

Pemanfaatan Seamless Wireless (EoIP) dan GPS pada Sistem Peringatan Perlintasan Kereta Tanpa Palang Pintu

Dio Pramantha¹, Gani Indriyanta², Laurentius Kuncoro Probo Saputra³

Informatika, Universitas Kristen Duta Wacana
Jl. Dr. Wahidin Sudirahusada No. 5-25, Yogyakarta

¹dio.pramantha@ti.ukdw.ac.id

²ganind@staff.ukdw.ac.id

³kuncoro@staff.ukdw.ac.id

Abstract-- The emergence of illegal railway crossings that have not barrier gate and guard is the beginning of frequent accidents at railway crossings. The solution to this problem is by creating a system that can detect train arrival and is able to warn crossers as well to be more vigilant and cautious around the barrierless railway crossing to reduce accident rate that occurred by human negligence. The development of this automatic train arrival warning system with barrier gate is using seamless wireless Ethernet over Internet Protocol (EoIP) technology. The process of this research was carried out by testing a prototype in the UKDW campus area with the length of the crossing is around 70-80 meters, using 1 MCU (Micro Controller Unit) ESP8266, 1 GPS Module (train node), 2 MCU, 2 LCD modules, 2 Servo motor and 2 Buzzers become 2 railway crossing nodes. This system was tested 30 times with a test checklist. The test starts when the train node sends information in the form of longitude and latitude to the server. The server will calculate the distance between the train nodes with each crossing node and send the results to each train crossing node. The distance checking will occur at each train crossing node and at the specified distance the barrier will be closed or opened. From 30 trials it was concluded that the train arrival warning system with automatic barrier gate can detect train arrivals and has been successfully built properly and runs according to the designed flow system. It is equipped EoIP seamless wireless technology that enables network construction even in blind spots which can not be reached by cellular or GSM operators. Therefore this seamless wireless EoIP technology is very suitable in assisting the construction of train arrival warning system with automatic barrier gate as evidenced by each device being able to be connected and uninterrupted during 30 trials.

Keywords—train system, automatic system, seamless EoIP, nodes, railway crossings

Intisari— Perlintasan kereta liar tanpa palang pintu dan tanpa penjaga merupakan awal mula sering terjadinya kecelakaan di perlintasan kereta api. Salah satu solusi untuk menyelesaikan permasalahan ini dengan pembangunan sistem otomatis peringatan kedatangan kereta dengan memanfaatkan teknologi *seamless wireless Ethernet over Internet Protocol (EoIP)*. Penelitian ini dilakukan dengan pengujian prototype di lingkungan UKDW dengan panjang pelintasan kurang lebih 70 - 80 meter, menggunakan 1 MCU(Micro Controller Unit) ESP8266 dan 1 GPS Module yaitu node kereta, 2 MCU, 2 unit LCD module, 2 Servo motor dan 2 Buzzer menjadi 2 node perlintasan kereta. Sistem ini diuji coba sebanyak 30 kali dengan checklist pengujian. Pengujian dimulai saat node kereta

mengirimkan informasi dalam bentuk longitude dan latitude ke server lalu server akan menghitung jarak antara node kereta dengan setiap node perlintasan kemudian hasil jarak tersebut akan di kirim ke setiap node perlintasan kereta, akan terjadi pengecekan jarak di setiap node perlintasan kereta, pada jarak yang telah ditentukan palang akan ditutup ataupun dibuka. Dari 30 kali percobaan disimpulkan bahwa sistem otomatis peringatan kedatangan kereta dapat mendeteksi kedatangan kereta dan telah berhasil dibangun dengan benar dan berjalan sesuai flow sistem yang sudah dirancang. Dilengkapi dengan pemanfaatan Teknologi seamless wireless EoIP yang memungkinkan pembangunan jaringan bahkan di titik buta tidak dapat dijangkau oleh operator seluler ataupun GSM. Oleh karena itu seamless wireless EoIP ini sangat cocok dalam membantu pembangunan sistem peringatan kedatangan kereta api dengan palang otomatis dibuktikan dengan setiap alat dapat terhubung dan tidak terputus selama 30 kali percobaan.

Kata Kunci—sistem kereta, sistem otomatis, seamless EoIP, nodes, perlintasan kereta

I. PENDAHULUAN

Kereta api merupakan suatu sarana transportasi yang banyak digunakan oleh masyarakat Indonesia untuk bepergian dari satu tempat ke tempat lainnya terlebih khusus di pulau Jawa, karena kereta api memiliki rute sendiri sehingga dapat menghemat waktu bagi orang-orang yang mempunyai mobilitas yang cukup tinggi. Kebutuhan akan fungsi kereta ini juga harus diikuti dengan tingkat keselamatan para penyeberang jalan, baik kendaraan roda dua, roda empat maupun pejalan kaki yang akan melintasi rel kereta.

Perlntasan kereta api tanpa palang pintu dan tanpa penjaga merupakan awal mula sering terjadinya kecelakaan di perlntasan kereta api. Berdasarkan latar belakang permasalahan mengenai kurangnya waspadanya orang-orang yang akan melewati perlntasan kereta api. Kecelakaan kereta api kerap terjadi, seperti yang terjadi di 08 Januari 2018 hari Senin pada sekitar pukul 15.30 WIB kereta api Bengawan 143 jurusan Solo-Jakarta yang melintas di perlntasan tanpa palang pintu menabrak seorang wanita yang mengendarai sepeda motor, korban terseret sekitar 10 meter dan meninggal dunia. Kejadian terjadi tepatnya di Desa Titang, Kecamatan Jogonalan km 146,5 antara Stasiun Srowot-Prambanan, Klaten [1] maka dibangunlah sebuah sistem yang dapat membuat para pelintas lebih waspada dan

berhati-hati di perlintasan kereta api tanpa palang pintu untuk mengurangi kecelakaan yang kerap terjadi karena kelalaian manusia. Sebuah sistem otomatis yang dapat mendeteksi kedatangan kereta dan dapat memberi peringatan ketika kereta sudah sangat dekat dengan memanfaatkan teknologi seamless wireless Ethernet over Internet Protocol (EoIP).

Sistem ini implementasikan di UKDW dalam bentuk *prototype*. Penelitian ini diharapkan memberikan manfaat untuk memberikan solusi kepada PT. KAI dalam mengembangkan jalur komunikasi yang awalnya jalur telepon menjadi jalur wireless dan dalam mengontrol setiap titik perlintasan kereta yang tidak dapat dicapai oleh penjaga (karena sistem bekerja otomatis) serta meningkatkan kewaspadaan dalam mencegah bahkan meminimalisir angka kecelakaan.

II. TINJAUAN PUSTAKA & LANDASAN TEORI

A. Tinjauan Pustaka

Pada penelitian terdahulu yang dilakukan oleh Firdaus [2] dengan judul Miniatur Palang Pintu Kereta Api Otomatis Dengan Menampilkan Kecepatan Kereta Serta Waktu Tunggu Menggunakan Arduino. Perancangan menggunakan *arduino*, *gps module*, *servo*, *LCD 1602* dan *rangkain buzzer*. Sensor digunakan untuk mendeteksi posisi kereta, pada miniatur yang telah di buat menggunakan 4 sensor srf-04. Fungsi tiap sensor adalah pendeteksi kedatangan kereta, mengaktifkan sistem penghitung kecepatan dan pendeteksi bahwa kereta api telah melewati palang pintu. Palang pintu pada miniatur digerakkan oleh motor DC servo. LCD pada miniatur mampu menampilkan kecepatan kereta serta waktu tunggu kedatangan kereta. Pada miniatur yang telah di buat juga dilengkapi dengan kontrol manual, sehingga saat terjadi kegagalan sistem otomatis palang pintu dapat di operasikan secara manual. Diharapkan untuk penelitian lebih lanjut menggunakan arduino seri mega karena jumlah pin input dan output lebih banyak sehingga dapat di tambahkan sensor lain agar pendeteksian posisi kereta lebih akurat dan dapat dijalankan pada jenis lintasan rel ganda.

Penelitian lain yang dilakukan oleh Cahyadi [3] dengan judul Pemanfaatan Fitur Tunneling Menggunakan Virtual Interface EoIP di MikrotikRouterOS Untuk Koneksi Bridging Antar Kantor Melalui Jaringan ADSL Telkom Speedy. Fitur tunneling dengan EoIP dapat dimanfaatkan untuk membuat bridging antar perangkat MikrotikRouterOS untuk menghubungkan dua atau lebih kantor suatu organisasi/lembaga melalui jaringan ADSL Speedy Telkom (public network/internet), sehingga seolah-olah kantor-kantor tersebut terhubung dalam satu segmen jaringan intranet. Dalam aspek keamanan, walau EoIP tidak memberlakukan enkripsi seperti VPN-IP, namun administrator dapat mengaktifkan fungsi firewall/filtering dan monitoring pada interface-interface EoIP. Dari sisi biaya, dengan menggunakan jaringan ADSL Speedy Telkom, dapat dibangun sebuah jaringan private (antar perangkat MikrotikRouterOS) di atas jaringan public yang biayanya relatif lebih murah dibandingkan dengan sewa layanan VPN-IP. Saran dari peneliti Jangan menggunakan fitur Bridge Loop sebagai fail over system (melalui protokol STP/RSTP), karena fail over tersebut tidak akan berguna apabila menggunakan ISP yang sama, dalam artian jika 1 link

down maka link lainnya yang dalam satu ISP bisa dipastikan juga down.

Briantoro, Arifin, & Hendriawan [4] melakukan penelitian dengan judul Sistem Informasi Posisi Kereta Api Menggunakan GPS. Pada setiap pengujian pengukuran besar lintang dan bujur pada masing-masing stasiun dan shelter tidak selalu sama. Pada saat pengiriman SMS data lintang dan bujur stasiun/shelter ke server pada kereta api selalu ada waktu tunda. Rata-rata waktu tundanya ialah 6.624 detik. Terdapat selisih jarak antara stasiun/shelter sebenarnya dengan stasiun/shelter yang ditunjukkan oleh alat Sistem Informasi Posisi Kereta Api Menggunakan GPS. Pengiriman SMS. Waktu tunda (detik) Alat Sistem Informasi Posisi Kereta Api Menggunakan GPS menampilkan nama stasiun/shelter pada LCD dengan tingkat keberhasilan 100% dan mendeteksi perlintasan kereta api dengan tingkat keberhasilan 100%. Alat mendeteksi perlintasan kereta api rata-rata berjarak 157.05 meter sebelum melewati perlintasan kereta api. Selisih waktu estimasi dengan posisi kereta api sebenarnya rata-rata sebesar 2.229 detik. Alat memiliki tingkat keakurasian 98,17% dalam memberikan informasi posisi. Untuk pengembangan selanjutnya, perlu penggunaan GPS yang memiliki tingkat akurasi lebih tinggi.

Penelitian yang peneliti lakukan berfokus kepada pembuatan sistem peingatan kedatangan kereta otomatis untuk meningkatkan kewaspadaan dalam melewati perlintasan kereta api dengan memanfaatkan seamless wireless EoIP akan diteliti dengan cara menganalisis waktu yang diperlukan untuk pengiriman informasi dari aplikasi web yang akan dibangun peneliti kepada sistem peringatan kedatangan kereta otomatis dan bagaimana sistem akan dibangun dan dirancangan. Peneliti berharap sistem dapat mencegah, meningkat kewaspadaan kepada orang-orang bahkan pengemudi yang akan melintasi perlintasan kereta tanpa palang pintu, dan mengurangi angka kecelakaan dan kematian sia-sia oleh karena kurangnya kewaspadaan.

B. Landasan Teori

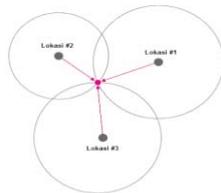
1) Global Positioning System (GPS)

Global Positioning System (GPS) merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunaanya dimana dia berada (secara global) dipermukaan bumi yang berbasis satelit. Data dikirim dari satelit berupa sinyal radio dengan data digital.

GPS (*Global Positioning System*) adalah sistem navigasi yang berbasis satelit yang saling berhubungan yang berada di orbitnya. Satelit-satelit itu milik Departemen Pertahanan (*Departemen of Defense*) Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978 dan pada tahun 1994 sudah memakai 24 satelit. Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama GPS receiver yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. Posisi diubah menjadi titik yang dikenal dengan sebutan Way-point nantinya akan berupa titik-titik koordinat lintang dan bujur dari posisi seseorang atau suatu lokasi kemudian di layar pada peta elektronik. GPS adalah satu-satunya sistem satelit navigasi global untuk penentuan lokasi, kecepatan, arah, dan waktu yang telah beroperasi secara penuh didunia saat ini (*undergraduate thesis Wildan Habibi, ITS, Surabaya Januari : 2011*).

GPS menggunakan konstelasi 27 buah satelit yang mengorbit bumi, dimana sebuah GPS receiver menerima

informasi dari tiga atau lebih satelit tersebut seperti terlihat dalam Gambar 1 dibawah, untuk menentukan posisi. GPS receiver harus berada dalam line-of sight (LoS) terhadap ketiga satelit tersebut untuk menentukan posisi, sehingga GPS hanya ideal untuk digunakan dalam outdoor positioning [5].

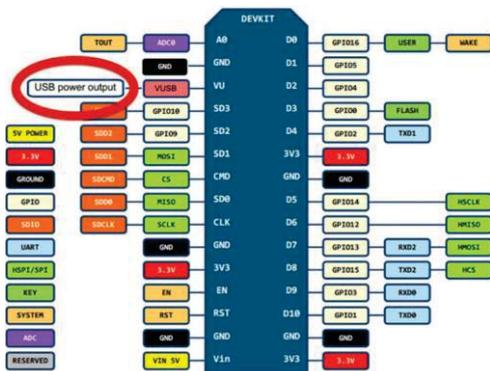


Gambar. 1. Trilaterasi Dalam Global Positioning System (GPS)

2) NodeMCU ESP8266

NodeMCU merupakan sebuah open source platform IoT dan pengembangan kit yang menggunakan bahasa pemrograman Lua untuk membantu dalam membuat prototype produk IoT atau bisa dengan memakai sketch dengan adruino IDE. Pengembangan kit ini didasarkan pada modul ESP8266, yang mengintegrasikan GPIO, PWM (Pulse Width Modulation), IIC, 1-Wire dan ADC (Analog to Digital Converter) semua dalam satu board. GPIO NodeMCU ESP8266 seperti Gambar 2

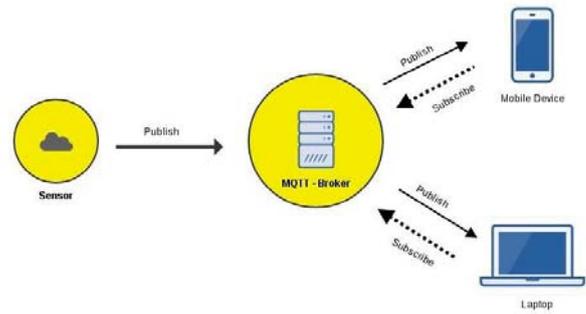
NodeMCU berukuran panjang 4.83cm, lebar 2.54cm, dan berat 7 gram. Board ini sudah dilengkapi dengan fitur WiFi dan Firmwarena yang bersifat opensource [6].



Gambar. 2. GPIO NodeMCU ESP8266 v3

3) MQTT (Message Queuing Telemetry Transport)

Salah satu model komunikasi yang digunakan dalam sistem Internet of Things adalah publish/subscribe (PUB/SUB). MQTT menggunakan topik dalam mem-publish ataupun men-subscribe pada pengiriman pesan. MQTT (Message Queuing Telemetry Transport) adalah protokol yang berjalan pada diatas stack TCP/IP dan mempunyai ukuran paket data dengan low overhead yang kecil (minimum 2 bytes) sehingga berefek pada konsumsi daya yang juga cukup kecil. Protokol ini adalah jenis protokol data-agnostic yang artinya anda bisa mengirimkan data apapun seperti data binary, text bahkan XML ataupun JSON dan protokol ini memakai model publish/subscribe daripada model client-server [7].



Gambar. 3. Sistem Umum MQTT

4) WDS (Wireless Distribution System)

WDS (Wireless Distribution System) merupakan sebuah sistem yang memungkinkan interkoneksi antar access point. Sistem ini digunakan untuk memperluas jangkauan area wireless, dengan menggunakan beberapa perangkat access point yang menjadi satu kesatuan, tanpa membangun backbone jaringan. Syarat dalam membangun jaringan WDS yaitu access point harus menggunakan Band, Frequency, dan SSID yang sama.

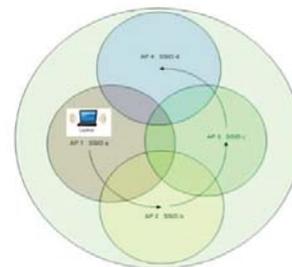
Mode Wireless Distribution System (WDS) yaitu :

- WDS Dynamic adalah Interface WDS secara otomatis akan terbuat ketika sudah menentukan perangkat AP lain yang kompetibel.
- WDS Static adalah mode WDS yang Interface WDS dibuat secara manual.
- WDS Dynamic-Mesh adalah mode WDS yang sama seperti WDS Dynamic, namun hanya menggunakan protocol HWMP+ yaitu pengembangan dari WDS standar.
- WDS Static-Mesh adalah mode WDS yang sama dengan mode Static, hanya menggunakan protocol HWMP+.

□ Perbedaan sistem Non-WDS dengan WDS

a. Wireless non-WDS

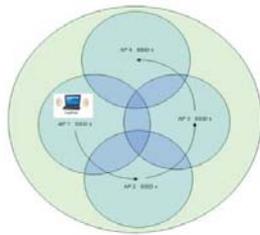
Apabila user berpindah tempat dari area AP1 ke area AP lain (AP2/AP3), maka user tersebut akan mengalami kehilangan koneksi beberapa saat sebelum terhubung ke area AP yang baru.



Gambar. 4. Wireless non-WDS

b. Wireless WDS

Apabila user berpindah tempat dari area AP1 ke area AP lain (AP2/AP3), maka user seakan-akan tetap berada di area yang sama tanpa perlu kehilangan koneksi [8].



Gambar. 5. Wireless WDS



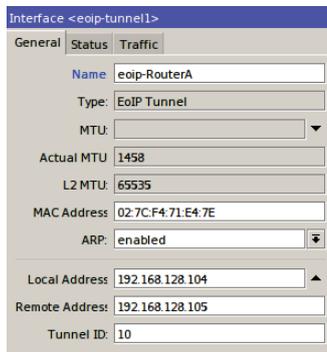
Gambar. 7. Modul GPS (NEO-6M)

(<https://store.roboticsbd.com/plc-programming-cable/797-u-blox-neo-6m-gps-module-robotics-bangladesh.html>)

5) *Ethernet over Internet Protocol (EoIP) Mikrotik*

Ethernet over IP (EoIP) Tunneling adalah protokol MikroTik RouterOS yang menciptakan terowongan Ethernet antara dua router di atas koneksi IP. Terowongan EoIP dapat berjalan di atas terowongan IP, terowongan PPTP atau koneksi lain yang mampu mengangkut IP. Ketika fungsi bridging dari router diaktifkan, semua lalu lintas *Ethernet* (semua *protokol Ethernet*) akan dijembatani seolah-olah ada di mana antarmuka *Ethernet* fisik dan kabel antara dua *router* (dengan *bridging* diaktifkan). Protokol ini memungkinkan skema jaringan ganda. Pengaturan jaringan dengan antarmuka EoIP:

- memungkinkan untuk menjembatani LAN melalui Internet
 - memungkinkan untuk menjembatani LAN melalui terowongan yang terenkripsi
 - memungkinkan untuk menjembatani LAN melalui jaringan nirkabel
- 802.11b *ad-hoc Protokol EoIP* mengenkapsulasi bingkai *Ethernet* dalam paket *GRE (IP protocol number 47)* (seperti PPTP) dan mengirimnya ke sisi remote dari terowongan *EoIP* [9].



Gambar. 6. Contoh Konfigurasi EoIP Router gateway

6) *Modul GPS (NEO-6M)*

Modul GPS dengan jenis NEO-6M berukuran ringkas ini (25x35mm untuk modul, 25x25mm untuk antena) berfungsi sebagai penerima GPS yang dapat mendeteksi lokasi dengan menangkap dan memproses sinyal dari satelit navigasi. Aplikasi dari modul ini melingkupi sistem navigasi, sistem keamanan terhadap kemalingan pada kendaraan/berpindahnya kendaraan.. Sumber tenaga dapat menggunakan catu daya antara 3 Volt hingga 5 Volt [10].

7) *Perhitungan Jarak antara 2 Lokasi*

$$\cos d = \sin \varphi_1 \sin \varphi_2 + \cos \varphi_1 \cos \varphi_2 \cos(L1 - L2) \quad (1)$$

$$s = \frac{6371 \pi d}{180} \text{ kilometers} \quad (2)$$

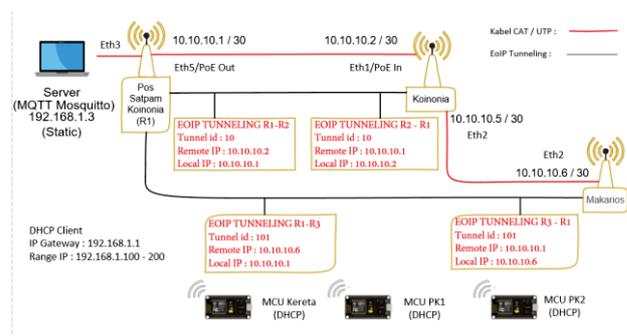
Di dalam buku Jean Meus [11], ada 2 rumus yang dapat digunakan dalam menghitung jarak antara 2 lokasi yang berbeda yang pertama diasumsikan bumi itu bulat dan yang kedua diasumsikan bahwa bumi itu elipse. Dalam studi kasus ini peneliti menggunakan asumsi bahwa bumi itu bulat. Peneliti melakukan pengujian rumus hasil jarak antar 2 lokasi yang sama dengan rumus yang berbeda dan membandingkan pada hasil jarak pada *google maps*. Hasil jarak dengan rumus asumsi bumi itu bulat lebih mendekati hasil *google maps*.

Keterangan :

- d = sudut pusat bumi
- S = Jarak
- φ_1 = latitude kereta
- φ_2 = latitude perlintasan
- L1 = longitude kereta
- L2 = longitude perlintasan

III. METODOLOGI PENELITIAN

A. *Topologi Jaringan*

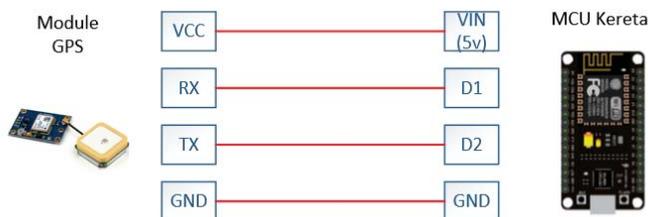


Gambar. 8. Topologi Jaringan

Topologi Jaringan sistem peringatan kedatangan kereta api yang akan di bangun ditunjukkan oleh Gambar 8.

Implementasi jaringan menggunakan 3 router Mikrotik RB951Ui-2nd (hAP) dengan *backbone* menggunakan kabel UTP, lalu untuk terhubung langsung dengan user/client, setiap router akan menggunakan mode *AP Bridge*. Mode *AP Bridge* memungkinkan untuk setiap router memancarkan sinyal masing-masing, kemudian diperlukannya EoIP Tunneling untuk membuat setiap router terkoneksi satu sama lain antara router pusat dan setiap cabang-cabangnya. Server akan terhubung langsung dengan router pusat di pos satpam koinonia menggunakan kabel UTP dengan *IP Static*, kemudian setiap MCU akan terkoneksi melalui *wireless dengan IP Dinamis atau DHCP (Dynamic Host Configuration Protocol)*.

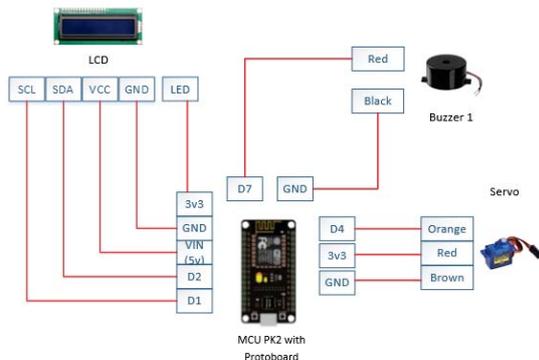
B. Rangkaian Node



Gambar. 9. Rangkaian Pemasangan pin GPS Module pada MCU Kereta

Rangkaian pemasangan pin untuk pemasangan GPS Modul pada *nodeMCU* kereta di tunjukan pada Gambar 9.

- Sensor memiliki 4 pin yang di tempatkan seperti berikut:
- VCC, di hubungkan pada pin VIN (5v) nodeMCU kereta
 - RX, dihubungkan pada pin input D1
 - TX, dihubungkan pada pin input D2
 - GND, dihubungkan ke GND pada nodeMCU



Gambar. 8. Rangkaian Pemasangan pin LCD, Buzzer dan Servo pada MCU Kereta

Rangkaian pemasangan pin untuk pemasangan LCD, Buzzer dan Servo pada *nodeMCU* PK di tunjukan pada Gambar 10. LCD memiliki 4 pin yang di tempatkan seperti berikut:

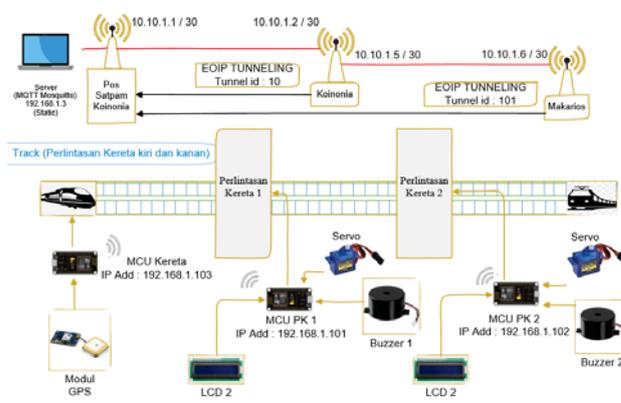
- SCL, di hubungkan pada pin input D1
- SDA, dihubungkan pada pin input D2
- VCC, di hubungkan pada pin VIN (5v) nodeMCU kereta
- GND, dihubungkan ke GND pada nodeMCU
- LED, dihubungkan ke 3v3 pada nodeMCU

Buzzer memiliki 2 pin :

- Kabel Merah, dihubungkan pada pin input D7
- Kabel Black, dihubungkan ke GND pada nodeMCU

Servo memiliki 3 pin :

- Kabel Orange dihubungkan dengan pin inputan yaitu D4
- Kabel Merah, dihubungkan ke 3v3 pada nodeMCU
- Kabel Coklat, dihubungkan ke GND pada nodeMCU



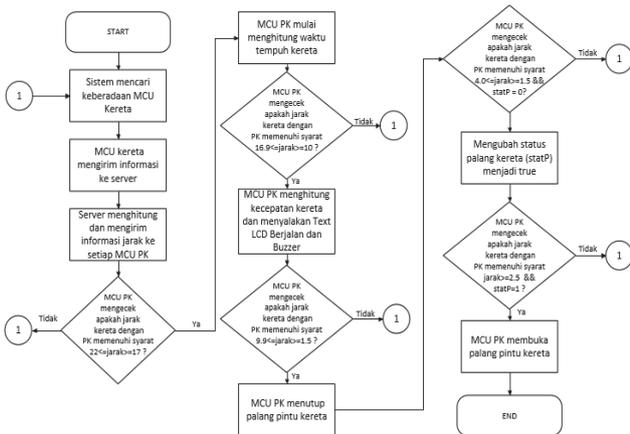
Gambar. 9. Topologi Jaringan dan Tata Letak secara Keseluruhan

Tata letak sistem secara keseluruhan yang akan dibangun ditunjukan oleh Gambar 9 Implementasi di eksekusi di halaman kampus UKDW akan menggunakan 2 buah *nodeMCU* (yang tersambung dengan 2 Buzzer, 2 device LCD dan 2 Servo) yang diletakan di area perlintasan kereta tanpa palang pintu dan 1 *Node MCU* di dalam kereta (yang terhubung dengan *module GPS*) . Setiap *Node MCU* akan terhubung menuju Mikrotik *RouterOS* dengan tujuan mendapat akses jaringan lokal. Kemudian akan tersedia 1 server untuk mengolah data melalui perhitungan yang akan dibangun melalui pemrograman Web. Semua *Node MCU* akan terhubung ke server untuk pengolahan data secara terus menerus.

C. Flowchart Cara Kerja Sistem

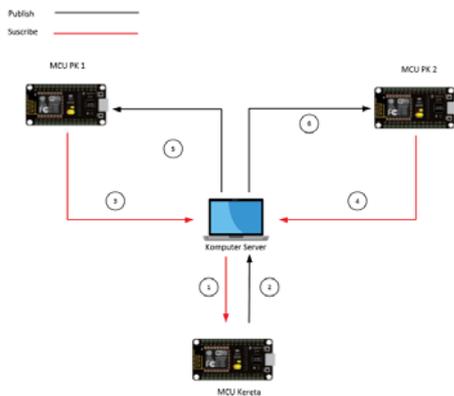
Pembuatan *flowchart* cara kerja sistem bertujuan untuk mempermudah memahami proses kerja alat ini. Gambar 10 merupakan *flowchart* dari sistem yang akan diimplementasikan. Alur dimulai dengan sistem melacak jarak kereta yang sudah terpasang *GPS* melalui *Node MCU*, *Node MCU* yang terletak di kereta mengirim informasi berupa Longitude dan Latitude ke *MQTT Broker*, setelah itu *MQTT* akan menerima dan meneruskan data untuk diolah di komputer server. Kemudian hasil perhitungan jarak oleh server, kemudian server akan mengirim hasil perhitungan jarak antara kereta dengan setiap perlintasan kereta melalui *MQTT*. Pengecekan jarak dilakukan di masing-masing perlintasan kereta, apakah jarak antara kereta dengan PK kurang dari 4,5 KM. Jika tidak, maka sistem akan melakukan pengecekan secara berkala dan jika iya, sistem akan mengirim perintah ke *MCU* di perlintasan untuk mulai menghitung waktu tempuh kereta dalam dengan counter up. Kemudian *MCU PK* (Perlintasan Kereta) mengecek apakah jarak kereta dengan PK masuk dalam syarat selanjutnya yaitu apakah jarak antara kereta dengan PK kurang dari 3,5 KM, jika iya *MCU PK* menghitung kecepatan kereta serta menyalakan *text LCD berjalan* dan *Buzzer*, jika tidak, pengecekan akan terus berkelanjutan. Dilanjutkan dengan pengecekan lain apakah jarak antara kereta dengan PK kurang dari 2 KM jika iya, *MCU PK* akan menutup palang pintu kereta, jika tidak, pengecekan akan terus berkelanjutan

sampai jarak memenuhi syarat. Selanjutnya pengecekan antara jarak kereta dan PK apakah sudah dalam jarak kurang dari 0,5 KM status palang *false*, jika tidak MCU PK akan melakukan pengecekan ulang, jika iya, maka ubah status palang menjadi *true*. Perubahan *status* ini untuk menandai bahwa kereta akan lewat, sehingga pada if selanjutnya dapat dilakukan pembukaan pintu palang kereta tanpa harus menunggu hasil jarak yang sama dengan penutupan palang kereta. Diakhiri dengan pengecekan apakah jarak antara MCU dan kereta sudah mencakup jarak lebih besar dari 1 KM dan apakah *status* palang kereta *true*, jika iya, MCU PK akan membuka palang perlintasan kereta api, jika tidak MCU PK akan melakukan pengecekan secara terus-menerus. Sistem akan terus melakukan looping pengecekan kembali posisi kereta secara berkala dan terus mengikuti proses seperti Gambar 10 flowchart sistem.



Gambar. 10. Flowchart Cara Kerja Sistem Peringatan Kedatangan Kereta Api

D. Perancangan Komunikasi Data dengan MQTT



Gambar. 11. Gambar Publish dan Subscribe di setiap MCU

MQTT berfungsi sebagai media penengah antara server dan MCU, MQTT menghubungkan dengan menggunakan topic atau alamat rumah sebuah device supaya tidak ada satupun data yang akan tersesat atau salah alamat. MQTT memakai metode *Publish & Subscribe*, simpelnya publisher adalah si pengirim dan subscriber adalah si penerima. Pada sistem peringatan kedatangan kereta otomatis ini MQTT bekerja seperti pada gambar 11 diawali dengan server melakukan *subscribe* berdasarkan topic yang disetujui dengan MCU kereta disaat bersamaan MCU kereta *mempublish longitude dan latitude kereta ke Server*. Server menghitung jarak antara MCU kereta dengan MCU

perlintasan masing-masing, kemudian kedua MCU perlintasan *subscribe* ke komputer server sesuai topic yang disetujui antara komputer server dan MCU perlintasan masing-masing. Komputer Server akan *mempublish* hasil jarak antara kereta dengan masing masing mcu perlintasan. Hasil jarak yang diterima dari *server* akan diolah di setiap MCU perlintasan kereta untuk menghitung kecepatan kereta dan sistem peringatan kedatangan kereta secara otomatis akan aktif ketika semua syarat terpenuhi, dan jika syarat tidak terpenuhi sistem sistem peringatan kedatangan kereta tidak akan aktif.

E. Perancangan Antarmuka

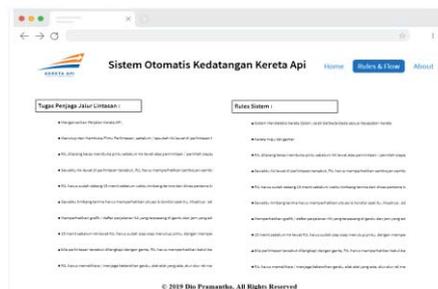
Antarmuka web monitoring sistem peringatan otomatis kedatangan kereta api ini di buat menjadi 3 Bagian yaitu halaman utama seperti ditunjukkan di gambar 12 berisi tabel data lokasi setiap MCU, halaman rules dan flow sistem seperti di gambar 13 yang berisi rules sistem dan flow dari sistem. Pada gambar 14 adalah halaman about yang berisi biodata dari peneliti.

1) Halaman Utama (Home)



Gambar. 12. Perancangan antarmuka halaman home

2) Rules & Flow



Gambar. 13. Perancangan antarmuka Rules & Flow

3) About



Gambar. 14. Tampilan About

IV. HASIL DAN ANALISIS

A. Sistem Database

Sistem Database pada penelitian ini menggunakan *mysql*. Database akan tersimpan pada komputer server yang berperan sebagai *broker* sekaligus *web monitoring* lokal.

Gambar 15 menjelaskan masing-masing tabel yang ada dalam database sistem_kereta.sql.

- Tabel koordinat merupakan tabel yang digunakan untuk menyimpan semua lokasi *static* perlintasan kereta yang ada pada sistem. Atribut yang dimiliki oleh table ini yaitu kode_koordinat sebagai *primary key*, kemudian ada *longitude*, *latitude* dan juga waktu.
- Tabel Koordinat_kereta merupakan tabel yang digunakan untuk menyimpan lokasi kereta yang terus berubah-ubah atau *dinamis*.

DATABASE SISTEM KERETA

koordinat		koordinat_kereta	
PK	kode_koordinat : int (100)	PK	kode_koordinat : int (100)
	longitude : decimal(65,6)		longitude : decimal(65,6)
	latitude : decimal(65,6)		latitude : decimal(65,6)
	waktu : datetime		waktu : datetime
	keterangan : VARCHAR(255)		

Gambar. 15. Database Sistem Kereta

B.Implementasi Web

Web ini dibuat menggunakan bahasa pemrograman *php* sedangkan untuk merapikan tampilan menggunakan *Bootstrap* versi 4.3.1 sebagai *framework Cascading Style Sheets* (CSS). Semua file *php*, *css* dan *javascript* tersimpan di *server* lokal. Web ini mempunyai beberapa section :

a. Section Home

Tabel Koordinat Perlintasan dan Kereta

No	Keterangan	Longitude	Latitude	Waktu
1	MCU PK 1	110.378731	-7.785155	2019-10-28 12:17:24
2	MCU PK 2	110.378685	-7.785371	2019-10-28 13:18:29
3	MCU KERETA			2019-11-26 00:08:18am

Jarak antara Kereta dan Perlintasan Kereta (1) : (M)
 Jarak antara Kereta dan Perlintasan Kereta (2) : (M)

Gambar. 16. Section Home

- Gambar 16 adalah tampilan section Home dimana section index adalah section pertama yang akan ditampilkan saat memasuki web.
 - Pada tampilan section ini terdapat tampilan lokasi MCU PK 1, MCU PK 2 dan MCU Kereta, serta jarak antara MCU Kereta dengan setiap MCU PK. MCU Kereta dan hasil perhitungan jarak antar setiap lokasi akan terus diupdate secara berkala.
- #### b. Section Rule Sistem

RULES SISTEM

- Sistem berjalan Otomatis
- Kereta telah dibekali GPS untuk melacak posisi kereta secara realtime
- Jarak antara kereta dan setiap perlintasan kereta dihitung dengan rumus
- Ketika kereta berada Range 17 - 22 (m), sistem akan mengambil waktu tempuh dan jarak tempuh yang kereta lewat
- Ketika kereta berada Range 10 - 16 (m), sistem akan mengirimkan buzzer, led led berjalan dan menghitung kecepatan rata-rata kereta
- Ketika kereta berada Range 1.5 - 9 (m) sistem akan menutup palang
- Ketika kereta berada di jarak < (2.5) m sistem akan membuka palang

Gambar. 17. Section Rules

- Section rules adalah section yang menampilkan Rule Base dari sistem peringatan kedatangan kereta otomatis.
- Section rules telah berhasil di implementasikan ditunjukkan pada gambar 17.

c. Section PJJ (Penjaga Jalan Lintasan)

TUGAS PENJAGA JALAN LINTASAN (PJJ)

- Menutup dan Membuka Palang Perlintasan, otomatis / manual / KA lewat di perlintasan tersebut sesuai kebutuhan untuk mengemponkan PERKA.
- PJJ dilarang keras membuka palang sebelum KA lewat atau pemantauan / pemilih siap-siaga, kecuali ada alasan langsung yang bertanggung jawab.
- Sewaktu KA lewat di perlintasan tersebut, PJJ harus memperhatikan semaksimal mungkin yang tidak dibantu oleh KA di lok masauin di rangkaian tersebut.
- PJJ harus sudah datang 15 menit sebelum waktu lintang kereta dari diran pertama ke diran berikutnya.
- Mengemponkan Perlintasan Kereta API.
- Bila perlintasan tersebut dilangkahi dengan kereta, PJJ harus memperhatikan betul betul yang gerak tersebut, sehingga PJJ yakin dari mana KA akan lewat dari stop untuk menutup palang.
- PJJ harus memperhatikan / menjaga ketertutupan palang, dari saat yang ada, dari saat saat istirahat jalan raya yang di perlintasan tersebut sehingga sewaktu KA lintang kemudian akan lewat tidak terganggu.
- Selama tidak ada KA lewat, PJJ harus membersihkan buan keran / Api PJJ, setiap 100 - 200m

Gambar. 18. Section PJJ

- Tampilan section PJJ adalah section yang menampilkan tugas dari penjaga jalan Lintasan kereta api pada umumnya.
- Tugas yang dicantumkan di web adalah tugas asli dari seorang penjaga perlintasan kereta.
- Section telah berhasil di implementasikan ditunjukkan pada gambar 18.

C.Implementasi MQTT Broker

Broker MQTT digunakan sebagai penghubung antara *node-node* yang ada dengan *web*. Aplikasi *Mosquitto* merupakan aplikasi yang memberikan layanan *private broker* yang dapat di *install* pada *windows*. Aplikasi *Mosquitto* tidak memiliki tampilan *Graphical User Interfaces* (GUI), sehingga *broker* hanya dapat dijalankan melalui *Command Prompt* dengan cara mengakses folder yang sudah *terinstall*, kemudian menjalankan perintah seperti pada Gambar 19

```
C:\Program Files\mosquitto>mosquitto -v -c mosquitto.conf
```

Gambar. 19. Perintah untuk menjalankan broker MQTT

Broker MQTT mulai menjalankan layanan dengan membuka *port 1883* dan *port 8883*. *Port 1883* digunakan oleh protokol MQTT untuk pengiriman pesan, sedangkan *port 8883* digunakan untuk *websocket* karena pada sisi klien *web* akan melakukan *request* terhadap *broker MQTT*, maka dibutuhkan *port 8883*. Semua pengaturan *port* ada di dalam *file mosquito.conf*.

Proses selanjutnya ditunjukkan seperti pada Gambar 20 yaitu *broker* akan menerima *request* pengenalan dari klien baru (*node1*, *node2*, *webclient*) yang ditandai dengan *CONNACK* yang artinya sudah terhubung. Kemudian klien akan mendaftarkan semua topik yang *unsubscribe* ke *broker* di tandai dengan *SUBSCRIBE*, sedangkan *SUBACK* berarti semua topik telah diterima dan didaftarkan oleh *broker*.

```

1578443671: mosquitto version 1.5.4 starting
1578443671: Config loaded from mosquitto.conf.
1578443671: Opening ipv6 listen socket on port 1883.
1578443671: Opening ipv4 listen socket on port 1883.
1578443671: Opening websockets listen socket on port 8883.
1578443671: New client connected from 192.168.18.25 as web_dio (ci, k0).
1578443671: No will message specified.
1578443671: Sending CONNACK to web_dio (0, 0)
1578443671: Received SUBSCRIBE from web_dio
1578443671: outTopic# (qos 0)
1578443671: web_dio 0 outTopic/#
1578443671: inTopic# (qos 0)
1578443671: web_dio 0 inTopic/#
1578443671: Sending CONNACK to web_dio
1578443672: New connection from 192.168.18.25 on port 1883.
1578443672: New connection from 192.168.18.25 on port 1883.
1578443672: New client connected from 192.168.18.25 as Luar1578443866499 (ci, k3b, u LCD)
1578443672: Will message specified (0 bytes) (r0, q0).
1578443672: server/
1578443672: Sending CONNACK to Luar1578443866499 (0, 0)
1578443672: New client connected from 192.168.18.25 as LCD_PK_da189cf4-d448-40a5-88a8-17c47f82a0ab1578443866572 (ci, 0, u LCD)
1578443672: Will message specified (0 bytes) (r0, q0).
1578443672: server/pk1/hasilJarak
1578443672: Sending CONNACK to LCD_PK_da189cf4-d448-40a5-88a8-17c47f82a0ab1578443866572 (0, 0)
    
```

Gambar. 20. Cara kerja broker MQTT (1)

Gambar 21 merupakan proses PUBLISH nilai yang dimulai dari *NodeKereta*. Nilai GPS akan di PUBLISH oleh *Node Kereta* dengan topik “outTopic/koordinat”, kemudian oleh *broker* MQTT, nilai tersebut diterima dan di PUBLISH kembali menuju semua klien yang telah mendaftarkan atau melakukan *subscribe* terhadap topik “outTopic/koordinat”. Terdapat 1 klien yang melakukan *subscribe* terhadap topik “outTopic/koordinat” yaitu klien dari *web_dio*.

```

1578445127: Received PUBLISH from nodekereta (d0, q0, r0, m0, 'outTopic/koordinat', ... (20 bytes))
1578445127: Sending PUBLISH to web_dio (d0, q0, r0, m0, 'outTopic/koordinat', ... (20 bytes))
1578445127: Sending PUBLISH to nodemcuPK (d0, q0, r0, m0, 'outTopic/koordinat', ... (20 bytes))
1578445127: Received PUBLISH from web_dio (d0, q0, r0, m0, 'server/pk1/hasilJarak', ... (8 bytes))
1578445127: Sending PUBLISH to nodemcuPK (d0, q0, r0, m0, 'server/pk1/hasilJarak', ... (8 bytes))
1578445127: Received PUBLISH from web_dio (d0, q0, r0, m0, 'server/pk2/hasilJarak', ... (8 bytes))
    
```

Gambar. 21. Cara kerja broker MQTT (2)

```

1578446241: Sending PINGRESP to nodemcuPK
1578446243: Received PINGREQ from LCD_PK_da189cf4-d448-40a5-88a8-17c47f82a0ab1578443866572
1578446243: Sending PINGRESP to LCD_PK_da189cf4-d448-40a5-88a8-17c47f82a0ab1578443866572
1578446243: Received PINGREQ from Luar1578443866499
1578446243: Sending PINGRESP to Luar1578443866499
1578446244: Received PINGREQ from nodekereta
1578446244: Sending PINGRESP to nodekereta
1578446244: Received PUBLISH from nodekereta (d0, q0, r0, m0, 'outTopic/koordinat', ... (20 bytes))
1578446244: Sending PUBLISH to web_dio (d0, q0, r0, m0, 'outTopic/koordinat', ... (20 bytes))
1578446244: Sending PUBLISH to nodemcuPK (d0, q0, r0, m0, 'outTopic/koordinat', ... (20 bytes))
1578446245: Received PUBLISH from web_dio (d0, q0, r0, m0, 'server/pk1/hasilJarak', ... (8 bytes))
1578446245: Sending PUBLISH to nodemcuPK (d0, q0, r0, m0, 'server/pk1/hasilJarak', ... (8 bytes))
1578446245: Received PUBLISH from web_dio (d0, q0, r0, m0, 'server/pk2/hasilJarak', ... (8 bytes))
1578446248: Received PINGREQ from nodemcuPK
1578446248: Sending PINGRESP to nodemcuPK
1578446253: Received PINGREQ from LCD_PK_da189cf4-d448-40a5-88a8-17c47f82a0ab1578443866572
1578446253: Sending PINGRESP to LCD_PK_da189cf4-d448-40a5-88a8-17c47f82a0ab1578443866572
1578446253: Received PINGREQ from Luar1578443866499
1578446254: Sending PINGRESP to Luar1578443866499
1578446254: Received PINGREQ from nodekereta
1578446254: Sending PINGRESP to nodekereta
1578446254: Received PUBLISH from nodekereta (d0, q0, r0, m0, 'outTopic/koordinat', ... (20 bytes))
1578446254: Sending PUBLISH to web_dio (d0, q0, r0, m0, 'outTopic/koordinat', ... (20 bytes))
1578446254: Sending PUBLISH to nodemcuPK (d0, q0, r0, m0, 'outTopic/koordinat', ... (20 bytes))
1578446255: Received PUBLISH from web_dio (d0, q0, r0, m0, 'server/pk1/hasilJarak', ... (8 bytes))
1578446255: Sending PUBLISH to nodemcuPK (d0, q0, r0, m0, 'server/pk1/hasilJarak', ... (8 bytes))
1578446255: Received PUBLISH from web_dio (d0, q0, r0, m0, 'server/pk2/hasilJarak', ... (8 bytes))
1578446256: Received PINGREQ from nodemcuPK
1578446256: Sending PINGRESP to nodemcuPK
    
```

Gambar. 22. Cara kerja broker MQTT (3)

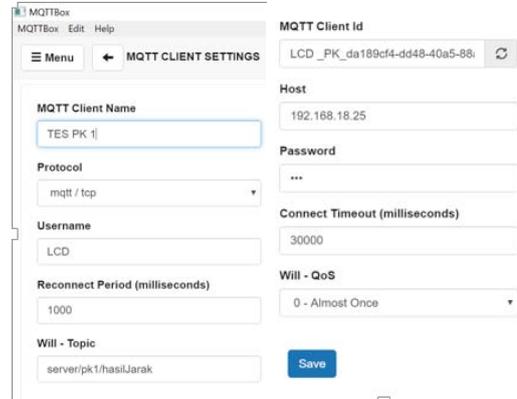
Gambar 22 merupakan proses PUBLISH hasil Jarak yang dimulai dari *web_dio*. Proses ini sama seperti proses PUBLISH sebelumnya yang dijelaskan pada Gambar 21, hanya berbeda topik dan berbeda *node*.

Hasil jarak akan di PUBLISH oleh *web_dio* dengan 2 topik berbeda, pertama “server/pk1/hasilJarak” dan kedua “server/pk2/hasilJarak”, kemudian oleh *broker* MQTT, nilai tersebut di PUBLISH kembali menuju semua klien yang telah mendaftarkan atau melakukan *subscribe* terhadap topik “server/pk1/hasilJarak” dan “server/pk2/hasilJarak”. Terdapat 1 klien yang melakukan *subscribe* terhadap topik “server/pk1/hasilJarak” yaitu klien dari *Node PK1*, dan demikian dengan topik “server/pk2/hasilJarak” memiliki klien dari *Node PK2*.

D. Testing dengan MQTTBox

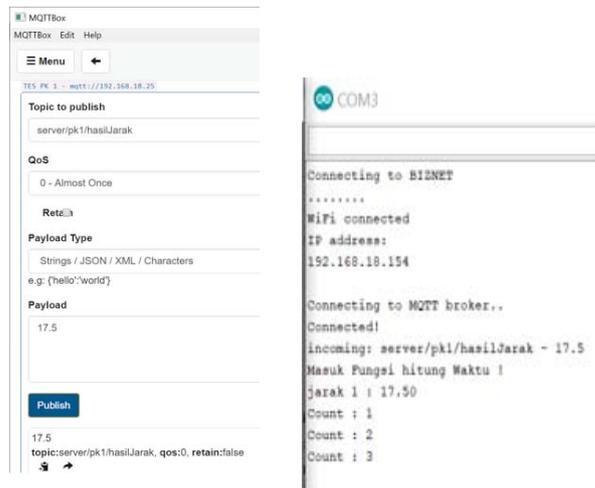
MQTTBox adalah sebuah aplikasi yang dapat membantu memudahkan *developer* arduino yang menggunakan MQTT dalam hal pengujian atau *testing* program. MQTTbox dapat mengirimkan dan menerima data atau dalam pemrograman arduino disebut *publish* dan *subscribe* dalam bentuk strings,

JSON, binary, dll. MQTT box memiliki *Graphical User Interfaces (GUI)*, Gambar 23 adalah tampilan untuk membuat koneksi baru kepada *node-node*. Dimulai dengan membuat koneksi baru yang diperlukan untuk membuat koneksi baru “TES PK 1” sebagai Nama MQTT klien, identitas klien MQTT “LCD(*code auto generate*)”, Protocol yang digunakan “mqtt/tcp”, host “192.168.18.25” (IP Server MQTT Broker), username “LCD”, password “dio”, dan topic “server/pk1/hasilJarak”.

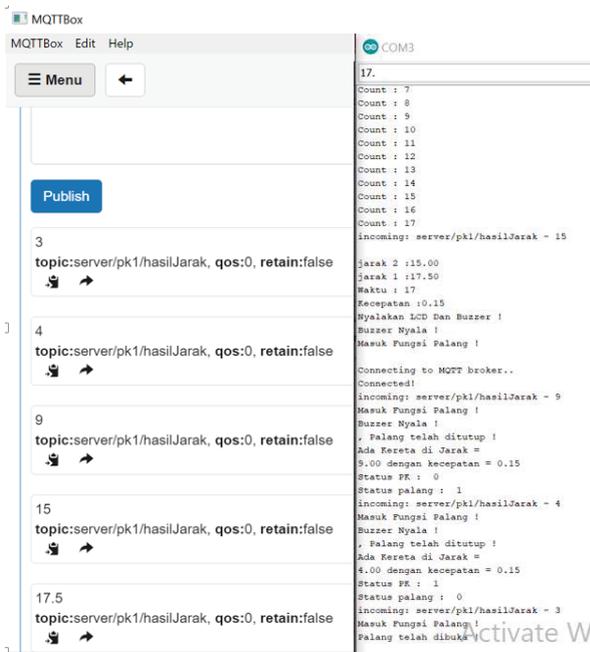


Gambar. 23. Membuat klien baru MQTTBox

Pengujian akan dilakukan sesuai flowchart cara kerja sistem pada Gambar 10. Pengujian di tunjukkan pada Gambar, MQTTBox *publish* nilai hasil jarak “17” dengan topik “server/pk1/hasilJarak”. Pada penerapan sistem di lapangan *publish* hasil jarak ini seharusnya dilakukan dari web, tetapi dengan MQTTBox data yang seharusnya diterima dari web telah digantikan. Pengujian sistem palang kereta secara total ditunjukkan pada Gambar 25.



Gambar. 24. MQTTBox Publish

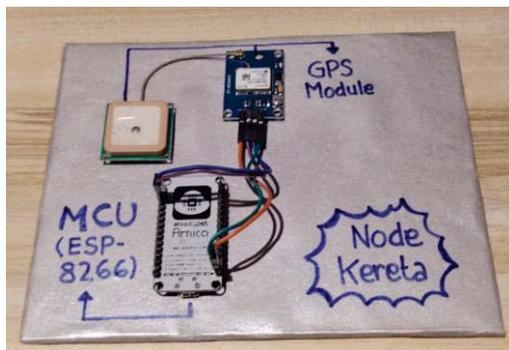


Gambar. 25. Pengujian MQTTBox

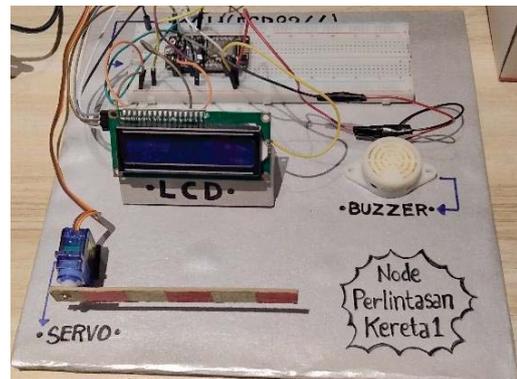
E. Implementasi Alat

Implementasi jaringan *seamless wireless* menggunakan 3 buah mikrotik Router RB951Ui-2nD (hAP). Backbone implementasi jaringan ini memakai kabel sebagai jalur bertukar data. Router 1 dipasang di depan gedung koinonia bagian utara UKDW. Dari router 1 membagi listrik dengan PoE (Power of Ethernet) ke router 2, dikarenakan lokasi router 2 tidak dapat dijangkau oleh terminal. Router 2 dipasang di gedung koinonia bagian selatan. Router 3 dipasang di depan gedung makarios. Pengecekan keberhasilan implementasi jaringan *seamless wireless EoIP* dengan berjalan dari pos satpam koinonia sampai ke gedung makarios tanpa putus koneksi wireless dan ping dari client Router 3 ke Server.

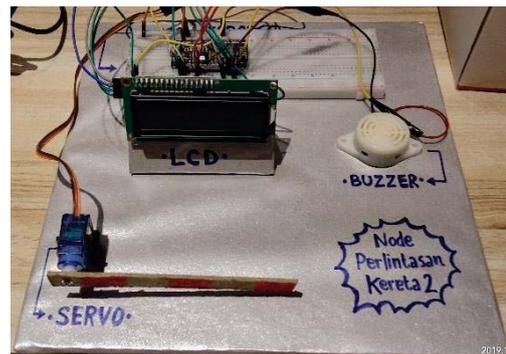
Implementasi sistem peringatan kedatangan kereta otomatis menggunakan 3 *node* pembangun sistem yaitu *node* kereta, *node* perlintasan kereta 1, dan *node* perlintasan kereta 2. *Node* kereta terdiri dari 2 alat yaitu *MCU ESP8266 Amica* dan *GPS Module Neo-6mv2*. *Node* kereta berhasil di implementasikan, karena *node* kereta telah dapat mengirimkan lokasi kereta selama pengujian sistem. Gambar 26 adalah hasil dari implementasi *node* kereta.



Gambar. 26. Node Kereta



Gambar. 27. Node Perlintasan kereta 1



Gambar. 28. Node Perlintasan kereta 2

Node Perlintasan Kereta terdiri dari *servo*, *buzzer*, *LCD* dan *MCU ESP8266 amica*. Dua *node* dipasang pada gedung koinonia dan gedung makarios. Gambar 27 dan 28 adalah hasil Implementasi dari *node* pk1 dan *node* pk2.

F. Hasil Pengujian Alat

Pengujian sistem peringatan kedatangan kereta otomatis diuji dengan *checklist* pengujian yang telah dirancang di bab sebelumnya. *Checklist* pengujian di bagi menjadi dua bagian yaitu *checklist* pk 1 dan cek list pk 2. *Checklist* pk 1 terdiri dari 12 poin dan *checklist* pk 2 terdiri dari 6 poin, *checklist* pk 1 memiliki 6 poin lebih banyak dari pk 2 dikarenakan pada pk 1 ditambahkan *checklist* dari Server. Percobaan pengujian dilakukan 30 kali untuk setiap cek list, Pengujian dilakukan dalam waktu 2 hari berturut-turut. Selama 30 kali percobaan dilakukan tidak ada masalah yang terlihat. Sistem berjalan sesuai flowchart kerja. Dari hasil cek list pk 1 dan cek list pk 2 dinyatakan sistem peringatan kedatangan kereta otomatis berhasil dibangun dengan benar ditunjukkan pada Gambar 29 dan pada Gambar 30. Dan bahwa implementasi jaringan dengan teknologi *wireless seamless EoIP* sangat cocok dengan sistem peringatan kedatangan kereta otomatis yang membutuhkan jaringan yang stabil dan tidak terputus oleh pergantian kawasan wireless.

Perlintasan Kereta 1
Percobaan : 6

NO	Task	YA	Tidak	Keterangan
1	Komputer server (Localhost & MQTT) ON	✓		
2	Seamless Wireless 1, 2, 3 ON	✓		
3	Node MCU Kereta ON	✓		
4	Node MCU PK1 ON	✓		
5	MCU Kereta dapat mengirimkan (publish) longitude dan latitude kereta kepada Komputer Server	✓		
6	Komputer server dapat menghitung jarak antara lokasi kereta (MCU Kereta) dengan lokasi perlintasan (MCU PK1)	✓		
7	Komputer server dapat menghitung jarak antara lokasi kereta (MCU Kereta) dengan lokasi perlintasan (MCU PK2)	✓		
8	Komputer Server dapat mengirim / publish hasil perhitungan jarak antar lokasi kereta (MCU Kereta) dengan lokasi perlintasan (MCU PK1)	✓		
9	MCU PK1 dapat menghitung waktu dengan range jarak $22 \leq \text{hasiljarak} \leq 17$	✓		
10	MCU PK1 dapat menghitung kecepatan kereta api dengan range jarak $16.9 \leq \text{hasiljarak} \leq 12$ dan MCU PK1 dapat menyalakan LCD dan Buzzer	✓		
11	MCU PK1 mendeteksi kedatangan kereta dan dapat menutup palang perlintasan kereta dalam range jarak	✓		
12	MCU PK1 dapat membuka palang perlintasan kereta ketika hasiljarak ≥ 2.5 , setelah kereta melewati palang	✓		

Felix

Kecepatan Kereta 0,29 m/s

Gambar. 29. Cek List Pengujian Lapangan Perlintasan Kereta 1

Perlintasan Kereta 2
Percobaan : 6

NO	Task	YA	Tidak	Keterangan
1	Node MCU PK2 ON	✓		
2	Komputer Server dapat mengirim / publish hasil perhitungan jarak antar lokasi kereta (MCU Kereta) dengan lokasi perlintasan (MCU PK2)	✓		
3	MCU PK1 dapat menghitung waktu dengan range jarak $22 \leq \text{hasiljarak} \leq 17$	✓		
4	MCU PK1 dapat menghitung kecepatan kereta api dengan range jarak $16.9 \leq \text{hasiljarak} \leq 12$ dan MCU PK1 dapat menyalakan LCD dan Buzzer	✓		
5	MCU PK2 mendeteksi kedatangan kereta dan dapat menutup palang perlintasan kereta dalam range jarak	✓		
6	MCU PK2 dapat membuka palang perlintasan kereta ketika hasiljarak ≥ 2.5 , setelah kereta melewati palang	✓		

Renyji : katon Bilug

Kecepatan Kereta 0,31 m/s

Gambar. 30. Cek List Pengujian Lapangan Perlintasan Kereta 2

V. KESIMPULAN DAN PEMNGEMBANGAN SISTEM

Penelitian yang dilakukan sebanyak 30 kali percobaan mendapatkan hasil bahwa sistem otomatis peringatan kedatangan kereta dapat mendeteksi kedatangan kereta dan telah berhasil dibangun dengan benar dan berjalan sesuai flow sistem yang sudah dirancang dan dengan pembangunan jaringan dengan menggunakan teknologi *seamless wireless EoIP* yang memungkinkan pembangunan jaringan bahkan di titik buta tidak dapat dijangkau oleh operator seluler ataupun GSM. Oleh karena itu *seamless wireless EoIP* ini sangat cocok dalam membantu pembangunan sistem peringatan kedatangan kereta api dengan palang otomatis dibuktikan dengan setiap alat dapat terhubung dan tidak terputus selama 30 kali percobaan. Sistem masih butuh beberapa pengembangan dari sisi jaringan lokal menjadi jaringan publik dengan memikirkan keamanan data dan sistem, dari sisi perhitungan kecepatan rata-rata masih dalam batasan jarak tertentu, pengembangan selanjutnya dalam menghitung kecepatan tanpa harus memikirkan batasan jarak. Penelitian selanjutnya palang pintu perlintasan dapat mendeteksi keberadaan benda atau

mahluk hidup pada saat penutupan palang pintu kereta dan juga kedepannya sistem dapat mengatasi perlintasan kereta 2 jalur.

DAFTAR PUSTAKA

- [1] H. P. Klaten, "Laka Kereta Api di Perlintasan Tanpa Palang Pintu, Pengendara Motor Terseret Hingga 10 Meter," 09 Januari 2018. [Online]. Available: <http://jateng.tribunnews.com/2018/01/09/laka-kereta-api-di-perlintasan-tanpa-palang-pintu-pengendara-motor-terseret-hingga-10-meter>.
- [2] M. A. Firdaus, "Miniatur Palang Pintu Kereta Otomatis Dengan Menampilkan Kecepatan Kereta Serta Waktu Tunggu Menggunakan Arduino," pp. 1-100, 2015.
- [3] D. Cahyadi, "Pemanfaatan Fitur Tunneling Menggunakan Virtual Interface EoIP di MikrotikRouterOS Untuk Koneksi Bridging Antar Kantor Melalui Jaringan ADSL Telkom Speedy," pp. 1-5, 2010.
- [4] H. Briantoro, F. Arifin and A. Hendriawan, "Sistem Informasi Posisi Kereta Api," pp. 1-6, 2011.
- [5] A. Pranindya, "Pendeteksi dan Pelacakan Keberadaan manusia menggunakan GPS," pp. 7-33, 2014.
- [6] D. F. Arranda, *KONTROL LAMPU RUANGAN BERBASIS WEB MENGGUNAKAN NODEMCU ESP8266*, pp. 3-4, 2017.
- [7] A. Destiara, "Sistem Pemberi Pakan Ayam Terjadwal Menggunakan Modul WiFi," pp. 2-3, 2017.
- [8] K. Anggoro, "ANALISIS WIRELESS DISTRIBUTION SYSTEM (WDS) DENGAN 4 BUAH RB951Ui-2HnD," pp. 9-12, 2017.
- [9] "Manual:Interface/EoIP," 22 Mei 2018. [Online]. Available: <https://wiki.mikrotik.com/wiki/Manual:Interface/EoIP>.
- [10] M. P. Naully, "Detektor Posisi Kendaraan Bermotor dengan Sensor GPS dan Aplikasi Android (Transmitter)," *eprints repository software*, p. 11, 2015.
- [11] M. Jean, *Astronomical Algorithms* (second edition), Richmon, Virginia: Willmann-Bell, Inc, 1998.