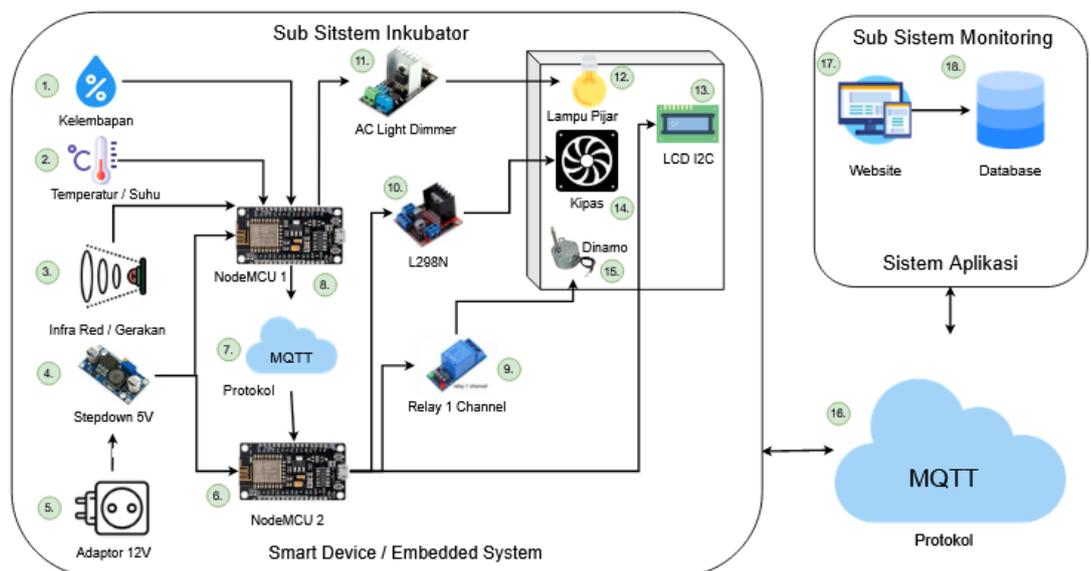
5. Halaman *Select Device*

Pada Gambar 4.14 merupakan halaman *select device* yang mana pengguna dapat memilih device atau mesin mana yang ingin dimonitoring dengan cara memilih mesin pada menu *dropdown* kemudian menekan tombol *connect* maka *dashboard* akan menampilkan data sesuai dengan mesin yang telah dipilih. Selain itu pengguna juga dapat melakukan *disconnect* mesin dengan cara memilih mesin yang akan di-*disconnect* kemudian menekan tombol *disconnect* maka *dashboard* tidak akan menampilkan data.



Gambar 4. 14 Halaman *Select Device*

4.5.8 Arsitektur Sistem



Gambar 4. 15 Arsitektur Sistem Prototipe

Pada arsitektur sistem prototipe yang ditunjukkan pada Gambar 4.15 terdiri dari 2 sub sistem, yaitu sebagai berikut :

1. Sub Sistem Inkubator

Pada sub sistem inkubator 2 inputan data yang digunakan untuk pengolahan *fuzzy* sugeno yaitu suhu dan kelembapan ditunjukkan pada nomor 1 dan 2. Selain itu terdapat data infrared dan gerakan dari sensor pir ditunjukkan pada nomor 4 digunakan untuk mendeteksi telur yang telah menetas. *Stepdown* digunakan untuk menurunkan tegangan dari adaptor 12v menjadi tegangan 5V untuk menyalakan kedua mikrokontroller ditunjukkan pada nomor 4 dan 5. Semua data inputan diolah oleh NodeMCU 1 dan di-*publish* menggunakan protokol mqtt menuju NodeMCU 2 ditunjukkan pada nomor 6, 7, 8. Hasil perhitungan *fuzzy* pada NodeMCU 1 digunakan sebagai *input* dari *AC light dimmer* untuk menyalakan lampu ditunjukkan pada nomor 11 dan 12, kemudian hasil *output fuzzy* kipas digunakan sebagai *input* dari modul l298n untuk menyalakan kipas ditunjukkan pada nomor 10 dan 14. Relay 1 channel digunakan untuk menyalakan dan mematikan *dynamo synchronus* yang dikontrol oleh NodeMCU 2 ditunjukkan pada nomor 9 dan 15. Terdapat juga lcd I2C yang digunakan untuk menampilkan suhu, kelembapan, kekuatan lampu, dan kecepatan kipas ditunjukkan pada nomor 13.

2. Sub Sistem *Monitoring*

Pada sub sistem *monitoring*, *website* akan mensubscribe topic yang sama dengan topic yang di-*publish* oleh NodeMCU 1 dan 2 dimana *website* akan menampilkan data suhu, kelembapan, status telur menetas, kekuatan lampu dan kecepatan kipas secara *realtime* ditunjukkan pada nomor 17. Komunikasi antara sub sistem *monitoring* dan sub sistem inkubator menggunakan protokol mqtt yang ditunjukkan pada nomor 16. Data yang telah diterima akan disimpan dalam database ditunjukkan pada nomor 18.