

BAB I

PENDAHULUAN

1.1. Latar Belakang

Secara umum hal yang paling diperhatikan dalam memelihara hewan peliharaan adalah pembagian makanan yang harus diberikan secara teratur supaya hewan peliharaan yang kita miliki tidak mengalami kekurangan gizi. Namun pada kondisi sekarang ini banyak para pemelihara hewan yang sibuk bekerja namun lupa untuk memberikan makanan pada hewan peliharaannya, dan dapat menyebabkan kematian pada hewan peliharaannya. Maka dari itu ada urgensi untuk pembuatan alat yang memanfaatkan teknologi IoT (*Internet Of Things*) dan *robotika* dapat memberi makanan terhadap hewan pemeliharaan yang menerima perintah dari jarak jauh untuk bergerak secara otomatis memberikan makanan ke wadah. Mekanisme yang digunakan sebagai perintah jarak jauh adalah Protokol *Websocket* dan *RestAPI* dengan mikrokontroler *NodeMCU ESP32* yang berisikan prototype sistem IoT menggunakan jaringan nirkabel (WiFi) yang menghubungkan node node dimana setiap node berisikan sensor atau aktuator yang dikendalikan oleh mikrokontroler.

Alur kerja sistem disini yakni aplikasi *mobile* berisikan instruksi pemberian makan dan volume isi tempat makanan utama yang dideteksi dari sensor ultrasonik secara online melalui jaringan internet. *User* memberikan instruksi pemberian makan sesuai jadwal jam yang telah diinput melalui aplikasi *mobile*. Semua instruksi yang ada di aplikasi *mobile* ini akan

diteruskan ke sistem *webserver* dan database. Setiap aktivitas data yang telah diproses di *webserver* akan diteruskan Mikrokontroler *NodeMCU ESP32* untuk mengatur jalannya alat pemberi makan dan motor servo menerima instruksi membuka katup makanan untuk menuangkan ke wadah. Sensor ultrasonik pun akan menerima instruksi kembali dari Mikrokontroler *NodeMCU ESP32* untuk mendeteksi volume sisa makanan pada wadah utama setiap kali intruksi pemberian makan.

Berdasarkan latar belakang tersebut maka penulis mengembangkan IoT (*Internet Of Things*) dengan menggabungkan Aplikasi *Mobile (Smartphone)* menggunakan *ReactNative.JS*, *Webserver* menggunakan *Laravel*, dan Mikrokontroler *NodeMCU ESP32*.

IoT (*Internet Of Things*) adalah teknologi komunikasi antar mesin dengan menggunakan internet dengan bentuk komunikasi *Machine-to-Machine* yang bekerja dengan menggunakan argumentasi pemrograman dimana setiap perintah argument dapat menghasilkan suatu interaksi pada alat atau mesin yang telah terkoneksi otomatis tanpa bantuan manusia dan tanpa batasan jarak. Fungsi internet di sini menjadi faktor penting dalam penghubung antara kedua interaksi. Manfaat penggunaan teknologi IoT (*Internet Of Things*) mampu mempermudah proses usaha interaksi karena mesin atau alat yang digunakan dapat bekerja 24 jam penuh tanpa istirahat, atau bisa disebut dengan *real time*. Internet disini sebagai konektivitas antara sensor atau perangkat yang akan saling berkomunikasi di *cloud*. Data dari sensor yang dikirim ke *cloud* akan diproses oleh software yang akan menentukan *action* selanjutnya. *Action* ini bisa berbentuk pengiriman *alert*, penyesuaian jadwal, penutupan akses pada

alat, atau lainnya. (*Internet Of Things*) dapat juga dikontrol oleh user lewat dashboard dari komputer, laptop, atau *mobile app* lainnya. *User* atau pengguna yang sudah diberikan izin dapat mengatur dan mengubah *action* dan *rules* sesuai kebutuhan. Perubahan ini akan dikirim lagi ke *cloud* lalu sensor yang berkaitan akan segera ter-update sesuai dengan apa yang kita perintah. Namun tanpa konektivitas, data pada *device* tidak akan sampai ke sistem. Sarana komunikasi device dengan sistem IoT bisa beragam. Mulai dari koneksi selular, satelit, WiFi, bluetooth, *low power wide area network* (LPWAN), dan lainnya. Pemilihan konektivitas selalu disesuaikan dengan kebutuhan pengguna. Untuk industri yang menggunakan banyak *device* kecil di area yang luas seperti pada pertanian dan penyaluran listrik, LPWAN adalah jenis konektivitas tepat. Sedangkan untuk industri finansial yang memerlukan keamanan tinggi, *SD-WAN* dan *Managed Service Connectivity*. Peran pengguna pada teknologi IoT bertugas sebagai pengawas dan pengatur alat yang bekerja secara langsung. Namun adapun tantangan yang dihadapi dalam teknologi IoT yakni cara penyusunan jaringan komunikasi, sebab memerlukan sistem keamanan yang ketat. Pengembangan teknologi IoT terbilang mahal yang dimana sistem *Webserver* harus menggunakan *VPS* guna memberikan kostumisasi pada sistem yang dibuat, contohnya *Websocket* dan *Pusher*.

WebSocket adalah standar baru untuk komunikasi *realtime* pada *Webserver*. *Websocket* dirancang untuk diterapkan di browser web dan server web, tetapi dapat digunakan oleh aplikasi client atau server. *WebSocket* adalah protokol yang menyediakan saluran komunikasi *full-duplex* melalui koneksi *TCP* tunggal. Lalu-lintas jaringan yang tidak penting dan *latency* dibandingkan

dengan solusi *polling* dan *long-polling* yang telah digunakan untuk mensimulasikan koneksi dua arah dengan cara menjaga dua koneksi tetap terhubung dan mengurangi *traffic* atau lalu lintas jaringan yang tidak perlu dan *latency* menggunakan *full duplex* melalui koneksi tunggal. (A.B., 2017). Developer bisa membuat aplikasi berbasis web secara *real time* atau *real-time apps*. Aplikasi *real time* adalah dimana ketika ada perubahan data, maka saat itu juga website di *browser client* juga ada perubahan atau setidaknya muncul notifikasi. Ada alternatif lain untuk permasalahan developer tersebut, diantaranya adalah metode *polling* dan *long polling*. Metode *polling* ini mengirimkan request data ke server secara terus menerus. Kalau hanya satu *client* yang melakukan request berulang seperti itu, mungkin tidak masalah, tapi bagaimana jika ada beberapa *client* yang mengakses satu server dan berulang kali melakukan metode *Polling*. Maka server akan jadi sibuk dan rentan terkena serangan *DDOS*. Metode *long polling* adalah metode polling dengan interval waktu yang berkala. Jadi request tidak sesering metode *Polling*. Manfaatnya adalah server jauh lebih stabil dibandingkan dengan metode *polling*. Namun permasalahannya sederhana, *Long Polling* tidak menjawab *real time*, karena ada interval waktu yang digunakan.

Pembuatan aplikasi *mobile (smartphone)* dirancang menggunakan framework *ReactNative.JS* yang dapat bekerja di 2 OS yaitu *Android* dan *IOS*. *React Native* memiliki banyak library dan komunitas *open source* yang dapat membantu penulis dalam mengembangkan sebuah aplikasi. *Database Server* penulis menggunakan *RestAPI* yang merupakan salah satu implementasi dari *web service* sebagai sebuah standar untuk pertukaran data antar aplikasi atau

system. Kerja dari *RestAPI* yakni harus ada sebuah *REST server* yang akan menyediakan data ataupun resource. Sebuah *REST client* akan membuat *HTTP* request ke server melalui sebuah global *ID* atau *URIs* dan server akan merespon dengan mengirimkan balik sebuah *HTTP* response sesuai yang diminta client. Metode *REST* yang sering digunakan dalam *HTTP* antara lain, *GET*, *POST*, *PUT*, dan *DELETE*. (Fakhrun & Gumilang, 2018). Namun ada juga beberapa metode lainnya seperti *PACTH* dan *OPTIONS*. *RestAPI* disini berperan sebagai perantara untuk ngobrol dengan server, jadi untuk akses data di sebuah server dilakukan tidak secara direct melainkan melalui *API* ini, sedangkan *REST* ini sebagai aturan-aturan atau perintah apa saja yang diperbolehkan untuk dilakukan oleh client melalui protokol *HTTP*.

1.2. Rumusan Masalah

Berdasarkan uraian pada latar belakang di atas, berikut merupakan perumusan masalah dari pembahasan dan pelaksanaan *Otomatisasi Pemberian Makanan Kucing Jarak Jauh* berbasis *Internet Of Things* :

1. Bagaimana merancang dan membuat sistem untuk *Otomatisasi Pemberian Makanan Jarak Jauh*?
2. Bagaimana cara mengimplementasi *Internet Of Things* terhadap pengaturan jam makan secara online.
3. Bagaimana merancang dan membuat sistem *webservice* untuk komunikasi antara aplikasi dengan alat *Otomatisasi Pemberian Makanan Jarak Jauh* menggunakan protokol *Websocket* dan *RestAPI*?
4. Bagaimana merancang dan membuat *Alat Otomatisasi Pemberian Makanan Jarak Jauh* menggunakan Mikrokontroler *NodeMCU ESP32*?

5. Bagaimana *NodeMCU ESP32* mengirim dan mengambil data ke *Websserver*?
6. Bagaimana hasil pengujian sensor ultrasonik sebagai sensor digital pendeteksi volume dari wadah penyimpanan makanan secara *real time*?
7. Bagaimana kerja motor servo pada *Alat Otomatisasi*?
8. Bagaimana cara hasil pengujian lcd i2c sebagai sensor digital pendeteksi volume dari wadah penyimpanan makanan.

1.3. Batasan Masalah

Berdasarkan hasil rumusan masalah yang telah disebutkan oleh penulis, akan dipaparkan dari pembahasan dan pelaksanaan penelitian *Internet Of Things*:

1. Pemanfaatan teknologi *Internet Of Things* membutuhkan jaringan Internet yang stabil.
2. Alat otomatisasi diaplikasikan menggunakan wadah akrilik sederhana yang berukuran tinggi 35cm x panjang 10cm x lebar 12,5cm dengan objek makanan kucing.
3. Alat otomatisasi pemberian makanan jarak jauh menggunakan Mikrokontroler *NodeMCU ESP32*.
4. Alternatif sumber daya alat otomatisasi menggunakan powerbank.
5. Hasil deteksi perhitungan jarak oleh sensor Ultrasonik hanya bisa diproses dari jarak 2cm atau selebihnya.
6. Setting WiFi pada Mikrokontroler *NodeMCU ESP32* secara manual.
7. Protokol *Websocket* untuk akses antara mikrokontroler dan *websserver*.
8. Protokol *RestAPI* untuk akses antara aplikasi *mobile* dan *websserver*.

9. Pembuatan *webservice* menggunakan *Framework Laravel* menggunakan bahasa pemrograman *PHP* dengan konsep protokol *Websocket* dan *RestAPI*.
10. Pembuatan aplikasi *mobile (Smartphone)* menggunakan *Framework UI ReactNative.js* menggunakan bahasa pemrograman *JavaScript* dengan konsep *RestAPI* sebagai media penyimpanan dan pertukaran data.

1.4. Tujuan Penelitian

Tujuan yang dicapai dari penelitian tugas akhir ini adalah alat otomatisasi pemberian makanan melalui jarak jauh yang memanfaatkan teknologi IoT (*Internet Of Things*) dan *robotika* yang dikendalikan oleh aplikasi *mobile (smartphone)* secara online untuk mengurangi dampak kematian pada hewan peliharaan karena terlambat memberikan makanan.

1.5. Manfaat Penelitian

Manfaat yang diharapkan dapat diperoleh dari tugas akhir ini oleh penulis:

1. Bagi Penulis, bermanfaat untuk menambah ilmu pengetahuan yang diperoleh sewaktu pada perkuliahan dan mendapatkan pengetahuan dan pemahaman bagaimana cara kerja Mikrokontroler, aplikasi *mobile (smartphone)*, dan *Internet Of Things*.
2. Bagi Pembaca, untuk menambah wawasan dan pengetahuan terkait *Internet Of Things* dari mikrokontroler, *webservice*, dan aplikasi *mobile (smartphone)* yang menggunakan *ReactNative.js*, dengan menggunakan protokol *Websocket*, dan *RestAPI*.

3. Bagi Pengguna, membantu mengatur jam makan secara *real time* dan meringankan pekerjaan untuk memberi makanan pada hewan peliharaan karena menggunakan sistem otomatisasi.

1.6. Sistematika Penulisan

Pada penyusunan tugas akhir, sistematika pembahasan diatur dan disusun dalam 5 bab, dan tiap-tiap bab terdiri dari sub-sub bab. Untuk memberikan gambaran yang lebih jelas, maka diuraikan secara singkat mengenai materi dari bab-bab dalam penulisan tugas akhir sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini menjelaskan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat dari pembuatan tugas akhir.

BAB II TINJAUAN PUSTAKA

Pada bab ini akan menjelaskan mengenai landasan teori – teori pendukung yang akan menunjang pembuatan tugas akhir diantaranya konsep dan metode yang digunakan.

BAB III METODOLOGI

Pada bab ini akan membahas metode dan analisa perancangan sistem dalam pembuatan tugas akhir. Pada bab ini akan membahas juga terkait metode, algoritma, diagram, sketsa dan analisa perancangan sistem dalam pembuatan tugas akhir rancang bangun otomatisasi pemberian makanan kucing jarak jauh berbasis *Internet Of Things* menggunakan protokol *Websocket* Mikrokontroler *ESP32*.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini akan menjelaskan hasil dan pembahasan tentang kerja dari rancang bangun otomatisasi pemberian makanan kucing jarak jauh berbasis *Internet Of Things* menggunakan protokol *Websocket* Mikrokontroler *ESP32*.

BAB V KESIMPULAN

Bab ini membahas tentang bagian akhir pada sebuah laporan tugas akhir yaitu kesimpulan dan saran yang didapatkan dari hasil evaluasi BAB IV. Kesimpulan akan menjelaskan tentang hasil apa yang telah didapat dari pembuatan alat, aplikasi dan laporannya. Sedangkan saran akan menjelaskan bagaimana peneliti visi tentang sistem yang telah dibuat.

DAFTAR PUSTAKA

Pada bagian ini memaparkan sumber – sumber dan literature yang digunakan dalam pembuatan laporan tugas akhir ini.