

## IMPLEMENTASI ON BOARD DIAGNOSTIC II UNTUK PENGUMPULAN DATA PADA KENDARAAN RODA EMPAT

### DATA COLLECTION SYSTEM FOR AUTOMOBILE BASED ON BOARD DIAGNOSTIC-II INTERFACE

Muhammad Rasyad Mustafa, Ir. Ahmad Tri Hanuranto M.T<sup>2</sup>, Dr. Nyoman Bogi A. Karna, MSEE<sup>3</sup>

<sup>1,2,3</sup> Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Telkom University, Bandung

<sup>1</sup>rasyadz@student.telkomuniversity.ac.id,

{<sup>2</sup>Athanuranto,<sup>3</sup>nyoman.bogi}@telkomuniversity.ac.id}

#### Abstrak

Beberapa kejadian kecalakan lalu lintas terjadi karena kerusakan pada mesin mobil dan hal tersebut tanpa diketahui sebelumnya oleh pengemudi mobil tersebut. Di era moderen ini internet menjadi hal yang lumrah di setiap kalangan pun sudah berdampingan dengan internet mulai dari anak-anak hingga orang dewasa. Implementasi dari OBD-II juga bisa disangkutkan dengan adanya *Internet of Thing* (IoT) perkembangan teknologi pun semakin maju salah satunya di bidang otomotif. *On Board Diagnostic II* (OBD-II) adalah sebuah soket dalam kendaraan beroda empat (mobil) yang berfungsi untuk mengetahui kondisi mesin mobil tersebut melalui *Engine Control Unit* (ECU) pada mobil.

Pada sistem dalam tugas akhir ini dengan memanfaatkan fitur *On Board Diagnostic - II* (OBD-II) untuk membaca parameter pada data *Engine Control Unit* (ECU) mobil melalui protokol yang sesuai dengan ECU mobil tersebut lalu data akan direkam oleh ELM327 dan diteruskan ke Arduino Nano untuk ditranslasikan dari data yang telah direkam oleh ELM327. Lalu data akan dikirimkan ke internet atau server melalui GPRS dengan menggunakan SIM 800L untuk kebutuhan analisa lebih lanjut.

Dalam implementasi ini manfaat yang diharapkan yaitu ELM327 berhasil merekam parameter pada ECU dan data berhasil dikirimkan ke cloud server melalui *General Packet Radio Service* (GPRS) agar nanti pada pengembangan studi ini *user* bisa mengetahui informasi tentang kondisi mobilnya berdasarkan parameter yang diperoleh dan agar bisa dianalisa lebih lanjut lagi pada pengembangannya kedepannya.

**Kata Kunci:** OBD-II, ECU, Internet

#### Abstract

A number of traffic accidents occur because of damage to the car's engine and it is unknown to the driver of the car. In this modern era the internet has become a common thing in every circle that has side by side with the internet ranging from children to adults. The implementation of OBD-II can also be linked to the existence of the Internet of Thing (IoT), the development of technology is increasingly advanced one of them in the automotive sector. *On Board Diagnostic II* (OBD-II) is a socket in a four-wheeled vehicle (car) that functions to determine the condition of the car's engine through the Engine Control Unit (ECU) on the car.

In the system in this final project by utilizing the *On Board Diagnostic - II* (OBD-II) feature to read the parameters in the car's Engine Control Unit (ECU) through a protocol that matches the car's ECU, the data will be recorded by ELM327 and forwarded to Arduino Nano to be translated from data that has been recorded by ELM327. Then the data will be sent to the internet or server via GPRS by using the 800L SIM for further analysis needs.

In this implementation the expected benefits, namely ELM327, succeeded in recording parameters on the ECU and the data was successfully sent to the cloud server through General Packet Radio Service (GPRS) so that later in the development of this study the user could find out information about the condition of the car based on parameters obtained and be analyzed go further on the future development.

**Keywords:** OBD-II, ECU, Internet

## 1. Pendahuluan

Menurut “Peraturan kepala kepolisian negara republik Indonesia tentang tata cara penanganan kecelakaan lalu lintas” pasal 1 ayat 3 yang berbunyi kecelakaan lalu lintas adalah suatu peristiwa di jalan yang tidak diduga dan tidak disengaja melibatkan kendaraan dengan atau tanpa pengguna jalan lain yang mengakibatkan korban manusia dan/atau kerugian harta benda [3]. 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 masukan dari beberapa sensor untuk mengontrol aktuator dan mendapatkan performa yang diinginkan [4].

Di era modern ini perkembangan internet sangatlah pesat banyak aspek kehidupan masyarakat pun tidak terlepas dari peran internet. Di Indonesia sendiri istilah *Internet of Thing* (IoT) sudah bukan hal yang aneh lagi, Bahkan sekarang pun kendaraan roda empat bisa terintegrasi dengan IoT atau istilahnya dengan sebutan *Internet of Vehicle* (IoV) dengan menggunakan alat *On Board Diagnostic-II* (OBD-

II) sebagai interface dari *Electronic Control Unit* (ECU) agar bisa mendapatkan informasi dari mobil tersebut. Umumnya OBD-II bekerja pada layer 4 atau trans- port layer dan layer 3 atau network layer pada *Control Area Network* (CAN) suatu kendaraan [5].

OBD-II pertama kali dikembangkan pada tahun 1996 oleh SAE, EPA, CARB di Amerika serikat sistem tersebut menjadi sebuah standar bagi mayoritas mobil di Amerika Serikat. Sistem tersebut berfungsi untuk memantau parameter-parameter yang beda dari mobil lalu akan disimpan dalam memori bila terjadi sesuatu sistem akan memperingatkan user melalui *Malfunction Indication Light* (MIL) [6]. Di tahun 2008 CAN ditambah untuk menjadi protokol dalam OBD-II sehingga kompatibel di mobil-mobil yang sekarang.

Contoh pemanfaatan OBD-II yaitu untuk mendiagnosis keadaan mesin mobil tanpa harus membawanya ke bengkel salah satunya dengan menggunakan *Data Acquisition System* (DAS) data dari OBD-II yang berisikan informasi status mobil akan direkam dan akan muncul di display LCD secara real time [7].

Diharapkan dari tugas akhir ini permasalahan tersebut butuh suatu sistem agar informasi dari ECU bisa dikirimkan ke cloud server untuk diolah yang hasilnya nanti bisa membantu user untuk mengetahui status mobilnya tanpa harus pergi ke bengkel terlebih dahulu.

## 2. Dasar Teori

### 2.1 *Internet of Thing* (IOT)

Sebuah hal yang wajar bahwa konsep IoT sekarang memang sedang populer 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 [1].

### 2.2 OBD-II

OBD-II pertama kali diperkenalkan pada pertengahan tahun 90-an dimana pada pemakaian sebelumnya OBD berfungsi sebagai solusi untuk masalah emisi gas buang kendaraan dan pemakaian bahan bakar agar lebih irit. Lalu pada tahun 1996 OBD II pertama kali muncul dimana fungsinya lebih canggih lagi yaitu sudah bisa membaca informasi dari ECU mobil serta memonitor beberapa parameter juga salah satunya yaitu device pada mobil, body, chasis juga OBD-II sekarang sudah bisa terhubung jaringan internet dengan menggunakan 4G maupun Wifi [6]. Cara kerja dari OBD-II yaitu sensor akan membaca ECU dengan parameter yang berbeda yaitu meliputi status baterai, temperature, level bahan bakar, RPM, waktu sejak mesin mulai dan total pengoperasian mesin dalam jam. Beberapa parameter akan langsung terbaca dari kendaraan dan data yang masuk akan diolah untuk diproses dan akan memantau sensor serta parameter yang lainnya juga setelah itu data yang masuk akan disimpan dalam suatu memori [8].

### 2.3 *Engine Control Unit* (ECU)

*Engine Control Unit (ECU)* adalah bagian penting dari sistem kontrol mesin kendaraan yang dibentuk oleh mikroprosesor untuk perangkat kerasnya yang terdiri dari beberapa sensor dan kontrol perangkat lunak untuk mengatur sensor- sensor pada kendaraan. Fungsi ECU adalah menerima data dan sinyal status mesin kemudian menggerakkan sirkuit aktuator untuk mencapai tujuan kontrolnya [9]. Agar bisa berkomunikasi dengan OBD-II diperlukan sebuah protokol khusus yang kompatibel sehingga proses komunikasi antar ECU dan OBD-II bisa jalan contoh protokolnya seperti CAN, SAE J1850 PWM, SAE J1850 VWM, ISO 9141-2, ISO

14230-4 (CAN) dan protokol-protokol yang lainnya [10].

### 2.4 Embedded System

*Embedded System* adalah sebuah sistem yang bertujuan untuk melakukan suatu pekerjaan spesifik dan biasanya sistem tertanam ini tidak dapat berdiri sendiri. *Embedded System* mencakup *hard ware* sebagian besarnya lalu dikombinasikan dengan perangkat lunak untuk menunjang fungsi digitalnya perangkat keras yang digunakan berupa mikrokontroler karena dalam mikrokontroler mempunyai prosesor dan memori yang dapat diprogram sesuai kebutuhan *Embedded System* seperti berikut [14]. Adapun beberapa jenis *Embedded System* seperti berikut [15].

Mandiri (*standalone*) adalah sebuah sistem yang berfungsi secara independen dari perangkat keras lainnya dan tidak memerlukan *software* selain sistem operasi untuk menjalankannya.

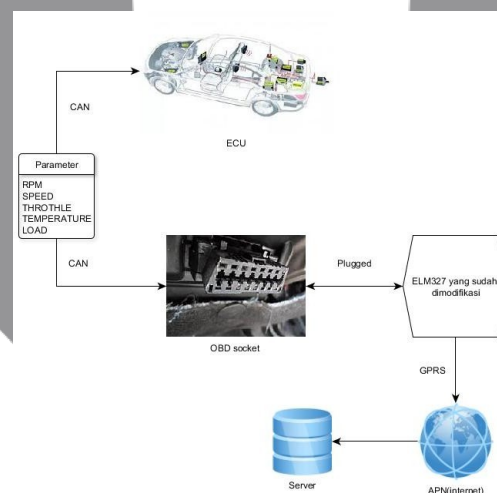
*Real Time* adalah sebuah sistem yang dilakukan dalam periode waktu spesifik dengan tugas - tugas yang spesifik.

*Hard Real Time* mirip seperti sistem *real time* namun memiliki sebuah *dead- line* sering kali digunakan sebagai alat pengontrol untuk aplikasi yang dedicated, mempunyai batasan waktu yang tetap yang sudah didefinisikan dengan baik. Pemrosesan harus selesai dalam batasan-batasan yang sudah didefinisikan, atau sistem akan gagal.

*Soft Real Time* adalah sistem yang tidak memerlukan deadline. Delay ini tidak akan berakibat sesuatu yang serius. Sistem *soft real-time* mempunyai lebih sedikit batasan waktu yang keras, dan tidak mendukung deadline dengan menggunakan batas akhir.

### 3. Perancangan Sistem

Pada sistem *data collection* ini menggunakan device ELM327 yang sudah dimodifikasi dengan menambah Arduino Nano dan SIM 800L sehingga menjadi sebuah *embedded system*. Terdapat tiga segmen dimulai dari rekam data ECU sampai pengiriman data secara nirkabel ke internet dengan menggunakan GPRS. Gambar 2 merupakan Diagram alur sistem



Gambar 2 : Diagram alur sistem.

Dimulai dari ELM327 membaca parameter sesuai protokol pada ECU yang meliputi *Revolutions Per Minute (RPM)*, *Speed*, *Throthle*, *Temperatur*, *Load* lalu data ditransfer ke Arduini Uno melalui altsoftserial untuk di *framing* agar bisa diterjemahkan setelah itu melalui softwareserial data akan dikirim oleh SIM 800L ke jaringan internet menggunakan GPRS.

#### 4. Hasil dan Analisis

Pada pengujian device dilakukan dalam 3 skenario dan kondisi yang berbeda yaitu saat berada di jalan tol, jalan kota, dan kondisi idle. Dilakukan dalam skenario berbeda untuk melihat perilaku hasil data parameter apakah sesuai dengan kondisi yang dilakukan atau tidak karena untuk melihat presisi sistem harus dilakukan di beberapa medan. Terdapat 5 parameter yang ditinjau yaitu Load, Temperature, RPM, Speed, dan Throttle.

##### A. Kondisi Idle

Pada pengujian Ok dan Error value untuk melihat data yang berhasil terupdate atau terkirim ke server dari ketiga skenario yang telah dilakukan apakah sistem ini layak atau tidak dilihat dari nilai ok yang didapat.

Tabel 1 : Ok dan Error Value Saat Idle

Idle					
Parameter	All Value (Ok only)				
	average	SD	Max	Min	Std Error
Load	35.83333	3.174	40	27	0.579503
Temperature	56.2	8.318654	69	42	1.518771
RPM	901.8	49.64731	1085	837	9.064317
Speed	0	0	0	0	0
Throttle	25.4	0.894427	27	23	0.163299

Tabel 1 dari hasil yang didapatkan nilai standar deviasi tidak relatif beda jauh yang berarti sistem ini layak karena Ok value dan All value relatif tidak berbeda jauh.

##### B. Kondisi Highway

Setelah melakukan pengujian pada beberapa skenario didapatkan hasil rata-rata, standar deviasi, Max, Min, Std Error

Tabel 2 : Ok dan Error Value Saat Highway

Highway										
Parameter	All Value					Ok Value				
	average	SD	Max	Min	Std Error	average	SD	Max	Min	Std Error
Load	50.06	24.4619	100	9	3.459434	51.8	24.35377	100	9	4.116538
Temperature	82.76	1.0214	86	80	0.144448	82.57143	1.008368	85	80	0.170445
RPM	2958.12	341.6016	3625	1884	51.27961	2913.057	376.5748	3526	1884	63.65337
Speed	91.1	13.397	113	47	1.894622	89.25714	14.19889	111	47	2.40005
Throttle	43.76	19.0986	83	27	2.700954	45.05714	19.89375	83	27	3.362658

dari hasil yang didapatkan nilai standar deviasi tidak relatif beda jauh yang berarti sistem ini layak karena Ok value dan All value relatif tidak berbeda jauh.

##### C. Kondisi Downtown

Setelah melakukan pengujian pada beberapa scenario didapatkan hasil rata – rata, standar devisiasi, Max, Min, Std Error.

Tabel 3 : Ok dan Error Value Saat Downtown

Parameter	Downtown									
	All Value					Ok Value				
	average	SD	Max	Min	Std Error	average	SD	Max	Min	Std Error
Load	46.52	27.1001	100	16	3.83254	49.37208	28.08662	100	21	4.283169
Temperature	84.62	0.9875	87	83	0.13965	84.62791	1.047065	87	83	0.159676
RPM	1475.18	441.8586	2659	574	62.48824	1470.651	455.4807	2659	574	69.46015
Speed	27.58	9.4028	42	7	1.329751	27.32558	8.809413	43	10	1.343423
Throttle	32.64	10.2275	67	23	1.446392	33.30233	10.74939	67	23	1.639266

Tabel 3 dari hasil yang didapatkan nilai standar deviasi tidak relatif beda jauh yang bearti sistem ini layak karena Ok value dan All value relatif tidak berbeda jauh.

## 5. Kesimpulan

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

1. Dari hasil pengujian dalam beberapa kondisi, itu menunjukkan bahwa sistem ini cukup presisi karena nilai error rendah.
2. Nilai parameter dari 3 kondisi cukup masuk akal tidak terlalu menyimpang jauh.
3. Sistem ini baik karena perbandingan setiap nilai error standar deviasi dan nilai OK tidak jauh berbeda.

## 6. Daftar Pustaka

- [1]B. Dorsemaine, J.-P. Gaulier, J.-P. Wary, N. Kheir, and P. Urien, "Internet of things: a definition & taxonomy," in *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*. IEEE, 2015, pp. 72–77.
- [2]A. IDE, "Introduction arduino," <https://www.arduino.cc/en/Guide/Introduction>, accessed: 2019-05-23.
- [3]Kapolri, "Peraturan kapolri," <https://www.peraturankapolri.com/2016/01/peraturan-kapolri-nomor-15-tahun-2013>, accessed: 2019-05-23.
- [4]S. A. Nugroho, E. Ariyanto, and A. Rakhmatsyah, "Utilization of onboard diagnostic 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, pp. 7–11.
- [5]K. H. Johansson, M. T. Örngrén, and L. Nielsen, "Vehicle applications of controller area network," in *Handbook of networked and embedded control systems*. Springer, 2005, pp. 741–765.
- [6]T. O. I. Homepage, "Obd-ii background," <http://www.obdi.com/background.html>, accessed: 2019-05-23.
- [7]Y. Xie, X. Su, Y. He, X. Chen, G. Cai, B. Xu, and W. Ye, "Stm32-based vehicle data acquisition system for internet-of-vehicles," in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*. IEEE, 2017, pp. 895–898.
- [8]N. N. Hasan, A. Arif, U. Pervez, M. Hassam, and S. S. U. Husnain, "Micro-controller based on-board diagnostic (obd) system for non-obd vehicles," in *2011 Uksim 13th International Conference on Computer Modelling and Simulation*. IEEE, 2011, pp. 540–544.
- [9]ELM, "Elm 327 obd to rs232," <https://www.elmelectronics.com/wp-content/uploads/2016/07/ELM327DS.pdf>, accessed: 2019-05-23.
- [10]M. K. Ajudia, M. T. Kolte, and P. Sarkar, "Designs of input and output driver circuits for 16-bit electronic control unit (ecu) and development of control strategy using these i," *O Driver Circuits*, pp. 102–106, 2014.

- [11]F. Kong, L. Zhang, J. Zeng, and Y. Zhang, "Automatic measurement and control system for vehicle ecu based on can bus," in *2007 IEEE International Conference on Automation and Logistics*. IEEE, 2007, pp. 964–968.
- [12]H. Zeltwanger, "Can standard review: Changes and enhancements of the iso 11898," SAE Technical Paper, Tech. Rep., 2000.
- [13]A. Cheng, Y. Yao, Z. Duan, A. Zhou, and W. Hong, "Ecu loader design of in-vehicle can network based on iso15765," in *International Conference on Information Science and Technology*. IEEE, 2011, pp. 1215–1217.
- [14]R. Kamal, *Embedded systems: architecture, programming and design*. Tata McGraw-Hill Education, 2011.
- [15]F. Santosa, "Embedded system," <https://febriadisantosa.weebly.com/knowledge/embedded-system>, accessed: 2019-05-23.
- [16]Y. A. Badamasi, "The working principle of an arduino," in *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*. IEEE, 2014, pp. 1–4.
- [17]Nettigo, "Sim 8001," <https://nettigo.eu/products/sim8001-gsm-gprs-module>, accessed: 2019-05-23.

