Jurnal Teknik Elektro dan Komputer vol ? no ? Agustus 2020, hal. ?-?
p-ISSN : 2301-8402, e-ISSN : 2685-368X ,dapat diakses melalui https://ejournal.unsrat.ac.id/index.php/elekdankom

Implementasi *Google Internet of Things Core* pada *Monitoring* Volume Ban Angin Mobil

Praisye E. A. Kaunang [1)], Sherwin R. U. A. Sompie [2)], Arie S. M. Lumenta [3)]
Teknik Elektro Universitas Sam Ratulangi, Jl. Kampus Bahu-Unsrat Manado, 95115
E-mail:15021106152@student.unsrat.ac.id.[1)], a_sompie@yahoo.com [2)], al@unsrat.ac.id [3)]
Diterima: tgl: direvisi: tgl: disetujui: tgl

Abstract* — Tires are one of the supporting components of a vehicle that plays a very important role. The condition of a good vehicle tire with air pressure in accordance with specifications is a safety requirement in driving. Tire pressure that is less or excessive can cause accidents when the tire breaks and will shorten the life of the tire. From the existing problems, a device that functions to monitor the state of the wind inside the tire is created in realtime by implementing the Google IoT Core platform and using the ESP32 device as a microcontroller that controls the MPX5700 sensor. Devices and sensors that take wind pressure data on tires connected to the internet can find, monitor and maintain air pressure stability on tires so that they drive safely and comfortably. The results of this study are building a system of air pressure monitoring on website-based vehicles.

Keyword*: *Vehicle Tires; Google IoT Core; ESP32 device; MPX5700 sensor; Air pressure.

Abstrak— Ban merupakan salah satu komponen pendukung suatu kendaraan yang berperan sangat penting. Kondisi keadaan ban kendaraan yang baik dengan tekanan udara yang sesuai dengan spesifikasi merupakan syarat keamanan dalam berkendara. Tekanan udara ban yang kurang ataupun berlebihan dapat membuat terjadinya kecelakaan ketika ban pecah dan akan memperpendek umur pakai ban. Dari permasalah yang ada dibuat suatu perangkat yang berfungsi untuk memonitor keadaan angin yang ada didalam ban secara *realtime* dengan menerapkan platform *Google IoT Core* dan menggunakan perangkat ESP32 sebagai mikrokontroler yang mengendalikan sensor MPX5700. Perangkat dan sensor yang mengambil data tekanan angin pada ban terhubung dengan jaringan internet dapat mengetahui, memantau dan menjaga stabilitas tekanan udara pada ban agar berkendara dengan aman dan nyaman. Hasil dari penelitian ini adalah membangun sistem *monitoring* tekanan udara pada kendaraan berbasis *website*.

Kata Kunci: Ban Kendaraan; *Google IoT Core*; Perangkat ESP32; Sensor MPX5700; Tekanan Udara.

I. PENDAHULUAN

Kendaraan merupakan alat transportasi yang digerakan oleh mesin maupun dan juga manusia. Kendaraan dianggap mampu membantu mempermudah hidup manusia sehingga gerak hidup manusia mampu berubah menjadi mudah. Dalam mengemudikan kendaraan, ada hal yang harus diperhatikan yaitu berkendara dengan memikirkan keselamatan dalam berkendara, karena kecelakaan tidak dapat dicegah namun bisa diminimalisirkan cedera yang mungkin bisa terjadi saat kecelakaan. Oleh sebab itu pada dasarnya kendaraan memerlukan komponen pendukung selain mesin agar dapat digunakan dengan aman yang salah satunya adalah Ban. Ban merupakan salah satu komponen pendukung suatu kendaraan yang berperan sangat penting. Kondisi keadaan ban kendaraan yang baik dengan tekanan udara yang sesuai dengan spesifikasi merupakan syarat keamanan dalam berkendara. Ban pada setiap kendaraan seudah memiliki standar yang telah ditetapkan oleh pabrik. Salah satu masalah yang sering dihadapi pengendara adalah berkurang dan berlebihnya tekanan udara pada ban yang mengakibatkan kenyamanan berkendara berkurang. Tekanan udara ban yang kurang ataupun berlebihan dapat membuat terjadinya kecelakaan ketika ban pecah dan akan memperpendek umur pakai ban.

Dari permasalah diatas diperlukan suatu perangkat yang berfungsi untuk memonitor keadaan angin yang ada didalam ban secara *realtime* dengan menggunakan platform *Google IoT Core* yang dimana perangkat dan sensor yang mengambil data tekanan angin pada ban terhubung dengan jaringan internet yang bertujuan agar penggendara roda empat dapat mengantisipasi terjadinya kecelakaan secara tiba-tiba sehinga dengan perangkat yang dibuat penggendara dapat mengetahui, memantau dan menjaga stabilitas tekanan udara pada ban agar berkendara dengan aman dan nyaman.

A. *Internet*

Internet Internet adalah kumpulan atau jaringan dari komputer yang ada diseluruh dunit. Internet (kependekan dari *interconnection-networking*) secara harafiah ialah sistem global dari seluruh jaringan komputer yang saling terhubung menggunakan standar *Internet Protocol Suit* (TCP/IP) untuk melayani miliayaran pengguna di seluruh dunia. Selain itu internet dapat disebut sebagai sumber daya informasi yang dapat digunakan oleh seluruh dunia dalam mencari informasi.[1]

B. *Internet of Things*

Internet of Things (IoT) adalah sebuah konsep yang menggunakan internet sebagai jaringan infrastruktur utama yang mengkoneksikan objek – objek tertentu (Miorandi, Sicari De Pellegrini & Chlamtac, 2012). Dalam hal ini IoT juga bisa diartikan internet yang menghubungkan antar *things*, dimana *things* disini berarti informasi seperti meta data (Bari, Mani & Berkovich, 2013).

C. *MQTT (Message Queuing Telemetry Transport)*

Perangkat MQTT merupakan sebuah protokol pertukaran pesan dengan model *publish/subscribe* yang sederhana dan ringan serta didesain untuk perangkat yang memiliki

kemampuan terbatas dan *bandwidth* yang kecil, *latency* tinggi, atau jaringan yang tidak andal. Prinsip desain MQTT adalah untuk meminimalisasi *bandwidth* jaringan dan kebutuhan resource perangkat dan tetap menjamin keandalan dan beberapa tingkat jaminan tersampaikannya sebuah pesan. Prinsip inilah yang membuat protokol ini ideal untuk diaplikasikan pada komunikasi *machine-to-machine* (M2M) atau Internet of Things dan untuk aplikasi mobile dimana *bandwidth* dan kapasitas baterai terbatas. Sejak 29 Oktober 2014, MQTT telah distandarisasi oleh *Organization for the Advancement of Structured Information Standards* (OASIS). Spesifikasi protokol MQTT telah dipublikasikan secara terbuka dengan lisensi *open-source*. Pertukaran pesan dengan model *publish/subscribe* pada MQTT merupakan alternatif dari model *client-server*, dimana sebuah client (publisher/subscriber) berkomunikasi langsung dengan sebuah endpoint lainnya pada sebuah topik melalui sebuah broker yang bertugas melakukan penyaringan pesan dan mendistribusikannya.[2]

D. Google Internet of Things Core

Cloud IoT Core adalah layanan terkelola sepenuhnya yang bisa menghubungkan, mengelola dan menyerap data secara mudah dan aman dari jutaan perangkat yang tersebar secara global. *Cloud* IoT Core memberikan solusi lengkap untuk mengumpulkan, memproses, menganalisis, memvisualisasikan data IoT secara *real-time* untuk mendukung peningkatan efisiensi operasional.[3] *Cloud* IoT Core yang menggunakan *Cloud pub/sub*, dapat menggabungkan data perangkat yang tersebar menjadi satu sistem global yang terintegrasi secara lancar dengan layanan analisis data *Google Cloud*.[3] *Cloud IoT Core* juga dapat menghubungkan beberapa atau jutaan perangkat yang tersebar secara global dengan aman melalui *endpoint* protokol yang menggunakan *load balancing* otomatis dan penskalaan horizontal, untuk memastikan penyerapan data yang lancar dalam kondisi apapun. Clout IoT Core mendukung protokol MQTT dan HTTP standar sehingga dapat menggunakan perangkat yang ada dengan perubahan minimum pada firmware. *Cloud* IoT Core berjalan pada infrastruktur tanpa server milik google, yang ditentukan skalanya secara otomatis sebagai respons terhadap perubahan *realtime* dan mematuhi protokol keamanan standar industri yang ketat untuk melindungi data bisnis.[3]

Fitur inti dari *Google IoT Core* memiliki dua komponen utama ialah pengaturan perangkat dan jembatan protokol. Pengaturan perangkat memungkinkan masing-masing perangkat dikonfigurasikan dan dikelola secara aman dengan cara yang kasar, manajemen dapat dilakukan melalu konsol atau secara terprogram. Ini juga dapat memelihara konfigurasi logis dari setiap perangkat dan dapat digunakan untuk mengontrol perangkat dari jauh. Jembatan protokol menyediakan titik akhir koneksi untuk protokol dengan penyeimbangan beban otomatis untuk semua koneksi perangkat. Jembatan protokol menerbitkan semua telemetri perangkat ke *Cloud Pub/Sub*, yang kemudian dapat dikonsumsi oleh sistem.[3]

Google IoT Core sendiri baru dirilis pada bulan Mei tahun 2017 namun baru bisa digunakan untuk umum pada bulan februari tahun 2018. *Google IoT Core* memudahkan pengguna untuk menyambungkan perangkat yang terdistribusi secara global ke GCP (*Google Cloud Platform*), sehingga dapat mengelolah secara terpusat dan membangun aplikasi yang kaya dan berintegritas dengan layanan data analisis, sehingga semua data, skalabilitas, dan kebutuhan dikelola otomatis.[3]

Google IoT Core mempunyai keuntungan dalam bidang industry sebagai contoh dapat memantau, menganalisis dan memprediksi penggunaan secara *real-time,* sehingga dirancang untuk membantu menyelesaikan masalah dengan meminimalisirkan resiko dari proses pemantauan dan manajemen perangkat.

E. Monitoring

Monitoring adalah suatu kegiatan yang dilakukan terus-menerus dan bersifat utuh dari manajemen perusahaan yang isinya adalah penilaian yang bersifat sistimatis terhadap kemajuan suatu pekerjaan. (Mudjahidin & Pahang Pu, 2010). Dengan kata lain, kegiatan monitoring adalah proses pencatatan dan pengumpulan informasi terhadap tugas-tugas projek secara periodik. Monitoring juga berguna untuk melihat dan memantau perkembangan suatu pekerjaan yang sedang berjalan.[4]

F. Bahasa Pemograman C++

Bahasa C dan C++ merupakan bahasa yang sangat popular 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++.[5]

G. Integrated Development Environment (IDE)

IDE singkatan dari *Integrated Development Environment*. *IDE* merupakan program komputer sebagai lingkungan pengembangan aplikasi atau program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari *IDE* adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak.

H. 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.[6]

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.[6] Untuk tampilan arduino IDE dapat dilihat pada Gambar 1 dan Gambar 2.

I. Node.Js

Perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis web dan ditulis dalam sintaks bahasa pemrograman JavaScript. Bila selama ini kita mengenal JavaScript sebagai bahasa pemrograman yang berjalan di sisi client atau browser saja, maka Node.js ada untuk melengkapi peran JavaScript sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi server, seperti halnya PHP, Ruby, Perl, dan sebagainya. Node.js dapat berjalan di sistem operasi Windows, Mac OS X dan Linux tanpa perlu ada perubahan kode program. Node.js memiliki pustaka server HTTP sendiri sehingga memungkinkan untuk menjalankan server web tanpa menggunakan program server web seperti Apache atau Nginx.

Gambar 1 Arduino IDE


Gambar 2 Tampilan Arduino IDE

J. Bootstrap

Bootstrap merupakan sebuah library framework CSS yang telah dibuat khusus uintuk mengembangkan front end sebuah website. Bootstrap juga dikenal sebagai salah satu framework CSS, HTML, Javascript yang begitu populer di kalangan website developer atau pengembang website.[7]

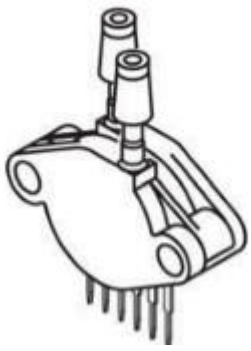
Sebagai pengguna Anda hanya perlu memanggil setiap kelas yang digunakan, contohnya seperti navigasi, tabel, grind, tombol atau sebagainya. Banyak fungsi bootstrap yang bisa dipakai untuk sebuah website.[7] Berikut fungsinya:

1) Bisa mempercepat waktu untuk memproses pembuatan *front end* sebuah website
2) Menampilkan sisi website yang lebih modern dan juga khas anak jaman sekarang
3) Tampilan dari bootstrap sendiri sudah sangat responsive sehingga sangat mendukung untuk segala jenis resolusi, entah itu tablet, smartphone ataupun juga PC dan laptop.
4) Website yang menggunakan bootstrap umumnya lebih ringan karena lebih terstuktur.

K. Firebase

Firebase memiliki produk utama, yaitu menyediakan database *realtime* dengan *backend* sebagai layanan (*Backend as a service*). Layanan ini menyediakan pengembangan aplikasi API yang memungkinkan aplikasi data yang akan disinkronisasi di klien dan disimpan di *cloud* firebase ini. Firebase menyediakan *library* untuk berbagai client platform yang memungkinkan integrase dengan Android, iOs, JavaScript, Java, Objective-C, dan Node aplikasi Js dan dapat juga disebut sebagai layanan DbaaS (*Database as a Service)* dengan konsep *realtime*. Firebase digunakan untuk mempermudah dalam penambahan fitur-fitur yang akan dibangun oleh *developer*.[8]


Gambar 3 Mikrokontroller DOIT ESP32 DEVKIT VI


Gambar 4 Konfigurasi Sensor *Pressure Gauge* MPX5700

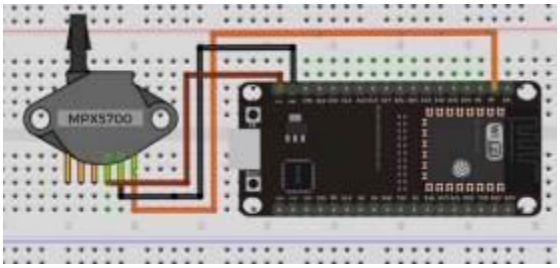
Semua data Firebase *Realtime* Database disimpan sebagai objek JSON. Bisa dianggap basis data sebagai JSON tree yang di-host di awan. Tidak seperti basis data SQL, tidak ada tabel atau rekaman. Ketika ditambahkan JSON tree data akan menjadi simpul dalam struktur JSON yang ada. Meskipun basis data menggunakan JSON tree, data yang tersimpan dalam basis data bisa diwakili sebagai tipe bawaan tertentu yang sesuai dengan tipe JSON yang tersedia untuk membantu menulis lebih banyak kode yang bisa dipertahankan.[8]

L. Mikrokontroller DOIT ESP32 DEVKIT VI

Mikrokontroller DOIT ESP32 DEVKIT V1 adalah mikrokontroler yang dikenalkan oleh *Espressif System* merupakan penerus dari mikrokontroler ESP8266. Tampilan atau bentuk dari mikrokontroller ini dapat dilihat pada Gambar 3. Mikrokontroller DOIT ESP32 DEVKIT V1 ini sudah tersedia modul WiFi dalam chip sehingga mendukung untuk membuat sistem aplikasi *Internet of Things.* Mikrokontroller DOIT ESP32 DEVKIT V1 ini memiliki unggulan yaitu sistem berbiaya rendah, dan juga berdaya rendah yang terintergrasi dengan chip mikrokontroler serta memiliki bluetooth dengan mode ganda dan fitur hemat daya menjadikan Mikrokontroller DOIT ESP32 DEVKIT V1 kompatibel dengan perangkat selular. Mikrokontroler sederhana ini bisa digunakan sebagai sistem mandiri yang lengkap atau dapat dioperasikan sebagai perangkat pendukung mikrokontroler host (Biswar, 2010).[9]

TABEL I
WIRING MIKROKONTROLER NODEMCU ESP32 DENGAN SENSOR MPX5700

No	Pin sensor MPX5700	Warna kabel	Pin NodeMCU ESP32
1	Vout	Orange	GPIO36
2	GROUND	Hitam	GND
3	Vcc	Coklat	3.3 Volt



Gambar 5 Rangkaian Sensor MPX5700 dengan Mikrokontroller ESP32


Gambar 6 Tampilan *Prototype* ESP32 dan Sensor MPX5700

M. Sensor Pressure Gauge MPX5700

Pressure gauge adalah alat yang digunakan untuk mengukur tekanan fluida (gas atau liquid) dalam tabung tertutup. Satuan dari alat ukur tekanan ini berupa psi (*pound per square inch*), psf (*pound per square foot*), mmHg (*millimeter of mercury*), inHg (*inch of mercury*), bar, atm (*atmosphere*).[10] Konfigurasi sensor pressure gauge MPX5700 dapat dilihat pada Gambar 4.

MPX5700 merupakan sensor tekanan udara dengan output analog, sensor ini merupakan sensor produk dari *Frescaal Semikonduktor, Inc*. MPX5700 dapat mengukur tekanan udara, oil maupun cairan lain dengan batas tekanan maksimum sebesar 700 kPa. Sensor MPX5700 dapat mengukur tekanan dengan 3 macam mode pengukuran yaitu, pengukuran *gauge, absolute* maupun *diferential.* Sedangkan paket dari sensor MPX5700 banyak jenisnya. Gambar 5 merupakan konfigurasi sensor *pressure gauge* seri MPX5700DPCASE 867C-05. Sensor ini memiliki daerah ukur untuk tekanan dari 0-700 kPa, dengan tingkat akurasi ±2,5%. Idealnya cocok untuk *Microprosessor* atau *Mikrokontroleller*.[10]

Konfigurasi pin sensor MPX5700 terdiri dari 6 pin dan yang digunakan hanya 3 pin saja, yaitu pin 1 sebagai tegangan output, pin 2 sebagai ground sedangkan pin 3 sebagai masukan dari tegangan *supply* sebesar 5 volt, sedangkan 3 pin yang lain NC (*not connects*). Dari spesifikasi, sensor MPX5700 bekerja pada tegangan 5 volt. Tingkat sensivitas dari sensor sebesar 6,4mV/kPa dengan tegangan output dari 0,2-volt hingga maksimum 4,7 volt.[10]

Gambar 7 Tampilan *Flowchart* Aplikasi

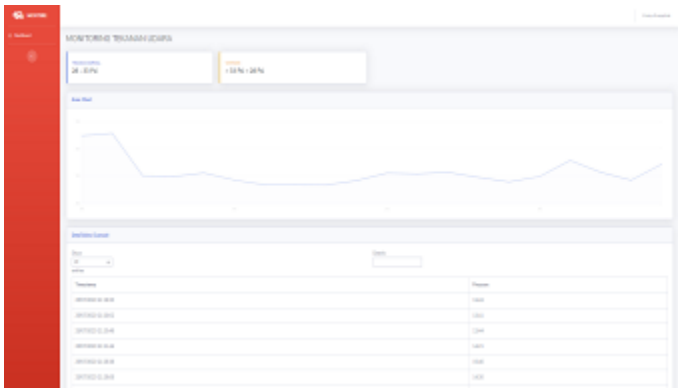
Kelebihan dari sensor ini adalah memiliki kepekaan yang baik terhadap tekanan yang dihasilkan, masa aktif yang lama, dan membutuhkan biaya yang lebih rendah. Sensor ini juga memiliki tingkat sensivitas yang rendah sehingga mudah dibawa kemana-mana, karena sensor ini akan bekerja jika ada tekanan udara yang diberikan. Dengan memanfaatkan prinsip kerja dari sensor MPX5700 ini, tekanan didalam ruang tertutup dapat diukur.

N. Ban

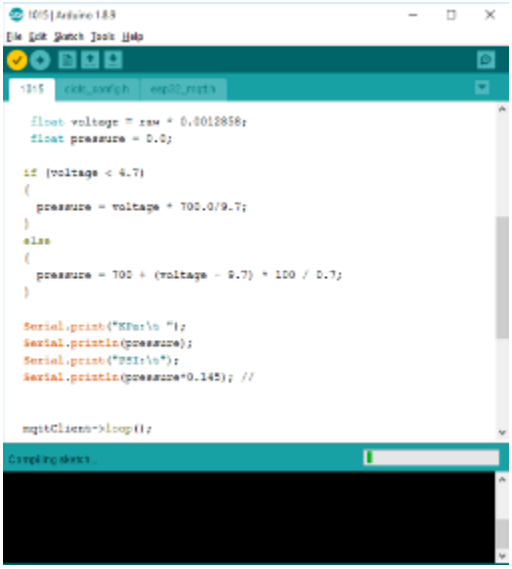
Ban dalah bagian yang berhubungan langsung dengan permukaan jalan. Fungsi ban adalah untuk memperoleh gaya gesek yang lebih besar dengan permukaan jalan dan memperoleh jalannya mobil yang lebih nyaman dengan menyerap kejutan-kejutan jalan.

Jumlah udara di dalam ban dapat diukur dengan menggunakan alat pengukur tekanan udara (*air pressure*). Bergantung pada tekanan udara, ban dapat digolongkan pada yang bertekanan tinggi (*high pressure tire*), ban tekanan rendah (*ballon tire*), dan ekstra ban tekanan rendah.[11]

1) Ban tekanan tinggi tekanan udaranya 4.22 sampai dengan 6.32 kg/cm^2 (60 – 90 psi). Ban dilengkapi dengan *case* yang tebal untuk menahan beban yang berat.



Gambar 8 Tampilan Halaman Utama Website



Gambar 9 Proses Upload Program pada Arduino IDE

2) Ban tekanan rendah tekanan udaranya 2.10 sampai 2.53 kg/cm^2 (30 – 60 psi). Luas penampang melintangnya kira-kira dua kali lebih besar dan ban tekanan tinggi. Luas permukaan yang bersinggungan dengan jalan lebih besar, karena volume udara lebih besar dan tekanan rendah, maka efek empuknya lebih baik.

Ban tekanan ektra renda (*extra low-pressure*). Tekanan udara 1.00 sampai dengan 2.10 kg/cm^2 (14-30 psi) dan digunakan terutama pada mobil penumpang.[11]

II. METODE PENELITIAN

A. Observasi dan Pengumpulan Data

Observasi merupakan metode pengumpulan data dengan mengamati secara langsung. Pada tahap ini dilakukan pengumpulan data dengan mengambil bukti beberapa gambar, serta video dengan mengamati langsung keadaan pada benda seperti apa agar bisa dijadikan sampel pada tahap uji coba.

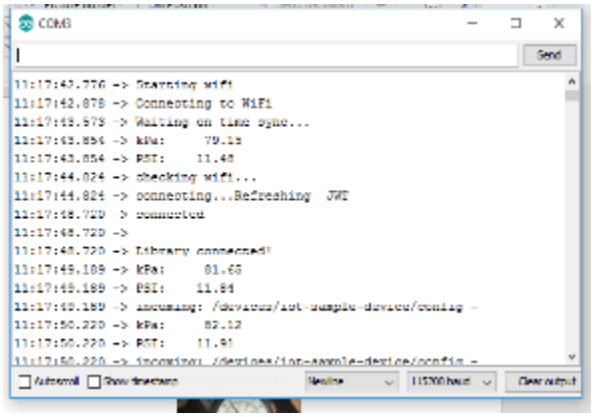
Penelitian ini melakukan pengumpulan data dengan 2 metode, yaitu:

1) Data Primer

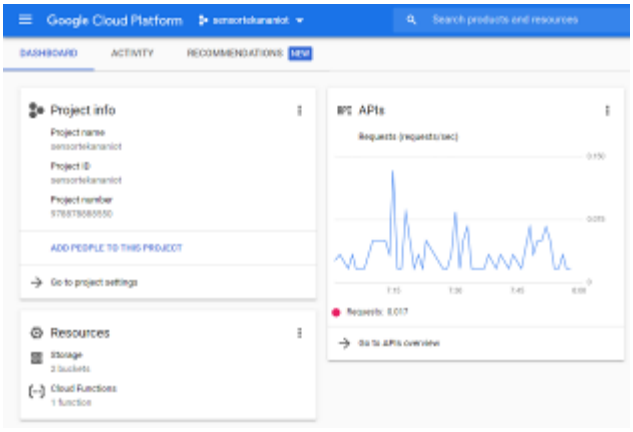
Pengujian yang dilakukan pada ban mobil secara manual menggunakan alat tekanan ban biasa (*tire gauge*)

2) Data Sekunder

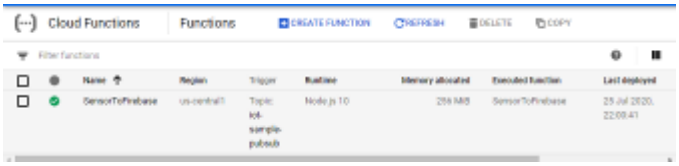
Sumber data sekunder berasal dari studi literatur yang digunakan sebagai referensi dalam proses pembuatan *monitoring* tekanan angin pada ban mobil



Gambar 10 Hasil Teknanan Udara menggunakan Sensor MPX5700

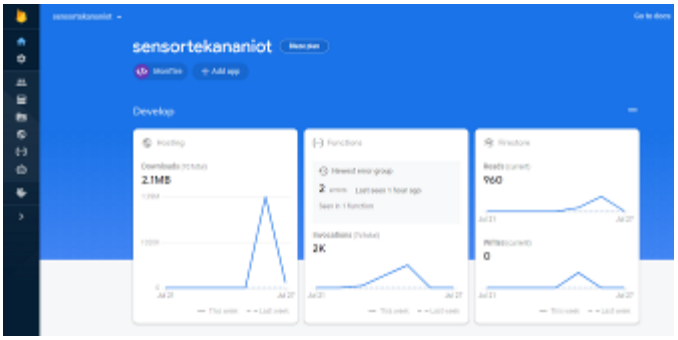


Gambar 11 *Function* yang Dibuat pada *Google IoT Core*

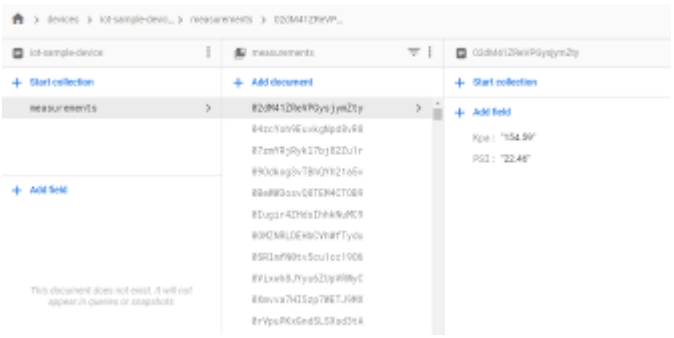


Gambar 12 *Function* yang dibuat di *Cloud Function*

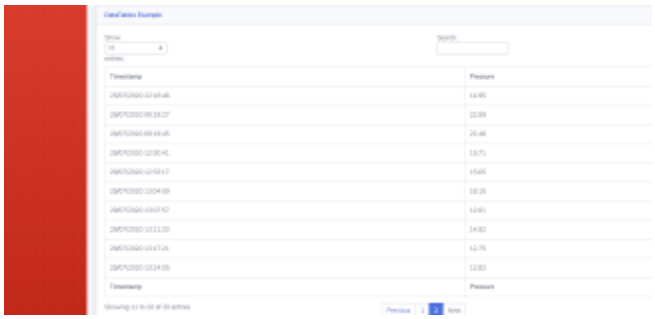
Gambar 13 Tampilan *Function* yang Siap Dihubungkan dengan Firebase



Gambar 14 Tampilan *Project* yang Dibuat pada Firebase



Gambar 15 Tampilan Database yang Dikirimkan Sensor



Gambar 16 Tampilan Kumpulan Data Sensor yang Masuk

Gambar 17 Tampilan Hasil Pengukuran Ban Normal

TABEL II
DATA PENGUJIAN AKURASI ALAT PADA BAN NORMAL

Percobaan	Ban Normal			
	I	II	III	IV
Tekanan pada ESP32&MPX5700	30.23	28.20	29.30	29.87
Tekanan pada *manual tire pressure gauge*	32	31	32	30
Selisih	± 1.7	± 2.8	± 2.7	± 0.1

TABEL III
DATA PENGUJIAN AKURASI PERAANGKAT

No	Output Data Sensor pada Website	Output Data dengan *Tire Pressure* manual	Selisih
1	30.35	32 Psi	1.65
2	30.35	32 Psi	1.65
3	30.35	32 Psi	1.65
4	30.35	32 Psi	1.65
5	30.35	32 Psi	1.65
6	28.28	30 Psi	1.72
7	28.28	30 Psi	1.72
8	28.28	30 Psi	1.72
9	28.28	30 Psi	1.72
10	17.93	18 Psi	0.07
11	17.93	18 Psi	0.07
12	17.93	18 Psi	0.07
13	17.93	18 Psi	0.07
14	17.93	18 Psi	0.07
15	17.93	18 Psi	0.07
16	26.32	28 Psi	1.68
17	26.32	28 Psi	1.68
18	26.31	28 Psi	1.67
19	26.32	28 Psi	1.68
20	26.32	28 Psi	1.68

B. Perancangan dan Pembuatan Perangkat Keras (Hardware)

Perancangan *hardware* diperlukan untuk memberikan gambaran yang jelas dan lengkap untuk nantinya digunakan dalam pembuatan program komputernya. Node MCU dengan chip ESP32 sudah didukung koneksi WiFi. Sensor MPX5700 yaitu sensor yang bisa menghitung tekanan angin didalam ban seperti pada kasus yang di ambil yang memonitoring tekanan angin pada ban mobil Daihatsu Sirion. Agar mikrokontroller ESP32 bisa terhubung dengan sensor MPX5700, pin Vout pada sensor di hubungkan dengan pin GPIO36 pada

mikrokontroller, pin GROUND pada sensor dihubungkan dengan pin GND pada mikrokontroller, dan pin VCC pada sensor dihubungkan dengan pin 3.3-volt pada mikrokontroller. Pada Tabel I adalah penyambungan kabel pada mikrokontroller NodeMCU ESP32 dengan sensor MPX5700 dan Gambar 5 adalah rangkaian sensor MPX5700 dengan mikrokontroller ESP32.

Prototipe merupakan rancangan visual yang menggambarkan suatu produk yang dikembangkan sebelum dibuat dalam skala yang sebenarnya atau sebelum diproduksi secara massal. Pembuatan prototipe dilakukan dengan *wiring* sensor tekanan MPX5700dengan mikrokontroler ESP32. Tampilan prototipe dapat dilihat pada Gambar 6.

C. *Perancangan dan Pembuatan Perangkat Lunak (Software)*

Perancangan ini terbagi atas perancangan perangkat lunak mikrokontroller dan sensor serta perancangan aplikasi-aplikasi pendukungnya. Pada penelitian ini dibuat tampilan web sederhana yang menampilkan data sensor secara *realtime*. Website yang dibuat dengan menerapkan platform *Google IoT Core* sebagai media penghubung sekaligus tempat menyimpan, mengelola data dan mengimplementasikannya dalam bentuk website.

Flowchart adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses dan hubungan antara suatu proses dengan proses yang lain dalam suatu program. *Flowchart* aplikasi merupakan alur dari aplikasi yang akan digunakan oleh pengguna dalam memonitor tekanan angin pada ban mobil. Dibawah ini merupakan bagan *flowchart*.

Alur kerja aplikasi yang dapat dilihat pada Gambar 7 ini dimulai dengan proses menghubungkan NodeMCU ESP32 ke jaringan WiFi kemudian sensor MPX5700 mengambil data tekanan udara yang ada pada dalam ban dan data tersebut dikirimkan pada software *Google IoT Core* dan *firebase*. Data yang telah tersimpan kemudian ditampilkan pada tampilan antarmuka website.

III. HASIL DAN PEMBAHASAN

A *Implementasi Perangkat Lunak*

Antar muka atau User interface web ini menggunakan bootsrap sebagai kerangka kerja dalam pembuatan web dan *Google IoT Core* sebagai server. Beberapa fungsi yang telah kita buat seperti chart dan table akan menjadi 2 fungsi utama pada halaman dashboard utama website. Tampilan halaman utama website *monitoring* tekanan udara pada ban mobil dapat dilihat pada Gambar 8.

B. *Hasil Pengujian*

1) *Pengujian Mikrokontroler NodeMCU ESP32*

Pengujian dilakukan dengan menghubungkan mikrokontroler ESP32 pada *USB connection* PC (*Personal Computer*) menggunakan kabel USB. Apabila LED pada mikrokontroler ESP32 berkedip sekali, maka menandakan bahwa mikrokontroler NodeMCU berfungsi. Setelah melakukan pengecekan *hardware*, Kemudian dilakukan pengujian *software* mikrokontroler ESP32. Mikrokontroler ESP32 dapat dinyatakan berkerja secara baik apabila LED berkedip sesuai perintah program yang telah upload. Proses upload program pada Arduino IDE dapat dilihat pada Gambar 9.

2) *Pengujian Sensor Pressure Gauge MPX5700*

Sensor MPX5700 dihubungkan dengan ESP32 dan program yang ditulis di Arduino IDE diupload dan tunggu sebentar selama 3-10 menit untuk menghubungkan ke jaringan *cloud* menggunakan jaringan internet. Jika semuanya sudah terhubung, hasil pengukuran tekanan udara ditampilkan dalam COM Arduino IDE seperti pada Gambar 10.

Dilihat pada Gambar 10 hasil pengujian tekanan di konversikan dari kPa ke psi.Untuk perhitungan nilai tekanan dalam kPa dan psi, penulis merumuskan dimana tingkat sensivitas sensor 9.7mV dengan maksimum tekanan(P) bawaan sebesar 700 kPa (101.5 psi), raw 0.0012858 serta tegangan output berada di range 0.2 – 4.7 volt (dapat dilihat pada gambar?), maka secara teori hasil dari psi adalah $\text{Palat (psi)} = \frac{268.44 \; x \; 100}{0,7}$ = 38.92 psi

Dari hasil konversi diatas dapat disimpulkan bahwa hasil keluaran tegangan yang dihasilkan adalah adalah 268,44 kPa kemudian dikoversikan menjadi 38.92 psi. Pada hasil pengujian ini sensor dinyatakan baik digunakan.

3) *Pengujian Platform Google IoT Core*

Pembuatan website pada tugas akhir ini dibuat dengan menghubungkan *Google IoT Core* dengan menggunakan protokol MQTT dan Firebase yang dimana data sensor yang dikirimkan ke *Google IoT Core* tersimpan dalam Firebase agar data bisa di tampilkan dalam tampilan antar muka website. *Project* yang telah dibuat seperti Gambar 11 pada *Google IoT Core* dihubungkan pada firebase. Untuk dapat menghubungkan project yang telah dibuat di *Google IoT Core* ke Firebase, perlu untuk men*deploy* function yang ada di *Cloud Function.*

Sebelumnya kita harus membuat function yang baru dengan meng-create function, seperti pada Gambar 12 *function* yang telah dibuat di beri nama *SensorToFirebase* dan menulis kan source code *index.js* dan *package.*json. Pada Gambar 13 merupakan *function* yang telah dibuat dan siap dihubungkan dengan firebase. Agar data sensor dapat terhubung dengan firebase, sebelumnya kita harus membuat project pada console firebase (*console.firebase.google.com)*. *Sensortekananiot* merupakan project firebase yang telah dibuat dapat dilihat pada Gambar 14. Dan ini merupakan hasil data sensor yang terlihat pada firebase seperti pada Gambar 15.

4) *Pengujian Aplikasi Website Monitoring*

Pada aplikasi *website monitoring* ini dilakukan pengujian dengan menggunakan platform *Google IoT Core*. Data yang diambil oleh sensor akan ditampilkan pada tampilan depan website seperti pada Gambar 16. Data yang ditampilkan berupa satuan Psi yang dimana tekanan ban normal 28 – 33 Psi. dan pada Gambar 17 merupakan hasil pengeluaran data sensor yang dihubungkan pada ban dengan tekanan angin normal.

5) Pengujian Alat

Pengujian dilakukan untuk memastikan fungsi dan kinerja dari alat yang digunakan yaitu mikrokontroler ESP32 dan sensor tekanan udara MPX5700. Pengujian alat yang dibuat dibandingan dengan *Manual tire pressure gauge* agar dapat dicocokkan dengan hasil yang diperoleh dengan menggunakan alat yang dibuat penulis. Pengujian dilakukan dengan menggunakan ban normal dan ban kempes agar supaya dapat dilihat perbandingan nilai pengeluaran yang dihasilkan. Berikut merupakan tabel pengujian dari hasil alat ukur yang dibuat penulis dan alat ukur manual.

Dari hasil pengujian pada Tabel II dapat diketahui bahwa nilai tekanan yang ditetapkan pada acuan pembanding (*Manual Tire Pressure Gauge*) dengan alat yang dibuat penulis (ESP32 & MPX5700) menghasilkan pengukuran yang tidak jauh berbeda. Hal ini sesuai dengan tingkat koreksi yang dimiliki oleh sensor tekanan MPX5700 itu sendiri. Dari data diatas tekanan pada alat yang dibuat penulis pada ban normal adalah 30.23 psi, dan tekanan pada alat manual ban normal 32 psi. Sehingga hasil pengukuran keseluruhan mendekati dengan hasil yang sebenarnya.

6) Pengujian Akurasi Alat

Pada pengujian akurasi alat ini digunakan sensor tekanan udaraa MPX5700 yang dirangkai dengan mikrokontroller Node MCU ESP32. Tujuan dari pengujian akurasi alat ini adalah untuk mengetahui apakah prototype ini dapat memberkan nilai tekanan yang sesuai jika dibandingkan dengan nilai tekanan yang aktual. Pengujian dilakukan dengan menghubungkan perangkat pada pentil ban mobil untuk mengukur tekanan udara yang ada dalam ban.

Pada pengujian ini dilakukan langsung menghubungkan perangkat pada pentil ban mobil untuk mengukur tekanan udara yang ada dalam ban.

Dari hasil pengujian diatas dapat diketahui bahwa nilai tekanan yang ditetapkan pada acuan pembanding (*Manual Tire Pressure Gauge*) dengan alat yang dibuat penulis (ESP32 & MPX5700) menghasilkan pengukuran yang tidak jauh berbeda. Hal ini sesuai dengan tingkat koreksi yang dimiliki oleh sensor tekanan MPX5700 itu sendiri Sehingga hasil pengukuran keseluruhan mendekati dengan hasil yang sebenarnya.

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian mengenai Implementasi *Google Internet of Things Core* pada Monitoring Tekanan Volume Ban Mobil maka dapat disimpulkan bahwa pengukuran volume angin pada ban mobil menggunakan sensor tekanan MPX5700 cukup baik dalam hal pengukuran volume angin. Dan data sensor yang di kirim ke *Google IoT Core* dapat disimpan pada *Pub/Sub* dan selanjutnya di tampilkan pada aplikasi monitoring yang di telah dibuat. Namun dalam menghubungkan perangkat dengan internet, membutuhkan koneksi internet yang stabil agar data yang akan ditampilkan akurat. Dan wadah pelindung perangkat yang digunakan masih perlu di tata kembali.

B. Saran

Berdasarkan kesimpulan diatas, maka disarankan untuk dilakukan penelitian lebih lanjut dengan mempertimbangkan beberapa faktor seperti koneksi jaringan internet yang stabil agar data yang ditampilkan akurat, kemudian wadah yang merupakan pelindung perangkat perlu di kemas lebih menarik dan aman agar terhindar dari hal yang tidak diinginkan.

V. KUTIPAN

[1] Fitri, “Perancangan Website Penjualan Online pada Queensha Boutiquen,” Universitas Komputer Indonesia, 2013.

[2] S. Michael, “RANCANG BANGUN SMART RICE COOKER MENGGUNAKAN PROTOKOL KOMUNIKASI WIFI DAN PROTOKOL PERTUKARAN PESAN MQTT,” Universitas Multimedia Nusantara, 2016.

[3] Dugi, “*Cloud Iot Core*,” *www.du-gi.com*, 2019. https://du-gi.com/dugi/google-cloud-platform-gcp/cloud-iot-core/.

[4] A. Conan, “SISTEM MONITORING DAN EVALUASI PENGELOLAAN PROGRAM STUDI BERBASIS BAN-PT,” Universitas Atma Jaya Yogyakarta, 2017.

[5] Wikipedia, “C++,” *www.wikipedia.org*, 2019. https://id.wikipedia.org/wiki/C%2B%2B.

[6] Juliani, “RANCANG BANGUN ALAT UKUR KETEBALAN KAYU MENGGUNAKAN TAMPILAN LCD BERBASIS ARDUINO,” Universitas Sumatera Utara, 2016.

[7] Z. A. Rozi and SmitDev, *Bootstrap Design Framework*. Jakarta: PT Elex Media Komputindo, 2015.

[8] W. Desta, “IMPLEMENTASI TEKNOLOGI FIREBASE PADA APLIKASI PENCARIAN LOKASI SERVICE KAMERA BERDASARKAN RATING BERBASIS ANDROID,” Sekolah Tinggi Manajemen Informatika dan Komputer Akakom Yogyakarta, 2017.

[9] W. Ida, “SISTEM IOT BERBASIS PROTOKOL MQTT DENGAN MIKROKONTROLER ESP8266 DAN ESP32,” in *Posiding SNATIF*, Denpasar Bali, 2018, pp. 324–334.

[10] A. Fandy, “PERANCANGAN ALAT UKUR TEKANAN UDARA DENGAN MENGGUNAKAN SENSOR PRESSURE GAUGE MPX5700,” Universitas Sumatera Utara, 2018.

[11] A. Tio, “ANALISIS FRONT WHEEL ALLIGMENT (FWA) PADA KENDARAAN DAIHATSU GRAN MAX PICK UP,” Universitas Pendidikan Indonesia, 2015.

Penulis bernama lengkap Praisye Evangelista Anastasia Kaunang anak ke tiga dari empat bersaudara, lahir di Manado pada tanggal 11 Agustus 1998. Penulis menempuh pendidikan pertama di SD GMIM 13 Manado (2003-2009), kemudian melanjutkan ke SMP Katolik ST. Rafael Manado (2009-2012), setelah itu melanjutkan sekolah di SMA Negeri 9 Manado (2012-2015). Tahun 2015, penulis melanjutkan studi di Fakultas Teknik, Jurusan Teknik Elektro Program Studi Teknik Informatika, Universitas Sam Ratulangi Manado Sulawesi Utara. Pada bulan Januari tahun 2019 Penulis mengajukan proposal Skripsi untuk memenuhi syarat meraih gelar sarjana (S1) dengan judul Implementasi *Google Internet of Things Core* pada Monitoring Tekanan Volume Ban Mobil yang kemudian disetujui dan dibimbing oleh dua dosen pembimbing yaitu Sherwin R.U.A Sompie,ST,MT dan Ir. Arie S.M. Lumenta, ST,MT.