# BABI PENDAHULUAN

## 1.1 Latar Belakang

*Internet Of Things* (IoT) menjadi salah satu teknologi yang ramai diperbincangkan saat ini. Kemajuannya yang pesat dan penerapannya yang bisa digunakan di semua bidang menjadikan IoT menjadi salah satu teknologi yang paling berkembang. Menurut hasil studi dari Gartner [1], perusahaan riset dan teknologi dari Amerika Serikat. Pada tahun 2017 ini akan ada 1,5 miliar perangkat baru yang terhubung ke internet. Jumlah tersebut akan meningkat hingga 20 miliar perangkat pada tahun 2020. Perangkat IoT sendiri dapat dibedakan menjadi 3 kategori, yaitu *Wearables*, Perangkat *Smart Home* dan Perangkat M2M *(Machine to Machine)*.

*Wearables* merupakan perangkat yang selalu dibawa oleh pengguna. Biasanya terhubung melalui koneksi *Bluetooth* ke perangkat seluler yang kemudian tersambung ke Internet. Perangkat di kategori ini termasuk jam pintar dan fitness band. Perangkat *Smart Home* merupakan perangkat yang bisa ditemukan di dalam rumah. Perangkat ini meliputi *motion sensor*, pintu dan saklar otomatis hingga oven. Kategori ketiga yaitu M2M merupakan perangkat yang langsung terhubung ke jaringan seperti mobil yang mampu memberitahu lokasinya saat terjadi kecelakaan, atau sebuah kulkas yang mampu memesan sendiri ketika stok buah yang disimpannya habis.

Dengan semakin banyaknya perangkat IoT yang terhubung secara online, pekerjaan manusia tentu akan terbantu. Namun dilain pihak, juga akan menimbulkan masalah baru ketika perangkat-perangkat yang terhubung tersebut memiliki tingkat keamanan yang rendah. Perangkat IoT yang dikuasai hacker dapat diubah menjadi botnet. Botnet ini mampu dikendalikan secara remote oleh hacker untuk melakukan serangan *Distributed Denial Of Service Attacks* (DDOS) ke jaringan tertentu. Pada akhir tahun 2016 kemarin, Dyn sebuah perusahaan provider *Domain Operated System* (DNS) mengalami serangan DDOS pada server mereka. Hal ini mengakibatkan situs-situs besar yang menggunakan layanan DSN Dyn seperti Amazon, Airbnb, CNN, Netflix dan Spotify tidak dapat diakses oleh pengguna. Setelah dilakukan investigasi menyeluruh, ditemukan bahwa serangan tersebut dilakukan lebih dari 100.000 perangkat IoT yang terkena malware Mirai botnet [2].

Berkaca dari hal diatas, sisi keamanan dari IoT perlu ditingkatkan agar kasus tersebut tidak terulang kembali. Baik dari sisi perangkat itu sendiri, komunikasi data ataupun dari sisi *Application Programming Interface* (API). API merupakan penghubung antara bagian *backend* dan *frontend*. Di pengembangan aplikasi berbasis IoT, API menjadi sistem penghubung penghubung antara sensor dengan database, ataupun database dengan antarmuka aplikasi. Penggunaan API diimplementasikan dalam bentuk web service. Generasi pertama web service yang diperkenalkan adalah Simple Object Acces Protocol (SOAP) namun karena perkembangan perangkat IoT yang semakin banyak dan SOAP tidak mampu menghandle perangkat yang berbeda standar maka penggunaan SOAP mulai ditinggalkan.

Representational State Transfer (REST) menjadi pengembangan selanjutnya dari web service. REST web service / RESTful memiliki keunggulan mampu menghandle perangkat-perangkat yang berbeda standar karena menggunakan basis Resource Oriented Architecture (ROA) [3]. REST sendiri memiliki 3 attribute yaitu :

1. Addresssability

Semua resource akan diimplementasikan menggunakan Uniform Resource Identifiers (URI). Setiap resource tersebut akan memiliki alamat URI sendiri. Ketika alamat URI dipanggil dia akan mengembalikan respon dalam bentuk JSON atau XML.

1. Connectedness

Resource yang ada dalam RESTful harus memiliki relasi dengan resource yang lain agar dapat dipresentasikan melalui URI.

1. Homogeneus Interface

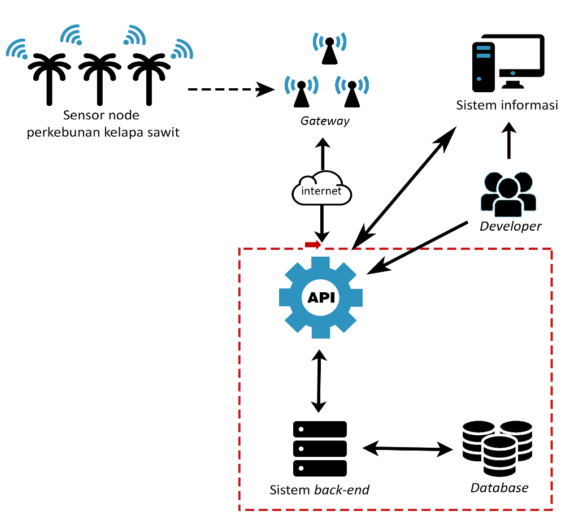
Resource akan dipanggil menggunakan 4 metode HTTP yaitu GET, PUT, POST dan DELETE dengan 2 tambahan metode yaitu HEAD dan OPTION. HEAD digunakan untuk menunjukkan Metadata sedang OPTION digunakan untuk memeriksa metode yang ada

1. Statelessness

Stateles menunjukkan bahwa server tidak menyimpan data dari klien dari setiap koneksi yang terbentuk.

Protokol RESTful ini banyak digunakan sebagai standar komunikasi data pada sistem-sistem IoT. Salah satu sistem IoT yang menggunakan RESTful adalah Smart Farm[4]. Pada Smart Farm, perangkat IoT digunakan sebagai alat monitoring pada perkebunan kelapa sawit yang dapat diakses melalui <http://smartcity.wg.ugm.ac.id/webapp/smartfarm/index.php> . Ilustrasi sistem ini dapat dilihat pada Gambar 1.1. Ada 4 bagian utama didalam sistem ini yaitu : Sensor Node, Gateway, *Back End* dan *Front End*. Sensor Node akan memonitoring kondisi lingkungan dan mengambil data berupa suhu udara, suhu tanah, kelembapan tanah, kelembapan udara, intensitas cahaya matahari, curah hujan, arah angin, kecepatan angin, ketinggian air dan kelembapan daun tanaman kelapa sawit. Data-data tersebut akan dikirimkan secara nirkabel melalui gateway yang akan disimpan kedalam Back End atau Database. Agar data dari database tersebut dapat divisualisasikan dengan baik maka perlu adanya layanan Front End atau Sistem Informasi. Untuk menjembatani antara Back End dan Front End maka diperlukan pengembangan API menggunakan protokol RESTful. API didalam sistem Smart Farm ini memiliki 2 fungsi yaitu sebagai jalur penghubung antara sensor node dengan database dan penghubung antara database dengan Sistem informasi.

Di dalam sistem Smart Farm ini, RESTful sudah dikembangkan secara baik namun masih memiliki beberapa kekurangan karena dalam pengembangan sistemnya aspek keamanan data belum menjadi prioritas. Sebagai contoh, belum adanya autentifikasi user saat mengakses API tersebut, sehingga jika alamat URI dari API tersebut diketahui maka dapat diakses oleh semua orang. Masalah yang lain yaitu setiap user bisa melihat semua data dalam database. Baik itu data dari sensor maupun data yang bersifat private seperti username dan password user. Untuk itu diperlukan suatu layanan manajemen user.



Gambar 1.1 Konsep sistem Smart Farm pemantauan perkebunan Kelapa Sawit

## 1.2 Perumusan masalah

Berdasar latar belakang diatas maka maka dirumuskan masalah sebagai berikut :

1. Data API pada system Smart Farm masih bisa diakses oleh semua user
2. Belum ada pembagian user berdasarkan hak akses
3. Perlunya penambahan metode autentifikasi didalam protokol REST yang digunakan

## 1.3 Keaslian penelitian

Keaslian penelitian ditunjukkan dengan menampilkan beberapa metode penelitian sebelumnya (seperti tinjauan pustaka), kemudian tunjukkan rencana metode penelitian Anda yang membedakan penelitian-penelitian sebelumnya.

## 1.4 Tujuan Penelitian

Penelitian ini bertujuan :

1. Mengembangkan metode autentifikasi pada API Smart Farm
2. Membuat manajemen user dalam sistem Smart Farm

## 1.5 Batasan Masalah

Penelitian ini memiliki batasan masalah sebagai berikut :

1. Difokuskan pada sistem API yang telah ada
2. Fokus pada pengembangan autentifikasi dan layanan hak ases user dalam mengakses URI
3. Hardware dan *sensor node* yang digunakan tidak menjadi perhatian dalam penelitian ini
4. Perancangan sistem informasi dan *user interface* tidak menjadi perhatian dalam penelitian ini

## 1.6 Manfaat Penelitian

Dengan penelitian ini, maka diharapkan akan memberikan manfaat antara lain

1. Sistem smart farm yang telah dibuat akan lebih aman
2. Sistem memiliki pembagian user, dan setiap user memiliki hak akses sendiri
3. Sistem keamanan API yang dibuat dapat dikembangkan lagi untuk penelitian selanjutnya

# BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

## 2.1 Tinjauan Pustaka

Tinjauan pustaka dituliskan berdasar apa yang sudah Anda pelajari dalam rangka penelitian tesis S2. Susunlah tinjauah pustaka dari yang bersifat umum menuju khusus (*general to specific*). Tinjauan pustaka ini dipelajari dari paper-paper seminar maupun jurnal.

## 2.2 Landasan Teori

Landasan teori dituliskan berdasar tinjauan pustaka, sebagai bentuk yang lebih spesifik sesuai dengan arah penelitian Anda. Landasan teori ini didapat dari paper maupun buku, yang mendasari metodologi penelitian yang dibahas di Bab III.

### 2.2.1 Tabel dan Gambar

Contoh dan template untuk tabel terlihat diTabel 2.1.

Tabel 2.1Contoh template untuk tabel

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Huruf | Lafal | Huruf | Lafal | Huruf | Lafal | Huruf | Lafal | Huruf | Lafal |
| A | Ha | n | Na | c | Ca | R | Ra | k | Ka |
| F | Da | t | Ta | s | Sa | W | Wa | l | La |
| P | Pa | d | Dha | j | Ja | Y | Ya | v | Nya |
| M | Ma | g | Ga | b | Ba | Q | Tha | z | Nga |

Tabel 2.1 adalah contoh *template* untuk tabel, dengan memanfaatkan *Caption* dan *Cross Reference*. Tabel jangan sampai terpotong ke halaman lain. Jika terpaksa terpotong maka harus dibuat judul tabel baru pada halaman berikutnya yang merupakan lanjutan halaman sebelumnya. Contoh *template* untuk gambar terlihat di Gambar 2.1.



Gambar 2.1. Kuliner - Yimmi Tomyam

Gambar 2.1 menunjukkan contoh *template* gambar. Hati-hati dalam menggunakan *cross reference*. Jika posisinya diubah maka harus update, atau tulisan nanti akan berantakan.

Gambar grafik dimungkinkan berwarna. Semua warna akan dipertahankan pada CDROM. Grafik jangan menggunakan pola titik-titik karena ada kemungkinan tidak dapat dicetak sesuai aslinya. Gunakan SOLID FILL dan warna yang kontras untuk tampilan di layar komputer, dan gunakan warna hitam-putih untuk hardcopy.

### 2.2.2 Persamaan dan Bagian *Listing* Program

Persamaan dapat menggunakan fungsi persamaan atau ditulis dalam text biasa. Ketentuannya seperti dinyatakan dengan Persamaan (2-0).

|  |  |
| --- | --- |
|  | (2-0) |

dengan

 = nilai piksel keluaran  
 = nilai piksel masukan  
 = nilai ambang

Dalam membuat persamaan seperti Persamaan (2-0) terpaksa dibuat dalam tabel 2 kolom tanpa *border*. Hal ini dimaksudkan agar ketika dilakukan Cross Reference tidak akan muncul persamaannya. Disamping itu ketika muncul nomor persamaan harus ditambahi sendiri dengan tanda kurung tutup.

Kadang-kadang diperlukan bagian program untuk diterangkan di dalam tesis. Dengan demikian perlu penulisan bagian program tersebut, yang disebut dengan *listing* program. Penampilan*listing* program ditaruh di Lampiran.

## 2.3 Hipotesis

Hipotesis bersifat dugaan hasil tentang penelitian yang Anda lakukan.

## 2.4 Keterangan Lain-lain (tidak dimasukkan dalam format penulisan)

Dokumen ini adalah template untuk penulisan tesis. Sebuah salinan elektronik yang dapat di-download dari situs web program studi S2 jurusan Teknik Elektro UGM. Jika ada hal-hal yang belum jelas, silakan hubungi pengelola program studi atau dosen pembimbing[1].

Informasi tentang makalah akhir penyerahan tersedia dari situs web S2 jurusan Teknik Elektro UGM atau Kantor pengelola program studi[2].