

BAB 1

ANALISIS KEBUTUHAN

1.1 Latar Belakang Masalah

Persinyalan kereta api merupakan suatu bentuk, warna atau cahaya untuk memberikan isyarat kepada petugas perkeretaapian agar perjalanan kereta api aman [1]. Jenis persinyalan kereta api terbagi menjadi persinyalan elektrik dan mekanik. Persinyalan Elektrik adalah suatu peralatan untuk memberikan isyarat berupa cahaya atau warna yang memiliki arti tertentu dan biasanya berupa lampu LED (*Light Emitting Diode*) [2]. Sinyal mekanik adalah perangkat sinyal yang digerakkan secara mekanik, contohnya papan atau lengan instruksi yang dinaikkan dan diturunkan untuk memberi perintah kepada masinis kereta api [3].

Kondisi persinyalan kereta api dapat dipantau melalui CTC yaitu sebuah sistem *monitoring* dan kontrol persinyalan kereta api yang berasal dari Amerika Utara. CTC juga dapat digunakan untuk memantau kondisi dari persinyalan kereta api apakah berjalan dengan keadaan normal atau sedang terjadi gangguan yang dipantau oleh operator dan teknisi *maintenance* di ruang kontrol [4].

Dalam pemantauan persinyalan kereta api di Indonesia operator harus berada di ruang kontrol dalam melakukan pemantauan kondisi. Untuk membagikan tampilan CTC saat ini hanya menggunakan sistem *Local Area Network* (LAN) yang memiliki batas jarak sehingga pemantauan menjadi tidak efisien dan memerlukan waktu yang cukup lama untuk melakukan perbaikan jika terjadi kegagalan pada kondisi persinyalan [5].

Untuk mengatasi ketidakefisienan dalam pemantauan dan masalah waktu yang cukup lama dalam melakukan perbaikan jika terjadi kegagalan pada kondisi persinyalan, kami memberikan solusi dengan memanfaatkan teknologi *Internet of Things* (IoT) dimana persinyalan yang ditampilkan di SCADA datanya dikirimkan ke *cloud* lalu dikirimkan ke *smartphone* sehingga data kondisi persinyalan dapat dipantau operator dan teknisi *maintenance* tanpa harus berada di *control room* dan dapat meningkatkan kecepatan respon teknisi *maintenance* pada perbaikan dan pemeliharaan instrumen pada persinyalan kereta api [6].

1.2 Informasi Pendukung



Gambar 1.1 Arsitektur IoT

Arsitektur IoT terdiri dari berbagai sistem penyusun yang bisa dilihat pada Gambar 1.1 yaitu *things*, *gateway*, *cloud*, *analytics*, dan *user interface*. *Things* adalah objek yang dilengkapi sensor untuk mengumpulkan data dan dikirim ke jaringan internet atau aktuator untuk bertindak. *Gateway* merupakan fasilitas konektivitas yang menghubungkan jaringan satu dengan jaringan yang lain. *Cloud* adalah teknologi berbasis internet yang dapat menyimpan, mengakses, dan mengolah data. *Analytics* adalah proses menganalisis data besar untuk mengetahui kinerja perangkat. Pada bagian *User Interface* terdapat komponen lunak dari sistem IoT yang menghubungkan pengguna dengan sistem IoT untuk memantau dan mengontrol sistem melalui tampilan di aplikasi [7].

Protokol komunikasi pada IoT merupakan suatu aturan yang mengatur pertukaran informasi dan menjembatani komunikasi antara perangkat IoT, protokol ini berfungsi sebagai bahasa untuk perangkat saling memahami dan berinteraksi. Macam-macam protokol komunikasi yaitu MQTT (*Message Queuing Telemetry Transport*), CoAP (*Constrained Application Protocol*), HTTP (*Hypertext Transfer Protocol*).

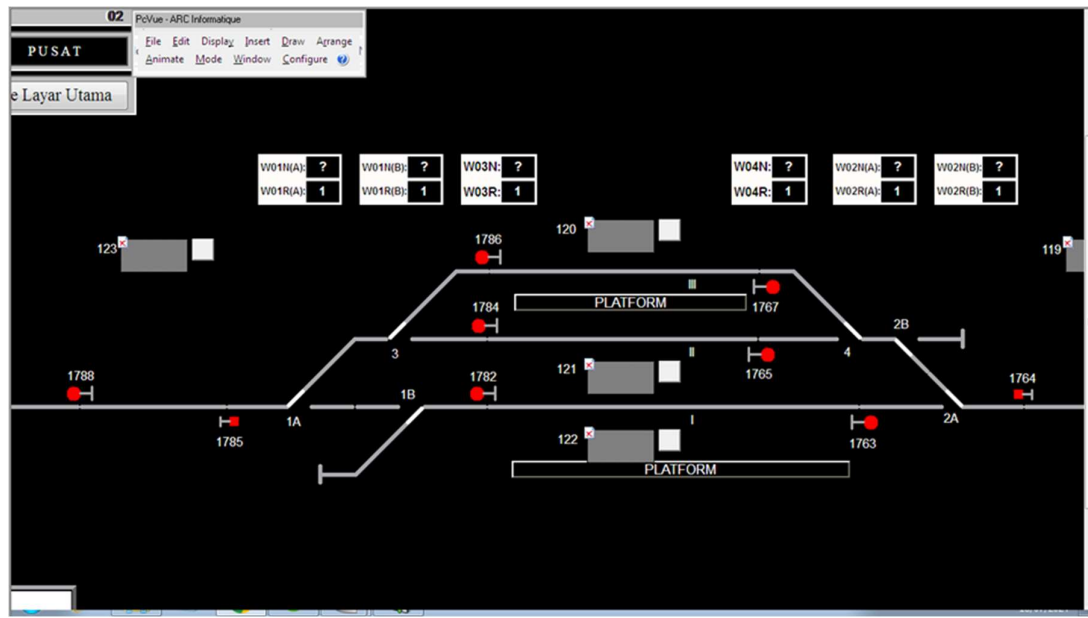
MQTT merupakan protokol komunikasi *publish/subscribe*. Kelebihannya yaitu pengurangan konsumsi energi dan *overhead*, cocok untuk perangkat dengan sumber daya terbatas. kekurangannya yaitu kurangnya *built-in* keamanan tanpa implementasi tambahan [8].



Gambar 1.2 PcVue

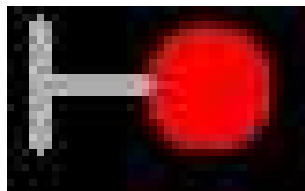
PcVue adalah *platform* SCADA untuk mengontrol, mengawasi proses industri. PcVue memiliki fitur untuk pengumpulan data, pemantauan, pengendalian, dan analisis dari perangkat lapangan seperti sensor, aktuator, PLC (*Programmable Logic Controllers*), dan RTU (*Remote Terminal Units*). PcVue dapat dimanfaatkan sebagai *software* SCADA untuk mengawasi persinyalan kereta api pada CTC melalui komputer dan data persinyalan juga dapat dikirimkan ke Microsoft SQL Server Management Studio untuk dianalisis dan disimpan [9].

Arsitektur pada PcVue terdiri atas *Field Device Layer*, *Communication Layer*, *Server Layer*, *Application Layer*. Pada *Field Device Layer* terdapat sensor, aktuator, PLC, RTU untuk mengumpulkan data di lapangan kemudian layer ini dibantu oleh *Communication Layer* untuk mengirimkan datanya ke PcVue sebagai Server SCADA yang terdapat di *Server Layer*. PcVue berfungsi sebagai server SCADA untuk mengumpulkan, memproses, mengontrol, dan menyimpan data. Pada *Application Layer* berisi *Human Machine Interface* (HMI) sebagai grafis antarmuka bagi operator untuk memonitor, mengontrol, dan menganalisis.



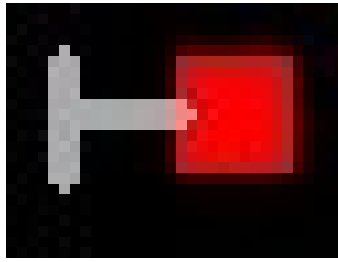
Gambar 1.3 CTC pada gedung N

Gambar 1.3 merupakan tampilan simulasi CTC yang berada pada gedung N, CTC yang ditampilkan berupa 5 stasiun yaitu stasiun FTE, pada *Capstone Design* kami menggunakan simulasi pada stasiun FTE.



Gambar 1.4 Sinyal STR

Pada Tampilan PcVue di CTC stasiun FTE terdapat sinyal bulat yang pada pengimplementasiannya bernama sinyal *starter* yang berfungsi sebagai utk tanda kereta dari stasiun ke luar. Pada PcVue sinyal ini memiliki 2 kondisi yaitu kondisi normal dengan indikasi berwarna merah dan kondisi gangguan dengan indikasi berwarna hijau.



Gambar 1.5 Sinyal kotak

Pada Tampilan PcVue di CTC stasiun FTE terdapat sinyal kotak yang pada pengimplementasiannya bernama sinyal *shunt* yang berfungsi sebagai tanda langsir kereta untuk menyusun, menambah, menghapus dan memindahkan rangkaian kereta. Sinyal ini memiliki 2 kondisi yaitu kondisi normal dengan indikasi berwarna merah dan kondisi gangguan dengan indikasi berwarna hijau.



Gambar 1.6 Rel

Gambar 1.6 merupakan gambar rel dimana rel ini memiliki 2 warna sebagai indikasi pada kondisinya, warna abu-abu sebagai indikasi bahwa rel dalam kondisi normal sedangkan warna kuning sebagai indikasi bahwa rel dalam kondisi gangguan.

1.3 *Constraint*

Berdasarkan informasi pendukung yang ada, maka ada beberapa aspek yang diperhatikan untuk membatasi penyelesaian masalah.

1.3.1 Aspek Pengguna

Petugas yang diberi kewenangan untuk memegang sistem *monitoring* kereta api adalah individu atau personel yang memiliki tanggung jawab dalam mengoperasikan dan mengelola sistem tersebut. Sistem ini membatasi para pengguna, yang diberi kewenanganlah yang dapat mengakses sistem *monitoring* CTC seperti teknisi *maintenance* dan operator.

1.3.2 Aspek Keselamatan

Fungsi-fungsi seperti pemantauan kondisi persinyalan merupakan salah satu fitur dalam sistem CTC yang membantu mengoptimalkan operasi dan meningkatkan keselamatan. Namun, penting untuk mencatat bahwa sistem CTC harus selalu diimplementasikan dan dioperasikan dengan mempertimbangkan keselamatan sebagai prioritas utama untuk memastikan perlindungan penumpang, kru, teknisi dan masyarakat sekitar, sehingga durasi perubahan data terbaru tertampil ke *smartphone* maksimal 5 menit.

1.3.3 Aspek Fungsional

SCADA CTC digunakan hanya untuk mengawasi kondisi persinyalan. Operator dan teknisi *maintenance* dalam mengawasi kondisi persinyalan hanya dapat mengawasi pada 1 stasiun. Pengawasan dapat dilakukan oleh operator dan teknisi *maintenance* yang bersangkutan melalui PcVue dan melalui *smartphone*.

1.4 Kebutuhan yang Harus Dipenuhi

Berdasarkan latar belakang masalah, analisa yang telah dilakukan dan *interview* dengan *user* maka yang harus dipenuhi adalah sebagai berikut:

1. *Software* untuk melihat HMI melalui *smartphone*
2. Sistem proteksi untuk membatasi mengakses sistem ini
3. Konektivitas antar *platform*
4. *Data transfer*
5. Durasi pengiriman data yang optimal

1.5 Tujuan

Tujuan dari penelitian *Capstone Design* ini adalah untuk menjawab kebutuhan akan sistem pemantauan yang modern, aman, dan efisien dalam *monitoring* CTC kereta api dengan merancang sebuah perangkat lunak berbasis IoT yang dapat memantau kondisi persinyalan kereta api melalui aplikasi di *smartphone*. Aplikasi ini akan dilengkapi dengan sistem proteksi yang membatasi akses pengguna, sehingga hanya pengguna yang

berwenang seperti operator dan teknisi *maintenance* yang dapat mengakses data dan fungsi pengawasan.