

Implementasi Komunikasi CAN-Bus Baterai pada Simulator Kendaraan Listrik Melalui Visualisasi Nextion

1st M. Aulia Rahman
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

rhhmanfte@student.telkomuniversity.a
c.id

2nd Irwan Purnama
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
Verified email at brin.go.id

3rd Angga Rusdinar
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
anggarusdinar@telkomuniversity.ac.id

Abstrak — penelitian ini menciptakan Electric Vehicle (EV) CAN-Bus Simulator sebagai alat pendidikan di SMK/PT. Simulator ini memungkinkan pengembangan keahlian dan pengetahuan tentang sistem komponen utama pada mobil listrik. Fokus utama simulator ini adalah kemampuannya untuk berkomunikasi melalui CAN-Bus dan menampilkan informasi data serta visual dari komponen-komponen yang akan digunakan. Hal ini menandakan bahwa komponen-komponen tersebut dapat berkomunikasi secara efektif melalui CAN-Bus

Kata kunci— Electric Vehicle, Controller Area Network Bus, Simulator

I. PENDAHULUAN

Seiring dengan kemajuan teknologi, keterampilan yang dimiliki juga harus update dan upgrade ke bidang automotive advanced, salah satunya adalah keterampilan menjelaskan komponen dan prinsip kerja Controlled Area Network Bus (CAN Bus) pada mobil listrik. Salah satu solusi adalah dengan adanya Electric Vehicle (EV) CAN-Bus Simulator yang membantu SMK/PT di bidang teknik otomotif dalam mengembangkan keahlian dan pengetahuan terutama pada mobil listrik.

Sistem kontrol kendaraan bermotor listrik saat ini pada umumnya berbasis CAN-Bus. CAN-Bus adalah teknologi yang mengintegrasikan komunikasi kontrol, akuisisi sinyal, dan pemrosesan data dari sistem komponen utama pada mobil listrik. CAN-Bus memiliki kemampuan komunikasi fleksibel yang beroperasi dalam mode multi-master, yang memungkinkan setiap node dalam jaringan untuk secara aktif mengirim informasi ke node lainnya kapan saja. Teknologi ini juga dapat meminimalisasi interferensi dan mendeteksi kesalahan pada sistem melalui pemeriksaan Cyclic Redundancy Check (CRC) dan metode pemeriksaan kendala komunikasi data lainnya.

Dalam EV CAN-Bus Simulator dapat memberikan informasi dari sistem komponen yang harus di-monitoring dalam mobil listrik. Pada sistem mobil listrik memiliki komponen sistem kendali salah satunya yaitu Battery Management System (BMS), On-Board Charger (OBC), Traction Inverter dan sensor lainnya. Pada Battery Management System (BMS) berfungsi untuk memonitor tegangan, arus, suhu, dan persentase baterai. sistem propulsi kendaraan. Serta sensor seperti, sensor suhu pada baterai.

Semua sistem tersebut diintegrasikan melalui Electric Control Unit (MASTER). Nantinya sistem tersebut akan menghasilkan keluaran yang dapat di-monitoring melalui dashboard Graphic User Interface (GUI). Parameter keluaran yang ada pada dashboard GUI yaitu Persentase baterai (State of Charge), Kesehatan baterai (State of Health), tegangan dan arus pada baterai, temperature baterai.

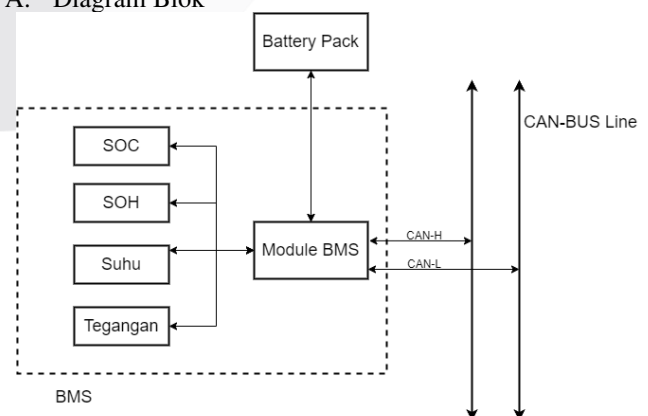
II. KAJIAN TEORI

A. Penjelasan Sistem

Battery management system berfungsi untuk memantau dan mengelola data baterai pada kendaraan listrik. Dalam sistem ini, data baterai yang berasal dari modul CAN slave dikumpulkan dan dianalisis. Data yang dikumpulkan mencakup berbagai parameter penting seperti State of Charge (SoC), State of Health (SoH), Suhu, dan tegangan baterai. Setelah data diterima dan diproses oleh modul CAN slave, informasi ini kemudian dikirimkan ke modul CAN master untuk pengolahan lebih lanjut. Baterai yang digunakan adalah baterai dari motor Gesits dengan spesifikasi 72V 20Ah.

III. METODE

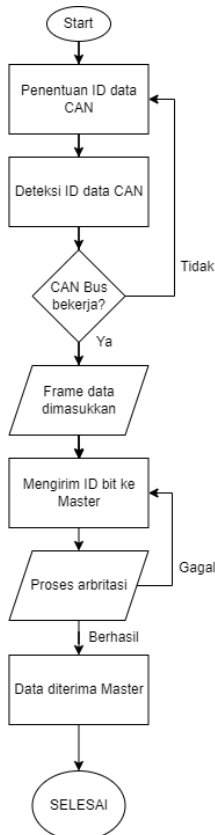
A. Diagram Blok



GAMBAR 1
Diagram Blok CAN-Bus Slave Baterai

Diagram blok ini menggambarkan arsitektur Sistem Manajemen Baterai (BMS) di mana data yang diperoleh dari baterai, yaitu tegangan, SoC (State of Charge), SoH (State of Health), dan suhu, akan dikirimkan melalui CAN bus ke ECU (Electronic Control Unit) sebagai master.

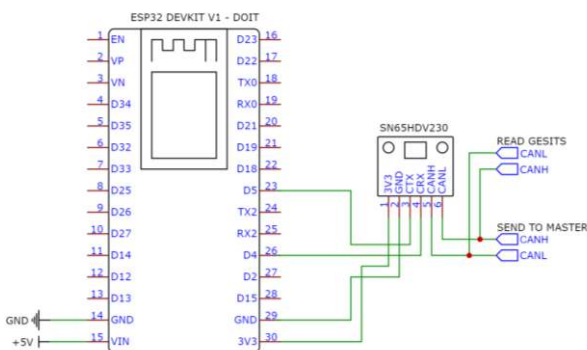
B. Flowchart



GAMBAR 2
Flowchart Komunikasi CAN-Bus

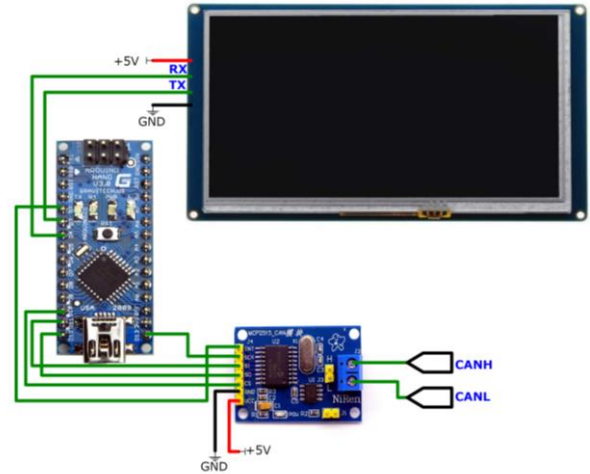
Diagram alir atau *flowchart* pada gambar 2 menggambarkan proses pengiriman sampai penerimaan data melalui komunikasi CAN-Bus. Proses dimulai penentuan *frame ID* BMS baterai Gesits, kemudian setelah mendeteksi *frame ID* yang diinginkan maka data komunikasi dapat diterima. *Frame ID* mengalami proses arbitrase bit untuk menghindari node bit yang tidak diinginkan masuk ke jalur komunikasi. Jika arbitrase berhasil, maka *frame ID* berhasil data dapat diterima oleh master.

C. Schematic Diagram



GAMBAR 3
Skematik sistem CAN-Bus slave baterai

Pada gambar 3 terlihat bahwa sistem ini menggunakan ESP32 sebagai mikrokontroler untuk CAN-Bus *slave* baterai dan CAN tranceiver 3.3V menggunakan *module* SN65hDV230 sebagai jembatan antara komunikasi CAN-Bus yang mengubah dari sinyal digital menjadi diferensial yang dapat dikirim melalui kabel CAN dan sebaliknya juga. Pada sistem menerima data baterai gesits yang didalam baterai sudah memiliki CAN tranceiver. Setelah data diterima data CAN kemudian dikirimkan ke sistem CAN-bus ECU (*master*).



GAMBAR 4
Skematik sistem CAN-Bus ECU (Master)

Sedangkan pada gambar 4 sistem CAN-Bus master menggunakan Arduino Nano sebagai mikrokontroler dan CAN tranceiver 5v menggunakan module MCP2515. Fungsi sistem ini sebagai penerima seluruh data *frame ID* yang telah ditentukan. Data yang diperoleh kemudian ditampilkan melalui GUI (*Graphic User Interface*) menggunakan Nextion NX8048K050. GUI pada sistem berfungsi representasi visual data CAN-bus yang akan ditampilkan.

IV. HASIL DAN PEMBAHASAN

A. Implementasi Sistem

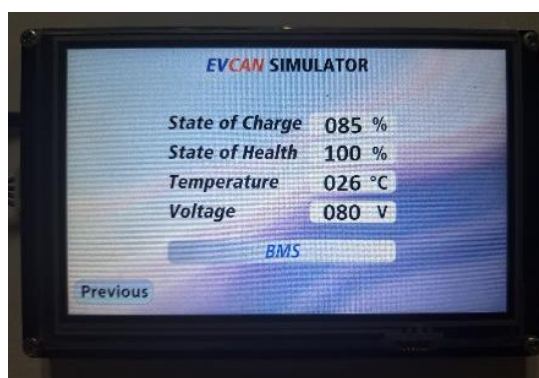


GAMBAR 5
Implementasi Sistem

Gambar 5 diatas merupakan implementasi pengujian komunikasi CAN-bus slave baterai. Pengujian komunikasi dilakukan selama 20 detik untk mengetahui apakah data yang dikirimkan sesuai Frame ID yang diterima.

TABEL 1
Hasil Pengujian komunikasi CAN-Bus Baterai

No	Waktu	Status	Data yang Diterima
1	20:49:00.965	Terkirim	SoC:86 SoH:100 Temp:26 V:80
2	20:49:01.948	Terkirim	SoC:86 SoH:100 Temp:26 V:80
3	20:49:02.965	Terkirim	SoC:86 SoH:100 Temp:26 V:80
4	20:49:03.951	Terkirim	SoC:86 SoH:100 Temp:26 V:80
5	20:49:04.968	Terkirim	SoC:86 SoH:100 Temp:26 V:80
6	20:49:05.968	Terkirim	SoC:86 SoH:100 Temp:26 V:80
7	20:49:06.950	Terkirim	SoC:86 SoH:100 Temp:26 V:80
8	20:49:07.941	Terkirim	SoC:86 SoH:100 Temp:26 V:80
9	20:49:08.962	Terkirim	SoC:86 SoH:100 Temp:26 V:80
10	20:49:09.963	Terkirim	SoC:86 SoH:100 Temp:26 V:80
11	20:49:10.965	Terkirim	SoC:86 SoH:100 Temp:26 V:80
12	20:49:11.948	Terkirim	SoC:86 SoH:100 Temp:26 V:80
13	20:49:12.965	Terkirim	SoC:86 SoH:100 Temp:26 V:80
14	20:49:13.951	Terkirim	SoC:86 SoH:100 Temp:26 V:80
15	20:49:14.968	Terkirim	SoC:86 SoH:100 Temp:26 V:80
16	20:49:15.968	Terkirim	SoC:86 SoH:100 Temp:26 V:80
17	20:49:16.950	Terkirim	SoC:86 SoH:100 Temp:26 V:80
18	20:49:17.941	Terkirim	SoC:86 SoH:100 Temp:26 V:80
19	20:49:18.962	Terkirim	SoC:86 SoH:100 Temp:26 V:80
20	20:49:19.963	Terkirim	SoC:86 SoH:100 Temp:26 V:80



GAMBAR 6
Hasil Tampilan GUI Master

Berdasarkan tabel 1 dan gambar 6, ditunjukkan hasil dari pengujian pengiriman data dari slave baterai ESP32 ke master Arduino selama 20 detik, data berhasil dikirim melalui CAN bus dan data berhasil ditampilkan pada GUI. Data yang dikirim tidak terjadi kesalahan dalam komunikasi CAN bus selama pengujian, baik pada serial monitor slave ESP32 maupun master Arduino.

V. KESIMPULAN

Pada pengujian ini, CAN bus pada slave baterai dapat berkomunikasi dengan ESP32. Data *frame ID* pada CAN baterai Gesits dapat diklasifikasikan sehingga data yang diinginkan terbaca dengan baik. Data *frame ID* CAN dapat diterima tanpa kesalahan oleh CAN-Bus master sehingga berhasil ditampilkan oleh *Graphic User Interface*.

REFERENSI

- [1] Aziz, Mochammad dkk. (2020, 1 Maret). Studi Analisis Perkembangan Teknologi dan Dukungan Pemerintah Indonesia Terkait Mobil Listrik. TESLA, Vol 22 No 1.
- [2] Suranto.G.(2019, 4 September). Menristekdikti: Pengembangan Kendaraan Listrik Butuh Ekosistem Pendukung. Diakses dari <https://ristekdikti.go.id/kabar/menristekdiktipengembangan-kendaraan-listrik-butuh-ekosistem-pendukung>.
- [3] Dreeskandar, Willy & Pandjaitan, R.H. (2020, 13 February). Peluang Sosialisasi Edukatif Kendaraan Eelektrik Melalui Kerja sama Perguruan Tinggi Dengan Produsen. Prosiding Konferensi Pendidikan Nasional "Strategi dan Implementasi Pendidikan Karakter pada Era Revolusi Industri 4.0".
- [4] Wang, Yingshun. (2021, Mar). Design of Electric Drive System of Electric Vehicle Based on CAN Bus. Journal of Physics: Conference Series. Vol 1982.
- [5] Datanesia. (2022, Mei). Gairah Transisi Energi: Sektor usaha sudah mulai bergerak ke bisnis energi baru terbarukan. Komitmen dan keseriusan pemerintah menjadi taruhannya. White Paper, Edisi: 123.
- [6] Indra Kusuma, Fuad dkk. (2023, 11 December). Peningkatan Penguasaan Konsep dan Struktur Ilmu Guru Teknik Otomotif Melalui Diseminasi Teknologi Controlled Area Network Bus (CAN Bus) di SMKN Palang, IRA Jurnal Pengabdian Kepada Masyarakat (IRAJPKM), Vol. 1, No. 3, 2023, pp. 21-27, e-ISSN: 3024-8299.
- [7] Sekretariat Jendral, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi. Inovasi. (2022, 7 December). Pendidikan Vokasi untuk Kendaraan Listrik Indonesia Telah Berjalan & akan Terus Ditingkatkan. Diakses dari <https://setjen.kemdikbud.go.id/berita-inovasi-pendidikan-vokasi-untuk-kendaraan-listrik-indonesia-telah-berjalan--akan-terus-ditingkatkan.html>.
- [8] Woojin. J, Sungmin. H. Eunmin Choi, Seonghun Lee, and Ji-Woong Choi. (2020, November). CNN-Based Adaptive Source Node Identifier for Controller Area

Network (CAN). IEEE Transactions on Vehicular Technology, VOL. 69, NO. 11.

[9] Skywork. Isolation in Electric Vehicle Systems, Quick Reference Guide. Skyworkinc.com, 2022.

[10] Prassanna Madhavan, A. Ranjeev. (2020, September). Battery Management System Integrated with

CAN BUS Safety Control Environment for Electric Vehicle. International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 9 Issue 09.

[11] Steve Corrigan, Texas Instruments. (2019) Introduction to the Controller Area Network (CAN). Application Report, SLOA101 – August 2002, Revised 2019.

