

# **ECU LOGGER : PERANCANGAN SISTEM PENYIMPANAN DAN MONITORING DATA ELEKTRONIK MOBIL**

## **ECU LOGGER : DESIGN SYSTEM STORAGE AND MONITORING OF CAR ELECTRONIC DATA**

**Doni Yosef Daniel<sup>1</sup>, Rizki Ardianto Priramadhi<sup>2</sup>, Denny Darlis<sup>3</sup>**

<sup>1,2</sup>Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

<sup>3</sup>Prodi D3 Teknik Telekomunikasi, Fakultas Industri Terapan, Universitas Telkom

<sup>1</sup>[doniyosefdaniel@student.telkomuniversity.ac.id](mailto:doniyosefdaniel@student.telkomuniversity.ac.id), <sup>2</sup>[rizki.ap@telkomuniversity.ac.id](mailto:rizki.ap@telkomuniversity.ac.id),

<sup>3</sup>[dennydarlis@telkomuniversity.ac.id](mailto:dennydarlis@telkomuniversity.ac.id)

---

### **Abstrak**

Di era modern saat ini mobil sudah di dominasi alat-alat elektronik. Sistem elektronik pada otomotif biasanya menggunakan ECU (*Engine Control Unit*) yang merupakan sirkuit elektronik utama pada sebuah mobil. ECU dapat mendeteksi kerusakan sistem kelistrikan di mobil melalui sensor-sensor di mesin pada mobil. Mengetahui performa sistem elektronik pada mobil sangat diperlukan supaya pengguna mobil dapat melakukan perbaikan pada mobilnya.

Penelitian ini bertujuan untuk membuat sistem atau perangkat yang dapat memonitor performa sistem elektronik pada mesin mobil secara *real time* melalui program tampilan serial menggunakan mikrokontroler ATmega328 dan menyimpan data tersebut ke piranti memori *SD Card*.

*ECU Logger* dapat melakukan monitoring performa sistem elektronik pada mobil dan menyimpan data tersebut ke *SD Card* sehingga pengguna kendaraan mobil dapat menampilkan data yang di hasilkan ECU pada perangkat *PC* atau laptop. Beberapa parameter sensor yang di monitoring oleh *ECU Logger* adalah *Vehicle Speed* (0 – 255 km/h), *Throttle Position* (0 – 100%), *Engine Speed* (0-16.383,7 RPM), *Engine Coolant Temperature* (-40°C s/d 215 °C), menggunakan 97% kapasitas memori internal ATmega328. Sistem juga dapat menampilkan waktu pada saat itu juga seperti detik, menit, jam, hari, bulan, dan tahun.

**Kata Kunci : ECU, ECU Logger, monitoring, CAN bus, ATmega 328.**

---

### **Abstract**

In the modern era, cars are now dominated by electronic devices. The electronic system is located on the ECU (*Engine Control Unit*) which is the main electronic circuit in a car. The ECU can detect damage to the electrical system in the car through sensors in the engine on the car. Knowing the performance of the electronic system in a car is very necessary so that drivers can make repairs to the car.

This research create systems or devices that can monitor the performance of electronic systems on car engines in real time through serial display programs using the ATmega328 microcontroller and storing the data to *SD Card* memory devices.

The *ECU Logger* can monitor the performance of the electronic system on the car and store the data to *SD Card* so that drivers can display the data generated by the ECU on *PC* or laptop devices. Some sensor parameters monitored by the *ECU Logger* are *Vehicle Speed* (0-255 km/h), *Throttle Position* (0-100%), *Engine Speed* (0-16,383.7 RPM), *Engine Coolant Temperature* (-40°C to 215 °C), uses 97% ATmega328 internal memory capacity. The system can also display the time at that time as well as seconds, minutes, hours, days, months and years.

**Keywords: ECU, ECU Logger, monitoring, CAN bus, ATmega 328.**

---

## 1. Pendahuluan

Mobil zaman sekarang sudah di dominasi alat-alat elektronik. Kerusakan-kerusakan sistem elektronik menjadi salah satu penyebab terjadinya kecelakaan. Mengacu pada masalah diatas, perlu adanya suatu perangkat yang dapat merekam dan menganalisa suatu penyebab kecelakaan seperti kerusakan sistem elektronik kendaraan. *ECU Logger* merupakan perangkat yang tepat untuk menganalisa kerusakan sistem elektronik pada kendaraan khususnya mobil. Piranti ini merupakan alat yang dapat mendeteksi sekaligus menyimpan data performa sistem elektronik pada mobil.

*ECU Logger* ini dapat mendeteksi performa sistem elektronik pada mobil yang sudah dilengkapi dengan suatu EFI (*electronic fuel injection*), dengan cara mendapatkan data dari ECU yang merupakan singkatan dari *Engine Control Unit* yang berfungsi sebagai sirkuit elektronik utama pada sebuah mobil. *ECU Logger* dapat mendeteksi kerusakan sistem komponen misalnya sistem elektrikal, transmisi, *ABS*, *airbag*, *immobilizer*, *AC*, sampai sistem injeksi dan sensor-sensor di mesin pada mobil.

Data yang dihasilkan dari ECU disimpan di ECU Logger dan nantinya dapat ditampilkan pada perangkat PC atau laptop. Penerapan sistem ini diharapkan dapat membantu pengguna mobil untuk mengetahui keadaan mesin atau sistem elektronik pada mobil agar dapat mencegah hal-hal yang tidak diinginkan saat berkendara misalkan terjadinya kecelakaan yang diakibatkan oleh kerusakan mesin atau sistem elektronik pada mobil.

## 2. Tinjauan Pustaka

### 2.1. Engine Control Unit (ECU)

*ECU* merupakan singkatan dari *Engine Control Unit*, sesuai namanya komponen ini pasti berfungsi sebagai sirkuit elektronik utama pada sebuah mobil. *ECU* sebagai sirkuit utama memiliki peran sebagai *central controller* atau jika pada komputer maka *ECU* itu seperti *CPU* yang mengendalikan fungsi *hardware*.

Beberapa modul pada *ECU* antara lain :

- a) *ECM (Engine Control Module)*  
Merupakan module yang khusus mengatur kinerja mesin. Dari mulai proses starting, penyalan busi, injeksi bahan bakar, hingga proses cooling.
- b) *TCM (Transmission Control Module)*  
Module ini hanya ada pada mobil bertransmisi otomatis. Fungsinya jelas untuk mengatur perpindahan dan moment transmisi sesuai RPM mesin dan kondisi pengemudian.
- c) *PCM (Powertrain Control Module)*  
*Module* yang khusus mengatur kinerja sistem *powertrain* mobil. Biasanya *module* ini akan memastikan aliran tenaga dari mesin sampai ke roda dengan efisien. Dan module ini hanya ada pada beberapa mobil saja.
- d) *BCM (Body Control Module)*  
Adalah *module* yang khusus mengatur kinerja kelistrikan *body* seperti lampu, *horn*, *wiper* otomatis/manual. Dan sistem hiburan pada *dashboard*.



Gambar 2.1 Engine Control Unit (ECU)

(sumber : <https://mobilmu.com/perawatan-mobil/mengenal-lebih-dalam-apa-itu-electronic-control-unit-ecu-aid3432>)

### 2.2 OBD

*On Board Diagnostics* atau OBD, merupakan istilah otomotif yang mengacu pada kemampuan diagnostik pada kendaraan. Sistem OBD memberikan fasilitas kepada pengguna kendaraan atau teknisi untuk dapat mengakses status kendaraan dalam berbagai sub sistem. OBD membantu untuk memantau kondisi kendaraan dengan mengambil data dari *Engine Control Unit* (ECU). Untuk membaca ECU perlu peralatan tambahan yaitu *scanner* yang umumnya hanya untuk satu merk kendaraan dan harganya relatif mahal.

Sistim OBD memiliki sistim berbasis mikrokontroler yang dapat mendiagnosa kesalahan dalam parameter. Ketika kerusakan terdeteksi sistem harus dapat menyimpan detail kerusakan yang kemudian akan dianalisa oleh teknisi dan menghidupkan indikator kerusakan mesin *malfunction indicator light* (MIL) pada dasbor kendaraan.



Gambar 2.2 Malfunction Indicator Light (MIL)  
(sumber : <https://ripleysmufflerandbrakes.com/dash-lights-troubleshoot/>)

Seluruh fungsi OBD terhubung pada sebuah DLC (*Data Link Connector*) pada mobil, dengan satu panel maka kita bisa mengetahui kejanggalan apa saja yang terjadi pada mobil.

**2.3 CAN (11-bit) bus format**

*Query* dan *Response PID* terjadi pada *CAN bus* kendaraan. *Query* dan *Response OBD standard* menggunakan alamat fungsional. Pembaca diagnostik memulai sebuah permintaan menggunakan *CAN-ID* 7DFh, yang bertindak sebagai alamat *broadcast*, dan 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. *Response* memiliki *ID* dari *ID* yang ditugaskan ditambah 8 misalnya 7E8h hingga 7EFh.

*Query*

Fungsional *Query ID* dikirim ke kendaraan melalui *CAN bus* pada *ID* 7DFh, menggunakan 8 *data bytes*. Yaitu:

Tabel 2.1 *Query PID*

PID Type	Byte							
	0	1	2	3	4	5	6	7
SAE Standard	Number of additional data bytes: 2	Mode 01 = show current data; 02 = freeze frame; etc.	PID code (e.g.: 05 = Engine coolant temperature)	not used (may be 55h)				

*Response*

Kendaraan merespon *PID query* pada *CAN bus* dengan *message IDs*, yang tergantung pada modul mana yang merespon. Biasanya mesin atau *ECU* utama merespon di ID7E8h. Ini 8h lebih tinggi dari alamat fisik modul yang ditanggapi.

Tabel 2.2 *Response PID*

PID Type	Byte							
	0	1	2	3	4	5	6	7
SAE Standard 7E8h, 7E9h, 7EAh, etc.	Number of additional data bytes: 3 to 6	Custom mode Same 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)

Tabel 2.3 OBD2 PID

PID (hex)	Data bytes returned	Description	Min value	Max value	Units	Formula <sup>[a]</sup>
00	4	PIDs supported [01 - 20]				Bit encoded [A7..D0] == [PID \$01..PID \$20] See below
01	4	Monitor status since DTCs cleared. (Includes malfunction indicator lamp (MIL) status and number of DTCs.)				Bit encoded. See below
02	2	Freeze DTC				
03	2	Fuel system status				Bit encoded. See below
04	1	Calculated engine load	0	100	%	$\frac{100}{255}A$ (or $\frac{A}{2.55}$ )
05	1	Engine coolant temperature	-40	215	°C	$A - 40$
06	1	Short term fuel trim—Bank 1	-100 (Reduce Fuel: Too Rich)	99.2 (Add Fuel: Too Lean)	%	$\frac{100}{128}A - 100$ (or $\frac{A}{1.28} - 100$ )
07	1	Long term fuel trim—Bank 1				
08	1	Short term fuel trim—Bank 2				
09	1	Long term fuel trim—Bank 2				
0A	1	Fuel pressure (gauge pressure)	0	765	kPa	$3A$
0B	1	Intake manifold absolute pressure	0	255	kPa	$A$
0C	2	Engine RPM	0	16,383.75	rpm	$\frac{256A + B}{4}$
0D	1	Vehicle speed	0	255	km/h	$A$
0E	1	Timing advance	-64	63.5	° before TDC	$\frac{A}{2} - 64$
0F	1	Intake air temperature	-40	215	°C	$A - 40$
10	2	MAF air flow rate	0	655.35	grams/sec	$\frac{256A + B}{100}$
11	1	Throttle position	0	100	%	$\frac{100}{255}A$

Contoh cara menghitung *Query* dan *Response PID*:

Get RPM =

*Query*

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

*Response*

0x04, 0x41, 0x0C, 0x0F, 0xA0, 0x00, 0x00, 0x00

(41 0C) adalah sebuah *response* dari *query PID* (01 0C), dan dua *bytes* setelahnya merupakan nilai dari RPM (0F A0). Untuk mendapatkan nilai RPM yang sesungguhnya adalah dengan cara mengubah nilai RPM dari heksadesimal ke desimal kemudian hitung menggunakan formula RPM  $((A*256)+B)/4$ .

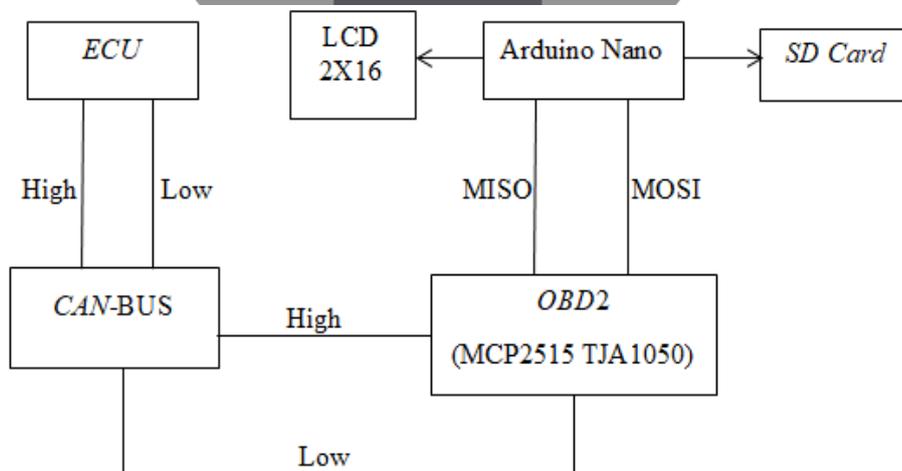
0Fh = 15d

A0h = 160d

$((15*256) + 160)/4 = 1000$  rpm

### 3. Perancangan Sistem

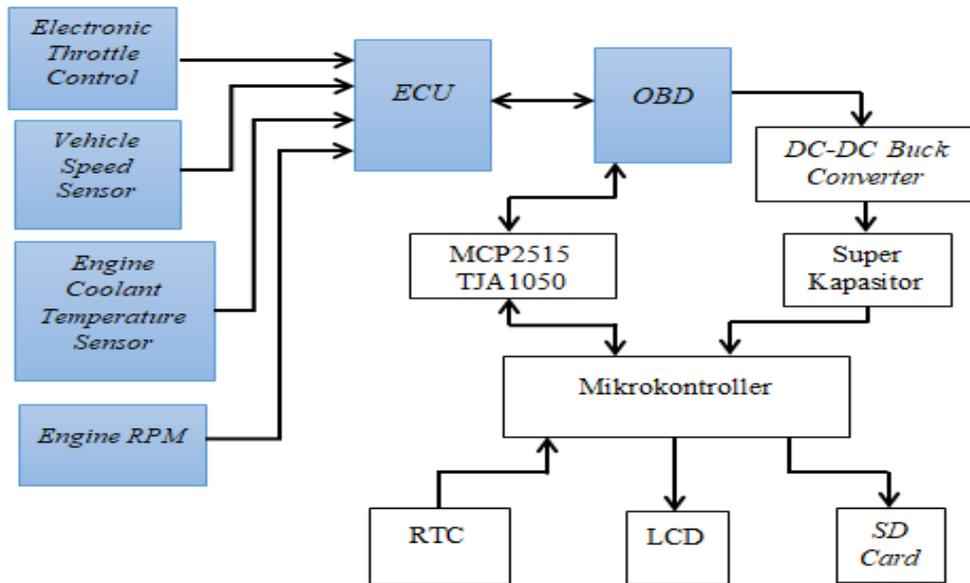
Perancangan sistem merupakan pengembangan secara teknik dan analisa. Perancangan sistem pada tugas akhir ini meliputi desain sistem yang akan dibangun, diagram blok sistem, desain perangkat keras dan perangkat lunak, proses kerja sistem, dan pemilihan komponen elektronika yang akan digunakan. Desain sistem secara umum sangat penting, karena dengan adanya gambaran umum sistem dapat dijadikan acuan untuk merancang perangkat keras yang akan digunakan dalam sistem.



Gambar 3.1 Diagram Blok Sistem Secara Umum

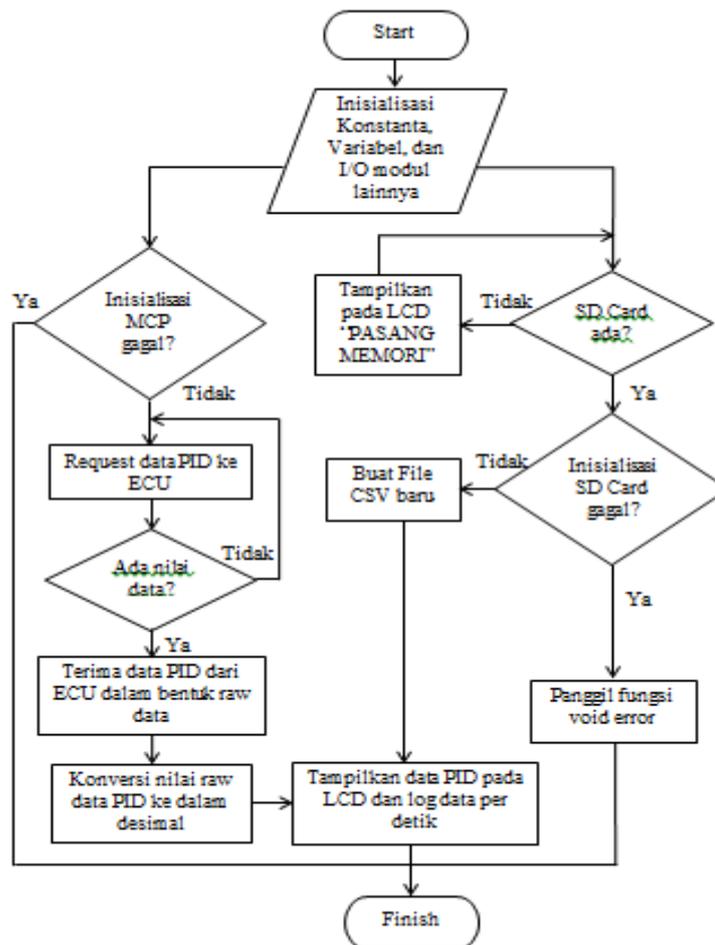
### 3.1. Diagram Blok Perangkat Keras

Diagram blok perangkat keras merupakan suatu hal yang penting dalam perancangan *sistem ECU Logger* agar berjalan dengan lancar. Pada gambar 3-2 merupakan diagram blok perangkat keras sistem *ECU Logger*.



Gambar 3.2 Diagram Blok Perangkat Keras

### 3.2. Flowchart Program



Gambar 3.3 Flowchart Program

### 3.3. Fungsi dan Fitur Perangkat

Pemilihan komponen elektronika yang digunakan pada ECU Logger adalah sebagai berikut :

#### 1. Arduino Nano (ATmega328)

Mikrokontroler yang digunakan adalah Arduino Nano berbasis ATmega 328. Arduino Nano memiliki ukuran yang kecil dan dapat memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah komputer dengan sebuah kabel *USB* atau mensuplainya dengan sebuah adaptor *AC* ke *DC* atau menggunakan baterai untuk memulainya

#### 2. MCP2515 CAN Bus Modul TJA1050

IC MCP2515 adalah pengontrol utama yang secara internal terdiri dari tiga sub-komponen utama yaitu modul *CAN*, logika kontrol, dan blok *SPI*. IC TJA1050 berfungsi sebagai antarmuka antara MCP2515 CAN kontroler dan fisik *CAN* bus, IC ini bertanggung jawab untuk mengambil data dari pengontrol dan mengirimkannya ke bus.

#### 3. Super Kapasitor

Pada ECU *Logger* terdapat superkapasitor yang bertujuan untuk menyimpan daya sehingga ECU *Logger* masih dapat bekerja pada saat tidak ada daya yang masuk dari aki mobil.

#### 4. Mini 360 DC-DC Buck Converter Step Down

Mini 360 DC-DC Buck Converter StepDown merupakan konverter penurun tegangan yang mengkonversikan tegangan masukan *DC* menjadi tegangan *DC* lainnya yang lebih rendah. Mini 360 DC-DC Buck Converter StepDown inilah yang akan menurunkan tegangan dari aki mobil sehingga sesuai dengan tegangan yang dibutuhkan oleh mikrokontroler.

#### 5. Micro SD Card reader modul SPI

Module micro SD merupakan modul untuk mengakses memori card yang bertipe *micro SD* untuk pembacaan maupun penulisan data dengan menggunakan sistem antarmuka *SPI* (*Serial Parallel Interface*). Modul ini cocok untuk berbagai aplikasi yang membutuhkan media penyimpanan data, maupun sistem aplikasi data *logging* lainnya.

#### 6. Micro SD Card

*SD Card* merupakan sebuah alat penyimpan berukuran kecil yang digunakan untuk menyimpan data. Hasil perekaman data yang terdapat pada data *logger* akan tersimpan di *SD Card* yang kemudian dapat dibaca pada komputer menggunakan *card reader*.

#### 7. RTC DS1307

Salah satu chip RTC yang mudah digunakan adalah DS1307. Komponen DS1307 berupa IC yang perlu dilengkapi dengan komponen pendukung lainnya seperti *crystal* sebagai sumber *clock* dan *battery external* 3,6 volt sebagai sumber energi cadangan agar fungsi penghitung tidak berhenti

#### 8. Liquid Crystal Display (LCD)

*Liquid Crystal Display* (*LCD*) adalah komponen elektronika untuk menampilkan data berupa karakter. *LCD* tipe 20x4 memiliki batas menampilkan karakter hanya 4 baris yang di tiap barisnya dapat memuat 20 karakter.

#### 9. Inter Integrated Circuit (I2C)

*Inter Integrated Circuit* (*I2C*) adalah komponen elektronika standar komunikasi serial dua arah yang menggunakan dua saluran untuk mengirim dan menerima data. Untuk membawa informasi data antara *I2C* dengan pengontrolnya menggunakan saluran *Serial Clock* (*SCL*) dan *Serial Data* (*SDA*).

## 4. Percobaan dan Analisa

### 4.1 Pengujian Engine Speed (Engine RPM)

Tujuan pengujian parameter *Engine Speed* untuk mengetahui nilai perbandingan antara keluaran *Engine Speed* melalui ECU *Logger* dengan nilai keluaran yang ditampilkan pada *MIL* (*Malfunction Indicator Light*).

### 4.2 Pengujian Engine Coolant Temperature

Tujuan pengujian parameter *Engine Coolant Temperature* untuk mengetahui nilai perbandingan antara keluaran *Engine Coolant Temperature* melalui ECU *Logger* dengan nilai keluaran yang ditampilkan pada *MIL* (*Malfunction Indicator Light*).

### 4.3 Pengujian Vehicle Speed

Tujuan pengujian parameter *Vehicle Speed* untuk mengetahui nilai perbandingan antara keluaran *Vehicle Speed* melalui ECU *Logger* dengan nilai keluaran yang ditampilkan pada *MIL* (*Malfunction Indicator Light*).

### 4.4 Pengujian Throttle Position

Tujuan pengujian parameter *Throttle Position* untuk mengetahui nilai perbandingan antara keluaran *Throttle Position* melalui *ECU Logger* dengan nilai keluaran yang ditampilkan pada *MIL (Malfunction Indicator Light)*.

**4.5 Menampilkan data ECU yang terdeteksi di LCD**

Tujuan pengujian ini adalah untuk menampilkan nilai parameter sensor yang di deteksi seperti *Engine Speed, Vehicle Speed, Engine Coolant Temperature, dan Throttle Position* pada *Liquid Crystal Display (LCD)*. Sehingga pengguna *ECU Logger* dapat melihat nilai parameter sensor yang terdeteksi melalui LCD secara *real time*. Pengujian ini dilakukan dengan menggunakan beberapa komponen yaitu *Arduino Nano ATmega328, MCP2515 TJA1050 CAN Bus Modul, Mini 360 DC-DC Buck Converter StepDown, Kapasitor 5.5V/4.0F, Soket OBD2 16 pin, I2C LCD 20x4 Modul*.

Berikut adalah gambar LCD yang menampilkan nilai parameter sensor *Engine Speed, Vehicle Speed, Engine Coolant Temperature, dan Throttle Position* :



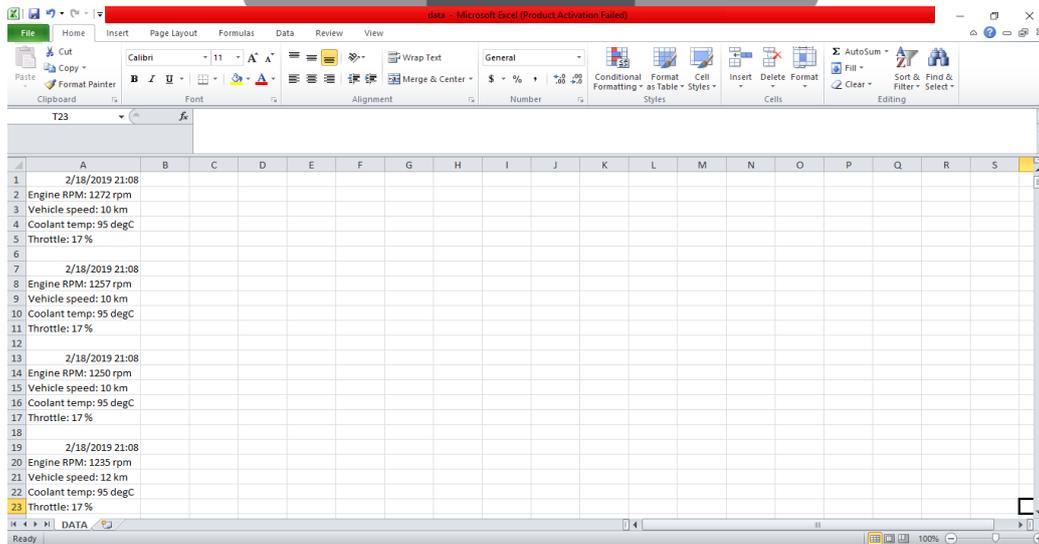
Gambar 4.1 Menampilkan data ECU yang terdeteksi di LCD

**4.6 Membaca data yang tersimpan di MicroSD Card menggunakan laptop**

**Tujuan Pengujian :**

Tujuan pengujian ini adalah untuk mengetahui semua data sensor yang dideteksi oleh ECU mobil. Data yang tersimpan dapat menjadi bahan penelitian dan analisa kerja sensor mobil pada saat mobil digunakan. Pengujian ini dilakukan dengan memasukkan *MicroSD Card* ke *SD Card Reader* lalu disambungkan ke *Laptop*.

Berikut adalah data yang tersimpan pada SD Card yang di tampilkan pada Laptop. Data yang tersimpan berupa waktu dan parameter sensor *Engine Speed, Vehicle Speed, Engine Coolant Temperature, dan Throttle Position*.



Gambar 4.2 Membaca data yang tersimpan di MicroSD menggunakan Laptop

## 5. Kesimpulan

Berdasarkan hasil penelitian dan pengerjaan tugas akhir ini, penulis mendapatkan beberapa kesimpulan, yaitu :

1. ECU *Logger* dapat mendeteksi sensor-sensor pada mobil dan menyimpan data tersebut ke dalam *microSD*.
2. ECU *Logger* ini dapat digunakan pada mobil yang menggunakan protokol ISO 15765 (menggunakan pin 4,5,6,14,16 pada soket OBD 16 pin).
3. Beberapa parameter sensor yang di monitoring oleh ECU *Logger* adalah *Vehicle Speed* (0 – 255 km/h), *Throttle Position* (0 – 100%), *Engine Speed* (0 - 16.383,7 RPM), *Engine Coolant Temperature* (-40°C s/d 215 °C), menggunakan 97% kapasitas memori *internal ATmega328*.
4. Sistem juga dapat menampilkan waktu pada saat itu juga seperti detik, menit, jam, hari, bulan, dan tahun.
5. Dengan kapasitas penyimpanan data pada SD Card sebesar 2GB dapat digunakan selama 1.042 hari (apabila mobil digunakan selama 12 jam/hari).

## 6. Saran

Saran untuk pengembangan sistem penyimpanan dan monitoring data elektronik mobil sebagai berikut:

1. Alat dapat dikembangkan dengan menambahkan *GPS* supaya pengguna kendaraan juga bisa melihat langsung data sensor yang terdeteksi.
2. Menambahkan parameter sensor pada ECU yang akan dideteksi.
3. Membuat ECU *Logger* supaya dapat digunakan untuk semua protokol dengan cara menggunakan pin-pin pada *socket OBD* sesuai dengan semua protokol.

## Daftar Pustaka :

- [1] Yanuar Prasajo Kusumo, Harianto, Madha Christian Wibowo. 2015. Rancang Bangun Sistem General Diagnostic Scanner Untuk Mengakses ECU Mobil Dengan Komunikasi Serial OBD-2. (<http://jurnal.stikom.edu/index.php/jcone/article/view/603>).
- [2] Yani Prabowo, I Wayan Degeng. 2016. Rancang Bangun Pembaca Display Data On Board Diagnostic (OBD) Mesin Mobil Berbasis Arduino.
- [3] Supriatna Adhisuwigno, Denda Dewatama. 2016. On Board Diagnostic (OBD) Reader Berbasis Arduino.
- [4] Di Natale, Marco dkk. 2012. *Understanding and Using the Controller Area Network Communication Protocol*. New York : Spinger –Verlag.