

PERANCANGAN DAN IMPLEMENTASI PERANGKAT PEMINDAI PORTABEL ON-BOARD DIAGNOSTIC-II PADA KENDARAAN RODA EMPAT BERBASIS MIKROKONTROLER

DESIGN AND IMPLEMENTATION OF ON-BOARD DIAGNOSTIC-II PORTABLE SCANNER DEVICE ON FOUR WHEEL VEHICLE BASED ON MICROCONTROLLER

Muhammad Fajar Saputro¹, Rizki Ardianto Priramadhi², Denny Darlis³

^{1,2}Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

³Prodi D3 Teknik Telekomunikasi, Fakultas Industri Terapan, Universitas Telkom

¹mfajaar@student.telkomuniversity.ac.id, ²rizki.ap@telkomuniversity.ac.id,

³dennydarlis@tass.telkomuniversity.ac.id

Abstrak

Mobil saat ini sudah menjadi transportasi utama bagi semua kalangan, baik itu mobil pribadi maupun mobil umum. Selain praktis, mobil dapat menjangkau tujuan yang diinginkan dengan lebih efektif jika dibandingkan dengan kereta api. Setiap mobil sudah pasti dilengkapi dengan ECU dan *On-Board Diagnostic* sebagai komponen penting pada mobil itu sendiri. Namun jika kita berbicara tentang ECU dan *On-Board Diagnostic*, mobil pabrikan sudah membuat *dashboard* untuk menampilkan parameter sensor yang di dapat dari ECU. Tampilan hasil sensor-sensor tersebut diletakkan di depan kemudi mobil agar dapat dilihat secara langsung oleh pengemudi dengan mudah.

Perangkat pembaca data OBD-II pada kendaraan roda empat ini dibuat agar dapat membaca parameter sensor-sensor yang ada pada mobil untuk ditampilkan pada *OBD Portable Scanner*. Parameter yang didapat yaitu berupa *vehicle speed, engine speed, oil temperature, fuel rate, engine cooled temperature, dan accu voltage*. Pada mobil pabrikan, parameter tersebut hanya ditampilkan di *dashboard* mobil itu sendiri. Namun dengan menggunakan *OBD Portable Scanner*, parameter-parameter tersebut dapat ditampilkan secara bergantian pada *OBD Portable Scanner*.

Dengan adanya *OBD Scanner Portable*, pengguna dapat dengan mudah membaca kondisi mobil yang mengalami kendala, sehingga dapat membaca parameter-parameter yang didapat dari ECU.

Kata Kunci: OBD-II, ECU.

Abstract

The car has now become the main transportation for all people, both private cars and public cars. Besides being practical, cars can reach their desired destinations more effectively when compared to trains. Every car is definitely equipped with ECU and On-Board Diagnostic as an important component in the car itself. But if we talk about the ECU and On-Board Diagnostic, the car manufacturer has made a dashboard to display the sensor parameters obtained from the ECU. Display the results of the sensors are placed in front of the steering wheel of the car so that it can be seen directly by the driver easily.

OBD-II data reader devices on four-wheeled vehicles are made in order to be able to read the parameters of sensors in the car to be displayed on the OBD Scanner Portable. The parameters obtained are in the form of vehicle speed, engine speed, oil temperature, fuel rate, engine cooled temperature, and battery voltage. On a car manufacturer, these parameters are only displayed on the dashboard of the car itself. But by using the OBD Scanner Portable, these parameters can be displayed alternately on the OBD Scanner Portable.

With the OBD Scanner Portable, users can easily read the condition of the car that is experiencing problems, so they can read the parameters obtained from the ECU.

Keywords: OBD-II, ECU.

1. Pendahuluan

Seperti yang kita ketahui, transportasi merupakan salah satu unsur penting pada zaman sekarang. Setiap orang membutuhkan transportasi untuk melakukan aktivitas sehari-hari. Salah satu transportasi yang umum digunakan yaitu mobil. Mobil dapat dikendalikan sendiri oleh pemiliknya sehingga dapat menuju tempat yang diinginkan tanpa ada batasan seperti transportasi umum. Tidak heran jika mobil menjadi transportasi utama untuk beraktivitas^[1].

Namun di sisi lain, tidak semua orang mampu untuk membeli mobil dikarenakan harganya yang tidak murah. Selain itu perawatan mobil membutuhkan biaya perawatan yang tidak murah. Perawatan mesin, pergantian oli secara berkala, dan juga pengisian bahan bakar harus dilakukan secara berkala agar memperpanjang usia pakai kendaraan^[2].

Mobil memiliki beberapa komponen penting didalamnya. Salah satu komponen penting nya yaitu *Electronic Control Unit* (ECU). Komponen ini berfungsi sebagai sirkuit elektronik utama pada sebuah mobil. Secara garis besar, cara kerja *Electronic Control Unit* (ECU) hampir sama dengan cara kerja *Central Processing Unit* (CPU) pada komputer. Komponen ini berfungsi sebagai pengendali dari sistem kelistrikan pada kendaraan. ECU bekerja apabila diberi masukan, kemudian melakukan pengambilan keputusan apakah sistem masukan tersebut berjalan dengan normal atau terdapat kesalahan didalamnya, dan mengeluarkan perintah baru yang sesuai. Keluaran dari ECU akan mengubah sistem operasi pada sistem sesuai dengan efek yang diinginkan^[3].

Parameter yang dapat diberikan oleh ECU bisa didapat dengan melakukan komunikasi antara perangkat *OBD Portable Scanner* dengan ECU melalui *OBD-II*. Data-data yang didapat dari *Electronic Control Unit* (ECU) ini akan di kirimkan ke *On-Board Diagnostic II* (OBD) kemudian diolah oleh mikrokontroler. Data yang didapat dari *On-Board Diagnostic II* (OBD-II) akan di tampilkan di LCD, kemudian pengguna dapat memilih parameter apa yang ingin ditampilkan pada LCD. Perangkat ini dapat menampilkan hingga 3 parameter dalam 1 waktu, dan juga parameter ini dapat bekerja apabila terkoneksi langsung dengan *On-Board Diagnostic II* (OBD) pada kendaraan.

Dengan menggunakan *OBD Scanner Portable*, pengguna dapat mengoperasikannya hanya dengan menghubungkan *OBD-II* dengan perangkat yang dibuat. Sehingga pengguna dapat dengan mudah mengetahui parameter-parameter sensor yang ada pada kendaraan roda empat yang ditampilkan di LCD pada perangkat tersebut.

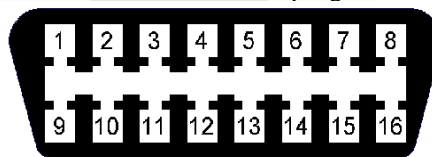
Melihat permasalahan diatas, penulis membuat perangkat ini untuk di terapkan dan di kembangkan pada teknologi mobil sekarang. Dengan menggunakan *OBD Scanner Portable* diharapkan dapat memberikan opsi tambahan kepada pemilik kendaraan untuk melihat parameter sensor-sensor yang ada pada mobil secara *real time*.

2. Tinjauan Pustaka

2.1. On-Board Diagnostic-II (OBD-II)

On-Board Diagnostic (OBD) adalah sistem pada kendaraan roda empat berbasis komputer yang dirancang untuk dapat memantau kondisi dan kinerja komponen-komponen pada mesin kendaraan dan mendeteksi malfungsi. Dapat diartikan bahwa *On-Board Diagnostic* adalah bahasa pada *Engine Control Unit*. Sistem OBD-II memberikan akses kepada manusia untuk mendapatkan informasi kesehatan kendaraan dan akses ke parameter-parameter serta sensor dari *Engine Control Unit* (ECU).

Konektor *On-Board Diagnostic* biasanya berada di posisi bawah *dashboard* bagian penumpang atau pengemudi, dengan tujuan agar mudah di akses dari kursi pengemudi.



Gambar 2.1. OBD-II Port.

(Sumber: <http://www.obdii.com/connector.html>)

Fungsi tiap pin pada port OBD-II yaitu sebagai berikut:

- Pin 2 - J1850 Bus+
- Pin 4 - *Chassis Ground*
- Pin 5 - *Signal Ground*
- Pin 6 - *CAN High*
- Pin 7 - *ISO 9141-2 K Line*
- Pin 10 - J1850 Bus
- Pin 14 - *CAN Low*
- Pin 15 - *ISO 9141-2 L Line*
- Pin 16 - *Battery Power*

2.2. MCP2515 CAN Bus

MCP2515 adalah *Controller Area Network* (CAN) *controller* yang dikembangkan untuk

menyederhanakan aplikasi yang membutuhkan interaksi dengan CAN Bus. Perangkat ini terdiri dari tiga blok utama, yaitu:

1. *CAN module.*
2. *Control logic.*
3. *Serial Peripheral Interface (SPI) protocol block.*

2.3. Protokol Komunikasi

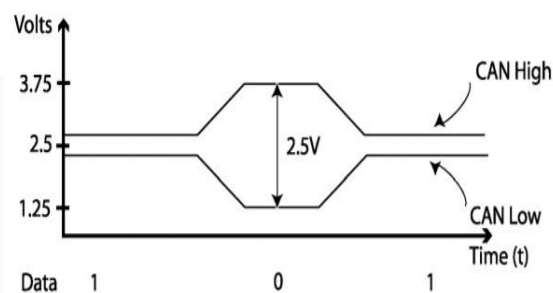
Terdapat beberapa protokol yang dapat diimplementasikan pada kendaraan roda empat agar dapat berkomunikasi dengan ECU[4][2]:

- SAE J1850 PWM – pin 2, 4, 5, 10 dan 16
- SAE J1850 VPW – pin 2, 4, 5 dan 16
- ISO 9141 – 4, 5, 7, 15 dan 16
- ISO 14230 KWP2000 – 4, 5, 7, 15 dan 16
- ISO 15765 (CAN) – pin 4, 5, 6, 14 dan 16.

Dari lima protokol diatas memiliki perbedaan yang signifikan yaitu pada pemempatan susunan pin untuk berkomunikasi. Satu tipe kendaraan hanya mengimplementasikan satu protokol.

2.4. Controller Area Network (CAN)

Controller Area Network (CAN) adalah suatu protokol komunikasi yang terdiri dari beberapa simpul yang terhubung melalui satu bus sebagai jalur untuk berkomunikasi. CAN dapat mengatur dan mengendalikan semua data yang masuk dan data yang keluar[10]. Untuk dapat berkomunikasi dengan *CAN Bus* yaitu melalui dua kabel yang disebut *CAN High* dan *CAN Low*. Pada kondisi *CAN Bus* tidak mengirim dan menerima data maka kedua kabel membawa 2,5V. Pada saat mentransmisikan data, *CAN High* menuju ke 3,75V dan *CAN Low* turun ke 1,25V, sehingga menghasilkan perbedaan 2,5V[11].



Gambar 2.3. CAN High dan CAN Low.

2.5. OBD-II PIDs

On-Board Diagnostic-II Parameter IDs (OBD-II PIDs) adalah kode yang akan digunakan untuk meminta data parameter dari kendaraan yang akan digunakan sebagai alat diagnostik. Kode-kode ini diterapkan di semua mobil yang beredar di pasaran Amerika Utara sejak tahun 1996. Kode-kode ini adalah standar SAE J / 1979. Pada Tabel 2.1 akan dijelaskan *Parameter IDs (PIDs)* yang akan digunakan pada Tugas Akhir ini.

Tabel 2.1. *Parameter IDs*[16].

PID (hex)	Data bytes returned	Description	Min value	Max value	Units	Formula
05	1	Engine coolant temperature	-40	215	°C	A-40
0C	2	Engine speed	0	16,383.75	rpm	((A*256)+B
0D	1	Vehicle speed	0	255	kmh	A
11	1	Throttle position	0	100	%	A*100/255

Pada CAN Bus kendaraan terdapat *Query* dan *Response*. Pada OBD *standard* menggunakan alamat fungsional. Pembaca diagnostik akan memulai sebuah permintaan menggunakan *CAN-ID 7DFh*, yang bertindak sebagai alamat *broadcast*, dan akan menerima *response* dari *ID* apa pun dalam rentang 7E8h hingga 7Efh. ECU yang dapat menanggapi OBD *query* mendengarkan fungsional *broadcast ID* dari 7DFh dan satu *ID* yang ditugaskan dalam kisaran 7E0h hingga 7E7h.

1. *Query*

Query PID fungsional akan dikirimkan ke kendaraan pada *CAN BUS* dengan *ID 7DFh* dan menggunakan 8 byte data. *Byte* tersebut yaitu:

Tabel 2.2. *Query* PID[19].

Byte -gt;	_ 0 _	_ 1 _	_ 2 _	_ 3 _	_ 4 _	_ 5 _	_ 6 _	_ 7 _
	Number of additional data bytes: 2	Mode01 = show current data; 02 = freeze frame; etc.						
SAE Standard			PID code(e.g.: 05 = Engine coolant temperature)	not used(may be 55h)				
	Number of additional data bytes: 3	Custom mode: (e.g.: 22 = enhanced data)	PID code(e.g.: 4980h)	not used(may be 00h or 55h)				
Vehicle specific								

2. Response

Query PID akan ditanggapi pada CAN BUS dengan message IDs yang bergantung pada modul mana yang akan merespon. Pada keadaan normal mesin atau ECU utama akan merespon pada ID 7E8h. Pada modul lainnya seperti pengontrol baterai pada Prius akan merespon pada 07E9h, 07EAh, 07Ebh, dll. Pesan tersebut menggunakan 8 byte data. Byte tersebut yaitu:

Tabel 2.3. Response PID[19].

Byte -gt;	_ 0 _	_ 1 _	_ 2 _	_ 3 _	_ 4 _	_ 5 _	_ 6 _	_ 7 _
SAE Standard 7E8h, 7E9h, 7EAh, etc.	Number of additional data bytes: 3 to 6	Custom modeSame as query, except that 40h is added to the mode value. So: 41h = show current data; 42h = freeze frame; etc.	PID code(e.g.: 05 = Engine coolant temperature)	value of the specified parameter, byte 0	value, byte 1 (optional)	value, byte 2 (optional)	value, byte 3 (optional)	not used(may be 00h or 55h)
Vehicle specific 7E8h, or 8h + physical ID of module.	Number of additional data bytes: 4 to 7	Custom mode: same as query, except that 40h is added to the mode value. (e.g.: 62h = response to mode 22h request)	PID code(e.g.: 4980h)		value of the specified parameter, byte 0	value, byte 1 (optional)	value, byte 2 (optional)	value, byte 3 (optional)
Vehicle specific 7E8h, or 8h + physical ID of module.	Number of additional data bytes: 3	7Fh this a general response usually indicating the module doesn't recognize the request.	Custom mode: (e.g.: 22h = enhanced diagnostic data by PID, 21h = enhanced data by offset)	31h	not used(may be 00h)			

Berikut adalah contoh perhitungan Query dan Response PID:

Engine Speed (rpm)=

Query

0x02, 0x01, 0x0C, 0x55, 0x55, 0x55, 0x55, 0x55

Response

0x04, 0x41, 0x0C, 0x0B, 0x09, 0x00, 0x00, 0x00

Dapat kita lihat dari nilai diatas, (41 dan 0C) merupakan Response dari Query (01 dan 0C), dan dua bytes setelahnya merupakan nilai dari Engine Speed yaitu (0B dan 09). Maka untuk mendapatkan nilai RPM yang sesungguhnya dengan cara mengubah terlebih dahulu nilai hexadecimal ke decimal dan kemudian hitung hasil konversi tersebut dengan menggunakan rumus $Engine\ Speed = ((A * 256) + B) / 4$.

0Bh = 11d (A)

09h = 09d (B)

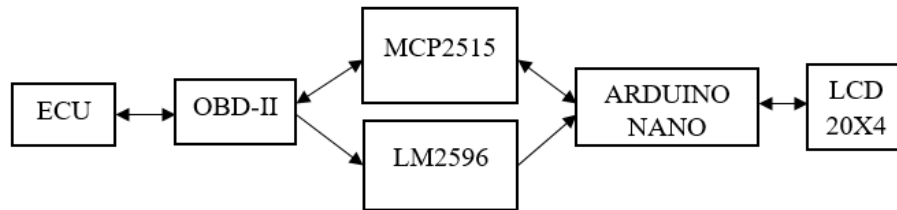
Engine Speed = ((11 * 256) + 09) / 4 = 706,25 rpm.

3. Perancangan Sistem

3.1. Diagram Blok Sistem

Sebelum membangun sebuah sistem, desain sistem sangat dibutuhkan agar perangkat dapat bekerja sesuai dengan desain yang telah dirancang. Pada tugas akhir ini, penulis akan membangun sistem pembaca data OBD-II pada mobil menggunakan mikrokontroler. Sistem yang akan dibangun ini mempunyai fungsi membaca, mengambil, mengolah, dan mengirimkan data-data yang terdapat pada ECU kemudian menampilkannya ke LCD

sehingga dapat dilihat oleh pengguna.



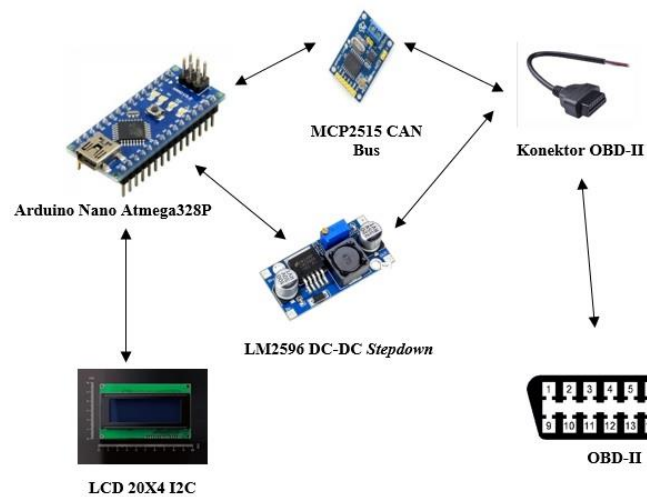
Gambar 3.1. Diagram Blok Perangkat Keras.

3.2. Desain Perangkat Keras

Adapun perangkat keras yang akan dibuat pada tugas akhir ini terdiri dari beberapa komponen utama yaitu:

- Arduino Nano (Atmega328P)
- 20X4 LCD *Display Module*
- MCP2515 CAN BUS
- LM2596 DC-DC *Step Down Converter Module*

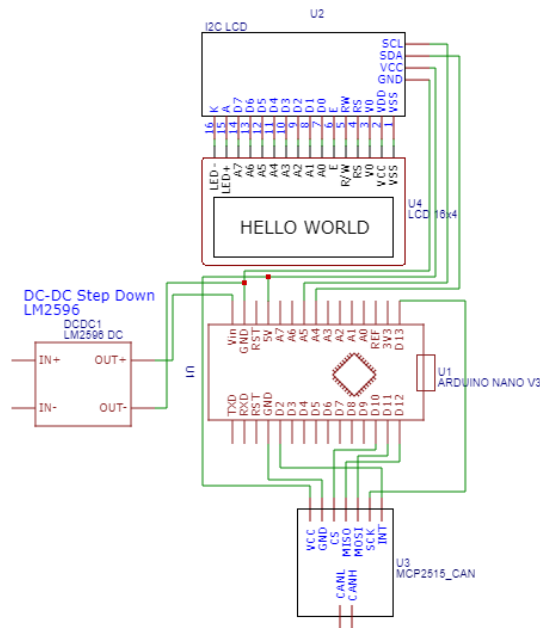
Kabel Konektor OBD-II.



Gambar 3.2. Desain Perangkat Keras.

3.3. Schematic Rangkaian

Pada *schematic* tersebut terdiri dari beberapa komponen yaitu Arduino Nano ATmega 328P, 20X4 LCD *Display Module*, MCP2515 CAN *Module*, *Inter Integrated Circuit (I2C) Module*, LM2596 DC-DC *Step Down Converter*.



Gambar 3.3. Schematic Rangkaian.

4. Pengujian dan Analisa

4.1. Pengujian Engine Speed

Pengujian ini dilakukan agar dapat mengetahui perbandingan nilai keluaran *Engine Speed* (rpm) pada *OBD Scanner Portable* dengan nilai keluaran *Engine Speed* (rpm) pada *Dashboard* mobil.

4.2. Pengujian Vehicle Speed

Pengujian ini dilakukan agar dapat mengetahui perbandingan nilai keluaran *Vehicle Speed* (kmh) pada *OBD Scanner Portable* dengan nilai keluaran *Vehicle Speed* (kmh) pada *Dashboard* mobil. Pengujian ini dilakukan dengan

4.3. Pengujian Throttle Position

Pengujian ini dilakukan agar dapat mengetahui perbandingan nilai keluaran *Throttle Position* (%) pada *OBD Scanner Portable* dengan posisi pedal gas yang ditekan.

4.4. Pengujian Engine Coolant Temperature

Pengujian ini dilakukan agar dapat mengetahui perbandingan nilai keluaran *Engine Coolant Temperature* (°C) pada *OBD Scanner Portable* dengan nilai keluaran *Engine Coolant Temperature* (°C) pada *Thermometer*.

4.5. Menampilkan Data ECU yang Terdeteksi di LCD

Pengujian ini dilakukan agar dapat mengetahui data ECU berupa parameter sensor yang dapat ditampilkan di *Liquid Crystal Display* (LCD). Parameter sensor yang akan ditampilkan di *Liquid Crystal Display* (LCD) yaitu *Engine Speed* (rpm), *Vehicle Speed* (kmh), *Throttle Position* (%), dan *Engine Coolant Temperature* (°C).



Gambar 4.1. Data yang Ditampilkan di LCD.

4.6. Verifikasi CAN 11bit 500kb Sebagai Protokol yang Digunakan

Pengujian ini dilakukan agar dapat memastikan bahwa data parameter sensor-sensor pada ECU sesuai dengan protokol komunikasi yang telah ditentukan pada penelitian Tugas Akhir ini.

OBD Scanner Portable ini melakukan pengambilan data parameter sensor-sensor pada protokol komunikasi ISO 15765 CAN 11bit 500kb.

```
//init CAN
if(Canbus.init(CANSPEED_500)) //init can
{
  Serial.println("ECU Access Success!");
  lcd.setCursor(0, 0);
  lcd.print("ECU Access Success!");
  delay(2000);
  lcd.clear();
}
```

Gambar 4.2. Inisialisasi CAN.

Dapat dilihat pada Gambar 4.8 bahwa inisialisasi pada Arduino IDE dilakukan pada protokol komunikasi ISO 15765 CAN 11bit 500kb dengan *specified speed* yaitu 500kb.

```
#define CANSPEED_500 1 // CAN speed at 500 kbps

#define COOLANT 0x05
#define ROTATION 0x0C
#define SPEED 0x0D
#define THROTTLE 0x11
#define MAF_SENSOR 0x10

#define PID_REQUEST 0x7DF
#define PID_REPLY 0x7E8
```

Gambar 4.3. CAN BUS *library*.

Mengacu pada CAN BUS *library* pada Gambar 4.9, ISO 15765 CAN 11bit 500kb didefinisikan dengan CANSPEED_500 yang menunjukkan bahwa *speed* pada 500kb. *Query* PIDs akan dikirimkan ke kendaraan pada CAN BUS dengan *ID 7DFh* dan mendefinisikan PID_REQUEST dengan nilai 0x7DF. Sedangkan untuk *Response* pada CAN BUS yaitu dengan *ID 7E8h* dan mendefinisikan PID_REPLY dengan nilai 0x7E8.

5. Kesimpulan

Berdasarkan hasil pengujian yang dilakukan oleh penulis, maka dapat disimpulkan:

1. *OBD-II* bekerja sebagai *port* yang digunakan untuk mengambil data parameter sensor pada ECU .
2. Data yang didapat dari *OBD-II* dapat diolah oleh mikrokontroler agar dapat ditampilkan pada *OBD Scanner Portable*.
3. *OBD Scanner Portable* yang dihubungkan dengan MCP2515 CAN BUS dapat membaca parameter sensor-sensor pada mobil.
4. *OBD Scanner Portable* dapat menampilkan parameter sensor-sensor pada mobil di LCD 20X4 dalam satu waktu.

OBD Scanner Portable dapat membaca parameter sensor-sensor pada semua mobil yang menggunakan protokol ISO 15765 - 4 CAN 11bit 500kb (dengan menggunakan pin 4, 5, 6, 14, dan 16 pada konektor OBD).

6. Saran

Berdasarkan hasil pengujian yang dilakukan oleh penulis, saran untuk penulis selanjutnya yang ingin melanjutkan penelitian ini adalah sebagai berikut:

1. Alat dapat dikembangkan dengan memanfaatkan Android agar parameter sensor-sensor dapat dilihat di Android.
2. Alat dapat membaca lebih banyak parameter sensor-sensor pada mobil.
3. Alat dapat membaca parameter sensor-sensor pada protokol lain.
4. Pengujian dilakukan pada pagi/malam hari agar mengurangi panas matahari yang bisa berdampak pada suhu mesin.

Reference:

- [1] Imanuel Christonny and Eka Firmansyah and I Wayan Mustika. 2015. Desain Sistem Elektronis Komunikasi Kendaraan Listrik Dengan Controller Area Network (CAN).
- [2] Diagnostics, Total Car. OBD2 PIDs for Programmers (Technical). (<http://www.totalcardiagnostics.com/support/Knowledgebase/Article/View/104/0/obd2-pids-for-programmers-technical>).
- [3] Resource, OBD-II. OBD-II PIDs. (<http://obdcon.sourceforge.net/2010/06/obd-ii-pids>).