

Implementasi *Internet of Things* pada *Monitoring Kecepatan Kendaraan Bermotor*

Christianto Wibisono Darmawan, Sherwin R. U. A. Sompie, Feisy D. Kambey
Jurusan Teknik Elektro, Universitas Sam Ratulangi Manado, Jl. Kampus Bahu, 95115, Indonesia
wibid26@gmail.com, aldo@unsrat.ac.id, feisykambey@unsrat.ac.id

Diterima: tgl; direvisi: tgl; disetujui: tgl

Abstract — *Internet of Things (IoT) is a concept in which an object is connected to the internet network. These days, the amount of motor vehicle users in Indonesia is increasing. The accident rate has also increase along with the increase in motor vehicle users. Most of the traffic accidents involved motorcycles, and predominantly occurred in student ages. The lack of parental role in watching over and educating their teenagers in riding motorcycle have become one of many causes of traffic accidents. Therefore, a prototype and an application had been made to help parents to monitor the motor vehicle users while riding the motorcycle. In this study, the application was made with Blynk platform and ThingSpeak platform and the prototype was made with NodeMCU ESP8266 microcontroller and GPS uBlox Neo 6M module. The result of this prototype experiment is able to integrate two platforms of IoT and able to remotely monitor the speed of motorcycle when stably connected to the internet and the prototype is outdoor.*

Keywords — *ESP8266; Global Positioning System (GPS); Internet of Things (IoT); NEO 6M; Speed Monitoring System).*

Abstrak — *Internet of Things (IoT) merupakan suatu konsep dimana suatu objek yang terhubung dengan jaringan internet. Dewasa ini jumlah pemilik kendaraan bermotor di Indonesia semakin bertambah. Seiring dengan bertambahnya pemilik kendaraan bermotor, tingkat kecelakaan juga semakin meningkat. Sebagian besar kecelakaan lalu lintas melibatkan sepeda motor, serta didominasi usia pelajar. Kurangnya peran orang tua dalam mengawasi dan mengedukasi anak remaja dalam mengendarai sepeda motor menjadi salah satu penyebab kecelakaan. Untuk itu dibuat sebuah prototipe dan aplikasi untuk membantu orang tua untuk memantau pengemudi kendaraan bermotor dalam berkendara. Pada penelitian ini aplikasi dibuat menggunakan platform Blynk dan ThingSpeak. Prototipe dibuat menggunakan mikrokontroler NodeMCU ESP8266 dan modul GPS uBlox Neo 6M. Hasil dari pengujian prototipe ini dapat mengintegrasikan dua platform IoT dan dapat memantau kecepatan kendaraan bermotor dari jarak jauh ketika koneksi internet berjalan dengan stabil serta prototipe berada di luar ruangan (outdoor).*

Kata kunci — *ESP8266; Global Positioning System (GPS); Internet of Things (IoT); NEO 6M ; Sistem Monitoring Kecepatan..*

I. PENDAHULUAN

Dewasa ini jumlah pemilik kendaraan bermotor di Indonesia semakin bertambah. Seiring dengan bertambahnya pemilik kendaraan bermotor, tingkat kecelakaan lalu lintas di Indonesia juga semakin meningkat. Tiga sampai empat orang meninggal setiap jam akibat kecelakaan lalu lintas. Sebagian besar kecelakaan lalu lintas melibatkan sepeda motor, serta didominasi usia pelajar.

Usia 17 tahun merupakan saat yang ditunggu-tunggu bagi

para remaja. Karena pada usia tersebut para remaja sudah diperbolehkan untuk memiliki Surat Izin Mengemudi (SIM) secara legal. Akan tetapi yang sangat disayangkan, banyak kecelakaan lalu lintas yang disebabkan oleh pengendara usia remaja.

Kepala Seksi Kemitraan Korps Lalu Lintas Kepolisian Republik Indonesia (Korlantas Polri), AKBP Aldo Siahaan, mengatakan, sebanyak 24.023 pelajar dan mahasiswa terlibat kecelakaan lalu lintas selama Januari-Mei 2017 di seluruh Indonesia [1]. Pelajar dan mahasiswa menjadi penyumbang tertinggi kecelakaan lalu lintas dibanding profesi lain. Data lain Korlantas Polri menunjukkan bahwa dari April-Juni 2017, mereka yang terlibat kecelakaan lalu lintas mencapai lebih dari 25 ribu insiden. Adapun 34,8 persen diantaranya melibatkan pengemudi/pengendara berusia 15-24 tahun. Aldo juga mengatakan angka kecelakaan terbanyak terjadi di jam-jam pergi dan pulang sekolah.

Berdasarkan riset yang dilakukan oleh *National Institute of Health (NIH)*, ditemukan hal yang menarik bahwa ternyata ada bagian dalam otak yang berhubungan dengan kemampuan untuk mengukur resiko dan mengontrol perilaku impulsive, bagian tersebut baru akan berkembang sepenuhnya ketika seseorang telah mencapai usia 25 tahun [2]. Hal tersebut bila dikaitkan dengan perilaku remaja seringkali membuat mereka melupakan resiko yang akan terjadi atas perilaku yang mereka lakukan. Salah satu contoh perilaku tersebut mengendarai kendaraan secara sembrono dan melewati batas kecepatan aman berkendara.

Melarang remaja untuk mengemudi bukanlah jalan keluar untuk mengurangi jumlah kecelakaan lalu lintas. Peran orang tua disini sangat dibutuhkan untuk mengawasi dan mengedukasi remaja agar mengendarai kendaraan secara benar dan menaati peraturan yang ada. Oleh karena itu untuk mengurangi jumlah kecelakaan dan melakukan pengawasan, dibutuhkan sebuah sistem *monitoring* kecepatan kendaraan bermotor.

Suatu alat yang mudah dikenali oleh internet dan setiap saat terkoneksi dengan internet yang dikenal dengan istilah *Internet of Things (IoT)*. *Internet of Things* mengacu pada pengidentifikasian suatu objek yang direpresentasikan secara virtual di dunia maya atau Internet. Jadi dapat dikatakan bahwa *Internet of Things* adalah bagaimana suatu objek yang nyata di dunia ini digambarkan di dunia maya (Internet). Metode yang digunakan oleh *Internet of Things* adalah nirkabel atau pengendalian secara otomatis tanpa mengenal jarak.

Dengan permasalahan yang ada penulis akan membuat

sebuah aplikasi *monitoring* kecepatan kendaraan bermotor yang dapat membantu orang tua atau pihak terkait untuk mengawasi pengemudi kendaraan bermotor dengan mengimplementasikan *Internet of Things*.

A. *Internet of Things (IoT)*

Internet of Things (IoT) adalah sebuah konsep/skenario dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer [3]. *IoT* telah berkembang dari konvergensi teknologi nirkabel, *micro-electromechanical systems (MEMS)*, dan *Internet. "A Things"* pada *Internet of Things* dapat didefinisikan sebagai subjek misalkan orang dengan monitor implant jantung, hewan peternakan dengan *transponder biochip*, sebuah mobil yang telah dilengkapi *built-in sensor* untuk memperingatkan pengemudi ketika tekanan ban rendah. Sejauh ini, *IoT* paling erat hubungannya dengan komunikasi *machine-to-machine (M2M)* di bidang manufaktur dan listrik, perminyakan, dan gas. Produk dibangun dengan kemampuan komunikasi *M2M* yang sering disebut dengan sistem cerdas atau "*smart*". (Contoh: *smart label, smart meter, smart grid sensor*). Meskipun konsep ini kurang populer hingga tahun 1999, namun *IoT* telah dikembangkan selama beberapa dekade. Alat Internet pertama, misalnya, adalah mesin Coke di Carnegie Mellon University di awal 1980-an. Para programmer dapat terhubung ke mesin melalui Internet, memeriksa status mesin dan menentukan apakah ada atau tidak minuman dingin yang menunggu mereka, tanpa harus pergi ke mesin tersebut [4]. Istilah *IoT (Internet of Things)* mulai dikenal tahun 1999 yang saat itu disebutkan pertama kalinya dalam sebuah presentasi oleh Kevin Ashton, *cofounder and executive director of the Auto-ID Center* di MIT.

B. *Monitoring*

Monitoring dapat didefinisikan sebagai suatu kegiatan yang dilakukan secara terus menerus dan bersifat utuh dari manajemen perusahaan yang isinya adalah penilaian yang bersifat sistimatis terhadap kemajuan suatu pekerjaan [5]. Dengan kata lain, kegiatan *monitoring* adalah suatu proses pencatatan dan pengumpulan informasi terhadap tugas-tugas proyek secara periodik. Selain itu, *monitoring* berguna untuk melihat dan memantau perkembangan suatu pekerjaan atau transaksi yang sedang berjalan.

C. *Kecepatan*

Kecepatan kendaraan adalah rata-rata jarak yang dapat ditempuh suatu kendaraan pada suatu ruas jalan dalam satu satuan waktu tertentu [6]. Kecepatan dari suatu kendaraan dipengaruhi oleh faktor-faktor manusia, kendaraan dan prasarana, serta dipengaruhi pula oleh arus lalu lintas, kondisi cuaca dan lingkungan alam sekitarnya. Dengan didapatnya waktu perjalanan dan jarak perjalanan maka kecepatan perjalanan dan kecepatan bergerak akan didapat. Sehingga, dapat dinyatakan dalam rumus (1) berikut:

$$V = s/t \quad (1)$$

Keterangan:

V = Kecepatan (km/jam, m/det).

s = Jarak yang ditempuh (km, m).

t = Waktu tempuh kendaraan (jam, det).

D. *Bahasa Pemrograman C++*

Bahasa C dan C++ merupakan bahasa yang sangat populer dalam dunia pengembangan perangkat lunak. Kedua bahasa ini digolongkan kedalam bahasa tingkat menengah. Semenjak dikembangkan, bahasa C dan C++ banyak digunakan untuk mengembangkan program-program aplikasi di bidang telekomunikasi *financial* atau bisnis dan sistem operasi. Bahkan sampai saat ini, pembuatan program-program untuk permainan komputer (*game*) sebagian besar masih menggunakan bahasa C/C++.

E. *Arduino IDE*

Dalam menuliskan kode sumber dibutuhkan Arduino IDE, dimana Arduino IDE ini merupakan program untuk menuliskan kode sumber ke dalam mikrokontroler arduino dan bahasa pemrogramannya sendiri merupakan penggabungan antara bahasa C dan Java dikarenakan struktur bahasa pemrograman dan penggunaan *library* yang mirip dengan C dan Java.

Software Arduino IDE terdiri dari 3 (tiga) bagian:

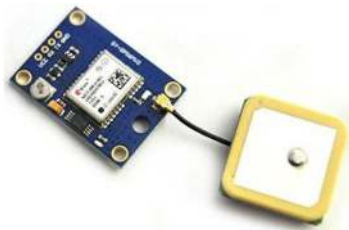
- 1) *Uploader*, modul yang berfungsi memasukan kode biner kedalam memori mikrokontroler.
- 2) Editor program, untuk menulis dan mengedit program. *Listing* program pada Arduino disebut *sketch*.
- 3) *Compiler*, modul yang berfungsi mengubah bahasa *processing* (kode program) kedalam kode biner karena kode biner adalah satu-satunya bahasa pemrograman yang dipahami oleh mikrokontroler.

Untuk struktur perintah pada arduino secara garis besar terdiri dari dua bagian yaitu *void setup* dan *void loop*. *Void setup* ini berisi perintah yang akan dieksekusi hanya satu kali sejak arduino dihidupkan sedangkan *void loop* berisi perintah yang akan dieksekusi berulang-ulang selama arduino dihidupkan.

F. *Global Positioning System (GPS)*

GPS atau *Global Positioning System* merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunaanya dimana dia berada (secara global) di permukaan bumi yang berbasis satelit [7]. Data dikirim dari satelit berupa sinyal radio berupa data digital. Dimanapun anda berada, maka *GPS* bisa membantu menunjukkan arah selama anda melihat langit. Layanan *GPS* ini tersedia gratis, bahkan tidak perlu mengeluarkan biaya apapun kecuali membeli *GPS receiver*-nya. *GPS* adalah sistem navigasi yang berbasis satelit yang saling berhubungan yang berada di orbitnya.

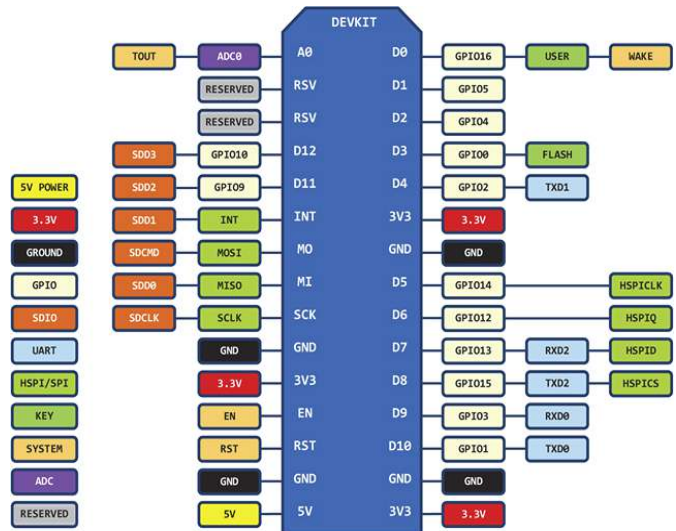
Perangkat *GPS* (biasanya merupakan perangkat khusus, bukan ponsel) menentukan lokasi dari minimal 3 satelit yang membentuk kawasan segitiga dengan mencari longitude, latitude, dan data lainnya yang diperlukan [8]. Hal ini membutuhkan waktu kira-kira 12 menit untuk menentukan lokasi ketika perangkat khusus *GPS* dinyalakan, cukup lama namun setidaknya pengguna tidak perlu memiliki akses ke operator ponsel untuk mengoperasikan *GPS*. Sedangkan *A-GPS* yang biasanya ditanamkan di ponsel dapat menemukan lokasi dengan lebih cepat, bahkan kurang dari 20 detik. Hal itu dikarenakan *A-GPS* langsung mencari satelit yang terdekat dengan lokasi ponsel saat itu melalui operator telekomunikasi. Hanya saja lokasi yang dilacak oleh *A-GPS* memang tidak seakurat *GPS*.



Gambar 1. GPS uBlox NEO 6M



Gambar 2. NodeMCU ESP8266



D0(GPI016) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

Gambar 3. Mapping Pin NodeMCU

G. GPS uBlox NEO 6M

Pada penelitian kali ini modul GPS yang digunakan adalah berjenis uBlox Neo 6M, jenis GPS ini cukup dapat diandalkan karena memiliki keakuratan yang cukup baik dan juga beberapa fitur yang cukup menguntungkan di antaranya terdapat baterai cadangan data, *built-in* elektronik kompas, dan *built-in* antena keramik untuk menangkap sinyal dengan kuat. Kemudian untuk dapat mengkomunikasikan GPS ini dengan Arduino diperlukan sebuah *library* yang bernama “TinyGPS++.h”. Bentuk dari modul GPS uBlox Neo 6M dapat dilihat pada Gambar 1.

H. NodeMCU ESP8266

NodeMCU adalah sebuah platform IoT yang bersifat *open source*. Terdiri dari perangkat keras berupa System On Chip ESP8266 dari ESP8266 buatan Espressif System [9].

NodeMCU bisa dianalogikan sebagai board arduino yang terkoneksi dengan ESP8266. NodeMCU telah *re-package* ESP8266 ke dalam sebuah board yang sudah terintegrasi dengan berbagai feature selayaknya *microcontroller* dan kapasitas ases terhadap wifi dan juga chip komunikasi yang berupa USB to serial. Sehingga dalam pemrograman hanya dibutuhkan kabel data USB.

Karena sumber utama dari NodeMCU adalah ESP8266 khususnya seri ESP-12 yang termasuk ESP-12E. Maka fitur – fitur yang dimiliki oleh NodeMCU akan lebih kurang serupa dengan ESP-12. Beberapa Fitur yang tersedia antara lain 10 Port GPIO dari D0 – D10, Fungsionalitas PWM, Antarmuka I2C dan SPI, Antarmuka 1 Wire, dan ADC. NodeMCU ESP8266 dan *mapping pin* dapat dilihat pada gambar 2 dan 3.

I. Blynk

Blynk adalah sebuah layanan server yang digunakan untuk mendukung project *Internet of Things*. Layanan server ini memiliki lingkungan mobile user baik Android maupun iOS. Blynk mendukung berbagai macam *hardware* yang dapat digunakan untuk project *Internet of Things*. Blynk adalah dashboard digital dengan fasilitas antarmuka grafis dalam pembuatan projectnya [10]. Penambahan komponen pada aplikasi Blynk dengan cara *Drag and Drop* sehingga memudahkan dalam penambahan komponen *Input/output* tanpa perlu kemampuan pemrograman Android maupun iOS.

Blynk diciptakan dengan tujuan untuk control dan *monitoring hardware* secara jarak jauh menggunakan komunikasi data internet ataupun intranet (jaringan LAN). Kemampuan untuk menyimpan data dan menampilkan data secara visual baik menggunakan angka, warna ataupun grafis semakin memudahkan dalam pembuatan project dibidang *Internet of Things*. Terdapat 3 komponen utama dalam Blynk:

1) Blynk Apps

Blynk Apps memungkinkan untuk membuat project interface dengan berbagai macam komponen *input output* yang mendukung untuk pengiriman maupun penerimaan data serta merepresentasikan data sesuai dengan komponen yang dipilih. Representasi data dapat berbentuk visual angka maupun grafik. Terdapat 4 jenis category komponen yang berdasar pada Aplikasi Blynk:

- a. *Controller* digunakan untuk mengirimkan data atau perintah ke *Hardware*.
- b. *Display* digunakan untuk menampilkan data yang berasal dari *hardware* ke smartphone.
- c. *Notification* digunakan untuk mengirim pesan dan notifikasi.
- d. *Interface* Pengaturan tampilan pada aplikasi Blynk dapat berupa menu ataupun tab.
- e. *Others* beberapa komponen yang tidak masuk dalam kategori sebelumnya diantaranya Bridge, RTC, Bluetooth.

2) Blynk Server

Blynk server merupakan fasilitas *Backend Service* berbasis *cloud* yang bertanggung jawab untuk mengatur komunikasi antara aplikasi smart phone dengan lingkungan *hardware*. Kemampuan untuk menangani puluhan *hardware* pada saat yang bersamaan semakin memudahkan bagi para pengembang sistem IoT. Blynk server juga tersedia dalam bentuk local server apabila digunakan pada lingkungan tanpa internet. *t* bersifat *open source* dan dapat diimplementasikan pada *Hardware* Raspberry Pi.

3) Blynk Library

Blynk Library dapat digunakan untuk membantu pengembangan code. Blynk library tersedia pada banyak platform perangkat keras sehingga semakin memudahkan para

pengembang *IoT* dengan fleksibilitas hardware yang didukung oleh lingkungan *Blynk*.

J. ThingSpeak

ThingSpeak merupakan sebuah layanan internet yang menyediakan layanan untuk pengaplikasian "*Internet of Things*". ThingSpeak merupakan layanan yang berisi aplikasi dan API yang bersifat open source untuk menyimpan dan mengambil data dari berbagai perangkat yang menggunakan HTTP (*Hypertext Transfer Protocol*) melalui Internet atau melalui LAN (*Local Area Network*) [11]. Dengan menggunakan ThingSpeak, seseorang dapat membuat aplikasi *logging sensor*, aplikasi pelacakan lokasi, dan jaringan sosial dari segala sesuatu yang terhubung ke internet dengan pembaruan status.

K. Fritzing

Fritzing merupakan *software* yang bersifat *open source* untuk merancang rangkaian elektronika [12]. Fritzing dikembangkan di University of Applied of Postdam. *Software* tersebut mendukung para penggemar elektronika untuk

membuat prototipe dengan merancang rangkaian berbasis abeltamikrokontroler Arduino. Memungkinkan para perancang elektronika pemula sekalipun untuk membuat layout PCB yang bersifat *custom*. Tampilan dan penjelasan yang ada pada Fritzing bisa dengan mudah dipahami oleh seseorang yang baru pertama kali menggunakannya.

Dalam perancangannya, Fritzing menggunakan tampilan *breadboard* sebagai prototipe penyusunan komponen elektronika. Beberapa komponen yang ada pada Fritzing mulai dari arduino, Raspberry Pi, berbagai sensor, voltage regulator, resistor, dan masih banyak lagi lainnya.

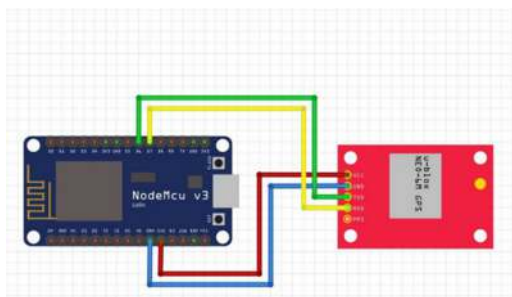
II. METODE PENELITIAN

A. *Observasi Pengumpulan Data*

Observasi ini merupakan metode pengumpulan data dengan mengamati langsung. Pada tahap ini akan dilakukan pengumpulan data dengan mengambil bukti beberapa foto gambar serta video dengan mengamati langsung keadaan di jalan seperti apa agar bisa dijadikan sampel nanti pada tahap uji coba. Dalam penelitian ini mengambil lokasi penelitian di Jalan Taratara-Kayawu, Kecamatan Tomohon Barat, Kota Tomohon Sulawesi Utara.

TABEL I
WIRING MIKROKONTROLER NODEMCU ESP8266 DENGAN MODUL GPS UBLOX NEO 6M

No.	Pin NodeMCU ESP8266	Pin Modul GPS
1.	3.3 Volt	VCC
2.	D7	RX
3.	D6	TX
4.	GND	GND



Gambar 4. Rangkaian modul *GPS* dengan mikrokontroler NodeMCU ESP8266



Gambar 5. Tampilan prototipe



Gambar 6. *Flowchart* aplikasi

B. Perancangan dan Pembuatan Perangkat Keras

Sebagai pengendali utama digunakan mikrokontroler NodeMCU dengan chip ESP8266 yang mendukung koneksi WiFi. Modul GPS yang digunakan yaitu GPS Neo 6M. Selain untuk menentukan lokasi, modul GPS dapat digunakan untuk mengukur kecepatan. Untuk menghubungkan NodeMCU ESP8266 dengan modul GPS, pin VCC pada modul dihubungkan dengan pin 3.3 volt pada mikrokontroler, pin RX pada modul di hubungkan dengan GPIO13 (D7) pada mikrokontroler, pin TX pada modul dihubungkan dengan pin GPIO12 (D6) pada mikrokontroler, dan terakhir pin GND pada modul dihubungkan dengan pin GND pada mikrokontroler. Pada table 1 adalah penyambungan kabel pada mikrokontroler NodeMCU ESP8266 dengan modul GPS. Rangkaian modul GPS dengan mikrokontroler NodeMCU ESP8266 dapat dilihat pada gambar 4.

Pembuatan prototipe dilakukan dengan *wiring* modul GPS dengan mikrokontroler NodeMCU ESP8266 (table I). Untuk sumber listrik menggunakan *Power Bank*. Setelah itu dilanjutkan dengan memasukkan ke tempat/wadah untuk prototipe. Wadah prototipe yang digunakan penulis pada penelitian ini yaitu wadah plastik yang berdimensi 13cm x 9cm x 6cm. Wadah tempat prototipe dapat dilihat pada gambar 5.

C. Perancangan dan Pembuatan Perangkat Lunak

Flowchart aplikasi merupakan alur dari aplikasi yang akan digunakan oleh pengguna dalam memonitor kecepatan kendaraan bermotor. Alur kerja aplikasi ini dimulai dengan proses NodeMCU ESP8266 menghubungkan ke jaringan WiFi,

jika sudah terhubung sensor GPS mengambil data *latitude*, *longitude* dan kecepatan, lalu data tersebut dikirim ke server *Blynk*, pada aplikasi *Blynk* dengan menggunakan widget *webhook* akan dikirim *HTTP Request* untuk menulis data ke *channel* ThingSpeak. *Flowchart* aplikasi dapat dilihat pada gambar 6.

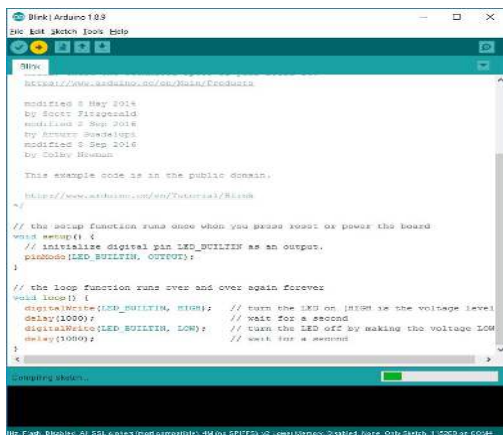
III. HASIL DAN PEMBAHASAN

A. Pengujian Hardware

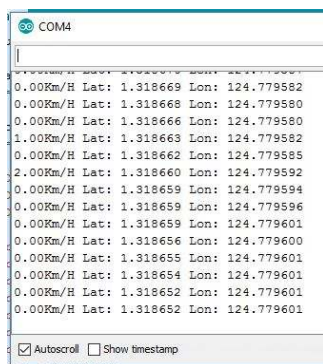
Penelitian implementasi *internet of things* pada *monitoring* kecepatan kendaraan bermotor memerlukan pengujian secara keseluruhan baik perangkat keras (*Hardware*) maupun perangkat lunak (*Software*). Pada pengujian *Hardware*, akan dilakukan pengujian mikrokontroler NodeMCU ESP8266 dan pengujian modul GPS.

1) Pengujian Mikrokontroler NodeMCU ESP8266

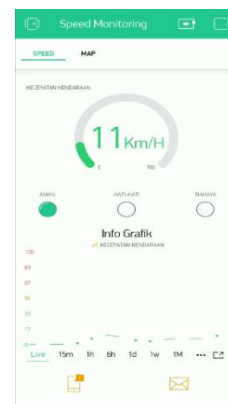
Pengujian dilakukan dengan menghubungkan mikrokontroler NodeMCU ESP8266 pada *USB connection* PC (*Personal Computer*) menggunakan kabel USB. Apabila LED pada mikrokontroler NodeMCU ESP8266 berkedip sekali, maka menandakan bahwa mikrokontroler NodeMCU berfungsi. Setelah melakukan pengecekan *hardware*, kemudian dilakukan pengujian *software* mikrokontroler NodeMCU ESP8266. Pengujian dilakukan dengan mengupload program bawaan *software* Arduino (Arduino IDE) dengan nama "Blink" seperti pada Gambar 7. Mikrokontroler NodeMCU ESP8266 dapat dinyatakan berkerja secara baik apabila LED berkedip sesuai perintah program yang telah upload.



Gambar 7. Proses upload program pada Arudino IDE



Gambar 8. Hasil pengujian modul GPS pada serial monitor



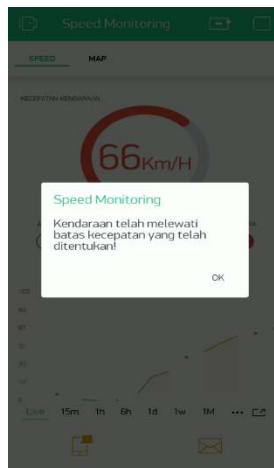
Gambar 9. Tampilan saat LED Aman menyala



Gambar 10. Tampilan saat LED Hati-hati menyala



Gambar 11. Tampilan saat LED Bahaya menyala



Gambar 12. Tampilan notifikasi pada smartphone

2) Pengujian Modul GPS

Pengujian dilakukan Setelah pin *GPS* dengan pin *NodeMCU ESP8266* terhubung, tunggu 2 hingga 10 menit untuk menangkap sinyal dari satelit. Apabila LED pada modul *GPS* telah berkedip, maka menandakan bahwa modul berfungsi. Setelah itu dilanjutkan dengan meng-upload *source code* menggunakan *software* *Arduino IDE* untuk mengukur kecepatan dan menentukan lokasi (latitude dan longitude) dengan modul *GPS*. Apabila modul *GPS* berfungsi maka hasil pengukuran kecepatan dan lokasi akan tampak seperti pada Gambar 8.

B. Pengujian Software

Pengujian *software* pada tugas akhir ini akan dilakukan dua pengujian, yaitu pengujian platform *Blynk* dan pengujian platform *ThingSpeak*.

1) Pengujian Platform Blynk

Data yang telah didapat dengan modul *GPS* akan dikirim ke server *Blynk*. Setelah itu data tersebut akan ditampilkan pada project sesuai dengan *auth token* yang digunakan. Data tersebut akan memberikan *output* berupa widget-widget.

Terdapat tiga data yang didapat dari modul *GPS* yaitu kecepatan (km/h), latitude, dan longitude. Data kecepatan akan ditampilkan pada widget *Gauge* dan *Superchart*, sedangkan latitude dan longitude akan ditampilkan pada widget *Map*.



Gambar 13. Tampilan inbox email



Gambar 14. Tampilan widget Map dengan point pada aplikasi Blynk

Terdapat tiga keadaan sebagai tanda peringatan yang divisualisasikan dengan widget LED, yaitu Aman, Hati-hati, dan Bahaya. LED Aman akan menyala jika kecepatan kurang dari 40 Km/H, LED Hati-hati akan menyala saat kecepatan lebih dari 40 Km/H dan kurang dari 60 Km/H, sedangkan LED Bahaya akan menyala jika kecepatan lebih dari 60 Km/H. Saat LED Bahaya menyala atau saat kecepatan lebih dari 60 Km/H, aplikasi *Blynk* akan memberikan notifikasi pada smartphone dan mengirimkan surat ke email yang telah ditentukan. Pada widget Map juga akan diberikan point ketika LED Bahaya menyala atau kecepatan lebih dari 60 Km/H. Tampilan pengujian pada platform *Blynk* dapat dilihat pada gambar 9, 10, 11, 12, 13, dan 14.

2) Pengujian Platform ThingSpeak

Data yang telah diukur dengan modul *GPS* akan dikirim secara berurutan ke server *Blynk*. Lalu dengan widget *WebHook* pada aplikasi *Blynk* akan mengirim *HTTP/S* request ke server *ThingSpeak* untuk menulis data ke channel *ThingSpeak*.

Pada platform *ThingSpeak* ini, data yang telah dikirim ke server *ThingSpeak* akan ditampilkan berupa *Chart* dan *widget*. Terdapat tiga *Chart* dan satu *widget Gauge* yang digunakan pada tugas akhir ini yaitu, *Chart Latitude*, *Chart Longitude*, *Chart Speed*, dan *widget Gauge Speed*. Tampilan pengujian pada platform *ThingSpeak* dapat dilihat pada gambar 15, 16, 17 dan 18.



Gambar 15. Tampilan chart Latitude



Gambar 18. Tampilan widget Gauge Speed



Gambar 16. Tampilan chart Longitude



Gambar 17. Tampilan chart Speed

TABEL II
 ANALISIS PENGUJIAN ALAT

No.	Kecepatan pada serial monitor (Km/H)	Kecepatan pada Platform Blynk (Km/H)
1.	6	6
2.	9	9
3.	13	13
4.	19	19
5.	15	15



Gambar 19. Tampilan prototipe pada bagasi depan motor

C. Pengujian Alat

Pada tahap ini ditujukan untuk memastikan fungsionalitas dan kinerja dari alat. Pengujian dilakukan dengan mengoperasikan alat kemudian memperhatikan kinerja dari komponen-komponen yang ada pada alat, yaitu mikrokontroler NodeMCU dan modul GPS.

Analisis pengujian alat ini dilakukan dengan mencocokkan data pada serial monitor dengan data pada platform IoT, sehingga nantinya dapat ditentukan alat bekerja dengan efektif. Terdapat tiga data yang diukur dengan modul GPS, tetapi hanya data kecepatan saja yang akan digunakan. Untuk platform IoT hanya digunakan platform Blynk. Percobaan ini dilakukan dengan membawa alat yang dihubungkan ke laptop, lalu mengujinya dengan menggunakan mobil. Analisis pengujian alat dapat dilihat pada tabel II.

D. Pengujian Akurasi Alat

Pada pengujian akurasi alat ini digunakan modul modul GPS yang dirangkai dengan mikrokontroler NodeMCU. Tujuan dari pengujian akurasi alat ini adalah untuk mengetahui apakah prototipe ini dapat memberikan nilai kecepatan yang sesuai jika dibandingkan dengan nilai kecepatan yang aktual.

Prototipe diletakkan pada bagasi/keranjang depan motor dan smartphone diletakkan bersampingan dengan speedometer dengan bantuan perekat. Penempatan alat dapat dilihat pada Gambar 19.

Pengujian akurasi alat ini dilakukan sebanyak sepuluh kali, dimana data pengukuran oleh modul GPS akan dibandingkan dengan kecepatan motor dan akan diambil selisihnya. Pada pengujian ini digunakan platform Blynk sebagai hasil pengukuran modul GPS. Pengujian ini dilakukan di jalan lurus dengan kecepatan konstan. Berikut adalah hasil pengujian yang telah dilakukan pada tabel III, IV, V, VI, VII, VIII, IX, X, XI, dan XII.

E. Statistik Deskriptif

Analisa deskriptif menampilkan rata-rata, standar deviasi, nilai maksimum, dan minimum suatu data. Tujuannya adalah untuk memudahkan pengamatan dalam suatu data. Terdapat 10 pengujian yang akan diuji pengujian statistik deskriptif. Hasil pengujian statistik deskriptif dan dapat dilihat pada tabel XIII dan gambar 21.

Pada table di atas menunjukkan nilai Obs atau jumlah observasi yang akan diteliti berjumlah 15 sampel, nilai Min atau nilai selisih minimum pada setiap pengujian, nilai Max atau nilai selisih maksimum pada setiap pengujian, nilai Mean atau nilai rata-rata selisih pada setiap pengujian, Standar Deviasi atau simpangan baku yang menggambarkan sebaran sampel terhadap rata-ratanya, dan Standar Error atau kesalahan baku yang menggambarkan sebaran rata-rata sampel terhadap rata-rata dari keseluruhan kemungkinan sampel (rata-rata populasi).

TABEL III
PENGUJIAN KE -1

No	Data Kecepatan GPS (Km/H)	Kecepatan Motor (Km/H)	Selisih
1	22	25	3
2	22	24	2
3	22	23	1
4	23	25	2
5	23	25	2
6	22	24	2
7	24	26	2
8	25	27	2
9	24	26	2
10	24	24	0
11	23	24	1
12	22	24	2
13	23	24	1
14	23	23	0
15	22	23	1
	Rata-rata selisih		1.4
	Standar Deviasi		0.833

TABEL IV
PENGUJIAN KE -2

No	Data Kecepatan GPS (Km/H)	Kecepatan Motor (Km/H)	Selisih
1	22	24	2
2	22	23	1
3	22	24	2
4	21	22	1
5	22	25	3
6	24	26	2
7	23	25	2
8	24	25	1
9	24	25	1
10	24	24	0
11	22	24	2
12	23	24	1
13	23	23	0
14	22	24	2
15	23	25	1
	Rata-rata selisih		1.47
	Standar Deviasi		0.828

Pada pengujian 1 memiliki Mean sebesar 1.53, Standar Deviasi sebesar 0.833, dan Standar Error sebesar 0.215. Pada pengujian 2 memiliki Mean sebesar 1.4, Standar Deviasi sebesar 0.828, dan Standar Error sebesar 0.213. Pada pengujian 3 memiliki Mean sebesar 1.33, Standar Deviasi sebesar 0.723, dan Standar Error sebesar 0.186. Pada pengujian 4 memiliki Mean sebesar 1.46, Standar Deviasi sebesar 0.915, dan Standar Error sebesar 0.236. Pada pengujian 5 memiliki Mean sebesar 1.47, Standar Deviasi sebesar 1.060, dan Standar Error sebesar 0.273. Pada pengujian 6 memiliki Mean sebesar 1.67, Standar Deviasi sebesar 0.617, dan Standar Error sebesar 0.159. Pada pengujian 7 memiliki Mean sebesar 1.6, Standar Deviasi sebesar 0.985, dan Standar Error sebesar 0.254. Pada pengujian 8 memiliki Mean sebesar 1.4, Standar Deviasi sebesar 0.910, dan Standar Error sebesar 0.235. Pada pengujian 9 memiliki Mean sebesar 1.6, Standar Deviasi sebesar 1.055, dan Standar Error sebesar 0.272. Pada pengujian 10 memiliki Mean sebesar 1.73, Standar Deviasi sebesar 0.883, dan Standar Error sebesar 0.228.

TABEL V
PENGUJIAN KE -3

No	Data Kecepatan GPS (Km/H)	Kecepatan Motor (Km/H)	Selisih
1	20	22	2
2	21	23	2
3	22	22	0
4	21	23	2
5	22	23	1
6	22	24	2
7	23	25	2
8	24	25	1
9	20	22	2
10	21	21	0
11	21	20	1
12	19	20	1
13	21	22	1
14	20	21	1
15	21	23	2
	Rata-rata selisih		1.33
	Standar Deviasi		0.723

TABEL VI
PENGUJIAN KE -4

No	Data Kecepatan GPS (Km/H)	Kecepatan Motor (Km/H)	Selisih
1	21	23	2
2	22	24	2
3	23	24	1
4	23	23	0
5	25	26	1
6	23	26	3
7	23	26	3
8	23	24	1
9	23	23	0
10	22	24	2
11	22	24	2
12	22	23	1
13	21	22	1
14	20	22	2
15	21	22	1
	Rata-rata selisih		1.46
	Standar Deviasi		0.915

TABEL VII
PENGUJIAN KE -5

No	Data Kecepatan GPS (Km/H)	Kecepatan Motor (Km/H)	Selisih
1	28	29	1
2	26	29	3
3	25	28	3
4	27	28	1
5	26	28	2
6	29	31	2
7	28	31	3
8	27	29	2
9	27	28	1
10	26	25	1
11	24	25	1
12	24	24	0
13	24	24	0
14	25	27	2
15	23	23	0
Rata-rata selisih			1.47
Standar Deviasi			1.060

TABEL X
PENGUJIAN KE -8

No	Data Kecepatan GPS (Km/H)	Kecepatan Motor (Km/H)	Selisih
1	34	36	2
2	32	34	2
3	33	34	1
4	33	35	2
5	35	35	0
6	30	33	3
7	30	32	2
8	31	33	2
9	31	32	1
10	31	31	0
11	31	32	1
12	31	31	0
13	29	31	2
14	29	30	1
15	31	33	2
Rata-rata selisih			1.4
Standar Deviasi			0.910

TABEL VIII
PENGUJIAN KE -6

No	Data Kecepatan GPS (Km/H)	Kecepatan Motor (Km/H)	Selisih
1	29	31	2
2	29	30	1
3	28	30	2
4	28	30	2
5	29	32	3
6	29	31	2
7	33	35	2
8	33	34	1
9	30	31	1
10	31	33	2
11	30	32	2
12	30	31	1
13	29	31	2
14	29	30	1
15	27	28	1
Rata-rata selisih			1.67
Standar Deviasi			0.617

TABEL XI
PENGUJIAN KE -9

No	Data Kecepatan GPS (Km/H)	Kecepatan Motor (Km/H)	Selisih
1	30	32	2
2	31	31	0
3	31	32	1
4	32	34	2
5	33	35	2
6	31	33	2
7	33	33	0
8	31	33	2
9	29	32	3
10	28	31	3
11	31	33	2
12	30	33	3
13	32	32	0
14	29	30	1
15	28	29	1
Rata-rata selisih			1.6
Standar Deviasi			1.055

TABEL IX
PENGUJIAN KE -7

No	Data Kecepatan GPS (Km/H)	Kecepatan Motor (Km/H)	Selisih
1	28	30	2
2	28	29	1
3	30	30	0
4	29	31	2
5	30	33	3
6	30	32	2
7	30	31	1
8	30	30	0
9	28	30	2
10	28	29	1
11	26	29	3
12	30	33	3
13	32	33	1
14	30	31	1
15	29	31	2
Rata-rata selisih			1.6
Standar Deviasi			0.985

TABEL XII
PENGUJIAN KE -10

No	Data Kecepatan GPS (Km/H)	Kecepatan Motor (Km/H)	Selisih
1	31	34	3
2	32	32	0
3	31	32	1
4	30	33	3
5	32	34	2
6	31	33	2
7	31	33	2
8	30	32	2
9	31	32	1
10	32	34	2
11	31	32	1
12	32	33	1
13	30	33	3
14	31	33	2
15	31	32	1
Rata-rata selisih			1.73
Standar Deviasi			0.883

TABEL XIII
STATISTIK DESKRIPTIF

No	Obs	Min	Max	Std. Deviasi	Std. Error
1	34	36	2	0.833	0.215
2	32	34	2	0.828	0.213
3	33	34	1	0.723	0.186
4	33	35	2	0.915	0.236
5	35	35	0	1.060	0.273
6	30	33	3	0.617	0.159
7	30	32	2	0.985	0.254
8	31	33	2	0.910	0.235
9	31	32	1	1.055	0.272
10	31	31	0	0.883	0.228
11	31	32	1	0.833	0.215
12	31	31	0	0.828	0.213
13	29	31	2	0.723	0.186
14	29	30	1	0.915	0.236
15	31	33	2	1.060	0.273

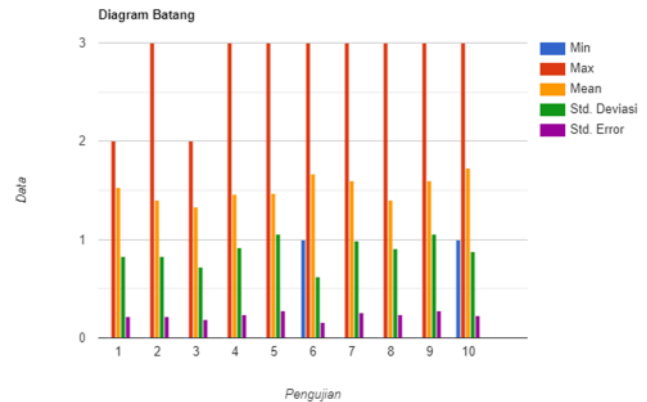
IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian mengenai Implementasi *Internet of Things* Pada *Monitoring* Kecepatan Kendaraan Bermotor maka dapat disimpulkan bahwa pengujian akurasi alat pada penelitian ini dilakukan di jalanan lurus dan pada kecepatan konstan, sehingga jika terjadi perubahan kecepatan yang signifikan dalam periode waktu yang singkat, maka data kecepatan *GPS* dengan data kecepatan aktual akan terjadi selisih yang besar serta koneksi internet dan lokasi alat pada saat dijalankan (*indoor/outdoor*) memiliki pengaruh besar dalam pengambilan dan pengiriman data.

V. KUTIPAN

- [1] L. Jemadu and I. Krisnamusi, "Remaja Paling Banyak Terlibat Kecelakaan Lalu Lintas," *www.suara.com*, 2017. [Online]. Available: <https://www.suara.com/otomotif/2017/10/04/181547/remaja-paling-banyak-terlibat-kecelakaan-lalu-lintas>. [Accessed: 24-Nov-2018].
- [2] M. Bagus Rachmanto, "Benarkah Pengemudi Usia Remaja Cenderung Ugal-ugalan?," *www.medcom.id*, 2020. [Online]. Available: <https://www.medcom.id/otomotif/tips-otomotif/RkjbGB3k-benarkah-pengemudi-usia-remaja-cenderung-ugal-ugalan>. [Accessed: 26-May-2020].
- [3] TRG Investama, "Internet of Things (IoT) & Cloud Computing," *www.trg-investama.com*, 2015. [Online]. Available: <http://trg-investama.com/internet-of-things-iot-cloud-computing/>. [Accessed: 25-Nov-2018].
- [4] Y. Yudhanto, "Apa itu IoT (Internet Of Things)?," *www.ilmukomputer.org*, 2015. [Online]. Available: <https://ilmukomputer.org/2015/05/15/apa-itu-iot-internet-of-things/>. [Accessed: 08-Feb-2019].
- [5] M. Mudjahidin and N. D. P. Putra, "RANCANG BANGUN SISTEM INFORMASI MONITORING PERKEMBANGAN PROYEK BERBASIS WEB STUDI KASUS DI DINAS BINA MARGA DAN PEMANTUSAN," *J. Tek. Ind.*, 2012.
- [6] F. D. Hobbs, *Perencanaan dan teknik lalu lintas*, 2nd ed. Yogyakarta: Gadjah Mada University Press, 1995.
- [7] A. Sunyoto and J. E. Istiyanto, "Integrasi modul GPS Receiver dan GPRS untuk penentuan posisi dan jalur pergerakan obyek bergerak," *Dr. Diss.*, 2007.
- [8] Afrizal, A. Sukmaaji, and T. Sutanto, "Android Personnel Monitoring Location Pada Institusi Kepolisian Berbasis Web," *J. Sist. Inf. dan Komput. Akunt.*, vol. 3, 2013.



Gambar 20. Diagram batang statistik deskriptif

- [9] T. T. Saputro, "Mengenal NodeMCU: Pertemuan Pertama," *www.embeddednesia.com*, 2017. [Online]. Available: <https://embeddednesia.com/v1/tutorial-nodemcu-pertemuan-pertama/>. [Accessed: 09-Feb-2019].
- [10] Arafat, "Sistem Pengamanan Pintu Rumah Berbasis *Internet Of Things* (IoT) Dengan ESP8266," *Technol. J. Ilm.*, vol. 7, 2016.
- [11] M. Chwalisz, "thingspeak Documentation," *SL Thingspeak*, 2016.
- [12] N. W. Nugraha and B. Rahmat, "SISTEM PEMBERIAN MAKANAN DAN MINUMAN KUCING MENGGUNAKAN ARDUINO," *J. Teknol. Inf. dan Komun.*, vol. 13, 2018.



Penulis bernama lengkap Christianto Wibisono Darmawan, lahir di Bekasi 26 Agustus 1997. Penulis merupakan anak ke-1 dari 2 bersaudara. Penulis menempuh pendidikan pertama di TK Santa Maria Monika Cibitung (2001-2003). Penulis kemudian melanjutkan ke Sekolah Dasar Strada Budi Luhur I Bekasi (2004-2006), kemudian pindah ke Sekolah Dasar Paskalis I Tanah Tinggi (2006-2007), kemudian pindah ke Sekolah Dasar Katolik Santa Clara Tomohon (2007-2009). Penulis lalu melanjutkan ke Sekolah Menengah Pertama Negeri 1 Tomohon (2009-2010). Kemudian penulis melanjutkan Sekolah Menengah Atas Lokon Santo Nikolaus Tomohon (2012-2015). Pada tahun 2015 penulis melanjutkan studi ke Perguruan Tinggi Negeri di Universitas Sam Ratulangi Manado dengan mengambil Program Studi S-1 Teknik Informatika di Jurusan Teknik Elektro Fakultas Teknik. Pada bulan November tahun 2019 Penulis mengajukan proposal Skripsi untuk memenuhi syarat meraih gelar sarjana (S1) dengan judul Implementasi *Internet of Things* Pada *Monitoring* Kecepatan Kendaraan Bermotor yang kemudian disetujui dan melanjutkan pembuatan penelitian skripsi. Pembuatan skripsi ini dibimbing oleh dua dosen pembimbing, yaitu Sherwin R. U. A. Sompie, ST, MT dan Feisy Diane Kambey, ST, MT. Pada 30 Juni 2020, penulis resmi menyelesaikan skripsi dengan menyandang gelar sarjana komputer dengan predikat sangat memuaskan.