

**PERANCANGAN PERANTI PENGUKURAN
SPEKTROMETRI IMPEDANSI ELEKTROKIMIA
BERBASIS EVAL-AD5941BATZ UNTUK KARAKTERISASI
BATERAI LITIUM-ION BEKAS**

SKRIPSI



Disusun oleh:

**Bonaventura Riko Kumara Datta
20/463235/TK/51227**

**PROGRAM STUDI TEKNIK ELEKTRO
DEPARTEMEN TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI
FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA
YOGYAKARTA
2024**

HALAMAN PENGESAHAN

PERANCANGAN PERANTI PENGUKURAN SPEKTROMETRI IMPEDANSI ELEKTROKIMIA BERBASIS EVAL-AD5941BATZ UNTUK KARAKTERISASI BATERAI LITIUM-ION BEKAS

SKRIPSI

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik
pada Departemen Teknik Elektro dan Teknologi Informasi
Fakultas Teknik
Universitas Gadjah Mada

Disusun oleh:

Bonaventura Riko Kumara Datta
20/463235/TK/51227

Telah disetujui dan disahkan

Pada tanggal

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Eka Firmansyah, S.T., M.Eng., Ph.D., IPM.
NIP 197903032002121004

Ridwan Wicaksono, S.T., M.Eng., Ph.D.
NIP 111199010202301101

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini :

Nama : Bonaventura Riko Kumara Datta
NIM : 20/463235/TK/51227
Tahun terdaftar : 2020
Program Studi : Teknik Elektro
Fakultas : Teknik Universitas Gadjah Mada

Menyatakan bahwa dalam dokumen ilmiah Skripsi ini tidak terdapat bagian dari karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/lembaga lain, kecuali yang secara tertulis disitasi dalam dokumen ini dan disebutkan sumbernya secara lengkap dalam daftar pustaka.

Dengan demikian saya menyatakan bahwa dokumen ilmiah ini bebas dari unsur-unsur plagiasi dan apabila dokumen ilmiah Skripsi ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis lain dan/atau dengan sengaja mengajukan karya atau pendapat yang merupakan hasil karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Yogyakarta, 27 Juni 2024



Bonaventura Riko Kumara Datta
20/463235/TK/51227

HALAMAN PERSEMBAHAN

Puji dan syukur kepada Tuhan Yang Maha Esa,
atas berkat dan rahmat yang diberikan sehingga
penulis dapat menyelesaikan penelitian ini yang berjudul
“Perancangan Peranti Pengukuran Spektrometri Impedansi Elektrokimia
Berbasis EVAL-AD5941BATZ untuk Karakterisasi Baterai Litium-ion Bekas”.
Tugas akhir ini penulis persembahkan kepada kedua orang tua, kepada keluarga dan
teman-teman semua, serta untuk bangsa, negara, dan agama.

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga tugas akhir berupa penyusunan skripsi ini telah terselesaikan dengan baik. Dalam hal penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Ir. Hanung Adi Nugroho, S.T., M.E., Ph.D., IPM. selaku Ketua Departemen Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada.
2. Bapak Ir. Lesnanto Multa Putranto, S.T., M.Eng., Ph.D., IPM. selaku Sekretaris Departemen Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada yang telah membantu kelancaran administrasi untuk pelaksanaan kerja praktik ini.
3. Bapak Ir. Eka Firmansyah, S.T., M.Eng., Ph.D., IPM. selaku Dosen Pembimbing yang telah membimbing dan memberi nasihat sehingga penelitian dapat berjalan dengan lancar.
4. Bapak Ridwan Wicaksono, S.T., M.Eng., Ph.D. selaku Dosen Pembimbing yang telah membimbing dan memberi nasihat sehingga penelitian dapat berjalan dengan lancar.
5. Seluruh dosen dan tenaga pengajar DTETI FT UGM atas ilmu yang diberikan se-lama masa studi.
6. Kedua Orang Tua dan adik-adik yang selalu memberikan dukungan selama belajar dan menyelesaikan tugas akhir ini.
7. Bapak Yaenuri, A.Md dan Bapak Sunandar, S.T. selaku Tenaga Kependidikan Laboratorium Instalasi yang dengan murah hati membantu menyediakan tempat dan peralatan penelitian sehingga penelitian dapat berjalan dengan nyaman dan lancar.
8. Sahabat Holy Garage tercinta yang dengan setia menemani dan menghibur dinamika pembelajaran dari awal kuliah hingga kelulusan.
9. Teman-teman Laboratorium Instalasi yang selalu menemani proses penelitian mulai hingga kelulusan.
10. Sheren yang dengan setia dan sabar menemani dan memberi masukan selama proses penelitian berlangsung.
11. Semua pihak yang telah berkontribusi secara langsung maupun tidak langsung dalam penyelesaian skripsi ini.

Penulis menyadari bahwa dalam proses penulisan laporan ini masih banyak kekurangan yang berkaitan dengan tata bahasa dan penulisan lainnya. Penulis menyadari akan keterbatasan pengetahuan dan keterampilannya dalam menulis laporan ini. Kritik dan saran ataupun keperluan lain dapat disampaikan melalui email pribadi penulis rikokumara0365@gmail.com. Akhir kata, penulis berharap semoga skripsi ini dapat memberikan manfaat bagi kita semua. Aamiin.

DAFTAR ISI

HALAMAN PENGESAHAN	ii
PERNYATAAN BEBAS PLAGIASI	iv
HALAMAN PERSEMBAHAN	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vi
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
DAFTAR SINGKATAN.....	xii
INTISARI.....	xiii
ABSTRACT	xiv
BAB I Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Batasan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan.....	3
BAB II Tinjauan Pustaka dan Dasar Teori	4
2.1 Tinjauan Pustaka	4
2.2 Dasar Teori	6
2.2.1 Baterai	6
2.2.2 Baterai Litium-ion	7
2.2.3 Baterai Litium-ion Berdasar Bahan Penyusun Katoda	8
2.2.4 Model Ekuivalen Baterai Litium-ion.....	9
2.2.4.1 Model Rint	9
2.2.4.2 Model Thevenin	9
2.2.4.3 Model RC Orde Kedua	10
2.2.4.4 Model PNGV	11
2.2.5 Penuaan dan Degradasi Baterai Litium-ion.....	11
2.2.6 <i>Electrochemical Impedance Spectroscopy</i> (EIS)	12
2.2.6.1 Isyarat Eksitasi.....	12
2.2.6.2 Pengolahan Hasil Pengukuran EIS.....	13
2.2.7 Karakterisasi Hasil EIS	14
2.2.7.1 Model Sirkuit Randel— <i>Charge Transfer Effect</i>	15
2.2.7.2 Model Sirkuit Warburg— <i>Diffusion Effect</i>	17
2.2.7.3 Gabungan Model Sirkuit Randel and Warburg	17

2.2.8	IC Analog Front-End AD5941	19
2.2.9	Modul EVAL-AD5941BATZ	20
2.2.10	STM32F411CEU6	21
2.2.10.1	<i>Serial Peripheral Interface (SPI)</i>	21
2.2.10.2	<i>Universal Asynchronous Receiver/Transmitter (UART)</i> ..	21
2.2.10.3	<i>General Purpose Input/Output (GPIO)</i>	22
BAB III	Metode Penelitian.....	23
3.1	Alat dan Bahan Penelitian	23
3.2	Alur Penelitian Tugas Akhir.....	24
3.3	Gambaran Umum Sistem	25
3.4	Perancangan Perangkat Keras	26
3.4.1	Rancangan <i>Serial Peripheral Interface (SPI)</i>	27
3.4.2	Rancangan <i>Universal Asynchronous Receiver Transmitter (UART)</i>	28
3.4.3	Rancangan <i>General Purpose Input/Output (GPIO)</i>	29
3.4.4	Desain Sirkuit PCB Antarmuka	30
3.4.5	Routing dan Layouting PCB Antarmuka	32
3.4.6	<i>Bill of Materials</i>	36
3.5	Perancangan Perangkat Lunak untuk Mikrokontroler.....	36
3.6	Perancangan Perangkat Lunak untuk <i>Graphical User Interface</i>	38
3.6.1	Gambaran Umum GUI.....	38
3.6.2	Penggunaan <i>Library</i> dalam Perancangan GUI.....	39
3.6.3	Tampilan Akhir GUI	41
BAB IV	Hasil dan Pembahasan.....	43
4.1	Implementasi Perangkat Keras	43
4.2	Pengujian USB-to-UART TTL CH340G	44
4.2.1	Protokol Komunikasi UART.....	44
4.2.2	Skema dan Pengaturan Pengujian	45
4.2.3	Hasil Pengujian UART	45
4.3	Pengujian Modul EVAL-AD5941BATZ	46
4.3.1	Skema dan Pengaturan Pengujian	47
4.3.2	Protokol Perintah SPI IC AD5941	47
4.3.3	Pengujian Inisialisasi	47
4.3.4	Pengujian Akses FIFO	50
4.4	Pengujian GUI	51
4.4.1	Skema Pengujian GUI	51
4.4.2	Pengujian COM Manager.....	51
4.4.3	Pengujian Pengaturan Parameter Pengukuran	52
4.4.4	Pengujian Pengukuran	54
4.4.5	Pengujian Tampilan Plot.....	54

4.5	Pengujian Fungsi Pengukuran Impedansi.....	56
4.5.1	Perhitungan Impedansi Hasil EIS	56
4.5.2	Pengaturan Parameter Pengukuran	57
4.5.3	Skema dan Susunan Peranti Pengukuran	58
4.5.4	Baterai Objek Pengukuran	59
4.5.5	Hasil Pengujian.....	59
4.6	Pengujian Akurasi Pengukuran dan Karakterisasi Baterai Litium-ion	61
4.6.1	Metode dan Objek Pengujian.....	61
4.6.2	Pengaturan Parameter Pengukuran	62
4.6.3	Hasil Pengukuran	62
4.6.4	Karakterisasi Hasil Pengukuran Berdasar Model dan Kurva Karakteristik	67
4.6.4.1	Perhitungan Resistansi Elektrolit	67
4.6.4.2	Perhitungan Resistansi Transfer Muatan	68
4.6.4.3	Perhitungan Kapasitansi Lapisan Ganda	69
4.6.5	Analisis Akurasi Pengukuran.....	70
4.6.5.1	Pengukuran Resistansi Elektrolit	71
4.6.5.2	Pengukuran Resistansi Transfer Muatan	72
4.6.5.3	Pengukuran Kapasitansi Lapisan Ganda.....	73
4.7	Pengujian Pengukuran Baterai Litium-ion Lainnya	75
4.8	Spesifikasi Akhir Peranti Pengukuran	77
BAB V	Kesimpulan dan Saran	78
5.1	Kesimpulan.....	78
5.2	Saran.....	78
DAFTAR PUSTAKA		79
LAMPIRAN		L-1
L.1	Program STM32	L-1
L.1.1	Program main.c	L-1
L.1.2	Program STM32F411Port.c	L-3
L.1.3	Program AD5940Main.c.....	L-6
L.2	Source Code GUI	L-8
L.2.1	Program Master.py	L-8
L.2.2	Program GUI_Master.py.....	L-8
L.2.3	Program Serial_Com_ctrl.py	L-15
L.2.4	Program Data_Com_ctrl.py	L-24
L.3	Tabel Hasil Pengukuran	L-26
L.3.1	Hasil Pengujian Fungsi Pengukuran	L-26
L.3.2	Hasil Pengujian Akurasi Pengukuran	L-27
L.4	Alat dan Bahan Penelitian	L-35

DAFTAR TABEL

Tabel 2.1	Tinjauan Pustaka	5
Tabel 2.2	Performa baterai litium-ion berdasarkan penyusun katoda.....	8
Tabel 3.1	Parameter minimum produksi PCB lokal	32
Tabel 3.2	Lebar jalur yang digunakan pada PCB Antarmuka	33
Tabel 3.3	<i>Bill of Materials</i> peranti pengukuran impedansi	36
Tabel 4.1	Perintah GUI untuk perangkat	44
Tabel 4.2	Respon Perangkat terhadap perintah GUI.....	44
Tabel 4.3	Byte perintah komunikasi SPI	47
Tabel 4.4	Perintah inisiasi SPI	48
Tabel 4.5	Spesifikasi baterai pengujian peranti pengukuran	59
Tabel 4.6	Parameter baterai objek pengujian.....	61
Tabel 4.7	Nilai resistansi elektrolit untuk setiap pengukuran baterai	68
Tabel 4.8	Nilai resistansi transfer muatan untuk setiap pengukuran baterai	69
Tabel 4.9	Nilai kapasitansi lapisan ganda untuk setiap pengukuran baterai	70
Tabel 4.10	Perhitungan RSD untuk pengukuran R_S	71
Tabel 4.11	Perhitungan RSD untuk pengukuran R_{CT}	72
Tabel 4.12	Perhitungan RSD untuk pengukuran C_{DL}	73
Tabel 4.13	Merek dan parameter baterai litium-ion objek pengujian	75
Tabel 4.14	Spesifikasi Akhir Peranti Pengukuran	77
Tabel 1	Hasil pengukuran pengujian fungsi perangkat	L-27
Tabel 2	Pengukuran impedansi baterai Samsung INR-25R2 Baru.....	L-28
Tabel 3	Pengukuran impedansi baterai Samsung INR-25R2 Bekas Baru	L-30
Tabel 4	Pengukuran impedansi baterai Samsung INR-25R2 Bekas Tua	L-32
Tabel 5	Rerata pengukuran setiap sampel baterai.....	L-34

DAFTAR GAMBAR

Gambar 2.1	Aliran arus ion litium saat <i>charging</i>	7
Gambar 2.2	Model Sirkuit Rint	9
Gambar 2.3	Model Sirkuit Thevenin	10
Gambar 2.4	Model Sirkuit RC Orde Kedua	10
Gambar 2.5	Model Sirkuit Ekuivalen PNGV.....	11
Gambar 2.6	Plot nyquist hasil EIS baterai litium-ion	14
Gambar 2.7	Model Sirkuit Randel	16
Gambar 2.8	Plot Nyquist Sirkuit Randel sederhana.....	16
Gambar 2.9	Model Sirkuit Warburg	17
Gambar 2.10	Plot Nyquist Sirkuit Warburg dengan efek difusi	17
Gambar 2.11	Model Sirkuit Warburg	18
Gambar 2.12	Plot Nyquist Model Sirkuit Randel yang dimodifikasi.....	18
Gambar 2.13	Diagram blok fungsional IC AD5491.....	19
Gambar 2.14	Sirkuit yang disederhanakan dari Modul EVAL-5941BATZ	20
Gambar 2.15	Modul EVAL-AD5941BATZ.....	20
Gambar 2.16	STM32F411CEU6 "Black Pill"	21
Gambar 3.1	Diagram alir alur penelitian	24
Gambar 3.2	Gambaran umum sistem	25
Gambar 3.3	Diagram rancangan perangkat keras	26
Gambar 3.4	Koneksi SPI STM32 dengan EVAL-AD5941BATZ	27
Gambar 3.5	<i>Timing diagram</i> komunikasi SPI IC AD5941	27
Gambar 3.6	Koneksi UART STM32 dengan EVAL-AD5941BATZ	29
Gambar 3.7	Koneksi GPIO STM32 dengan EVAL-AD5941BATZ	29
Gambar 3.8	<i>Root hierarchy schematic</i> PCB Antarmuka.....	30
Gambar 3.9	<i>Hierarchical sheet</i> Voltage Regulator	30
Gambar 3.10	<i>Hierarchical sheet</i> Microcontroller	31
Gambar 3.11	<i>Hierarchical sheet</i> USB & Serial	31
Gambar 3.12	<i>Hierarchical sheet</i> Connection	32
Gambar 3.13	Pengelompokan komponen pada PCB Antarmuka	34
Gambar 3.14	Hasil <i>routing</i> PCB Antarmuka pada <i>software</i> KiCAD	34
Gambar 3.15	Hasil perancangan PCB Antarmuka	35
Gambar 3.16	PCB Antarmuka dengan STM32 dan EVAL-AD5941BATZ	35
Gambar 3.17	Diagram alir program STM32.....	37
Gambar 3.18	Diagram penerimaan data serial dari STM32 ke PC/Laptop.....	38
Gambar 3.19	Gambaran umum GUI	39
Gambar 3.20	Hasil perancangan GUI	42
Gambar 4.1	Implementasi rancangan PCB Antarmuka.....	43
Gambar 4.2	Implementasi rancangan peranti pengukuran impedansi.....	43
Gambar 4.3	Skema pengujian komunikasi UART	45
Gambar 4.4	Konfigurasi parameter pada <i>software</i> Logic	45
Gambar 4.5	Respon IDLE oleh perangkat pada <i>software</i> Logic	46
Gambar 4.6	Perintah CONNECT oleh GUI pada <i>software</i> Logic	46
Gambar 4.7	Respon SAVED_PARAM oleh perangkat pada <i>software</i> Logic ...	46
Gambar 4.8	Skema pengujian komunikasi SPI.....	47

Gambar 4.9	Hasil pengujian inisialisasi sistem INIT-1 <i>software Logic</i>	48
Gambar 4.10	Hasil pengujian keseluruhan inisialisasi pada <i>software Logic</i>	49
Gambar 4.11	Hasil pengujian perintah kalibrasi ADC pada <i>software Logic</i>	50
Gambar 4.12	Hasil pengujian akses FIFO pada <i>software Logic</i>	50
Gambar 4.13	Skema pengujian komunikasi GUI.....	51
Gambar 4.14	Pengujian refresh COM Manager	52
Gambar 4.15	Proses pengujian COM Manager pada <i>software VSCode</i>	52
Gambar 4.16	Pembaharuan nilai parameter pada <i>software VSCode</i>	53
Gambar 4.17	Hasil pengujian penyimpanan parameter pada tampilan GUI	53
Gambar 4.18	Proses pengujian pengukuran pada <i>software VSCode</i>	54
Gambar 4.19	Tampilan plot hasil pengukuran pada GUI	55
Gambar 4.20	Tampilan plot pada GUI dengan memuat file .xlsx	55
Gambar 4.21	Ilustrasi proses pengukuran impedansi	56
Gambar 4.22	Skema pengujian komunikasi SPI.....	58
Gambar 4.23	Susunan perangkat dalam pengujian	59
Gambar 4.24	Plot Nyquist pengukuran baterai Samsung INR18650-25R2	60
Gambar 4.25	Plot Bode pengukuran baterai Samsung INR18650-25R2	60
Gambar 4.26	Plot hasil pengukuran baterai Samsung INR-25R2 Baru	63
Gambar 4.27	Plot rerata pengukuran baterai Samsung INR-25R2 Baru	63
Gambar 4.28	Plot hasil pengukuran baterai Samsung INR-25R2 Bekas Baru ...	64
Gambar 4.29	Plot rerata pengukuran baterai Samsung INR-25R2 Bekas Baru..	64
Gambar 4.30	Plot hasil pengukuran baterai Samsung INR-25R2 Bekas Tua	65
Gambar 4.31	Plot rerata pengukuran baterai Samsung INR-25R2 Bekas Tua ...	65
Gambar 4.32	Perbandingan hasil pengukuran antara 3 sampel baterai	66
Gambar 4.33	Nilai R_S pengukuran baterai Samsung INR-25R2 Baru	71
Gambar 4.34	Nilai R_S pengukuran baterai Samsung INR-25R2 Bekas Baru....	71
Gambar 4.35	Nilai R_S pengukuran baterai Samsung INR-25R2 Bekas Tua....	72
Gambar 4.36	Nilai R_{CT} pengukuran baterai Samsung INR-25R2 Baru	72
Gambar 4.37	Nilai R_{CT} pengukuran baterai Samsung INR-25R2 Bekas Baru..	73
Gambar 4.38	Nilai R_{CT} pengukuran baterai Samsung INR-25R2 Bekas Tua ...	73
Gambar 4.39	Nilai C_{DL} pengukuran baterai Samsung INR-25R2 Baru	74
Gambar 4.40	Nilai C_{DL} pengukuran baterai Samsung INR-25R2 Bekas Baru ..	74
Gambar 4.41	Nilai C_{DL} pengukuran baterai Samsung INR-25R2 Bekas Tua ...	74
Gambar 4.42	Hasil Pengukuran Baterai Litium-ion Lainnya	76
Gambar 1	Baterai Samsung INR18650-25R2 objek pengukuran akurasi....	L-35
Gambar 2	Perangkat ST-Link/V2 Programming	L-36
Gambar 3	Perangkat Logic Analyzer 24MHz	L-36
Gambar 4	Kabel dan <i>holder</i> baterai pengukuran impedansi.....	L-36

DAFTAR SINGKATAN

f	= Frekuensi
ϕ	= Sudut fase
\bar{x}	= Rata-rata
σ	= Standar Deviasi
R_S	= <i>Solution Resistance</i>
R_{CT}	= <i>Charge Transfer Resistance</i>
C_{DL}	= <i>Double Layer Capacitance</i>
W	= Impedansi Warburg
Z	= Impedansi
AC	= <i>Alternating Current</i>
ADC	= <i>Analog-to-Digital Converter</i>
AFE	= <i>Analog Front-End</i>
APB	= <i>Advanced Peripheral Bus</i>
DAC	= <i>Digital-to-Analog Converter</i>
DC	= <i>Direct Current</i>
DFT	= <i>Discrete Fourier Transform</i>
DIBS	= <i>Distributed Impulse Binary Sequence</i>
ECM	= <i>Equivalent Circuit Model</i>
EIS	= <i>Electrochemical Impedance Spectroscopy</i>
FIFO	= <i>First In, First Out</i>
FPGA	= <i>Field-Programmable Gate Array</i>
GPIO	= <i>General Purpose Input/Output</i>
GUI	= <i>Graphical User Interface</i>
IC	= <i>Integrated Circuit</i>
LED	= <i>Light-emitting Diode</i>
MLBS	= <i>Maximum Length Binary Sequence</i>
PCB	= <i>Printed Circuit Board</i>
PGA	= <i>Programmable Gain Amplifier</i>
PWM	= <i>Pulse Width Modulation</i>
RSD	= <i>Relative Standard Deviation</i>
SEI	= <i>Solid Electrolyte Interface</i>
SPI	= <i>Serial Peripheral Interface</i>
TIA	= <i>Trans-impedance Amplifier</i>
TTL	= <i>Transistor-transistor Logic</i>
UART	= <i>Universal Asynchronous Receiver-Transmitter</i>
USB	= <i>Universal Serial Bus</i>

INTISARI

Electrochemical impedance spectroscopy (EIS) merupakan teknik pengukuran non-invasif yang akurat dan cocok untuk karakterisasi baterai litium-ion. Penelitian ini bertujuan untuk mengembangkan perangkat pengukuran impedansi baterai litium-ion melalui metode EIS memanfaatkan modul EVAL-AD5941BATZ dan mikrokontroler STM32F411CEU6, serta menampilkannya pada perangkat komputer dengan *graphical user interface* (GUI) yang memiliki fungsi plot Nyquist, plot Bode, dan karakterisasi impedansi baterai litium-ion. Metodologi yang digunakan adalah perancangan perangkat dan sistem, implementasi perangkat dan sistem, pengujian, serta pengambilan data untuk analisis hasil perancangan. Nilai akurasi pengukuran dinilai berdasarkan *relative standard deviation* (RSD). Hasil penelitian menunjukkan bahwa perangkat yang dikembangkan mampu melakukan pengukuran impedansi baterai litium-ion dengan RSD di bawah 1% pada sapuan frekuensi di atas 1 kHz dan RSD di bawah 3% pada sapuan frekuensi di bawah 1 kHz. Perangkat juga mampu mengirimkan data ke GUI, melakukan plot karakterisasi impedansi, dan menentukan parameter internal baterai. Hasil ini menunjukkan bahwa perangkat tersebut memiliki tingkat akurasi yang tinggi dan dapat diandalkan dalam pengukuran impedansi baterai. Kesimpulan dari penelitian ini adalah perangkat yang dikembangkan akurat, berbiaya rendah, dan mudah dioperasikan, sehingga memudahkan karakterisasi baterai litium-ion untuk proses daur ulang.

Kata kunci : Baterai litium-ion, EVAL-AD5941BATZ, *Electrochemical impedance spectroscopy*

ABSTRACT

Electrochemical impedance spectroscopy (EIS) is a precise and non-invasive characterization method for lithium-ion batteries (LIBs). This study aims to develop a LIB impedance measurement device based on the high-precision impedance and electrochemical front-end EVAL-AD5941BATZ module and the STM32F411CEU6 microcontroller. This system utilizes a wide-frequency galvanostatic method, producing a constant current source as the excitation signal, and detects the battery response voltage signal to obtain EIS data ranging from 0.1 Hz to 200 kHz. Additionally, a graphical user interface is developed for the real-time display of Nyquist plots and Bode plots from the EIS results, allowing the user to perform LIB impedance characterization. The results show that for frequency sweeps above 1 kHz, the impedance measurement RSD is below 1%, and for frequency sweeps below 1 kHz, the impedance measurement RSD is below 3%. Compared to commercial impedance analyzer devices, the developed system is accurate, low-cost, and easy to operate, facilitating lithium-ion battery characterization for the recycling process.

Keywords : Lithium-ion battery, EVAL-AD5941BATZ, Electrochemical impedance spectroscopy

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penggunaan kendaraan listrik, terutama mobil listrik, diprediksi meningkat secara eksponensial pada tahun 2023 [1]. Akibatnya, kebutuhan baterai sebagai penyimpanan energi utama kendaraan listrik melonjak tinggi. Baterai litium-ion merupakan satu diantara beragam jenis baterai yang digunakan sebagai penyimpanan energi utama pada kendaraan listrik. Baterai ini pun banyak digunakan pada perangkat elektronik seperti *smartphone*, *powerbank*, laptop, dan perangkat elektronik portabel lainnya. Litium-ion marak digunakan karena memiliki *energy density* yang tinggi, *specific energy* yang tinggi, *self-discharge rate* yang rendah, minimalnya *memory effect*, dan tajamnya penurunan biaya produksi dalam kurun beberapa tahun terakhir [2], [3]. Baterai litium-ion hadir dalam tiga basis bentuk yaitu *cylinder*, *prismatic*, dan *pouch* sehingga memberi fleksibilitas dalam penggunaannya. Aplikasinya pada kendaraan listrik mengharuskan *battery pack* terdiri dari ratusan hingga ribuan sel baterai yang disusun dalam struktur seri dan paralel [2] untuk menghasilkan tegangan, mengalirkan arus, serta menyimpan besarnya energi yang dibutuhkan pada sistem kendaraan.

Ribuan sel baterai yang ada pada *battery pack* kendaraan listrik memiliki parameter seperti kapasitas dan resistansi internal yang tidak bernilai sama karena proses produksi dan manufaktur [2], [4]. Perbedaan ini akan menyebabkan ketidakseimbangan antara satu baterai dengan baterai lain setelah melewati siklus pengisian-pengosongan secara berulang. Akibat dari perbedaan parameter yang signifikan adalah terjadinya *overcharge* dan *overdischarge* pada sel baterai dengan kapasitas terkecil sehingga timbul degradasi dan perubahan zat kimia penyusun baterai [4], [5]. Masa pakai baterai yang terbatas disertai dengan degradasi zat kimia penyusun akan berdampak besar pada performa *battery pack* kendaraan listrik [6]. Akhirnya, *battery pack* kendaraan listrik akan mencapai penghujung hidupnya dan akan dihancurkan untuk didaur ulang. Namun, tidak semua sel baterai pada *battery pack* perlu dihancurkan untuk daur ulang. Masih terdapat sel baterai yang dapat berfungsi dengan baik dan memiliki potensi besar pada pasar baterai bekas. Maka dari itu, diperlukan metode yang tepat untuk mengetahui degradasi baterai dan menilai kelayakan baterai litium-ion sebagai media penyimpanan energi.

Degradasi pada baterai litium-ion umumnya diukur dengan dua parameter, yaitu kapasitas dan resistansi internal baterai [6], [7]. Resistansi internal baterai litium-ion menjadi parameter yang penting karena digunakan untuk mengkarakterisasi kemampuan daya sebuah sel baterai [8]. Karakterisasi ini memerlukan sebuah metode untuk mengukur resistansi internal baterai secara akurat tanpa memengaruhi komponen internal baterai

tersebut. Hal tersebut tidak mudah karena baterai litium-ion bukanlah komponen *ohmic* murni [3], [8]. Sifat kapasitif yang ada pada baterai litium-ion menyebabkan pengukuran ini menjadi lebih sulit. Salah satu metode yang dapat dilakukan adalah mengukur impedansi, yaitu resistansi ditambah reaktansi, melalui *electrochemical impedance spectroscopy* (EIS).

EIS merupakan metode noninvasif yang dapat mengukur impedansi baterai litium-ion dengan akurasi tinggi. Metode ini memperoleh nilai impedansi dengan mengukur respon tegangan terhadap isyarat eksitasi yang diberikan pada rentang frekuensi yang beragam [6], [9]. Nilai tersebut dapat digunakan untuk mengkarakterisasi parameter sel baterai litium-ion seperti resistansi internal dan kapasitansi lapisan ganda [9]. Proses karakterisasi ini memerlukan sebuah peranti yang mampu menghasilkan isyarat eksitasi dan mengolah respon yang didapat. Salah satu perangkat yang dapat melakukan fungsi ini adalah Modul EVAL-AD5941BATZ. Modul ini merupakan *evaluation board* dari IC AD5941, sebuah *analog front-end* pengukuran presisi tinggi impedansi sistem elektrokimia. Namun, perangkat ini harus dihubungkan ke perangkat komputer agar dapat melakukan dan menampilkan hasil pengukuran impedansi. Mikrokontroler STM32F411CEU6 dapat digunakan untuk menjembatani Modul EVAL-AD5941BATZ dan perangkat komputer.

Maka dari itu, penelitian ini diperuntukan untuk merancang peranti EIS berbasis Modul EVAL-AD5941BATZ dengan mikrokontroler STM32F411CEU6 dan menampilkan hasil pengukuran pada perangkat komputer melalui *graphical user interface* yang memiliki fungsi tampilan plot Nyquist, plot Bode, dan karakterisasi impedansi baterai litium-ion sehingga memudahkan karakterisasi impedansi baterai litium-ion dari kendaraan listrik untuk didaur ulang.

1.2 Rumusan Masalah

Baterai litium-ion bekas kendaraan listrik memiliki potensi besar dalam penggunaan kedua (*second life*), sehingga diperlukan metode karakterisasi kelayakan yang akurat dan noninvasif. Modul EVAL-AD5941BATZ mampu melakukan spektrometri impedansi elektrokimia secara presisi dan efisien, tetapi masih memerlukan peranti mikrokontroler dan antarmuka agar dapat berfungsi dengan baik.

1.3 Tujuan Penelitian

Tujuan dari penelitian tugas akhir ini adalah membuat peranti pengukuran spektrometri impedansi elektrokimia berbasis Modul EVAL-AD5941BATZ dan mikrokontroler STM32F411CEU6, serta menampilkannya pada perangkat komputer dengan *graphical user interface* yang memiliki fungsi tampilan plot Nyquist, plot Bode, dan karakterisasi impedansi baterai litium-ion.

1.4 Batasan Penelitian

Penelitian ini akan diberi batasan-batasan berupa:

1. Baterai yang dikarakterisasi adalah baterai berbasis litium-ion.
2. Penelitian ini berfokus pada perancangan peranti EIS berbasis Modul EVAL-AD5941BATZ dan STM32F411CEU6.
3. Penelitian ini belum mempertimbangkan simulasi model baterai litium-ion.
4. Pembahasan EIS hanya mencakup pengaturan parameter IC AD5941, bentuk tampilan hasil pengukuran, dan penentuan karakteristik baterai secara sederhana.

1.5 Manfaat Penelitian

Penelitian dan penulisan tugas akhir perancangan peranti EIS diharapkan dapat memudahkan karakterisasi baterai litium-ion bekas kendaraan listrik sehingga nantinya dapat diputuskan apakah memiliki *second life* atau harus dihancurkan.

1.6 Sistematika Penulisan

Sistematika penulisan penelitian mengenai perancangan perangkat EIS dipaparkan dalam lima bab.

Bab pertama adalah perndahuluan. Bab pertama memuat latar belakang penelitian, perumusan masalah, tujuan penelitian, batasan penelitian, manfaat dari penelitian yang dilakukan, dan sistematika penulisan.

Bab kedua adalah tinjauan pustaka dan dasar teori. Bab kedua memuat tinjauan pustaka, dasar teori yang digunakan dalam merancang dan menguji peranti pengukuran impedansi baterai, Modul EVAL-AD5941BATZ, mikrokontroler STM32F411CEU6, dan metode EIS yang diusulkan.

Bab ketiga adalah metode tugas akhir. Bab ketiga memuat alur penelitian tugas akhir, gambaran umum sistem, perancangan peranti yang berisi penjelasan perancangan PCB, perancangan perangkat lunak, dan perancangan *graphical user interface*.

Bab keempat adalah hasil dan pembahasan. Bab keempat memuat tentang hasil perancangan perangkat keras, tampilan pada GUI, dan analisis dari masing-masing pengujian yang dilakukan untuk menguji peranti pengukuran dan fitur lainnya.

Bab kelima adalah kesimpulan dan saran. Bab kelima memuat tentang kesimpulan dari penelitian yang sudah dilakukan dan saran berhubungan dengan penelitian yang telah dilaksanakan dan kaitannya dengan perancangan perangkat EIS.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

EIS merupakan teknik pengukuran non-invasif yang sangat akurat dan cocok untuk karakterisasi berbagai macam bahan termasuk baterai litium-ion. Karakteristik atau parameter utama baterai litium-ion yang dapat diketahui melalui proses EIS adalah impedansi. Respons impedansi diperoleh dengan memberikan sebuah isyarat eksitasi kecil kepada sel baterai. Baterai akan memberi respons berupa keluaran sinusoidal pada frekuensi yang sama persis tetapi dengan fase dan amplitudo yang bergeser. Respons tersebut kemudian dapat diubah menjadi bentuk riil dan imajiner, kemudian ditampilkan menjadi plot yang umumnya merupakan plot Nyquist. Melalui plot Nyquist, baterai litium-ion dapat dikarakterisasi berdasarkan spektrum impedansinya yang menunjukkan zona-zona tertentu dan merepresentasikan nilai parameter di dalamnya.

Terdapat penelitian terdahulu untuk membuat peranti EIS portabel dengan biaya rendah. Kelompok peneliti dari Politecnico di Torino melakukan penelitian berdasarkan pada TeensyDuino untuk melakukan pengukuran korosi pada logam. Penelitian ini mengimplementasikan beberapa mikrokontroler yaitu Teensy 3.2, Teensy 3.5, dan Teensy 3.6. Beberapa versi dari Teensy digunakan untuk membentuk beberapa sistem yang saling mengakomodir kekurangan fitur dari antara versi. Teensy 3.2 dipilih karena satunya Teensy seri 3 yang memiliki *programmable gain amplifier* (PGA) pada setiap ADC. Terlebih, PGA tersebut dapat menerima tegangan negatif. Pada Teensy 3.5 dan Teensy 3.6, terdapat ADC yang dapat digunakan pada *single mode* atau *differential mode*. Dengan digunakannya *differential mode*, ADC memiliki performa yang lebih baik. Ketiga versi mikrokontroler ini memiliki kemampuan PWM pin digital tetapi tidak dapat digunakan untuk rentang frekuensi EIS yang lebar tanpa penambahan komponen lain. Namun, penggunaan peranti ini hanya ditujukan pada pengukuran impedansi logam sehingga perangkat keras yang diusulkan hanya memerlukan beberapa resistor. Tanpa adanya komponen lain seperti OpAmp ataupun MOSFET, pengukuran tetap mampu dilakukan dalam rentang frekuensi 0,01 Hz hingga 50 kHz dengan isyarat eksitasi yang dihasilkan langsung oleh mikrokontroler [10]. Penyederhanaan rangkaian dicapai dengan memperkirakan arus pengukuran dari tegangan jatuh pada resistor shunt. Nilai impedansi yang berbeda dapat juga disesuaikan dengan mengganti nilai resistor shunt yang digunakan. Ketidakpastian sistem yang diusulkan kurang dari 5% untuk nilai impedansi hingga $50\text{k}\Omega$ [10]. Sistem yang diusulkan pun menggunakan perangkat lunak open source sehingga dapat disesuaikan dengan berbagai kebutuhan pengukuran.

Peneliti dari University of Padova mengembangkan peranti EIS menggunakan konverter daya komersial untuk menginjeksi pulsa arus DC ke dalam baterai dan mengukur respons tegangannya. Tipe konverter daya yang digunakan adalah Synchronous Buck Converter. Arus sinusoidal yang diinjeksi ke baterai diperoleh dengan menambahkan gangguan pada duty cycle isyarat PWM [11]. Kontrol arus injeksi dilakukan dengan menambahkan rangkaian eksternal untuk menerima umpan balik sistem yaitu parameter arus induktor dan arus eksitasi. Koneksi pengukuran menggunakan metode empat kabel sehingga meminimalkan resistansi kontak pada terminal baterai. Untuk melakukan EIS dengan akurat, digunakan closed-loop current control untuk mengatur duty cycle dari isyarat PWM yang diinjeksi. Perangkat pengujian terdiri dari Evaluation Board 6494L dari STMicroelectronics yang dilengkapi dengan dua MOSFET dan drivernya. Frekuensi pensaklaran sebesar 210 kHz dihasilkan melalui board khusus kontrol PWM. Kontrol arus digital dikembangkan dalam sistem LabVIEW dengan frekuensi pencuplikan sebesar 50 kHz. Perangkat kontrol arus tersebut adalah FPGA cRio dari National Instrument berisi ADC dan DAC dengan spesifikasi frekuensi pencuplikan 100 kHz, 16-bit konversi dan nilai skala penuh $\pm 10V$. Pengukuran EIS dilakukan pada rentang frekuensi 1 Hz sampai 1200 Hz.

Tabel 2.1. Tinjauan Pustaka

Tahun	Judul Penelitian	Keterangan
2021	<i>Electrochemical Impedance Spectroscopy System Based on a Teensy Board</i>	Perancangan peranti EIS menggunakan mikrokontroler Teensy 3.2, Teensy 3.5, dan Teensy 3.6 tanpa menggunakan komponen eksternal rumit untuk mengukur proses korosi pada logam.
2021	<i>An experimental validation of the adoption of dc-dc converters for the impedance measurement in Li-Ion batteries</i>	Perancangan peranti EIS berbasis konverter daya menggunakan Synchronous Buck Converter dan FPGA cRio dari National Instrument sebagai perangkat kontrol.

Tabel 2.1 memuat rangkuman dari penelitian terdahulu mengenai peranti EIS. Penelitian [10] dari Politecnico di Torino menggunakan mikrokontroler Teensy tanpa komponen eksternal kompleks dengan terbatasnya pengukuran hanya untuk benda pasif seperti logam. Terdapat juga penelitian [11] dari University of Padova yang memanfaatkan konverter daya dengan sistem kontrol yang kompleks. Maka penelitian ini akan mengembangkan sistem berdasar penelitian terdahulu, pengendali akan diimplementasikan pada mikrokontroler STM32F411CEU6 sebagai antarmuka Modul EVAL-5941BATZ yang memiliki cip khusus untuk melakukan pengukuran impedansi sehingga pengukuran impedansi menjadi lebih akurat dan mudah.

2.2 Dasar Teori

Perancangan dan pengujian peranti EIS berbasis Modul EVAL-AD5941BATZ membutuhkan pengetahuan mendasar mengenai objek yang diukur, yaitu baterai lithium-ion, metode *electrochemical impedance spectrometry*, cip AD5941, dan mikrokontroler STM32F411CEU6. Dengan demikian, implementasi pada perancangan peranti dapat dilakukan sesuai dengan ilmu yang ada.

2.2.1 Baterai

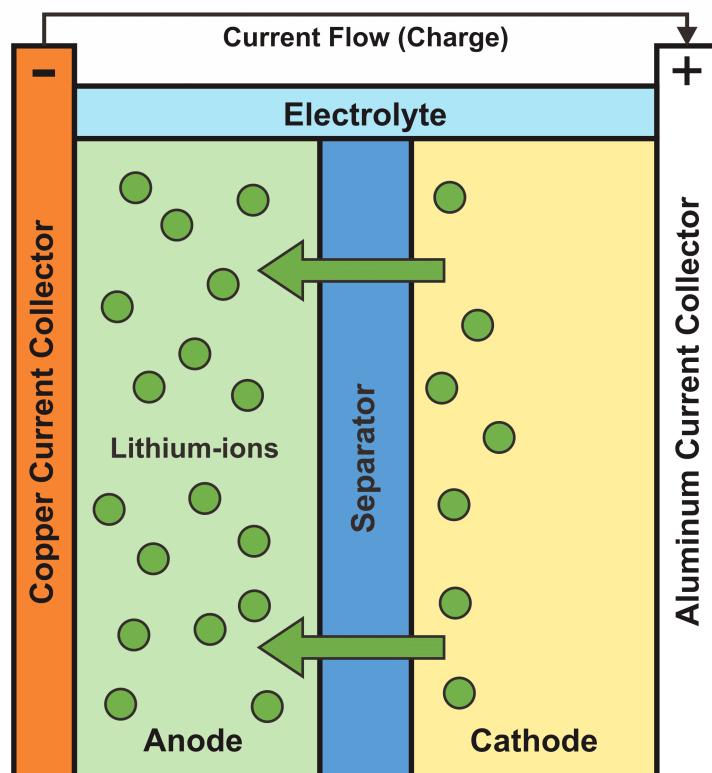
Baterai merupakan perangkat penyimpanan energi listrik yang berbentuk portabel. Secara spesifik, baterai adalah alat penyimpanan energi berbasis elektrokimia yang beroperasi dengan mengubah energi kimia menjadi energi listrik [2]. Baterai diklasifikasikan menjadi baterai primer dan baterai sekunder. Perbedaan umumnya adalah baterai primer tidak dapat diisi ulang, seperti baterai sekali pakai yang harus dibuang setelah habis, sedangkan baterai sekunder adalah baterai yang dapat diisi ulang.

Sel baterai umumnya tersusun dari lima komponen utama untuk menghasilkan reaksi kimia. Pertama, terdapat katoda, yaitu bagian "positif" sel baterai yang terdiri dari substrat berlapis bahan aktif. Kedua, terdapat anoda, yaitu bagian "negatif" sel baterai yang biasanya terbuat dari tembaga tipis dan dilapisi dengan bahan anoda aktif. Ketiga, terdapat *separator* atau "pemisah" yang mencegah katoda dan anoda bersentuhan sehingga tidak menimbulkan korsleting. Ketiga komponen tersebut kemudian dirakit untuk membentuk elektroda yang digulung atau ditumpuk. Komponen keempat adalah *enclosure*, yang umumnya berupa kaleng atau kantong sebagai wadah dari elektroda. Setelah tersusun, komponen kelima yaitu elektrolit ditambahkan ke dalamnya. Elektrolit adalah media yang memungkinkan ion-ion bergerak dari anoda ke katoda atau sebaliknya saat terjadi reaksi kimia. Inilah yang membuat sel baterai istimewa, karena baterai berfungsi sebagai tempat penyimpanan energi dan juga tempat energi tersebut dikonversi (dihasilkan) [2].

Karakteristik dan kinerja baterai yang berbeda sangat dipengaruhi oleh bahan kimia penyusunnya. Pada awal perkembangan baterai sekunder, marak digunakan bahan kimia berbasis non-lithium. Misalnya, lead acid (PbA), nickel cadmium (NiCd), nickel metal hydride (NiMH), sodium sulfur, dan sodium nickel chloride [2]. Proses elektrifikasi otomotif dan berkembangnya perangkat teknologi portabel yang masif menuntut teknologi penyimpanan energi menjadi lebih efisien dan ringkas. Kebanyakan baterai berbasis non-lithium memiliki ukuran yang cukup besar sehingga kepadatan energinya rendah, dengan tegangan yang dihasilkan hanya berkisar antara 1,2 V hingga 2,6 V [2]. Hal ini kurang sesuai dengan aplikasi otomotif dan perangkat elektronik. Oleh karena itu mulai dikembangkan sel baterai berbasis bahan kimia lain yaitu litium.

2.2.2 Baterai Lithium-ion

Baterai lithium-ion mulai diperkenalkan secara komersil pada tahun 1990 berbasis dari penelitian Dr. John Goodenough dari University of Texas [2]. Semenjak awal dikembangkan, baterai ini dengan cepat menjadi pilihan bagi sebagian besar perangkat elektronik karena memiliki kepadatan energi yang jauh lebih tinggi daripada sel baterai sejenisnya. Salah satu keunggulan lain dari lithium-ion dibandingkan baterai berbasis nikel dan timbal adalah tegangan yang dihasilkan antara 3,2 V dan 3,8 V. Profil tegangan yang lebih tinggi menjadi penting terutama dalam aplikasi kendaraan listrik. Artinya, lebih sedikit hubungan seri yang diperlukan untuk mencapai tegangan sistem yang diinginkan.



Gambar 2.1. Aliran arus ion litium saat *charging* [2]

Baterai lithium-ion bekerja seperti jenis baterai lainnya yaitu saat diberikan arus, ion dipaksa bergerak dari anoda ke katoda. Aliran energi tercipta ketika ion litium di dalam katoda dipindahkan melalui media elektrolit ke anoda, hal ini disebut pengisian/*charging*. Pengosongan/*discharging* ditunjukkan oleh ion litium yang dipindahkan melalui media elektrolit dari anoda ke katoda. Gambar 2.1 menampilkan peristiwa pengisian daya/*charging*, ion litium bergerak dari katoda melalui elektrolit ke separator dan kemudian bergerak lagi melalui elektrolit ke anoda. Peristiwa ini menciptakan aliran tegangan ke atas menuju *copper current collector* (terminal negatif) dan dalam kondisi sirkuit tertutup, menuju kembali ke kolektor arus positif (terminal positif) [2].

2.2.3 Baterai Litium-ion Berdasar Bahan Penyusun Katoda

Baterai litium-ion tersedia dengan performa dan karakteristik yang beragam. Berikut merupakan rangkuman performa baterai litium-ion secara umum berdasarkan bahan kimia penyusun katodanya, seperti nickel manganese cobalt (NMC), nickel cobalt aluminum (NCA), lithium iron phosphate (LFP), lithium titanate (LTO), lithium manganese oxide (LMO), dan lithium cobalt oxide (LCO). Rangkuman tersebut ditampilkan pada Tabel 2.2.

Tabel 2.2. Performa baterai litium-ion berdasarkan penyusun katoda [2]

Cathode Chemistry Descriptor	Lithium Iron Phosphate	Lithium Manganese Oxide	Lithium Titanate	Lithium Cobalt Oxide	Lithium Nickel Cobalt Aluminum	Lithium Nickel Manganese Cobalt
	LFP	LMO	LTO	LCO	NCA	NMC
Specific Energy (Wh/kg)	80 - 130	105 - 120	70	120 - 150	80 - 220	140 - 180
Energy Density (Wh/L)	220 - 250	250 - 265	130	250 - 450	210 - 600	325
Specific Power (W/kg)	1400 - 2400	1000	750	600	1500 - 1900	500 - 3000
Power Density (W/L)	4500	2000	1400	1200 - 3000	4000-5000	6500
Nominal Voltage (V)	3,2 - 3,3	3,8	2,2 - 2,3	3,6 - 3,8	3,6	3,6 - 3,7
Approx. Life Cycle	1000 - 2000	> 500	> 4000	> 700	> 1000	1000 - 4000
Operating Temp. Range (°C)	-20 to +60	-20 to +60	-40 to +55	-20 to +60	-20 to +60	-20 to +55

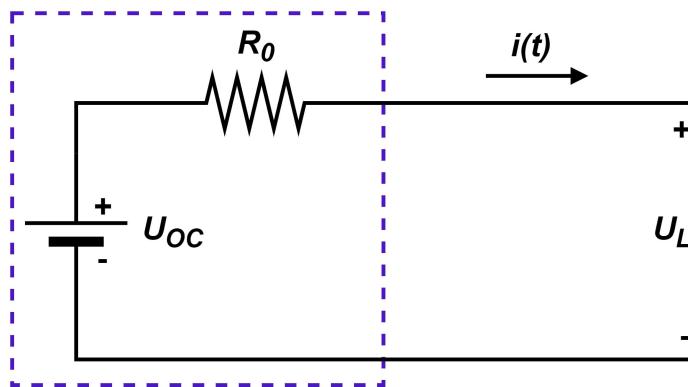
2.2.4 Model Ekuivalen Baterai Litium-ion

Baterai litium-ion merupakan sebuah komponen elektrokimia yang tersusun atas komponen kompleks. Perubahan kompleks yang terjadi ditunjukkan oleh fenomena histeresis, sifat perubahan terhadap waktu, nonlinear, dan sebagainya [12]. Oleh karena itu, diperlukan sebuah model ekuivalen yang akurat untuk bisa mengkarakterisasi baterai litium-ion.

Beragam penelitian telah dilakukan dan menghasilkan model ekuivalen dari baterai litium-ion. Berikut adalah penjelasan sederhana dari model-model tersebut.

2.2.4.1 Model Rint

Model Rint hanya memiliki dua komponen yaitu resistansi internal (R_{int}) dan sumber tegangan konstan, merupakan model yang sangat sederhana dan relatif kasar [12]. Model ini tidak hanya sederhana, tetapi juga menjadi dasar dari model baterai lainnya sehingga sering disebut sebagai model ideal baterai [12]. Model Rint ditunjukkan oleh Gambar 2.2.

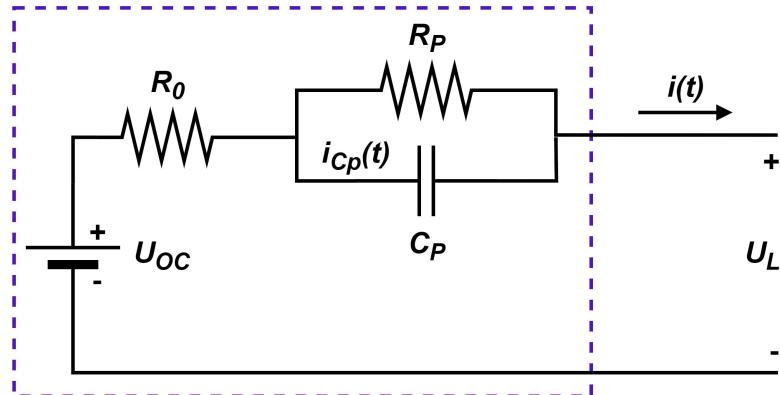


Gambar 2.2. Model Sirkuit Rint [12]

U_{OC} adalah sumber tegangan ideal yang ekuivalen sebagai tegangan hubung buka (*open circuit*); U_L adalah tegangan antara terminal positif dan negatif baterai; R_0 adalah resistansi internal ohmic baterai; $i(t)$ adalah *charge* dan *discharge* dari sirkuit.

2.2.4.2 Model Thevenin

Model Thevenin adalah model Rint yang dikembangkan dengan menambahkan faktor polarisasi internal dari baterai litium-ion. Proses *charge* dan *discharge* baterai yang nonlinear dimodelkan dengan ditambahkannya rangkaian RC paralel [12]. Oleh karena itu, model Thevenin dapat disebut sebagai sebuah model RC orde pertama. Model Thevenin ditunjukkan oleh Gambar 2.3.

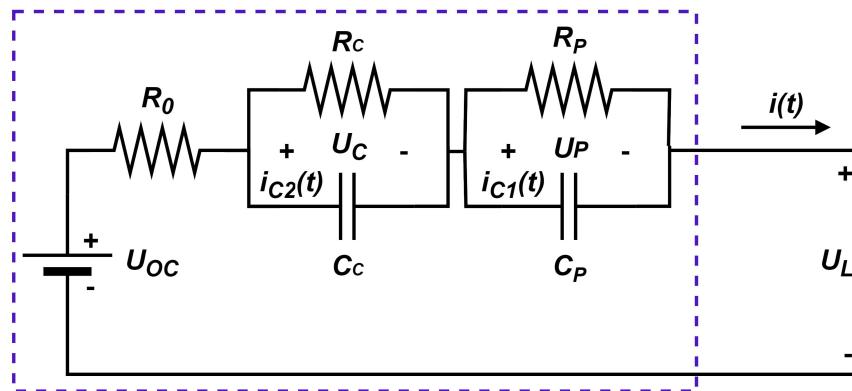


Gambar 2.3. Model Sirkuit Thevenin [12]

R_P adalah resistansi kontak nonlinear antara elektrolit dan plat elektroda; C_P adalah kapasitansi ekuivalen dari plat elektroda, sering disebut sebagai kapasitansi polarisasi. Penamaan komponen lainnya sama seperti yang ada pada Model Rint.

2.2.4.3 Model RC Orde Kedua

Pada model ini, sebuah rangkaian RC ditambahkan ke rangkaian Model Thevenin untuk memodelkan konsentrasi polarisasi di dalam baterai. Struktur model ini lebih kompleks dibandingkan dengan kedua model sebelumnya. Namun, model ini dapat menunjukkan polarisasi *ohmic* internal, polarisasi elektrokimia, dan polarisasi konsentrasi dengan lebih baik, bahkan mendekati karakter aktual baterai [12]. Model RC Orde Kedua ditunjukkan oleh Gambar 2.4.

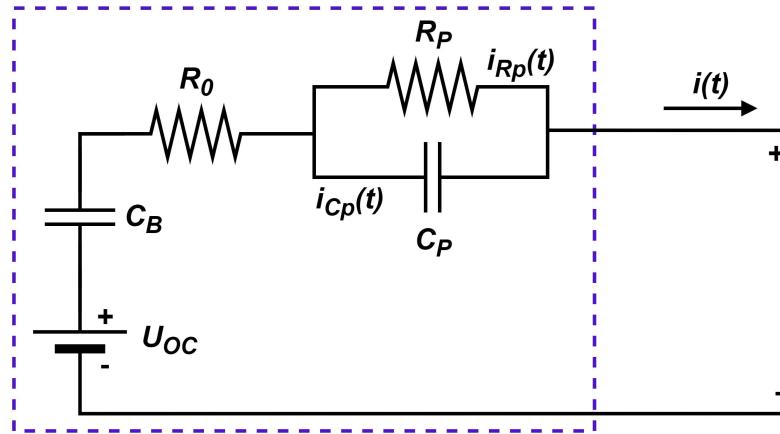


Gambar 2.4. Model Sirkuit RC Orde Kedua [12]

R_C adalah impedansi konsentrasi; C_C adalah kapasitansi konsentrasi. Penamaan komponen lainnya sama seperti yang ada pada Model Thevenin.

2.2.4.4 Model PNGV

Model ini pertama kali diajukan dalam 'PNGV Battery Test Manual' dan dikembangkan dari model Thevenin dengan menambahkan kapasitansi *bulk* C_b [12]. Ditambahkannya komponen ini bertujuan untuk meng karakterisasi perubahan tegangan hubung buka baterai yang disebabkan oleh akumulasi arus beban. Model PNGV menunjukkan perubahan integral dari tegangan hubung buka terhadap waktu dengan adanya kapasitansi C_b . Model PNGV ditunjukkan oleh Gambar 2.5.



Gambar 2.5. Model sirkuit ekuivalen PNGV [12]

2.2.5 Penuaan dan Degradasi Baterai Lithium-ion

Baterai lithium-ion tetap dapat mengalami degradasi yang signifikan meskipun memiliki siklus pakai yang tinggi dan *self-discharge rate* yang rendah. Salah satu contoh percepatan degradasi dapat ditemukan dalam aplikasi skala besar yaitu kendaraan listrik. Ribuan sel baterai yang ada pada *battery pack* kendaraan listrik memiliki parameter seperti kapasitas dan resistansi internal yang tidak bernilai sama karena proses produksi dan manufaktur [2], [4]. Perbedaan ini akan menyebabkan ketidakseimbangan antara satu baterai dengan baterai lain setelah melewati siklus pengisian-pengosongan secara berulang [8]. Akibat dari perbedaan parameter yang signifikan adalah terjadinya *overcharge* dan *overdischarge* pada sel baterai dengan kapasitas terkecil sehingga timbul degradasi dan perubahan zat kimia penyusun baterai [4], [5].

Penuaan dan degradasi sel baterai dapat diketahui dengan mengukur laju peningkatan impedansi internal sehingga memungkinkan penentuan masa hidup sel baterai [8]. Namun, hal ini tidak mudah karena impedansi internal juga mengukur pertumbuhan lapisan *solid electrolyte interface* (SEI) pada anoda [9]. Seiring dengan penggunaan sel baterai, ion-ion litium akan terperangkap dan menumpuk di lapisan SEI [2], [9]. Tumpukan ion pada lapisan SEI akan menyebabkan ion litium bebas kesulitan untuk melewati katoda sehingga nilai impedansi internal baterai meningkat.

2.2.6 *Electrochemical Impedance Spectroscopy (EIS)*

Sebagai komponen elektrokimia dengan struktur yang kompleks, baterai lithium-ion mengalami interaksi materi elektroda dalam konsentrasi elektroaktif, transfer muatan, dan transfer massa dari larutan massal ke permukaan elektroda [4], [13]. Fenomena ini dapat dikarakterisasi sebagai rangkaian listrik yang terdiri dari resistansi, kapasitansi, dan/atau elemen fase konstan yang dihubungkan secara seri dan paralel untuk membentuk rangkaian ekuivalen. EIS dapat digunakan untuk mempelajari sifat material intrinsik atau proses spesifik yang mempengaruhi konduktansi, resistansi, atau kapasitansi pada sistem elektrokimia, dalam hal ini baterai lithium-ion [13]. Karakteristik atau parameter utama yang dapat diketahui melalui proses EIS adalah impedansi. Impedansi berbeda dengan resistansi karena resistansi yang diamati pada rangkaian DC mematuhi Hukum Ohm secara langsung [13].

2.2.6.1 Isyarat Eksitasi

Isyarat eksitasi yang digunakan dalam EIS memiliki peran penting dalam memperoleh respons baterai sehingga data yang diperoleh akurat dan dapat diandalkan. Respons impedansi diperoleh dengan memberikan sebuah isyarat eksitasi kecil kepada sel baterai. Kemudian, baterai akan memberi respons yang bersifat pseudo-linear; masukan sinusoidal menghasilkan keluaran sinusoidal pada frekuensi yang sama, tetapi dengan fase dan amplitudo yang bergeser.

EIS merupakan metode yang noninvasif karena tidak memerlukan penetrasi fisik ke dalam bahan atau sistem yang diteliti. Isyarat eksitasi yang dieberikan kepada baterai memiliki nilai arus yang sangat kecil, dalam rentang mili Ampere, sehingga tidak mengubah dan merusak komponen internal baterai. EIS juga efektif dalam mendekripsi perubahan awal pada sifat listrik bahan, sebelum terjadi kerusakan fisik.

Dalam penelitian ini, IC AD5941 dapat membangkitkan isyarat eksitasi AC dengan dua pengaturan: sapuan frekuensi linear dan logaritmik. Selain isyarat eksitasi tersebut, terdapat beberapa opsi yang dapat digunakan dalam EIS, seperti MultiSine, Distributed Impulse Binary Sequence (DIBS), dan Maximum Length Binary Sequence (MLBS) [14]. Namun, sapuan frekuensi linear dan logaritmik menunjukkan efisiensi energi tertinggi, yaitu lebih dari 96% [14]. Efisiensi yang tinggi ini menjadikannya pilihan yang lebih baik dibandingkan dengan metode lain yang memiliki tingkat efisiensi lebih rendah dan rentan terhadap artefak sinyal serta noise kuantisasi.

Sapuan frekuensi logaritmik, yang karakteristiknya mirip dengan sapuan eksponensial, memiliki beberapa keunggulan dibandingkan metode lainnya. Sapuan ini mampu mencakup rentang frekuensi yang lebih luas dengan lebih banyak titik data pada frekuensi rendah dan tinggi, yang meningkatkan resolusi frekuensi secara keseluruhan [14]. Selain

itu, distribusi energi yang lebih merata di seluruh rentang frekuensi tertentu mengurangi kemungkinan tumpang tindih spektrum yang dapat mengaburkan informasi penting dalam respons sistem elektrokimia [14]. Efisiensi energi dari sweep logaritmik yang tinggi menjadikannya sangat efektif untuk aplikasi yang memerlukan analisis impedansi yang mendalam [14].

Sebaliknya, metode seperti multisine dan DIBS, meskipun memiliki distribusi frekuensi yang terencana untuk mengurangi tumpang tindih spektrum dan noise kuantisasi, menunjukkan tingkat efisiensi yang lebih rendah. Isyarat eksitasi multisine memiliki efisiensi energi hanya 54% dan DIBS 9% karena adanya kebocoran spektral dan tambahan harmonik yang mengganggu isyarat [14]. Metode MLBS juga menghadapi masalah serupa, dengan efisiensi energi yang lebih rendah dan spektrum yang dipengaruhi oleh artefak sinyal dan noise [14].

Dengan mempertimbangkan semua faktor ini, pemilihan sapuan frekuensi logaritmik untuk isyarat eksitasi EIS menjadi pilihan yang optimal. Sapuan logaritmik tidak hanya meningkatkan kualitas data yang diperoleh, tetapi juga mendukung interpretasi data yang lebih akurat sesuai dengan model yang digunakan.

2.2.6.2 Pengolahan Hasil Pengukuran EIS

Pengolahan hasil pengukuran respons baterai terhadap isyarat eksitasi pada EIS melibatkan beberapa langkah penting untuk memperoleh nilai impedansi (Z). EIS dapat diterapkan dalam mode galvanostatik atau potensiostatik. Namun, dalam penelitian ini, mode galvanostatik digunakan untuk pengukuran EIS karena lebih mudah mengontrol beban yang diterapkan pada baterai [15]. Pada pengukuran mode galvanostatik, isyarat eksitasi arus bolak-balik diterapkan ke baterai dan respons yang dihasilkan, dalam hal ini tegangan, diukur [15]. Isyarat eksitasi dan respon yang diberikan oleh baterai dinyatakan secara matematis oleh Persamaan (2-1) dan (2-2) [15].

$$\Delta I = I_{max} \cdot \sin(2\pi ft) \quad (2-1)$$

$$\Delta V = V_{max} \cdot \sin(2\pi ft + \phi) \quad (2-2)$$

Isyarat eksitasi yang diberikan pada baterai memberi respons pergeseran fase yang dievaluasi sebagai jumlah fungsi sinusoidal [15]. Langkah ini kemudian diulang untuk setiap titik dalam sapuan frekuensi. Impedansi kemudian dapat dihitung menggunakan generalisasi kompleks dari hubungan Hukum Ohm [15]. Penentuan impedansi kompleks dinyatakan dalam Persamaan (2-3) [15].

$$Z = \frac{V_{max} \cdot \sin(2\pi ft + \phi)}{I_{max} \cdot \sin(2\pi ft)} \quad (2-3)$$

Di mana V_{max} dan I_{max} adalah nilai puncak dari isyarat tegangan dan isyarat arus, f adalah frekuensi sudut, dan ϕ adalah pergeseran fase. Kemudian, bagian riil (R) dan (X) imajiner dari impedansi ini dapat dipisahkan melalui Persamaan (2-4), (2-6), dan (2-5) [15].

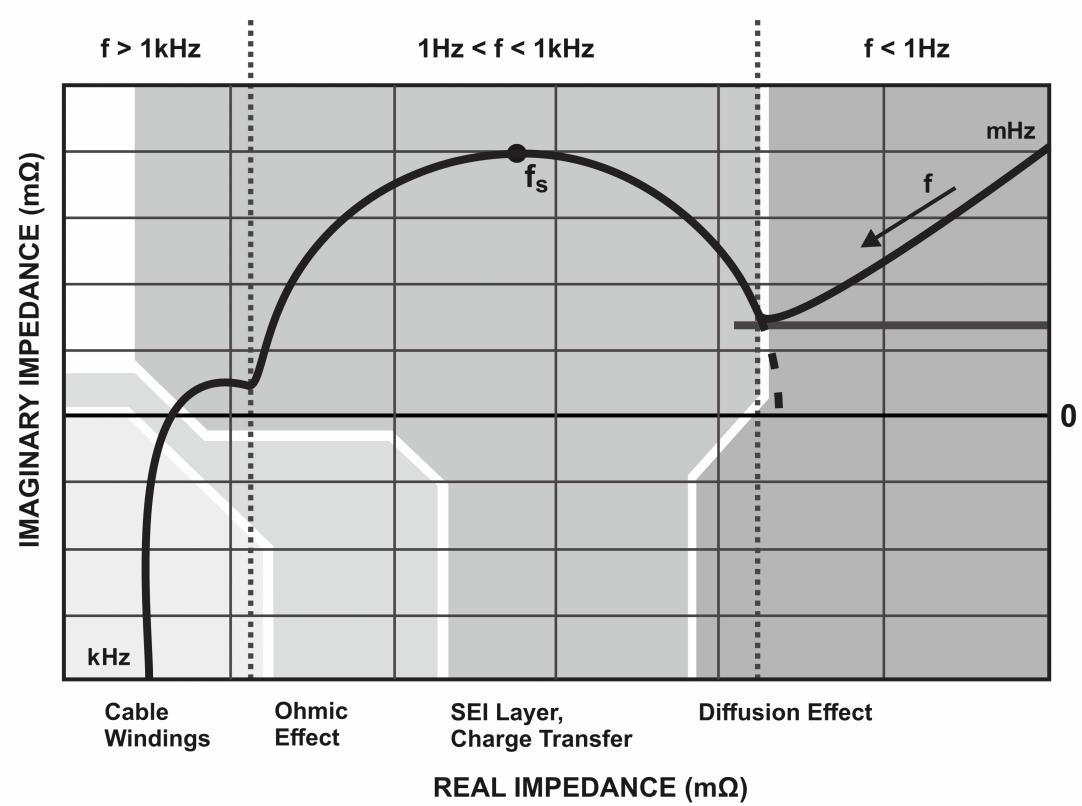
$$R = \frac{V_{max}}{I_{max}} \cos(\phi) \quad (2-4)$$

$$X = \frac{V_{max}}{I_{max}} \sin(\phi) \quad (2-5)$$

$$|Z| = \sqrt{R^2 + X^2} \quad (2-6)$$

2.2.7 Karakterisasi Hasil EIS

Informasi hasil EIS yang telah diproses menghasilkan nilai impedansi dengan komponen riil dan imajiner. Hasil ini sering ditampilkan sebagai plot Nyquist, tetapi juga dapat ditampilkan sebagai plot Bode. Umumnya plot Bode digunakan untuk mengkarakterisasi komponen kapasitif, sedangkan plot Nyquist digunakan untuk mengkarakterisasi komponen resistif baterai [13]. Kurva pada plot Nyquist menunjukkan daerah yang berhubungan dengan berbagai proses kimia dan fisik yang terjadi di dalam baterai [13]. Daerah tersebut ditampilkan pada Gambar 2.6.



Gambar 2.6. Plot nyquist hasil EIS baterai lithium-ion [9]

Proses fisik dan kimia pada baterai dimodelkan menggunakan resistor dan kapasitor seperti yang telah dijelaskan sebelumnya pada model ekuivalen baterai litium-ion. Namun, pada proses EIS terdapat komponen yang disebut resistansi Warburg yang diwakili oleh huruf W [9], [13]. Model-model ini didasarkan pada data yang dikumpulkan dari baterai yang diuji. Setelah plot Nyquist baterai dikarakterisasi, model rangkaian ekuivalen tersebut dapat disesuaikan untuk mendapatkan parameter internal baterai. Terdapat empat parameter umum yang mewakili komponen kimia baterai yaitu:

1. Electrolytic (Ohmic) Resistance— R_S

R_S berhubungan dengan resistansi elektrolit dalam baterai, parameter ini juga dipengaruhi oleh elektroda dan panjang kawat yang digunakan saat melakukan pengujian, meningkat seiring bertambahnya usia baterai, dan dominan pada frekuensi lebih dari 1 kHz [9].

2. Double Layer Capacitance— C_{DL}

C_{DL} muncul di antara elektroda dan elektrolit, terdiri dari dua lapisan paralel dengan muatan berlawanan yang mencakup elektroda, dan dominan pada rentang frekuensi 1 Hz hingga 1 kHz [9].

3. Charge Transfer Resistance— R_{CT}

R_{CT} muncul akibat perpindahan elektron dari zat padat (elektroda) ke zat cair (elektrolit), berubah sesuai suhu dan pengisian daya baterai, dominan pada rentang frekuensi 1 Hz hingga 1 kHz [9].

4. Warburg (Diffusion) Resistance— W

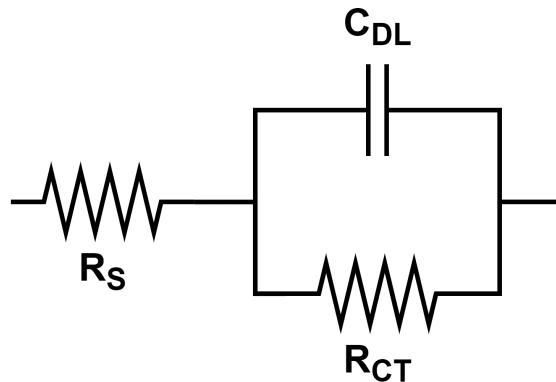
W adalah representasi dari resistansi terhadap perpindahan massa, yaitu kontrol difusi, biasanya menunjukkan pergeseran fase 45° , dominan pada frekuensi kurang dari 1 Hz [9].

Mengembangkan *equivalent circuit model* (ECM) pada EIS memerlukan eksperimen dengan berbagai model rangkaian hingga model tersebut cocok dengan plot Nyquist yang diukur. Berikut adalah penjelasan mengenai penyusunan ECM untuk membantu karakterisasi baterai litium-ion selama proses EIS.

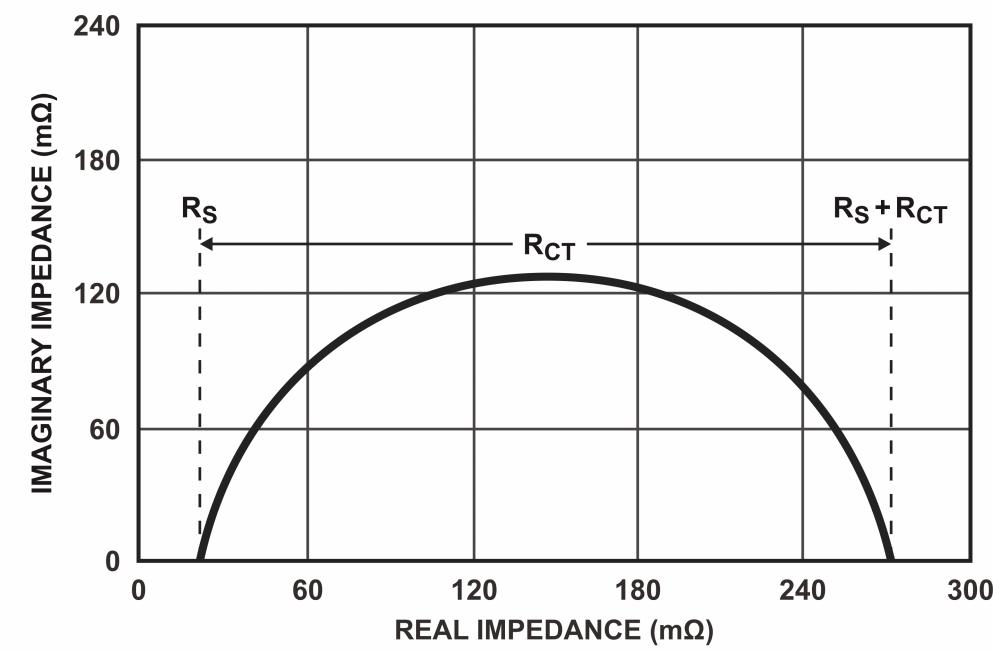
2.2.7.1 Model Sirkuit Randel—*Charge Transfer Effect*

Sirkuit Randel adalah ECM yang paling umum. Rangkaian Randel terdiri dari resistansi elektrolit (R_S), kapasitor lapisan ganda (C_{DL}), dan resistansi transfer muatan R_{CT} . Kapasitansi lapisan ganda sejajar dengan resistansi transfer muatan, yang menciptakan bentuk simulasi setengah lingkaran. Selain menjadi model dasar yang berguna, rangkaian Randel yang disederhanakan adalah titik awal untuk model lain yang lebih

kompleks. Model ini ditunjukkan oleh Gambar 2.7 dan menyerupai model Thevenin pada bagian sebelumnya.



Gambar 2.7. Model Sirkuit Randel [9]

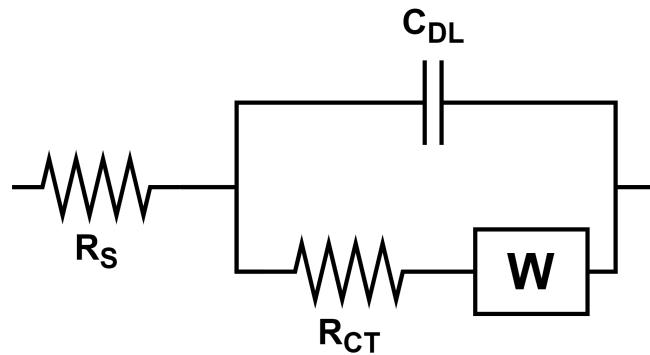


Gambar 2.8. Plot Nyquist Sirkuit Randel sederhana [9]

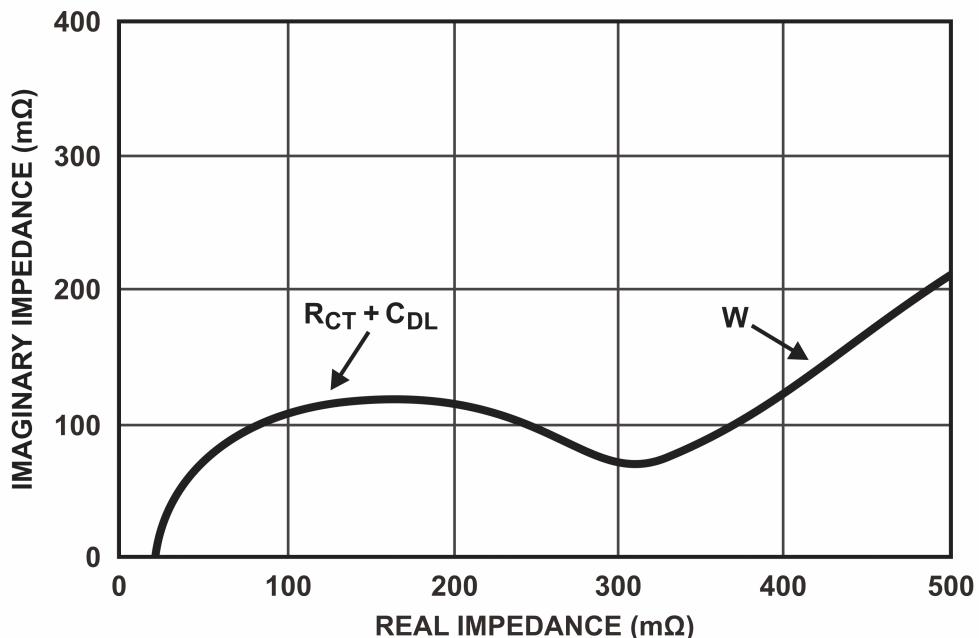
Plot Nyquist untuk sirkuit Randel yang disederhanakan selalu berbentuk setengah lingkaran. Resistansi elektrolit (R_S) ditentukan dengan membaca nilai sumbu nyata pada titik potong frekuensi tinggi karakterisasi baterai, yaitu di mana garis melintasi sumbu x di sisi kiri plot, yang merupakan zona frekuensi tinggi [13]. Pada Gambar 2.8, resistansi elektrolit (R_S) merupakan titik potong di dekat titik asal plot Nyquist, yaitu 30 mΩ. Nilai sumbu nyata pada titik potong lainnya (frekuensi rendah) adalah jumlah dari resistansi perpindahan muatan (R_{CT}) dan resistansi elektrolit (dalam hal ini 270 mΩ). Oleh karena itu, diameter setengah lingkaran sama dengan resistansi transfer muatan (R_{CT}).

2.2.7.2 Model Sirkuit Warburg—*Diffusion Effect*

Saat resistansi Warburg ditambahkan pada ECM, komponen W dipasang secara seri dengan R_{CT} . Penambahan resistansi Warburg ditunjukkan dengan garis 45° yang terlihat pada wilayah frekuensi rendah pada plot Nyquist [9], [13], ditunjukkan pada Gambar 2.10. Model sirkuit Warburg ditunjukkan oleh Gambar 2.9.



Gambar 2.9. Model Sirkuit Warburg [9]

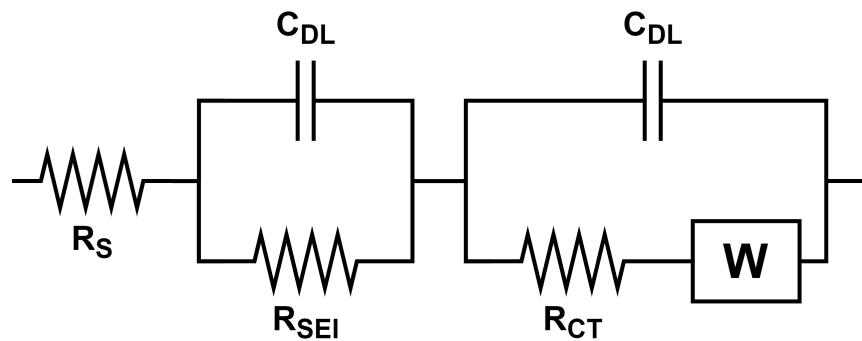


Gambar 2.10. Plot Nyquist Sirkuit Warburg dengan efek difusi [9]

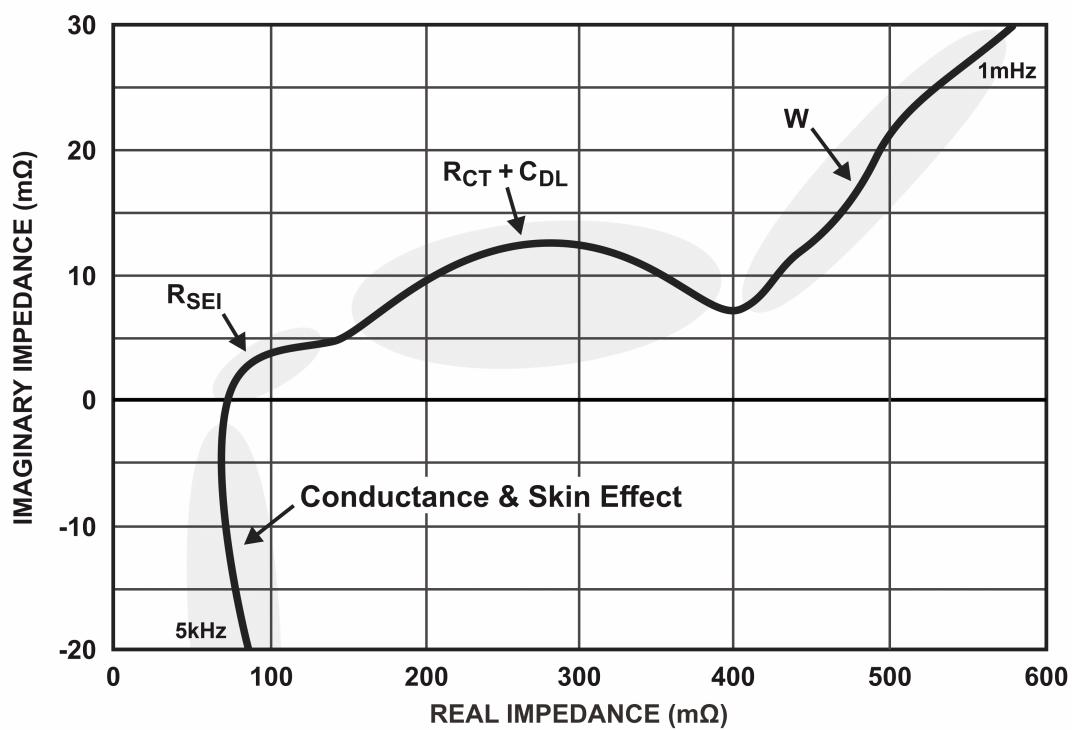
2.2.7.3 Gabungan Model Sirkuit Randel and Warburg

Baterai litium-ion yang sudah berada di penghujung masa pakainya sering menggambarkan bentuk dua setengah lingkaran. Setengah lingkaran pertama berhubungan dengan *solid electrolyte interface* (SEI) yang disebabkan oleh dekomposisi elektrokimia

elektrolit yang ireversibel [9]. Dalam kasus baterai litium-ion, SEI terbentuk di elektroda negatif seiring bertambahnya usia baterai. Produk penguraian ini membentuk lapisan padat pada permukaan elektroda. Setelah lapisan SEI awal terbentuk, ketidakmampuan molekul elektrolit untuk melakukan perjalanan melalui SEI ke permukaan bahan aktif, di mana mereka dapat bereaksi dengan ion litium dan elektron, menekan pertumbuhan SEI lebih lanjut [9], [13]. Resistor R_{SEI} memodelkan resistansi pada lapisan SEI, model ini ditunjukkan oleh Gambar 2.11. Sedangkan Gambar 2.12 menunjukkan karakteristik plot Nyquist yang lebih detail dari model ini.



Gambar 2.11. Model Sirkuit Warburg [9]

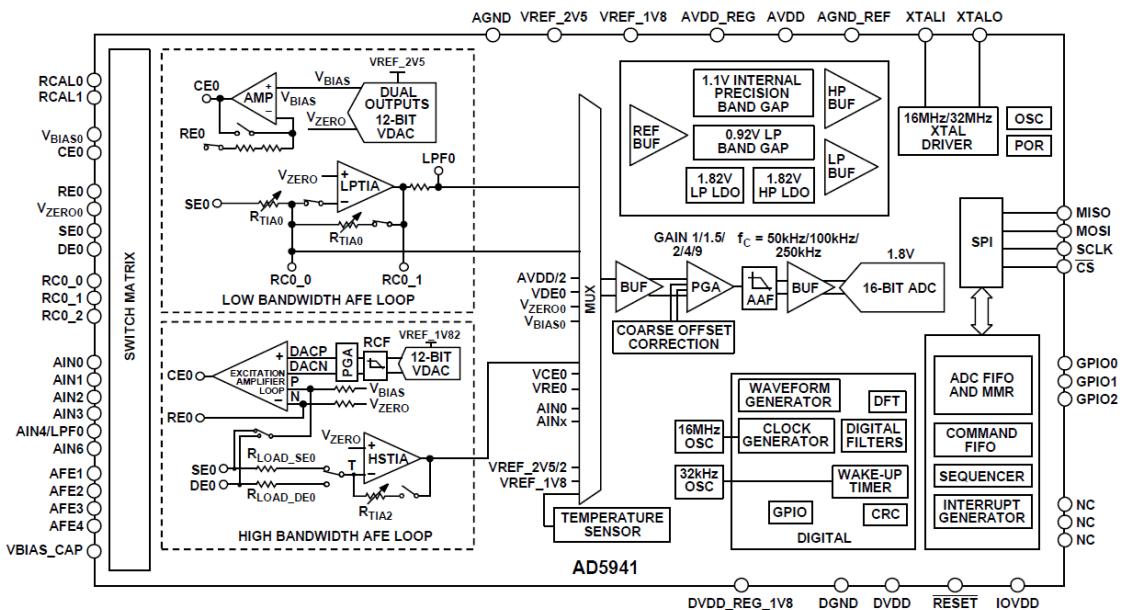


Gambar 2.12. Plot Nyquist Model Sirkuit Randel yang dimodifikasi [9]

2.2.8 IC Analog Front-End AD5941

AD5941 merupakan sebuah *integrated circuit* (IC) *analog front-end* (AFE) presisi tinggi berdaya rendah yang dirancang untuk aplikasi portabel dalam pengukuran berbasis metode elektrokimia, seperti amperometri, voltametri, atau pengukuran impedansi [16]. IC AD5941 memiliki dua *loop* eksitasi presisi tinggi dan satu buah saluran pengukuran umum yang memungkinkan kemampuan pengukuran secara luas [16]. *Loop* eksitasi pertama terdiri dari rangkaian keluaran ganda berdaya sangat rendah, *digital-to-analog converter* (DAC), dan potensiostat berdaya rendah dan berderau rendah. Satu keluaran DAC mengontrol masukan *non-inverting* dari potensiostat, dan keluaran lainnya mengontrol masukan *non-inverting* dari *transimpedance amplifier* (TIA) [16]. *Loop* eksitasi berdaya rendah ini mampu menghasilkan sinyal dari DC hingga 200 Hz. *Loop* eksitasi kedua terdiri dari 12-bit DAC, disebut sebagai DAC berkecepatan tinggi yang dapat menghasilkan isyarat eksitasi berfrekuensi tinggi hingga 200 kHz [16].

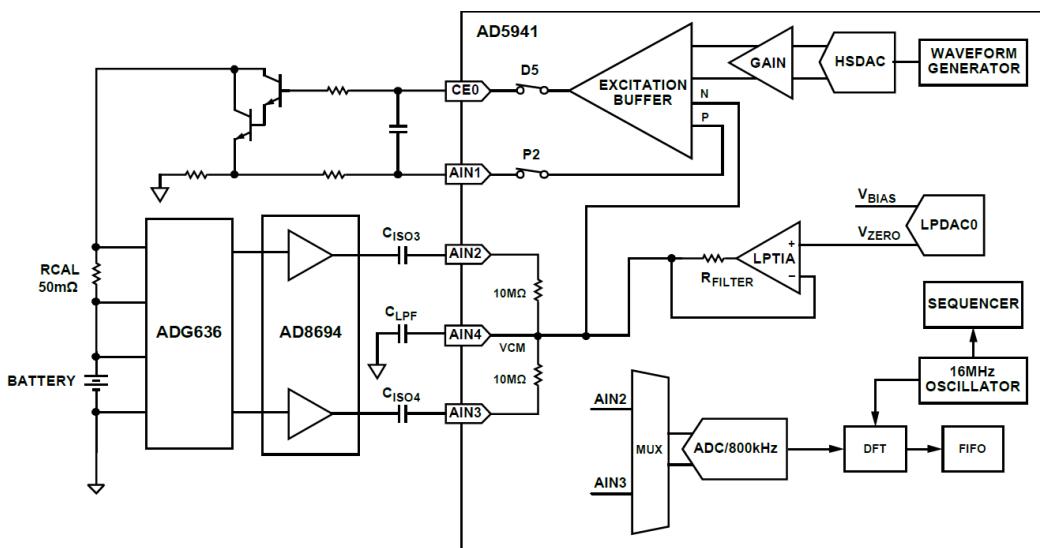
IC AD5941 dapat dikontrol dengan *direct register writes* melalui Serial Peripheral Interface (SPI), atau, sebagai alternatif, menggunakan sequencer yang dapat diprogram sebelumnya sehingga menyediakan kontrol otonom terhadap AFE. Terdapat *static random access memory* (SRAM) sebesar 6 kB yang dipartisi untuk *deep data first in, first out* (FIFO) dan *command FIFO* [16]. Cip ini juga memiliki sejumlah *general purpose input/output* (GPIO) yang dapat dikontrol menggunakan sequencer AFE. Sequencer AFE memungkinkan kontrol siklus yang akurat pada beberapa perangkat sensor eksternal. Gambar 2.13 menunjukkan diagram blok fungsional dari IC AD5941.



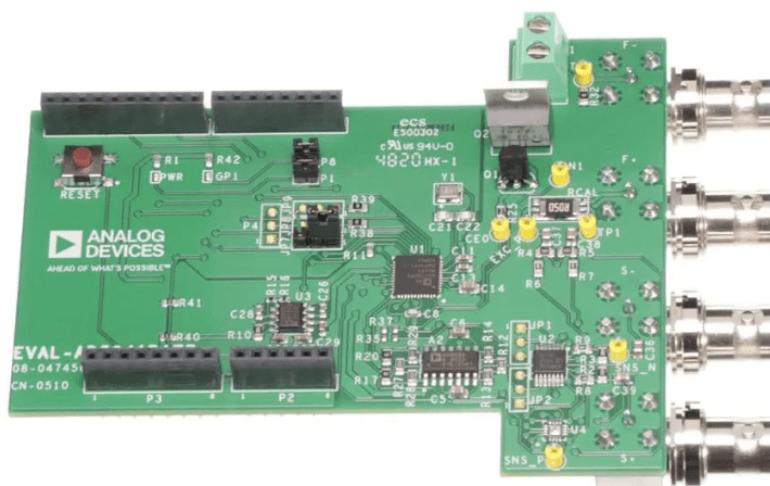
Gambar 2.13. Diagram blok fungsional IC AD5941 [14]

2.2.9 Modul EVAL-AD5941BATZ

Pengukuran impedansi dengan EIS menggunakan IC AD5941 memerlukan komponen tambahan untuk menghasilkan isyarat eksitasi dan mengukur respon dari baterai. Komponen tersebut ialah transistor, *low noise* OpAmp, dan *switch* CMOS. EIS pada baterai memerlukan eksitasi arus, bukan tegangan, dan nilai impedansi yang terbaca sangatlah kecil, dalam kisaran miliohm [9]. Modul EVAL-AD5941BATZ memuat seluruh rangkaian dan komponen yang dibutuhkan untuk injeksi arus ke baterai dan memungkinkan kalibrasi serta pengukuran impedansi pada baterai. Gambar 2.14 menunjukkan sirkuit yang disederhanakan dari Modul EVAL-AD5941BATZ. Gambar 2.15 menunjukkan perangkat fisik Modul EVAL-5941BATZ.



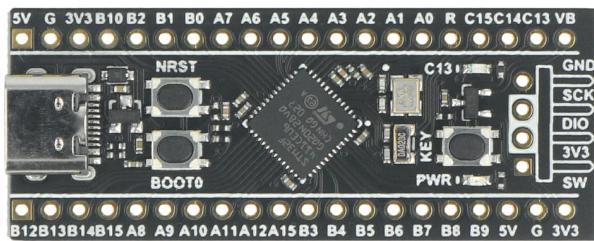
Gambar 2.14. Sirkuit yang disederhanakan dari Modul EVAL-5941BATZ [9]



Gambar 2.15. Modul EVAL-AD5941BATZ [9]

2.2.10 STM32F411CEU6

Mikrokontroler yang digunakan pada penelitian ini adalah STM32F411CEU6 dengan arsitektur ARM®Cortex®-M4. Mikrokontroler ini berbentuk *development board* dari pihak ketiga yang berbasis CPU seri STM32F411CE dari STMicroelectronics. CPU STM32F411CEU6 memiliki *high-speed embedded memories* (sampai 512 kBytes *flash memory* dan 128 kBytes SRAM) dan I/O dengan rentang yang luas [17]. Mikrokontroler ini juga dilengkapi dengan periferal yang terkoneksi dengan 32-bit multi-AHB bus matrix. Mikrokontroler ini memiliki 12-bit ADC, RTC berdaya rendah, enam general-purpose 16-bit timers dengan sebuah PWM timer, dua general-purpose 32-bit timers [17]. Gambar 2.16 menunjukkan perangkat fisik STM32F411CEU6 yang dikenal dengan nama "Black Pill".



Gambar 2.16. STM32F411CEU6 "Black Pill" [17]

Periferal pada mikrokontroller STM32F411CEU6 yang akan digunakan pada penelitian ini hanyalah *serial peripheral interface* (SPI), *universal asynchronous receiver/transmitter* (UART), dan juga *general port input/output* (GPIO). Berikut adalah penjelasan penggunaan periferal yang akan digunakan dalam penelitian ini.

2.2.10.1 *Serial Peripheral Interface* (SPI)

SPI merupakan antarmuka komunikasi komponen elektronik digital yang paling banyak digunakan. SPI merupakan komunikasi sinkron dan *full duplex* dengan tipe antarmuka berbasis *main-subnode* [18]. Data yang dikirimkan oleh *main* ataupun *subnode* disinkronkan pada tepi naik atau turun dari sinyal *clock* sehingga *main* ataupun *subnode* dapat mengirimkan data secara bersamaan [18]. Penelitian ini menggunakan periferal SPI untuk mengakses IC *analog front-end* AD5941 yang ada pada papan Modul EVAL-AD5941BATZ. Hal ini disebabkan oleh penggunaan antarmuka SPI pada IC AD5941 untuk mengakses register di dalamnya.

2.2.10.2 *Universal Asynchronous Receiver/Transmitter* (UART)

Periferal UART digunakan untuk mengirim data dari mikrokontroler ke PC atau laptop untuk keperluan *debugging* serta pencatatan data melalui perangkat lunak GUI

yang telah dibuat. Proses ini memungkinkan pemantauan dan analisis data secara *real-time*, yang sangat penting dalam pengembangan dan pengujian sistem berbasis mikrokontroler. Dengan menggunakan UART, komunikasi antara mikrokontroler dan perangkat eksternal menjadi lebih mudah dan efisien, memastikan data dapat ditransfer dengan cepat dan andal.

2.2.10.3 General Purpose Input/Output (GPIO)

Periferal GPIO digunakan untuk memberikan sinyal keluaran dan menerima sinyal masukan, seperti sinyal Reset dan Interrupt dari cip AD594, serta untuk berbagai fungsi lain yang memerlukan pembacaan atau penulisan nilai digital. Fungsi GPIO pada mikrokontroler memungkinkan kontrol yang fleksibel terhadap perangkat keras eksternal, memungkinkan mikrokontroler untuk berinteraksi dengan berbagai komponen dalam sistem secara efektif

BAB III

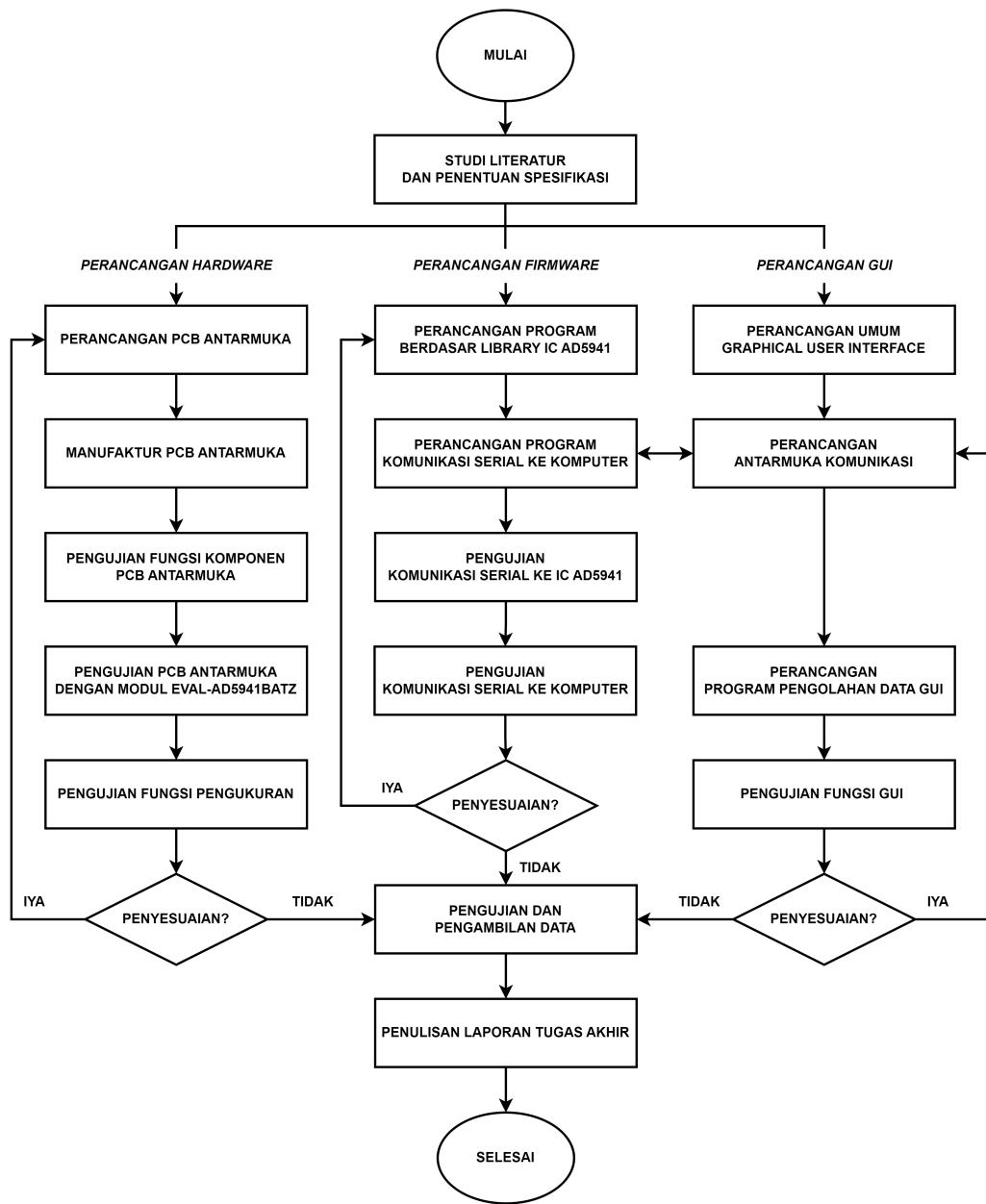
METODE PENELITIAN

3.1 Alat dan Bahan Penelitian

Alat dan bahan merupakan sarana dan prasarana penunjang proses penelitian dan pembuatan tugas akhir. Alat dan bahan ini berupa perangkat keras dan perangkat lunak yang digunakan selama proses penelitian tugas akhir.

1. Modul EVAL-AD5941BATZ, digunakan sebagai AFE pengukuran impedansi.
2. *Board* STM32F411CEU6 "Black Pill", digunakan sebagai mikrokontroler.
3. PCB Antarmuka, sebagai perangkat penghubung Modul EVAL-AD5941BATZ, STM32F411CEU6, dan komputer.
4. Modul CH340G, digunakan untuk menjembatani komunikasi serial UART antara komputer dan peranti pengukuran.
5. Modul Logic Analyzer, digunakan untuk membaca data komunikasi SPI dan UART antara STM32F411CEU6 dan IC AD5941 selama pengujian.
6. Multimeter Sanwa CD771, digunakan untuk mengukur tegangan dan koneksi pada titik tertentu selama proses pengujian PCB antarmuka.
7. Perangkat lunak KiCAD 7.0, digunakan untuk mendesain PCB Antarmuka.
8. Perangkat lunak Keil uVision 5, digunakan sebagai code editor dan compiler firmware mikrokontroler.
9. Perangkat lunak Visual Studio Code dengan ekstensi Python, digunakan dalam pengembangan GUI.
10. Perangkat STM32 ST-Link V2, digunakan untuk *programming* dan *verifying* firmware mikrokontroler.
11. Perangkat lunak Hterm, digunakan untuk menampilkan data selama pengujian komunikasi serial antara perangkat keras dan perangkat komputer.
12. Perangkat lunak Logic 2.4.14, digunakan untuk menampilkan data pengujian komunikasi SPI dan UART antara STM32F411CEU6 dan IC AD5941.
13. Perangkat lunak Fusion 360, digunakan untuk membuat model 3-dimensi dari Modul EVAL-AD5941BATZ
14. Baterai Samsung INR18650-25R2, digunakan sebagai objek pengukuran.

3.2 Alur Penelitian Tugas Akhir

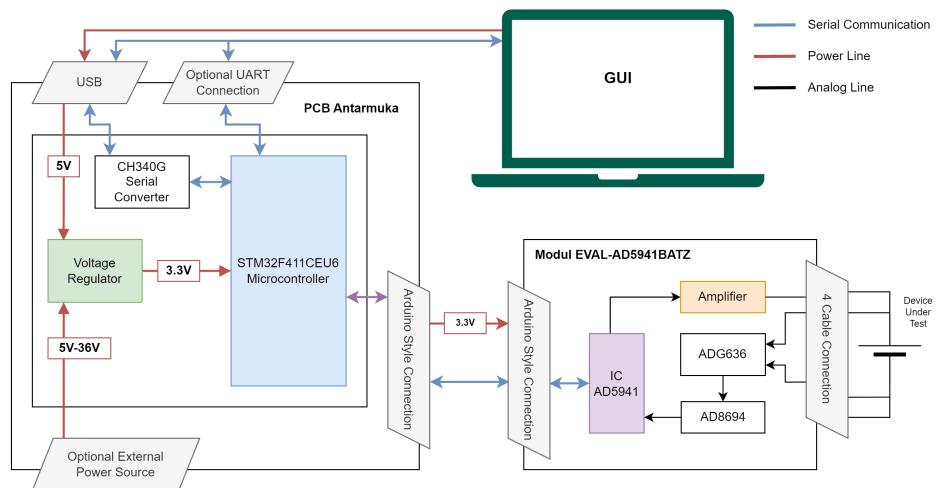


Gambar 3.1. Diagram alir alur penelitian

Penelitian ini terdiri dari empat tahapan: studi literatur, perancangan dan implementasi, pengujian, serta pengambilan dan analisis hasil. Alur penelitian ini dijelaskan oleh Gambar 3.1. Studi literatur bertujuan untuk memahami metode EIS dan perangkat yang digunakan sehingga perancangan dan implementasi dapat sesuai dengan referensi literatur. Perancangan dan implementasi mencakup perangkat keras, perangkat lunak, dan GUI, yang dilakukan secara simultan. Pengujian melibatkan pengukuran impedansi baterai litium-ion dan membandingkannya dengan data pabrik. Setelah pengujian, data dikumpulkan untuk memastikan program berjalan dengan baik dan pengukuran akurat.

3.3 Gambaran Umum Sistem

Secara garis besar, peranti pengukuran impedansi baterai litium-ion menggunakan metode EIS terdiri dari PCB antarmuka, modul EVAL-AD5941BATZ, dan perangkat komputer dengan GUI. Gambaran umum sistem yang dirancang dalam penelitian ini dapat dilihat pada Gambar 3.2.



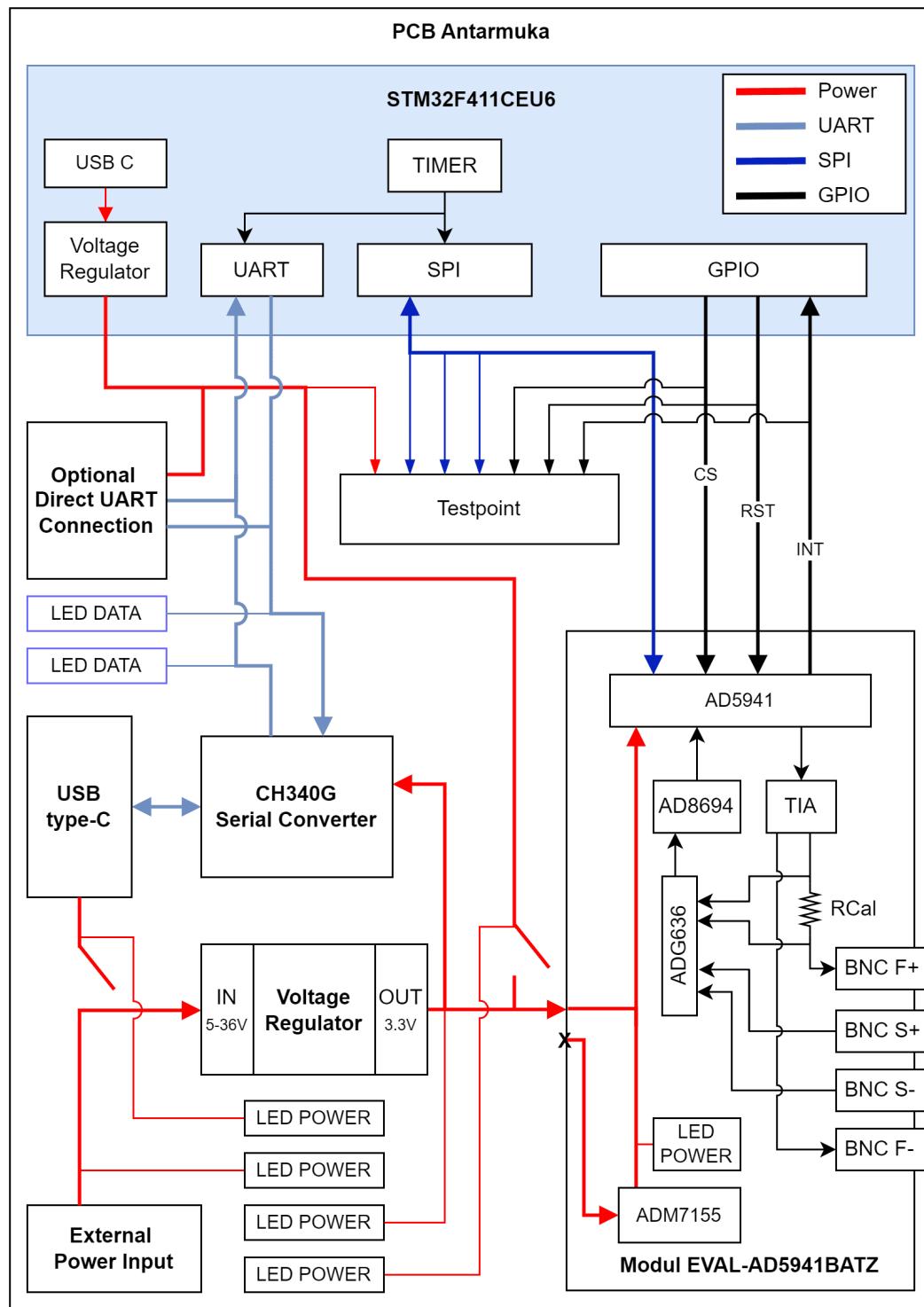
Gambar 3.2. Gambaran umum sistem

PCB antarmuka mengintegrasikan semua komponen yang diperlukan untuk menjadi jembatan antara modul EVAL-AD5941BATZ dan perangkat komputer. PCB antarmuka dapat terhubung ke komputer melalui koneksi USB tipe C yang menyediakan jalur data dan suplai tegangan 5V. Selain USB, tersedia opsi untuk koneksi daya eksternal serta koneksi UART eksternal untuk memudahkan troubleshooting. Tegangan suplai dari sumber eksternal atau USB akan dikonversi menjadi tegangan 3,3V yang diperlukan untuk mensuplai mikrokontroler dan modul EVAL-AD5941BATZ. PCB antarmuka terhubung langsung ke modul EVAL-AD5941BATZ melalui koneksi berbasis Arduino Uno, yang mencakup 4 jalur SPI, suplai daya 3,3V, dan GPIO tambahan.

Komunikasi dan ekstrasi data dari modul EVAL-AD5941BATZ dilakukan oleh mikrokontroler STM32F411CEU6. Mikrokontroler ini berfungsi untuk menerima perintah pengukuran dari komputer melalui GUI. Perintah tersebut kemudian diteruskan ke modul EVAL-AD5941BATZ untuk melakukan pengukuran pesistor kalibrasi dan pengukuran impedansi baterai. Selama proses pengukuran, modul EVAL-AD5941BATZ mengirimkan data dari register sesuai dengan permintaan mikrokontroler. Data yang dikumpulkan selama pengukuran dikirimkan secara real-time ke komputer, di mana data tersebut kemudian diolah dan ditampilkan dalam GUI. Hal ini memudahkan proses karakterisasi baterai.

3.4 Perancangan Perangkat Keras

Perangkat keras, yaitu PCB antarmuka, berfungsi sebagai penghubung antara Modul EVAL-AD5941BATZ, mikrokontroler STM32F411CEU6, dan PC/Laptop. Gambar 3.3 menampilkan diagram rancangan perangkat keras penelitian ini.

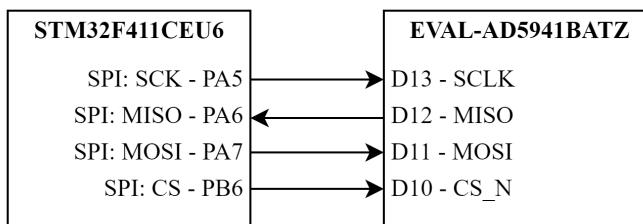


Gambar 3.3. Diagram rancangan perangkat keras

3.4.1 Rancangan Serial Peripheral Interface (SPI)

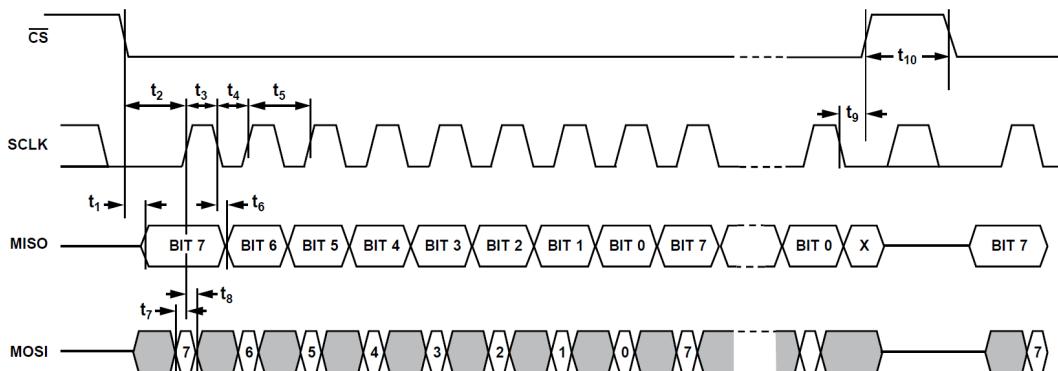
Antarmuka SPI digunakan untuk komunikasi antara STM32F411CEU6 dan modul EVAL-AD5941BATZ. IC AD5941 menyediakan antarmuka SPI yang memfasilitasi konfigurasi dan kontrol oleh mikrokontroler host. Mikrokontroler STM32F411CEU6 berperan sebagai *master device* untuk membaca dan menulis ke memori, register, dan FIFO dari IC AD5941 yang beroperasi sebagai *slave device*.

Setelah dilakukan studi literatur terhadap komponen-komponen yang ada pada modul EVAL-AD5941BATZ, antarmuka SPI menggunakan empat pin, yaitu CS (Chip Select), SCLK (Serial Clock), MISO (Master In Slave Out), dan MOSI (Master Out Slave In). Komunikasi SPI pada IC AD5941 memiliki batasan speed SCLK maksimum sebesar 16 MHz, kecepatan ini harus lebih rendah dari clock sistem IC AD5941. Setiap siklus komunikasi SPI mengirimkan sebanyak delapan bit data, di mana bit paling signifikan (Bit 7) dikirim dan diterima terlebih dahulu. Gambar 3.4 menunjukkan koneksi SPI antara STM32F411CEU6 dengan Modul EVAL-AD5941BATZ.



Gambar 3.4. Koneksi SPI STM32 dengan EVAL-AD5941BATZ

Operasi SPI dikonfigurasikan dalam mode *full-duplex* memungkinkan adanya isyarat data keluaran dari *slave device* yang dapat berjalan bersamaan dengan isyarat data keluaran dari *master device*. Selain pengaturan ini, terdapat dua parameter tambahan yang harus dikonfigurasi, yaitu polaritas *clock* dan fase *clock*. Konfigurasi ini didasarkan pada *timing diagram* yang terdapat di dalam *datasheet* [16] IC AD5941 pada bagian SPI. Timing diagram SPI ini ditunjukkan oleh Gambar 3.5.



Gambar 3.5. *Timing diagram* komunikasi SPI IC AD5941 [16]

Berdasarkan diagram dari *datasheet* IC AD5941 [16], waktu t_1 adalah waktu *setup* MISO dari *falling edge* CS dan waktu t_2 adalah waktu *setup* SCLK setelah CS menjadi LOW. Pada interval antara t_3 dan t_6 , terdapat perubahan nilai SCLK dari HIGH ke LOW yang menunjukkan penundaan sinyal MISO untuk mengirimkan data pada *falling edge* SCLK. Pada interval antara t_7 dan t_5 , terjadi perubahan nilai SCLK dari LOW ke HIGH yang menunjukkan bahwa data diterima oleh *slave device* pada *rising edge* SCLK. Berdasarkan informasi ini, sinyal MOSI dan MISO dikirimkan pada *falling edge* sinyal SCLK dan diambil sampelnya pada *rising edge* sinyal SCLK oleh host dan AD5941.

Oleh karena itu, operasi SPI diatur dalam Mode 0, di mana SCLK memiliki CPOL (Clock Polarity) = 0 (level awal sinyal clock) dan CPHA (Clock Phase) = 0 (kapan data disampel/dibaca dalam siklus SCLK).

Operasi dari antarmuka SPI perlu dikonfigurasi terlebih dahulu pada mikrokontroler untuk mengatur Mode dan frekuensinya. Terlebih, besar frekuensi SPI juga dipengaruhi oleh frekuensi *clock* sistem dan frekuensi *clock* dari bus APB yang menggerakkan periferal SPI. Dalam konteks ini, frekuensi SPI yang akan digunakan diatur agar tidak melebihi 8 MHz untuk memfasilitasi pengujian dalam penelitian. Pengaturan periferal SPI pada mikrokontroler dapat dipermudah dengan menggunakan HAL (Hardware Abstraction Layer), sehingga meminimalkan parameter yang harus diatur. Konfigurasi periferal SPI dijabarkan sebagai berikut:

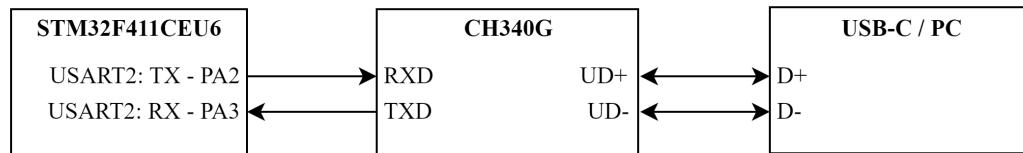
- *SPI Handler* = SPI1
- *Bus Handler* = APB2
- *APB2 Freq* = 100MHz
- *SPI1 Prescaler* = 16

Maka, nilai frekuensi SPI yang digunakan dijabarkan oleh Persamaan 3-1.

$$f_{SPI} = \frac{f_{APB2}}{SPI1Prescaler} = \frac{100MHz}{16} = 6,25MHz \quad (3-1)$$

3.4.2 Rancangan Universal Asynchronous Receiver Transmitter (UART)

Antarmuka UART digunakan sebagai komunikasi antara mikrokontroler dengan PC/Laptop selama proses pengaturan dan pengukuran impedansi baterai melalui perangkat lunak GUI. Pesat baud yang digunakan pada UART adalah 115200 bps untuk menghindari masalah *jamming* akibat kecepatan transfer data yang terlalu lambat. Penggunaan UART dengan kecepatan tinggi dalam penelitian ini dimungkinkan karena komunikasi dilakukan melalui kabel dengan jarak transmisi data yang tidak terlalu jauh. Gambar 3.6 menunjukkan koneksi UART antara STM32F411CEU6 dengan IC *USB-to-TTL Converter* CH340G dan PC/Laptop.



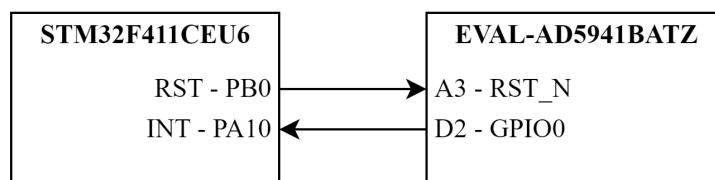
Gambar 3.6. Koneksi UART STM32 dengan EVAL-AD5941BATZ

Operasi dari antarmuka UART perlu dikonfigurasi terlebih dahulu pada mikrokontroler untuk mengatur pesat baud dan format data yang dikirimkan. Pengaturan periferal UART pada mikrokontroler dapat dipermudah dengan menggunakan HAL (Hardware Abstraction Layer), sehingga meminimalkan parameter yang harus diatur. Konfigurasi periferal UART adalah sebagai berikut:

- *UART Handler* = USART2
- *Baud Rate* = 115200
- *Word Length* = 8 bit
- *Stop Bits* = 1 bit
- *UART Parity* = None
- *UART Mode* = Rx & Tx

3.4.3 Rancangan General Purpose Input/Output (GPIO)

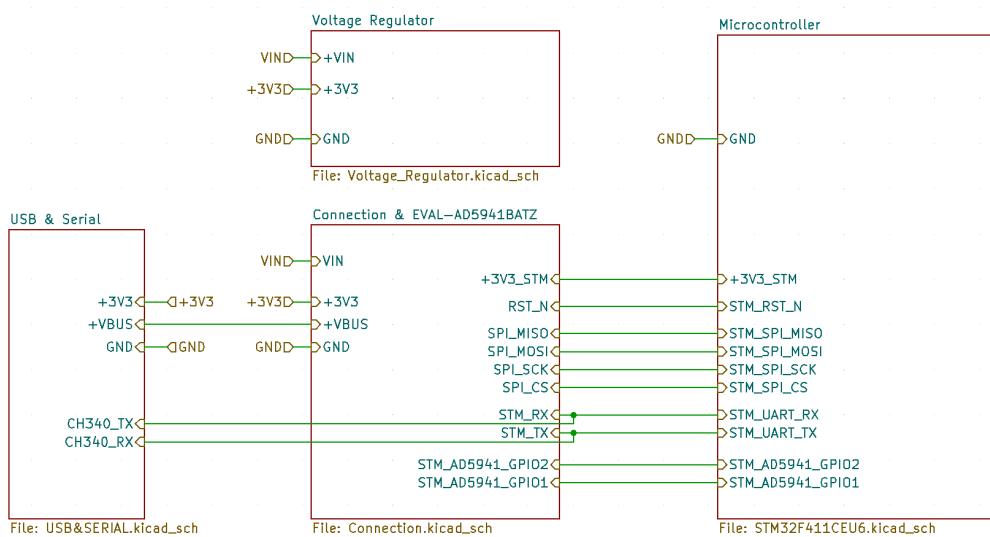
GPIO digunakan sebagai pemicu pin *RESET* pada IC AD5941 saat melakukan setup sebelum komunikasi SPI dengan mikrokontroler. Pin ini berfungsi sebagai enable chip yang dioperasikan dalam mode *active low*. Selain itu, GPIO juga digunakan untuk menerima flag dari IC AD5941 yang menandakan bahwa nilai pengukuran sudah tersimpan dalam register dan siap untuk dikirimkan kepada mikrokontroler melalui komunikasi SPI. Gambar 3.7 menunjukkan koneksi GPIO antara STM32F411CEU6 dengan Modul EVAL-AD5941BATZ.



Gambar 3.7. Koneksi GPIO STM32 dengan EVAL-AD5941BATZ

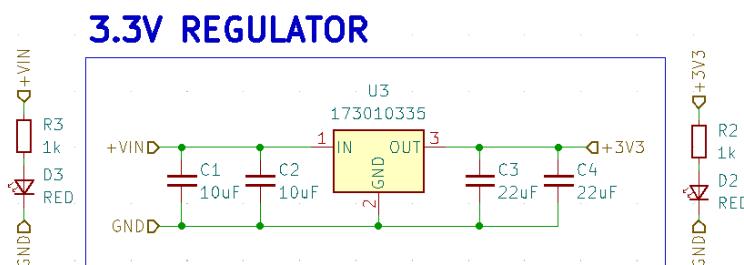
3.4.4 Desain Sirkuit PCB Antarmuka

PCB Antarmuka berfungsi sebagai penghubung Modul EVAL-AD5941BATZ dengan PC/Laptop. *Board* ini terdiri dari mikrokontroler, regulator tegangan, *USB-to-TTL Converter*, testpoint, LED indikator, koneksi untuk Modul EVAL-AD5941BATZ, dan koneksi eksternal lainnya. Gambar 3.8 menampilkan *root hierarchy schematic* dari PCB Antarmuka yang dibagi menjadi empat *hierarchical sheet* agar schematic menjadi lebih teratur. Pembagian ini meliputi *sheet* Mikrokontroler, Voltage Regulator, USB & Serial, dan Connection.



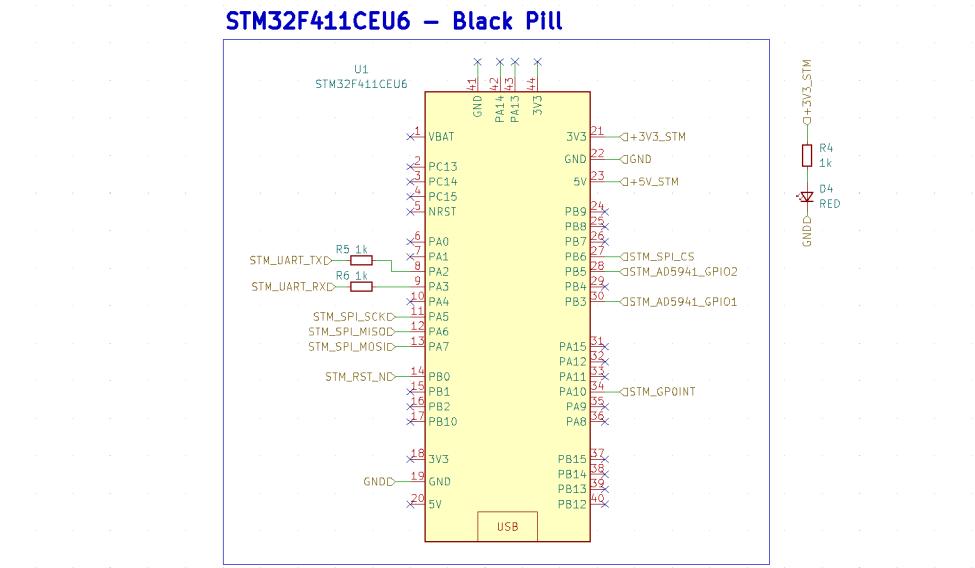
Gambar 3.8. *Root hierarchy schematic* PCB Antarmuka

Keseluruhan sistem dalam PCB Antarmuka akan beroperasi pada tegangan 3,3V. Untuk memastikan stabilitas tegangan ini, digunakan regulator tegangan seri WPME-FDSM 173010335 dengan spesifikasi tegangan masukan 5 – 36V dan tegangan keluaran 3,3V. Regulator tegangan ini dapat menerima dua opsi masukan, yaitu tegangan 5V langsung dari koneksi USB tipe C atau dari sumber daya eksternal melalui konektor Barrel Jack. Gambar 3.9 menunjukkan isi dari Sheet Voltage Regulator yang mencakup simbol komponen WPME-FDSM 173010335, kapasitor filter, dan LED indikator.



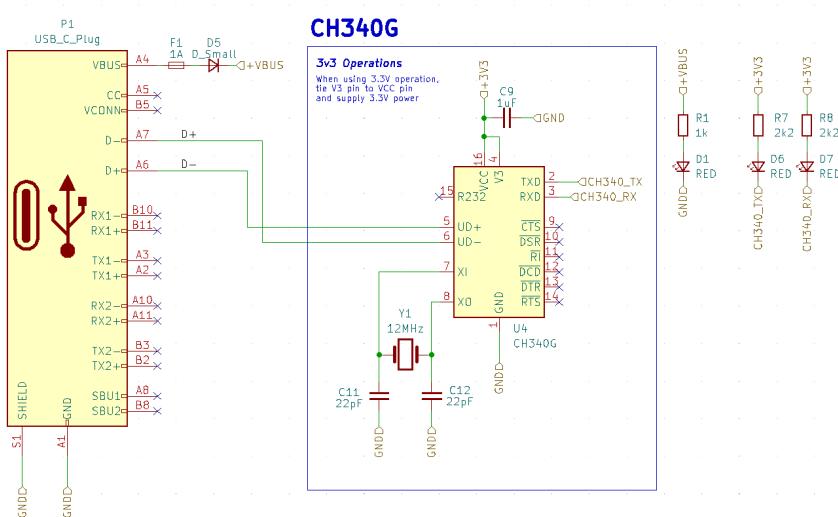
Gambar 3.9. *Hierarchical sheet* Voltage Regulator

Mikrokontroler yang digunakan adalah STM32F411CEU6 yang dikemas dalam format yang dikenal sebagai “black pill” dengan 40 pin GPIO. Gambar 3.10 menampilkan isi dari *sheet* Microcontroller yang mencakup simbol komponen STM32F411CEU6, resistor, dan LED indikator.



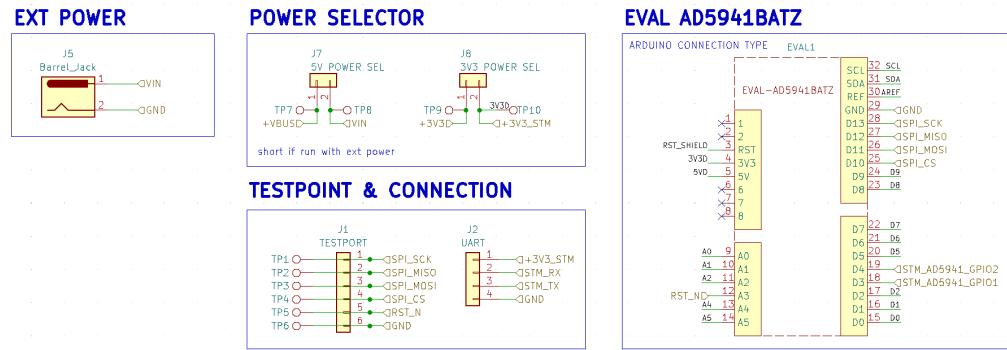
Gambar 3.10. *Hierarchical sheet* Microcontroller

Komunikasi antara mikrokontroler dan PC/Laptop membutuhkan konverter data serial USB yang merupakan pasangan diferensial (D+ dan D-) menjadi TTL (Transistor-Transistor Logic) yang mendukung protokol UART. Untuk memenuhi kebutuhan ini, digunakan IC CH340G yang memiliki fitur-fitur yang sesuai. Gambar 3.11 menampilkan isi dari *sheet* USB & Serial yang mencakup simbol komponen USB tipe C, IC CH340G, berbagai komponen pendukung, dan LED indikator.



Gambar 3.11. *Hierarchical sheet* USB & Serial

Gambar 3.12 menampilkan isi dari *sheet Connection* & EVAL-AD5941BATZ yang mencakup semua koneksi pada PCB dan Modul EVAL-AD5941BATZ. Bagian ini mencakup komponen seperti Barrel Jack sebagai masukan daya eksternal, pin selektor tegangan masukan untuk regulator tegangan, pin selektor tegangan 3,3V, tespoint untuk memudahkan pengujian, serta pin soket yang kompatibel dengan koneksi Arduino Uno dari Modul EVAL-AD5941BATZ.



Selain parameter minimum yang tercantum dalam Tabel 3.1, lebar jalur pada PCB juga disesuaikan dengan penggunaannya. Sebagai contoh, jalur untuk koneksi data atau sinyal umumnya lebih kecil dibandingkan dengan jalur untuk suplai daya. Setiap keperluan ini memiliki lebar jalur yang ditentukan berdasarkan besar arus yang dihantarkan dan target resistansi maksimum yang diizinkan sepanjang jalur.

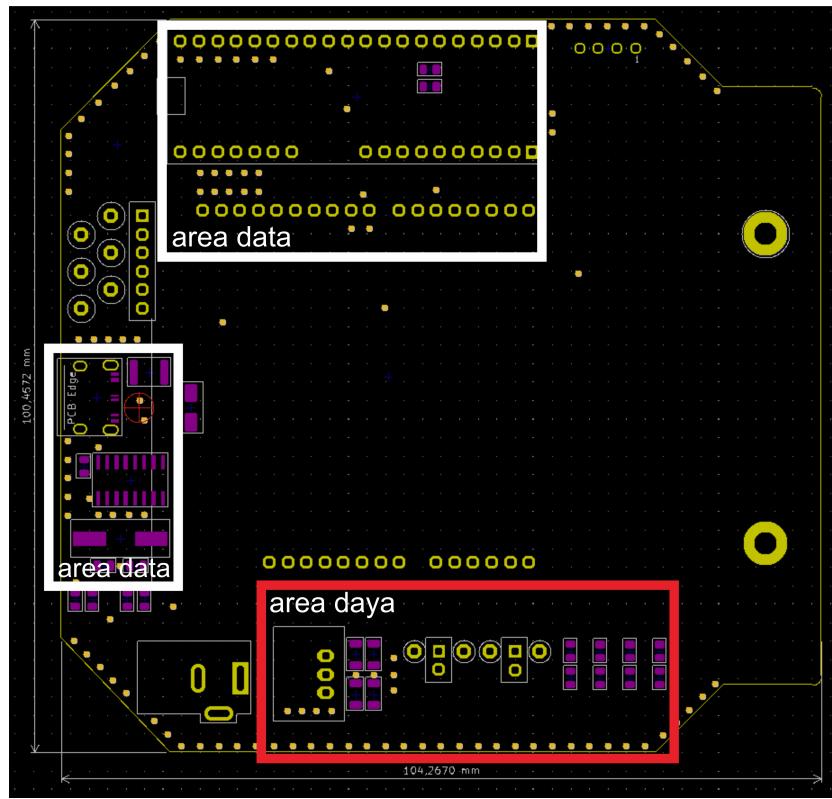
Lebar jalur minimum pada penelitian ini dihitung menggunakan kalkulator yang tersedia dalam perangkat lunak KiCAD. Perhitungan tersebut mengacu pada standar IPC 2221 dengan parameter kenaikan suhu $\Delta T = 10$ derajat. Keputusan tentang lebar jalur juga dipengaruhi oleh spesifikasi minimum dari produsen PCB, jarak antar *pad* komponen, serta arus maksimum yang akan melewati pin GPIO pada mikrokontroler yang digunakan, sebagaimana yang dijelaskan dalam Tabel 3.2.

Tabel 3.2. Lebar jalur yang digunakan pada PCB Antarmuka

Lebar Jalur	Arus Maksimum	Resistansi @ 10 cm	Deskripsi
0,254 mm	0,880 A	0,1934 Ω	Digunakan untuk menghubungkan USB tipe C dan cip CH340G
0,400 mm	1,230 A	0,1228 Ω	Digunakan pada semua jalur data
0,800 mm	2,034 A	0,0614 Ω	Digunakan pada semua jalur distribusi daya
> 4 mm	> 6 A	< 0,02 Ω	Digunakan pada jalur daya antara koneksi eksternal dan regulator tegangan

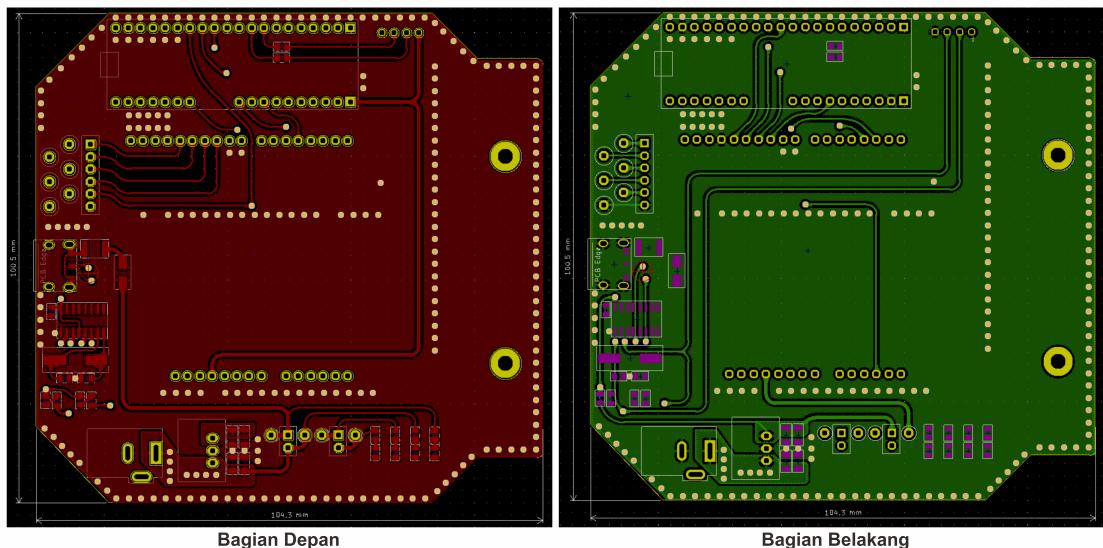
Setelah menentukan parameter-parameter di atas, langkah selanjutnya adalah penempatan komponen dalam PCB. Umumnya, kelompok komponen daya akan dipisahkan dari kelompok data untuk mengurangi gangguan yang mungkin terjadi. Pada desain PCB antarmuka, regulator tegangan dan kapasitor filternya dikelompokkan dan ditempatkan sedekat mungkin dengan sumber daya. Hal ini bertujuan untuk meminimalisir gangguan pada area IC CH340G dan terutama jalur SPI antara mikrokontroler dan Modul EVAL-AD5491BATZ. Penempatan komponen ini ditampilkan pada Gambar 3.13.

Tahap selanjutnya adalah menyusun jalur konduktor, yang sering disebut sebagai *routing*, untuk menghubungkan komponen-komponen di PCB. Jalur-jalur ini dapat ditempatkan di kedua sisi PCB dan dihubungkan menggunakan via. Beberapa hal yang perlu diperhatikan selama proses *routing* adalah jarak antar jalur, panjang jalur, interferensi, dan sudut belokan jalur.



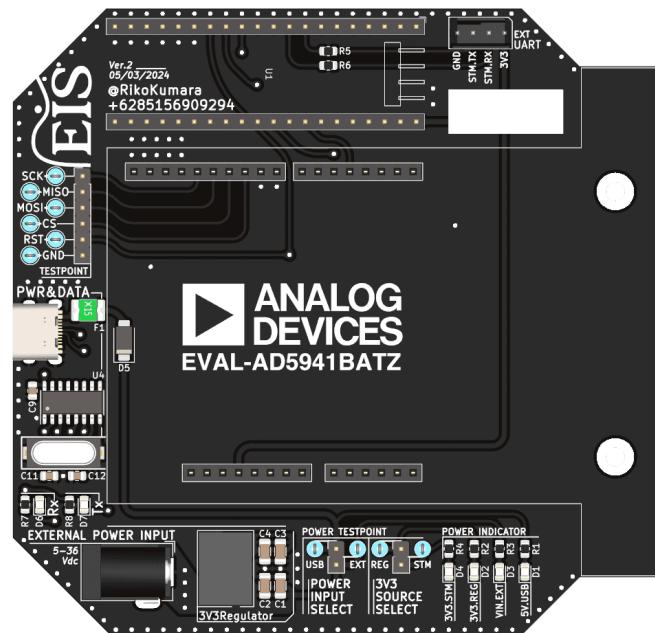
Gambar 3.13. Pengelompokan komponen pada PCB Antarmuka

Setelah proses *routing* selesai, area permukaan PCB yang tidak dilalui oleh jalur akan diisi dengan *ground plane*. *Ground plane* pada kedua sisi PCB akan dihubungkan menggunakan via pada sekeliling PCB. Tujuannya adalah untuk menyatukan *ground plane* dan menghindari pembentukan kapasitansi pada PCB. Gambar 3.14 menunjukkan hasil *routing* pada permukaan bawah PCB.

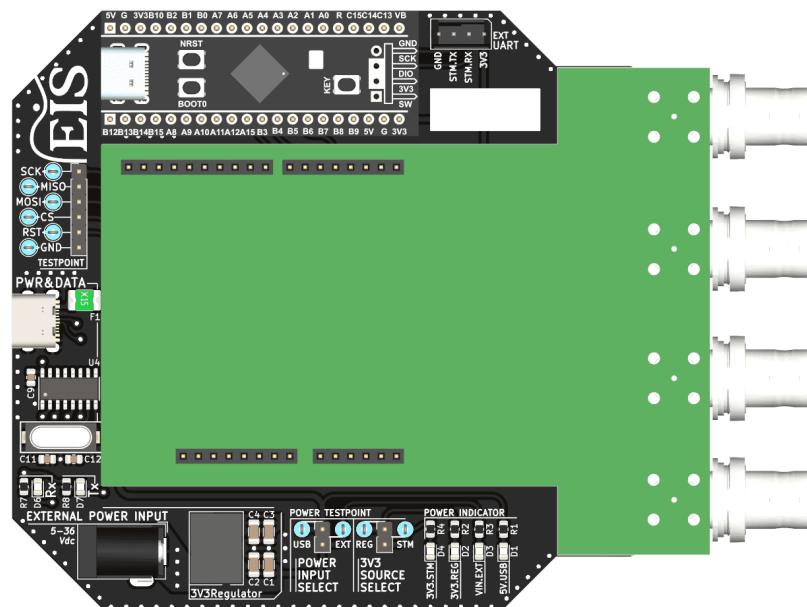


Gambar 3.14. Hasil *routing* PCB Antarmuka pada *software KiCAD*

Tahap terakhir dari perancangan PCB Antarmuka adalah penambahan label dan simbol yang diperlukan untuk memudahkan pengujian dan perakitan PCB. Gambar 3.15 menampilkan hasil perancangan PCB Antarmuka yang dihasilkan dari tampilan 3D dalam perangkat lunak KiCAD. Gambar 3.16 menunjukkan PCB Antarmuka tergabung dengan mikrokontroler STM32F411CEU6 dan modul EVAL-AD5941BATZ.



Gambar 3.15. Hasil perancangan PCB Antarmuka



Gambar 3.16. PCB Antarmuka dengan STM32 "Black Pill" dan EVAL-AD5941BATZ

3.4.6 Bill of Materials

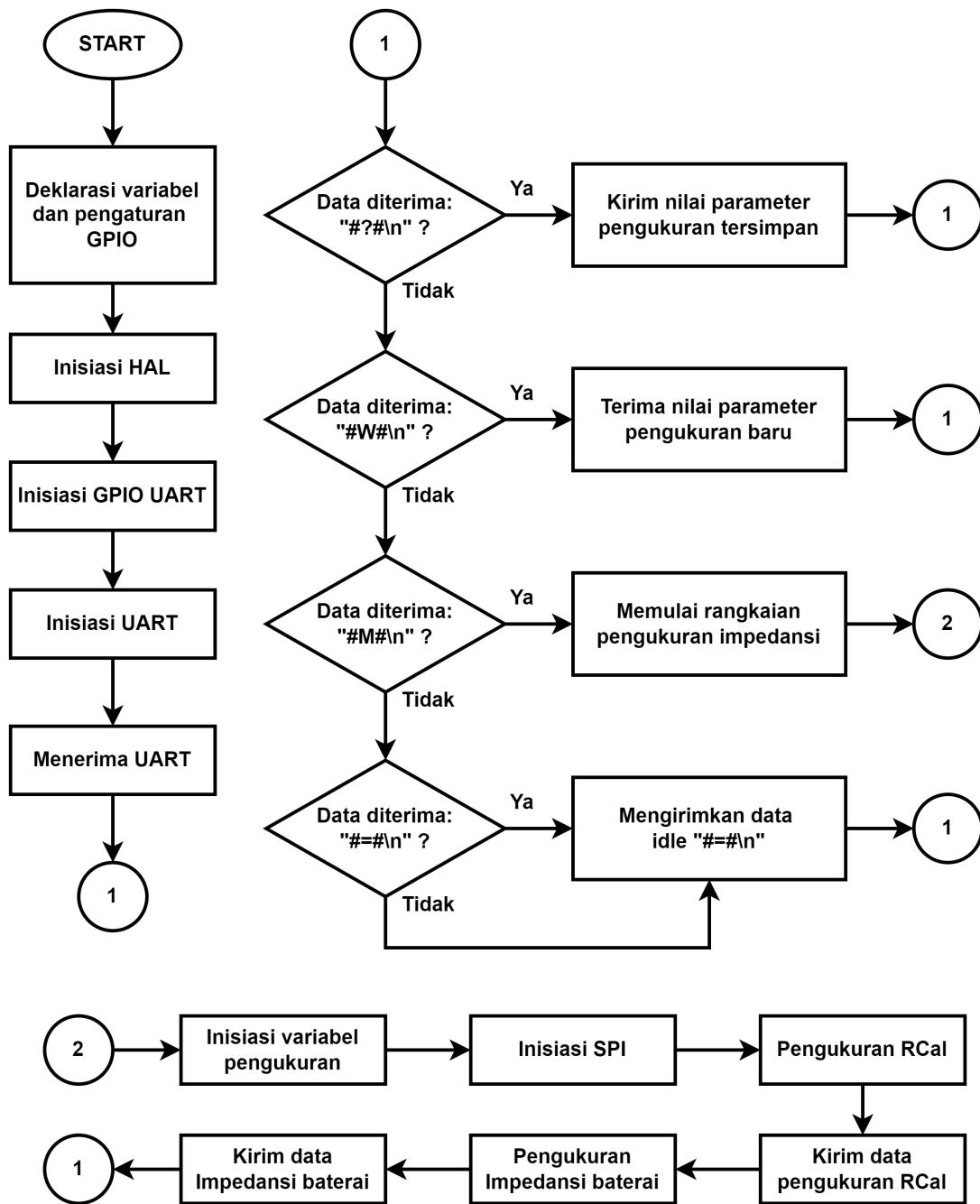
Bill of Materials (BoM) adalah sebuah list yang berisi komponen, bahan, jumlah, dan harga yang dibutuhkan untuk merakit atau membuat suatu produk akhir. BoM dibuat untuk menentukan jenis komponen yang diperlukan dan jumlah yang perlu dibeli untuk merakit suatu produk akhir. BoM perancangan PCB Antarmuka dan sistem pengukuran impedansi berbasis Modul EVAL-AD5941BATZ ditampilkan pada Tabel 3.3.

Tabel 3.3. *Bill of Materials* peranti pengukuran impedansi

No	Nama	Keterangan	Jml	Harga	Subtotal
1	Modul EVAL-AD5941BATZ		1	Rp 3.348.807,00	Rp 3.348.807,00
2	STM32F411 BlackPill WeAct		1	Rp 135.000,00	Rp 135.000,00
3	Biaya Cetak PCB Antarmuka		1	Rp 144.000,00	Rp 144.000,00
4	173010335 Voltage Regulator	3,3V Fixed	1	Rp 114.145,00	Rp 114.145,00
5	CH340G	SMD	1	Rp 12.000,00	Rp 12.000,00
6	USB-C Connector	16 pin	1	Rp 3.000,00	Rp 3.000,00
7	Fuse PPTC SMD1812	1 A 16V	1	Rp 3.000,00	Rp 3.000,00
8	Crystal SMD	12MHz	1	Rp 1.500,00	Rp 1.500,00
9	Resistor SMD0805	1 kOhm	8	Rp 100,00	Rp 800,00
10	Kapasitor SMD0805 MLCC 16V	1 uF	1	Rp 300,00	Rp 300,00
11	Kapasitor SMD1206 MLCC 50V	10 uF	2	Rp 1.999,00	Rp 3.998,00
12	Kapasitor SMD1206 MLCC 25V	22 uF	2	Rp 1.899,00	Rp 3.798,00
13	Kapasitor SMD0805 MLCC 16V	22 pF	2	Rp 1.490,00	Rp 2.980,00
14	LED SMD0805	Merah	4	Rp 200,00	Rp 800,00
15	LED SMD0805	Biru	2	Rp 200,00	Rp 400,00
16	PinSocket 2,54mm	1x16pin	4	Rp 1.000,00	Rp 4.000,00
17	PinHeader 2,54mm	1x40pin	1	Rp 2.000,00	Rp 2.000,00
Total					Rp 3.666.383,00

3.5 Perancangan Perangkat Lunak untuk Mikrokontroler

Perangkat lunak firmware untuk mikrokontroler STM32F411CEU6 dikembangkan menggunakan Keil uVision 5 dengan dukungan dari *library* AD5941 yang disediakan oleh Analog Devices. Firmware ini dirancang untuk mengatur dan mengelola pengukuran impedansi menggunakan IC AD5941, serta mentransmisikan data hasil pengukuran ke PC/Laptop melalui komunikasi serial. Keil uVision 5 dipilih karena menyediakan IDE yang lengkap untuk mikrokontroler STM32, termasuk fitur *debugger* yang canggih, manajemen proyek yang efisien, dan dukungan yang luas untuk mikrokontroler ARM Cortex-M.



Gambar 3.17. Diagram alir program STM32

Library AD5941 dari Analog Devices menjadi dasar dalam pengembangan firmware ini. *Library* tersebut menyediakan fungsi dan driver yang diperlukan untuk mengoperasikan IC AD5941 sebagai analog front-end untuk pengukuran impedansi. Dengan memanfaatkan *library* ini, pengembangan firmware menjadi lebih efisien dan terstruktur.

Firmware ini dirancang dengan struktur modular yang terdiri dari beberapa file sumber utama. File-file tersebut mencakup main.c sebagai kode utama yang mengontrol alur eksekusi program, STM32F411Port.c untuk mengatur port dan GPIO, BATImpedan-

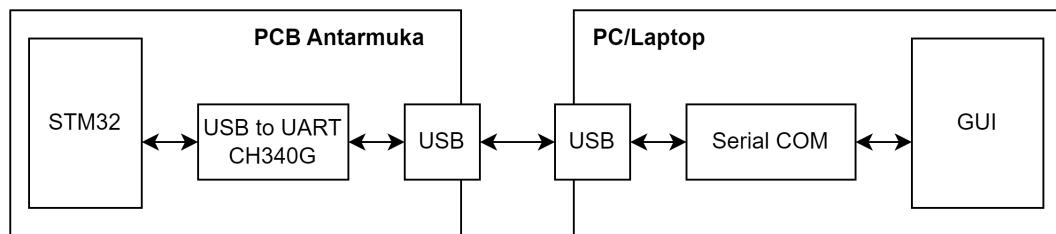
ce.c untuk konfigurasi pengukuran EIS, AD5941Main.c yang berisi fungsi-fungsi utama untuk pengukuran EIS menggunakan IC AD5941, dan ad5941.c sebagai *library* yang diperlukan untuk mengoperasikan IC AD5941.

Diagram alir yang ditampilkan pada Gambar 3.17 menjelaskan secara detail alur kerja firmware. Diagram ini mencakup langkah-langkah dari inisialisasi sistem, konfigurasi IC AD5941, pengambilan dan pengolahan data, hingga komunikasi data dengan PC/Laptop serta pengaturan parameter pengukuran. Setiap langkah dalam diagram alir tersebut diimplementasikan melalui fungsi-fungsi yang terdapat dalam file sumber utama firmware, memastikan bahwa proses pengukuran dan komunikasi berjalan secara efisien dan akurat.

3.6 Perancangan Perangkat Lunak untuk *Graphical User Interface*

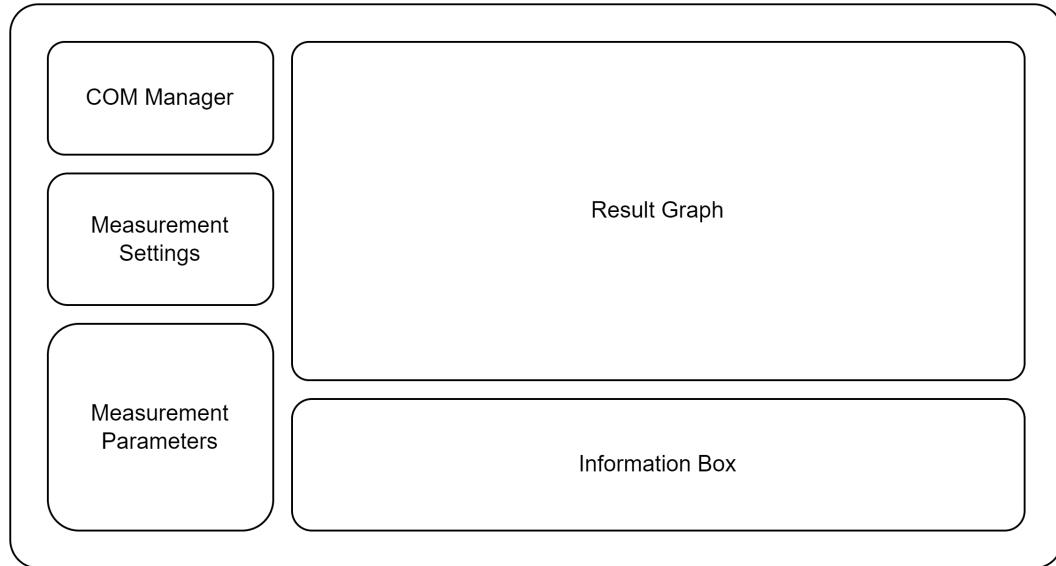
3.6.1 Gambaran Umum GUI

Perancangan *graphical user interface* (GUI) dikembangkan pada Python. Tujuan utama GUI adalah untuk memudahkan pengguna dalam berinteraksi dengan data yang diterima dari mikrokontroler STM32F411CEU6. Beberapa fitur yang tersedia di GUI ini meliputi COM Manager, pengaturan parameter pengukuran, serta tampilan data dalam bentuk plot Nyquist atau plot Bode. Proses penerimaan data serial dari mikrokontroler ke PC/Laptop ditunjukkan oleh Gambar 3.18.



Gambar 3.18. Diagram penerimaan data serial dari STM32 ke PC/Laptop

Gambaran umum dari antarmuka pengguna yang dirancang dapat dilihat pada Gambar 3.19 yang menunjukkan tata letak dasar dari GUI yang mencakup beberapa komponen utama seperti area pengaturan koneksi serial (COM Manager), pengaturan parameter pengukuran, serta area tampilan data yang dikirimkan oleh mikrokontroler.



Gambar 3.19. Gambaran umum GUI

3.6.2 Penggunaan *Library* dalam Perancangan GUI

Perancangan GUI memerlukan penggunaan *library* untuk membantu implementasi dari fitur yang telah ditentukan. Perancangan GUI ini menggunakan *library* ‘tkinter’ dan beberapa *library* pendukung lainnya. *Library* yang digunakan meliputi ‘tkinter’, ‘matplotlib’, ‘mpl_interactions’, ‘math’, ‘threading’, ‘messagebox’, ‘filedialog’, ‘openpyxl’, ‘datetime’, dan ‘os’. Berikut penjelasan masing-masing *library* dalam konteks perancangan GUI:

1. Tkinter

‘Tkinter’ adalah *library* standar Python untuk pembuatan GUI. *Library* ini menyediakan berbagai widget yang dapat digunakan untuk membangun antarmuka pengguna, seperti label, tombol, dan berbagai elemen GUI lainnya.

2. Matplotlib

Library ‘matplotlib’ digunakan untuk membuat plot dan grafik dalam GUI. Dengan menggunakan ‘FigureCanvasTkAgg’, plot yang dihasilkan oleh Matplotlib dapat ditampilkan dalam antarmuka Tkinter.

3. mpl_interactions

Library ‘mpl_interactions’ digunakan untuk menambahkan interaktivitas pada plot Matplotlib, seperti fitur zoom dan panning. Fitur ini memungkinkan pengguna untuk berinteraksi dengan plot hasil pengukuran impedansi dalam proses analisis.

4. math

Library 'math' menyediakan berbagai fungsi matematika dasar dan lanjutan yang diperlukan dalam berbagai perhitungan dan operasi numerik. Beberapa fungsi yang disediakan oleh math adalah fungsi trigonometri, logaritma, eksponensial, dan operasi pembulatan.

5. threading

Library 'threading' digunakan untuk menangani *thread*, memungkinkan aplikasi untuk menjalankan operasi secara paralel atau *asynchronous*. Fungsionalitas ini penting untuk menjaga responsivitas antarmuka pengguna, terutama dalam aplikasi yang menjalankan tugas berat atau membutuhkan operasi I/O. *Library* ini akan digunakan saat melakukan komunikasi serial dengan perangkat keras.

6. Messagebox

Library 'messagebox' digunakan untuk menampilkan kotak dialog pesan yang berfungsi sebagai alat interaksi antara aplikasi dan pengguna. Kotak dialog ini dapat digunakan untuk menampilkan informasi, peringatan, atau pertanyaan konfirmasi, yang berguna dalam memberikan umpan balik atau meminta input dari pengguna.

7. Filedialog

Library 'filedialog' digunakan untuk membuka dialog pemilihan file atau direktori. Fungsionalitas ini sangat penting karena digunakan dalam interaksi sistem file, seperti membuka, menyimpan, atau mengekspor file.

8. Openpyxl

Library 'openpyxl' digunakan untuk bekerja dengan file Excel. *Library* ini memungkinkan aplikasi untuk membuat, membaca, dan memodifikasi file Excel, yang berguna dalam penyimpanan data hasil pengukuran impedansi.

9. Datetime

Library 'datetime' digunakan untuk menangani operasi yang berkaitan dengan tanggal dan waktu. Fungsi ini digunakan untuk mencatat waktu dan tanggal saat menyimpan file Excel hasil pengukuran impedansi.

10. os

Library 'os' menyediakan berbagai fungsi untuk berinteraksi dengan sistem operasi, seperti manipulasi direktori dan file. Fungsi-fungsi digunakan untuk mengelola file dan mengubah direktori kerja saat menyimpan file Excel hasil pengukuran impedansi.

Penggunaan *library-library* tersebut memungkinkan pembuatan GUI yang interaktif dan fungsional. Dengan mengkombinasikan kemampuan dan fitur dari masing-masing *library*, implementasi GUI diharapkan dapat memberi tampilan yang membantu proses karakterisasi baterai litium-ion.

3.6.3 Tampilan Akhir GUI

Tampilan akhir GUI yang dikembangkan memiliki beberapa fitur yang mendukung proses pengukuran dan karakterisasi data impedansi baterai. Fitur-fitur tersebut diuraikan sebagai berikut:

1. Pengaturan port komunikasi

Terdapat *frame* "COM Manager" dengan fitur yang memungkinkan pengguna untuk mengelola koneksi serial antara PC/Laptop dan mikrokontroler. Pengguna dapat memilih port COM yang tersedia, tetapi parameter baud rate ditetapkan pada satu nilai sesuai dengan pengaturan perangkat keras.

2. Pengaturan parameter pengukuran

Terdapat *frame* "Measurement Settings" yang memungkinkan pengguna untuk mengatur parameter pengukuran sesuai dengan kebutuhan. Pengaturan ini mencakup nilai awal sapuan frekuensi, nilai akhir sapuan frekuensi, dan jumlah titik sapuan frekuensi.

3. Menampilkan data

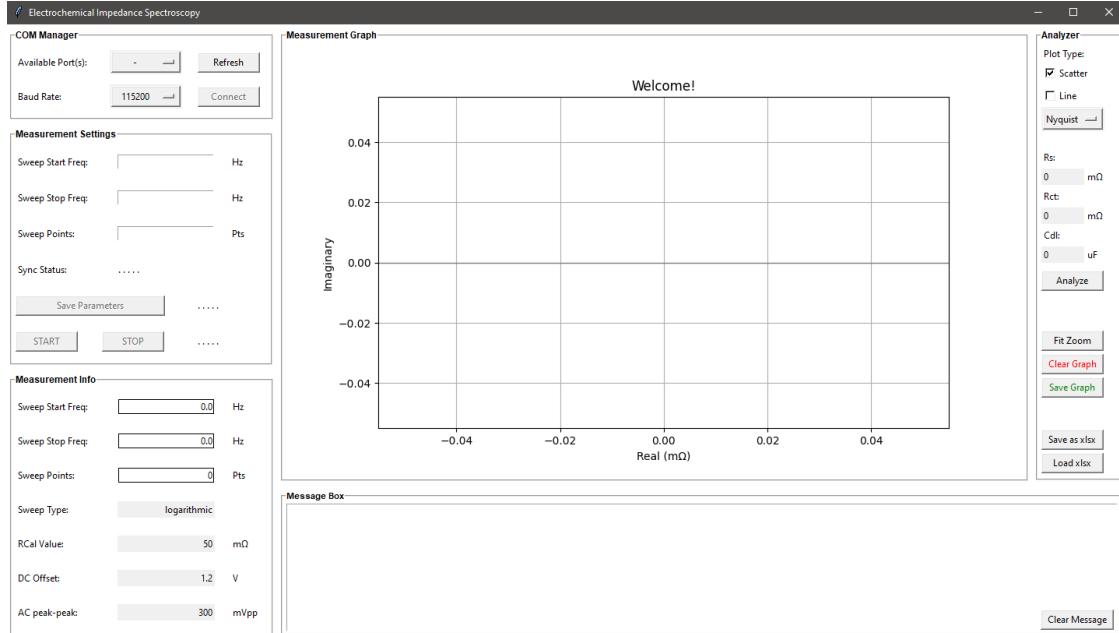
Terdapat *frame* "Measurement Graph" untuk menampilkan data impedansi yang dikirimkan oleh mikrokontroler atau memuat file tersimpan dalam bentuk:

- Nyquist Plot: menampilkan data impedansi dalam bentuk plot Nyquist yang berisi nilai riil dan imajiner.
- Bode Plot: menampilkan data impedansi dalam magnitude dan fasanya terhadap frekuensi.

4. Pengaturan plot dan analisis nilai komponen internal baterai

Terdapat *frame* "Analyzer" yang memungkinkan pengguna untuk memilih jenis tampilan plot (Nyquist atau Bode) dengan bentuk *scatter* dan/atau *line*. Pengguna juga dapat menyimpan plot sebagai gambar PNG dan juga menyimpan data pengukuran impedansi sebagai *file* Excel. Selain itu, terdapat fitur untuk menentukan nilai komponen internal baterai dengan melakukan perhitungan sederhana berdasarkan kurva karakteristik yang ditampilkan.

Dengan adanya GUI, pengguna dapat lebih mudah dalam mengatur dan memantau proses pengukuran serta menganalisis data yang dihasilkan. GUI ini dirancang untuk memudahkan pengambilan serta pengolahan data impedansi. Tampilan awal GUI yang telah dirancang ditunjukkan oleh Gambar 3.20.



Gambar 3.20. Hasil perancangan GUI

BAB IV

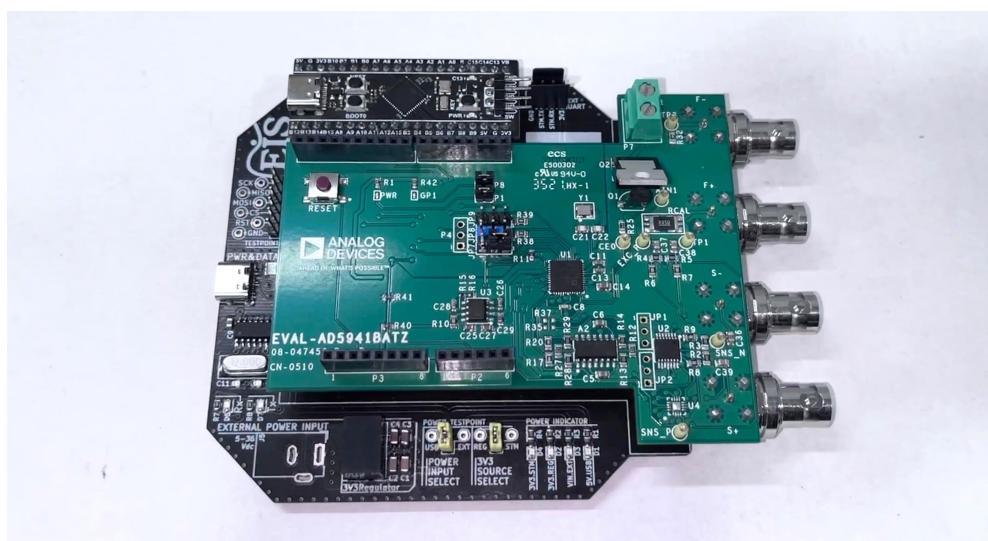
HASIL DAN PEMBAHASAN

4.1 Implementasi Perangkat Keras

Pada penelitian ini, peranti EIS berbasis Modul EVAL-AD5941BATZ dibuat untuk membantu karakterisasi daur ulang baterai litium-ion sehingga dibutuhkan sistem yang portabel dan meminimalisir biaya yang dikeluarkan. PCB Antarmuka yang telah dirancang memiliki ukuran 105 x 105 mm dan saat keseluruhan sistem digabungkan memiliki ukuran 105 x 131 mm. Gambar 4.1 menunjukkan rancangan perangkat PCB Antarmuka yang telah diimplementasikan. Sedangkan Gambar 4.2 menunjukkan implementasi keseluruhan sistem pengukuran impedansi.



Gambar 4.1. Implementasi rancangan PCB Antarmuka



Gambar 4.2. Implementasi rancangan peranti pengukuran impedansi

4.2 Pengujian USB-to-UART TTL CH340G

Pengujian komunikasi UART dilakukan untuk mengetahui apakah parameter dalam proses komunikasi serial sudah berjalan dengan semestinya.

4.2.1 Protokol Komunikasi UART

Komunikasi serial antara PCB Antarmuka dan PC/Laptop memiliki protokol pertintah yang telah ditentukan agar komunikasi dapat berjalan dengan baik. Protokol ini menggunakan metode *parsing* dengan pemisah berupa karakter '#'. Perintah dari GUI ditampilkan pada Tabel 4.1 dan respon perangkat ditampilkan pada Tabel 4.2.

Tabel 4.1. Perintah GUI untuk perangkat

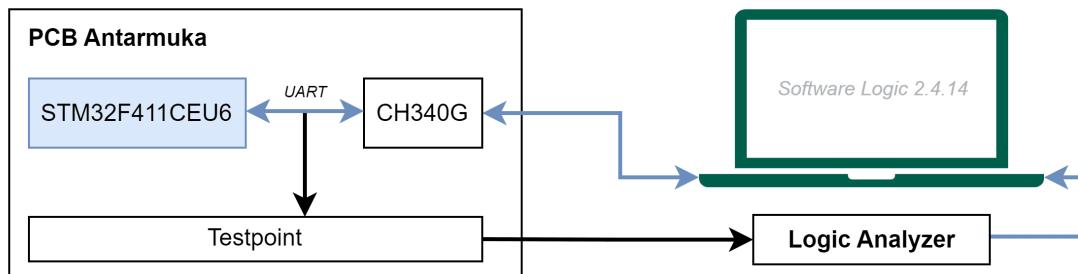
Nama	Nilai	Deskripsi
IDLE	#=#\n	Memerintahkan perangkat masuk ke <i>idle mode</i>
CONNECT	#?#\n	Memerintahkan sinkronisasi dengan perangkat
WRITE_PARAM	#W#\n	Memerintahkan penulisan parameter EIS baru
	#1#[]#\n	Penulisan parameter <i>sweep start frequency</i>
	#2#[]#\n	Penulisan parameter <i>sweep stop frequency</i>
	#3#[]#\n	Penulisan parameter <i>sweep points</i>
MEASURE_INIT	#M#\n	Memerintahkan inisialisasi EIS

Tabel 4.2. Respon Perangkat terhadap perintah GUI

Nama	Nilai	Deskripsi
IDLE	#=#\n	Respon bahwa perangkat dalam <i>idle mode</i>
CMD_OK	#K#\n	Respon bahwa perintah telah diterima dan siap melakukan perintah
SAVED_PARAM	#P#[]#[]#[]#\n	Memberi respon perintah CONNECT dengan mengirim parameter tersimpan
RCAL_INIT	#RC#\n	Isyarat bahwa pengukuran RCAL dimulai
IMP_INIT	#IM#\n	Isyarat bahwa pengukuran impedansi dimulai
RCAL_DATA	#R#[]#[]#[]#\n	Mengirim data pengukuran resistor kalibrasi
IMP_DATA	#I#[]#[]#[]#\n	Mengirim data pengukuran impedansi

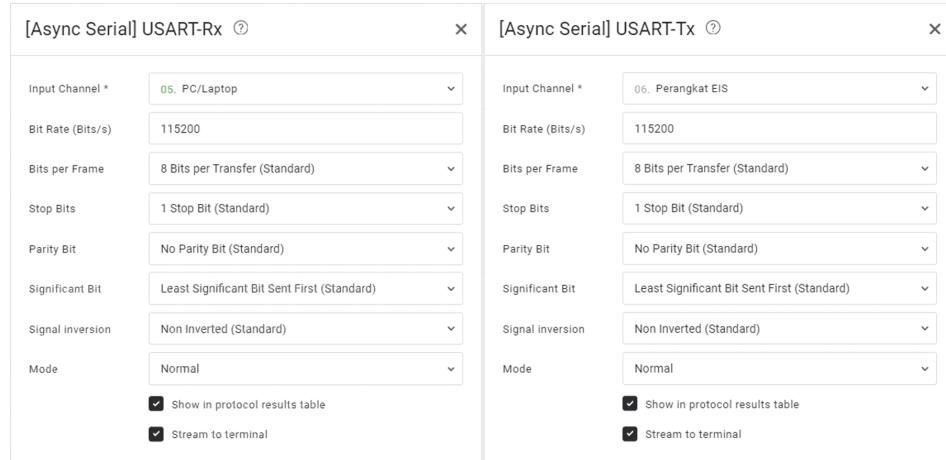
4.2.2 Skema dan Pengaturan Pengujian

Pengujian ini dilakukan menggunakan perangkat lunak Logic 2.4.14 dan perangkat modul Logic Analyzer untuk membaca komunikasi serial UART antara PC/Laptop dan perangkat PCB Antarmuka. Instalasi dan koneksi perangkat modul Logic Analyzer ditampilkan pada Gambar 4.3. Kemudian, fungsi pengiriman perintah akan dilakukan dengan bantuan perangkat lunak Hterm.



Gambar 4.3. Skema pengujian komunikasi UART

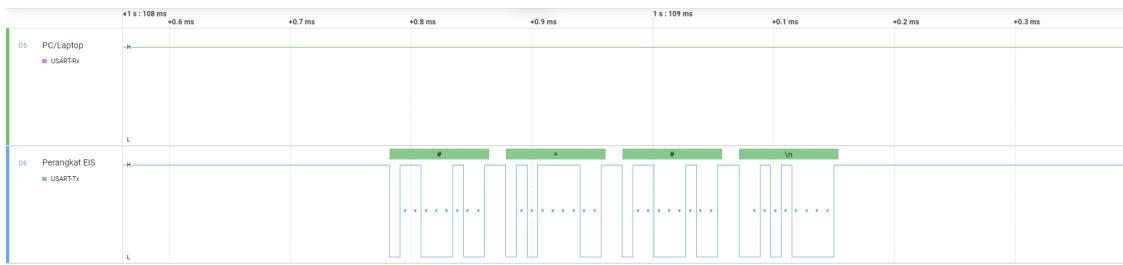
Untuk mendapatkan hasil pengujian yang baik, perlu dilakukan konfigurasi komunikasi UART yang sesuai. Konfigurasi parameter pada perangkat lunak Logic 2.4.14 ditunjukkan pada Gambar 4.4.



Gambar 4.4. Konfigurasi parameter pada *software Logic*

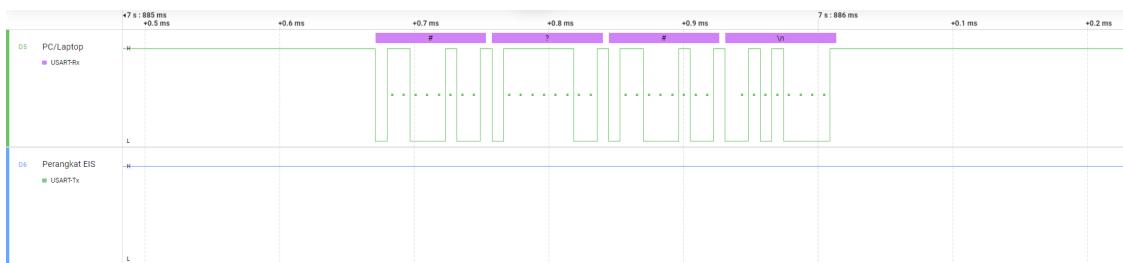
4.2.3 Hasil Pengujian UART

Hasil pengujian komunikasi UART dengan bantuan perangkat modul Logic Analyzer dapat dipulihkan pada *software Logic* sesuai dengan yang data yang dikirim oleh mikrokontroler. Gambar 4.5 menampilkan bahwa saat awal perangkat dinyalakan, perangkat mengirim respon IDLE.



Gambar 4.5. Respon IDLE oleh perangkat pada *software Logic*

Selanjutnya, dilakukan pengujian sinkronisasi perangkat dengan mengirim perintah CONNECT. Gambar 4.6 menampilkan bahwa perintah terkirim dengan benar.



Gambar 4.6. Perintah CONNECT oleh GUI pada *software Logic*

Setelah menerima perintah CONNECT, perangkat langsung mengirim respon SAVED_PARAM yang ditunjukkan oleh Gambar 4.7. Data yang dikirimkan oleh perangkat menunjukkan bahwa parameter sweep start = 1 Hz, sweep stop = 50.000 Hz, dan sweep points = 50 titik.



Gambar 4.7. Respon SAVED_PARAM oleh perangkat pada *software Logic*

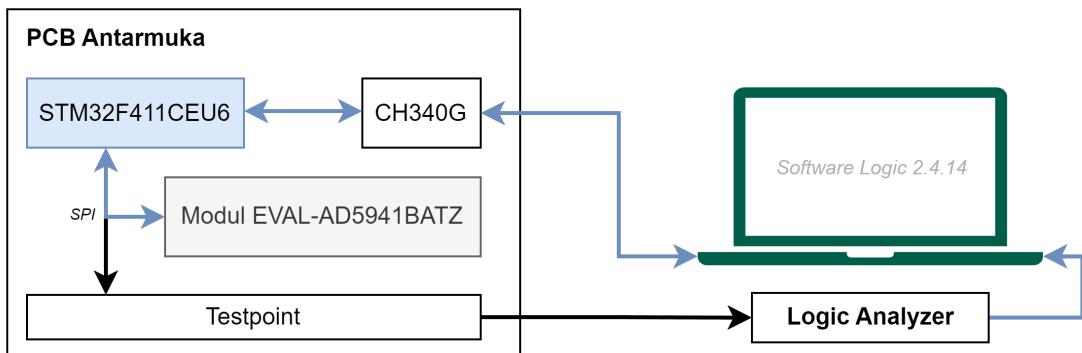
Berdasarkan pengujian ini, parameter konfigurasi komunikasi UART pada mikrokontroler sudah diatur dengan benar.

4.3 Pengujian Modul EVAL-AD5941BATZ

Pada tahap ini akan dilakukan pengujian untuk mengakses IC AD5941 yang berada pada modul EVAL-AD5941BATZ melalui protokol komunikasi SPI. Berikut adalah berbagai pengujian yang dilakukan.

4.3.1 Skema dan Pengaturan Pengujian

Pengujian ini dilakukan menggunakan perangkat lunak Logic 2.4.14 dan perangkat modul Logic Analyzer untuk membaca komunikasi serial SPI antara perangkat PCB Antarmuka dan Modul EVAL-AD5941BATZ. Instalasi dan koneksi perangkat modul Logic Analyzer ditampilkan pada Gambar 4.8.



Gambar 4.8. Skema pengujian komunikasi SPI

4.3.2 Protokol Perintah SPI IC AD5941

Komunikasi SPI pada IC AD5941 memiliki protokol perintah yang telah ditentukan agar komunikasi dapat berjalan dengan baik. Byte perintah ini menentukan protokol SPI yang akan digunakan untuk setiap transaksi SPI. Byte perintah yang dapat digunakan ditunjukkan pada Tabel 4.3.

Tabel 4.3. Byte perintah komunikasi SPI

Perintah	Nilai	Deskripsi
SPICMD_SETADDR	0x20	Mengatur alamat register komunikasi SPI
SPICMD_READREG	0x6D	Menentukan SPI sebagai <i>read transaction</i>
SPICMP_WRITEREG	0x2D	Menentukan SPI sebagai <i>write transaction</i>
SPICMD_READFIFO	0x5F	Perintah untuk membaca FIFO

4.3.3 Pengujian Inisialisasi

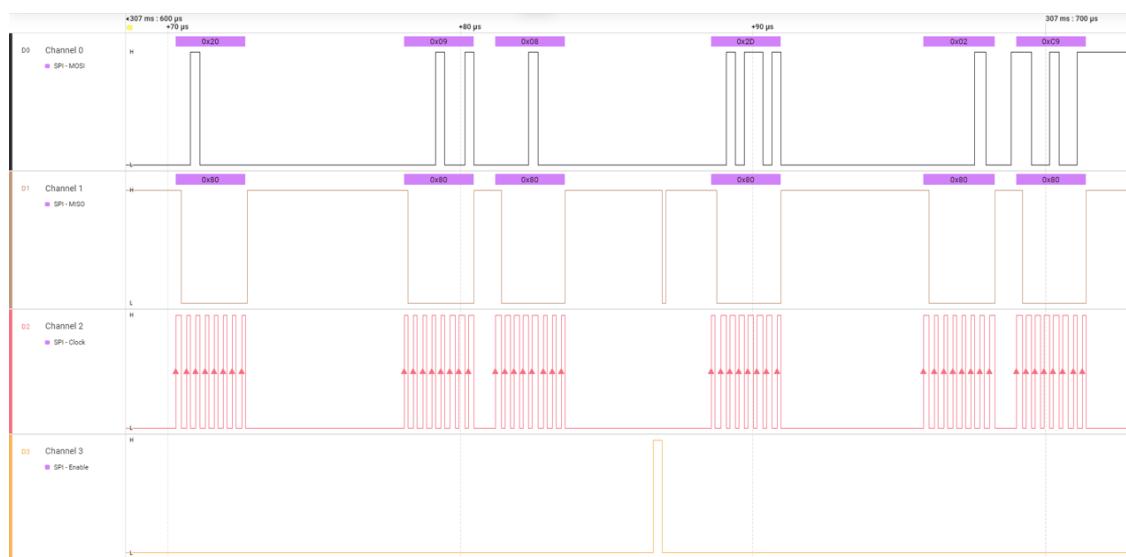
Berdasarkan *datasheet* IC AD5941 [16], hal pertama yang harus dilakukan adalah melakukan inisialisasi terhadap IC AD5941. Inisialisasi ini berfungsi untuk memberikan pengalaman pada IC AD5941 dalam melakukan transaksi melalui SPI. Hal ini diperlukan karena setiap kali IC AD5941 dimatikan atau di-RESET, alamat perangkat akan kembali menjadi 0x00. Proses inisialisasi dapat dimulai dengan mengirimkan byte perintah untuk mengatur alamat register dalam komunikasi SPI. Setelah mengirimkan byte

perintah SPICMD_SETADDR, terdapat urutan inisialisasi sistem yang harus diterapkan untuk memastikan pengoperasian perangkat AD5941 dengan benar. Tabel 4.4 menunjukkan alamat-alamat register serta data yang harus ditulis ke dalam register tersebut. Urutan inisialisasi ini harus dieksekusi dengan benar dan urut, jika tidak perangkat tidak akan berfungsi dengan baik.

Tabel 4.4. Perintah inisiasi SPI

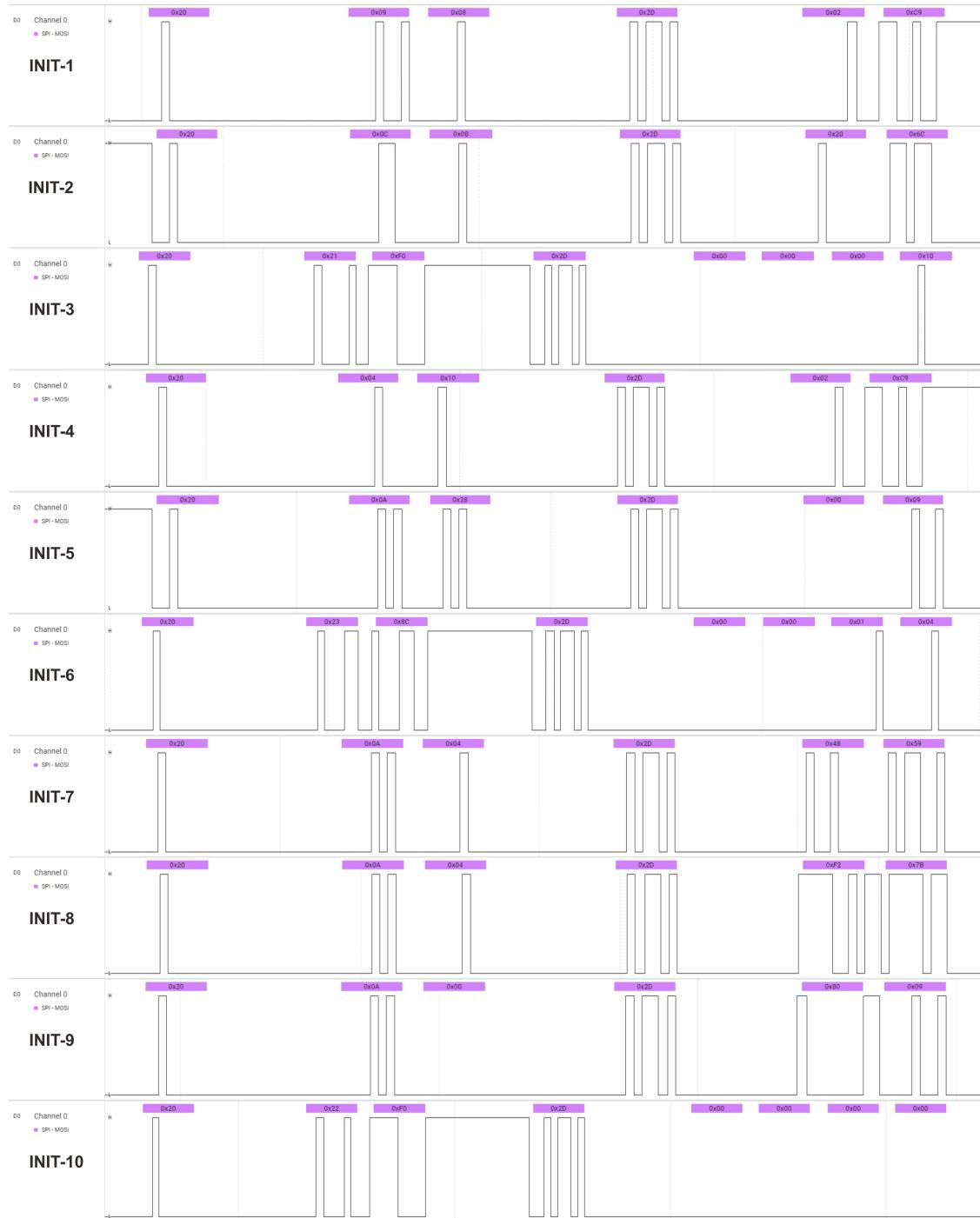
Label	Alamat Register	Data Inisiasi
INIT-1	0x0980	0x029C
INIT-2	0x0C08	0x206C
INIT-3	0x21F0	0x0010
INIT-4	0x0410	0x02C9
INIT-5	0x0A28	0x0009
INIT-6	0x238C	0x0104
INIT-7	0x0A04	0x4859
INIT-8	0x0A04	0xF27B
INIT-9	0x0A00	0x8009
INIT-10	0x22F0	0x0000

Setelah mengimplementasikan perintah inisialisasi tersebut ke dalam program mikrokontroler, maka dilakukan pengujian dengan menggunakan modul logic analyzer untuk melihat gelombang keluaran komunikasi SPI. Hasil pengujian pada inisialisasi INIT-1 ditampilkan pada Gambar 4.9.



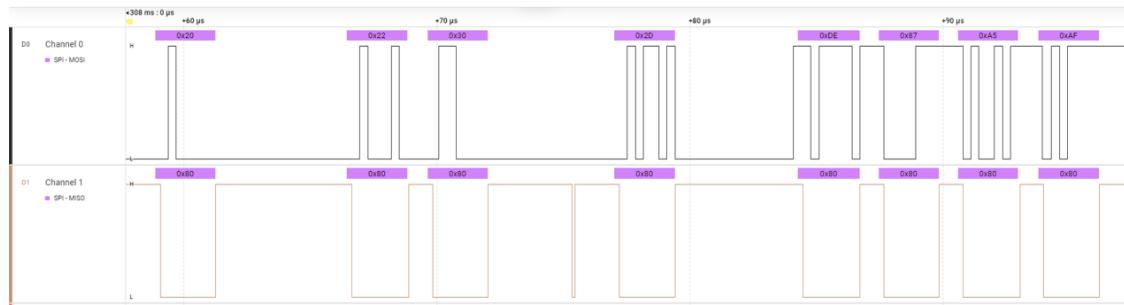
Gambar 4.9. Hasil pengujian inisialisasi sistem INIT-1 pada *software Logic*

Berdasarkan hasil yang ditunjukkan oleh Gambar 4.9, perintah pertama yang dikirim adalah SPICMD_SETADDR (0x20, diikuti oleh prosedur inisialisasi sistem dengan mengirim alamat register yang akan diberikan data (0x0908), diikuti oleh perintah SPI-CMD_WITEREG (0x2D), dan diikuti oleh pengiriman data (0x02C9) untuk mengisi register yang sebelumnya sudah ditentukan. Keseluruhan proses inisialisasi sistem ditampilkan pada Gambar 4.10.



Gambar 4.10. Hasil pengujian keseluruhan inisialisasi sistem pada *software Logic*

Proses selanjutnya adalah mengakses register untuk melakukan kalibrasi ADC dari IC AD5941. Perintah pertama yang dikirim adalah SPICMD_SETADDR (0x20) diikuti oleh alamat register kunci kalibrasi ADC CALDATLOCK (0x00002230). Kemudian, untuk mengakses register ini, diperlukan sebuah kata kunci yang tujuannya adalah untuk mencegah penimpaan data setelah proses kalibrasi. Sebelum mengirim kata kunci, dikirimkan perintah SPICMD_WRITEREG (0x2D) dan diikuti oleh kata kunci (0xDE87A5AF). Hasil pengujian ini ditampilkan pada Gambar 4.11.

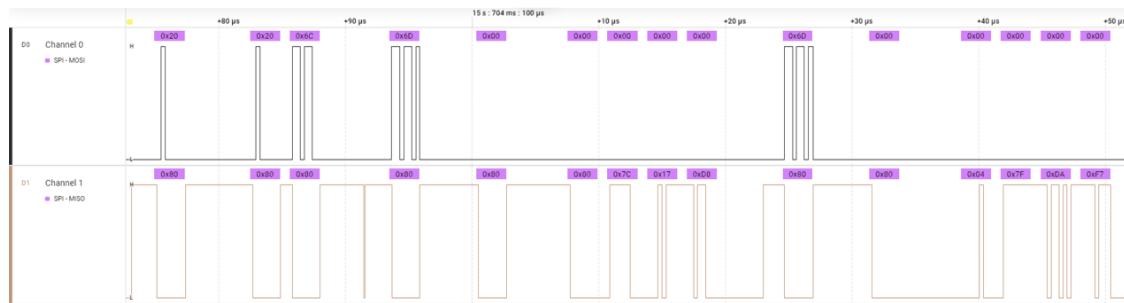


Gambar 4.11. Hasil pengujian perintah kalibrasi ADC *software Logic*

4.3.4 Pengujian Akses FIFO

Berdasarkan datasheet IC AD5941 [16], terdapat dua metode untuk mengakses data yang disimpan dalam FIFO. Metode pertama digunakan jika terdapat kurang dari tiga data pada FIFO sehingga proses membaca register DATAFIFORD dilakukan secara langsung. Metode kedua digunakan jika ada lebih dari tiga data pada FIFO sehingga diterapkan protokol transaksi SPI yang lebih efisien yaitu *fast FIFO read*. Pada penelitian ini hanya menggunakan dua data FIFO (riil dan imajiner) sehingga dilakukan pembacaan data langsung pada FIFO yang diuraikan sebagai berikut:

1. Mengirim 8-bit perintah SPICMD_SETADDR (0x20)
 2. Mengirim 16-bit alamat register DATAFIFORD (0x206C)
 3. Mengirim 8-bit perintah SPICMD_READREG (0x6D)
 4. Mengirim 8-bit *dummy byte* (0x00) untuk menginisiasi pembacaan register
 5. Menerima 32-bit data pertama dari register FIFO



Gambar 4.12. Hasil pengujian akses FIFO pada *software Logic*

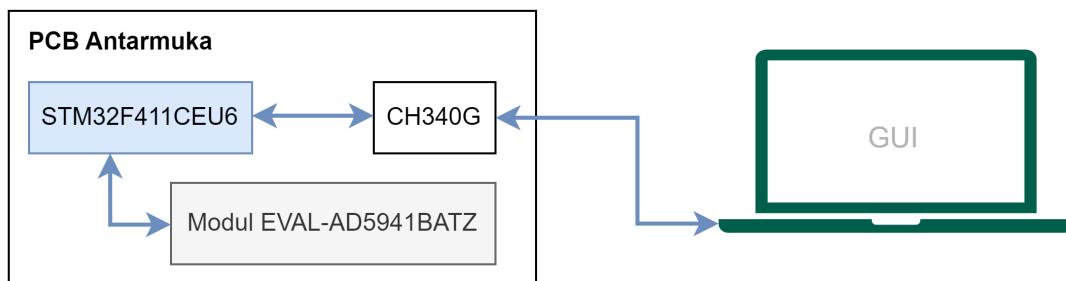
Hasil pengujian yang ditampilkan pada Gambar 4.12 menunjukkan bahwa proses akses data dari FIFO berjalan sesuai dengan protokol yang telah ditetapkan. Perintah pertama yang dikirim adalah SPICMD_SETADDR (0x20), diikuti oleh pengiriman 16-bit alamat register DATAFIFORD (0x206C), diikuti oleh perintah SPICMD_READREG (0x6D), pengiriman 8-bit *dummy byte*, dan diikuti oleh pengiriman 32-bit data (0x02C9) hasil pengukuran yang tersimpan dalam FIFO.

4.4 Pengujian GUI

Pengujian GUI dilakukan untuk memastikan bahwa GUI bekerja sesuai dengan perancangan dan memberikan pengalaman pengguna yang optimal. Pengujian ini dilakukan pada beberapa aspek utama yang diuraikan di bawah.

4.4.1 Skema Pengujian GUI

Pengujian ini dilakukan pada perangkat lunak GUI untuk memberi perintah ke perangkat keras PCB Antarmuka yang terhubung dengan Modul EVAL-AD5941BATZ dan STM32F411CEU6. Instalasi dan koneksi pengujian ditampilkan pada Gambar 4.13.



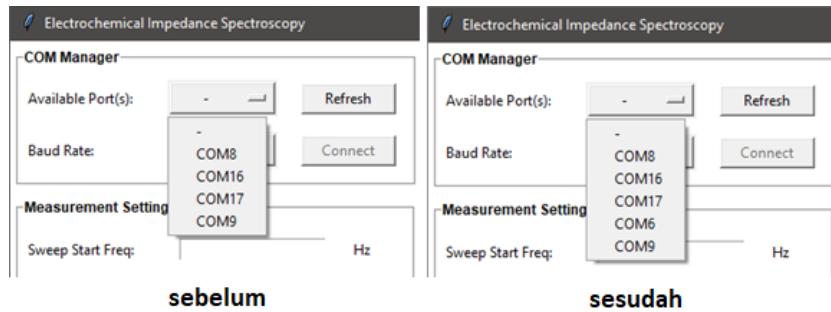
Gambar 4.13. Skema pengujian komunikasi GUI

4.4.2 Pengujian COM Manager

Pengujian COM dilakukan untuk memastikan bahwa GUI dapat menyegarkan daftar COM yang tersedia pada PC/Laptop dengan benar serta dapat melakukan sinkronisasi dengan perangkat STM32. Pengujian ini terdiri dari dua langkah utama:

1. *Refresh* Daftar COM pada Frame "COM Manager".

Pengujian dilakukan dengan menekan tombol "Refresh" pada frame "COM Manager". Gambar 4.14 menunjukkan bahwa daftar COM berhasil diperbarui dengan akurat sesuai dengan perubahan yang terjadi pada port COM komputer.



Gambar 4.14. Pengujian refresh COM Manager

2. Menghubungkan GUI dengan Perangkat STM32.

Setelah melakukan *refresh* daftar COM dan memilih COM yang sesuai, GUI dan perangkat dapat dihubungkan dengan menekan tombol "Connect". Kemudian, proses sinkronisasi dimulai dengan GUI mengirimkan perintah CONNECT ke STM32. Setelah perangkat tersinkronisasi, parameter pengukuran *default* secara otomatis ditampilkan pada frame "Measurement Info" GUI. Gambar 4.15 menampilkan proses pengujian COM Manager. Pengujian menunjukkan bahwa sinkronisasi berhasil dilakukan dan parameter *default* ditampilkan dengan benar.

```
[Running] python -u "d:\KULIAH\1. SKRIPSI\EIS_GUI\Master.py"
Refresh COM List
Available COMs:
['-', 'COM8', 'COM16', 'COM17', 'COM9']
Refresh COM List
Available COMs:
['-', 'COM8', 'COM16', 'COM17', 'COM6', 'COM9']
Successful UART Connection Using COM6
Synchronizing Serial...
[ '=', '\n' ]
[ 'P', '1', '50000', '50', '\n' ]
[ 'P', '1', '50000', '50', '\n' ]
[ '=', '\n' ]
Synchronizing Done...
```

Gambar 4.15. Proses pengujian COM Manager pada *software* VSCode

4.4.3 Pengujian Pengaturan Parameter Pengukuran

Pengujian pengaturan parameter pengukuran bertujuan untuk memastikan bahwa nilai parameter dapat diisi dan disinkronkan dengan perangkat melalui GUI.

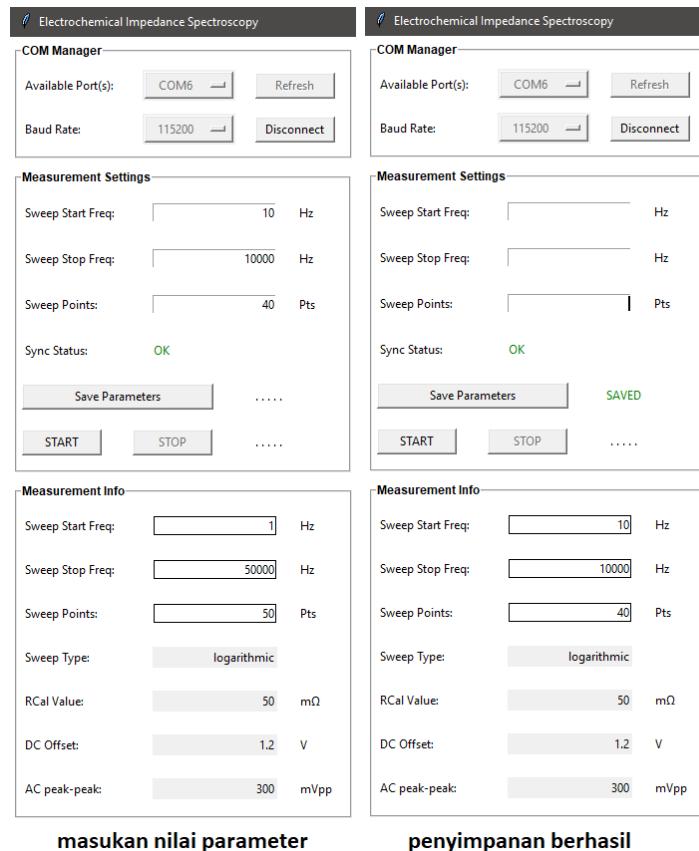
Hal pertama dalam pengujian ini adalah mengisi nilai parameter pada frame "Measurement Settings". Nilai yang dimasukkan pada kotak parameter dibatasi sehingga hanya angka yang dapat ditampilkan dan diterima sebagai nilai akhir. Nilai "Sweep Start Freq" dibatasi hingga 9999, nilai "Sweep Stop Freq" dibatasi hingga 200.000, dan nilai "Sweep Points" dibatasi hingga 999. Selanjutnya, dilakukan sinkronisasi dan pengiriman data ke perangkat dengan menekan tombol "Save Parameters" pada GUI. Selama proses sinkronisasi, PC/Laptop akan terus mengirim perintah WRITE_PARAM sampai mene-

rima respon CMD_OK dari perangkat. Kemudian PC/Laptop akan mengirimkan data parameter sesuai yang diisi dan jika perangkat berhasil menerima data, perangkat akan memberi konfirmasi berupa nilai parameter yang diterima. Gambar 4.16 menampilkan bahwa prosedur ini berhasil dikirim dan diterima oleh perangkat.

```
Synchronizing...
['=','\n']
['=','\n']
['=','\n']
['=','\n']
['=','\n']
['=','\n']
['=','\n']
['K','\n']
Sending Param...
['10','\n']
['10000','\n']
['40','\n']
['=','\n']
Successfully Saving All Measurement Parameters...
```

Gambar 4.16. Pembaharuan nilai parameter pada *software* VSCode

Setelah konfirmasi diterima, parameter yang telah disinkronkan ditampilkan pada frame "Measurement Info". Gambar 4.17 menunjukkan bahwa parameter ditampilkan dengan benar dan sesuai dengan yang diinputkan pada GUI.



Gambar 4.17. Hasil pengujian penyimpanan parameter pada tampilan GUI

4.4.4 Pengujian Pengukuran

Pengujian pengukuran memastikan bahwa GUI dapat menginisiasi pengukuran dan menerima hasil pengukuran dari perangkat dengan benar.

Pengujian pertama adalah sinkronisasi dengan perangkat. Setelah pengguna menekan tombol “START” pada frame “Measurement Settings”, perintah MEASURE_INIT dikirimkan dari PC/Laptop ke perangkat untuk melakukan sinkronisasi data. Selama proses sinkronisasi, PC/Laptop akan terus menerus mengirim perintah MEASURE_INIT sampai menerima respon CMD_OK dari perangkat. Perangkat mengirimkan respon RCAL_INIT bahwa pengukuran akan dimulai. Kemudian, perangkat mengirimkan hasil pengukuran resistor kalibrasi terlebih dahulu dengan format RCAL_DATA, diikuti dengan hasil pengukuran impedansi dengan format IMP_DATA. Gambar 4.18 menampilkan proses hasil pengukuran dan penerimaan data pada GUI.

```
Synchronizing...
['=', '\n']
['K', '\n']
['RC', '\n']
Initiating RCal Measurement Procedure...
['R', '10.0', '19.0', '6.0', '\n']
['R', '11.9', '-45.0', '-10.0', '\n']
['R', '14.3', '-14.0', '-12.0', '\n']
['R', '17.0', '1.0', '20.0', '\n']
['R', '20.3', '28.0', '-41.0', '\n']
['R', '24.2', '-9.0', '-7.0', '\n']

['=', '\n']
['IM', '\n']
['I', '10.0', '-5.4', '-8.8', '\n']
['I', '11.9', '-48.5', '-8.1', '\n']
['I', '14.3', '-51.5', '-5.9', '\n']
['I', '17.0', '-47.3', '-4.9', '\n']
['I', '20.3', '-49.4', '0.8', '\n']
['I', '24.2', '278.5', '122.3', '\n']
```

Gambar 4.18. Proses pengujian pengukuran pada *software* VSCode

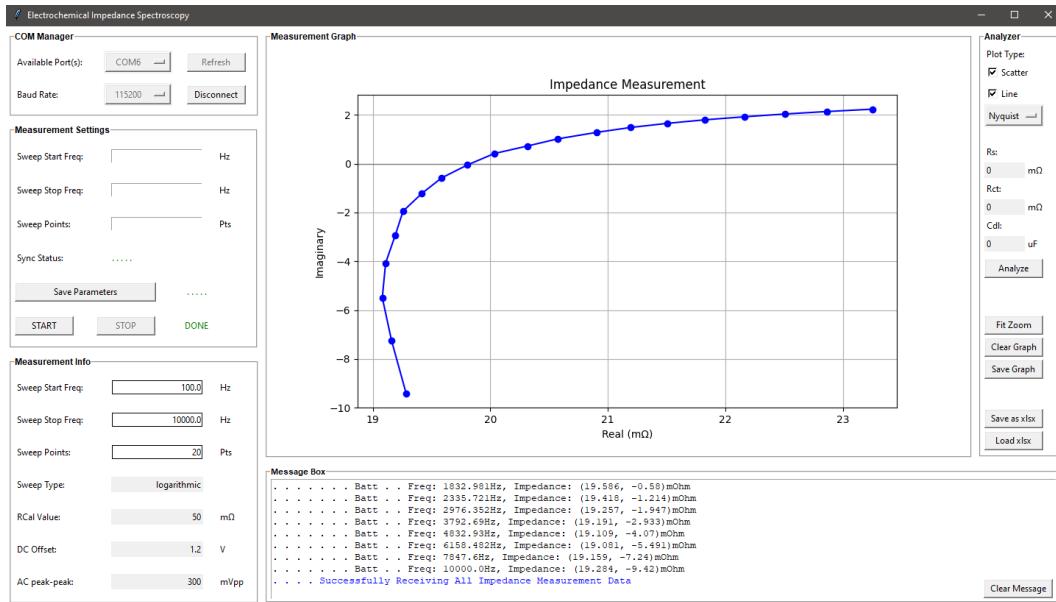
4.4.5 Pengujian Tampilan Plot

Pengujian tampilan plot bertujuan untuk memastikan bahwa GUI dapat menampilkan data pengukuran dalam bentuk plot dengan benar. Terdapat dua sumber data yang dapat diplot:

1. Hasil Pengukuran Langsung

Data yang diperoleh dari pengukuran langsung oleh STM32 diplot pada GUI. Pengujian menunjukkan bahwa plot ditampilkan dengan akurat dan memberikan vi-

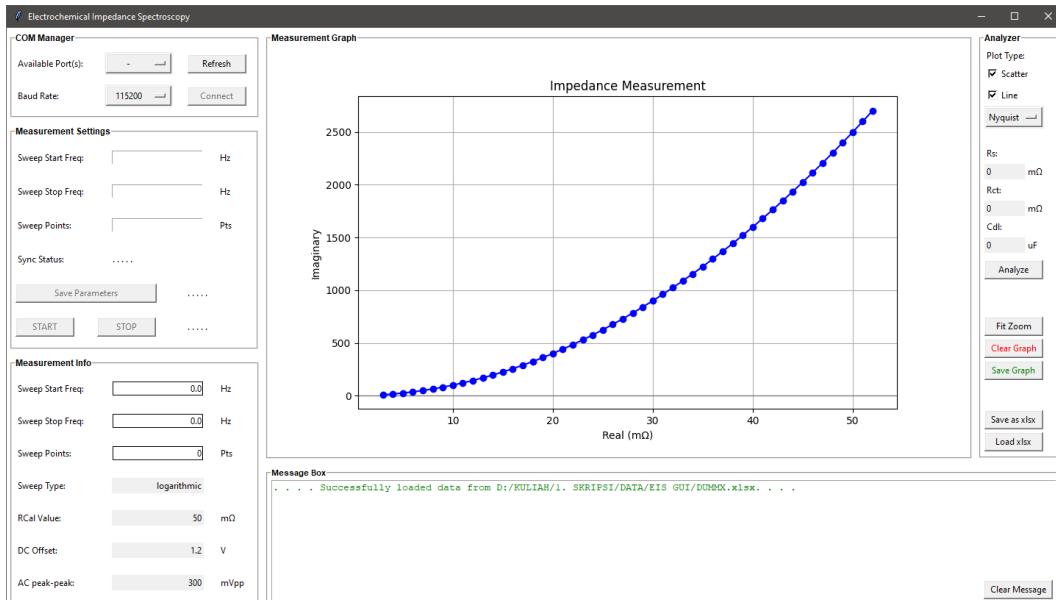
sualisasi yang jelas dari hasil pengukuran. Gambar 4.19 menampilkan tampilan GUI setelah berhasil menampilkan plot hasil pengukuran impedansi.



Gambar 4.19. Tampilan plot hasil pengukuran pada GUI

2. Memuat File Hasil Penyimpanan

GUI juga mampu memplot data dari *file* pengukuran sebelumnya yang disimpan atau *file* dengan format data yang sama. Gambar 4.20 menampilkan tampilan GUI setelah berhasil menampilkan plot hasil memuat *file* Excel.



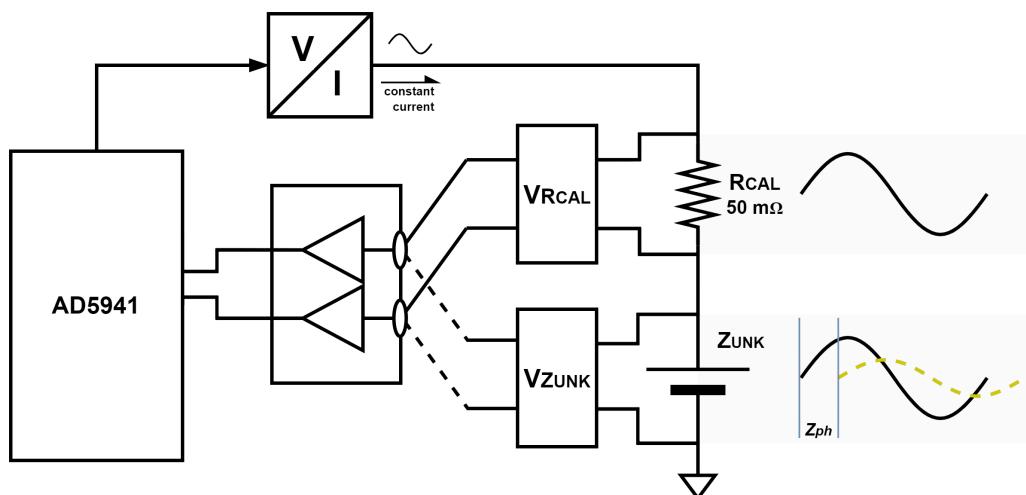
Gambar 4.20. Tampilan plot pada GUI dengan memuat *file* .xlsx

4.5 Pengujian Fungsi Pengukuran Impedansi

Pengujian ini bertujuan untuk mengevaluasi fungsi peranti pengukuran impedansi baterai litium-ion yang dikembangkan, dengan mengoperasikan sistem secara menyeluruh termasuk PCB Antarmuka, Modul EVAL-AD5941BATZ, dan GUI.

4.5.1 Perhitungan Impedansi Hasil EIS

Pengukuran impedansi menggunakan metode EIS memiliki sifat yang rasioometrik. Untuk mengukur impedansi yang tidak diketahui nilainya (Z_{UNK}), isyarat eksitasi AC diberikan pada resistor yang diketahui nilai hambatannya (resistor kalibrasi, R_{CAL}) dan tegangan responsnya ($V_{R_{CAL}}$) diukur. Selanjutnya, isyarat yang sama diberikan pada obyek yang impedansinya tidak diketahui (Z_{UNK}), dalam hal ini adalah baterai litium-ion, dan tegangan responnya ($V_{Z_{UNK}}$) diukur. Gambar 4.21 menunjukkan proses EIS dan parameter yang diukur.



Gambar 4.21. Ilustrasi proses pengukuran impedansi

Proses pertama, yaitu pengukuran respon tegangan R_{CAL} , penting untuk dilakukan sebelum melakukan pengukuran impedansi baterai. Besarnya tegangan respon pada resistor R_{CAL} dapat diwakili oleh Persamaan 4-1 [19].

$$V_{R_{CAL}} = \sqrt{{R_{R_{CAL}}}^2 + {X_{R_{CAL}}}^2} \cdot I_{R_{CAL}} \sin(2\pi f_0 t + \phi_0) \quad (4-1)$$

Di mana $R_{R_{CAL}}$ dan $X_{R_{CAL}}$ adalah komponen riil dan imajiner dari R_{CAL} , sedangkan $I_{R_{CAL}}$, f_0 , dan ϕ_0 adalah amplitudo, frekuensi, dan fase dari arus isyarat eksitasi yang diberikan. Selain itu, fase dari $V_{R_{CAL}}$, yaitu ϕ_0 , sama dengan fase dari arus isyarat eksitasi karena tegangan respon pada R_{CAL} sefase dengan arus eksitasi.

Setelah proses pertama selesai dilakukan untuk setiap sapuan frekuensi, isyarat eksitasi yang sama diberikan dan besar tegangan respon baterai (V_{UNK}) yang melintasi baterai (Z_{UNK}) dapat dinyatakan dengan Persamaan 4-2 [19].

$$V_{Z_{UNK}} = \sqrt{{R_{Z_{UNK}}}^2 + {X_{Z_{UNK}}}^2} \cdot I \sin(2\pi ft + \phi_1) \quad (4-2)$$

Di mana $R_{Z_{UNK}}$ dan $X_{Z_{UNK}}$ adalah komponen riil dan imajiner dari Z_{UNK} dan ϕ_1 adalah pergeseran fase antara isyarat eksitasi dan tegangan respon karena komponen elektrokimia internal baterai. Menggabungkan Persamaan 4-1 dan 4-2, fase dari baterai litium-ion yang diukur pada frekuensi f_0 dinyatakan dengan Persamaan 4-3.

$$Z_{ph} = \phi_1 - \phi_0 \quad (4-3)$$

Setelah didapatkan nilai $V_{R_{CAL}}$ dan nilai $V_{Z_{UNK}}$, maka nilai impedansi baterai yang tidak diketahui dapat dihitung menggunakan Persamaan 4-4 [9], [19].

$$Z_{UNK} = \frac{V_{Z_{UNK}}}{V_{R_{CAL}}} \cdot R_{CAL} \quad (4-4)$$

Komponen riil dan imajiner dari Z_{UNK} pada frekuensi f_0 dapat dipisahkan dengan menghitung magnitude dan fasenya. Hal ini dinyatakan pada Persamaan 4-5 dan 4-6.

$$R_{Z_{UNK}} = |Z_{UNK}| \cdot \cos(Z_{ph}) \quad (4-5)$$

$$X_{Z_{UNK}} = |Z_{UNK}| \cdot \sin(Z_{ph}) \quad (4-6)$$

Proses pengukuran tegangan respon ini dilakukan secara otomatis melalui fitur DFT *hardware accelerator engine* dalam IC AD5941. Hasil ini tersimpan dalam register ADC pada alamat DFTREAL dan DFTIMAG [16] yang juga ditempatkan pada register FIFO untuk akses cepat data register.

4.5.2 Pengaturan Parameter Pengukuran

Pengujian bertujuan untuk mencoba rentang sapuan frekuensi yang dapat dihasilkan oleh IC AD5941 dan mengukur impedansi baterai pada setiap nilai frekuensi tersebut. Dalam *datasheet* IC AD5941 [16], disebutkan bahwa rentang frekuensi yang dapat dihasilkan adalah 0,015 Hz sampai dengan 200 kHz. Namun, pada penelitian ini dibatasi hanya dari 0,1 Hz sampai dengan 200 kHz untuk menghindari ketidakpastian pengukuran pada frekuensi rendah dan waktu pengukuran yang terlalu lama. Terlebih, karakterisasi

komponen impedansi seringkali hanya dilakukan pada rentang 0,1 Hz sampai 10 kHz karena pada rentang ini sudah didapatkan parameter yang diperlukan.

Pengukuran EIS memiliki beberapa parameter yang harus diatur nilainya seperti amplitudo isyarat eksitasi dan DC *offset* untuk rangkaian penguat dan tapis. Terdapat parameter tetap dan parameter yang dapat dikonfigurasi melalui GUI yang diuraikan sebagai berikut.

Parameter tetap:

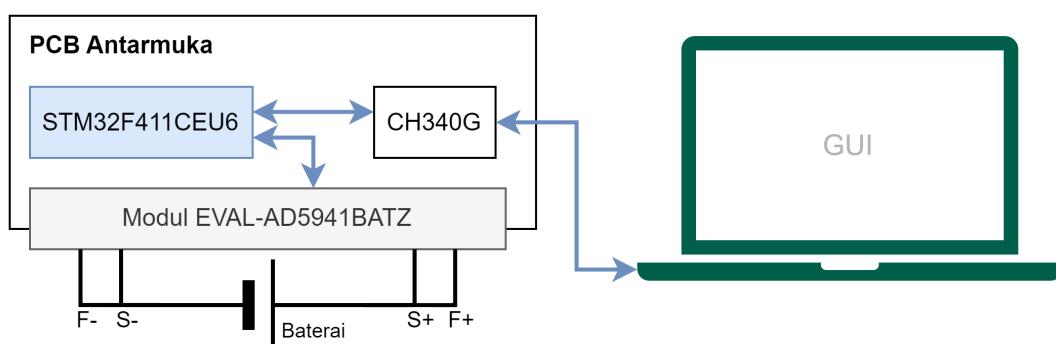
- *Sweep Type* = *Logarithmic*
- *DC Offset* = 1.200 mV
- *AC Amplitude* = 300 mVpp

Parameter yang dapat dikonfigurasi:

- *Start Frequency* = 0,1 Hz
- *Stop Frequency* = 200.000 Hz
- *Sweep Points* = 30

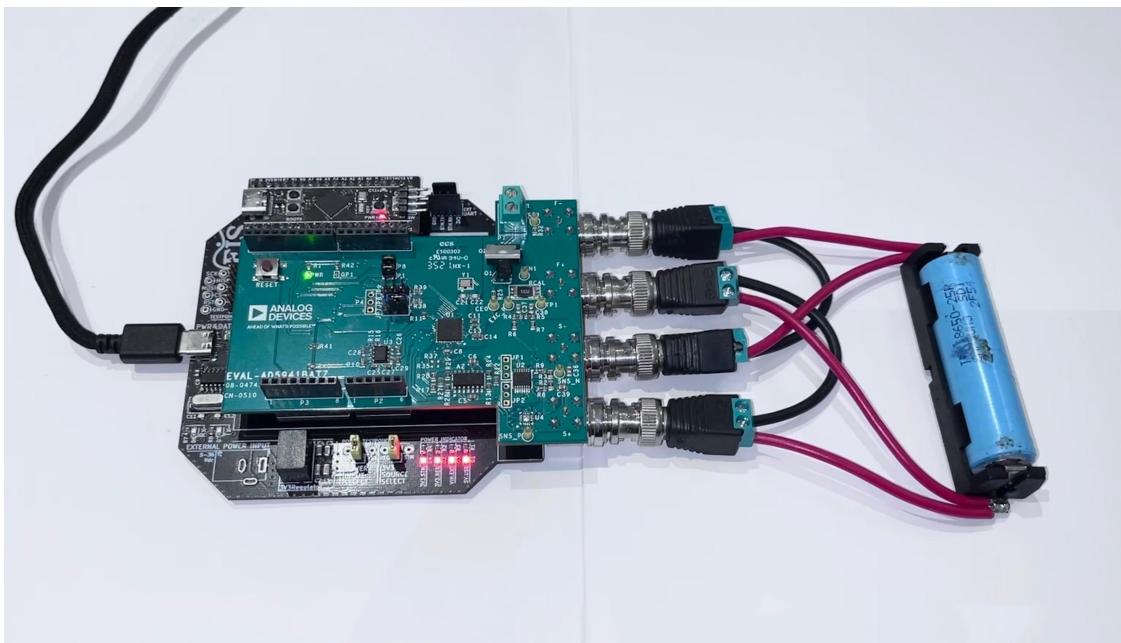
4.5.3 Skema dan Susunan Peranti Pengukuran

Pengujian peranti pengukuran impedansi dilakukan sesuai dengan parameter yang telah dikonfigurasi sebelumnya. Perangkat dihubungkan ke laptop menggunakan USB tipe C, tanpa sumber daya eksternal. Koneksi ke baterai dilakukan menggunakan metode empat kabel, yang merupakan teknik yang umum digunakan untuk mengukur impedansi secara akurat. Koneksi ini dibuat sendiri dengan menyolder kabel ke *holder* baterai, meminimalisir resistansi kontak tambahan sehingga hasil yang diperoleh lebih akurat. Skema penyusunan dalam pengujian ditampilkan pada Gambar 4.22.



Gambar 4.22. Skema pengujian komunikasi SPI

Susunan perangkat keras dalam pengujian ini ditampilkan pada Gambar 4.23.



Gambar 4.23. Susunan perangkat dalam pengujian

4.5.4 Baterai Objek Pengukuran

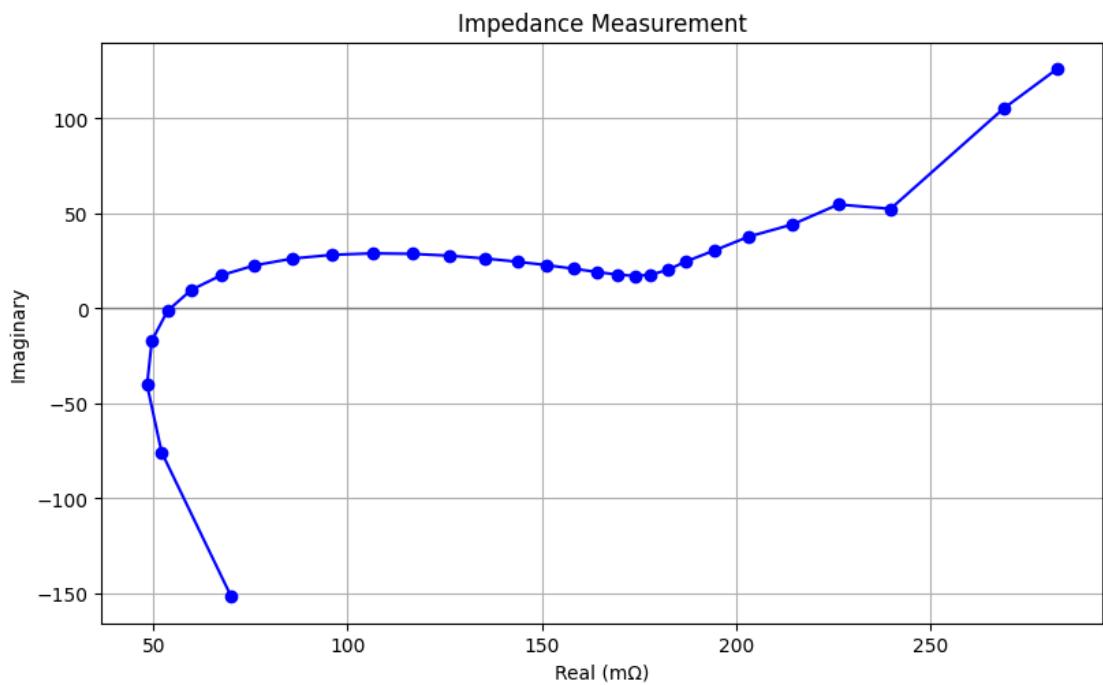
Pengukuran impedansi dalam pengujian ini dilakukan pada baterai dengan spesifikasi baterai yang ditampilkan pada Tabel 4.5 [20].

Tabel 4.5. Spesifikasi baterai pengujian peranti pengukuran

Parameter	Samsung INR18650-25R2
Tegangan Baterai (V)	3,621
Tegangan Nominal (V)	3,600
Kapasitas Nominal (mAh)	2500
Bahan Katoda	NCA

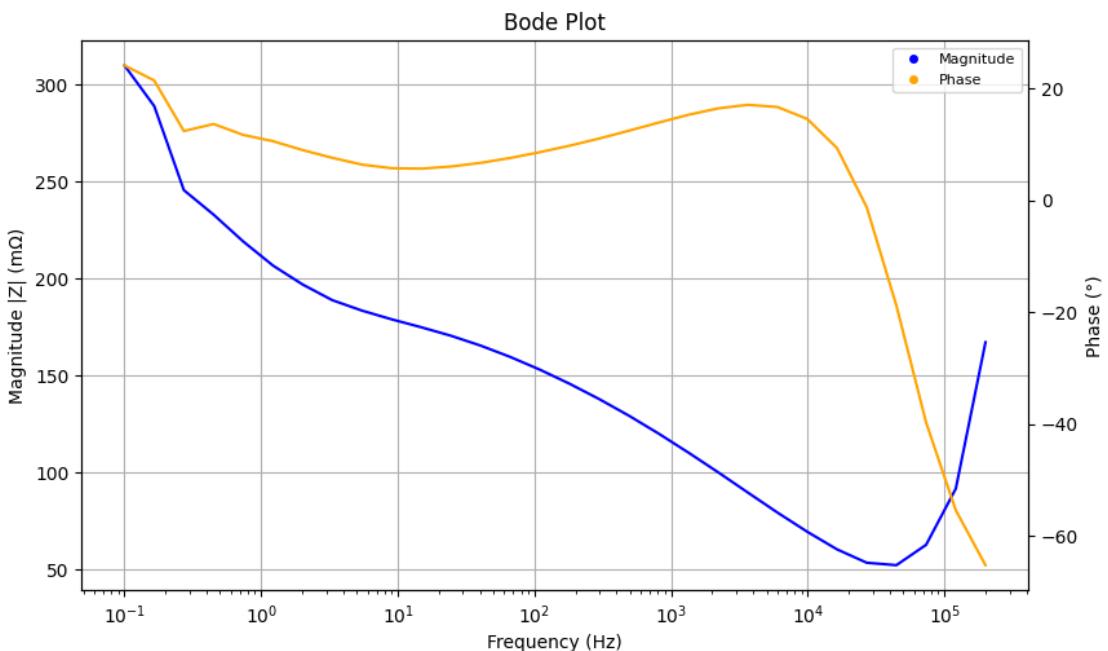
4.5.5 Hasil Pengujian

Hasil pengukuran ditampilkan pada Gambar 4.24 dalam bentuk plot Nyquist yang didapatkan dari fitur GUI untuk menyimpan plot dalam bentuk gambar. Plot menunjukkan bahwa pengukuran berhasil dilakukan dari rentang frekuensi rendah yang dimulai dari sisi kanan atas plot hingga frekuensi tinggi yang berakhir di sisi bawah kiri plot.



Gambar 4.24. Plot Nyquist pengukuran baterai Samsung INR18650-25R2

Hasil pengukuran juga dapat ditampilkan dalam bentuk plot Bode seperti pada Gambar 4.25 yang didapatkan dari fitur GUI untuk menyimpan plot dalam bentuk gambar. Opsi ini dapat membantu proses karakterisasi dari baterai jika dibutuhkan.



Gambar 4.25. Plot Bode pengukuran baterai Samsung INR18650-25R2

Hasil pengujian menunjukkan bahwa perangkat dapat melakukan pengukuran impedansi sesuai dengan parameter yang ditentukan. Nilai impedansi hasil pengukuran dapat ditampilkan dalam GUI berupa plot Nyquist dan plot Bode yang karakteristiknya sesuai dengan dasar teori EIS yang telah dibahas sebelumnya.

4.6 Pengujian Akurasi Pengukuran dan Karakterisasi Baterai Litium-ion

Pengujian ini bertujuan untuk mengevaluasi akurasi peranti pengukuran impedansi baterai litium-ion yang dikembangkan. Hasil pengukuran kemudian dikarakterisasi berdasarkan plot Nyquist dari ECM untuk mendapatkan parameter internal baterai. Setiap nilai tersebut kemudian dihitung besar simpangannya untuk mendapatkan nilai akurasi pengukuran.

4.6.1 Metode dan Objek Pengujian

Pengujian dilaksanakan dengan mengukur dan membandingkan hasil pengukuran tiga sampel baterai: baterai baru, baterai bekas yang masih baik, dan baterai bekas yang sudah tidak layak digunakan. Setiap baterai diukur impedansinya sebanyak 10 kali untuk mendapatkan data yang akurat dan mengurangi variabilitas hasil. Pengujian dilakukan dalam kondisi suhu ruangan yang konstan pada 21°C dan setiap baterai diisi hingga mencapai tegangan nominalnya dengan batas sebesar $\pm 5\%$. Data yang diperoleh dari setiap pengukuran mencakup frekuensi dan impedansi (resistansi dan reaktansi) pada setiap titik frekuensi yang ditentukan. Parameter umum dari baterai yang digunakan sebagai objek pengukuran ditunjukkan oleh Tabel 4.6 [20].

Tabel 4.6. Parameter baterai objek pengujian

Parameter	Samsung	Samsung	Samsung
	INR-25R2	INR-25R2	INR-25R2
	Baru	Bekas Baru	Bekas Tua
Tegangan Baterai (V)	3,604	3,595	3,592
Tegangan Nominal (V)	3,600	3,600	3,600
Tegangan <i>Cut-off</i> (V)	2,500	2,500	2,500
Kapasitas Nominal (mAh)	2500	2500	2500
Bahan Katoda	NCA	NCA	NCA
Impedansi Internal ($\text{m}\Omega$)	≤ 18	≤ 18	≤ 18

4.6.2 Pengaturan Parameter Pengukuran

Pengujian ini akan menerapkan parameter pengukuran yang sama pada setiap sampel baterai. Parameter tersebut adalah parameter yang dapat diatur pada GUI yaitu frekuensi sapuan dan jumlah titik nilai, nilai parameter diuraikan sebagai berikut:

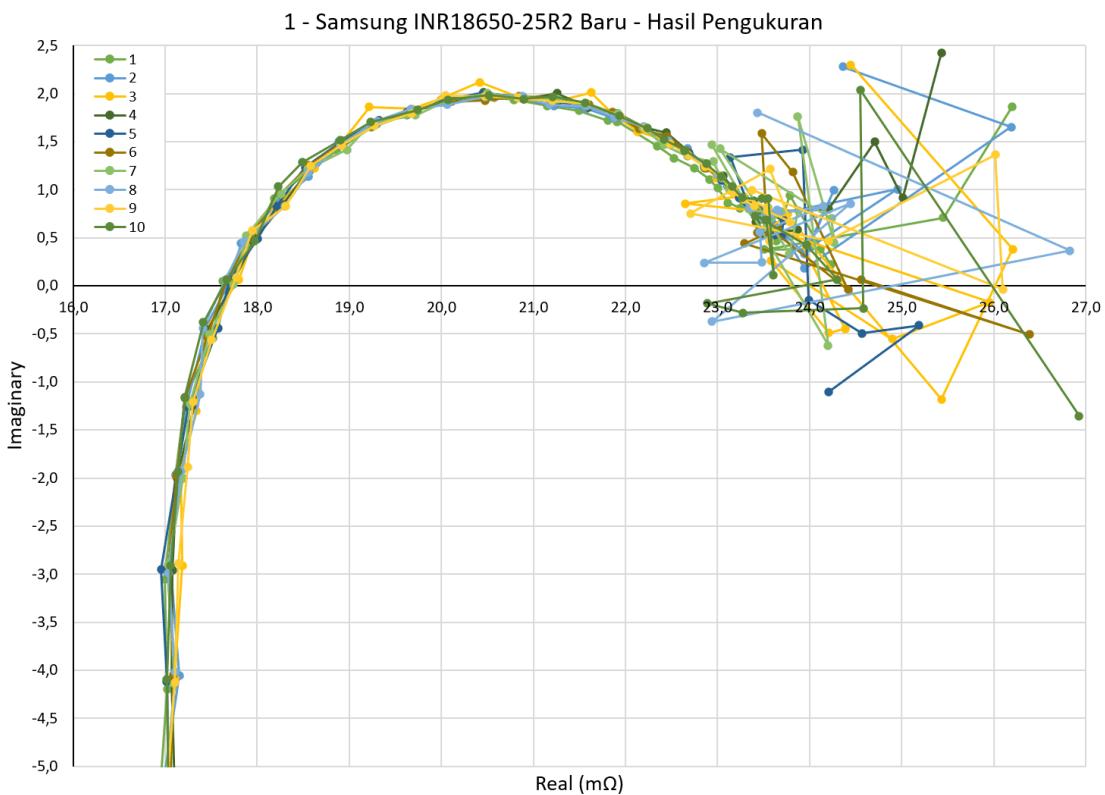
- *Start Frequency* = 1 Hz
- *Stop Frequency* = 10.000 Hz
- *Sweep Points* = 40

Pengujian ini dibatasi pada rentang frekuensi 1 Hz sampai dengan 10 kHz untuk mendapatkan tiga karakteristik baterai litium-ion yaitu R_S , R_{CT} , dan C_{DL} . Ketiga parameter ini pasti dijumpai pada rentang frekuensi ini dan akan membentuk plot Nyquist yang menyerupai Gambar 2.6. Nilai resistansi Warburg tidak diikutsertakan karena berada pada rentang frekuensi kurang dari 1 Hz yang menyebabkan pengukuran terlalu lama. Nilai ini juga membutuhkan analisis yang lebih mendalam dan melibatkan konstanta Warburg yang dipengaruhi oleh bahan penyusun katoda [19].

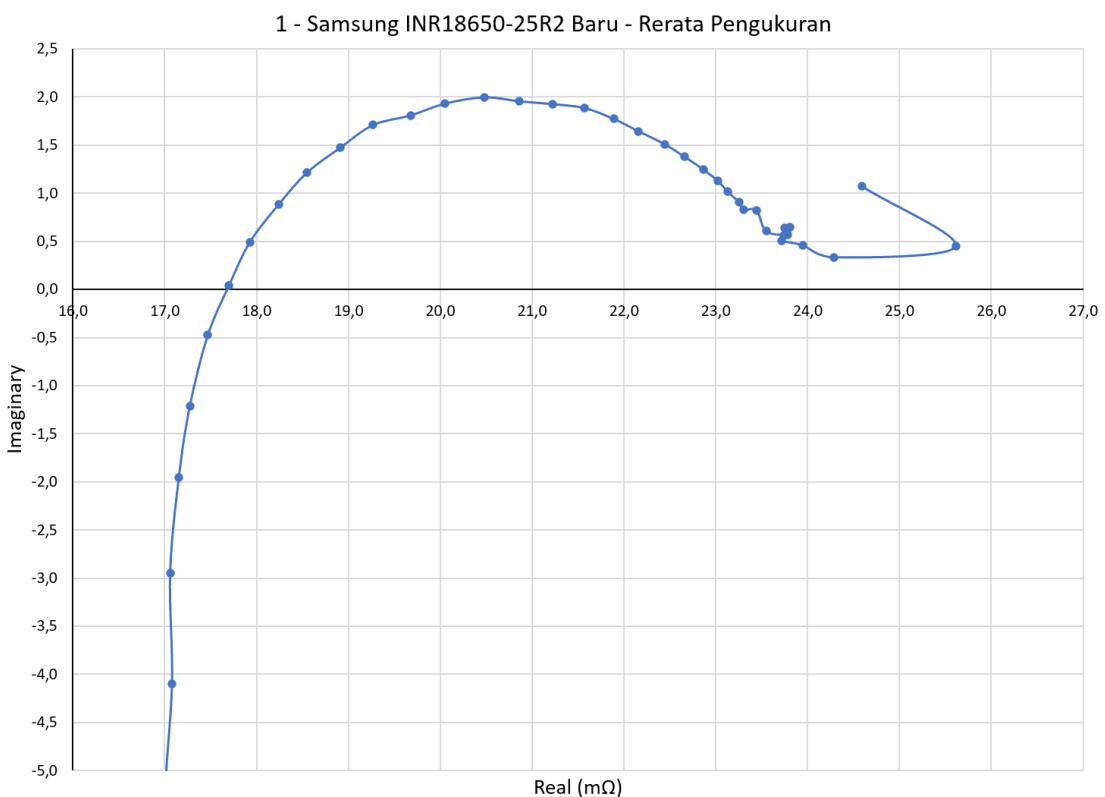
4.6.3 Hasil Pengukuran

Pengukuran impedansi dilakukan melalui GUI dan hasil pengukuran impedansi setiap baterai kemudian disimpan ke dalam *file* Excel. Sepuluh hasil pengukuran sampel baterai diambil dan diplot untuk memudahkan visualisasi dan perbandingan antar iterasi pengukuran. Gambar 4.26, Gambar 4.28, dan Gambar 4.30 menunjukkan hasil pengukuran impedansi untuk setiap sampel baterai, memungkinkan kita mengamati performa masing-masing baterai dalam sepuluh pengukuran yang berbeda. Proses *plotting* ini membantu dalam melihat tren dan variasi yang terjadi pada masing-masing baterai dalam berbagai iterasi pengukuran.

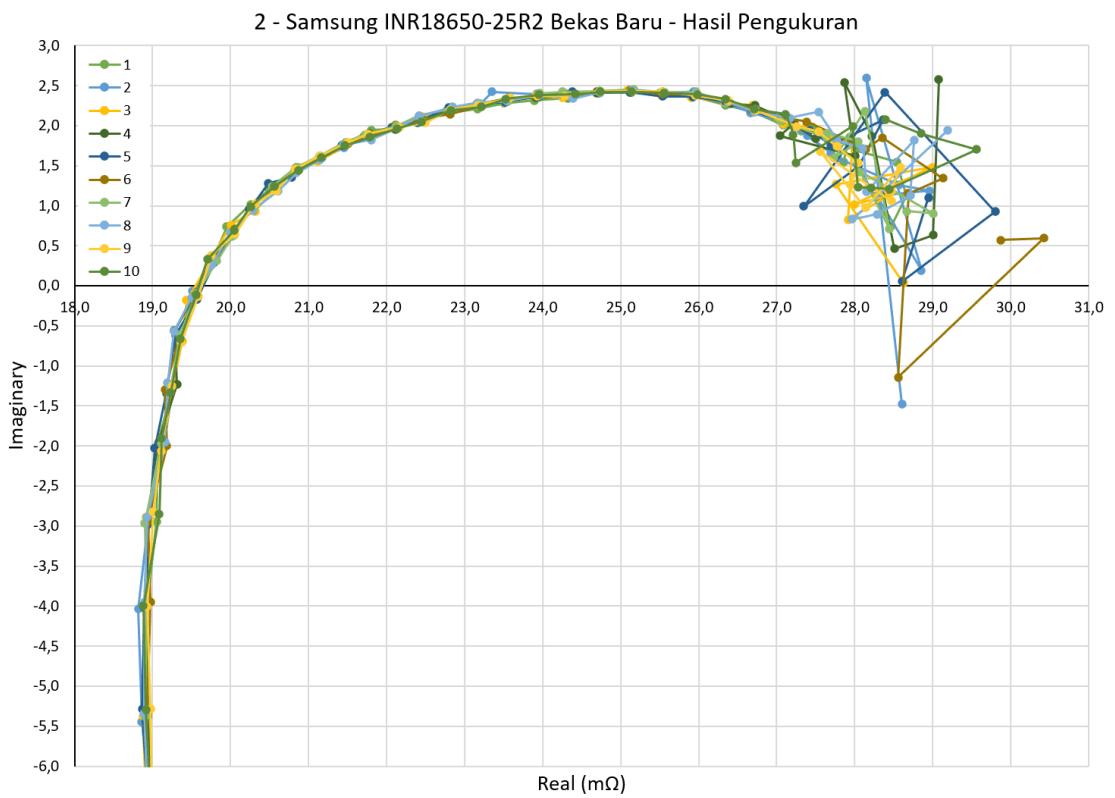
Setelah melakukan pengukuran dan *plotting* data, rerata dari kesepuluh pengukuran dihitung menggunakan metode rerata titik koordinat untuk memperoleh nilai rerata yang akurat dan representatif. Hasil rerata ini ditampilkan pada Gambar 4.27, Gambar 4.29, dan Gambar 4.31, menunjukkan nilai rerata impedansi untuk setiap sampel baterai setelah sepuluh kali pengukuran. Analisis hasil rerata ini memberikan gambaran stabil hasil pengukuran impedansi baterai, memungkinkan evaluasi dan karakterisasi masing-masing baterai..



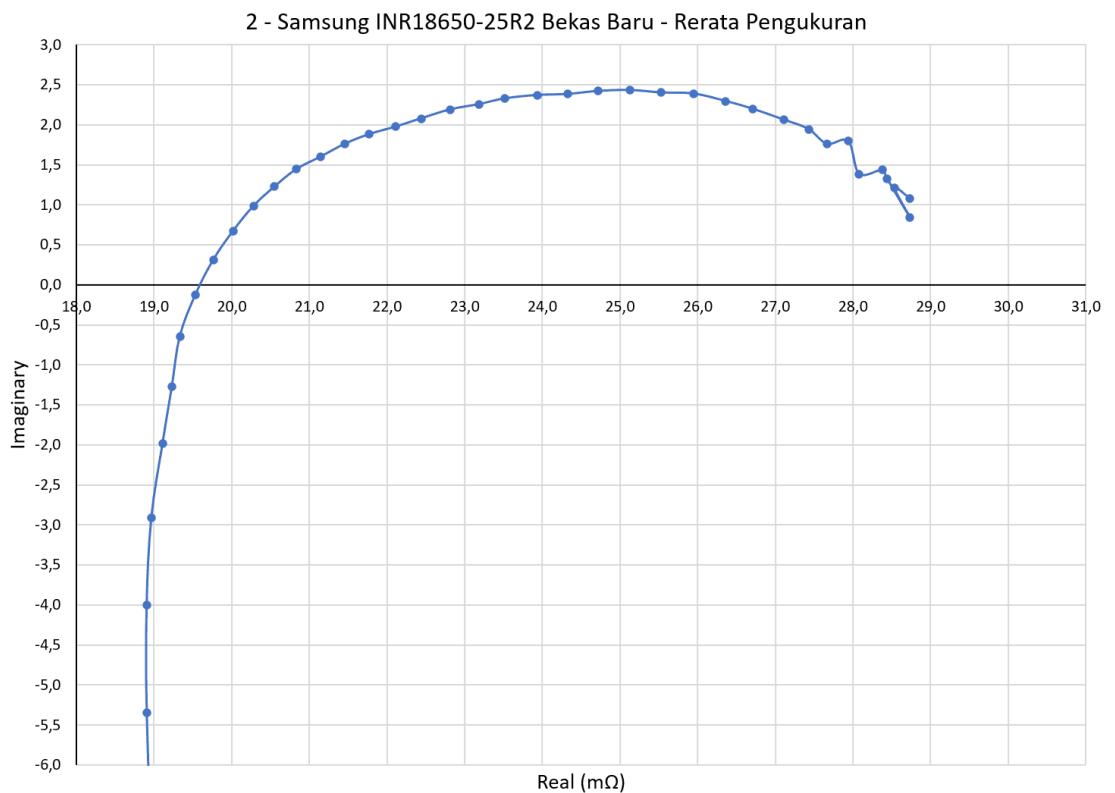
Gambar 4.26. Plot hasil pengukuran baterai Samsung INR-25R2 Baru



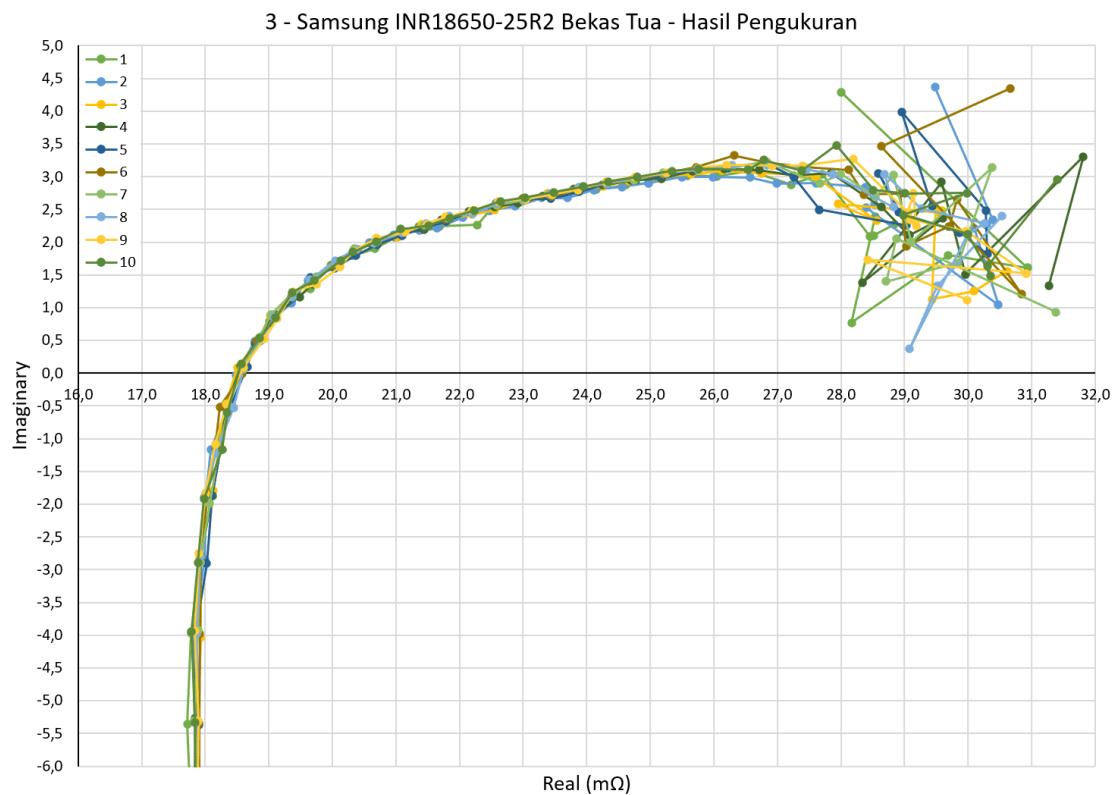
Gambar 4.27. Plot rerata hasil pengukuran baterai Samsung INR-25R2 Baru



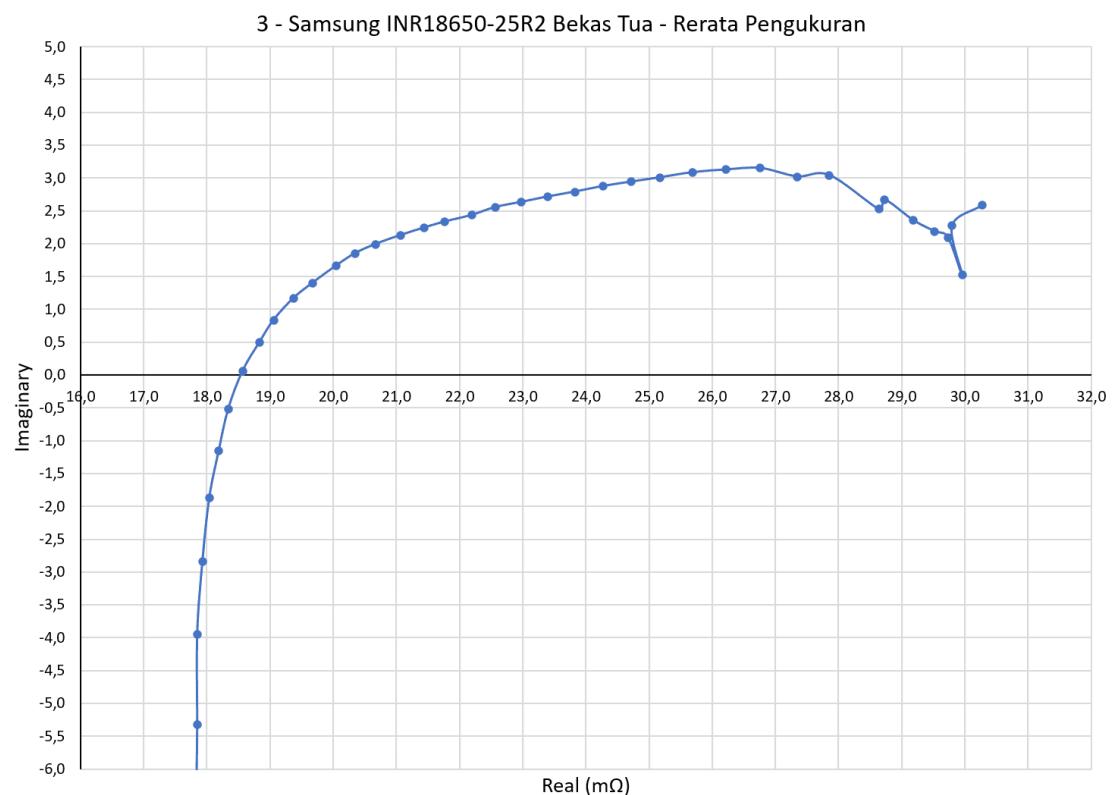
Gambar 4.28. Plot hasil pengukuran baterai Samsung INR-25R2 Bekas Baru



Gambar 4.29. Plot rerata hasil pengukuran baterai Samsung INR-25R2 Bekas Baru



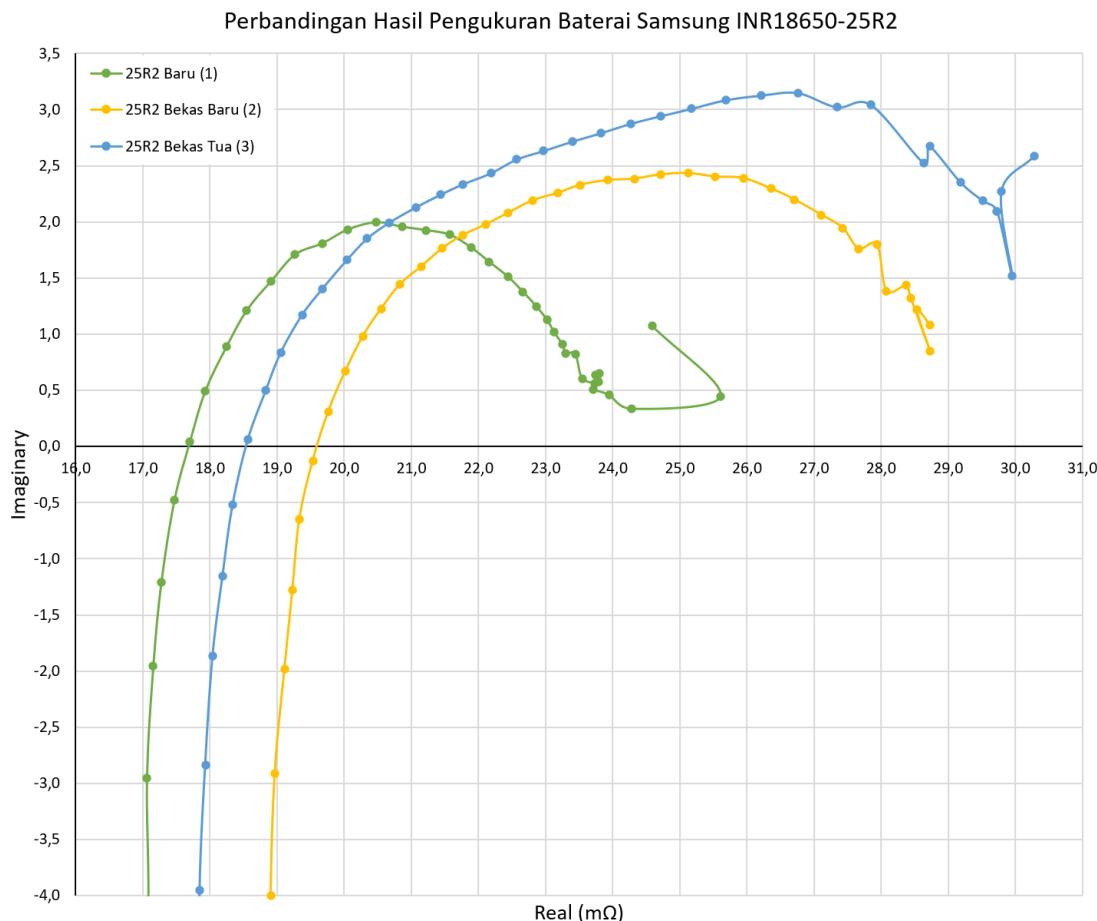
Gambar 4.30. Plot hasil pengukuran baterai Samsung INR-25R2 Bekas Tua



Gambar 4.31. Plot rerata hasil pengukuran baterai Samsung INR-25R2 Bekas Tua

Hasil pengukuran menunjukkan bahwa pada frekuensi rendah, terdapat nilai yang cukup inkonsisten antar pengukuran. Hal ini sangat terlihat pada baterai INR-25R2 baru, seperti ditunjukkan pada Gambar 4.26. Jika dibandingkan dengan ECM Randles sederhana, plot dari baterai ini memiliki bentuk setengah lingkaran yang sempurna seperti pada Gambar 2.8, dan model ini tidak memiliki komponen Warburg. Oleh karena itu, dapat diasumsikan bahwa nilai impedansi pada frekuensi rendah yang menyebar pada area pangkal setengah lingkaran merupakan karakteristik baterai baru.

Hasil pengukuran untuk baterai INR-25R2 bekas baru dan bekas tua menunjukkan bentuk plot setengah lingkaran yang sedikit memanjang. Walaupun pengukuran tidak dilakukan pada rentang $< 1 \text{ Hz}$, terlihat adanya bentuk ekor 45 derajat di sisi kanan plot. Hasil ini menyerupai bentuk ECM Warburg seperti pada Gambar 2.10. Oleh karena itu, dapat diasumsikan bahwa kedua baterai ini sudah mengalami degradasi, dengan adanya tanda peningkatan SEI pada memanjangnya plot (kedua baterai) dan adanya efek difusi dengan adanya bentuk garis 45 derajat (baterai bekas tua). Perbandingan hasil rerata antara ketiga sampel baterai ditampilkan pada Gambar 4.32.



Gambar 4.32. Perbandingan hasil pengukuran antara 3 sampel baterai

4.6.4 Karakterisasi Hasil Pengukuran Berdasar Model dan Kurva Karakteristik

Karakteristik model dan kurva hasil pengukuran baterai sangat dipengaruhi oleh usia dan tingkat degradasi yang terjadi dalam baterai tersebut. Seiring waktu, berbagai faktor seperti siklus pengisian dan pengosongan, suhu operasional, serta kondisi penyimpanan dapat menyebabkan perubahan dalam sifat elektrokimia baterai, yang pada gilirannya mempengaruhi hasil pengukuran impedansi.

Baterai litium-ion umumnya dimodelkan sebagai rangkaian Randles, rangkaian Warburg, atau kombinasi keduanya seperti yang ditunjukkan oleh Gambar 2.12. Model tersebut berisi beberapa komponen resistif dan kapasitif yang merepresentasikan fenomena kimia di dalam baterai. Terdapat empat parameter umum yang mewakili komponen kimia baterai yaitu resistansi elektrolit (R_S), resistansi transfer muatan (R_{CT}), kapasitansi lapisan ganda (C_{DL}), dan resistansi Warburg (W). Namun, hanya tiga komponen yang akan digunakan dalam analisis pengujian dan karakterisasi baterai. resistansi Warburg tidak digunakan karena memerlukan analisis yang lebih dalam dengan mempertimbangkan bahan penyusun baterai dan konstanta Warburg.

4.6.4.1 Perhitungan Resistansi Elektrolit

Salah satu parameter yang selalu ada dalam semua model baterai adalah resistansi elektrolit/*solution resistance* (R_S). R_S seringkali memiliki komponen tambahan seperti resistansi *current collector*, resistansi *separator*, dan juga resistansi kabel pengukuran sehingga terkadang disebut sebagai resistansi *bulk* (R_b). Nilai R_S dapat ditentukan pada titik potong sumbu nyata dalam area frekuensi tinggi ($f > 1 \text{ kHz}$), yaitu di mana garis melintasi sumbu x di sisi paling kiri plot. Komponen riil dari impedansi cenderung mendekati nilai yang sama pada zona ini sehingga impedansi yang tersisa sebagian besar adalah resistansi murni. Oleh karena itu, nilai R_S dapat ditemukan pada titik ini.

Hasil pengukuran seringkali tidak memberi titik yang secara langung berada pada sumbu riil. Maka digunakan pendekatan interpolasi linier antara dua titik yang berdekatan dengan sumbu riil untuk mendapatkan nilai R_S . Misalkan dua titik tersebut adalah (R_1, jX_1) dan (R_2, jX_2) , di mana R adalah bagian riil dan X adalah bagian imajiner. Menggunakan persamaan interpolasi linier, nilai internal resistance R_S dapat dinyatakan dengan Persamaan 4-7.

$$R_S = R_1 + \frac{(0 - X_1) \cdot (R_2 - R_1)}{X_2 - X_1} \quad (4-7)$$

Melalui pendekatan ini, didapat nilai R_S dari hasil pengukuran impedansi pada plot Nyquist yang ditampilkan pada Tabel 4.7.

Tabel 4.7. Nilai resistansi elektrolit untuk setiap pengukuran baterai

Resistansi Elektrolit (mΩ)			
n	25R2 Baru	25R2 Bekas Baru	25R2 Bekas Tua
1	17,619	19,604	18,572
2	17,674	19,561	18,497
3	17,753	19,548	18,478
4	17,718	19,608	18,517
5	17,682	19,648	18,614
6	17,647	19,599	18,587
7	17,733	19,622	18,545
8	17,626	19,595	18,578
9	17,735	19,632	18,556
10	17,628	19,594	18,521

4.6.4.2 Perhitungan Resistansi Transfer Muatan

Parameter selanjutnya yang selalu ada dalam semua model baterai adalah resistansi transfer muatan (R_{CT}). Nilai R_{CT} dari karakterisasi plot Nyquist dipengaruhi oleh nilai R_S dan/atau nilai R_{SEI} . SEI terbentuk pada elektroda negatif seiring bertambahnya usia baterai sehingga pada baterai yang usianya masih muda atau baru, SEI belum terbentuk. Maka dari itu, dalam menghitung R_{CT} perlu ditentukan apakah terdapat karakteristik yang menunjukkan adanya R_{SEI} . Plot Nyquist baterai litium-ion yang sudah terbentuk SEI biasanya menampilkan karakteristik impedansi dengan adanya dua bentuk setengah lingkaran. Setengah lingkaran pertama setelah zona Warburg ($f > 1$ Hz) berhubungan dengan charge transfer effect dan setengah lingkaran kedua sebelum zona ohmic ($f < 1$ kHz) berhubungan dengan SEI. Hasil pengukuran dari ketiga baterai menunjukkan tidak adanya karakteristik SEI. Plot hanya menunjukkan lengkungan besar pada frekuensi rendah (1 Hz $< f < 1$ kHz) yang khas untuk zona R_{CT} . Titik transisi ini ($R_{R_{CT}+R_S}$) dapat ditemukan dengan menganalisis perubahan tajam pada turunan kedua dari kurva impedansi, yang menandakan awal dari lengkungan besar. Setelah menentukan kedua titik impedansi, nilai R_{CT} dapat dihitung menggunakan Persamaan 4-8.

$$R_{CT} = R_{R_{CT}+R_S} - R_S \quad (4-8)$$

Melalui pendekatan ini, didapat nilai R_{CT} dari hasil pengukuran impedansi pada plot Nyquist yang ditampilkan pada Tabel 4.8.

Tabel 4.8. Nilai resistansi transfer muatan untuk setiap pengukuran baterai

Resistansi Transfer Muatan ($\text{m}\Omega$)			
n	25R2 Baru	25R2 Bekas Baru	25R2 Bekas Tua
1	6,016	8,865	11,116
2	6,271	9,292	10,981
3	5,995	9,079	10,955
4	6,152	8,901	11,444
5	6,241	8,957	11,664
6	6,172	8,961	11,243
7	6,137	9,381	11,301
8	5,856	9,120	10,939
9	6,056	8,510	11,421
10	6,337	8,845	11,475

4.6.4.3 Perhitungan Kapasitansi Lapisan Ganda

Komponen kapasitansi lapisan ganda (C_{DL}) merupakan bagian dari rangkaian RC yang disusun secara paralel. Plot Nyquist dari rangkaian RC paralel ini akan membentuk sebuah setengah lingkaran sempurna, seperti yang terdapat pada model Randles pada Gambar 2.8.

Nilai C_{DL} kemudian dapat ditentukan setelah mengetahui nilai diameter setengah lingkaran (besar nilai RCT) dan nilai frekuensi resonansi f_S [13], yaitu frekuensi di mana impedansi imajiner (Z_{im}) mencapai nilai maksimum setengah lingkaran di plot Nyquist. Pada titik ini didapatkan bahwa bentuk setengah lingkaran memiliki nilai yang direpresentasikan oleh Persamaan 4-9 [6], [13].

$$\omega R_{CT} C_{DL} = 1 \quad (4-9)$$

Melalui bentuk Persamaan 4-9, nilai kapasitansi lapisan ganda dapat dihitung menjadi Persamaan 4-10.

$$C_{DL} = \frac{1}{2\pi f_S R_{CT}} \quad (4-10)$$

Melalui pendekatan ini, didapat nilai C_{DL} dari hasil pengukuran impedansi pada plot Nyquist yang ditampilkan pada Tabel 4.9.

Tabel 4.9. Nilai kapasitansi lapisan ganda untuk setiap pengukuran baterai

Kapasitansi Lapisan Ganda (μF)			
n	25R2 Baru	25R2 Bekas Baru	25R2 Bekas Tua
1	115,712	657,778	1339,164
2	111,004	627,540	1365,808
3	116,121	642,284	1369,030
4	113,139	655,076	1310,530
5	111,528	650,986	1285,743
6	112,779	650,758	1333,915
7	113,424	621,591	1327,016
8	118,862	639,353	1370,959
9	114,943	685,174	1313,098
10	109,844	659,271	1306,923

4.6.5 Analisis Akurasi Pengukuran

Akurasi peranti pengukuran impedansi akan ditentukan berdasar nilai *relative standard deviation* (RSD) dari parameter R_S , R_{CT} , dan C_{DL} yang didapat. Nilai ini diharapkan dapat mewakili akurasi peranti pengukuran karena merepresentasikan area frekuensi yang menjadi titik penting karakterisasi baterai litium-ion. RSD adalah ukuran seberapa besar variasi dalam data pengukuran terhadap rata-rata dari data tersebut. Nilai RSD ditentukan berdasarkan besar standar deviasi, yaitu ukuran besar variasi atau penyebaran data dalam suatu himpunan data. Secara matematis, perhitungan nilai standar deviasi ditunjukkan oleh Persamaan 4-11.

$$\sigma = \sqrt{\frac{\sum (x_i - \bar{x})^2}{N}} \quad (4-11)$$

Di mana σ adalah standar deviasi, x_i adalah masing-masing nilai data, \bar{x} adalah rata-rata dari nilai data, dan N adalah jumlah total nilai data. Setelah mendapat nilai standar deviasi dari data pengukuran, RSD dapat dihitung dengan membagi standar deviasi dengan rata-rata dan dikalikan persen yang ditunjukkan oleh Persamaan 4-12.

$$RSD = \frac{\sigma}{\bar{x}} \cdot 100\% \quad (4-12)$$

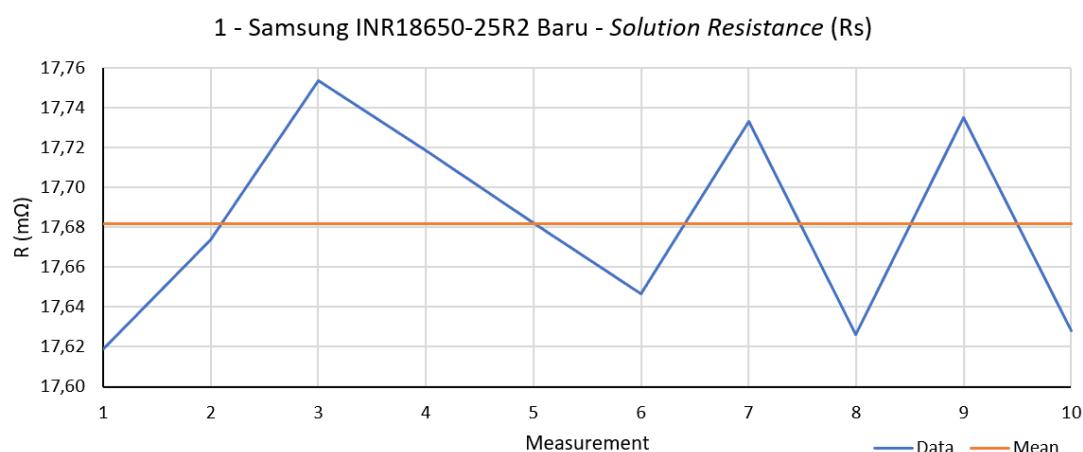
Melalui RSD, tingkat variasi dan akurasi dari pengukuran dapat ditentukan. Hasil perhitungan RSD untuk tiap parameter baterai diuraikan di bawah.

4.6.5.1 Pengukuran Resistansi Elektrolit

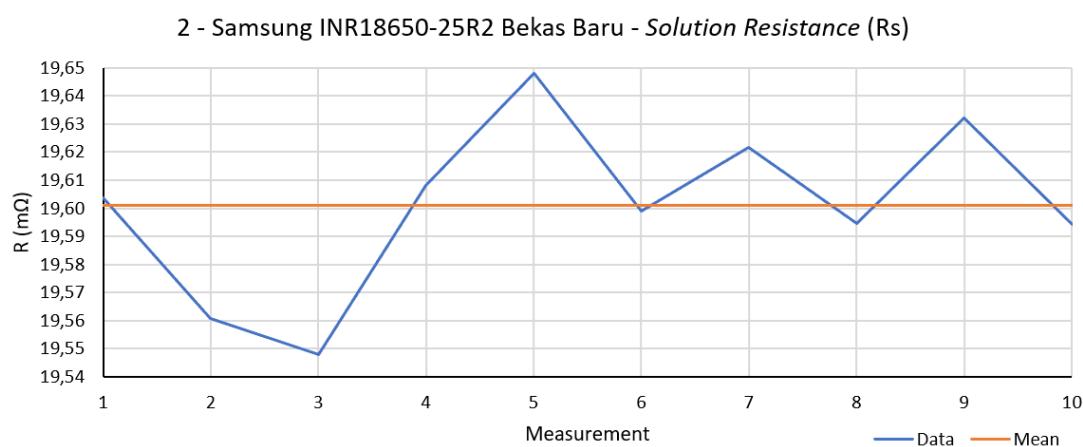
Tabel 4.10. Perhitungan RSD untuk pengukuran R_S

Parameter	25R2 Baru	25R2 Bekas Baru	25R2 Bekas Tua
Mean	17,682	19,601	18,547
Standar Deviasi	0,048	0,029	0,041
RSD (%)	0,272	0,146	0,220

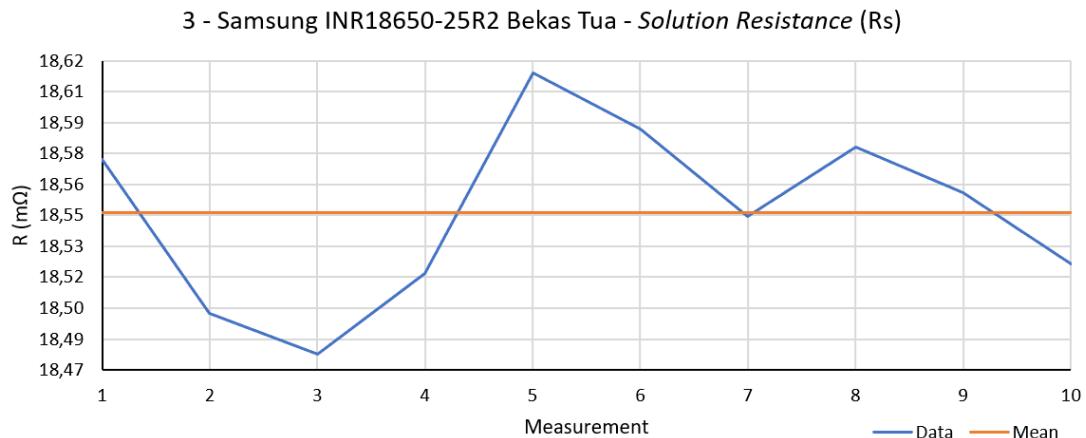
Nilai RSD dari pengukuran R_S menunjukkan nilai error kurang dari 0,5% pada area frekuensi >1 kHz yang merupakan zona *ohmic*. Hasil ini sangat baik mengingat bahwa pengukuran impedansi baterai memerlukan akurasi yang tinggi. Simpangan dari masing-masing pengukuran dan nilai reratanya digambarkan oleh plot yang ditampilkan pada Gambar 4.33, Gambar 4.34, dan Gambar 4.35.



Gambar 4.33. Nilai R_S pengukuran baterai Samsung INR18650-25R2 Baru



Gambar 4.34. Nilai R_S pengukuran baterai Samsung INR18650-25R2 Bekas Baru



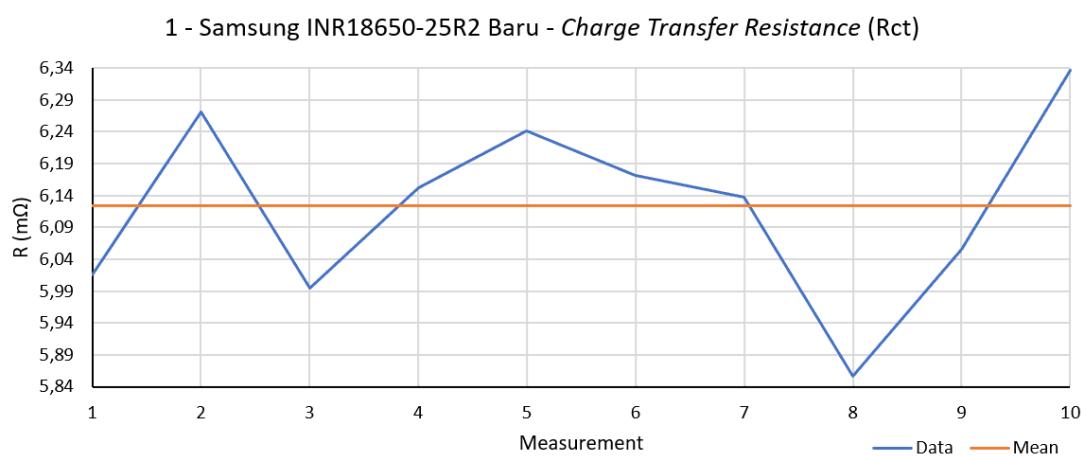
Gambar 4.35. Nilai R_s pengukuran baterai Samsung INR18650-25R2 Bekas Tua

4.6.5.2 Pengukuran Resistansi Transfer Muatan

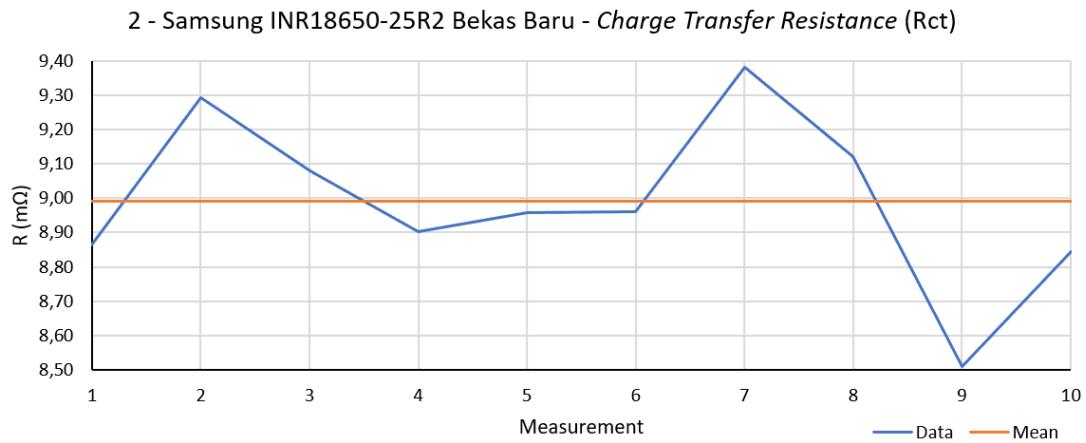
Tabel 4.11. Perhitungan RSD untuk pengukuran R_{CT}

Parameter	25R2 Baru	25R2 Bekas Baru	25R2 Bekas Tua
Mean	6,123	8,991	11,254
Standar Deviasi	0,137	0,234	0,238
RSD (%)	2,244	2,597	2,114

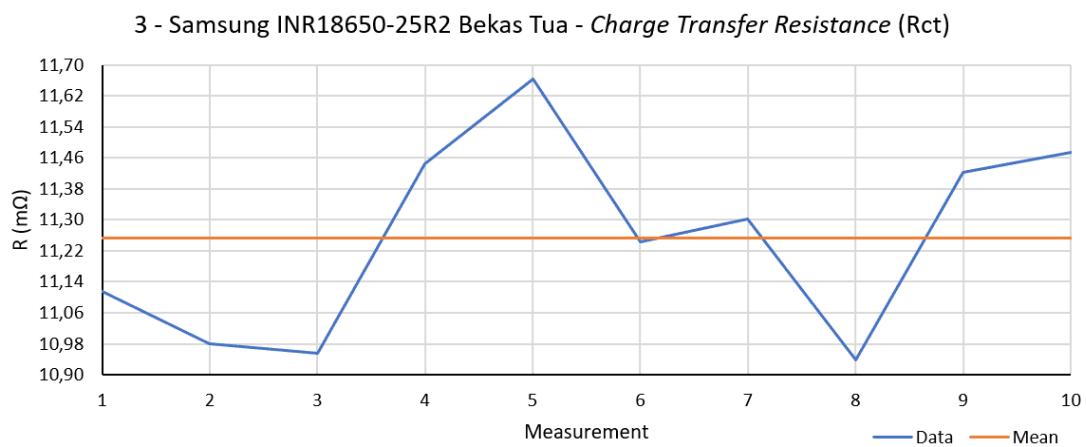
Nilai RSD dari pengukuran R_{CT} menunjukkan nilai error kurang dari 3% pada area frekuensi < 10 Hz yang merupakan area transisi dari zona Warburg. Hasil ini tergolong baik melihat bahwa penentuan titik transisi ini memiliki nilai cukup fluktuatif. Simpangan dari masing-masing pengukuran dan nilai reratanya digambarkan oleh plot yang ditampilkan pada Gambar 4.36, Gambar 4.37, dan Gambar 4.38.



Gambar 4.36. Nilai R_{CT} pengukuran baterai Samsung INR18650-25R2 Baru



Gambar 4.37. Nilai R_{CT} pengukuran baterai Samsung INR18650-25R2 Bekas Baru



Gambar 4.38. Nilai R_{CT} pengukuran baterai Samsung INR18650-25R2 Bekas Tua

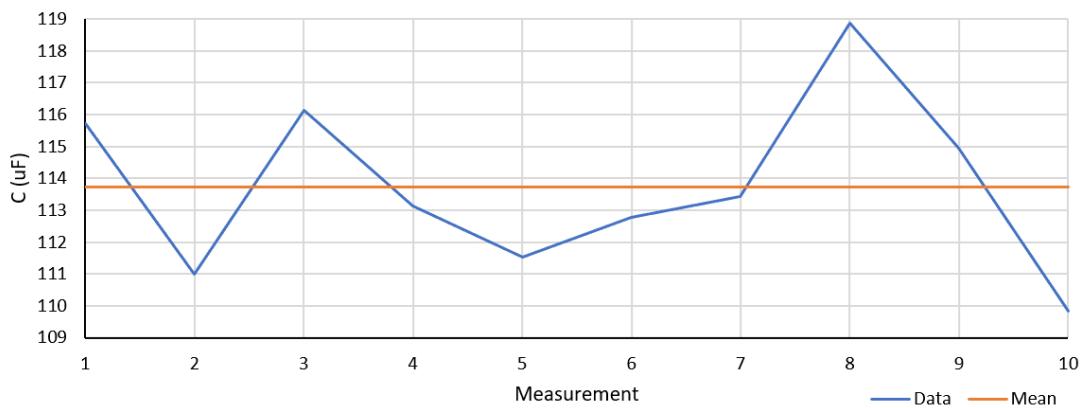
4.6.5.3 Pengukuran Kapasitansi Lapisan Ganda

Tabel 4.12. Perhitungan RSD untuk pengukuran C_{DL}

Parameter	25R2 Baru	25R2 Bekas Baru	25R2 Bekas Tua
Mean	113,736	648,981	1333,219
Standar Deviasi	2,573	16,985	28,155
RSD (%)	2,262	2,617	2,112

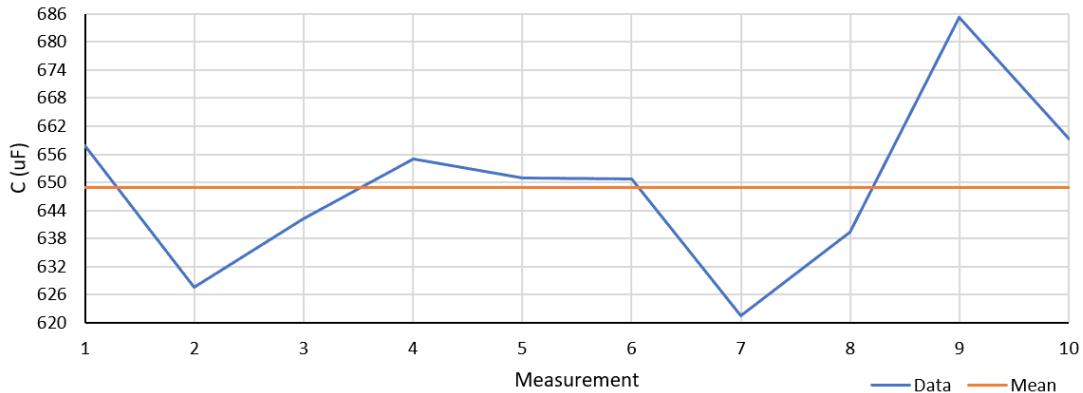
Nilai RSD dari pengukuran C_{DL} menunjukkan error kurang dari 3% pada area frekuensi 1 Hz sampai 1 kHz yang membentuk setengah lingkaran. Hasil ini tergolong baik mengingat bahwa perhitungan nilai C_{DL} bergantung pada nilai R_{CT} . Simpangan dari masing-masing pengukuran dan nilai reratanya digambarkan oleh plot yang ditampilkan pada Gambar 4.39, Gambar 4.40, dan Gambar 4.41.

1 - Samsung INR18650-25R2 Baru - Double Layer Capacitance (CdL)



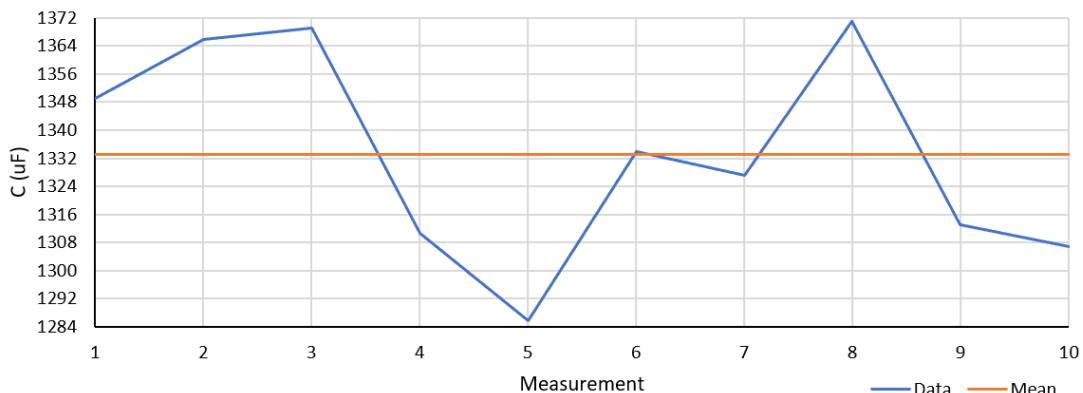
Gambar 4.39. Nilai C_{DL} pengukuran baterai Samsung INR18650-25R2 Baru

2 - Samsung INR18650-25R2 Bekas Baru - Double Layer Capacitance (CdL)



Gambar 4.40. Nilai C_{DL} pengukuran baterai Samsung INR18650-25R2 Bekas Baru

3 - Samsung INR18650-25R2 Bekas Tua - Double layer Capacitance (CdL)



Gambar 4.41. Nilai C_{DL} pengukuran baterai Samsung INR18650-25R2 Bekas Tua

4.7 Pengujian Pengukuran Baterai Litium-ion Lainnya

Pengujian bertujuan untuk melihat kemampuan perangkat dalam mengukur baterai litium-ion lainnya dan menghasilkan plot yang menunjukkan perbedaan karakter. Pengukuran akan dilakukan kepada beberapa merek baterai dengan kondisi usia yang berbeda-beda. Baterai yang diukur dalam pengujian ini ditampilkan dalam Tabel 4.13.

Tabel 4.13. Merek dan parameter baterai litium-ion objek pengujian

No	Merek	Seri	Tegangan Baterai (V)
1	Sony	US18650VTC6	4,128
2	Sony	US18650VTC6	4,134
3	Sony	US18650VTC6	3,936
4	Sony	US18650VTC6	3,995
5	Sony	US18650VTC5	3,632
6	AWT	INR18650	3,731
7	AWT	INR18650	3,720

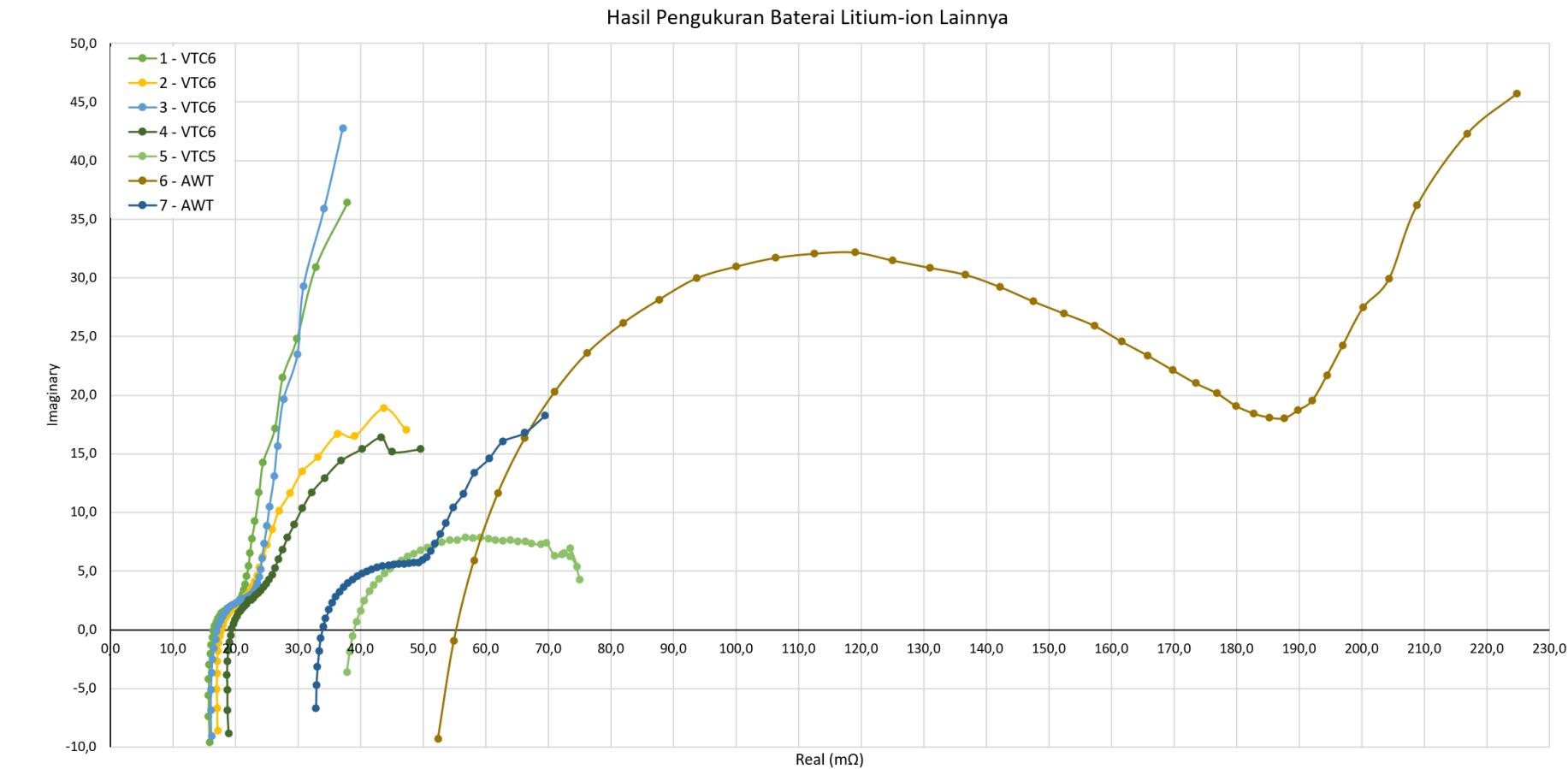
Pengujian ini akan menerapkan parameter pengukuran yang sama pada setiap sampel baterai, nilai parameter diuraikan sebagai berikut:

- *Start Frequency* = 1 Hz
- *Stop Frequency* = 10.000 Hz
- *Sweep Points* = 40

Hasil pengukuran setiap baterai ditampilkan dalam satu plot Nyquist pada Gambar 4.42. Plot menunjukkan bahwa setiap baterai memiliki karakter yang berbeda, menandakan perbedaan degradasi dan perubahan komponen di dalamnya.

Baterai VTC6 nomor 1-4 memiliki nilai potong sumbu riil yang cukup rendah, tetapi membentuk karakter setengah lingkaran kecil di sisi kiri dan setengah lingkaran besar di sisi kanan yang belum selesai. Bentuk setengah lingkaran kecil ini berhubungan dengan terbentuknya SEI dan setengah lingkaran besar menunjukkan adanya nilai R_{CT} yang besar. Namun, baterai nomor 1 dan 2 memiliki bentuk frekuensi rendah dengan nilai imajiner yang lebih tinggi dan kemungkinan menuju tak terhingga.

Baterai AWT nomor 6 memiliki nilai impedansi yang tinggi dengan bentuk menyerupai plot baterai VTC6 nomor 4, tetapi dengan bentuk yang lebih memanjang. Sedangkan baterai AWT nomor 6 membentuk kurva yang lebih besar lagi, menandakan bahwa baterai tersebut sudah kehilangan kapasitasnya dan membentuk hambatan yang besar di dalam. Baterai VTC5 nomor 5 memiliki nilai impedansi yang tinggi tetapi memiliki bentuk yang berbeda dibandingkan baterai yang lain. Kurva yang dihasilkan hanya membentuk setengah lingkaran yang besar.



Gambar 4.42. Hasil Pengukuran Baterai Litium-ion Lainnya

4.8 Spesifikasi Akhir Peranti Pengukuran

Berdasarkan perancangan, implementasi, dan hasil pengujian peranti pengukuran impedansi serta informasi berdasarkan *datasheet* IC AD5941 [16], maka spesifikasi peranti pengukuran dapat dirangkum pada Tabel 4.14.

Tabel 4.14. Spesifikasi Akhir Peranti Pengukuran

Parameter Fisik					
Dimensi		Panjang	Lebar	Tinggi	Satuan
PCB Antarmuka		105	105	18	mm
PCB Antarmuka + STM32 + EVAL-AD5941BATZ		131	105	30	mm
Keseluruhan sistem + Kabel pengukuran		242	105	30	mm
Parameter Elektris					
Simbol	Parameter	Min	Nom	Maks	Satuan
V_{IN}	Tegangan masukan pada USB	4,5	5	5,5	V
	Tegangan masukan pada pin EXT	4,5	-	36	V
V_{OUT}	Tegangan keluaran regulator		3,3		V
$I_{REG_{max}}$	Arus keluaran maksimum regulator		1		A
f_{SPI}	Frekuensi clock SPI		6,25		MHz
Bd_{UART}	Pesat baud UART		115200		bps
f_{sweep}	Frekuensi sapuan	0,1	-	200.000	Hz
n_{sweep}	Titik sapuan	2	-	999	pts
DC_{offset}	DC bias		1200		mV
A_{AC}	AC amplitude		300		mVpp
$Sweep$	Jenis sapuan logaritmik		Ya		
i_{exc}	Arus eksitasi maksimum		67		mA
f_{sADC}	Mode daya rendah ($f_{sweep} < 80\text{kHz}$)		800		kSPS
	Mode daya tinggi ($f_{sweep} > 80\text{kHz}$)		1600		kSPS
RSD	Saat $f_{sweep} < 1\text{kHz}$	2,20	2,40	3	%
	Saat $f_{sweep} > 1\text{kHz}$	0,15	0,22	1	%

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Telah berhasil direalisasikan sebuah peranti pengukuran impedansi baterai dengan modul EVAL-AD5941BATZ dan dihubungkan ke perangkat komputer melalui mikrokontroler STM32F411CEU6.
2. Peranti memiliki ukuran yang portabel dengan dimensi akhir sebesar 242 x 105 x 30 mm dan harga yang cukup terjangkau.
3. Perangkat komputer dapat menerima data impedansi dan menampilkannya dalam bentuk plot Nyquist maupun plot Bode melalui GUI.
4. Peranti memiliki akurasi yang baik dengan nilai RSD kurang dari 1% untuk parameter R_S pada frekuensi sapuan $> 1 \text{ kHz}$ dan RSD kurang dari 5% untuk parameter R_{CT} dan C_{DL} pada frekuensi sapuan $< 1 \text{ kHz}$.

5.2 Saran

Saran untuk penelitian selanjutnya yaitu:

1. Mengembangkan konektor baterai untuk bentuk dan tipe baterai litium-ion lainnya.
2. Mengembangkan GUI dengan menambahkan fitur analisis dengan bantuan ECM.
3. Melakukan analisis lebih lanjut untuk merk dan tipe baterai berbasis litium lainnya.
4. Mengembangkan perangkat sehingga dapat melakukan pengukuran impedansi yang lebih luas seperti impedansi antena dan impedansi jaringan biologis.

DAFTAR PUSTAKA

- [1] “Trends in electric light-duty vehicles,” <https://www.iea.org/reports/global-ev-outlook-2023/trends-in-electric-light-duty-vehicles>, accessed: 2023-09-26.
- [2] J. Warner, *The Handbook of Lithium-Ion Battery Pack Design: Chemistry, Components, Types and Terminology*. Elsevier, 2015.
- [3] Y. Liu, Y. G. Liao, and M.-C. Lai, “Effects of depth-of-discharge, ambient temperature, and aging on the internal resistance of lithium-ion battery cell,” in *2021 International Conference on Electrical, Computer and Energy Technologies (ICE-CET)*, 2021, pp. 1–5.
- [4] W. Liu, Q. Ma, X. He, Q. Li, M. Dong, and M. Ren, “Study on impedance characteristics and detection methods of lithium-ion battery overcharge and over discharge based on electrochemical impedance spectroscopy,” in *2023 IEEE 6th International Electrical and Energy Conference (CIEEC)*, 2023, pp. 966–971.
- [5] A. Gailani, R. Mokidm, M. El-Dalahmeh, M. El-Dalahmeh, and M. Al-Greer, “Analysis of lithium-ion battery cells degradation based on different manufacturers,” in *2020 55th International Universities Power Engineering Conference (UPEC)*, 2020, pp. 1–6.
- [6] W. Choi, H.-C. Shin, J. M. Kim, J. Choi, and W.-S. Yoon, “Modeling and applications of electrochemical impedance spectroscopy (eis) for lithium-ion batteries,” *Journal of Electrochemical Science and Technology*, vol. 11, pp. 1–13, 03 2020.
- [7] B. Balagopal, C. S. Huang, and M.-Y. Chow, “Sensitivity analysis of lithium ion battery parameters to degradation of anode lithium ion concentration,” in *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1, 2019, pp. 4543–4548.
- [8] D.-I. Stroe, M. Swierczynski, S. K. Kær, and R. Teodorescu, “A comprehensive study on the degradation of lithium-ion batteries during calendar ageing: The internal resistance increase,” in *2016 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2016, pp. 1–7.
- [9] “Cn0510 electrochemical impedance spectroscopy (eis) for batteries,” <https://www.analog.com/media/en/reference-design-documentation/reference-designs/CN0510.pdf>, accessed: 2023-10-05.
- [10] L. E. Sebar, L. Iannucci, E. Angelini, S. Grassini, and M. Parvis, “Electrochemical impedance spectroscopy system based on a teensy board,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.
- [11] A. Zilio and P. Mattavelli, “An experimental validation of the adoption of dc-dc converters for the impedance measurement in li-ion batteries,” in *2021 IEEE International Conference on Environment and Electrical Engineering and 2021 IEEE Industrial and Commercial Power Systems Europe (EEEIC / ICPS Europe)*, 2021, pp. 1–6.

- [12] K. HUANG, Y. Wang, and J. FENG, “Research on equivalent circuit model of lithium-ion battery for electric vehicles,” in *2020 3rd World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*, 2020, pp. 492–496.
- [13] H. Magar, Y. Rabeay, R. Hassan, and A. Mulchandani, “Electrochemical impedance spectroscopy (eis): Principles, construction, and biosensing applications,” *Sensors*, vol. 21, p. 6578, 10 2021.
- [14] A. Fischer, A. Y. Kallel, and O. Kanoun, “Comparative study of excitation signals for microcontroller-based eis measurement on li-ion batteries,” in *2021 International Workshop on Impedance Spectroscopy (IWIS)*, 2021, pp. 44–47.
- [15] O. Alao and P. Barendse, “Characterization of lithium ncm and sealed lead acid batteries using electrochemical impedance spectroscopy,” in *2018 IEEE PES/IAS PowerAfrica*, 2018, pp. 166–171.
- [16] A. Devices, “Ad5940/5941 datasheet,” <https://www.analog.com/en/products/AD5941.html>, accessed: 2023-10-12.
- [17] STMicroelectronics, “Stm32f411ce,” <https://www.st.com/en/microcontrollers-microprocessors/stm32f411ce.html#overview>, accessed: 2024-01-26.
- [18] P. Dhaker, “Introduction to spi interface,” <https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html>, accessed: 2024-01-27.
- [19] J. Wu, W. Bai, L. Zhang, X. Zhang, H. Lin, H. Dai, J. Liu, F. Zhang, and Y. Yang, “Design of a portable electrochemical impedance spectroscopy measurement system based on ad5941 for lithium-ion batteries,” *Journal of Energy Storage*, vol. 84, p. 110856, 2024.
- [20] “Samsung inr18650-25r datasheet,” <https://www.powerstream.com/p/INR18650-25R-datasheet.pdf>, accessed: 2024-5-20.

LAMPIRAN

L.1 Program STM32

L.1.1 Program main.c

```
1 /*  
2  * @file      main.c  
3  * @author    Riko Kumara  
4  * @version   V1.6  
5  * @date      20-June-2024  
6  * @brief     This file provides is the main code  
7 */  
8 #include "stdio.h"  
9 #include "AD5940.h"  
10 #include "stm32f4xx_hal.h"  
11 #include "BATImpedance.h"  
12 /* Definition for AD5940 Pins */  
13 #define DEBUG_UART              USART2  
14 #define DEBUG_UART_IRQN         USART2_IRQn  
15 #define DEBUGUART_CLK_ENABLE()  __HAL_RCC_USART2_CLK_ENABLE()  
16 #define DEBUGUART_GPIO_CLK_ENABLE() __HAL_RCC_GPIOA_CLK_ENABLE()  
17 #define DEBUGUART_TX_PIN          GPIO_PIN_2  
18 #define DEBUGUART_TX_GPIO_PORT   GPIOA  
19 #define DEBUGUART_TX_AF           GPIO_AF7_USART2  
20 #define DEBUGUART_RX_PIN          GPIO_PIN_3  
21 #define DEBUGUART_RX_GPIO_PORT   GPIOA  
22 #define DEBUGUART_RX_AF           GPIO_AF7_USART2  
23 /* Definition for Private Variables */  
24 #define UART_RX_BUFF_SIZE        16  
25 float    SWEEP_START_FREQ    = 0.1f;  
26 float    SWEEP_STOP_FREQ     = 10000.0f;  
27 int     SWEEP_POINTS        = 50;  
28 int     GUI_COMMAND         = 0;  
29 int     input_stat          = 0;  
30 int     state               = 0;  
31 char    rx_data[UART_RX_BUFF_SIZE];  
32 char    rx_buff[UART_RX_BUFF_SIZE];  
33 char    temp[10];  
34 /* Functions that used to initialize MCU platform */  
35 uint32_t MCUPlatformInit(void *pCfg);  
36 UART_HandleTypeDef UartHandle;  
37 void UserBATStructInit(void) {  
38     AppBATCfg_Type *pBATCfg;  
39     AppBATGetCfg(&pBATCfg);  
40     pBATCfg->SeqStartAddr      = 0;  
41     pBATCfg->MaxSeqLen        = 512;  
42     pBATCfg->RcalVal          = 50.0;  
43     pBATCfg->ACVoltPP         = 300.0f;  
44     pBATCfg->DCVolt           = 1200.0f;  
45     pBATCfg->DftNum           = DFTNUM_8192;  
46     pBATCfg->NumOfData        = -1;  
47     pBATCfg->BatODR           = 20;  
48     pBATCfg->FifoThresh       = 2;  
49     pBATCfg->SinFreq          = 200;  
50     pBATCfg->SweepCfg.SweepEn = bTRUE;  
51     pBATCfg->SweepCfg.SweepStart = SWEEP_START_FREQ;  
52     pBATCfg->SweepCfg.SweepStop  = SWEEP_STOP_FREQ;
```

```

53     pBATCfg->SweepCfg.SweepPoints    = SWEEP_POINTS;
54     pBATCfg->SweepCfg.SweepLog      = bTRUE;
55 }
56 int main(void) {
57     MCUPlatformInit(0);
58     while(1) {
59         RX_Receive();
60         if(rx_data[0] == '#' && rx_data[2] == '#') {
61             switch(rx_data[1]) {
62                 case '=': // Idle Mode
63                     printf("#=#\n");
64                     break;
65                 case '?': // Sync
66                     SyncSerial(); // Send Param Data to GUI
67                     break;
68                 case 'M': // Start Measuring
69                     StartMeasure();
70                     break;
71                 case 'W': // Write User Param
72                     UserParamInput();
73                     input_stat = 0;
74                     break;
75             } } else printf("#=#\n");
76     }
77 void StartMeasure(void) {
78     state = 1;
79     HAL_Delay(100);
80     printf("#K#\n");
81     UserBATStructInit();
82     void AD5940_Main(void);
83     void AD5940_RCal(void);
84     AD5940 MCUResourceInit(0);
85     printf("#RC#\n"); HAL_Delay(300);
86     AD5940_RCal();
87     printf("#=#\n"); HAL_Delay(1000);
88     printf("#IM#\n"); HAL_Delay(300);
89     AD5940_Main();
90     printf("#=#\n");
91     state = 0;
92 }
93 void SyncSerial(void) {
94     state = 1;
95     int sync_stat = 0;
96     while(!sync_stat) {
97         printf("#P");
98         printf("%.1f%.1f#%d#\n", SWEEP_START_FREQ, SWEEP_STOP_FREQ, SWEEP_POINTS);
99         RX_Receive();
100        if(rx_data[1] == '=') sync_stat=1;
101    }
102    printf("#=#\n");
103    state = 0;
104 }
105 void UserParamInput(void) {
106     state = 1;
107     HAL_Delay(50);
108     int cnt=0;
109     printf("#K#\n");
110     HAL_Delay(10);
111     while(!input_stat) {

```

```

112     RX_Receive();
113     if(rx_data[1]=='1' && rx_data[2]=='#') {
114         HAL_Delay(10);
115         for(int i=3; i<10; i++) {
116             if(rx_data[i]!='#') temp[i-3]+=rx_data[i];
117             else if (rx_data[i]=='#') break;
118         }
119         HAL_Delay(20);
120         SWEEP_START_FREQ = atof(temp);
121         printf("#%.2f#\n", SWEEP_START_FREQ);
122         for(int i=0; i<10; i++) temp[i] = 0;
123     }
124     else if(rx_data[1]=='2' && rx_data[2]=='#') {
125         HAL_Delay(10);
126         for(int i=3; i<10; i++) {
127             if(rx_data[i]!='#') temp[i-3]+=rx_data[i];
128             else if (rx_data[i]=='#') break;
129         }
130         HAL_Delay(20);
131         SWEEP_STOP_FREQ = atof(temp);
132         printf("#%.2f#\n", SWEEP_STOP_FREQ);
133         for(int i=0; i<10; i++) temp[i] = 0;
134     }
135     else if(rx_data[1]=='3' && rx_data[8]=='#') {
136         HAL_Delay(10);
137         for(int i=3; i<10; i++) {
138             temp[i-3] += rx_data[i];
139         }
140         HAL_Delay(20);
141         float tempe = atof(temp);
142         SWEEP_POINTS = (uint8_t)(tempe);
143         printf("#%d#\n", SWEEP_POINTS);
144         for(int i=0; i<10; i++) temp[i] = 0;
145     }
146     else if(rx_data[1]=='=') {
147         input_stat = 1;
148         for(int i=0; i<10; i++) temp[i] = 0;
149         break;
150     }
151     else if(cnt>10) { printf("#K#\n"); cnt=0; }
152     else { HAL_Delay(10); cnt++; }
153 }
154 printf("#=#\n");
155 state = 0;
156 }
```

Listing 1. Kode main.c untuk firmware STM32

L.1.2 Program STM32F411Port.c

```

1 ****
2 * @file      NUCLEOF411Port.c
3 * @version    V1.1
4 * @author     Riko Kumara
5 * @date       20-June-2024
6 * @brief      STM32F411 board port file.
7 ****
8 #include "ad5940.h"
```

```

9 | #include "stdio.h"
10| #include "stm32f4xx_hal.h"
11| /* Definition for STM32 SPI clock resources */
12| #define AD5940SPI           SPI1
13| #define AD5940_CLK_ENABLE()   __HAL_RCC_SPI1_CLK_ENABLE()
14| #define AD5940_SCK_GPIO_CLK_ENABLE() __HAL_RCC_GPIOA_CLK_ENABLE()
15| #define AD5940_MISO_GPIO_CLK_ENABLE() __HAL_RCC_GPIOA_CLK_ENABLE()
16| #define AD5940_MOSI_GPIO_CLK_ENABLE() __HAL_RCC_GPIOA_CLK_ENABLE()
17| #define AD5940_CS_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
18| #define AD5940_RST_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
19| #define AD5940_GP0INT_GPIO_CLK_ENABLE() __HAL_RCC_GPIOA_CLK_ENABLE()
20| #define ARDUINO_D3_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
21| #define ARDUINO_D4_GPIO_CLK_ENABLE() __HAL_RCC_GPIOB_CLK_ENABLE()
22| #define AD5940SPI_FORCE_RESET() __HAL_RCC_SPI1_FORCE_RESET()
23| #define AD5940SPI_RELEASE_RESET() __HAL_RCC_SPI1_RELEASE_RESET()
24| /* Definition for AD5940 Pins */
25| #define AD5940_SCK_PIN        GPIO_PIN_5
26| #define AD5940_SCK_GPIO_PORT  GPIOA
27| #define AD5940_SCK_AF         GPIO_AF5_SPI1
28| #define AD5940_MISO_PIN       GPIO_PIN_6
29| #define AD5940_MISO_GPIO_PORT GPIOA
30| #define AD5940_MISO_AF        GPIO_AF5_SPI1
31| #define AD5940_MOSI_PIN       GPIO_PIN_7
32| #define AD5940_MOSI_GPIO_PORT GPIOA
33| #define AD5940_MOSI_AF        GPIO_AF5_SPI1
34| #define AD5940_CS_PIN         GPIO_PIN_6
35| #define AD5940_CS_GPIO_PORT  GPIOB
36| #define AD5940_RST_PIN        GPIO_PIN_0
37| #define AD5940_RST_GPIO_PORT  GPIOB
38| #define AD5940_GP0INT_PIN     GPIO_PIN_10
39| #define AD5940_GP0INT_GPIO_PORT GPIOA
40| #define AD5940_GP0INT_IRQn    EXTI15_10_IRQHandler
41| #define ARDUINO_D3_PIN        GPIO_PIN_3
42| #define ARDUINO_D3_PORT      GPIOB
43| #define ARDUINO_D4_PIN        GPIO_PIN_5
44| #define ARDUINO_D4_PORT      GPIOB
45| SPI_HandleTypeDef SpiHandle;
46| #define SYSTICK_MAXCOUNT ((1L<<24)-1)
47| #define SYSTICK_CLKFREQ  100000000L
48| volatile static uint8_t ucInterrupted = 0;
49| void AD5940_ReadWriteNBytes(unsigned char *pSendBuffer,unsigned char
50|                             *pRecvBuff,unsigned long length) {
51|     HAL_SPI_TransmitReceive(&SpiHandle, pSendBuffer, pRecvBuff, length, (uint32_t)-1);
52| }
53| void AD5940_CsClr(void) {
54|     HAL_GPIO_WritePin(AD5940_CS_GPIO_PORT, AD5940_CS_PIN, GPIO_PIN_RESET);
55| }
56| void AD5940_CsSet(void) {
57|     HAL_GPIO_WritePin(AD5940_CS_GPIO_PORT, AD5940_CS_PIN, GPIO_PIN_SET);
58| }
59| void AD5940_RstSet(void) {
60|     HAL_GPIO_WritePin(AD5940_RST_GPIO_PORT, AD5940_RST_PIN, GPIO_PIN_SET);
61| }
62| void AD5940_RstClr(void) {
63|     HAL_GPIO_WritePin(AD5940_RST_GPIO_PORT, AD5940_RST_PIN, GPIO_PIN_RESET);
64| }
65| void AD5940_Delay10us(uint32_t time) {
66|     time/=100;
|     if(time == 0) time =1;

```

```

67     HAL_Delay(time);
68 }
69 uint32_t AD5940_GetMCUIntFlag(void) {
70     return ucInterrupted;
71 }
72 uint32_t AD5940_ClrMCUIntFlag(void) {
73     ucInterrupted = 0;
74     return 1;
75 }
76 /* This function is used to set Dn on Arduino shield(and set it to output) */
77 void Arduino_WriteDn(uint32_t Dn, BoolFlag bHigh) {
78     if(Dn&(1<<3)) { //set D3, P0.13
79         if(bHigh) HAL_GPIO_WritePin(ARDUINO_D3_PORT, ARDUINO_D3_PIN, GPIO_PIN_SET);
80         else       HAL_GPIO_WritePin(ARDUINO_D3_PORT, ARDUINO_D3_PIN, GPIO_PIN_RESET);
81     }
82     if(Dn&(1<<4)) { //Set D4, P0.9
83         if(bHigh) HAL_GPIO_WritePin(ARDUINO_D4_PORT, ARDUINO_D4_PIN, GPIO_PIN_SET);
84         else       HAL_GPIO_WritePin(ARDUINO_D4_PORT, ARDUINO_D4_PIN, GPIO_PIN_RESET);
85     }
86 }
87 uint32_t AD5940_MCUResourceInit(void *pCfg) {
88     GPIO_InitTypeDef GPIO_InitStruct;
89     /* Step1, initialize SPI peripheral and its GPIOs for CS/RST */
90     AD5940_SCK_GPIO_CLK_ENABLE();
91     AD5940_MISO_GPIO_CLK_ENABLE();
92     AD5940_MOSI_GPIO_CLK_ENABLE();
93     AD5940_CS_GPIO_CLK_ENABLE();
94     AD5940_RST_GPIO_CLK_ENABLE();
95     /* Enable SPI clock */
96     AD5940_CLK_ENABLE();
97     GPIO_InitStruct.Pin      = AD5940_SCK_PIN;
98     GPIO_InitStruct.Mode    = GPIO_MODE_AF_PP;
99     GPIO_InitStruct.Pull    = GPIO_PULLDOWN;
100    GPIO_InitStruct.Speed   = GPIO_SPEED_FREQ_VERY_HIGH;
101    GPIO_InitStruct.Alternate = AD5940_SCK_AF;
102    HAL_GPIO_Init(AD5940_SCK_GPIO_PORT, &GPIO_InitStruct);
103    /* SPI MISO GPIO pin configuration */
104    GPIO_InitStruct.Pin = AD5940_MISO_PIN;
105    GPIO_InitStruct.Alternate = AD5940_MISO_AF;
106    HAL_GPIO_Init(AD5940_MISO_GPIO_PORT, &GPIO_InitStruct);
107    /* SPI MOSI GPIO pin configuration */
108    GPIO_InitStruct.Pin = AD5940_MOSI_PIN;
109    GPIO_InitStruct.Alternate = AD5940_MOSI_AF;
110    HAL_GPIO_Init(AD5940_MOSI_GPIO_PORT, &GPIO_InitStruct);
111    /* SPI CS GPIO pin configuration */
112    GPIO_InitStruct.Pin = AD5940_CS_PIN;
113    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
114    HAL_GPIO_Init(AD5940_CS_GPIO_PORT, &GPIO_InitStruct);
115    /* SPI RST GPIO pin configuration */
116    GPIO_InitStruct.Pin = AD5940_RST_PIN;
117    HAL_GPIO_Init(AD5940_RST_GPIO_PORT, &GPIO_InitStruct);
118    AD5940_CsSet();
119    AD5940_RstSet();
120    /* Set the SPI parameters */
121    SpiHandle.Instance          = AD5940SPI;
122    SpiHandle.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_16;
123    SpiHandle.Init.Direction     = SPI_DIRECTION_2LINES;
124    SpiHandle.Init.CLKPhase      = SPI_PHASE_1EDGE;
125    SpiHandle.Init.CLKPolarity   = SPI_POLARITY_LOW;

```

```

126 SpiHandle.Init.FirstBit      = SPI_FIRSTBIT_MSB;
127 SpiHandle.Init.TIMode        = SPI_TIMODE_DISABLE;
128 SpiHandle.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
129 SpiHandle.Init.CRCPolynomial = 7;
130 SpiHandle.Init.NSS          = SPI_NSS_SOFT;
131 SpiHandle.Init.Mode         = SPI_MODE_MASTER;
132 HAL_SPI_Init(&SpiHandle);
133 /* Step 2: Configure external interrupt line */
134 AD5940_GPOINT_GPIO_CLK_ENABLE();
135 GPIO_InitStruct.Pin          = AD5940_GPOINT_PIN;
136 GPIO_InitStruct.Mode         = GPIO_MODE_IT_FALLING;
137 GPIO_InitStruct.Pull         = GPIO_PULLUP;
138 GPIO_InitStruct.Speed        = GPIO_SPEED_FREQ_VERY_HIGH;
139 GPIO_InitStruct.Alternate    = 0;
140 HAL_GPIO_Init(AD5940_GPOINT_GPIO_PORT, &GPIO_InitStruct);
141 /* Enable and set EXTI Line0 Interrupt the lowest priority */
142 HAL_NVIC_EnableIRQ(AD5940_GPOINT IRQn);
143 ARDUINO_D3_GPIO_CLK_ENABLE(); //D3 D4 GPIO
144 ARDUINO_D4_GPIO_CLK_ENABLE();
145 GPIO_InitStruct.Pin          = ARDUINO_D3_PIN;
146 GPIO_InitStruct.Mode         = GPIO_MODE_OUTPUT_PP;
147 GPIO_InitStruct.Pull         = GPIO_PULLDOWN;
148 GPIO_InitStruct.Speed        = GPIO_SPEED_FREQ_VERY_HIGH;
149 GPIO_InitStruct.Alternate    = 0;
150 HAL_GPIO_Init(ARDUINO_D3_PORT, &GPIO_InitStruct);
151 GPIO_InitStruct.Pin          = ARDUINO_D4_PIN;
152 HAL_GPIO_Init(ARDUINO_D4_PORT, &GPIO_InitStruct);
153 return 0;
154 }
155 /* MCU related external line interrupt service routine */
156 void EXTI15_10_IRQHandler() {
157 ucInterrupted = 1;
158 __HAL_GPIO_EXTI_CLEAR_IT(AD5940_GPOINT_PIN);
159 }
```

Listing 2. Kode STM32F411Port.c untuk firmware STM32

L.1.3 Program AD5940Main.c

```

1 ****
2 @file      AD5940Main.c
3 @author    ADI - Modified by Riko kumara
4 @date      20-June-2024
5 @brief     Used to control EIS application and process data.
6 ****
7 #include "ad5940.h"
8 #include <stdio.h>
9 #include "string.h"
10 #include "math.h"
11 #include "BATImpedance.h"
12 #define APPBUFF_SIZE 512
13 uint32_t AppBuff[APPBUFF_SIZE];
14 float maxfreq;
15 uint8_t meas_stat = 0;
16 int32_t BATShowResult(uint32_t *pData, uint32_t DataCount) {
17     fImpCar_Type *pImp = (fImpCar_Type*)pData;
18     float freq;
19     AppBATCtrl(BATCTRL_GETFREQ, &freq); /*Process data*/
```

```

20     for(int i=0;i<DataCount;i++)
21         printf("#I%.3f%.3f%.3f#\n", freq, pImp[i].Real, pImp[i].Image);
22     if(freq == maxfreq)      meas_stat = 1;
23     return 0;
24 }
25 /* Initialize AD5940 basic blocks like clock */
26 static int32_t AD5940PlatformCfg(void) {
27     CLKCfg_Type clk_cfg;
28     FIFOcfg_Type fifo_cfg;
29     AGPIOcfg_Type gpio_cfg;
30     AD5940_HWReset();                                /* Use hardware reset */
31     AD5940_Initialize();                            /* Platform configuration */
32     /* Step1. Configure clock */
33     clk_cfg.ADCClkDiv = ADCCLKDIV_1;
34     clk_cfg.ADCClkSrc = ADCCLKSRC_HFOSC;
35     clk_cfg.SysClkDiv = SYSCLKDIV_1;
36     clk_cfg.SysClkSrc = SYSCLKSRC_HFOSC;
37     clk_cfg.HfOSC32MHzMode = bFALSE;
38     clk_cfg.HFOSCEn = bTRUE;
39     clk_cfg.HFXTALEn = bFALSE;
40     clk_cfg.LFOSCEn = bTRUE;
41     AD5940_CLKCfg(&clk_cfg);
42     /* Step2. Configure FIFO and Sequencer*/
43     fifo_cfg.FIFOEn = bFALSE;
44     fifo_cfg.FIFOMode = FIFOMODE_FIFO;
45     fifo_cfg.FIFOSize = FIFOSIZE_4KB;                /* 4kB for FIFO */
46     fifo_cfg.FIFOSrc = FIFOSRC_DFT;
47     fifo_cfg.FIFOThresh = 4;                          /* DFT result, one pair for RCAL */
48     AD5940_FIFOcfg(&fifo_cfg);                    /* Disable to reset FIFO */
49     fifo_cfg.FIFOEn = bTRUE;
50     AD5940_FIFOcfg(&fifo_cfg);                    /* Enable FIFO here */
51     /* Step3. Interrupt controller */
52     AD5940_INTCfg(AFEINTC_1, AFEINTSRC_ALLINT, bTRUE);
53     AD5940_INTCfg(AFEINTC_0, AFEINTSRC_DATAFIFOTHRESH, bTRUE);
54     AD5940_INTCClrFlag(AFEINTSRC_ALLINT);
55     /* Step4: Reconfigure GPIO */
56     gpio_cfg.FuncSet = GP0_INT|GP2_SYNC;
57     gpio_cfg.InputEnSet = AGPIO_Pin2;
58     gpio_cfg.OutputEnSet = AGPIO_Pin0|AGPIO_Pin2;
59     gpio_cfg.OutVal = 0;
60     gpio_cfg.PullEnSet = 0;
61     AD5940_AGPIOcfg(&gpio_cfg);
62     AD5940_SleepKeyCtrls(SLPKEY_UNLOCK);           /* Allow AFE to enter sleep mode. */
63     return 0;
64 }
65 void AD5940_RCal(void) {
66     AD5940PlatformCfg();
67     AppBATCfg_Type *pBATCfg;
68     AppBATGetCfg(&pBATCfg);
69     maxfreq = pBATCfg->SweepCfg.SweepStop;
70     AppBATInit(AppBuff, APPBUFF_SIZE);              /* Initialize BAT application*/
71     AppBATCtrl(BATCTRL_MRCAL, 0);                  /* Measure RCAL each point in sweep */
72 }
73 void AD5940_Main(void) {
74     uint32_t temp;
75     AD5940PlatformCfg();
76     AppBATInit(AppBuff, APPBUFF_SIZE);              /* Initialize BAT application*/
77     AppBATCtrl(BATCTRL_START, 0);
78     while(!meas_stat) {

```

```

79     /* Check if interrupt flag which will be set when interrupt occurred. */
80     if(AD5940_GetMCUIntFlag()) {
81         AD5940_ClrMCUIntFlag();           /* Clear this flag */
82         temp = APPBUFF_SIZE;
83         AppBATISR(AppBuff, &temp);      /* Provide a buffer to store data sent*/
84         AD5940_Delay10us(100000);
85         BATShowResult(AppBuff, temp);    /* Print measurement results over UART*/
86
87         AD5940_SEQMmrTrig(SEQID_0);     /* Trigger next measurement by MMR write*/
88     }
89     meas_stat = 0;
90     AppBATCtrl(BATCTRL_STOPSYNC, 0);
91     AppBATCtrl(BATCTRL_SHUTDOWN, 0);
92 }
```

Listing 3. Kode AD5940Main.c untuk firmware STM32

L.2 Source Code GUI

L.2.1 Program Master.py

```

1 """
2 ****
3 * @file      Master.py
4 * @author    Riko Kumara
5 * @version   V1.1
6 * @date      15-May-2024
7 * @brief     This file provides is the Master file
8 ****
9 """
10 from GUI_Master import RootGUI
11 from Serial_Com_ctrl import SerialCtrl
12 from Data_Com_ctrl import DataMaster
13 # Initiate the Root class that will manage the other classes
14 MySerial = SerialCtrl()
15 MyData = DataMaster()
16 RootMaster = RootGUI(MySerial, MyData)
17 # Start the Graphic User Interface adn loop
18 RootMaster.root.mainloop()
```

Listing 4. Kode Master.py untuk GUI

L.2.2 Program GUI_Master.py

```

1 """
2 ****
3 * @file      GUI_Master.py
4 * @author    Riko Kumara
5 * @version   V1.0
6 * @date      20-June-2024
7 * @brief     This file provides all the GUI management and handler
8 ****
9 """
10 from tkinter import *
11 from tkinter import messagebox
12 from tkinter import ttk
13 from tkinter import filedialog
```

```

14 | from openpyxl import Workbook, load_workbook
15 | from openpyxl.styles import Alignment
16 | from mpl_interactions import (panhandler, zoom_factory)
17 | from datetime import datetime
18 | from PIL import Image, ImageTk
19 | import os
20 | import math
21 | import threading
22 | import numpy as np
23 | import matplotlib.pyplot as plt # pip install matplotlib
24 | from matplotlib.backends.backend_tkagg import (FigureCanvasTkAgg, NavigationToolbar2Tk)
25 | from matplotlib.figure import Figure
26 | from matplotlib.lines import Line2D
27 |
28 | class RootGUI():
29 |     def __init__(self, serial, data):
30 |         """
31 |             Initializing the root GUI and other components of the program
32 |         """
33 |         self.root = Tk()
34 |         self.root.title("Electrochemical Impedance Spectroscopy")
35 |         self.root.geometry("1440x790")
36 |         self.root.config(bg="white")
37 |         self.serial = serial
38 |         self.data = data
39 |         # Passing each other class
40 |         self.info = InfoGUI(self.root, self.serial, self.data)
41 |         self.dis = DisGUI(self.root, self.serial, self.data)
42 |         self.mbox = MsgBoxGUI(self.root, self.data, self.dis)
43 |         self.set = SetGUI(self.root, self.serial, self.data, self.dis, self.mbox)
44 |         self.meas = MeasGUI(self.root, self.serial, self.data, self.info, self.mbox,
45 |                             self.dis, self.set)
46 |         self.com = ComGui(self.root, self.serial, self.data, self.mbox, self.meas,
47 |                           self.info, self.set)
48 |         self.root.protocol("WM_DELETE_WINDOW", self.close_window)
49 |
50 |     def close_window(self):
51 |         self.root.destroy()
52 |         self.serial.SerialClose(self)
53 |         self.serial.threading = False
54 |
55 | # Class to setup and create the communication manager with MCU
56 | class ComGui():
57 |     def __init__(self, root, serial, data, mbox, meas, info, set):
58 |         self.frame = LabelFrame(root, text="COM Manager",
59 |                                font=('Helvetica 9 bold'), padx=5, pady=5, bg="white")
60 |         self.label_com = Label(self.frame, text="Available Port(s) : ",
61 |                               bg="white", width=15, height=2, anchor="w")
62 |         self.label_bd = Label(self.frame, text="Baud Rate: ",
63 |                               bg="white", width=15, height=2, anchor="w")
64 |         self.baudOptionMenu()
65 |         self.ComOptionMenu()
66 |         self.btn_refresh = Button(self.frame, text="Refresh",
67 |                                   width=10, command=self.com_refresh)
68 |         self.btn_connect = Button(self.frame, text="Connect",
69 |                                   width=10, state="disabled", command=self.
70 |                                         serial_connect)
71 |         self.publishCom()

```

```

69
70     def publishCom(self):
71         """
72             Method to display all the Widget of the main frame
73         """
74
75     def ComOptionMenu(self):
76         """ Method to Get the available COMs connected to the PC
77             and list them into the drop menu """
78         self.serial.getCOMList()
79         self.clicked_com = StringVar()
80         self.clicked_com.set(self.serial.com_list[0])
81         self.drop_com = OptionMenu(self.frame, self.clicked_com, *self.serial.com_list,
82                                     command=self.connect_ctrl)
83         self.drop_com.config(width=8)
84
85     def baudOptionMenu(self):
86         """ Method to list all the baud rates in a drop menu """
87         self.clicked_bd = StringVar()
88         bds = ["115200"]
89         self.clicked_bd .set(bds[0])
90         self.drop_baud = OptionMenu(
91             self.frame, self.clicked_bd, *bds, command=self.connect_ctrl)
92         self.drop_baud.config(width=8)
93
94     def connect_ctrl(self, widget):
95         """ Mehtod to keep the connect button disabled if all the
96             conditions are not cleared """
97         if "--" in self.clicked_bd.get() or "--" in self.clicked_com.get():
98             self.btn_connect["state"] = "disabled"
99         else:
100             self.btn_connect["state"] = "active"
101
102     def com_refresh(self):
103         # Get the Widget destroyed
104         self.drop_com.destroy()
105         print("Refresh COM List") # Refresh the list of available Coms
106         self.ComOptionMenu() # Publish the this new droplet
107         self.drop_com.grid (column=1, row=0, padx=10, pady=self.pady) # Just in case to
108         secure the connect logic
109         logic = []
110         print(f"Available COMs: \n{self.serial.com_list}")
111         self.connect_ctrl(logic)
112
113     def serial_connect(self):
114         """ Method that Updates the GUI during connect / disconnect status
115             Manage serials and data flows during connect / disconnect status """
116         global swpstrt_val, swpstp_val, swppts_val
117         if self.btn_connect["text"] in "Connect":
118             # Start the serial communication
119             self.serial.SerialOpen(self)
120             # If connection established move on
121             if self.serial.ser.status:
122                 self.serial.t1 = threading.Thread(
123                     target=self.serial.SerialSync, args=(self, ), daemon=True)
124                 self.serial.t1.start()
125             else:pass
126         else: # Set the EIS-device to idle condition
127             self.serial.SerialClose(self)

```

```

126         self.data.ClearData()      # Update the buttons parameters
127
128 # Class to setup the Measurement Settings
129 class MeasGUI():
130     def __init__(self, root, serial, data, info, mbox, dis, set):
131         ''' Initialize main Widgets for measurement GUI
132         '''
133         # Build MeasGUI Static Elements
134         self.frame = LabelFrame(root, text="Measurement Settings",
135                             font=('Helvetica 9 bold'), padx=5, pady=5, bg="white")
136         # Extending the GUI
137         self.publishMeas()
138     def publishMeas(self):
139         '''
140             Method to display all the widgets
141         '''
142     def validateEntry(self, txt, newtext):
143         if len(newtext) > 5:
144             return False
145         if newtext == "":
146             return True # Allow deletion
147         try:
148             self.inputstr = float(newtext) # Try to convert to float
149             return True
150         except ValueError:
151             return False
152
153     def save_param(self):
154         global swpstrt_val, swpstp_val, swppts_val, swpstrt_str, swpstp_str, swppts_str
155         swpstrt_str = self.write_sweep_start.get().replace(" ", "")
156         swpstp_str = self.write_sweep_stop.get().replace(" ", "")
157         swppts_str = self.write_sweep_points.get().replace(" ", "")
158         self.data.ClearData()
159         if self.start >0 or self.stop >0 or self.point >0:
160             self.serial.t1 = threading.Thread(
161                 target=self.serial.SerialSaveParam, args=(self, ), daemon=True)
162             self.serial.t1.start()
163
164     def start_meas(self):
165         self.serial.t1 = threading.Thread(
166             target=self.serial.Measurement, args=(self, ), daemon=True)
167         self.serial.t1.start()
168
169     def stop_meas(self):
170         self.btn_stop_meas ["state"] = "disabled"
171         global cnt
172         global meas_mode
173         meas_mode = 0
174         self.serial.ser.write(self.data.idle.encode())
175         self.serial.t1 = threading.Thread(
176             target=self.serial.setIdle, args=(self, ), daemon=True)
177         self.serial.t1.start()
178
179 # Class to display the Measurement Settings and Results
180 class InfoGUI():
181     def __init__(self, root, serial, data):
182         '''
183             Initializing the info GUI and other components of the program
184         '''

```

```

185         self.frame = LabelFrame(root, text="Measurement Info", font=('Helvetica 9 bold'))
186         , padx=5, pady=5, bg="white")
187         # Create Label for the Measurement Parameters
188         # Create Label for Displaying the Valuesm
189         # Create Label for the Parameters Unit
190         self.publishInfo()
191
192     def publishInfo(self):
193         """
194             Publishing and update all of the info GUI static component
195         """
196
197     class SetGUI():
198         def __init__(self, root, serial, data, dis, mbox):
199             """
200                 Initialize main Widgets for analyzer GUI
201             """
202             self.PlotOptionMenu()
203             self.publishFunctionBar()
204             self.dis.publishDis(temp_title)
205
206         def plot_display(self):
207             global stat_scat, stat_line, pscat, pline
208             stat_scat = pscat.get()
209             stat_line = pline.get()
210             print(f"Scat: {stat_scat}, Line: {stat_line}\n")
211             self.dis.publishDis(temp_title)
212             if plot_mode == 0:
213                 self.dis.plotParam()
214
215         def publishFunctionBar(self):
216             """ Publishing and update all of the info GUI static component """
217         def PlotOptionMenu(self):
218             self.clicked_pltyp = StringVar()
219             self.clicked_pltyp.set(self.plot_type[0])
220             self.drop_plot = OptionMenu(self.frame, self.clicked_pltyp, *self.plot_type,
221                                         command=self.plot_ctrl)
222             self.drop_plot.config(width=6)
223         def plot_ctrl(self, value):
224             global plot_mode
225             if "Nyquist" in value:
226                 plot_mode = 0
227                 self.btn_analyze.config(state="active")
228                 self.dis.publishDis(3)
229             elif "Bode" in value:
230                 plot_mode = 1
231                 self.btn_analyze.config(state="disabled")
232                 self.dis.publishDis(4)
233         def analyze_plot(self):
234             try:
235                 Rs = self.data.find_Rs(self)
236                 if Rs > 0:
237                     self.publishFunctionBar()
238                     self.dis.plotParam()
239                 else: pass
240                 RsRct = self.data.find_Rct(self)
241                 if RsRct > 0:
242                     Rct = round(RsRct-Rs, 3)
243                     fCdl, xCdl, yCdl = self.data.find_Cdl(self)

```

```

242         Cdl = 1000000/(2*(22/7)*fCdl*Rct)
243         Cdl = round(Cdl, 3)
244     else: pass
245     self.publishFunctionBar()
246     self.dis.plotParam()
247     plot_param = 1
248 except Exception as e: pass
249     for col, width in column_widths.items():
250         ws.column_dimensions[col].width = width
251     wb.save(self.filename)
252 else:pass
253 except Exception as e:pass
254 # Class to setup the data display GUI
255 class DisGUI():
256     def __init__(self, root, serial, data):
257         ''' Initializing the display graph GUI and the objects that will be used to
258         manage the GUI '''
259         # Master Frame controls
260         self.frame = LabelFrame(root, text="Measurement Graph",
261                             font=('Helvetica 9 bold'), padx=5, pady=5, bg="white")
262         self.publishDis(2)
263     def zoom_reset(self):
264         self.publishDis(temp_title)
265     def publishDis(self, title):
266         ''' Publishing and update all of the display graph GUI static component '''
267         self.frame.grid(row=0, column=3, rowspan=18, columnspan=6, padx=5, pady=5,
268         sticky="N")
269         self.f = Figure(figsize=(9.46,5.52), dpi=100)
270         self.a = self.f.add_subplot(111)
271         self.a.grid()
272     try:
273         # Nyquist Plot
274         if plot_mode == 0:
275             self.a.set_title(self.plot_title[title])
276             if stat_scat:
277                 self.a.scatter(x, y, color='blue', zorder=5)
278             if stat_line:
279                 self.a.plot(x, y, color='blue', zorder=5)
280             self.axhline(y=0, color='gray', linestyle='-', linewidth=1)
281             temp_title = title
282             self.plotParam()
283         # Bode Plot
284         elif plot_mode == 1:
285             self.b = self.a.twinx()
286             self.a.set_xlabel('Frequency (Hz)')
287             self.a.set_title(self.plot_title[title])
288             z = [math.sqrt(real**2 + imag**2) for real, imag in zip(x, y)]
289             ph= [math.degrees(math.atan2(imag, real)) for real, imag in zip(x, y)]
290             if stat_scat:
291                 self.a.scatter(fr, z, color='blue', zorder=5, label='Magnitude')
292                 self.b.scatter(fr, ph, color='orange', zorder=5, label='Phase')
293                 custom_legend = [Line2D([0], [0], marker='o', color='w',
294                                         markerfacecolor='blue', markersize=6, label='Magnitude'),
295                                         Line2D([0], [0], marker='o', color='w',
296                                         markerfacecolor='orange', markersize=6, label='Phase')]
297                 self.a.legend(handles=custom_legend, loc='upper right', fontsize=8)
298             if stat_line:
299                 self.a.plot(fr, z, color='blue', zorder=5, label='Magnitude')
300                 self.b.plot(fr, ph, color='orange', zorder=5, label='Phase')

```

```

297         custom_legend = [Line2D([0], [0], marker='o', color='w',
298                               markerfacecolor='blue', markersize=6, label='Magnitude'),
299                               Line2D([0], [0], marker='o', color='w',
300                               markerfacecolor='orange', markersize=6, label='Phase')]
301         self.a.legend(handles=custom_legend, loc='upper right', fontsize=8)
302         self.a.set_xscale('log')
303         temp_title = title
304     except Exception as e:pass
305     self.canvas = FigureCanvasTkAgg(self.f, master=self.frame)
306     self.canvas.get_tk_widget().grid(row=0, column=1, rowspan=18, padx=0, pady=0,
307                                     sticky="NE")
308     self.disconnect_zoom = zoom_factory(self.a)
309     self.pan_handler = panhandler(self.f, 1)
310     ''''''
311 def plotParam(self):
312     ''' Displaying the critical parameter point on the graph '''
313     if stat_line==0 and stat_scat==0: pass
314     else:
315         try:
316             if Rs > 0:
317                 self.a.plot(Rs, 0, 'rs', markersize=8, zorder=6)
318                 self.a.annotate('Rs', (Rs, 0), textcoords="offset points", xytext
319 = (10, -15), ha='left', fontsize=12)
320                 self.a.annotate(f'({Rs}, {0})', (Rs, 0), textcoords="offset points",
321 xytext=(25, -15), ha='left', fontsize=8)
322                 self.canvas = FigureCanvasTkAgg(self.f, master=self.frame)
323                 self.canvas.get_tk_widget().grid(row=0, column=1, rowspan=18, padx
324 = 0, pady=0, sticky="NE")
325             else:
326                 pass
327             if Rct > 0:
328                 self.a.plot(RsRct, 0, 'rs', markersize=8, zorder=6)
329                 self.a.annotate('Rs+Rct', (RsRct, 0), textcoords="offset points",
330 xytext=(-10, -15), ha='right', fontsize=12)
331                 self.a.annotate(f'({RsRct}, {0})', (RsRct, 0), textcoords="offset
332 points", xytext=(-10, -15), ha='left', fontsize=8)
333                 self.canvas = FigureCanvasTkAgg(self.f, master=self.frame)
334                 self.canvas.get_tk_widget().grid(row=0, column=1, rowspan=18, padx
335 = 0, pady=0, sticky="NE")
336             else:
337                 pass
338         except Exception as e:
339             pass
340 class MsgBoxGUI():
341     ''' Class to Display Command Info '''
342     def __init__(self, root, data, dis):
343         ''' Initializing the message box GUI and the objects that will be used to manage
the GUI '''

```

```

343     self.frame = LabelFrame(text="Message Box",
344                             font=('Helvetica 8 bold'), padx=5, pady=2, bg="white")
345     self.btn_clear_msg = Button(self.frame, text="Clear Message", width=12, command
346 =self.clear_msg)
347     self.publishMsgBox()
348     '////
349 def publishMsgBox(self, tag="info"):
350     ''' Publishing and update all of the Message Box static component '''
351     # Setup so the new message is shown in a new line
352     self.text.delete(1.0, END)
353     # Setup the widget grid
354     self.frame.grid (row=18, column=3, columnspan=7, padx=5, pady=5, sticky="W")
355     self.btn_clear_msg.grid (row=4, column=0, columnspan=1, padx=5, pady=2, sticky=
356 "SE")
357     self.text.config (state="normal")
358     self.text.insert(END, self.text_msg, tag)
359     self.text.config(state="disabled")
360     self.text.see(END)
361     self.text.grid (row=0, column=0, rowspan=5, sticky="NW")
362     '////
363 def clear_msg(self):
364     ''' Clearing the Message Box '''
365     self.text.config(state="normal")
366     self.text.delete("1.0", END)
367     self.text.config(state="disabled")
368     self.text.grid (row=0, column=0, rowspan=5, sticky="NW")
369     '////
370 if __name__ == "__main__":
371     RootGUI()
372     ComGui()
373     MeasGUI()
374     InfoGUI()
375     SetGUI()
376     DisGUI()

```

Listing 5. Kode GUI_Master.py untuk GUI

L.2.3 Program Serial_Com_ctrl.py

```

1 /**
2 ****
3 * @file      Serial_Com_ctrl.py
4 * @author    Riko Kumara
5 * @version   V1.5
6 * @date      13-June-2024
7 * @brief     This file provides all the Serial Connection and Communication procedure
8 ****
9 /**
10 import serial.tools.list_ports
11 import time
12
13 class SerialCtrl():
14     def __init__(self):
15         self.sync_cnt = 50
16         self.ser = serial.Serial()
17         self.Batt = True
18         self.freq = 0
19         self.real = 0

```

```

20         self.imaj = 0
21 def getCOMList(self):
22     ports = serial.tools.list_ports.comports()
23     self.com_list = [com[0] for com in ports]
24     self.com_list.insert(0, "-")
25 def SerialOpen(self, ComGUI):
26     try:
27         self.ser.is_open
28     except:
29         PORT = ComGUI.clicked_com.get()
30         BAUD = ComGUI.clicked_bd.get()
31         self.ser = serial.Serial()
32         self.ser.baudrate = BAUD
33         self.ser.port = PORT
34         self.ser.timeout = 0.1
35     try:
36         if self.ser.is_open:
37             print("Already Open")
38             self.ser.status = True
39         else:
40             PORT = ComGUI.clicked_com.get()
41             BAUD = ComGUI.clicked_bd.get()
42             self.ser = serial.Serial()
43             self.ser.baudrate = BAUD
44             self.ser.port = PORT
45             self.ser.timeout = 0.01
46             self.ser.open()
47             self.ser.status = True
48     except:
49         self.ser.status = False
50 def SerialClose(self, gui):
51     try:
52         self.ser.is_open
53         self.ser.write(gui.data.idle.encode())
54         time.sleep(0.2)
55         self.ser.close()
56         self.ser.status = False
57     except:
58         self.ser.status = False
59 def SerialSync(self, gui):
60     self.threading = True
61     global swpstr
62     global swpstp
63     global swppts
64     cnt = 0
65     gui.meas.sync_status["text"] = ".Sync. "
66     gui.meas.sync_status["fg"] = "orange"
67     time.sleep(0.3)
68     while self.threading:
69         try:
70             self.ser.write(gui.data.sync.encode())
71             gui.meas.sync_status["text"] = ".Sync. ."
72             gui.meas.sync_status["fg"] = "orange"
73             time.sleep(0.2)
74             gui.data.RowMsg = self.ser.readline()
75             gui.data.DecodeMsg()
76             if gui.data.msg[0] == 'P':
77                 gui.meas.sync_status["text"] = ".Loading Param.   "
78                 gui.meas.sync_status["fg"] = "orange"

```

```

79         swpstr = float(gui.data.msg[1])
80         swpstp = float(gui.data.msg[2])
81         swppts = int(gui.data.msg[3])
82         self.ser.write(gui.data.idle.encode())
83         time.sleep(0.3)
84         gui.meas.sync_status["text"] = ".Loading Param. . ."
85         gui.meas.sync_status["fg"] = "orange"
86         gui.data.RowMsg = self.ser.readline()
87         gui.data.DecodeMsg()
88         while gui.data.msg[0] != '=':
89             self.ser.write(gui.data.idle.encode())
90             time.sleep(0.2)
91             gui.meas.sync_status["text"] = ".Loading Param. . ."
92             gui.meas.sync_status["fg"] = "orange"
93             gui.data.RowMsg = self.ser.readline()
94             gui.data.DecodeMsg()
95             time.sleep(0.3)
96         gui.mbox.text_msg = ". . . . . Successfully Load All Default
Parameters\n"
97         gui.mbox.publishMsgBox("success")
98         gui.meas.sync_status["text"] = "OK"
99         gui.meas.sync_status["fg"] = "green"
100        gui.meas.btn_save_param["state"] = "active"
101        gui.meas.btn_start_meas["state"] = "active"
102        gui.info.value_swpstrt.config(text=swpstr)
103        gui.info.value_swpstp.config(text=swpstp)
104        gui.info.value_swppts.config(text=swppts)
105        gui.swpstrt_val = swpstr
106        gui.swpstp_val = swpstp
107        gui.swppts_val = swppts
108        gui.data.msg = []
109        time.sleep(0.5)
110        self.threading = False
111        print(gui.swpstrt_val)
112        print(gui.swpstp_val)
113        print(gui.swppts_val)
114        print(f"Synchronizing Done...")
115        break
116    else:
117        time.sleep(0.3)
118        gui.meas.sync_status["text"] = ".Sync. . ."
119        gui.meas.sync_status["fg"] = "orange"
120        time.sleep(0.3)
121    except Exception as e:
122        time.sleep(0.1)
123        #print(e)
124    cnt += 1
125    if self.threading == False:
126        cnt = 0
127        break
128    if cnt > self.sync_cnt:
129        gui.meas.sync_status["text"] = "failed"
130        gui.meas.sync_status["fg"] = "red"
131        time.sleep(0.5)
132        cnt = 0
133        if self.threading == False:
134            break
135    def SerialSaveParam(self, gui):
136        global swpstrt_val, swpstp_val, swppts_val

```

```

137     print("Synchronizing...")
138     gui.btn_start_meas["state"] = "disabled"
139     gui.btn_stop_meas["state"] = "disabled"
140     gui.btn_save_param["state"] = "disabled"
141     gui.label_meas_param["text"] = ". . . . ."
142     cnt = 0
143     self.P1stat = True
144     self.P2stat = True
145     self.P3stat = True
146     self.threading = True
147     gui.sync_status["text"] = " .Sync. "
148     gui.sync_status["fg"] = "orange"
149     time.sleep(0.1)
150     while self.threading:
151         try:
152             self.ser.write(gui.data.write.encode())
153             gui.sync_status["text"] = " .Sync. . "
154             gui.sync_status["fg"] = "orange"
155             time.sleep(0.2)
156             gui.data.RowMsg = self.ser.readline()
157             gui.data.DecodeMsg()
158             if gui.data.msg[0] == 'K':
159                 gui.label_save_param["text"] = ". .Init. ."
160                 gui.label_save_param["fg"] = "green"
161                 time.sleep(0.2)
162                 gui.sync_status["text"] = "OK"
163                 gui.sync_status["fg"] = "green"
164                 if gui.start > 0:
165                     gui.mbox.text_msg = ". . . . . Sending Sweep Start Freq
Parameter\n"
166                     gui.mbox.publishMsgBox("process")
167                     while(self.P1stat):
168                         try:
169                             self.plmsg = "#1#{:2f}#\n".format(gui.start)
170                             self.ser.write(self.plmsg.encode())
171                             time.sleep(0.8)
172                             gui.label_save_param["text"] = ". Writing. "
173                             gui.label_save_param["fg"] = "orange"
174                             gui.data.RowMsg = self.ser.readline()
175                             gui.data.DecodeMsg()
176                             if gui.data.msg[0] == "{:.2f}".format(gui.start) :
177                                 try:
178                                     swpststrt_val = gui.start
179                                     gui.info.value_swpstrt.config(text = swpststrt_val
)
180                                     gui.label_save_param.config(text="SAVED", fg="green")
181                                     gui.mbox.text_msg = ". . . . . Successfully
Saving Sweep Start Freq Parameter\n"
182                                     gui.mbox.publishMsgBox("success")
183                                     self.P1stat = False
184                                 except:
185                                     time.sleep(0.1)
186                                 else:
187                                     time.sleep(0.4)
188                                     gui.label_save_param["text"] = " .Writing ."
189                                     gui.label_save_param["fg"] = "orange"
190                                 except:
191                                     print("P1 error")

```

```

192             time.sleep(0.5)
193         if gui.stop > 0:
194             gui.mbox.text_msg = "..... Sending Sweep Stop Freq
Parameter\n"
195             gui.mbox.publishMsgBox("process")
196             while(self.P2stat):
197                 try:
198                     self.p2msg = "#2#{:.2f}#\n".format(gui.stop)
199                     self.ser.write(self.p2msg.encode())
200                     time.sleep(0.8)
201                     gui.label_save_param["text"] = ". Writing. "
202                     gui.label_save_param["fg"] = "orange"
203                     gui.data.RowMsg = self.ser.readline()
204                     gui.data.DecodeMsg()
205                     if gui.data.msg[0] == "{:.2f}".format(gui.stop):
206                         try:
207                             swpstp_val = gui.stop
208                             gui.info.value_swpstp.config(text = swpstp_val)
209                             gui.label_save_param.config(text="SAVED", fg=
green")
210                             gui.mbox.text_msg = "..... Successfully
Saving Sweep Stop Freq Parameter\n"
211                             gui.mbox.publishMsgBox("success")
212                             self.P2stat = False
213                         except:
214                             time.sleep(0.1)
215                         else:
216                             time.sleep(0.4)
217                             gui.label_save_param["text"] = ".Writing ."
218                             gui.label_save_param["fg"] = "orange"
219                         except:
220                             print("P2 error")
221                             time.sleep(0.5)
222                     if gui.point > 0:
223                         gui.mbox.text_msg = "..... Sending Sweep Points Parameter\
n"
224                         gui.mbox.publishMsgBox("process")
225                         while(self.P3stat):
226                             try:
227                                 self.p3msg = "#3#{:05d}#\n".format(gui.point)
228                                 self.ser.write(self.p3msg.encode())
229                                 time.sleep(0.8)
230                                 gui.label_save_param["text"] = ". Writing. "
231                                 gui.label_save_param["fg"] = "orange"
232                                 gui.data.RowMsg = self.ser.readline()
233                                 gui.data.DecodeMsg()
234                                 if gui.data.msg[0] == str(gui.point):
235                                     try:
236                                         swppts_val = gui.point
237                                         gui.info.value_swppts.config(text = swppts_val)
238                                         gui.label_save_param.config(text="SAVED", fg=
green")
239                                         gui.mbox.text_msg = "..... Successfully
Saving Sweep Points Parameters\n"
240                                         gui.mbox.publishMsgBox("success")
241                                         self.P3stat = False
242                                     except:
243                                         time.sleep(0.1)
244                                     else:

```

```

245             time.sleep(0.4)
246             gui.label_save_param["text"] = " .Writing ."
247             gui.label_save_param["fg"] = "orange"
248         except:
249             print("P3 error")
250             time.sleep(0.1)
251         if (gui.start>0 and self.P1stat==False) or (gui.stop>0 and self.
252 P2stat==False) or (gui.point>0 and self.P3stat==False):
253             self.threading = False
254             self.ser.write(gui.data.idle.encode())
255             time.sleep(0.1)
256             gui.data.RowMsg = self.ser.readline()
257             gui.data.DecodeMsg()
258             while gui.data.msg[0] != '=':
259                 self.ser.write(gui.data.idle.encode())
260                 time.sleep(0.3)
261                 gui.data.RowMsg = self.ser.readline()
262                 gui.data.DecodeMsg()
263                 time.sleep(0.4)
264             gui.mbox.text_msg = ". . . . Successfully Saving All
Measurement Parameters\n"
265             print("Successfully Saving All Measurement Parameters...")
266             gui.mbox.publishMsgBox("info")
267             gui.sync_status["text"] = "OK"
268             gui.sync_status["fg"] = "green"
269             gui.btn_start_meas["state"] = "active"
270             gui.btn_save_param["state"] = "active"
271         else:
272             gui.sync_status["text"] = " .Sync. . ."
273             gui.sync_status["fg"] = "orange"
274         except Exception as e:
275             print("Write Req Error")
276             time.sleep(0.1)
277         cnt += 1
278         if self.threading == False:
279             cnt = 0
280             break
281         if cnt > self.sync_cnt:
282             self.threading = False
283             self.ser.write(gui.data.idle.encode())
284             time.sleep(.2)
285             self.ser.write(gui.data.idle.encode())
286             gui.label_save_param["text"] = "failed"
287             gui.label_save_param["fg"] = "red"
288             time.sleep(0.3)
289             cnt = 0
290             self.threading == False
291             break
292     def Measurement(self, gui):
293         print("Synchronizing...")
294         self.sync_cnt = 200
295         gui.set.btn_analyze ["state"] = "disabled"
296         gui.set.btn_load_xlsx ["state"] = "disabled"
297         gui.btn_start_meas ["state"] = "disabled"
298         gui.btn_stop_meas ["state"] = "disabled"
299         gui.btn_save_param ["state"] = "disabled"
300         gui.label_save_param["text"] = ". . . ."
301         self.threading = True
302         self.RCAL = False

```

```

302     self.BATT = True
303     self.MeasStat = True
304     self.freq = 0
305     self.real = 0
306     self.imag = 0
307     self.wait = 0
308     cnt = 0
309     gui.sync_status["text"] = " .Sync. "
310     gui.sync_status["fg"] = "orange"
311     gui.data.ClearTemp()
312     time.sleep(0.1)
313     self.ser.write(gui.data.meas.encode())
314     while self.threading:
315         try:
316             gui.sync_status["text"] = " .Sync. . "
317             gui.sync_status["fg"] = "orange"
318             time.sleep(0.1)
319             gui.data.RowMsg = self.ser.readline()
320             gui.data.DecodeMsg()
321             if gui.data.msg[0] == 'K':
322                 time.sleep(0.1)
323                 gui.sync_status["text"] = "OK"
324                 gui.sync_status["fg"] = "green"
325                 while self.MeasStat:
326                     gui.data.RowMsg = self.ser.readline()
327                     gui.data.DecodeMsg()
328                     time.sleep(0.2)
329                     if gui.data.msg[0] == 'RC':
330                         self.RCAL = True
331                         print("Initiating RCal Measurement Procedure...")
332                         gui.mbox.text_msg = f". . . Initiating RCal Measurement
Procedure\n"
333                         gui.mbox.publishMsgBox("process")
334                         gui.label_meas_param["text"] = ". Init. ."
335                         gui.label_meas_param["fg"] = "green"
336                         self.tempFreq = float('0.0')
337                         while self.RCAL:
338                             gui.data.RowMsg = self.ser.readline()
339                             gui.data.DecodeMsg()
340                             time.sleep(0.1)
341                             if gui.data.msg[0] == 'R':
342                                 gui.label_meas_param["text"] = ". Receive. ."
343                                 gui.label_meas_param["fg"] = "green"
344                                 try:
345                                     self.freq = float(gui.data.msg[1].strip())
346                                     self.real = float(gui.data.msg[2].strip())
347                                     self.imag = float(gui.data.msg[3].strip())
348                                     if self.freq > self.tempFreq:
349                                         gui.data.fTemp.append(self.freq)
350                                         gui.data.rTemp.append(self.real)
351                                         gui.data.iTemp.append(self.imag)
352                                         gui.mbox.text_msg = f". . . . . RCal . .
Freq: {self.freq}Hz, RCalVolt: ({self.real}, {self.imag})V\n"
353                                         gui.mbox.publishMsgBox("normal")
354                                     else:
355                                         time.sleep(0.1)
356                                         gui.label_meas_param["text"] = ".Receive ."
357                                         gui.label_meas_param["fg"] = "green"
358                                         self.tempFreq = self.freq

```

```

359         except ValueError:
360             print("ERR CONVERTING")
361     elif gui.data.msg[0] == '=':
362         try:
363             gui.data.fRCal = gui.data.fTemp
364             gui.data.rRCal = gui.data.rTemp
365             gui.data.iRCal = gui.data.iTemp
366             gui.set.clear_graph()
367             gui.dis.f.clf()
368         except:
369             print("ERR PLOT")
370             gui.mbox.text_msg = f". . . Failed to Show
371             Graph. . . . \n"
372             gui.mbox.publishMsgBox("warning")
373             gui.label_meas_param["text"] = "DONE"
374             gui.label_meas_param["fg"] = "green"
375             self.RCAL = False
376             gui.mbox.text_msg = f". . . Successfully Receiving
377             All RCal Measurement Data\n"
378             gui.mbox.publishMsgBox("info")
379             break
380         else:
381             time.sleep(0.15)
382             if self.wait == 0:
383                 gui.label_meas_param["text"] = ". Waiting"
384                 gui.label_meas_param["fg"] = "orange"
385                 self.wait = 1
386             elif self.wait == 1:
387                 gui.label_meas_param["text"] = "Waiting ."
388                 gui.label_meas_param["fg"] = "orange"
389                 self.wait = 0
390         elif gui.data.msg[0] == 'IM':
391             gui.set.btn_reset_zoom ["state"] = "disabled"
392             gui.set.btn_clr_graph ["state"] = "disabled"
393             gui.set.btn_save_graph ["state"] = "disabled"
394             gui.set.btn_save_xlsx ["state"] = "disabled"
395             gui.mbox.text_msg = f". . Initiating Impedance Measurement
396             Procedure\n"
397             gui.mbox.publishMsgBox("process")
398             gui.label_meas_param["text"] = ". Init. ."
399             gui.label_meas_param["fg"] = "green"
400             gui.sync_status["text"] = "OK"
401             gui.sync_status["fg"] = "green"
402             self.freq = float('0.0')
403             self.real = float('0.0')
404             self.imag = float('0.0')
405             gui.set.clear_graph()
406             gui.data.ClearTemp()
407             while self.BATT:
408                 gui.data.RowMsg = self.ser.readline()
409                 gui.data.DecodeMsg()
410                 time.sleep(0.1)
411                 if gui.data.msg[0] == 'I':
412                     gui.label_meas_param["text"] = ". Receive ."
413                     gui.label_meas_param["fg"] = "green"
414                     try:
415                         self.freq = float(gui.data.msg[1].strip())
416                         self.real = float(gui.data.msg[2].strip())
417                         self.imag = float(gui.data.msg[3].strip())

```

```

415             if self.freq > self.tempFreq:
416                 if self.real > float('10000.0'):
417                     self.real = self.tempReal + 0.5
418                     self.imag = self.tempImaj
419                 elif self.imag > float('10000.0'):
420                     self.imag = self.tempReal + 0.5
421                     self.real = self.tempImaj
422                 gui.data.fTemp.append(self.freq)
423                 gui.data.rTemp.append(self.real)
424                 gui.data.iTemp.append(self.imag)
425                 gui.mbox.text_msg = f"..... Batt . .
426 Freq: {self.freq}Hz, Impedance: ({self.real}, {self.imag})mOhm\n"
427                 gui.mbox.publishMsgBox("normal")
428                 gui.dis.publishDis(0)
429             else:
430                 time.sleep(0.1)
431                 gui.label_meas_param["text"] = ". Receive. "
432                 gui.label_meas_param["fg"] = "green"
433                 self.tempFreq = self.freq
434                 self.tempReal = self.real
435                 self.tempImaj = self.imag
436             except ValueError:
437                 print("ERR CONVERTING")
438             elif gui.data.msg[0] == '=':
439                 try:
440                     gui.data.fBatt = gui.data.fTemp
441                     gui.data.rBatt = gui.data.rTemp
442                     gui.data.iBatt = gui.data.iTemp
443                     gui.dis.f.clf()
444                     gui.dis.publishDis(1)
445                 except:
446                     print("ERR PLOT")
447                     gui.mbox.text_msg = f".... Failed to Show
448 Graph . . . .\n"
449                     gui.mbox.publishMsgBox("warning")
450                     gui.label_meas_param["text"] = "DONE"
451                     gui.label_meas_param["fg"] = "green"
452                     gui.btn_start_meas["text"] = "START"
453                     gui.btn_start_meas["state"] = "active"
454                     gui.btn_save_param["state"] = "active"
455                     gui.sync_status["text"] = "....."
456                     self.threading = False
457                     self.BATT = False
458                     self.MeasStat = False
459                     gui.mbox.text_msg = f".... Successfully Receiving
460 All Impedance Measurement Data\n"
461                     gui.mbox.publishMsgBox("info")
462
463                     gui.set.btn_analyze ["state"] = "active"
464                     gui.set.btn_reset_zoom ["state"] = "active"
465                     gui.set.btn_clr_graph ["state"] = "active"
466                     gui.set.btn_save_graph ["state"] = "active"
467                     gui.set.btn_save_xlsx ["state"] = "active"
468                     gui.set.btn_load_xlsx ["state"] = "active"
469                     break
470             else:
471                 time.sleep(0.15)
472                 if self.wait == 0:
473                     gui.label_meas_param["text"] = ". Waiting"

```

```

471             gui.label_meas_param["fg"] = "orange"
472             self.wait = 1
473         elif self.wait == 1:
474             gui.label_meas_param["text"] = "Waiting ."
475             gui.label_meas_param["fg"] = "orange"
476             self.wait = 0
477         else:
478             gui.sync_status["text"] = ".Sync. . ."
479             gui.sync_status["fg"] = "orange"
480             time.sleep(0.1)
481     except Exception as e:
482         print("Meas Req Error")
483         time.sleep(0.2)
484     cnt += 1
485     if self.threading == False:
486         break
487     if cnt > self.sync_cnt:
488         self.ser.write(gui.data.meas.encode())
489         time.sleep(0.2)
490         gui.label_meas_param["text"] = "..Resync.."
491         gui.label_meas_param["fg"] = "red"
492         cnt = 0
493     if self.threading == False:
494         break
495 def setIdle(self, gui):
496     self.threading = True
497     while self.threading:
498         gui.data.RowMsg = self.ser.readline()
499         gui.data.DecodeMsg()
500         while True:
501             self.ser.write(gui.data.idle.encode())
502             time.sleep(0.3)
503             gui.data.RowMsg = self.ser.readline()
504             gui.data.DecodeMsg()
505             if gui.data.msg[0] == '=':
506                 self.threading = False
507                 break
508             time.sleep(0.3)
509 if __name__ == "__main__":
510     SerialCtrl()

```

Listing 6. Kode Serial_Com_ctrl.py untuk GUI

L.2.4 Program Data_Com_ctrl.py

```

1 """
2 ****
3 * @file    Data_Com_ctrl.py
4 * @author  Riko Kumara
5 * @version V1.0
6 * @date    13-April-2024
7 * @brief   This file provides all the Data Stream and Calculation
8 ****
9 """
10 import pandas as pd
11
12 class DataMaster():
13     def __init__(self):

```

```

14     """
15     Method to initialize the Data class
16     This classes will manage data transformation within the whole program
17     """
18     # Program Logic to interface with MCU
19     self.sync = "#?\n"
20     self.idle = "#=\n"
21     self.write= "#W#\n"
22     self.meas = "#M#\n"
23     self.rcal = "#R#\n"
24     self.batt = "#B#\n"
25     self.msg = []
26     # Build data
27     self.fBatt = []
28     self.rBatt = []
29     self.iBatt = []
30     self.fRCal = []
31     self.rRCal = []
32     self.iRCal = []
33     self.fTemp = []
34     self.rTemp = []
35     self.iTemp = []
36     self.YData = []
37     self.XData = []
38     """
39     def DecodeMsg(self):
40         """
41             Method used to get the message coming from UART and converted to a python
42             string
43             it is also used to get defferent type of messages based on the Message protocol
44             """
45             temp = self.RowMsg.decode('utf8')
46
47             if len(temp) > 0:
48                 if "#" in temp:
49                     self.msg = temp.split("#")
50                     del self.msg[0]
51                     print(self.msg)
52             """
53
54     def ClearData(self):
55         """Method used to clear the data after each new serial upload"""
56
57         self.RowMsg = ""
58         self.msg = []
59         self.YData = []
60         self.fData = []
61         self.rData = []
62         self.iData = []
63
64         self.fBatt = []
65         self.rBatt = []
66         self.iBatt = []
67
68         self.fRCal = []
69         self.rRCal = []
70         self.iRCal = []
71         """

```

```

72     def ClearTemp(self):
73         self.fTemp = []
74         self.rTemp = []
75         self.iTemp = []
76
77     def find_Rs(self, gui):
78         df = pd.DataFrame({'f': gui.freq, 'x': gui.real, 'y': gui.imaj})
79         dfx = df[df['f'] > 1000]
80         dfneg = (dfx['y'] < 0).any()
81
82         if dfneg:
83             df = df[df['f'] > 10]
84             crossings = []
85             for i in range(1, len(df)):
86                 if df['y'].iloc[i-1] * df['y'].iloc[i] < 0:
87                     crossings.append((df['x'].iloc[i-1], df['x'].iloc[i]))
88
89             if not crossings:
90                 return None
91             x1, x2 = crossings[0]
92             y1 = df[df['x'] == x1]['y'].values[0]
93             y2 = df[df['x'] == x2]['y'].values[0]
94             x_target = x1 + (0 - y1) * (x2 - x1) / (y2 - y1)
95             return round(x_target, 3)
96         else:
97             return 0
98
99     def find_Rct(self, gui):
100        df = pd.DataFrame({'f': gui.freq, 'x': gui.real, 'y': gui.imaj})
101        dff = (df['f'] < 10).any()
102
103        if dff:
104            df_filtered = df[(df['f'] > 2) & (df['f'] < 100)]
105            low_imaj_idx = df_filtered['y'].idxmin()
106            lowest_pts = df_filtered.loc[low_imaj_idx]
107            return round(lowest_pts['x'], 3)
108        else:
109            return 0
110
111    def find_Cdl(self, gui):
112        df = pd.DataFrame({'f': gui.freq, 'x': gui.real, 'y': gui.imaj})
113        df_filtered = df[(df['f'] > 10) & (df['f'] < 2000)]
114        max_imaj_idx = df_filtered['y'].idxmax()
115        max_pts = df_filtered.loc[max_imaj_idx]
116        return round(max_pts, 3)
117
118 if __name__ == "__main__":
119     DataMaster()

```

Listing 7. Kode Data_Com_ctrl.py untuk GUI

L.3 Tabel Hasil Pengukuran

L.3.1 Hasil Pengujian Fungsi Pengukuran

Tabel 1. Hasil pengukuran pengujian fungsi perangkat

n	Frequency	Real (mΩ)	Imaginary
1	0,10	282,678	126,221
2	0,17	268,818	105,237
3	0,27	239,733	52,473
4	0,45	226,271	54,745
5	0,74	214,466	44,238
6	1,22	203,141	37,786
7	2,01	194,470	30,619
8	3,32	187,151	24,834
9	5,47	182,297	20,270
10	9,03	178,048	17,787
11	14,89	174,000	17,144
12	24,55	169,524	17,846
13	40,49	164,266	19,199
14	66,77	158,159	20,942
15	110,12	151,267	22,829
16	181,62	143,613	24,647
17	299,52	135,262	26,352
18	493,98	126,234	27,807
19	814,68	116,552	28,775
20	1343,57	106,411	29,023
21	2215,84	96,094	28,286
22	3654,39	85,819	26,289
23	6026,87	76,100	22,731
24	9939,59	67,373	17,419
25	16392,51	59,753	9,782
26	27034,75	53,656	-1,183
27	44586,08	49,609	-16,944
28	73531,97	48,378	-40,201
29	121269,92	52,112	-75,586
30	200000,00	70,059	-151,721

L.3.2 Hasil Pengujian Akurasi Pengukuran

Tabel 2. Pengukuran impedansi baterai Samsung INR-25R2 Baru

n	1 (mΩ)		2 (mΩ)		3 (mΩ)		4 (mΩ)		5 (mΩ)		6 (mΩ)		7 (mΩ)		8 (mΩ)		9 (mΩ)		10 (mΩ)	
f (Hz)	Zre	Zim	Zre	Zim	Zre	Zim	Zre	Zim	Zre	Zim	Zre	Zim	Zre	Zim	Zre	Zim	Zre	Zim	Zre	Zim
1,00	26,197	1,862	24,360	2,280	24,450	2,299	25,432	2,423	24,209	-1,102	24,564	0,066	22,938	1,470	23,435	1,801	23,374	0,997	26,927	-1,355
1,27	25,455	0,710	26,193	1,651	26,206	0,381	25,014	0,916	25,192	-0,410	26,393	-0,507	24,201	-0,622	26,831	0,365	26,100	-0,037	24,551	2,041
1,60	23,510	0,377	23,945	0,182	25,434	-1,182	24,709	1,499	24,569	-0,494	23,295	0,440	23,870	1,761	22,940	-0,371	26,019	1,365	24,584	-0,235
2,03	24,121	0,380	23,584	0,728	23,748	0,740	24,205	0,796	23,989	-0,147	23,659	0,601	24,265	0,447	24,446	0,855	24,208	0,471	23,275	-0,280
2,57	24,228	0,218	24,956	1,006	24,387	-0,447	23,412	0,768	23,924	1,415	23,537	0,617	24,239	0,703	22,857	0,241	22,708	0,751	22,892	-0,181
3,26	23,785	0,941	23,992	0,395	24,209	-0,489	23,455	0,551	23,137	1,336	24,420	-0,042	23,027	1,428	23,482	0,245	23,569	1,215	24,300	0,067
4,13	23,635	0,469	24,261	0,994	23,583	0,260	23,871	0,583	23,314	0,887	23,819	1,184	23,781	0,333	23,459	0,561	23,791	0,668	23,965	0,424
5,22	23,549	0,818	23,946	0,331	24,899	-0,553	23,489	0,536	23,720	0,678	23,481	1,586	23,874	0,784	24,146	0,827	23,537	0,779	23,412	0,729
6,62	23,445	0,683	23,474	0,787	25,958	-0,172	23,420	0,661	23,693	0,522	23,533	0,580	23,690	0,772	23,561	0,740	23,569	0,464	23,527	0,688
8,38	23,380	0,766	23,592	0,737	23,567	0,599	23,563	0,529	23,624	0,530	23,433	0,550	23,536	0,624	23,648	0,789	23,544	0,810	23,609	0,114
10,61	23,283	0,834	23,504	0,753	23,763	0,750	23,391	0,887	23,317	0,869	23,408	0,813	23,586	0,754	23,487	0,706	23,166	0,972	23,551	0,909
13,43	23,244	0,807	23,327	0,824	22,648	0,856	23,366	0,828	23,422	0,811	23,435	0,798	23,340	0,798	23,357	0,807	23,400	0,828	23,478	0,906
17,01	23,114	0,863	23,240	0,918	23,283	0,935	23,253	0,910	23,231	0,914	23,278	0,905	23,300	0,905	23,266	0,914	23,283	0,901	23,270	0,914
21,54	23,004	1,018	23,104	1,047	23,121	1,033	23,159	1,024	23,155	0,986	23,129	1,016	23,129	1,007	23,164	1,024	23,163	0,998	23,160	1,033
27,28	22,913	1,106	23,056	1,125	22,991	1,122	23,061	1,147	23,042	1,100	23,030	1,143	23,043	1,126	23,000	1,157	23,030	1,143	23,039	1,134
34,55	22,746	1,226	22,888	1,226	22,871	1,236	22,858	1,232	22,854	1,258	22,862	1,227	22,955	1,294	22,876	1,253	22,867	1,240	22,881	1,270
43,76	22,529	1,328	22,671	1,427	22,691	1,379	22,687	1,392	22,695	1,383	22,662	1,406	22,673	1,349	22,682	1,362	22,681	1,358	22,644	1,406
55,41	22,342	1,455	22,423	1,529	22,444	1,515	22,447	1,593	22,507	1,486	22,454	1,558	22,420	1,469	22,469	1,505	22,486	1,479	22,422	1,520
70,17	21,913	1,706	22,164	1,635	22,132	1,607	22,208	1,650	22,186	1,642	22,147	1,632	22,190	1,642	22,195	1,655	22,216	1,641	22,238	1,640
88,86	21,812	1,720	21,862	1,751	21,873	1,781	21,913	1,795	21,896	1,792	21,858	1,808	21,917	1,790	21,884	1,754	21,942	1,771	21,937	1,767

L-29

n	1 (mΩ)		2 (mΩ)		3 (mΩ)		4 (mΩ)		5 (mΩ)		6 (mΩ)		7 (mΩ)		8 (mΩ)		9 (mΩ)		10 (mΩ)	
f (Hz)	Zre	Zim	Zre	Zim																
112,53	21,494	1,821	21,572	1,842	21,626	2,010	21,598	1,859	21,546	1,890	21,603	1,889	21,559	1,868	21,576	1,868	21,573	1,902	21,560	1,907
142,51	21,149	1,873	21,222	1,871	21,266	1,923	21,255	2,005	21,258	1,910	21,228	1,962	21,267	1,940	21,175	1,885	21,210	1,924	21,211	1,967
180,47	20,786	1,935	20,825	1,956	20,907	1,941	20,899	1,946	20,891	1,967	20,847	1,973	20,869	1,959	20,882	1,976	20,830	1,956	20,894	1,946
228,55	20,574	1,955	20,458	2,017	20,414	2,117	20,463	2,012	20,449	2,000	20,473	1,930	20,501	2,003	20,474	1,969	20,479	1,973	20,518	1,985
289,43	19,998	1,934	20,002	1,912	20,012	1,951	20,093	1,927	20,059	1,945	20,115	1,917	20,067	1,928	20,066	1,884	20,043	1,980	20,072	1,936
366,52	19,625	1,779	19,621	1,792	19,688	1,841	19,631	1,809	19,669	1,803	19,722	1,826	19,720	1,774	19,671	1,841	19,673	1,794	19,740	1,830
464,16	19,236	1,650	19,295	1,686	19,218	1,863	19,323	1,723	19,318	1,719	19,237	1,663	19,248	1,701	19,305	1,715	19,272	1,665	19,231	1,702
587,80	18,865	1,412	18,888	1,501	18,913	1,487	18,890	1,462	18,932	1,511	18,935	1,494	18,973	1,418	18,898	1,462	18,916	1,465	18,902	1,514
744,38	18,517	1,156	18,558	1,135	18,618	1,225	18,566	1,221	18,519	1,234	18,560	1,252	18,543	1,215	18,549	1,175	18,581	1,250	18,489	1,284
942,67	18,185	0,910	18,239	0,887	18,190	0,801	18,282	0,847	18,211	0,833	18,251	0,955	18,255	0,959	18,297	0,828	18,306	0,827	18,229	1,031
1193,78	17,924	0,449	17,827	0,441	17,951	0,489	17,957	0,470	18,004	0,490	17,942	0,547	17,882	0,525	17,856	0,463	17,946	0,577	17,974	0,464
1511,78	17,628	0,048	17,652	-0,039	17,798	0,066	17,728	0,017	17,694	0,050	17,671	0,067	17,741	0,015	17,658	0,065	17,772	0,089	17,667	0,068
1914,48	17,515	-0,546	17,449	-0,411	17,435	-0,470	17,461	-0,454	17,575	-0,440	17,452	-0,531	17,480	-0,507	17,411	-0,442	17,503	-0,561	17,411	-0,380
2424,46	17,217	-1,161	17,228	-1,263	17,337	-1,304	17,320	-1,241	17,250	-1,265	17,214	-1,156	17,270	-1,226	17,380	-1,128	17,302	-1,209	17,212	-1,164
3070,29	17,113	-1,972	17,116	-1,964	17,168	-1,900	17,130	-2,002	17,129	-1,955	17,123	-1,971	17,194	-2,003	17,179	-1,919	17,242	-1,887	17,142	-1,932
3888,16	16,995	-3,049	17,064	-2,948	17,184	-2,911	17,078	-2,960	16,956	-2,953	17,031	-2,995	17,035	-2,900	17,030	-2,989	17,147	-2,887	17,060	-2,907
4923,88	17,023	-4,198	17,152	-4,055	17,104	-4,081	17,069	-4,104	17,018	-4,118	17,091	-4,046	17,099	-4,145	17,106	-4,014	17,105	-4,128	17,017	-4,096
6235,51	16,930	-5,461	16,929	-5,501	17,007	-5,590	17,108	-5,523	17,037	-5,505	17,044	-5,477	16,965	-5,552	16,996	-5,492	17,001	-5,473	17,039	-5,496
7896,52	17,123	-7,271	17,125	-7,237	17,173	-7,228	17,172	-7,300	17,151	-7,125	17,160	-7,233	17,195	-7,125	17,213	-7,259	17,175	-7,238	17,156	-7,285
10000,00	17,302	-9,306	17,290	-9,302	17,262	-9,304	17,409	-9,300	17,298	-9,241	17,286	-9,303	17,391	-9,245	17,296	-9,237	17,309	-9,232	17,352	-9,347

Tabel 3. Pengukuran impedansi baterai Samsung INR-25R2 Bekas Baru

n	1 (mΩ)		2 (mΩ)		3 (mΩ)		4 (mΩ)		5 (mΩ)		6 (mΩ)		7 (mΩ)		8 (mΩ)		9 (mΩ)		10 (mΩ)	
f (Hz)	Zre	Zim	Zre	Zim	Zre	Zim	Zre	Zim	Zre	Zim	Zre	Zim	Zre	Zim	Zre	Zim	Zre	Zim	Zre	Zim
1,00	27,463	1,983	28,604	-1,478	28,627	0,038	29,078	2,574	28,945	1,104	29,875	0,574	28,670	0,931	29,191	1,943	28,473	1,059	28,394	2,079
1,27	28,007	1,689	28,149	2,594	27,758	1,266	29,004	0,635	28,605	0,057	30,423	0,594	29,003	0,903	27,969	0,831	27,554	1,680	28,851	1,904
1,60	28,541	1,546	28,853	0,193	28,997	1,480	28,510	0,462	29,801	0,927	28,560	-1,142	28,066	1,425	28,287	0,896	28,143	0,973	29,565	1,703
2,03	28,659	1,124	28,287	1,124	27,994	1,016	28,222	1,869	28,387	2,417	28,688	1,156	28,445	0,708	28,715	1,127	28,586	1,480	28,439	1,205
2,57	28,469	0,745	28,962	1,185	28,433	1,127	27,869	2,542	27,346	0,992	29,130	1,343	28,130	2,182	28,761	1,817	28,470	1,189	28,208	1,227
3,26	28,033	1,523	28,571	1,257	27,918	0,830	28,005	1,621	28,047	1,497	28,353	1,850	27,738	1,601	28,155	1,177	27,945	1,262	28,041	1,230
4,13	27,933	1,861	27,779	1,594	27,978	1,806	27,042	1,874	28,365	2,075	28,144	1,710	28,047	1,803	28,105	1,718	28,040	1,535	27,973	1,992
5,22	27,859	1,546	27,693	1,660	27,733	1,851	27,228	2,028	27,696	1,755	28,016	1,769	27,646	1,906	27,777	1,813	27,770	1,744	27,245	1,536
6,62	27,319	1,930	27,393	1,875	27,362	1,974	27,501	1,837	27,418	1,944	27,387	2,052	27,635	1,856	27,536	2,174	27,535	1,924	27,211	1,890
8,38	27,087	2,006	27,065	2,049	27,078	2,023	27,056	2,105	27,069	2,062	27,155	2,088	27,052	2,058	27,168	2,097	27,256	1,994	27,116	2,144
10,61	26,715	2,170	26,667	2,161	26,697	2,252	26,723	2,252	26,732	2,178	26,672	2,200	26,723	2,204	26,710	2,174	26,728	2,191	26,715	2,204
13,43	26,341	2,259	26,332	2,285	26,332	2,293	26,414	2,271	26,341	2,302	26,376	2,315	26,371	2,289	26,371	2,302	26,380	2,315	26,341	2,332
17,01	25,934	2,365	25,935	2,430	25,908	2,353	25,926	2,396	25,925	2,361	25,978	2,391	25,961	2,421	25,956	2,404	25,956	2,378	25,982	2,391
21,54	25,500	2,409	25,525	2,392	25,504	2,396	25,543	2,383	25,534	2,366	25,534	2,396	25,534	2,422	25,565	2,417	25,539	2,430	25,556	2,409
27,28	25,098	2,434	25,115	2,416	25,090	2,447	25,137	2,420	25,119	2,420	25,120	2,442	25,163	2,454	25,150	2,441	25,145	2,429	25,141	2,424
34,55	24,721	2,409	24,734	2,408	24,691	2,427	24,691	2,409	24,704	2,422	24,717	2,430	24,730	2,434	24,743	2,421	24,726	2,426	24,738	2,425
43,76	24,318	2,343	24,281	2,409	24,253	2,349	24,367	2,384	24,381	2,423	24,281	2,396	24,256	2,423	24,391	2,340	24,280	2,370	24,423	2,395
55,41	23,895	2,314	23,920	2,396	23,910	2,374	23,918	2,348	23,949	2,364	23,923	2,369	23,964	2,407	23,958	2,381	23,932	2,365	23,954	2,385
70,17	23,508	2,282	23,352	2,421	23,532	2,316	23,573	2,348	23,522	2,295	23,532	2,324	23,553	2,306	23,536	2,311	23,581	2,348	23,529	2,333
88,86	23,166	2,210	23,182	2,252	23,158	2,271	23,175	2,270	23,170	2,262	23,203	2,242	23,206	2,277	23,163	2,284	23,178	2,257	23,215	2,233

n	1 (mΩ)		2 (mΩ)		3 (mΩ)		4 (mΩ)		5 (mΩ)		6 (mΩ)		7 (mΩ)		8 (mΩ)		9 (mΩ)		10 (mΩ)	
f (Hz)	Zre	Zim	Zre	Zim																
112,53	22,794	2,157	22,807	2,161	22,803	2,178	22,777	2,191	22,795	2,230	22,820	2,143	22,829	2,212	22,842	2,234	22,816	2,200	22,829	2,191
142,51	22,438	2,081	22,425	2,055	22,452	2,115	22,460	2,076	22,399	2,038	22,438	2,081	22,417	2,124	22,422	2,120	22,494	2,037	22,446	2,050
180,47	22,096	1,962	22,096	1,962	22,083	1,975	22,113	1,957	22,066	1,984	22,114	2,009	22,127	1,995	22,165	1,982	22,161	1,982	22,139	1,965
228,55	21,806	1,943	21,770	1,884	21,752	1,871	21,775	1,905	21,714	1,876	21,774	1,858	21,731	1,889	21,803	1,822	21,770	1,897	21,786	1,853
289,43	21,435	1,762	21,451	1,723	21,469	1,752	21,452	1,757	21,430	1,754	21,504	1,777	21,470	1,791	21,443	1,762	21,500	1,790	21,460	1,753
366,52	21,133	1,582	21,177	1,602	21,119	1,553	21,148	1,625	21,143	1,612	21,139	1,604	21,152	1,625	21,146	1,577	21,148	1,625	21,164	1,598
464,16	20,838	1,469	20,810	1,418	20,820	1,452	20,843	1,477	20,784	1,355	20,865	1,480	20,848	1,481	20,844	1,412	20,833	1,460	20,872	1,445
587,80	20,494	1,200	20,578	1,220	20,559	1,200	20,549	1,253	20,487	1,283	20,566	1,243	20,536	1,245	20,610	1,188	20,588	1,189	20,557	1,239
744,38	20,255	1,017	20,256	0,982	20,317	0,933	20,277	0,967	20,241	0,957	20,298	1,009	20,305	0,999	20,287	0,940	20,316	1,025	20,257	0,986
942,67	19,952	0,738	20,020	0,648	20,002	0,759	20,022	0,622	20,040	0,633	20,053	0,710	20,017	0,618	19,996	0,664	20,059	0,648	20,048	0,697
1193,78	19,759	0,300	19,760	0,278	19,743	0,311	19,803	0,333	19,794	0,300	19,740	0,294	19,826	0,312	19,756	0,252	19,757	0,383	19,702	0,330
1511,78	19,523	-0,156	19,513	-0,067	19,435	-0,180	19,538	-0,120	19,564	-0,173	19,565	-0,072	19,561	-0,093	19,498	-0,152	19,589	-0,133	19,557	-0,114
1914,48	19,335	-0,700	19,348	-0,699	19,381	-0,693	19,279	-0,559	19,296	-0,620	19,346	-0,632	19,355	-0,612	19,277	-0,567	19,368	-0,708	19,358	-0,657
2424,46	19,242	-1,310	19,268	-1,222	19,254	-1,268	19,315	-1,232	19,179	-1,341	19,159	-1,299	19,235	-1,272	19,193	-1,214	19,247	-1,248	19,229	-1,338
3070,29	19,070	-1,963	19,143	-1,990	19,151	-1,947	19,063	-2,006	19,027	-2,029	19,181	-2,000	19,108	-1,972	19,157	-1,954	19,110	-2,066	19,115	-1,905
3888,16	19,051	-2,943	18,956	-2,872	18,921	-2,884	18,937	-2,979	18,951	-2,971	18,945	-2,934	18,898	-2,960	18,929	-2,888	19,006	-2,819	19,088	-2,850
4923,88	18,929	-3,993	18,818	-4,035	18,915	-4,015	18,929	-3,978	18,896	-4,019	18,975	-3,946	18,918	-4,018	18,893	-3,944	18,929	-4,009	18,883	-4,000
6235,51	18,938	-5,337	18,861	-5,449	18,875	-5,374	18,937	-5,371	18,867	-5,279	18,934	-5,380	18,906	-5,379	18,892	-5,343	18,973	-5,286	18,917	-5,295
7896,52	19,019	-7,004	19,011	-7,015	18,944	-7,028	19,044	-7,030	18,985	-7,046	18,938	-7,043	18,947	-7,091	18,964	-6,931	19,000	-6,964	19,015	-7,013
10000,00	19,246	-9,032	19,234	-9,037	19,193	-8,997	19,209	-9,196	19,235	-9,051	19,183	-9,168	19,221	-9,076	19,216	-9,212	19,208	-9,091	19,185	-9,091

Tabel 4. Pengukuran impedansi baterai Samsung INR-25R2 Bekas Tua

n	1 (mΩ)		2 (mΩ)		3 (mΩ)		4 (mΩ)		5 (mΩ)		6 (mΩ)		7 (mΩ)		8 (mΩ)		9 (mΩ)		10 (mΩ)	
f (Hz)	Zre	Zim	Zre	Zim																
1,00	28,003	4,291	29,478	4,367	30,612	1,556	31,273	1,335	29,436	2,547	30,670	4,347	31,383	0,931	30,535	2,405	29,989	1,118	31,400	2,956
1,27	30,933	1,610	30,388	2,345	30,090	1,248	31,811	3,303	28,959	3,990	28,635	3,460	28,872	2,054	29,517	1,333	28,420	1,733	30,305	1,633
1,60	29,688	1,800	30,291	2,276	29,432	1,127	29,960	1,511	30,278	2,481	30,843	1,213	28,711	1,401	29,083	0,380	30,913	1,524	30,359	1,489
2,03	28,167	0,768	30,097	2,211	29,491	2,498	29,582	2,919	30,304	1,820	29,830	2,654	29,846	1,687	29,997	2,116	29,977	2,167	29,997	2,125
2,57	28,539	2,384	30,479	1,044	29,591	2,476	28,343	1,387	29,869	2,147	29,701	2,244	30,374	3,140	30,256	2,285	29,092	2,343	28,977	2,415
3,26	28,837	2,268	28,402	2,522	29,539	2,453	29,607	2,368	28,905	2,457	29,028	1,939	29,105	2,002	29,268	2,515	29,185	2,252	29,987	2,748
4,13	28,511	2,103	28,530	2,575	27,959	2,580	29,074	2,108	28,593	3,049	28,933	2,745	28,823	3,024	28,688	3,038	29,141	2,746	29,007	2,743
5,22	28,458	2,092	28,388	2,835	28,561	2,328	28,641	2,537	29,034	2,260	28,365	2,728	28,543	2,691	28,822	2,535	29,053	2,447	28,507	2,799
6,62	27,706	3,041	27,613	2,901	27,613	2,989	27,846	3,057	27,663	2,496	28,118	3,104	27,993	3,037	27,868	3,040	28,194	3,274	27,927	3,478
8,38	27,213	2,874	26,988	2,895	27,369	3,033	27,252	3,015	27,261	2,993	27,433	3,146	27,668	2,892	27,450	3,107	27,398	3,167	27,381	3,093
10,61	26,569	3,092	26,574	2,988	26,753	3,049	26,714	3,105	26,800	3,213	26,834	3,217	26,830	3,226	26,839	3,179	26,917	3,166	26,791	3,252
13,43	26,049	3,004	25,993	2,991	26,128	3,091	26,171	3,121	26,223	3,130	26,318	3,324	26,283	3,151	26,283	3,173	26,206	3,177	26,534	3,103
17,01	25,577	3,013	25,499	2,992	25,642	3,032	25,664	3,084	25,733	3,101	25,720	3,144	25,755	3,105	25,742	3,122	25,764	3,135	25,759	3,118
21,54	24,973	2,905	24,973	2,901	25,199	3,004	25,182	2,970	25,174	3,013	25,200	3,039	25,204	3,060	25,243	3,047	25,265	3,055	25,339	3,084
27,28	24,614	2,873	24,553	2,843	24,664	2,933	24,694	2,941	24,768	2,944	24,777	2,953	24,751	2,945	24,791	3,000	24,764	2,992	24,799	2,996
34,55	24,149	2,803	24,110	2,795	24,253	2,851	24,240	2,873	24,318	2,888	24,301	2,889	24,328	2,910	24,314	2,888	24,311	2,923	24,341	2,922
43,76	23,712	2,733	23,697	2,673	23,805	2,814	23,791	2,802	23,876	2,769	23,865	2,808	23,870	2,821	23,879	2,842	23,860	2,795	23,944	2,849
55,41	23,352	2,712	23,294	2,676	23,386	2,746	23,302	2,698	23,434	2,666	23,444	2,700	23,416	2,731	23,411	2,736	23,459	2,725	23,478	2,763
70,17	22,887	2,587	22,872	2,545	22,967	2,637	22,921	2,592	23,002	2,657	22,990	2,671	23,006	2,635	23,020	2,660	23,041	2,655	23,034	2,681
88,86	22,480	2,482	22,468	2,483	22,556	2,490	22,555	2,542	22,581	2,592	22,609	2,564	22,587	2,566	22,590	2,596	22,613	2,616	22,648	2,622

L₃₃

n	1 (mΩ)		2 (mΩ)		3 (mΩ)		4 (mΩ)		5 (mΩ)		6 (mΩ)		7 (mΩ)		8 (mΩ)		9 (mΩ)		10 (mΩ)	
f (Hz)	Zre	Zim	Zre	Zim																
112,53	22,283	2,261	22,056	2,380	22,190	2,417	22,148	2,470	22,165	2,465	22,221	2,482	22,195	2,469	22,212	2,430	22,200	2,482	22,234	2,482
142,51	21,680	2,241	21,645	2,211	21,795	2,358	21,717	2,342	21,790	2,315	21,773	2,358	21,825	2,366	21,839	2,401	21,778	2,384	21,843	2,370
180,47	21,353	2,237	21,374	2,177	21,408	2,273	21,441	2,186	21,438	2,221	21,447	2,255	21,451	2,264	21,478	2,281	21,486	2,276	21,516	2,254
228,55	20,983	2,074	20,971	2,113	21,031	2,075	21,089	2,143	21,092	2,104	21,093	2,142	21,119	2,146	21,097	2,142	21,145	2,154	21,077	2,199
289,43	20,663	1,900	20,584	1,997	20,672	1,997	20,620	1,977	20,705	1,961	20,716	2,017	20,703	2,009	20,655	1,989	20,691	2,061	20,690	2,009
366,52	20,319	1,795	20,285	1,801	20,349	1,857	20,344	1,853	20,372	1,800	20,393	1,890	20,346	1,901	20,385	1,895	20,331	1,862	20,327	1,854
464,16	19,979	1,648	20,040	1,658	20,016	1,638	20,021	1,655	20,036	1,606	19,999	1,652	20,044	1,679	20,042	1,723	20,114	1,624	20,133	1,718
587,80	19,652	1,292	19,617	1,420	19,674	1,425	19,687	1,363	19,655	1,457	19,639	1,414	19,747	1,476	19,644	1,423	19,747	1,355	19,713	1,414
744,38	19,351	1,107	19,361	1,072	19,352	1,168	19,482	1,165	19,346	1,143	19,367	1,180	19,375	1,227	19,365	1,167	19,381	1,244	19,371	1,236
942,67	19,011	0,794	19,025	0,797	19,021	0,872	19,088	0,890	19,018	0,763	19,074	0,840	19,028	0,893	19,085	0,856	19,134	0,833	19,109	0,844
1193,78	18,810	0,466	18,783	0,492	18,840	0,496	18,848	0,491	18,778	0,457	18,804	0,488	18,852	0,547	18,872	0,505	18,922	0,520	18,850	0,534
1511,78	18,581	0,017	18,488	-0,036	18,506	0,088	18,557	0,105	18,657	0,101	18,581	-0,009	18,568	0,068	18,593	0,059	18,598	0,089	18,565	0,148
1914,48	18,318	-0,514	18,373	-0,469	18,324	-0,482	18,324	-0,500	18,356	-0,596	18,228	-0,513	18,375	-0,494	18,446	-0,532	18,330	-0,475	18,345	-0,602
2424,46	18,186	-1,140	18,087	-1,170	18,202	-1,181	18,242	-1,153	18,238	-1,151	18,159	-1,142	18,214	-1,154	18,160	-1,214	18,163	-1,084	18,267	-1,160
3070,29	18,015	-1,819	18,011	-1,841	18,123	-1,795	18,109	-1,868	18,108	-1,859	18,036	-1,841	18,068	-1,987	17,982	-1,899	18,006	-1,834	17,978	-1,921
3888,16	17,927	-2,831	17,991	-2,848	17,914	-2,818	17,929	-2,787	18,024	-2,907	17,927	-2,830	17,924	-2,896	17,919	-2,797	17,902	-2,754	17,887	-2,892
4923,88	17,777	-3,962	17,812	-3,964	17,928	-4,034	17,856	-3,898	17,854	-3,892	17,915	-3,973	17,886	-3,923	17,866	-3,978	17,837	-3,927	17,783	-3,939
6235,51	17,716	-5,356	17,837	-5,306	17,877	-5,293	17,841	-5,261	17,906	-5,372	17,901	-5,345	17,830	-5,293	17,879	-5,330	17,893	-5,314	17,838	-5,329
7896,52	17,791	-7,025	17,860	-6,898	17,847	-6,997	17,818	-7,046	17,806	-7,000	17,903	-6,977	17,884	-7,083	17,886	-6,896	17,873	-6,900	17,855	-7,001
10000,00	17,914	-9,086	17,904	-9,176	18,007	-9,096	18,070	-9,132	18,008	-9,110	18,055	-9,143	18,038	-9,136	18,098	-9,121	18,011	-9,095	18,040	-8,925

Tabel 5. Rerata pengukuran setiap sampel baterai

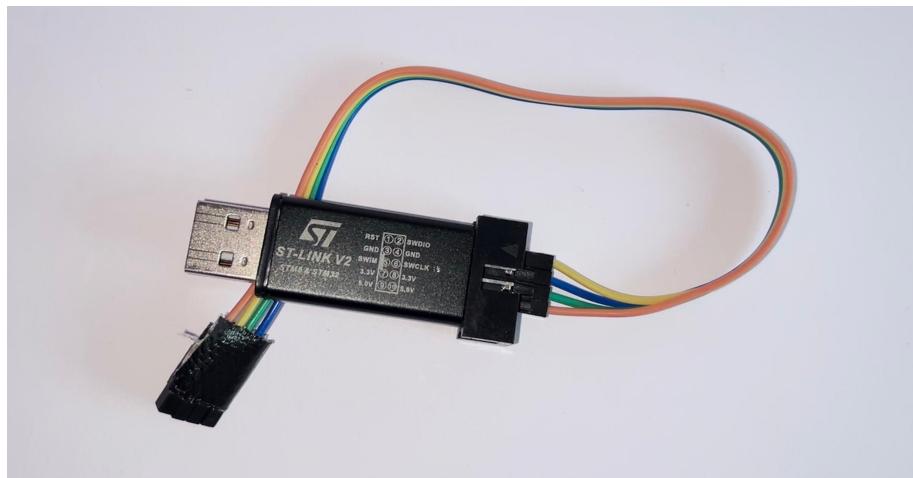
f (Hz)	25R2 Baru (mΩ)		25R2 Bekas Baru (mΩ)		25R2 Bekas Tua (mΩ)	
	Zre	Zim	Zre	Zim	Zre	Zim
1,00	24,589	1,074	28,732	1,081	30,278	2,585
1,27	25,614	0,449	28,532	1,215	29,793	2,271
1,60	24,287	0,334	28,732	0,846	29,956	1,520
2,03	23,950	0,459	28,442	1,323	29,729	2,096
2,57	23,714	0,509	28,378	1,435	29,522	2,186
3,26	23,738	0,565	28,081	1,385	29,186	2,352
4,13	23,748	0,636	27,941	1,797	28,726	2,671
5,22	23,805	0,652	27,666	1,761	28,637	2,525
6,62	23,787	0,573	27,430	1,946	27,854	3,042
8,38	23,550	0,605	27,110	2,063	27,341	3,021
10,61	23,446	0,825	26,708	2,199	26,762	3,149
13,43	23,302	0,826	26,360	2,296	26,219	3,127
17,01	23,252	0,908	25,946	2,389	25,685	3,085
21,54	23,129	1,019	25,533	2,402	25,175	3,008
27,28	23,020	1,130	25,128	2,433	24,717	2,942
34,55	22,866	1,246	24,719	2,421	24,266	2,874
43,76	22,661	1,379	24,323	2,383	23,830	2,791
55,41	22,442	1,511	23,932	2,370	23,398	2,715
70,17	22,159	1,645	23,522	2,328	22,974	2,632
88,86	21,889	1,773	23,182	2,256	22,569	2,555
112,53	21,571	1,886	22,811	2,190	22,190	2,434
142,51	21,224	1,926	22,439	2,078	21,769	2,335
180,47	20,863	1,955	22,116	1,977	21,439	2,242
228,55	20,480	1,996	21,768	1,880	21,070	2,129
289,43	20,053	1,931	21,461	1,762	20,670	1,992
366,52	19,676	1,809	21,147	1,600	20,345	1,851
464,16	19,268	1,709	20,836	1,445	20,042	1,660

	25R2 Baru (mΩ)		25R2 Bekas Baru (mΩ)		25R2 Bekas Tua (mΩ)	
	f (Hz)	Zre	Zim	Zre	Zim	Zre
587,80	18,911	1,473	20,552	1,226	19,678	1,404
744,38	18,550	1,215	20,281	0,982	19,375	1,171
942,67	18,244	0,888	20,021	0,674	19,059	0,838
1193,78	17,926	0,491	19,764	0,309	18,836	0,500
1511,78	17,701	0,045	19,534	-0,126	18,569	0,063
1914,48	17,469	-0,474	19,334	-0,645	18,342	-0,518
2424,46	17,273	-1,212	19,232	-1,274	18,192	-1,155
3070,29	17,154	-1,951	19,113	-1,983	18,043	-1,866
3888,16	17,058	-2,950	18,968	-2,910	17,934	-2,836
4923,88	17,078	-4,098	18,909	-3,996	17,851	-3,949
6235,51	17,006	-5,507	18,910	-5,349	17,852	-5,320
7896,52	17,164	-7,230	18,987	-7,016	17,852	-6,982
10000,00	17,320	-9,282	19,213	-9,095	18,014	-9,102

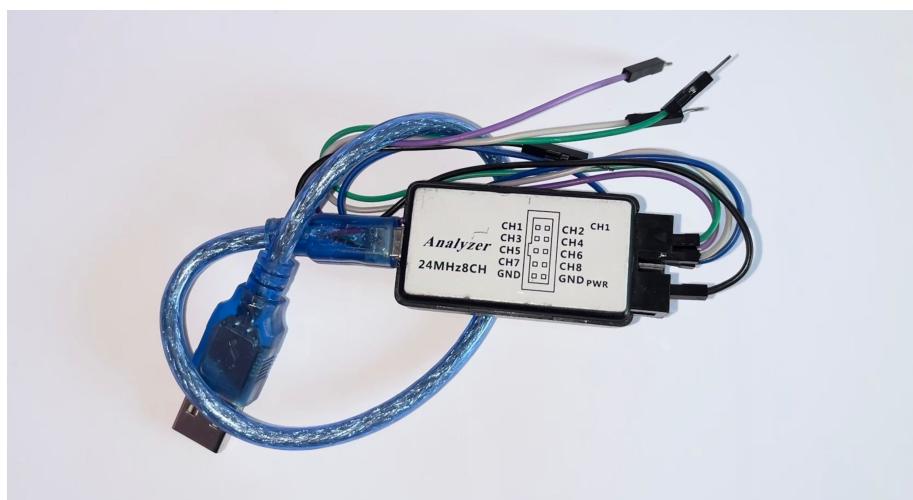
L.4 Alat dan Bahan Penelitian



Gambar 1. Baterai Samsung INR18650-25R2 objek pengukuran akurasi



Gambar 2. Perangkat ST-Link/V2 Programming



Gambar 3. Perangkat Logic Analyzer 24MHz



Gambar 4. Kabel dan *holder* baterai pengukuran impedansi