BAB 2
TINJAUAN PUSTAKA

2.1 Peternakan

Peternakan merupakan kegiatan mengembangbiakkan maupun memelihara hewan ternak seperti ayam, bebek, sapi, kambing, domba, dan hewan ternak lainnya. Peternakan pada umumnya memiliki kandang sebagai tempat tinggal hewan yang diternakan dan memiliki peternak sebagai pengurus hewan ternak.

2.1.1 Peternakan Ayam

Peternakan ayam merupakan kegiatan mengembangbaikkan maupun memelihara ayam sebagai hewan ternak untuk diambil daging, telur maupun bagaian lain dari ayam tersebut. Ayam merupakan salah satu unggas yang populer untuk diternakkan karena banyak sekali manfaat yang dapat diambil dari ayam dan juga ayam merupakan salah satu hewan ternak yang banyak dikonsumsi oleh manusia pada kebutuhan sehari-hari.

2.1.2 Kandang ayam

Kandang ayam merupakan suatu tempat yang diperuntukan bagi ayam untuk dapat tinggal di dalamnya agar dapat berlindung, makan, minum, dan melakukan aktivitas lainnya didalam kandang dengan nyaman. Kandang ayam disesuaikan dengan kebutuhan peternakan ayam oleh karena itu bentuk kandang ayam petelur dan pedaging akan berbeda dalam segi bentuknya. Pada umumnya bentuk kandang ayam petelur akan diberi sekat antara satu ayam dan diberi jalur uttuk telur ayam dengan ayam lainnya sedangkan ayam pedaging akan dilepas bebas di dalam kandang.

Kondisi lingkungan pada kandang ayam sangatlah berpengaruh dengan pertumbuhan ayam yang ada pada kandang tersebut. Pada suhu 34 ^{o}C, ayam akan mengalami kesulitan dalam membuang panas, terutama jika diikuti dengan kelembaban yang tinggi dalam keadaan demikian ayam tidak dapat lagi membuang panasnya, sehingga suhu tubuh cenderung melambung [4]. terdapat suhu yang ideal pada kandang ketika fase ayam sedang diternakan pada kandang sehingga ayam akan merasa nyaman berada pada kandang

tersebut. Adapun suhu ideal bagi kandang ayam sesuai dengan umur ayam tersebut dapat dilihat pada **Tabel 2.1**.

Tabel 2.1 Suhu Ideal Kandang Ayam

Umur (Hari)	Suhu (°C)
1	32 – 29
3	30 – 27
6	28 – 25
9	27 – 25
>12	26 – 24

Kelembaban relatif agar ayam dapat tumbuh dengan optimal adalah antara 50% sampai 70% [7]. Tingginya kelembaban yang ada pada kandang akan menyebabkan tingginya tingkat amonia dikarenakan pertumbuhan mikroba pada unggas yang pesat pada kondisi lingkungan yang memiliki kelembaban tinggi .

Gambar 2.1 Kandang ayam

Gambar 2.1 merupakan contoh kandang ayam pedaging yang diambil dari peternakan ayam pedaging di daerah Lembang, Kabupaten Bandung, Jawa Barat. Pada kandang ayam tersebut terdapat sekitar 3600 ekor ayam di dalam satu kandang ayam.

2.1.3 Peternak

Peternak merupakan seseorang yang bertugas memelihara hewan ternak beserta kandang dari hewan ternak. Peternak pada umumnya memiliki tugas memberi pakan hewan ternak dan melakukan pengawasan serta pemeliharaan terhadap kandang baik itu dari segi kebersihan, suhu, kelembaban kandang. Peternak juga akan melakukan pengawasan terhadap kesehatan hewan ternak dan pertumbuhan hewan ternak. Pengawasan dan pemeliharaan terhadap hewan ternak mupun kandang akan dilakukan setiap harinya oleh peternak agar setiap aktivitas yang terjadi pada kandang dapat berjalan dengan baik.

2.2 Sistem Internet of Things

Internet of Things, atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang terhubung secara terus menerus [8]. IoT biasanya digunakan sebagai media pengendalian serta pemantauan dari sebuah objek agar didapatkan data-data yang dibutuhkan oleh pengguna untuk dapat disampaikan dalam berupa informasi-informasi seputar objek tersebut secara terus menerus. Data-data ini dapat diperoleh dengan memasang beberapa perangkat tambahan seperti sensor serta aktraktor untuk melakukan aksi-aksi sesuai dengan tujuan pembangunan sistem IoT tersebut.

IoT erat hubungannya dengan pemanfaatkan internet untuk melakukan hal-hal seperti pemantauaan serta pengendalian dari sebuah objek guan kebutuhan yang berbeda-beda. Hal ini akan sangat bergnatung dengan penggunaan internet sebagai media komunikasi data yang terjadi antara perangkat IoT maupun terhadap pemprosesan lain.

2.2.1 Sensor Tetap

Sensor tetap adalah salah satu sensor yang ada pada sistem yang selalu diletakan di dalam kandang. Sensor ini berfungsi sebagai pengambil data

suhu, kelembaban dan kecepatan angin secara menyeluruh di dalam kandang. Sensor

tetap akan mengirim data secara realtime sesuai dengan pengaturan yang telah dimasukan peternak sebelumnya. Sensor tetap menggunakan beberapa komponen elektronika seperti DHT22, ESP8266 dan anemometer.

2.2.2 Sensor Portabel

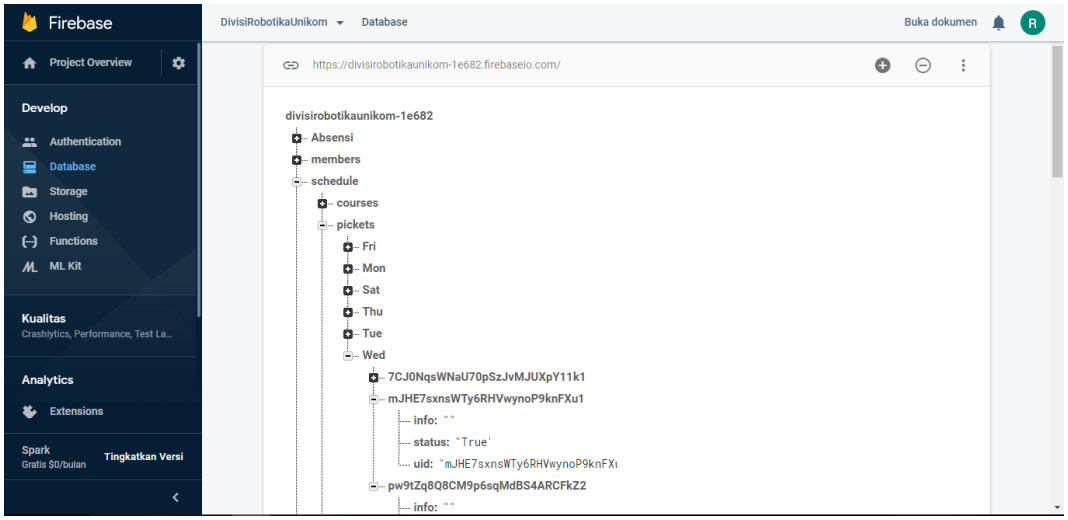
Sensor portabel adalah salah satu sensor yang ada pada sistem yang dapat dibawa menyusuri setiap *grid* yang ada di kandang. Sensor ini berfungsi sebagai pengambil data suhu, kelembaban, tingkat amonia dan 5 *sampel* berat ayam pada setiap *grid* yang ada pada kandang. Sensor portabel menggunakan beberapa komponen elektronika sebagai pembangun sensor portabel diantaranya adalah DHT22, ESP8266, arduino mega, MQ-137, RFID reader 125KHz, timbangan digital.

2.2.3 Pengontrol Blower

Pengontrol blower adalah suatu perangkat IoT yang terdapat pada kandang yang berfungsi sebagai pengontrol utama untuk mengoperasikan blower secara otomatis dengan menyesuaikan dengan suhu yang ada pada kandang. Pengontrol blower menggunakan beberapa komponen elektronika sebagai pembangun pengontrol blower diantaranya adalah ESP8266, relay 8 *channel* dan komponen pendukung lainnya.

2.2.4 Firebase Realtime Database

Firebase realtime database adalah basis data yang di-*host* di cloud dengan menyimpan data JSON dan dilakukan sinkronisasi secara *realtime* ke setiap klien yang tehubung dengan basis data. Firebase realtime basis data akan secara realtime memberikan setiap update kepada seluruh klien walaupun dengan berbagai platfrom yang berbeda seperti SDK Android, iOS, dan javaScript.



Gambar 2.2 Firebase Realtime Basis data

Gambar 2.2 meupakan gambar dari halaman webside Firebase Realtime Basis data, ini dapat diakses pada https://console.firebase.google.com/. Firebase Realtime Basis data memanfaaatkan data yang berformat *document base* yang mana data ini berupa data JSON untuk dapat menyimpan data-data yang dikirimkan ke Firebase Realtime Basis data. *Document oriented database* merupakan sebuah pendekatan penyimpanan data yang memiliki karakteristik informasi berorientasi dokumen tanpa mengunakan skema, SQL, dan ACID. Dengan kemampuannya untuk dapat memberikan pembaruan data secara *realtime* hal ini akan mendukung dalam mengenbangkan sebuah sistem yang membutuhkan pembaruan data secara langsung.

Firebase Realtime Database memiliki beberapa kelebihan yang mana diantaranya adalah bentuk data yang dinamis dan konsep *embedded documnet* sehingga dengan menggunaakn Firebase Realtime Database yang mana meruakan *Document oriented database* sangat baik bagi sebuah sistem yang menerapkan penyimpanan data dengan data yang dinamis. Selain itu Firebase juga memiliki kelemahan yang mana hal ini bila sistem yang dibangun membutuhkan query yang kompleks [9].

Pada *Document oriented database* memiliki beberapa istilah yang digunakan. Instilah-istilah ini memiliki fungsi yang hampir sama dengan

relational database sebagai penanda istilah yang ada didalam basis data tersebut. Adapun perbandingan instilah ini dapat dilihat pada **Tabel 2.2**.

Tabel 2.2 Padanan Istilah basis data relasional dan berbasis dokument

No	Istilah *Relational Databse*	Istilah *Document oriented database*
1	Database	Database
2	Tabel	Collection
3	Field	Field
4	Record	Document

Pada versi gratis Firebase Realtime Database menyediakan kapasitas penyimpanan pada basis data sebesar 1 GB serta jumlah *download* sebesar 10 GB / bulan. Akan tetapi terdapat versi berbayar dengan $5/GB untuk penyimpanan data pada basis data serta $1/GB untuk jumlah *download.*

2.2.5 UDP (*User Datagram Protocol*)

UDP atau *User Datagram Protocol* adalah salah satu protokol lapisan transpor TCP/IP yang mendukung komunikasi yang tidak andal (unreliable), tanpa koneksi (connectionless) antara host-host dalam jaringan yang menggunakan TCP/IP [10]. Menurut (Dwiyankuntoko, 2009) UDP merupakan protokol internet yang mana sangat mengutamakan kecepatan dalam proses pengiriman data yang mana UDP biasanya digunakan untuk kebutuhan data *realtime* karena UDP tidak memerlukan adanya *setup* koneksi terlebih dahulu karena hal tersebut dapat menyebabkan adanya tambahan *delay* [11].

UDP memiliki karakteristik connectionless dimana tidak ada akifitas *handshaking* antara UDP dan penerimanya saat melakukan komunikasi data antar perangkat, sehingga hal ini memungkinkan untuk perangkat mengirimkan data melalui jaringan ke perangkat tujuan tanpa membuat suatu koneksi langsung.

Dengan menggunakan UDP memungkinkan bagi aplikasi maupun software dapat saling mengirim pesan kepada komputer atau perangkat lain di dalam jaringan lain tanpa perlu melakukan komunikasi awal terlebih dahulu. Hal ini dikarenakan UDP menerapkan komunikasi secara sederhana dengan mekanisme yang minimalis. Terdapat proses checksum guna memproteksi integritas data.

Di dalam UDP terdapat salah satu komponen yang disebut dengan port. Port merupakan mekanisme yang memberikan izin kepada sebuah komputer untuk mendukung beberapa saat koneksi dengan computer yang lainnya. Port dapat mengidentifikasi aplikasi dan layanan yang berada di jaringan TCP/IP [12]. Port dapat dikenali dengan angka 16-bit yang dimaksud port number dan diklasifasikan dengan jenis protocol transport yang dimanfaatkan ke dalam port TCP dan port UDP. Karena memiliki batas tolerir jumlah port untuk setiap protocol transport yang digunakan sebanyak 65536 buah. Port UDP adalah channel yang ada di dalam UDP. Dengan menggunakan UDP, setiap aplikasi socket bisa mengirimkan paket-paket yang berbentuk datagram. Datagram selanjutnya diperuntukkan terhadap paket yang bersifat tidak handal (unreliable service). Sementara koneksi handal (biasanya berlaku bagi TCP) selalu menyertakan keterangan jikalau terjadi kegagalan dalam pengiriman data.

UDP (User Data Protocol) memiliki cara kerja yang jelas berbeda dengan TCP. UDP mempunyai saluran atau channel yang berfungsi untuk mengoneksi antar host untuk saling kirim informasi data. Channel itulah yang kemudian disebut dengan port UDP. Agar dapat terhubung dengan protocol UDP, aplikasi sebuah komputer harus dilengkapi alamat IP. Port UDP nantinya berguna untuk multiplexed message queue. Maksudnya, port UDP sangat dimungkinkan mampu bekerja menerima dan mengirimkan pesan data secara bersamaan. Lalu setiap nomor channel UDP atau port UDP dilengkapi dengan nomor unik khusus dalam pembagiannya sendiri [12]. Adapun beberapa tahapan yang ada pada UDP ketika dalam melakukan komunikasi data sebagai berikut.

1. Paket berisi client port dan port sumber berbentuk file text dikirim di dalam UDP Header
2. Paket isi port client dan port sumber audio dikirim ke server dalam UDP Header
3. Tujuan UDP membaca nomor port tujuan dan proses data
4. Paket asli mempunyai port tujuan, sehingga server bisa mengirimkan kembali data ke FTFP Client
5. Point 3 dan 4 berulang kembali ketika file audio dikirimkan client

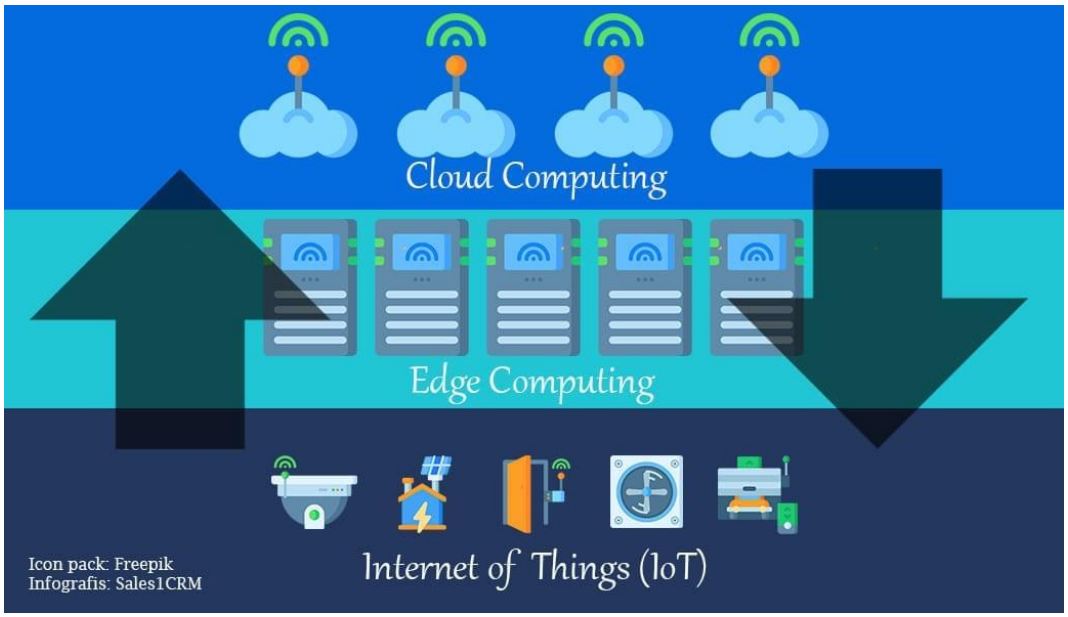
6. Ketika aplikasi ingin mengirim data, UDP tidak akan membuffer ataupun memfragmen tiap data yang masuk
7. Karena data tidak mengalami *prossesing* data yang berbelit terutama karena tidak memfragmen data, maka jika data lebih besar daripada MTU, lapisan IP yang harus memfragmennya.

2.2.6 Edge Computing (komputasi tepi)

Edge computing merupakan suatu konsep pengkomputasian data yang mana terjadi sebelum pada bagian "*edge*" dengan cara data dari seluruh perangkat IoT akan dikumpulkan, disimpan, dan dianalisis secara lokal pada bagian "*edge"* sebelum data tersebut dikirim ke cloud. *Edge computing* memiliki potensi untuk mengatasi permasalahan waktu respon yang terbilang lambat serta bandwitdth yang besar serta *edge computing* juga mempu meningkatkan keamanan dan privasi data yang ada pada sebuah perangkat IoT [13]. *Edge computing* sudah banyak digunakan dalam melakukan efisiensi pada perangkat IoT seperti halnya penerapan pada konsep rumah pintar hingga kota cerdas sekalipun sudah banyak yang menerapkan sistem *edge computing* dikarenakan cocok dalam melakukan efisiensi agar latensi serta badwitdth yang dibutuhkan dapat diminimalisir.

Banyaknya data-data mentah yang dihasilkan oleh perangkat IoT membuat komputasi awan secara konvesional tidak cukup efisiens dalam menangani semua data-data ini dalam artian akan menambah latensi serta bandwidth ketika komunikasi data terjadi. *Edge computing* memungkinkan untuk melakukan komputasi pada bagian tepi dari sebuah sistem IoT yang berada diantara perangkat IoT dan juga cloud. [13]. Pada dasarnya penggunaan *edge computing* adalah menenpatkan komputsai pada bagian yang lebih dekat pada sistem IoT yaitu bagian *edge* sehingga hal ini sangat berpengaruh dalam latensi komunikasi data dibandingkan dengan perangkat IoT yang langsung menggunakan *cloud computing*. Dengan menggunakan *edge compuing* dapat mengurangi sekitar 30% - 40% konsumsi energi yang digunakan serta mengurangi lantensi dari 900 ms menjadi 169 ms [13] sehingga didapatkan 20x lipat lebih efisien dalam proses menjalankan sistem dari segi latensi serta energi yang dibutuhkan.

Edge computing akan melakukan komputasi sedekat mungkin dengan perangkat IoT sehingga dapat meminimalisir latensi serta bandwitdh yang tidak diperlukan ketika melakukan komputasi dari data-data yang ada dibandingkan dengan cara melakukan komutasi pada *cloud* dengan jarak antara perangkat IoT dan *cloud* yang saling berjauhan sehingga menimbulkan nilai latensi serta bandwitdh yang besar dikarenakan lokasi yang saling berjauhan.



Gambar 2.3 Skema Edge Computing

Gambar 2.3 merupakan skema sederhana dalam penggunaan *edge computing* pada sistem IoT. Di mana, proses lalu lintas jaringan dari IoT akan berhenti pada jaringan edge computing. Kemudian, data-data pada edge computing akan kembali di dorong pada sistem cloud. Ketika perangkat (IoT) menerima perintah dari smart device, perintah tersebut akan masuk dan dikirimkan melalui edge. Sehingga latensi serta bandwidth yang dibutuhkan untuk memproses perintah tersebut dapat dilakukan dengan lebih singkat dibandingkan tanpa menggunakan *edge computing*.

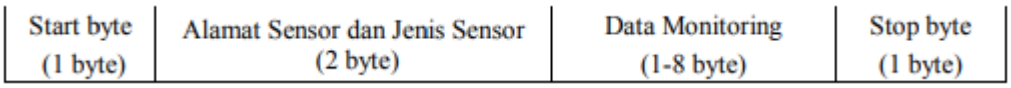
2.2.7 Pemaketan Data

Pemaketan data merupakan sebuah aktifitas yang dilakukan oleh sistem ketika sistem tersebut akan melakukan pengiriman data dari suatu perangkat

ke perangkat lainnya dengan maksud dan tujuan tertentu. Data-data yang ada didalam sebuah perangkat akan dipaketkan terlebih dahulu agar dapat

dilakukan pengiriman yang efisien dari segi tingkat peminimalisisran komunikasi data yang terjadi antar perangkat yang saling berkomunikasi serta juga bertujuan untuk mengurangi kegagalan dalam melaukukan komunikasi data yang terjadi, kegagalan dapat terjadi dikarenakan kesalahan dalam pengiriman data tersebut hingga ketidak sesuaian dengan data yang dikirim dengan data yang diterima oleh perangkat penerimanya.

Dalam melakukan pemaketan data pada umumnya menggunakan *header* tertentu untuk melakukan protokol pemaketan data. Dengna menggunakan *header* dalam pemprosesan pemaketan data, data tersebut dapat dikenali oleh perangkat yang menerima data tersebut dengan baik sehinga tidak terjadi *miss comunication* antara perangkat yang melakukan pengiriman data dengan perangkat yang menerima data tersebut karena keduanya sudah mengetahui *header* yang digunakan oleh protokol pemaketan data tersebut. *Header* yang digunakan dalam melakukan pemaketan data dapat berbeda tergantung dari format *header* yang diinisialisasi oleh masing-masing sistem yang dibangun dengan menyesuaikan kebutuhan masing-masing sistem yang dibangun.



Gambar 2.4 Contoh Format *Header* Pemaketan Data

Gambar 2.4 merupakan contoh dari format *header* protokol pemaketan data dari sebuah sistem IoT yang ada [14]. Dapat dilihat bahwasanya *header* yang dibuat dapat menyesuaikan degan kebutuhan data yang akan dipaketkan serta melihat dengan kebutuhan sistem tentang data paket tersebut. Data yang telah terpaketkan pada bagian perangkat pengirim akan dikirim ke perangkat penerima sehingga perangkat penerima akan melakukan pemecahan terhadap data peket tersebut sehingga akan didapatkan data-data yang sesuai dengan nilai data-data yang terpaketkan tersebut.