IMPLEMENTASI SISTEM PEMANTAUAN KONDISI KENDARAAN RODA EMPAT DENGAN MENGGUNAKAN ON BOARD DIAGNOSTIC-II

IMPLEMENTATION ON BOARD DIAGNOSTIC (OBD-II) FOR AUTO MOBILE MONITORING CONDITION SYSTEM

Mochammad Rachmandani, Ir. Ahmad Tri Hanuranto M.T[2], Dr. Nyoman Bogi A. Karna, MSEE[3]

[1,2,3] Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Telkom University, Bandung

[1]mochrachmandani@student.telkomuniversity.ac.id,
{[2]Athanuranto [3]aditya}@telkomuniversity.ac.id}

Abstrak

Salah satu perkembangan teknologi dalam bidang otomotif adalah dengan hadirnya *On Board Diagnostic-II* (OBD-II) yang memudahkan transfer data dari *Engine Control Unit* (ECU) yang terdapat sensor-sensor yang mengatur ke listrikan mobil seperti pada bagian RPM, temperatur, *ignation*, transmisi dan yang lainnya. Komunikasi antara ECU dan OBD-II diperlukan sebuah protokol dimana tiap standar protokol berbeda-beda tergantung dari tipe dan jenis kendaraan itu sendiri.

Kerusakan pada kendaraan adalah kondisi yang pasti terjadi. Hal tersebut disebabkan oleh beberapa faktor, dan untuk mengetahui faktor penyebab kerusakan kendaraan membutuhkan waktu yang lama. Pada penelitian ini dibangun sebuah sistem yang dapat digunakan pengemudi untuk mengetahui kondisi kendaraannya serta dapat melakukan diagnosa dan menentukan kondisi pada kendaraan. Implementasi dari OBD-II juga bisa disangkutkan dengan *Internet of Thing* (IoT) dimana data yang direkam oleh OBD-II bisa dikirikan ke *cloud server*.

Dengan memanfaatkan *On Board Diagnostic* berhasil mengirim dan menyimpan data yang sudah terbaca kedalam *database*. Proses perekaman data dapat dilakukan akan tetapi terdapat selisih nilai yang didapat antara serial monitor dengan *database* pada setiap parameternya. Seperti pada *Load* memiliki nilai *average* sebesar 6,9 dengan *standar error* sebesar 2,62 untuk parameter *temperature* memiliki nilai *average* sebesar 0,46 dengan *standar error* 0,22 untuk parameter RPM dan *Speed* memiliki *average* 149 km/h dan 4,7 km/h dan *standar error* 49,9 dan 1,86 dan untuk parameter *Throttle* memiliki *average* 1,16 dengan *standar error* 0,33.

Kata Kunci: *On Board Diagnostics, Database, Engine Control Unit*

Abstract

One of the technological developments in the automotive field is the presence of the On Board Diagnostic-II (OBD-II) which facilitates the transfer of data from the Engine Control Unit (ECU) which has sensors that regulate the car's electricity such as RPM, temperature, ignation, transmission and whatnot. Communication between ECU and OBD-II requires a protocol where each protocol standard varies depending on the type and type of vehicle itself.

Damage to the vehicle is a condition that is certain to occur. This is caused by several factors, and to find out the factors that cause damage to the vehicle takes a long time. In this study a system was built that can be used by the driver to determine the condition of the vehicle and can diagnose and determine the condition of the vehicle. Implementation of OBD-II can also be linked to Internet of Thing (IoT) where data recorded by OBD-II can be typed into the cloud server.

The process of recording data can be done but there is a difference in the value obtained between the serial monitor and the database on each parameter. As with Load has an average value of 1.2 with a standard error of 0.39 for temperature parameters having an average value of 0.46 with a standard error of 0.22 for RPM and Speed parameters having an average of 45,9 km/h and 2.73 km/h and standard error 16,99 and 0,83 and for Throttle parameters have an average of 1.16 with a standard error of 0.33.

Keywords: *On Board Diagnostics, Database, Engine Control Unit*

1. Pendahuluan

Banyaknya faktor penyebab kecelakaan lalu lintas, seperti perilaku berkendara, lingkungan jalan yang kurang baik, atau kondisi kendaraan yang tidak terawat den-gan benar. Menurut Database Komite Nasional Keselamatan Transportasi 31 Oktober 2016 faktor manusia menjadi penyebab yang paling tinggi dalam kecelakaan lalu lintas dan angkutan jalan [1].

Di era modern ini, kendaraan bermotor khususnya kendaraan beroda empat sudah dilengkapi *dengan Engine Control Unit* (ECU) serta berbagai macam sensor elektronik yang lainnya. Pada dasarnya *On Board Diagnostics* (OBD) terdiri dari ECU yang menggunakan masukkan dari beberapa sensor untuk mengontrol akuator dan mendapatkan performa yang diinginkan[2].

Salah satu implementasi OBD adalah untuk *safety driving* dengan cara memantau dan menganalisis *driving style* pengendara. Khususnya untuk keadaan lalu lintas di Indonesia yang sulit diprediksi karena beberapa faktor salah satunya banyak pengendara yang ugal-ugalan sehingga angka kecelakan lalu lintas di Indonesia menjadi salah satu yang tertinggi di dunia [3].

Dalam daftar pustaka dapat ditemukan beberapa karya tulis yang memanfaatkan fitur *On Board Diagnostics* II (OBD-II) pada kendaraan roda empat. Pada pustaka[4] penelitian tersebut mencoba membuat sistem seperti pada *black box* pesawat. Sistem ini dapat merekam data kendaraan berupa posisi *Load, Temperature, Speed*, RPM dan *Throttle* yang disimpan pada *Database* yang terhubung ke *Web Server*.

Salah satu faktor yang sering menjadi penyebab kecelakaan di jalan raya adalah pengendara kendaraan bermotor yang mengendarai kendaraannya melebihi batas kecepatan yang diperbolehkan [5]. Menanggapi hal ini, pihak berwenang harus mengambil tindakan tegas dengan memberi sanksi hukum yang sesuai hukum yang berlaku. Salah satu cara yang dapat dilakukan untuk membuktikan kesalahan pengendara adalah dengan mengukur kecepatan kendaraan yang melintas di jalan raya sehingga bisa diketahui pengendara tersebut melebihi batas kecepatan yang di-zinkan atau tidak [8]

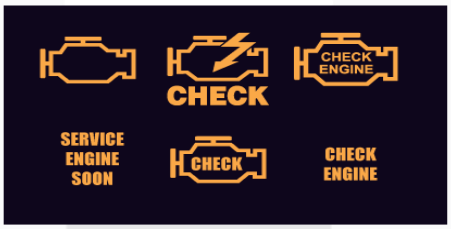
Dengan menggunakan metode *data acquisition* informasi dari ECU yang diterima OBD-II akan disimpan oleh *data acquisition system* (DAS) parameter-parameter yang telah direkam menggunakan OBD dikirimkan ke *cloud server* untuk disimpan pada *database*.

2. Dasar Teori

2.1 On Board Diagnostic

Pada awal 1980-an pabrikan kendaraan bermotor khususnya kendaraan beroda empat sudah mulai berkembang salah satunya dengan mengimplementasikan berbagai perangkat elektronik pada kendaraannya. Tujuan utamanya adalah sebagai menekan emisi gas buang guna mengefisiensikan bahan bakar. Dengan tujuan tersebut pabrikan kendaraan bermotor menambahkan sensor-sensor yang dapat memonitoring emisi gas buang, selain itu sensor tersebut juga dapat membantu dalam diagnosa kerusakan pada mesin kendaraan [2].

Output dari sistem OBD yang disajikan kepada driver berupa lampu peringatan dengan simbol mesin yang dikenal sebagai lampu indikator kerusakan (MIL) yang akan menyala apabila terjadi kerusakan yang disimpan dalam ECU (*Engine Control Unit*). Terdapat 16 pin konektor data link (DLC) yang menyimpan informasi dalam sistem OBD yang biasanya terletak pada kabin kendaraan [7].



Gambar 1: Mulfunction Indicator Light

2.1.1 Protokol Komunikasi OBD

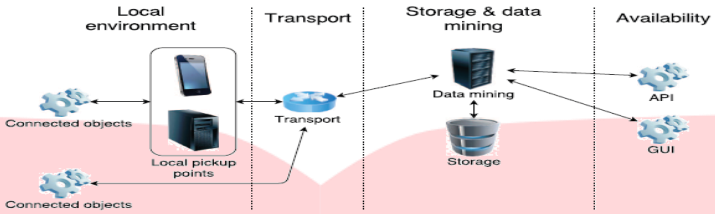
Agar dapat berkomunikasi dengan ECU, terdapat 5 (lima) protokol yang dapat diimplementasikan oleh pabrikan kendaraan yang diproduksi, yaitu [7]:

Standard	Pin2	Pin6	Pin7	Pin10	Pin14	Pin15
J1850 PWM	X	-	-	X	-	-
J1850 VPW	X	-	-	-	-	-
ISO15675(CAN)	-	X	-	-	X	-
ISO9141/14230	-	-	X	-	-	opsional

Gambar 2: Protokol Komunikasi OBD

2.2 Internet Of Things (IoT)

Sebuah hal yang wajar bahwa konsep IoT sekarang memang sedang popular bahkan banyak perusahaan–perusahaan yang sekarang sudah berbasis IoT karena dapat memudahkan suatu pekerjaan dan bisa dijadikan sebagai lahan bisnis juga. Kasarnya IoT bisa didefinisikan sebagai komunikasi antar 2 mesin yang bisa saling bertukar informasi secara mandiri dimana tiap objek terasosiasi satu sama lain dalam sebuah arsitektur sistem data dimana tiap bagian perlu transportasi, penyimpanan, proses, ketersediaan [9]. Tak heran IoT pun merambat ke ranah otomotif menjadikan kendaraan–kendaraan sekarang khususnya mobil sudah bisa terintegrasi dengan internet atau isitilahnya biasa disebut *Internet of Vehicle* (IoV) [8].



Gambar 3: Arsitektur IoT

2.3 *Embedded System*

Embedded System adalah kombinasi antara perangkat keras dan perangkat lunak yang didesain dengan fungsi spesifik untuk sistem yang lebih besar seperti mesin industri, kendaraan bermotor, peralatan medis, kamera, dan lainnya. Karena didesain hanya untuk fungsi spesifik, untuk mendesain embedded system perlu mempertimbangkan ukuran, konsumsi daya, keandalan sistem dan performansi [9]. *Embedded system* memiliki sifat dasar layaknya komputer, yaitu menerima input,memproses input, dan menghasilkan *output*. Pada umumnya, input pada *embedded system* berupa sensor-sensor tertentu tergantung kebutuhan sistem. *Input* kemudian akan diproses mikrokontroler/mikroprosesor yang sudah diprogram untuk fungsi spesifik. *Output* pada *embedded system* bisa berupa aktuator atau hanya berupa indikator [9]. *Embedded system* bisa berfungsi dengan sedikit atau tanpa kontak dengan manusia. Pemrograman *embedded system* menggunakan komputer untuk menulis program. Bahasa Pemrograman pada *embedded system* umumnya menggunakan beberapa bahasa, seperti assembly, C, atau phyton dengan menggunakan *compiler* sesuai dengan *platform* yang dipakai [9].

2.4 PHP *(Hypertext Preprocessor)*

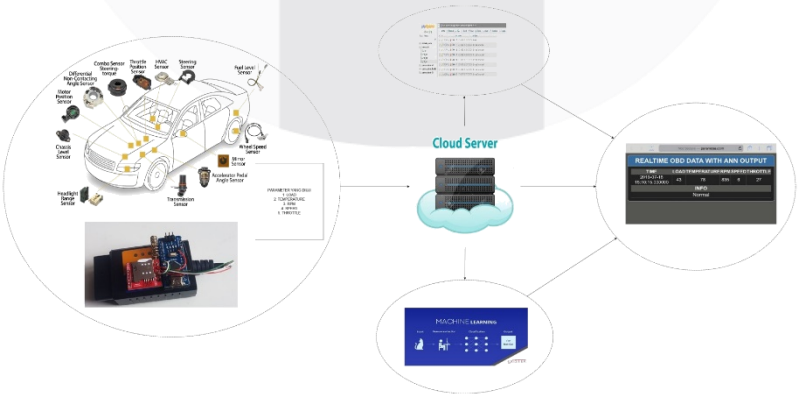
Hypertext Preprocessor (PHP) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis. Karena PHP merupakan *server-side scripting* maka sintaks dan perintah perintah PHP akan dieksekusi diserver kemudian hasilnya dikirimkan ke *browser* dalam format HTML. Dengan demikian, kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman *web* lebih terjamin [10].

2.5 MySQL *Database*

My Stucture Query Language atau MySQL merupakan sebuah perangkat lunak *(software)* sistem manajemen basis data SQL atau DBMS *multi thread* dan *multi user* yang merupakan turunan dari salah satu konsep utama dalam *database* untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan secara mudah dan otomatis.

3. Sistem yang Dibangun

3.1 Skema Utama



Gambar 4: Skema Utama

Sistem yang akan dibangun dalam tugas akhir ini merupakan implementasi *car recorder* pada kendaraan roda empat yang memanfaatkan fitur dari *on board diagnostic* pada kendaraan roda empat. *Prototipe car recorder* mempunyai fungsi merekam sensor kendaraan yang telah ditentukan pada batasan masalah. Dalam tugas akhir ini membahas mengenai bagaimana proses perekaman menggunakan *on board diagnostic* sampai proses penyimpanan data pada *database*

3.2 Komponen Penyusun

Komponen penyusun terdiri dari perangkat keras *(hardware)* dan perangkat lunak *(software*):

3.2.1 Perangkat Keras yang dibutuhkan:

1. ELM 327 merupakan perangkat mikrokontroller yang memungkinkan untuk mendapatkan data kendaraan melalui OBD II yang terintegrasi dengan *bluetooth.*
2. Arduiono Nano adalah salah satu papan pengembangan mikrokontroler yang berukuran kecil, lengkap dan mendukung penggunaan *breadboard*. Arduino Nano diciptakan dengan basis mikrokontroler Atmega328 (untuk ArduinoNano versi 3.x) atau Atmega168 (untuk Arduino versi 2.x). Arduino Nano dirancang dan diproduksi oleh perusahaan Gravitech.
3. Modul GSM Sim800L adalah perangkat yang bisa digunakan untuk meng-gantikan fungsi handphone. Untuk komunikasi data antara sistem jaringan seluler, maka digunakan Modul GSM SIM800 yang digunakan sebagai media pengiriman data ke *database* melalui *server.*

3.2.2 Perangkat Lunak yang dibutuhkan:

1. Arduino IDE digunakan untuk memprogram mikrokontroller yang digunakan pada tugas akhir ini yaitu arduino nano
2. Domainesia sebagai local hosting yang digunakan untuk program MySQL.
3. MySQL adalah *relational database management system* (RDBMS) *opensource* berdasarkan *structure query language*. MySQL berjalan secara virtual pada beberapa platform seperti, Linux, UNIX, dan windows. Pada tugasakhir ini, MySQL digunakan sebagai database yang telah terinstall pada *server* untuk menyimpan data.

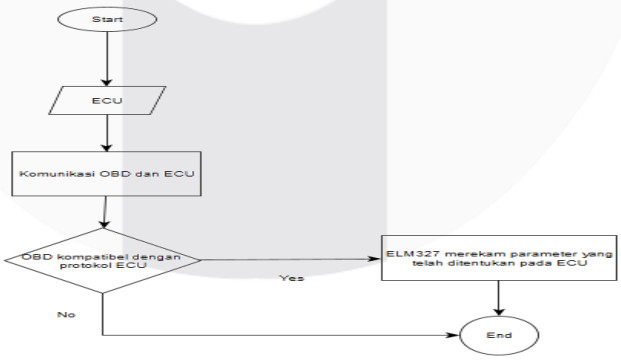
3.3 Skenario Sistem

Skenario pengujian dilakukan agar dapat mengetahui akurasi serta kinerja darisistem perekaman hingga pengiriman data. Pada sistem ini terbagi menjadi 3 fungsiyaitu sistem dapat berkomunikasi dengan dengan ECU kendaraan, Sistem dapat membaca data dari ELM327 sesuai dengan parameter yang sudah ditentukan, dan sistem dapat mengirim dan menyimpan data yang dibaca ke *Cloud*. Skenario pertama yang dilakukan yaitu membaca data sesuai parameter yang sudah ditentukan, setelah skenario pertama berjalan maka masuk ke skenario ke kedua yaitu data yang sudah dibaca sesuai parameter yang ditentukan akan dikirim ke dalam internet menggunakan *server Cloud*, data yang dikirim berupa *Load, Temperature, RPM,Speed, Throttle* kemudian data yang sudah didapatkan dikirim ke *Cloud* untuk memantau kondisi kendaraan.

Pengujian rekam data dilakukan dengan membandingkan data yang ada padakendaraan dan data yang disimpan pada *database*. Perekaman data kendaraan menggunakan Arduino IDE dan berdasarkan jarak tempuh kendaraan pada saat perekaman.

3.4 Fungsi dan Sistem

3.4.1 Pengambilan Data Kendaraan

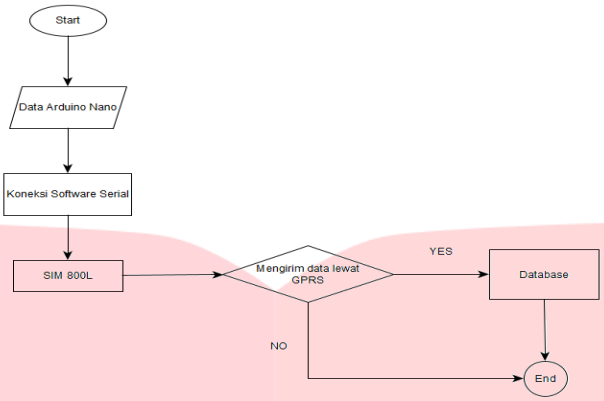


Gambar 5: Proses Pengambilan Data

Pada gambar 5 merupakan proses pengambilan data kendaraan dimulai saat ELM327 *pluged* dengan socket OBD. Dimana, pada ELM327 terdapat beberapa sensor untuk mendeteksi .Apabila protokol kompatibel maka ELM327 akan merekam data pada ECU yang akan menjadi parameter. Jika protokol antara ELM327 dan ECU tidak kompatibel maka proses rekam data tidak dapat terjadi. Proses rekam data ECU harus pada kondisi kendaraan *ignation on* agar ELM327

bisa terkoneksi. Proses pengambilan data dari ECU berupa *Load, Temperature, RPM, Speed, Throttle.*

3.4.2 Pengiriman Data



Gambar 6: Pengiriman Data

Pada gambar 6 merupakan proses pengiriman data sampai ke *database*, data yang sudah didapatkan dari ELM327 dikirimkan ke *cloud* dan harus tersedia jaringan internet melalui transmisi GPRS melalui komunikasi *software* serial selanjutnya data akan dikirim ke server atau internet melalui GPRS dengan menggunakan paket data menggunakan *simcard* yang terpasang pada SIM800L. Jika *simcard* tidak memiliki paket data maka tidak akan terjadi proses pengiriman data ke server.

4. Hasil dan Analisis

. Pada bab ini akan dibahas hasil pengujian dari skenario pengujian yang telah ditulis pada bab sebelumnya.

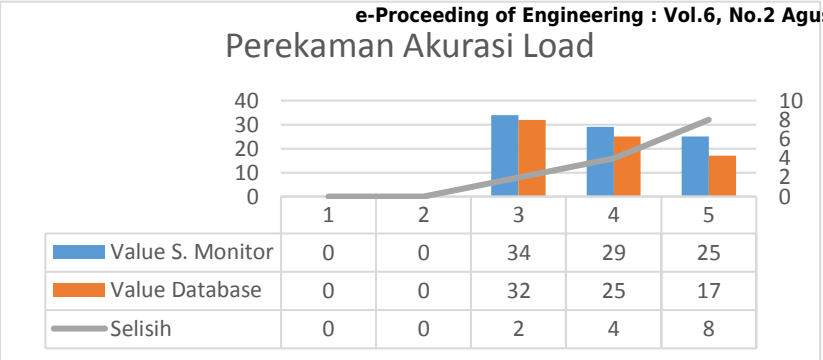
4.1 Perekaman Akurasi Load

Tabel 1: Hasil Pengujuan Load

Perekaman Akurasi Load				
No	Time	Value S. Monitor	Value Database	Selisih
1	18:51:26	0	0	0
2	18:51:47	0	0	0
3	18:57:08	34	32	2
4	18:57:25	29	25	4
5	18:58:24	25	17	8

Tabel 2: Selisih Nilai Load

Selisih Nilai Load	
Average	1,2
Max	8
Min	0
Std Deviasi	2,187740135
Std Error	0,399424874



Gambar 7: Akurasi Load

Dari Gambar 7 didapatkan nilai selisih dari serial monitor dan server didapatkan *average* sebesar 6,9 dengan standar *error* sebesar 2,62.

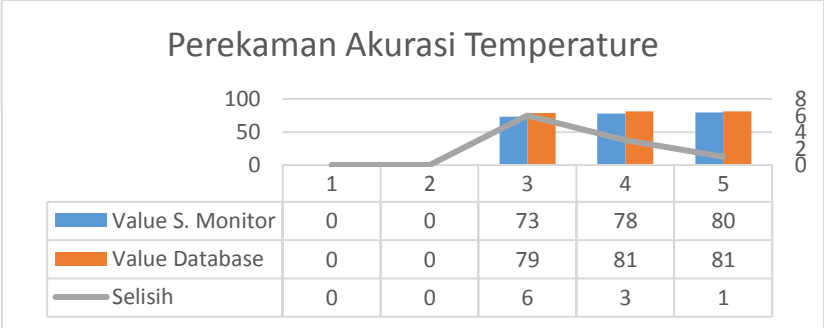
4.2 Perekaman Akurasi Temperature

Tabel 3: Hasil Pengujian Temperature

Perekaman Akurasi Temperature				
No	Time	Value S. Monitor	Value Database	Selisih
1	18:51:26	0	0	0
2	18:51:47	0	0	0
3	18:57:00	73	79	6
4	18:57:32	78	81	3
5	18:57:56	80	81	1

Tabel 4: Selisih Nilai Temperature

Selisih Nilai Temperature	
Average	0,466666667
Max	6
Min	0
Std Deviasi	1,252124631
Std Error	0,228605635



Gambar 8: Akurasi Temperature

Berdasarkan Gambar 8 terdapat perbedaan data yang diterima pada Serial Monitor dengan data yang diterima pada Server. Nilai selisih didapatkan *Average* sebesar 0,46 dengan standar *error* sebesar 0.2.

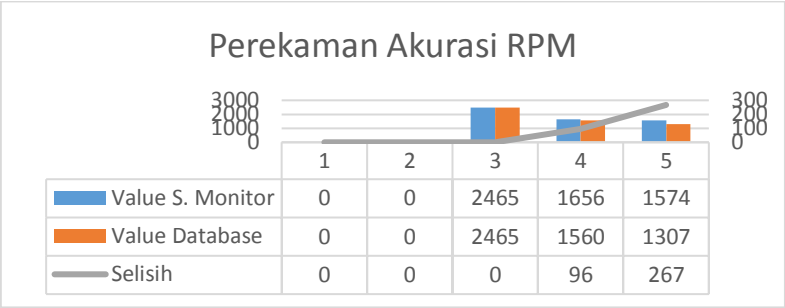
4.3 Perekaman Akurasi RPM

Tabel 5: Hasil Pengujian RPM

Perekaman Akurasi RPM				
No	Time	Value S. Monitor	Value Database	Selisih
1	18:51:26	0	0	0
2	18:51:47	0	0	0
3	18:55:56	2465	2465	0
4	18:56:29	1656	1560	96
5	18:56:38	1574	1307	267

Tabel 6: Selisih Nilai RPM

Selisih Nilai RPM	
Average	45,9
Max	384
Min	0
Std Deviasi	93,06406766
Std Error	16,99109638



Gambar 9: Akurasi RPM

Berdasarkan Gambar 9 terdapat perbedaan data yang diterima pada Serial Monitor dengan data yang diterima pada Server. Nilai selisih didapatkan *Average* sebesar 149 dengan standar *error* sebesar 49,9

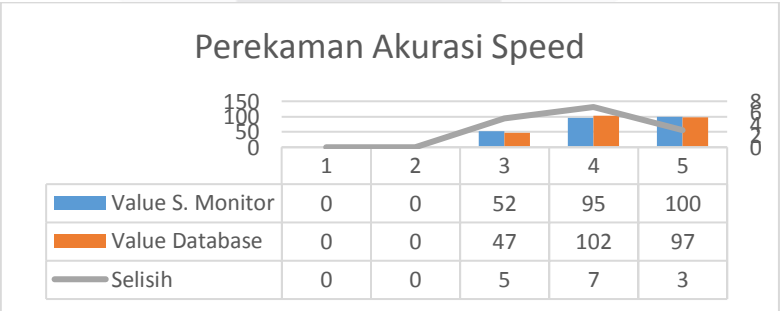
4.4 Perekaman Akurasi Speed

Tabel 7: Hasil Pengujian Speed

Perekaman Akurasi Speed				
No	Time	Value S. Monitor	Value Database	Selisih
1	18:51:26	0	0	0
2	18:51:47	0	0	0
3	18:58:57	52	47	5
4	19:01:18	95	102	7
5	19:01:36	100	97	3

Tabel 8: Selisih Nilai Speed

Selisih Nilai Speed	
Average	2,733333333
Max	16
Min	0
Std Deviasi	4,563221286
Std Error	0,833126411



Gambar 10: Akurasi Speed

Berdasarkan Gambar 10 didapatkan nilai *average Speed* sebesar 4,7 dengan standar *error* sebesar 1,86

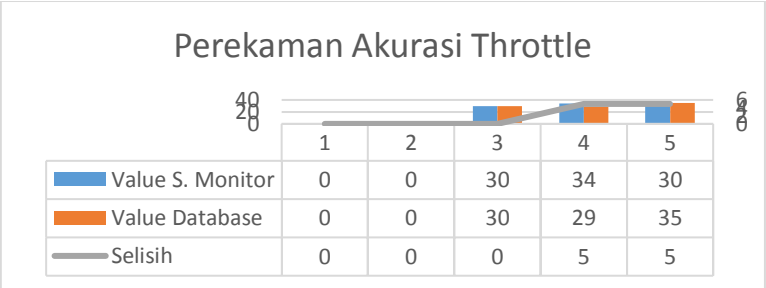
4.5 Perekaman Akurasi Throttle

Tabel 9: Hasil Pengujian Throttle

Perekaman Akurasi Throttle				
No	Time	Value S. Monitor	Value Database	Selisih
1	18:51:26	0	0	0
2	18:51:47	0	0	0
3	18:57:00	30	30	0
4	18:57:32	34	29	5
5	18:57:56	30	35	5

Tabel 10: Selisih Nilai Throttle

Selisih Nilai Throttle	
Average	1,166666667
Max	5
Min	0
Std Deviasi	1,858500199
Std Error	0,339314161



Gambar 11: Akurasi Throttle

Berdasarkan Gambar 11 terdapat perbedaan data yang diterima pada Serial Monitor dengan data yang diterima pada server dengan *average* sebesar 1,16 dan standar *error* sebesar 0,33

5. Kesimpulan

Berdasarkan perancangan dan pengimplementasian telah dilakukan pengujian dengan berbagai skenario pengujian dapat ditarik kesimpulan sebagai berikut:

1. Pada pengujian dengan berbagai skenario dengan memanfaatkan *On Board Diagnostics* (OBD-II) telah terbukti mampu merekam data Load, Temperature, RPM, Speed, dan Throttle.
2. Sistem berhasil mengirim dan meyimpan data yang sudah terbaca ke dalam *database*
3. Proses perekaman data dapat dilakukan akan tetapi terdapat selisih nilai yang didapat antara serial monitor dengan *server* pada setiap parameternya. Seperti pada Load memiliki nilai *average* sebesar 6,9 dengan standar *error* sebesar 2,62 untuk parameter temperature memiliki nilai *average* sebesar 0,46 dengan standar error 0,22 untuk parameter RPM dan *Speed* memiliki *average* 149 km/h dan 4,7 km/h dan standar *error* 49,9 dan 1,86 dan untuk parameter *Throttle* memiliki *average* 1,16 dengan standar *error* 0,33

6. Daftar Pustaka

[1] J. Lin, S. Chen, Y. Shih, and S.-H. Chen, "A study on remote on-line diagnosticsystem for vehicles by integrating the technology of obd, gps, and 3g, "World Academy of Science, Engineering and Technology, 2009

[2] S. A. Nugroho, E. Ariyanto, and A. Rakhmatsyah, "*Utilization of onboard di-agnostic ii (obd-ii) on four wheel vehicles for car data recorder prototype*," in 2018 *6th International Conference on Information and Communication Technology* (ICoICT). IEEE, 2018.

[3] S. Kaplan, M. A. Guvensan, A. G. Yavuz, and Y. Karalurt, "*Driver behavior analysis for safe driving: A survey*,"*IEEE Transactions on Intelligent Transportation Systems*, 2015

[4] N. Kaur and S. Monga, *"Comparisons of wired and wireless networks: Areview,"International Journal of Advanced Engineering Technology*, 2014

[5] D. Hui, H. Bo, W. Dafang, and Z. Guifan, *"The ecu control of diesel enginebased on can," in2011 Fourth International Conference on Intelligent Computation Technology and Automation*. IEEE, 2011.

[6] A. Muchta, "Kelistrikan mesin,"https://www.autoexpose.org/2017/10/apa-itu-ecu-dan-apa-fungsi-ecu-mobil.html, *accessed*: 2019-05-23.

[7] A. Pereira, M. Alves, and H. Macedo, *"Vehicle driving analysis in regardsto fuel consumption using fuzzy logic and obd-ii devices,"* in 2016 *8th Euro American Conference on Telematics and Information Systems* (EATIS). IEEE,2016

[8] B. Dorsemaine, J.-P. Gaulier, J.-P. Wary, N. Kheir, and P. Urien, *"Internet of things: a definition & taxonomy," in2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*. IEEE,2015.

[9] P . R. Burje, K. J. Karande, and A. B. Jagadale, *"Embedded onboard diagnostics system using can protocol," in 2014 International Conference on Communication and Signal Processing*. IEEE, 2014.

[10] E. Winarno and A. Zaki, *"Easy web programming with php plus html 5*," *Elex Media Komputindo*, 2011.

[11] Komputer, "Pengertian mysql," http://www.teorikomputer.com/2015/10/pengertian-mysql-beserta kelebihan dan.html, accessed: 2019-05-23.

[12] P. Wei, Z. Hong, and M. Shi, *"Performance analysis of http and ftp based on opnet,"* in2016 IEEE/ACIS 15th *International Conference on Computer and Information Science* (ICIS). IEEE, 2016.