

Sistem Kendali Kunci Otomatis Pada Motor Matic Menggunakan Mikrokontroler Berbasis Android

Automatic Key Control System On Matic Motors Using Android-Based Microcontroller

Bonifacius Vicky Indriyono^{1*}, Eko Hari Rachmawanto², Choerul Umam³, Natalinda Pamungkas⁴, Faiz Yusa Saidalvi⁵

^{1,4,5}Sistem Informasi, Universitas Dian Nuswantoro

²Teknik Informatika, Universitas Dian Nuswantoro

³Manajemen, Universitas Dian Nuswantoro

Email: ¹bonifacius.vicky.indriyono@dsn.dinus.ac.id, ²eko.hari@dsn.dinus.ac.id, ³choerul.umam@dsn.dinus.ac.id, ⁴natalinda@dsn.dinus.ac.id, ⁵faizyusasaidalvi2000@gmail.com.

*Penulis korespondensi: bonifacius.vicky.indriyono@dsn.dinus.ac.id

ABSTRAK

Sistem kendali untuk mengatur hidup dan matinya mesin motor yang dapat difungsikan juga sebagai bentuk pengamanan kendaraan seharusnya perlu di desain serta diciptakan dengan seefektif mungkin sehingga keamanan motor dapat dijaga dan resiko kendaraan hilang dapat diminimalisir. Sistem yang dirancang dalam penelitian ini difokuskan pada bagaimana mesin sepeda motor matik dapat dihidupkan dan dimatikan menggunakan teknologi mikrokontroler yang berbasis pada sistem operasi android. Prosedur kerja dari sistem ini adalah dengan memasukkan data pada android menggunakan web browser dan asisten google, kemudian komponen mikrokontroler ESP 32 bertugas meneruskan dan mengontrol data tersebut. Proses pengujian yang dilakukan dalam penelitian ini menghasilkan kesimpulan bahwa maksimal jarak Wi-Fi antara mikrokontroler ESP32 dengan android yang berada pada sepeda motor untuk mengendalikan mesin motor kurang-lebih sekitar 16 meter. Sistem ini juga mampu menjadi mesin kendali yang pintar untuk menyalakan mesin motor dengan menggunakan perintah suara oleh pengemudi menggunakan bantuan asisten google. Pengamanan sistem ini tidak terbatas pada pengamanan saat sepeda motor *stationary* atau pada saat sepeda motor berada diparkiran, pengamanan ini juga bisa berada pada saat kendaraan digunakan, dan ketika sepeda motor ditinggalkan pengemudi pada saat menyala, maka sepeda motor akan otomatis mati jika mikrokontroler ESP32 berada diluar jangkauan Wi-Fi/*Hotspot* yang berasal dari perangkat android.

Kata kunci: WIFI, android, mikrokontroler ESP32, asisten google

ABSTRACT

The control system to regulate the on and off of the motorcycle engine which can also be used as a form of vehicle security should be designed and created as effectively as possible so that the safety of the motorcycle can be maintained and the risk of the vehicle being lost can be minimized. The system designed in this research is focused on how an automatic motorcycle engine can be turned on and off using microcontroller technology based on the android operating system. The working procedure of this system is to enter data on android using a web browser and google assistant, then the ESP 32 microcontroller component is in charge of forwarding and controlling the data. The testing process carried out in this study resulted in the conclusion that the maximum Wi-Fi distance between the ESP32 microcontroller and the android on the motorcycle to control the motorcycle engine was approximately 16 meters. This system is also capable of being a smart control machine to turn on the motor engine using voice commands by the driver using the help of the Google Assistant. This security system is not limited to security when the motorbike is stationary or when the motorbike is parked, this security can also be present when the vehicle is in use, and when the motorbike is left by the driver when it is on, the motorbike will automatically turn off if the ESP32 microcontroller is outside. Wi-Fi/*Hotspot* coverage originating from android devices.

Keywords: WIFI, android, ESP32 microcontroller, google assistant.

1. PENDAHULUAN

Industri 4.0 atau Revolusi Industri Keempat adalah istilah yang umum digunakan untuk tingkat perkembangan industri teknologi global. Pada level keempat ini, dunia terfokus pada teknologi digital seperti smartphone, komputer, dan robotika. Revolusi ini ada kaitannya pula dengan digitalisasi teknologi. Dengan digitalisasi ini, virtualisasi proses, instalasi serta beragam program di dunia dapat dikembangkan dan diwujudkan. Pengujian virtualisasi ini selanjutnya dapat digunakan untuk membuat versi ini menjadi sangat efektif, memiliki manfaat, dan dapat melakukan penghematan biaya. Virtualisasi ini bisa dirancang dan diwujudkan di dunia nyata melalui hubungan dengan teknologi *Internet of Things*. Proses ini menyebabkan dapat terjadinya komunikasi dan kerja sama timbal balik, serta pertukaran data real-time, yang disebut sistem cyber-fisik yaitu integrasi antara proses komputerisasi dan fisik melalui tiga proses yakni identifikasi, integrasi sensor dan aktuator, serta pengembangan sensor dan aktuator. Sebagai contoh penerapannya seperti rumah pintar dan kendaraan pintar. Rumah pintar dan kendaraan pintar ini juga dapat dirancang secara otomatis dalam revolusi industri saat ini. Selain itu, pengotomatisan ini meliputi berbagai bidang yakni jaringan antar proses, informasi yang transparan, serta arahan teknis untuk pengambilan keputusan yang terdesentralisasi.

Saat ini, tanpa kita sadari teknologi *Internet of Things* sudah kita menjamur dalam kehidupan sehari-hari, sebagai contoh alat-alat yang sering kita gunakan untuk aktifitas seperti telepon seluler yang “Smart”, sistem alarm yang “Smart”, hingga mobil yang sudah “Smart”. *Internet Of Things* adalah semacam konsep dengan tujuan untuk memperbanyak jangkauan faedah dari adanya koneksi internet yang berjalan secara berkelanjutan. Menurut [1] *Internet of Things*, didefinisikan sebagai suatu konsep dengan tujuan perluasan manfaat dari koneksi internet yang tersambung secara berkelanjutan dimana kita dimungkinkan untuk mengkoneksikan mesin, maupun peralatan serta benda fisik lainnya menggunakan sensor jaringan guna memperoleh data dan melakukan pengelolaan cara kerjanya sendiri, sehingga mesin dimungkinkan dapat melakukan kolaborasi serta mampu bekerja berdasarkan informasi yang didapatkan secara independen. *Internet of Things* dapat menjadikan suatu masalah yang rumit menjadi sangat sederhana melalui proses otomatisasi dan pengamatan secara langsung dan nyata membuat teknologi IoT ini cukup memerlukan sambungan jaringan internet menggunakan telepon seluler dan jaringan Wi-Fi (*Wireless Fidelity*). *Internet Of Things* merupakan suatu ide dan gagasan yang menyebutkan bahwa semua benda maupun peralatan yang ada di dunia nyata dapat melakukan komunikasi antara satu dengan yang lainnya sebagai satu kesatuan sistem yang terpadu dengan memanfaatkan jaringan internet sebagai media penghubung [2].

Dalam kehidupan sehari-hari, banyak aktifitas yang bisa dilakukan dengan memanfaatkan kehandalan teknologi IoT. Salah satunya adalah membuat sistem kendali untuk menyalakan dan mematikan mesin sepeda motor. Menurut [3], Sepeda motor adalah salah satu alat transportasi darat yang praktis untuk digunakan, dilihat dari sisi efisiensi waktu dan biaya. Namun sepeda motor juga memerlukan perawatan berkala terutama di bagian mesin. Secara sederhana, mesin dapat dirawat dengan jalan memanaskan mesin setiap hari sebelum pemakaian ataupun jika tidak digunakan.

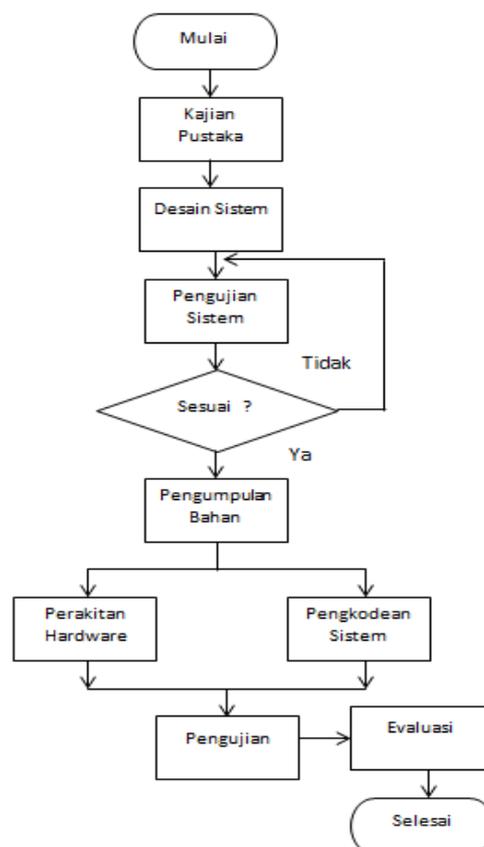
Beberapa penelitian yang membahas terkait sistem kendali berbasis IoT ini pernah dilakukan oleh beberapa peneliti, seperti yang dilakukan oleh [4] dengan judul Rancang Bangun Sistem Keamanan Rumah Berbasis Internet of Things Dengan Platform Android. Proses yang dilakukan tersebut memiliki keunggulan dan kekurangan yang disesuaikan dengan bagaimana jalannya sistem yang dirancang. Misalnya dari sisi keakuratan, sistem sering gagal dalam mengeksekusi program dan terjadinya keterlambatan respon setelah program dieksekusi. Solusi yang didapat dari keterlambatan notifikasi, maka dilakukan pengujian kembali dari proses transfer data. Pengujian perlu dilakukan untuk mengetahui kualitas layanan data. Kemudian yang dilakukan oleh [5] yang berjudul Sistem Keamanan Sepeda Motor Berbasis Internet Of Things (Iot) Dengan Smartphone Menggunakan Nodemcu. Hasil pengujian dari penelitian ini menyimpulkan bahwa proses kontrol dan transfer data dengan konsep IoT dapat diterapkan dengan baik dalam proses uji coba. Integrasi antara sensor getar sw-420, module nodemcu, relay 4 channel, dapat diketahui secara langsung pada saat alat dapat difungsikan. Secara prinsip perbedaan dengan penelitian yang dilakukan sekarang adalah dari media yang digunakan. Penelitian saat ini menggunakan media arduino UNO dan mikrokontroler ESP32. Selain itu perbedaan mendasar juga terletak pada fungsi dari aplikasi yang dibuat yakni aplikasi ini dapat digunakan untuk mengontrol hidup dan matinya mesin kendaraan. Penelitian yang dilakukan oleh [6] dengan judul Purwa Rupa Sistem Pengaman Sepeda Motor Berbasis IoT (Internet of Things) memaparkan berbagai tahapan dalam perancangan sistem pengamanan sepeda motor mulai dari perancangan *wiring* sistem, membuat sistem IoT menggunakan Arduino Mega, ESP8266, pengenalan sidik jari dan blynk di telepon pintar. Hasil dari pengujian ini menyimpulkan

diantaranya prototipe IoT untuk keamanan motor dapat dibangun dengan ESP8266, Modem, Arduino Mega 2560, aplikasi Blynk dan sensor sidik jari di telepon pintar. Keluaran dari pengujian ini berupa pesan peringatan dari aplikasi Blynk di telepon pintar yang berisi data pengguna yang sudah terverifikasi data sidik jarinya atau suksesnya mesin dinyalakan oleh pengguna. Tampilan *blynk* juga mampu menunjukkan kehandalan proses untuk memverifikasi sidik jari pengguna yang sudah disimpan dalam basis data. Perbedaan dengan penelitian sekarang adalah selain dari media yang digunakan juga fitur sistem yang dihasilkan. Sistem dalam penelitian ini mampu memberikan pesan peringatan apabila ada sesuatu yang terjadi pada kendaraan, misalnya kendaraan dinyalakan oleh orang yang tidak berkepentingan. Sistem yang dihasilkan juga mampu untuk mengontrol mesin agar mesin dapat dinyalakan atau dihidupkan secara otomatis pada jarak tertentu.

Tujuan utama dalam penelitian ini adalah merancang dan membangun sistem kendali berbasis *Internet of Things* untuk memperkuat sisi keamanan kendaraan bermotor. Hasil dari penelitian yang dilakukan ini sekaligus yang membedakan dengan penelitian lainnya adalah sistem yang dihasilkan memiliki keunggulan mampu melakukan kontrol pada mesin kendaraan secara otomatis baik menyalakan ataupun mematikan mesin menggunakan telepon seluler yang tersambung dengan wifi sehingga apabila terjadi sesuatu pada kendaraan bermotor maka dapat dilakukan pengendalian jarak jauh pada kendaraan tersebut.

2. METODE PENELITIAN

Dalam penelitian ini metode yang digunakan adalah eksperimen. Metode eksperimen merupakan suatu cara untuk melakukan percobaan tentang sesuatu hal, mengamati prosesnya serta menuliskan hasil percobaannya, kemudian hasil pengamatan itu dievaluasi, agar diperoleh hasil yang optimal. Berikut tahapan atau prosedur penelitian yang dilakukan.



Gambar-1. Metode Penelitian

- Kajian Pustaka : Kajian pustaka merupakan tahap awal dimana tim peneliti mengumpulkan berbagai sumber pustaka baik dari buku, jurnal maupun prosiding yang erat kaitannya dengan topik yang dibahas yakni seputar pemanfaatan Internet of Things dan perancangan sistem kendali kendaraan bermotor berbasis arduino.

- b) Desain Sistem : Desain sistem merupakan tahap paling awal dalam pembuatan sistem ini, pada tahap ini pembuatan desain seperti diagram blok dapat digunakan agar pemahaman dan kemudahan dalam perakitan perangkat keras dapat cepat dipahami dan perancangan antarmuka pengguna dapat digunakan agar kebutuhan dalam pembuatan aplikasi dapat lebih mudah dilakukan.
- c) Pengujian Sistem : setelah sistem dirancang, selanjutnya dilakukan pengujian. Pengujian dimaksudkan untuk mengetahui apakah jalannya sistem sudah sesuai dengan desain yang dirancang. Jika belum sesuai maka dilakukan kembali perubahan terhadap rancangan sistem.
- d) Pengumpulan Bahan : Pada langkah ini, kebutuhan alat dan bahan mencakup *hardware* maupun *software* penunjang yang akan digunakan dalam pembuatan sistem ini, perangkat *hardware* seperti perangkat arduino ESP32, dan lain-lain dapat diperoleh dan dibeli melalui *marketplace*, sedangkan untuk perangkat *software* (Arduino IDE) dapat diperoleh secara gratis dengan cara mendownload nya dari *official site* arduino.
- e) Perakitan Hardware : Perancangan pada sistem perangkat keras dapat diartikan pada kebutuhan perangkat keras yang akan dirancang pada sistem, dengan menyambungkan semua kabel-kabel dan kabel-kabel jumper pada semua pin yang telah ditentukan dan penempatan perangkat keras pada tempat di sepeda motor yang telah ditentukan.
- f) Pengkodean Sistem : Perancangan pemrograman pada sistem dapat didefinisikan pada pembuatan sistem dengan melakukan pemrograman pada arduino ESP32 didalam Arduino IDE, lalu setelah kode terverifikasi dan dapat berjalan maka kode-kode yang telah diverifikasi tersebut dapat di-*upload* kedalam arduino ESP32.
- g) Pengujian : Tahap pengujian ini adalah untuk menguji rangkaian perangkat keras dan kode sistem yang dibuat. Hal ini dimaksudkan untuk mengetahui apakah kode yang ditulis sudah sejalan dengan rangkaian perangkat kerasnya. Apakah kode sudah bisa berefek pada rangkaian perangkat keras. Pengujian sistem merupakan suatu proses pengujian perangkat lunak sekaligus perangkat keras yang telah dirancang, apakah telah sesuai dengan rancangan sistem yang dibuat atau belum. Pengujian tersebut dilakukan guna untuk menemukan apakah masih ada kesalahan dan kekurangan pada sistem yang telah dibuat.
- h) Evaluasi : Evaluasi merupakan tahap terakhir yang digunakan untuk menilai apakah semua prosedur sudah berjalan dengan baik ataukah belum. Apakah rangkaian perangkat keras dan kode sistem sudah berjalan sesuai dengan maksud dan tujuan dari penelitian.

2.1 WiFi (Wireless Fidelity)

Wireless Fidelity (Wi-Fi) merupakan standar dari jaringan tanpa kabel (nirkabel), hanya dengan komponen yang sesuai dapat terkoneksi ke jaringan. Wifi dapat dikatakan sebagai sebuah sarana penghubung komunikasi data dengan tidak menggunakan kabel yang dapat dimanfaatkan untuk melakukan komunikasi atau transfer data serta program dengan sangat cepat. Wi-Fi dapat diartikan pula sebagai teknologi yang menggunakan alat-alat elektronik untuk pertukaran data dengan memanfaatkan gelombang radio (nirkabel) dalam sebuah jaringan komputer [7]. Menurut [8], Wifi merupakan kumpulan dari standar yang diterapkan dalam sistem jaringan lokal tanpa kabel (WLAN). Wi-Fi tidak saja dapat dipergunakan untuk akses ke internet, namun dapat juga diterapkan dalam pembuatan jaringan nirkabel pada sebuah organisasi atau instansi. Wifi adalah sebuah jaringan komunikasi tanpa kabel yang menggunakan komputer LAN. Wifi memiliki jangkauan yang hanya terbatas pada area tertentu sehingga wifi dikenal dengan istilah hotspot.

Wifi memberikan banyak layanan yang menyerupai LAN seperti layanan surat elektronik (email), intranet, internet, musik, video streaming dan layanan lainnya yang berbasis IP base [9]. Secara sederhana sistem kerja dari Wifi ini adalah mula-mula menerima data dalam bentuk digital. Kemudian, dilakukan pengolahan data yang diubah menjadi gelombang radio yang selanjutnya diteruskan ke router. Proses pengiriman sinyal radio ini menggunakan media antena yang ada di adaptor. Selanjutnya router mengirimkan hasil pengolahan data dari internet ke komputer yang sudah terhubung dengan WiFi adaptor [9].



Gambar-2. Korvergansi Jaringan Wifi

2.2 Teknologi Android

Menurut [10], Android merupakan suatu sistem operasi yang berjalan pada perangkat mobile berbasis linux. Android adalah sistem operasi yang dikhususkan untuk dapat berjalan pada telepon seluler. Platform yang disediakan oleh android bersifat terbuka bagi para pengembang yang ingin membuat aplikasi mereka sendiri. Android adalah sebuah sistem operasi yang dibangun agar dapat digunakan pada perangkat *mobile* berbasis linux. Cakupannya meliputi sistem operasi, middleware, dan aplikasi. Android memiliki lisensi dibawah GNU, General Public Lisensi versi 2 (GPLv2), yang banyak disebut dengan istilah “copyleft” lisensi dimana setiap perbaikan pihak ketiga harus terus jatuh dibawah terms [10]. Android merupakan sistem operasi yang didesain untuk piranti bergerak layar sentuh seperti telepon seluler pintar dan komputer tablet berbasis Linux [11]. Android dapat juga didefinisikan sebagai sistem operasi linux yang di terapkan pada telepon seluler seperti komputer tablet (PDA) dan telepon pintar (smartphone) [12].

2.3 Arduino UNO

Menurut [13] Arduino adalah sebuah papan elektronik dengan basis mikrokontroller ATmega dimana bagian ini memenuhi sistem mikrokontroller minimum dengan tujuan supaya dapat berjalan secara mandiri (standalone controller). Komponen penting yang ada dalam papan Arduino adalah mikrokontroller 8 bit dengan merk ATmega yang diproduksi oleh Atmel corporation. Arduino memiliki banyak jenis papan yang menggunakan tipe Atmega yang bermacam-macam sesuai dengan spesifikasinya [14]. Arduino UNO merupakan sebuah papan mikrokontroller yang berbasis ATmega328. Arduino UNO ini memiliki pin sejumlah 14 pin yakni digital input/output. Dari 14 pin tersebut, 6 diantaranya dapat dipakai sebagai output (PWM), 6 pin analog input, satu buah pin osilator Kristal 16 MHz, satu buah power jack, satu buah koneksi USB, satu buah tombol reset dan satu buah ICSP header.

Didalam Arduino UNO terdapat semua yang diperlukan untuk kebutuhan mikrokontroller, memiliki kemudahan dalam menghubungkan ke sebuah personal komputer menggunakan kabel USB atau mensuplai dengan adaptor AC ke DC atau menggunakan baterai untuk memulai proses [15]. Menurut [16], Arduino adalah sistem elektronik yang bersifat terbuka yang fleksibel dan sangat mudah diterapkan baik dari sisi perangkat lunaknya maupun perangkat keras.



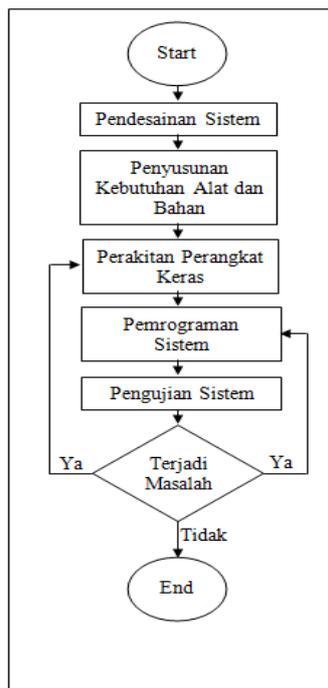
Gambar-3. Arduino UNO

3. HASIL DAN PEMBAHASAN

Pada bagian ini akan dipaparkan tentang pustaka penunjang penelitian dan tahapan-tahapan secara detail yang dilalui dalam penelitian sampai dihasilkan sebuah sistem kendali kontrol pada sepeda motor matik.

3.1 Flowchart Prosedur Penelitian

Prosedur penelitian yang dilakukan oleh peneliti dapat digambarkan 3. Pada gambar-3 dapat dijelaskan, awalnya pendesainan sistem yang terdiri dari pendesainan perangkat keras dan perangkat lunak setelah itu melakukan penyusunan/pembelian *hardware* dan peng-*install*-an program aplikasi, jika sistem sudah disusun maka sistem dapat dirakit *hardware*-nya dan pengkodean atau pemrograman sistem, setelah itu dilakuakn pengujian sistem, jika pada saat pengujian terdapat bug maka diperlukan pengecekan pada perakitan perangkat keras dan pemrograman sistem, jika sistem sudah tidak ada bug yang tersisa maka sistem sudah selesai dibuat.



Gambar-4. Diagram Alir Prosedur Penelitian

3.2 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras dalam perancangan sistem ini didasarkan pada kebutuhan penggunaan dan fungsi pada sistem yang dibuat ini, kebutuhan perangkat keras dapat dilihat pada tabel -1

Tabel -1. Daftar perangkat keras

No	Perangkat keras	Keterangan
1	ESP32	ESP32 sebagai perangkat utama yaitu perangkat yang berfungsi sebagai pemroses d inti dan alat untuk penyambungan wifi yang akan disambungkan dengan smartphone.
2	Kabel Jumper	Kabel Jumper berfungsi sebagai saluran data-data digital dan sebagai saluran listrik yang digunakan oleh semua perangkat keras yang terdapat pada sistem ini
3	Relay	Relay berfungsi sebagai saklar elektronik yang bergantung pada data digital untuk menghidupkan dan mematikan saklar, yang biasanya data digital berbentuk <i>HIGH</i> dan <i>LOW</i> , pada sistem ini relay yang digunakan merupakan relay dengan 2 channel yaitu channel-1 digunakan untuk kabel ke kunci dan channel-2 digunakan untuk kabel ke starter sepeda.
4	Buzzer	Buzzer digunakan sebagai alarm
5	Buck Converter Step Down	Buck Converter Step Down digunakan untuk penurun tegangan dari aki yang semulanya 12v agar diturunkan jadi 5v

3.3 Kebutuhan Perangkat Lunak

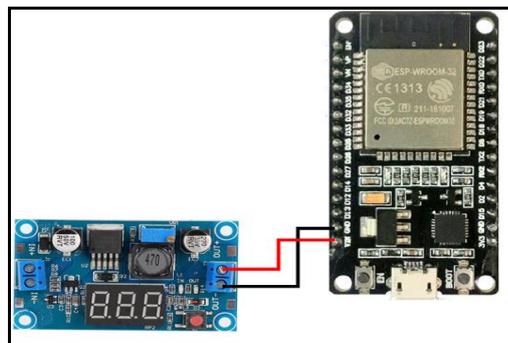
Perangkat lunak yang digunakan untuk membuat program pada *board* arduino ESP32 yaitu arduino IDE. Lalu untuk pengendalian sistem melalui asisten google, perangkat lunak yang digunakan yaitu Blynk dan IFTTT dan untuk pengendalian biasa dalam bentuk *interface* tombol-tombol dapat diakses melalui web browser yang telah terpasang pada semua smartphone android dengan memasukkan IP (*Internet Protocol*) pada *search bar* atau dapat juga dengan menekan ikon aplikasi pada beranda *smartphone* dengan mengatur pada aplikasi Google Chrome agar halaman web dapat cepat ditampilkan. Untuk lebih lengkapnya penjelasan dapat dilihat pada tabel -2.

Tabel -2. Daftar perangkat lunak

No	Perangkat keras	Keterangan
1	Arduino IDE	Perangkat lunak Arduino IDE digunakan untuk mengetik <i>sketch</i> dan digunakan sebagai sarana mengunggah <i>sketch</i> yang sudah jadi ke dalam perangkat keras ESP32.
2	Blynk	Perangkat lunak Blynk digunakan sebagai jembatan antara ESP32 dengan internet maka jika ESP32 diperintah oleh Pengemudi melewati Asisten Google maka perangkat lunak IFTTT akan merespon dan akan memerintah Blynk untuk mengontrol ESP32 dan ESP32 akan menjalankannya.
3	IFTTT	Perangkat lunak IFTTT digunakan untuk menjembatani antara Asisten Google dengan Blynk.
4	Android	Perangkat Android digunakan untuk memberi perintah melewati Asisten Google dan digunakan untuk membuka browser agar pengemudi dapat mengendalikan sistem kontrol.

3.4 Perakitan Perangkat Keras

Setelah perangkat keras sudah terkumpul semua dan siap untuk dirakit. Pertama-tama untuk perakitan, pasang kabel jumper pada Mikrokontroler ESP32 dibagian pin GND dan ujung satunya dipasang pada *Buck Converter Step Down* dibagian OUT-, lalu siapkan lagi kabel jumper untuk dipasang pada Mikrokontroler ESP32 dibagian pin VIN dan ujung satunya dipasang pada *Buck Converter Step Down* dibagian OUT+. Hasil proses ini ditunjukkan pada gambar-5.

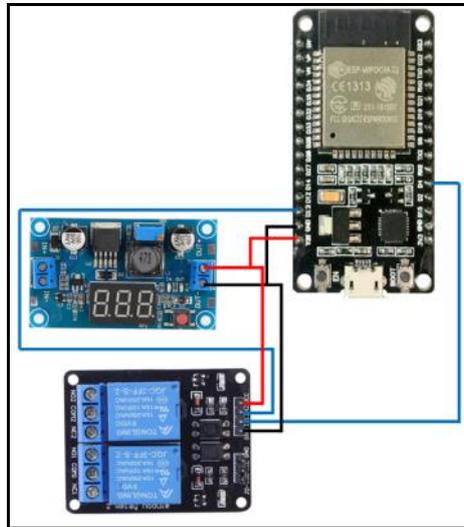


Gambar-5. Sambungan Esp32 Ke Buck Converter Step Down

Tahap perakitan bagian selanjutnya, pasang kabel jumper pada Mikrokontroler ESP32 dibagian pin D4 dan ujung satunya dipasang pada Relay dibagian IN1. Lalu siapkan lagi kabel jumper dan pasang kabel jumper tersebut pada Mikrokontroler ESP32 dibagian pin D13 dan ujung satunya dipasang pada Relay dibagian IN2. Langkah berikutnya, persiapkan sambungan kabel jumper dari langkah sebelumnya yaitu pada sambungan Mikrokontroler ESP32 GND ke Buck Converter Step Down OUT-, kupas pelindung karet pada kabel jumper tersebut sampai tembaga bagian dalam kabel jumper terlihat, siapkan kabel jumper lain dan potong bagian kepala ujung kabel jumper agar dapat disatukan dengan kabel Mikrokontroler ESP32 GND ke Buck Converter Step Down OUT-, dan bagian ujung satunya dipasang pada Relay dibagian GND.

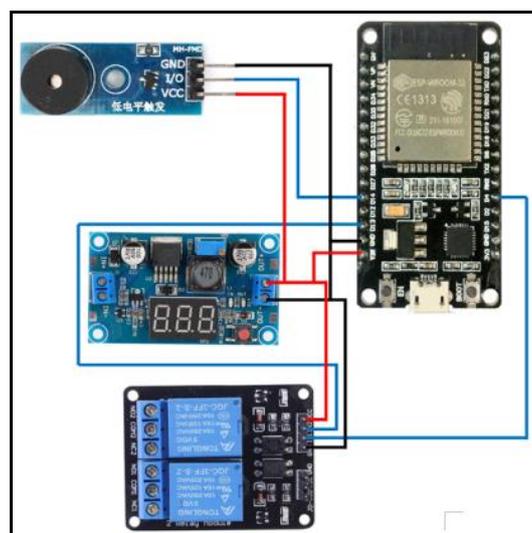
Proses selanjutnya ini hampir sama, persiapkan sambungan jumper dari langkah sebelumnya yaitu pada sambungan Mikrokontroler ESP32 VIN ke *Buck Converter Step Down* OUT+, kupas pelindung karet pada kabel jumper sampai tembaga bagian dalam jumper terlihat, setelah itu siapkan kabel jumper lain dan potong

bagian kepala ujung kabel jumper agar dapat disatukan dengan kabel Mikrokontroler ESP32 VIN ke ke *Buck Converter Step Down* OUT+, dan pasang bagian ujung satunya pada Relay dibagian VCC. Hasil kegiatan ini diperlihatkan dalam gambar-6.



Gambar-6. Sambungan Esp32 Ke Relay

Selanjutnya perakitan bagian Buzzer, pada bagian pemasangan pin VCC dan GND pada Buzzer langkah-langkahnya sama dengan pemasangan pada VCC dan GND pada Relay. Untuk langkah-langkah pemasangan kabel jumper pada Buzzer VCC, pasang kabel jumper pada Buzzer dibagian VCC, lalu potong kepala ujung lain kabel jumper agar dapat disatukan dengan sambungan pada Mikrokontroler ESP32 VIN ke *Buck Converter Step Down* OUT+. Proses pemasangan pada Buzzer GND langkah-langkahnya juga hampir sama dengan pemasangan Buzzer VCC, yaitu pasang kabel jumper pada pin GND pada Buzzer dan potong kepala kabel jumper pada ujung satunya kemudian sambungkan kabel yang sudah dipotong tersebut pada sambungan Mikrokontroler ESP32 GND ke *Buck Converter Step Down* OUT-. Berikut ini gambaran hasil proses praktikan.

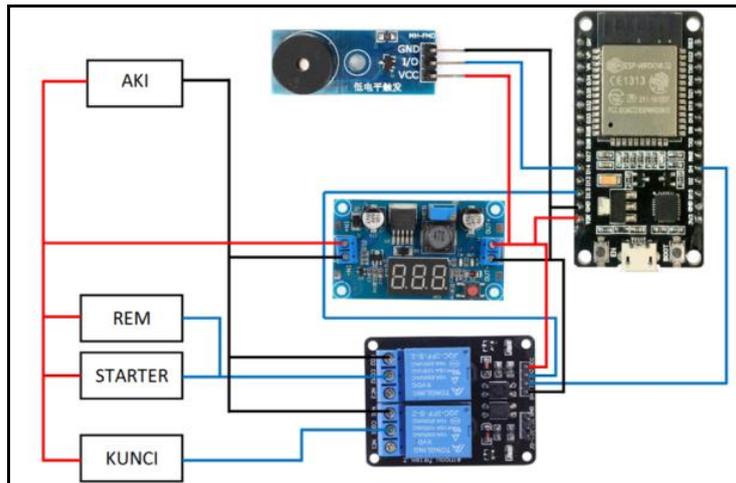


Gambar-7. Sambungan Esp32 Ke Buzzer

Setelah pemasangan komponen perangkat keras ke ESP32 telah selesai, untuk langkah selanjutnya adalah pemasangan komponen perangkat keras tersebut kedalam komponen sepeda motor matic Honda seperti BeAT, Scoopy dan Vario, yaitu dengan pemasangan *Buck Converter Step Down* IN+ dan *Buck Converter Step Down* IN- pada Aki, agar aliran listrik dari aki yang awalnya bertegangan 12v dapat diturunkan menjadi 5v, lalu aliran listrik dapat dialirkan ke perangkat keras yang lain. Apabila sambungan Aki ke *Buck Converter Step Down* telah dipasang, langkah selanjutnya memutus sambungan aliran listrik

sistem Kunci sepeda motor ke Aki dan menyambungkannya ke Relay dibagian COM1 dan NO1. Dan untuk sambungan aliran listrik sistem Rem dan Starter juga hampir sama seperti sambungan Kunci tapi bedanya setelah memutus sambungan aliran listrik sistem Rem dan Starter, ujung kabel Rem dan Starter yang telah terputus tersebut digabungkan. Fungsi dari penggabungan kabel ini adalah agar saat sepeda motor distarter menggunakan sistem web dan asisten google, em sepeda motor juga akan ikut aktif.

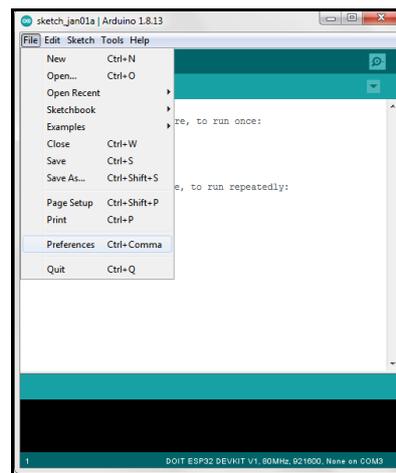
Setelah kabel Rem dan Starter menyatu, selanjutnya kabel gabungan tersebut dapat disambungkan ke Relay dibagian COM2 dan untuk kebel dari Aki yang semula tersambung ke rem dan starter sekarang disambungkan ke Relay dibagian NO1. Setelah selesai hasilnya dapat dilihat seperti gambar-8.



Gambar-8. Sambungan Aki, Rem, Starter Dan Kunci Ke Sistem Mikrokontroler

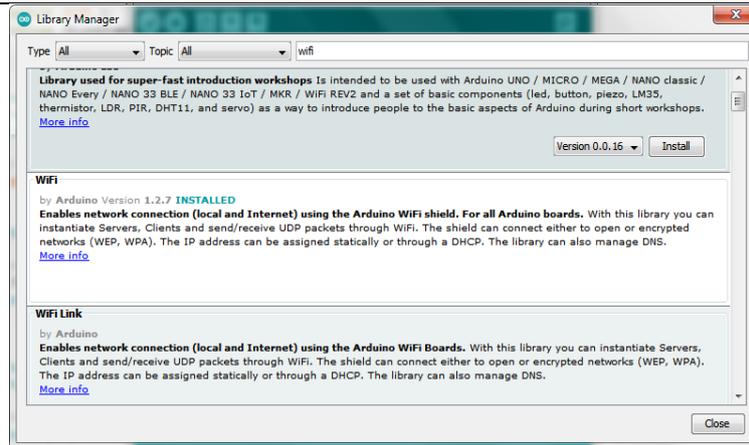
3.5 Pemrograman Sistem

Pemrograman pada Mikrokontroler ESP32 dilakukan didalam Arduino IDE, langkah awal dan persiapan sebelum melakukan pemrograman di Arduino IDE yaitu melakukan pemasangan atau peng-*instal*-an ESP32 kedalam Arduino IDE dengan cara membuka Arduino IDE lalu membuka *File -> Preference* dilihat seperti gambar-9.



Gambar-9. Arduino IDE Dengan Ribbon Tab File

Didalam *manage libraries* ketik *wifi* maka akan muncul beberapa *library* yang bersangkutan dengan WiFi, cari *library* yang bernama WiFi dan klik tombol *install*, setelah menunggu beberapa detik *library* WiFi akan ter-*install*. Hasilnya dapat dilihat pada gambar-10



Gambar-10. Manage Libraries

3.6. Kode Program Variabel Pin Pada Esp32

Variabel dibawah menampung data pin yang digunakan untuk memerintah Relay yang digunakan untuk mematikan dan menghidupkan starter dan kunci. Contohnya seperti gambar-11.

```

21 String output13State = "on";
22 String output4State = "on";
23
24 //-----
25
26 const int output13 = 13;//kunci
27 const int output4 = 4;//starter
28 const int output14 = 14;//buzzer
    
```

Gambar-11. Kode Program Variabel Pin ESP32

3.7. Kode Program Untuk Menetapkan Pin

Kode program dibawah ini digunakan untuk menetapkan pin yang ditetapkan sebagai *output*/keluaran. Contohnya seperti gambar-12

```

pinMode(output13, OUTPUT);// pin kunci
pinMode(output4, OUTPUT);// pin starter
pinMode(output14, OUTPUT);// pin buzzer
    
```

Gambar-12. Kode Program Penetapan ESP32

3.8. Kode program fungsi Auto WiFi Cut

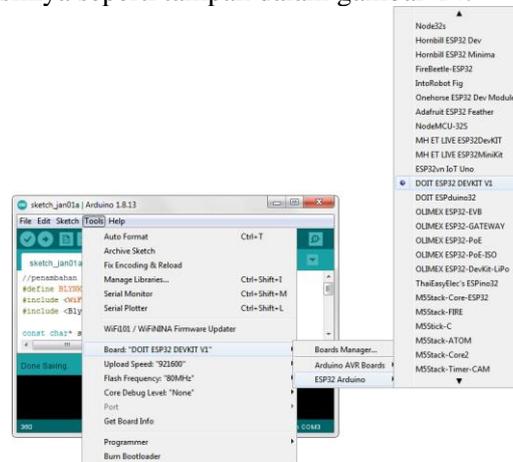
Kode program yang tersaji dalam gambar-12 merupakan kode program yang dijalankan jika sambungan WiFi antara *smartphone* dengan ESP32 terputus. Pengaturan *default* yang dijalankan yaitu dengan variabel (off_wifi_cut=0)(pengaturautowificut="off")(output13State="on") sehingga keluaran yang dihasilkan yaitu saat WiFi dimatikan maka sepeda motor tidak akan otomatis mati.

```

125 if(off_wifi_cut == 1 && pengaturautowificut=="on"){ //jika wifi cut hidup
126   Serial.println("GPIO 13 on");
127   output13State = "on";
128   digitalWrite(output13, HIGH);
129
130 }else if(off_wifi_cut == 0 && pengaturautowificut=="off"){ //jika wifi cut mati
131
132   if(output13State=="off"){ //jika motor hidup maka motor tetap hidup
133     Serial.println("GPIO 13 off");
134     output13State = "off";
135     digitalWrite(output13, LOW);
136   }else if(output13State=="on"){ //jika motor mati maka motor tetap mati
137     Serial.println("GPIO 13 on");
138     output13State = "on";
139     output4State = "on";
140     digitalWrite(output13, HIGH);
141     digitalWrite(output4, HIGH);
142   }
143 }
    
```

Gambar-13 Kode Program WiFi *Disconnected*

Setelah semua kode program selesai diketik, selanjutnya adalah peng-*upload*-an kode program kedalam ESP32, tapi sebelum itu harus dipilih dulu *Board* yang akan diupload klik *tolls-> Boards->ESP32 Arduino->DOIT ESP32 DEVKIT V1*. Hasilnya seperti tampak dalam gambar-14.



Gambar-14 Pengaturan *Board*

3.9. Pengujian Sistem

Pada pengujian sistem ini motor matik Honda yang telah dipasangi sistem kunci mikrokontroler diuji, untuk mengetahui apakah sistem sudah berjalan normal dan tidak ada bug yang dapat mempengaruhi kinerja sistem maka diperlukan pengujian sistem, pengujian sistem yang baik merupakan pengujian yang menyeluruh, baik pengujian *hardware* dan pengujian *software*, pengujian kinerja batas maksimal sistem.

a. Pengujian Hardware

Pengujian *hardware* diperlukan untuk mengetahui apakah ada kerusakan hardware dan apakah ada kesalahan dalam pemasangan kabel jumper. Dibawah ini merupakan contoh jika ada *hardware* yang rusak maka beberapa *hardware* yang lain akan tidak bisa digunakan.

Tabel -3. Probabilitas masalah kerusakan *hardware*

	Kabel	B.C.S.D	ESP32	Relay	Buzzer
	Rusak	Rusak	Rusak	Rusak	Rusak
Kabel	Rusak	X	X	X	X
B.C.S.D	✘	Rusak			
ESP32	✘	X	Rusak		
Relay	✘	X	X	Rusak	
Buzzer	✘	X	X		Rusak

Dari tabel-3 dapat diberikan penjelasan sebagai berikut :

- 1) Jika Kabel rusak ada kemungkinan beberapa hardware tidak akan bisa menyala tergantung kabel mana yang rusak.
- 2) Jika B.C.S.D atau Buck Converter Step Down rusak maka semua hardware tidak akan bisa menyala.
- 3) Jika ESP32 rusak maka Relay dan Buzzer tidak bisa digunakan.
- 4) Jika Relay rusak maka Relay tidak akan bisa merespon data dari ESP3 dan tidak bisa digunakan untuk menyalakan motor.
- 5) Jika Buzzer rusak maka Buzzer tidak bisa merespon data dari ESP32

b. Pengujian Software

Pengujian *software* diperlukan untuk mengetahui apakah ada kesalahan pemrograman, kesalahan keluaran tidak setara dengan masukan. Tabel-4 merupakan contoh hasil pengujian perangkat lunak (*software*).

Tabel -4. Pengujian software

Tes Uji	Cara Pengujian	Hasil yang diharapkan	Keterangan
Start up awal	Menyalakan B.C.S.D dengan mengalirkan listrik dari aki	Sistem aman, relay tidak klik dan buzzer tidak berbunyi aneh	Uji apakah ada masalah saat penyalakan awal
Penyambungan WiFi	Menyalakan <i>hotspot</i> pada <i>smartphone</i>	ESP32 otomatis tersambung dengan WiFi <i>smartphone</i>	Uji apakah bisa ESP32 dalam penyambungan WiFi
Cek IP	Setelah tersambung pada WiFi, mengecek IP yang didapatkan pada serial monitor	ESP32 mendapatkan IP yang sudah ditetapkan	Uji apakah bisa ESP32 mendapatkan alamat IP yang sudah ditetapkan
Konektifitas internet	Setelah tersambung pada WiFi, mengecek koneksi internet pada serial monitor	Tersambung dengan internet	Uji apakah bisa tersambung ke internet
Konektifitas Blynk	Setelah tersambung pada WiFi, mengecek koneksi Blynk pada serial monitor	Tersambung dengan Blynk	Uji apakah bisa tersambung ke Blynk
Halaman web	Setelah tersambung pada WiFi, membuka halaman dengan mengetik IP ESP32 pada browser atau bisa menekan ikon aplikasi pada beranda jika sudah di- <i>setting</i> pada Chrome	Halaman web bisa dibuka	Uji apakah halaman web bisa dibuka
Tampilan web	Setelah halaman web terbuka, mengecek seluruh tampilan web	Halaman web tidak ada kerusakan pada pemrograman CSS	Uji apakah tampilan web sama dengan CSS
Tombol nyalakan motor	Setelah halaman web terbuka, menekan tombol menghidupkan motor	Motor hidup setelah tombol ditekan	Uji tombol apakah bisa menyalakan motor
Tombol matikan motor	Setelah halaman web terbuka, menekan tombol mematikan motor	Motor mati setelah tombol ditekan	Uji tombol apakah bisa mematikan motor
Tombol starter motor	Setelah halaman web terbuka, menekan tombol starter motor	Starter motor hidup setelah tombol ditekan	Uji tombol apakah bisa menstarter motor
Tombol starter mati	Setelah halaman web terbuka, menekan tombol starter mati	Starter motor mati setelah tombol ditekan	Uji tombol apakah bisa mematikan starter motor
Google asisten nyalakan	Setelah halaman asisten google tertampil, mengatakan "hidupkan motor" atau "motor hidup"	Asisten google merespon dan motor dihidupkan	Uji apakah google asisten merespon perintah
Google asisten matikan	Setelah halaman asisten google tertampil, mengatakan "matikan motor" atau "motor mati"	Asisten google merespon dan motor mati	Uji apakah google asisten merespon perintah
Google asisten starter	Setelah halaman asisten google tertampil, mengatakan "starter motor" atau "motor start"	Asisten google merespon dan motor terstarter	Uji apakah google asisten merespon perintah
Pemutusan WiFi 1	Mematikan hotspot pada <i>smartphone</i>	Motor mati saat sambungan WiFi putus	Uji apakah saat WiFi terputus motor dapat mati jika pengaturan wifi cut hidup
Pemutusan WiFi 2	Mematikan hotspot pada <i>smartphone</i>	Motor tetap hidup saat sambungan WiFi putus	Uji apakah saat WiFi terputus motor tetap hidup jika pengaturan wifi cut mati
Pemutusan WiFi 3	Mematikan hotspot pada <i>smartphone</i>	Buzzer berbunyi	Uji apakah Buzzer akan berbunyi jika WiFi terputus

c. Pengujian Kinerja Batas Maksimal

Pengujian kinerja batas maksimal sistem digunakan untuk mengetahui bagaimana batasan sistem saat digunakan. Pengujian ini dibuat untuk memastikan saat sistem ini dihidupkan apakah ESP32 dapat

terkoneksi dengan WiFi dalam jarak tertentu. Disini WiFi yang digunakan bersumber dari *hotspot smartphone*. Pengujian ini dilakukan untuk mengetahui batas kinerja ESP32. Hasil pengujian terhadap ESP32 yang telah dilakukan disajikan pada tabel-5.

Tabel -5. Pengujian batas maksimal konektivitas wifi

Jarak dengan Smartphone (meter)	Hasil Koneksi	
	Diluar	Didalam rumah
8	√	√
9	√	√
10	√	√
11	√	√
12	√	√
13	√	√
14	√	√
15	√	X
16	√	X
17	X	X

4. KESIMPULAN

Berdasarkan hasil pengujian sistem dan implementasi yang dilakukan maka dapat ditarik kesimpulan bahwa penggunaan teknologi dari *Internet of Things* ini dapat diimplementasikan di berbagai sisi kebutuhan manusia. Salah satunya pemanfaatan teknologi *Internet of Things* untuk keamanan kendaraan bermotor. Melalui pengujian dalam penelitian ini dapat dilihat bahwa *Internet of Things* yang diterapkan pada aplikasi android berbasis arduino UNO mampu melakukan proses pengendalian sepeda motor baik untuk menyalakan maupun mematikan mesin dengan jarak-jarak tertentu sesuai dengan yang diharapkan. Penelitian ini menghasilkan sistem kendali untuk menyalakan maupun mematikan mesin motor secara otomatis menggunakan android. Maka untuk pengembangan selanjutnya dapat ditambahkan fitur deteksi sensor getar pada sepeda motor, yang gunanya untuk mendeteksi apakah sepeda motor sedang digerakkan oleh orang lain pada saat motor belum terkoneksi ke WiFi *smartphone*,

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM) Universitas Dian Nuswantoro yang telah mendukung penuh penulis dalam mengikuti agenda seminar nasional ini.

DAFTAR PUSTAKA

- [1] Arafat, "Sistem pengamanan pintu rumah berbasis internet of things (iot) dengan esp8266", *Technologia*, vol. 7, no. 4, pp. 262-268, 2016.
- [2] Y. Efendi, "Internet Of Things (Iot) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile", *Jurnal Ilmiah Ilmu Komputer*, vol. 4, no. 1, pp. 19-26, ISSN. 2442-4512, 2018.
- [3] Sumardi, "Perancangan Sistem Starter Sepeda Motor Menggunakan Aplikasi Android Berbasis Arduino Uno", *Prosiding Seminar Ilmu Komputer dan Teknologi Informasi*, vol. 2, no. 1, pp. 151-162, ISSN. 2540-7902, 2017.
- [4] R. Khana and U. Usnul, "Rancang Bangun Sistem Keamanan Rumah Berbasis Internet of Things Dengan Platform Android," *Jurnal Kajian Teknik Elektro*, vol. 3, no. 1, pp. 18–31, 2018.
- [5] A. Putra, D. Romahadi, "Sistem Keamanan Sepeda Motor Berbasis Internet Of Things (Iot) Dengan Smartphone Menggunakan Nodemcu", *Jurnal Teknologi Terpadu*, vol. 1, no. 9, pp. 77-87, ISSN. 2338 – 66649, 2021.
- [6] T. Juwariyah, D. Widiyanto S. Sulasmingsih, "Purwa Rupa Sistem Pengaman Sepeda Motor Berbasis IoT (Internet of Things)", *J.Oto.Ktrl.Inst (J.Auto.Ctrl.Inst)*, vol. 11, no. 1, pp. 49-57, ISSN. 2085-2517, 2019.

- [7] R. Karim, “Pentingnya Penggunaan Jaringan Wi-Fi Dalam Memenuhi Kebutuhan Informasi Pemustaka Pada Kantor Perpustakaan Dan Kearsipan Daerah Kota Tidore Kepulauan”, *e-journalActa Diurna*, vol. 5, no. 2, 2016.
- [8] R. Priantama, “Efektivitas Wifi Dalam Menunjang Proses Pendidikan Bagi Lembaga Perguruan Tinggi (Studi Kasus Terhadap Mahasiswa Pengguna Di Lingkungan Universitas Kuningan)”, *Jurnal Cloud Information*, vol. 1, no. 1, pp. 22-28, ISSN. 2527-5224, 2020
- [9] R. Corputty, Muriani, Y. Kolyaan, “Interworking Wimax Dan Wifi”, *Jurnal Teknologi Informasi (JTI)*, vol. 5, no. 2, pp. 38-50, ISSN. 2338 – 1434, 2017.
- [10] M. Ichwan, F. Hakiky, “Pengukuran Kinerja Goodreads Application Programming Interface (Api) Pada Aplikasi Mobile Android (Studi Kasus Untuk Pencarian Data Buku)”, *Jurnal Informatika*, vol. 2, no. 2, pp. 13-21, 2011.
- [11] H. Kusniyati, N.S.P. Sitanggang, “Aplikasi Edukasi Budaya Toba Samosir Berbasis Android”, *Jurnal Teknik Informatika*, vol. 9, no. 1, pp. 9-18, ISSN 1979-9160, 2016.
- [12] L. Safitri, S. Basuki, “Analisa Dan Perancangan Sistem Informasi Text Chatting Berbasis Android Web View”, *Jurnal Ipsikom*, vol. 8, no. 2, pp. 1-5, ISSN. 2338-4093, 2020.
- [13] Saptaji, ”Mudahbelajar Mikrokontroller dengan Arduino”. Bandung :Widya Media, 2015
- [14] F. Djuandi, “PengenalanArduino” : (<https://tobuku.com>) diakses 21 Juni 2015
- [15] A. Adriansyah, O. Hidayatama, “Rancang Bangun Prototipe Elevator Menggunakan Microcontroller Arduino Atmega 328p”, *Jurnal Teknologi Elektro*, vol. 4, no. 3, pp. 100-112, ISSN. 2086-9479, 2013.
- [16] P.Y.M Bate , A.S. Wiguna, D.A. Nugraha, “Sistem Penjemuran Otomatis Menggunakan Arduino Uno R3 Dengan Pendekatan Metode Fuzzy”, *KURAWAL Jurnal Teknologi, Informasi dan Industri*, vol. 3, no. 1, pp. 81-92, ISSN. 2615-6474, 2020.