

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **3.1 Analisis Sistem**

Analisis sistem adalah teknik pemecahan masalah yang menguraikan bagian-bagian komponen dengan mempelajari seberapa bagus bagian-bagian komponen tersebut bekerja dan berinteraksi untuk mencapai tujuan sistem yang akan dibuat (Kurniarum, 2013).

##### **3.1.1 Analisis Peran Sistem**

Sistem yang dirancang memiliki peranan sebagai berikut :

1. Sistem yang dibuat dapat mengukur suhu dan kelembaban, mengukur ketersediaan pakan pada tempat penampungan.
2. Dari informasi suhu dan kelembaban yang diukur, sistem dapat menyalakan lampu dan kipas secara otomatis.
3. Sistem dapat melakukan pemberian pakan berdasarkan waktu yang telah ditentukan dan mengirim perintah pakan tambahan.

##### **3.1.2 Analisis Peran Pengguna**

Sistem ini dirancang khusus untuk membantu peternak dalam pengolahan peternakan. Peternak dapat menentukan minimal dan maksimal suhu, yang menjadi acuan untuk mengontrol aktuator, menentukan waktu kapan pemberian pakan dilakukan dan memberikan makanan tambahan pada ayam ketika dibutuhkan.

### 3.1.3 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mengetahui fasilitas yang harus disediakan atau dimiliki oleh sistem agar dapat melayani kebutuhan peternakan. Komponen yang dibutuhkan berupa sensor dan mikrokontroler sebagai hardware, Server dan antarmuka *web* sebagai *software*. Fungsi utama dari sistem ini adalah untuk menghubungkan perangkat *IOT* , Server dan antarmuka *web* untuk melakukan pengolahan peternakan.

#### a) Kebutuhan Perangkat Keras

Tabel 3.1 Kebutuhan Perangkat Keras

Jenis Perangkat	Nama Perangkat	Spesifikasi
Komputer/Server	Asus VivoBook X409FJ	<ul style="list-style-type: none"><li>– Intel(R) Core(TM) i5-8265U CPU @ 1.60 GHz</li><li>– 8,00 GB Ram</li><li>– 64bit operating system</li></ul>
Microcontroller	NodeMcu Microcontroller ESP32U,	<ul style="list-style-type: none"><li>– Operating Voltage 3 - 5v</li><li>– Dual Core 32-bit</li><li>– Digital I/O Pins 32</li><li>– Clock Speed 80MHz - 240MHz</li><li>– Memory Flash: 4 MB</li></ul>

	ESP32Cam	<ul style="list-style-type: none"> <li>- <i>Low - Power Dual - Core 32bit CPU</i></li> <li>- <i>Clock Speed 80MHz - 240MHz</i></li> <li>- <i>Built-in 520KB SRAM, Camera OV2640 Built-in Flash</i></li> </ul>
Sensor	Sensor Ultrasonik  HCSR04	<ul style="list-style-type: none"> <li>- Tegangan <i>input</i>: 5v</li> <li>- Jangkauan : 3 – 350 cm</li> <li>- Frekuensi Ultrasonik: 40 kHz</li> </ul>
	<i>Water Level</i>  Sensor	<ul style="list-style-type: none"> <li>- Tegangan input 3 – 5v</li> <li>- Area Deteksi 40 mm x 16 mm</li> </ul>
	DHT22  Sensor suhu dan kelembaban,	<ul style="list-style-type: none"> <li>- Tegangan input 3,5 – 5v</li> <li>- Akurasi <math>\pm 20C</math> (<i>temperature</i>) <math>\pm 5\%</math> RH (<i>humidity</i>),</li> <li>- <i>Range</i> Suhu dan Kelembaban</li> <li>- (-40<sup>0</sup>C sampai 80<sup>0</sup>C) dan 20% – 90% RH</li> </ul>
	Lampu Pijar	<ul style="list-style-type: none"> <li>- Tegangan : 220v AC</li> <li>- <i>Watt</i> : 20 Watt</li> </ul>

Aktuator	Fan / Kipas	<ul style="list-style-type: none"> <li>- Tegangan : 12V DC</li> <li>- Ukuran: 120 x 120 x 25mm</li> </ul>
	Pompa Air	<ul style="list-style-type: none"> <li>- Tegangan : 12V DC</li> <li>- <i>Flow</i> : 4.0 Lpm, Amps : 3.5A,</li> <li>- <i>Pressure</i> : 0.6 Mpa</li> </ul>
	Motor Servo	<ul style="list-style-type: none"> <li>- <i>Type</i> : MG996R</li> <li>- Berat 55g</li> <li>- <i>Stall torque</i> 9.4kgfcm</li> <li>- 180° putaran</li> </ul>
	Modul Relay	<ul style="list-style-type: none"> <li>- <i>High and low trigger</i> 5v</li> <li>- 1 dan 4 <i>Channel</i> dengan <i>octocopuler</i></li> </ul>
Perangkat tambahan lainnya	Adaptor AC DC	<ul style="list-style-type: none"> <li>- Tegangan : 120 Watt</li> <li>- <i>Input</i> : 110-240V/50-60Hz</li> <li>- <i>Output</i> : 12V DC</li> </ul>

b) Kebutuhan Perangkat Lunak

Penggunaan perangkat lunak *Server* seperti *Node.js*, *Express.js*, *MongoDB*, dan *Socket.IO* sebagai fondasi utama yang mengelola logika aplikasi, data di *Server*, dan berinteraksi dengan perangkat *IOT* dan antarmuka *web*. Di sisi klien, diperlukan penggunaan library seperti *Socket.IO - client*, *Vue.js*, dan *ApexCharts* untuk menciptakan antarmuka pengguna yang responsif, visualisasi data, manajemen waktu, serta mendukung komunikasi *real-time* antara klien dan *Server*.

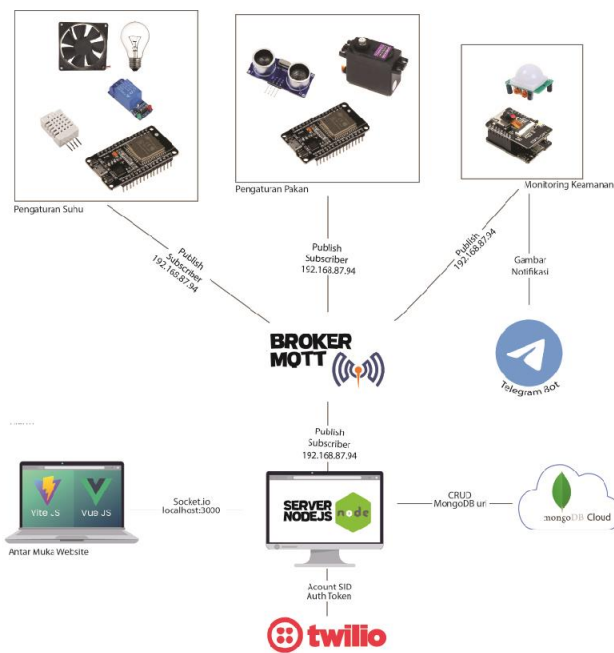
Tabel 3.2 Kebutuhan Perangkat Lunak

<i>Server</i>	<i>Client</i>
<i>Node.js</i>	<i>Vite.js</i>
<i>Express.js</i>	<i>Vue.js</i>
<i>MongoDB</i>	<i>ApexChart</i>
<i>MQTT</i>	<i>Socket.IO – client</i>
<i>Socket.IO</i>	

### 3.2 Perancangan Sistem

Sistem memiliki 3 *node* utama yaitu suhu, pakan, dan keamanan. *Node* suhu memantau dan merespon perubahan suhu. *Node* pakan mengelola pakan dan memberi pemberitahuan pengisian ulang. *Node* keamanan menggunakan sensor *PIR* untuk mendeteksi aktivitas mencurigakan, memberikan respons seperti pemberitahuan *via Telegram Bot*. Integrasi dengan *MongoDB Cloud*, *MQTT*

*Broker*, laptop, dan *WSN* serta *IOT* memungkinkan pengelolaan secara terintegrasi. Kebutuhan masing-masing *node* sekaligus kebutuhan dari perangkat lunak yang digunakan dapat dilihat pada (Gambar 3.1) berikut ini :



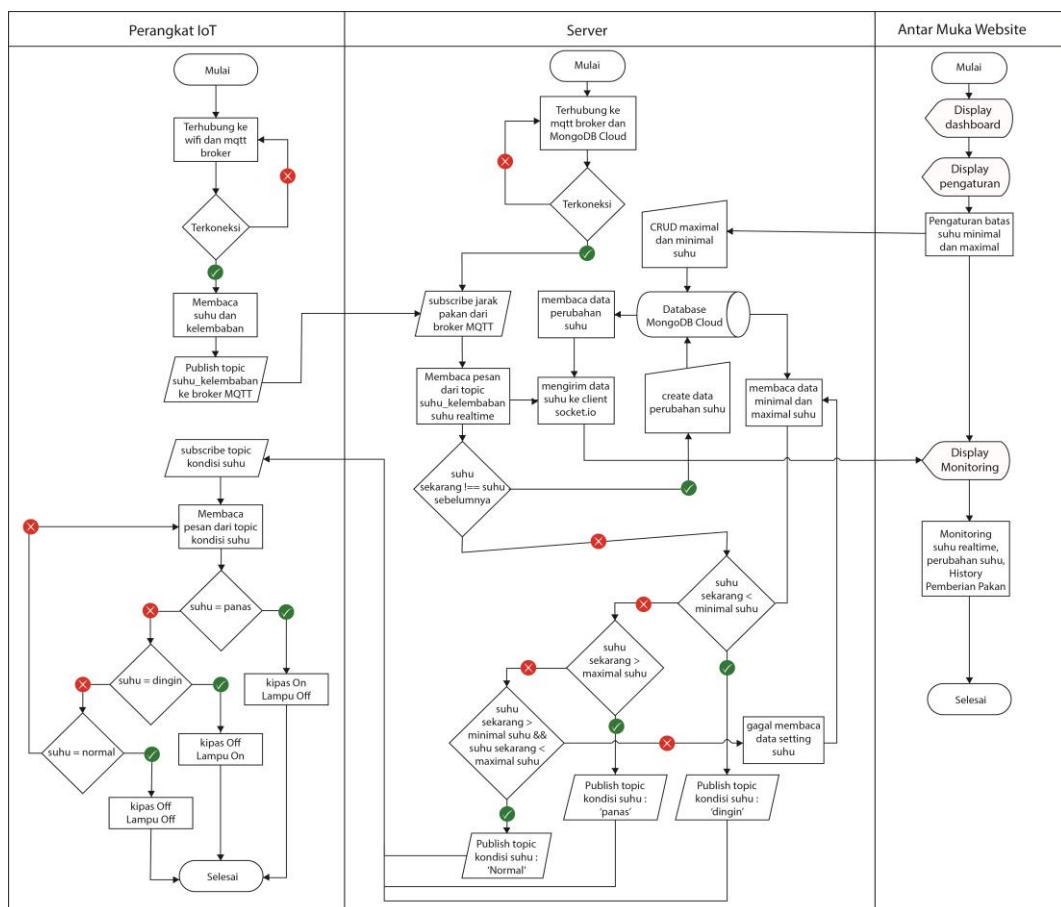
Gambar 3.1 Arsitektur Keseluruhan Sistem

### 3.2.1 Alur Sistem (*Flowchart*)

#### a. Sistem Suhu dan kelembaban

Sistem kontrol suhu dan kelembaban menggunakan perangkat *IOT* dengan ESP32 yang terhubung ke *broker MQTT*. Data suhu dan kelembaban di-*publish* dalam format JSON. *Server* terhubung ke *MQTT Server* dan *MongoDB Cloud*, melakukan *subscribe* terhadap topik suhu\_kelembaban, serta meneruskan data *real-time* ke antarmuka *web* melalui *Socket.IO*. *Server* juga menyimpan perubahan suhu dalam *collection* PerubahanSuhu di *database* jika terjadi perbedaan dengan data sebelumnya. Pengguna dapat memantau *real-time* suhu dan

perubahan suhu, sambil mengatur batas suhu minimal dan maksimal untuk mengontrol aktuator di kandang ayam pada halaman pengaturan. Perangkat *IOT* juga mengontrol aktuator berdasarkan pesan yang diterima dari *subscribe* topik kondisi suhu dari *MQTT Broker*. Alur kerja sistem pengaturan suhu dan kelembaban dapat dilihat pada Gambar 3.2 berikut ini :



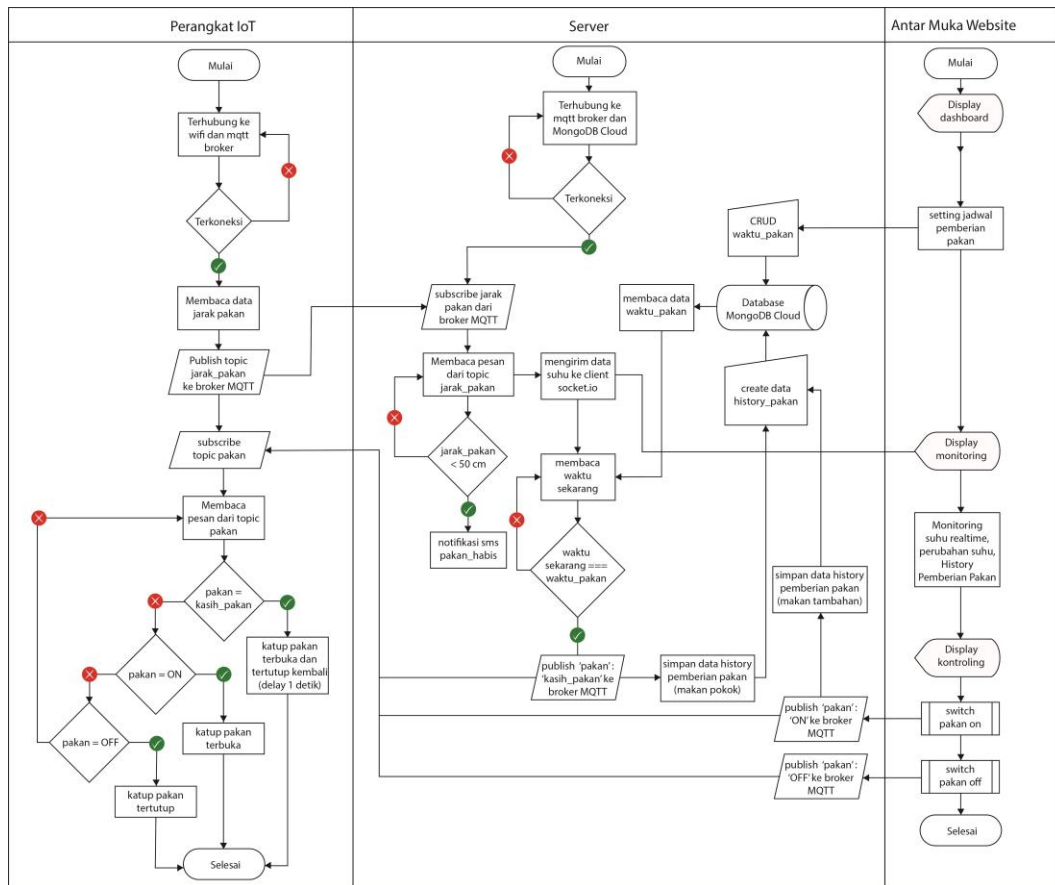
Gambar 3.2 Flowchart Sistem suhu dan kelembaban

## b. Sistem Pengaturan Pakan

Sistem Pengaturan Pakan Perangkat *IOT* dimulai dengan membaca data jarak pakan dan mem-*publish* ke *MQTT Broker* dengan topik "jarak\_pakan". Selanjutnya, sistem melakukan *subscribe* terhadap topik

"pakan" untuk menggerakkan servo agar dapat membuka dan menutup katup pakan. *Server* membaca jarak pakan secara *real-time* dan mengirimkan informasi tersebut ke *client Socket.IO*. Selain itu, *Server* mengirim notifikasi kepada peternak jika pakan hampir habis. *Server* juga mem-*publish* topik "pakan" dengan kondisi tertentu. Antarmuka *web* menampilkan informasi *real-time* mengenai jarak pakan. Peternak memiliki kemampuan untuk menentukan waktu pemberian pakan dan dapat melakukan kontrol manual pada halaman *controlling*. Dengan demikian, sistem ini memberikan kontrol yang efektif terhadap pemberian pakan dengan memanfaatkan teknologi *IOT*, memastikan peternak dapat memantau kondisi pakan secara *real-time* dan melakukan intervensi manual jika diperlukan. Untuk lebih memahami cara kerja sistem pengaturan pakan dapat dilihat pada (Gambar 3.3)

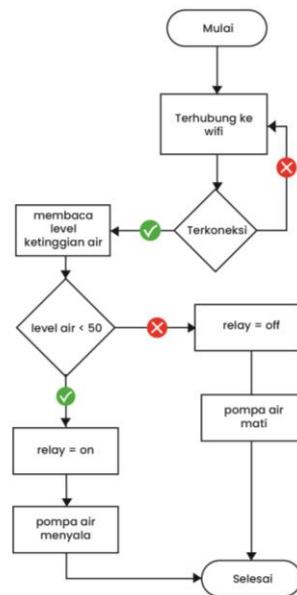




Gambar 3.3 Flowchart Sistem pengaturan pakan

### c. Sistem Pemberian Air

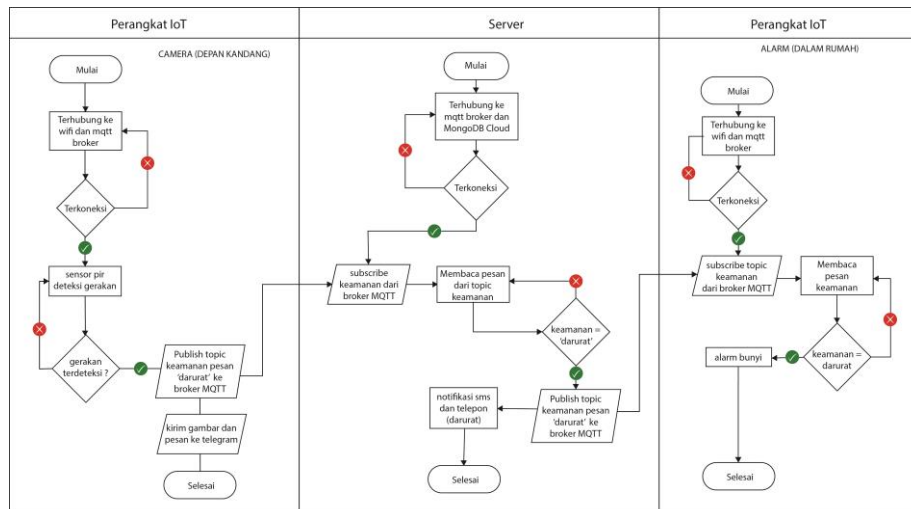
Sistem ini beroperasi secara otomatis dan tidak memerlukan kontrol dari peternak atau pesan dari *Server*, sehingga peternak tidak perlu mengatur lagi untuk kapan pemberian air dilakukan. Alur kerja sistem dapat dilihat pada (Gambar 3.4)



Gambar 3.4 *Flowchart* Sistem Pemberian Air

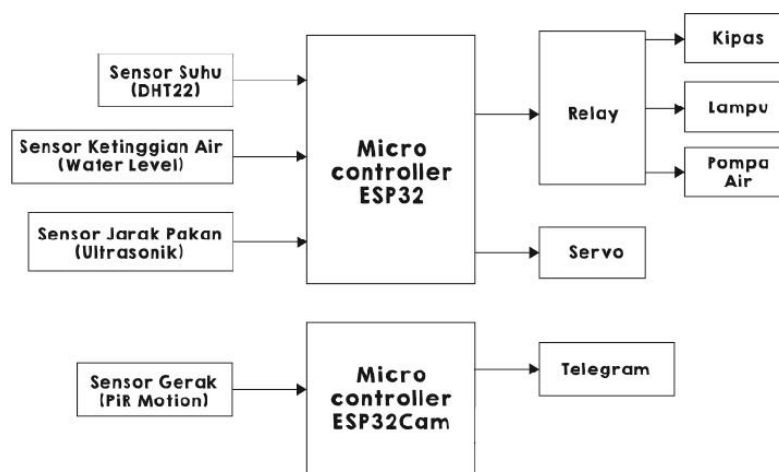
#### d. Sistem *Monitoring* Keamanan

Sistem *Monitoring* Keamanan kandang ayam dengan kamera ESP32 dan sensor *PIR*. Algoritma deteksi cerdas mengidentifikasi gerakan mencurigakan. Sensor *PIR* mendeteksi gerakan, mengirim gambar ke *bot* Telegram untuk notifikasi visual yang tidak melibatkan antar muka *website*, melainkan penggunaan sistem keamanan yang bekerja antara perangkat keras dengan perangkat keras. Dengan rincian pada (Gambar 3.5) ini, dapat dengan mudah dipahami bagaimana setiap komponen berkontribusi dalam menjalankan fungsi sistem *Monitoring* keamanan kandang ayam



Gambar 3.5 Flowchart Sistem Monitoring Keamanan

### 3.2.2 Perancangan Perangkat IOT

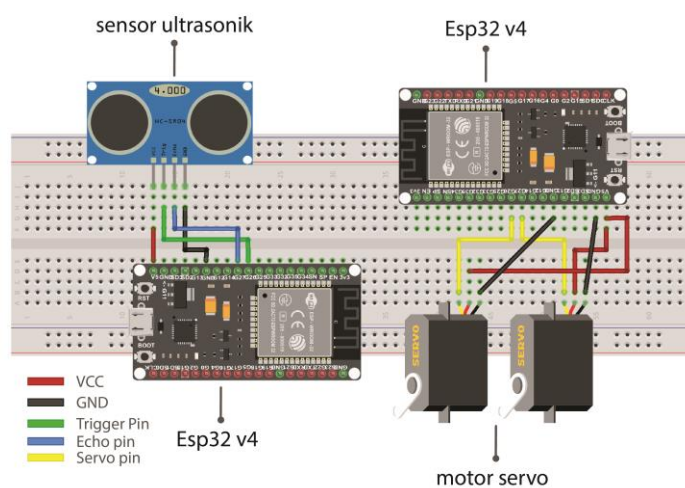


Gambar 3 6 Blok Diagram

Sistem pengendalian dan pemantauan lingkungan kandang ayam menggunakan ESP32 dan sensor-sensor seperti DHT22, *Water Level Sensor*, Ultrasonik, dan *PIR*. Sensor DHT22 digunakan untuk mengukur suhu dan kelembaban, dengan ESP32 mengontrol kipas dan lampu untuk menjaga suhu optimal. Sensor *Water Level* mendeteksi level air, dan jika rendah, ESP32 mengaktifkan pompa air. Sensor Ultrasonik me-*monitoring* persediaan pakan

dan memberikan pemberitahuan jika mendekati batas rendah. Sementara itu, ESP32 mengendalikan servo untuk pemberian pakan sesuai jadwal. Sensor *PIR* digunakan untuk keamanan, memberikan notifikasi jika mendeteksi gerakan mencurigakan. Kegunaan dari masing-masing alat di kelompokan menjadi tiga, sesuai cara kerja alat yang dapat dilihat pada (Gambar 3.6), alat *input* (kiri gambar), proses (tengah) dan yang kanan merupakan alat *output*. Penerimaan nilai atau pengkondisian cerdas dilakukan oleh mikrokontroller sebagai otak dari sistem diatas. Dari semua nilai atau data sensor yang dibaca dari sensor-sensor yang berbeda kemudian di proses dengan logika oleh ESP32 dan memberikan perintah agar terjadi sesuatu pada alat outputnya. Dengan demikian, sistem ini memberikan solusi terpadu untuk menjaga keseimbangan suhu, air, pakan, dan keamanan di kandang ayam dengan mudah dan efisien.

a) Pengaturan pakan

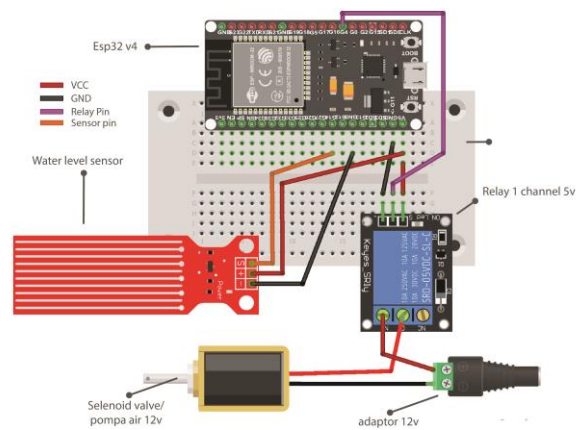


Gambar 3.7 Rangkaian Pengaturan Pakan

1. Pin Triger sensor ultrasonik ke pin GPIO 16 ESP32
2. Pin Echo sensor ultrasonik ke pin GPIO 17 ESP32

3. Pin VCC sensor ultrasonik ke pin 5V input ESP32
4. Pin GND sensor ultrasonik ke pin GND ESP32
5. Pin data servo 1 ke pin GPIO 25 ESP32
6. Pin data servo 2 ke pin GPIO 26 ESP32
7. Pin VCC servo 1 dan 2 ke pin 5V input ESP32
8. Pin GND servo 1 dan 2 ke pin GND ESP32

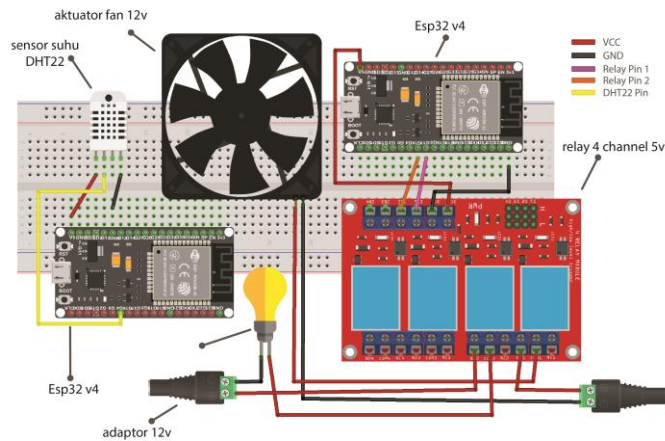
b) Pengaturan air minum



Gambar 3.8 Rangkaian Pengaturan Air Minum

1. Pin data *water level sensor* ke pin GPIO 4 ESP32
2. Pin VCC *water level sensor* ke pin 5V input ESP32
3. Pin GND *water level sensor* ke pin GND ESP32
4. Pin kontrol *relay* ke pin GPIO 26 ESP32
5. Pin NC *relay* pompa ke kabel VCC kipas
6. Pin COM *relay* pompa ke kabel VCC adaptor
7. Pin NC pompa kipas ke kabel VCC pompa
8. Pin COM *relay* pompa ke kabel VCC adaptor pompa
9. Kabel GND pompa kes GND adaptor pompa

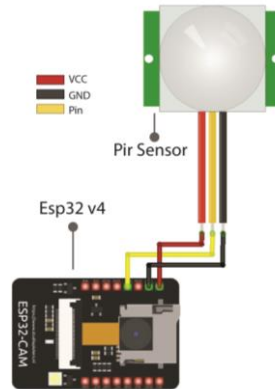
c) *Monitoring suhu*



Gambar 3.9 Rangkaian Pengaturan Suhu

1. Pin data sensor suhu DHT22 ke pin GPIO 4 ESP32
2. Pin VCC sensor suhu DHT22 ke pin 5V input ESP32
3. Pin GND sensor suhu DH22 ke pin GND ESP32
4. Pin kontrol *relay* kipas ke pin GPIO 16 ESP32
5. Pin kontrol *relay* lampu ke pin GPIO 17 ESP32
6. Pin NC *relay* kipas ke kabel VCC kipas
7. Pin COM *relay* kipas ke kabel VCC adaptor kipas
8. Kabel GND kipas kes GND adaptor kipas
9. Pin NC *relay* lampu ke kabel VCC lampu
10. Pin COM *relay* lampu ke kabel VCC adaptor lampu

d) *Monitoring* keamanan

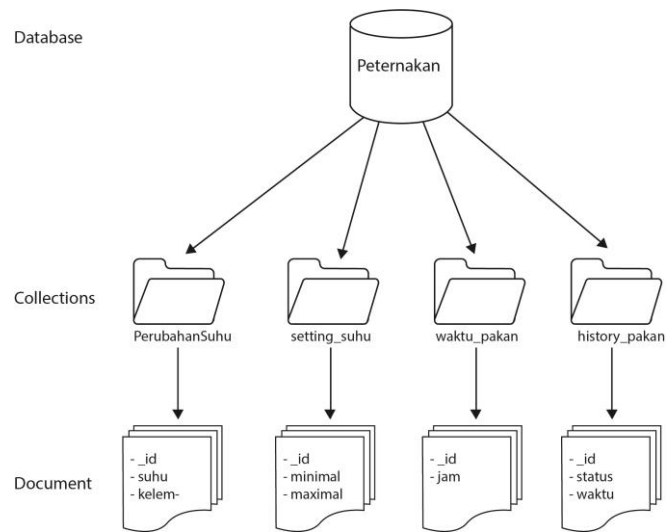


Gambar 3.10 Rangkaian *Monitoring* Keamanan (kamera)

1. Pin data *PIR* sensor ke pin GPIO 13 ESP32CAM
2. Pin VCC *PIR* sensor ke pin 5V input ESP32CAM
3. Pin GND *PIR* sensor ke pin GND ESP32CAM

### 3.2.3 Perancangan *Database*

*MongoDB* dipilih untuk implementasi *IOT* pada peternakan karena fleksibilitas skema yang memungkinkan penyimpanan data dari berbagai sensor dan perangkat *IOT* tanpa memerlukan skema tetap, memberikan adaptabilitas terhadap variasi format data yang dihasilkan. Tabel pada *database* disebut dengan *collections*, dan data yang terdapat dalam tabel disebut sebagai *document* yang menyimpan data dalam format BSON (Binary JSON) yang dapat berisi berbagai jenis data. Adapun implementasi *database MongoDB* dapat dilihat pada (Gambar 3.12).

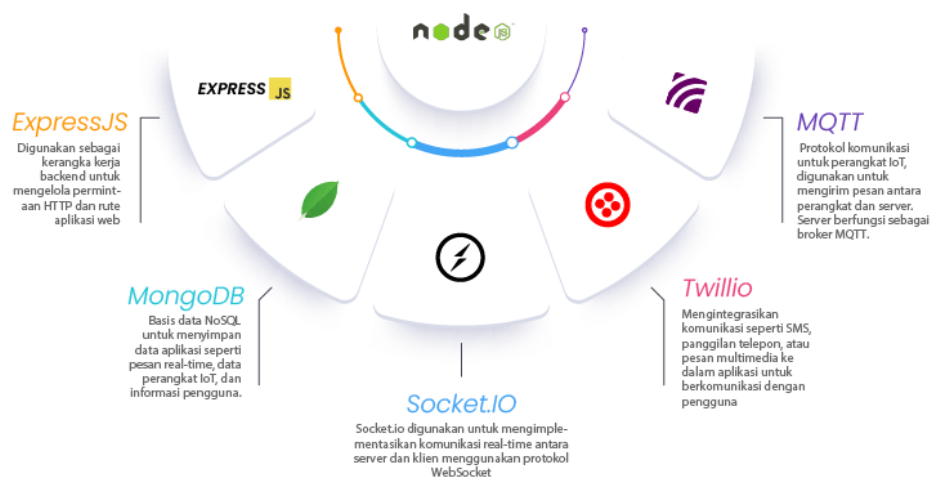


Gambar 3. 11 Struktur *Database*

### 3.2.4 Perancangan *Server*

Perangkat lunak *Server* menggunakan protokol komunikasi *MQTT* untuk menerima data dari perangkat *IOT* dan mengelola permintaan *HTTP* dari klien. Terintegrasi dengan *MongoDB* untuk menyimpan data, menggunakan *Express.js* sebagai kerangka kerja *HTTP*, *Socket.IO* untuk komunikasi *real-time* dua arah, dan *Twilio* untuk mengelola pesan notifikasi. Ini menciptakan pengalaman komunikasi terpadu antara perangkat *IOT*, pengguna, dan sistem basis data. Teknologi apa saja yang digunakan pada *Server* dapat dilihat pada (Gambar 3.12).





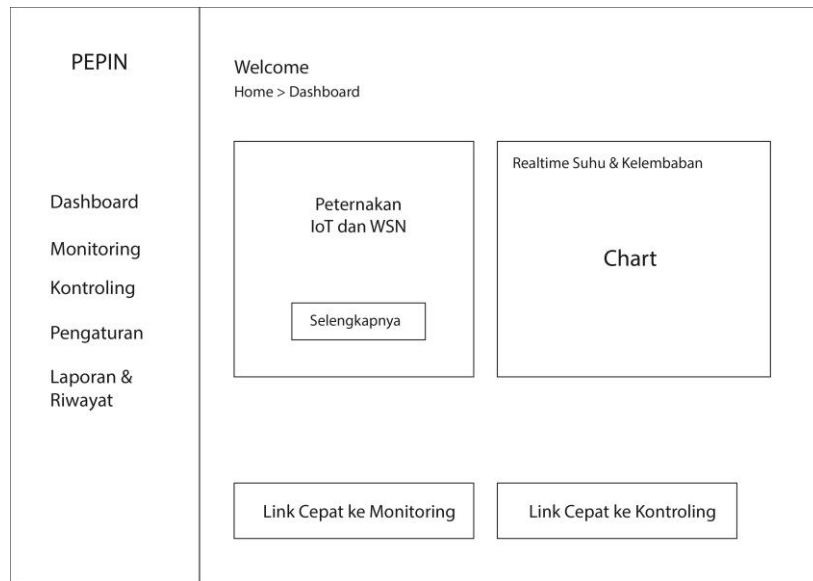
Gambar 3.12 Server NodeJS

### 3.2.5 Perancangan Antarmuka

Perancangan antarmuka bertujuan agar pengguna dan komputer dapat saling berinteraksi sehingga pengguna merasakan adanya kemudahan dari sistem komputer. Adapun desain *interface* dari penerapan teknologi *WSN* dan *IOT* untuk *me-monitoring* dan mengontrol peternakan adalah sebagai berikut:

a. Halaman *Dashboard*

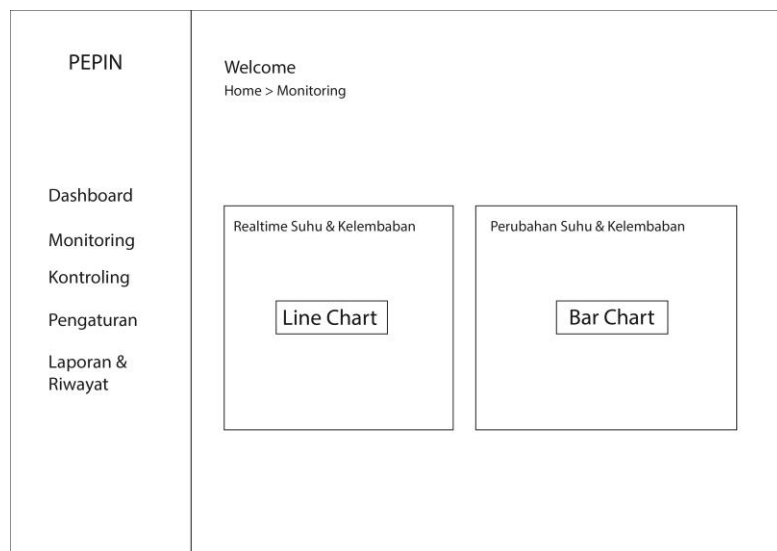
Halaman *dashboard* adalah titik awal yang memberikan pandangan singkat tentang kondisi suhu dan kelembaban *real-time*. Ini membantu peternak untuk mendapatkan gambaran cepat tentang kandang dan memutuskan apakah perlu tindakan koreksi. (Gambar 3.13).



Gambar 3.13 Desain Halaman *Dashboard*

b. Halaman *Monitoring*

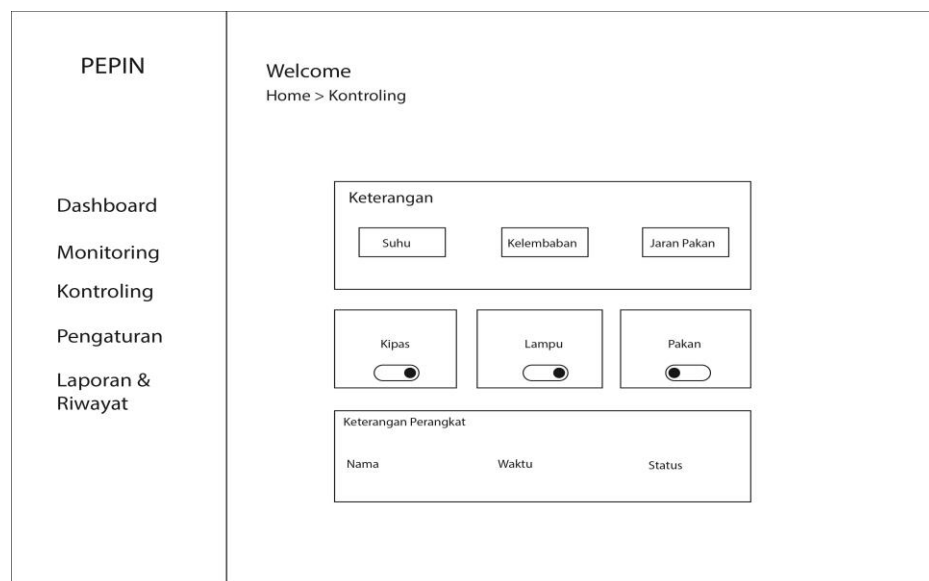
Halaman *Monitoring* memberikan tampilan rinci tentang data *real-time* dari sensor suhu dan jarak. Dan menampilkan chart *real-time* suhu dan perubahan suhu yang terjadi pada (Gambar 3.14).



Gambar 3.14 Desain Halaman *Monitoring*

c. Halaman *Controlling*

Peternak dapat mengendalikan perangkat seperti pengatur suhu, pemberian pakan, atau lampu pemanas secara manual jika diperlukan. Halaman kontrol juga dapat mengintegrasikan sistem otomatisasi yang memungkinkan peternak mengatur parameter kandang, seperti suhu yang diinginkan atau jadwal pemberian pakan (Gambar 3.15).



Gambar 3.15 Desain Halaman *Controlling*

d. Halaman Pengaturan

Peternak memiliki kemampuan untuk mengelola dan mengontrol berbagai aspek operasional kandang secara tepat dan efisien. Melalui halaman ini, peternak dapat mengatur parameter penting seperti suhu yang diinginkan dalam kandang, yang sangat berpengaruh pada kesejahteraan hewan ternak, serta jadwal pemberian pakan yang memastikan nutrisi yang tepat pada waktu yang tepat. Dalam hal ini, halaman pengaturan memberikan fleksibilitas kepada peternak untuk

menyesuaikan operasi sesuai dengan kebutuhan khusus, baik secara manual maupun dengan mengintegrasikan sistem otomatisasi yang dapat memantau dan mengendalikan perangkat kandang secara otomatis. Ini membantu meningkatkan efisiensi produksi, memastikan kualitas hidup hewan ternak, dan mengoptimalkan hasil peternakan secara keseluruhan (Gambar 3.16).

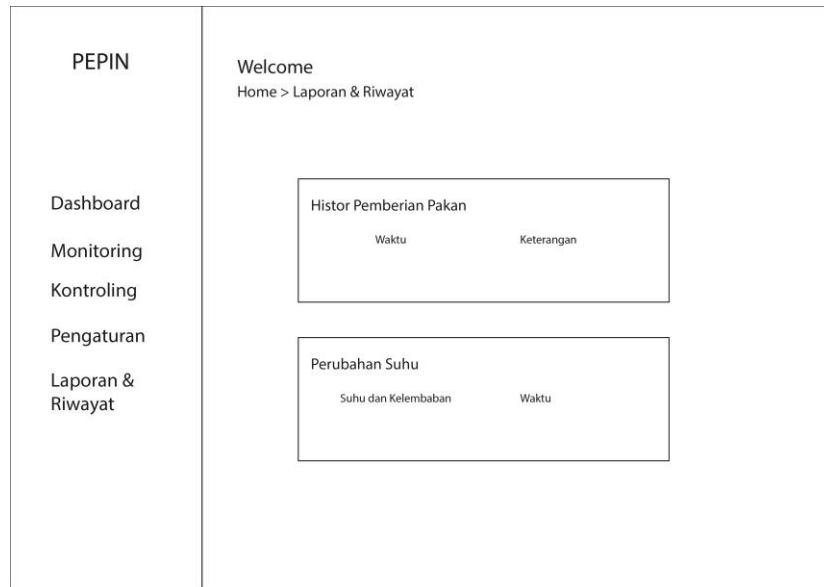
PEPIN	Welcome Home > Pengaturan
Dashboard	Atur Waktu Pemberian Pakan 00:00 [Clock Icon] [Simpan]
Monitoring	Jadwal Pemberian Pakan
Kontroling	Waktu      Keterangan      Aksi [edit] [hapus]
Pengaturan	Setting suhu
Laporan & Riwayat	Minimal:      Maximal:      [edit] [setting]

Gambar 3.16 Desain Halaman Pengaturan

e. Halaman Laporan

Halaman laporan sistem peternakan menyajikan informasi historis, termasuk perubahan suhu dan catatan pemberian pakan. Ini memungkinkan pemantauan evolusi suhu dan detail pakan, mendukung pemahaman tren kandang, kesejahteraan hewan, dan performa produksi. Laporan ini juga berfungsi sebagai alat deteksi masalah atau perubahan

yang tidak diinginkan, memfasilitasi pengambilan tindakan perbaikan dalam manajemen peternakan (Gambar 3. 17).



Gambar 3. 17 Desain Halaman Laporan