

**PEMODELAN DEGRADASI SISTEM BATERAI PADA
KENDARAAN LISTRIK RODA DUA BERBASIS
PEMBELAJARAN MESIN**

TUGAS AKHIR



Oleh

Ammar Akila Azhar : 13316048

Faishal Rafi Elian : 13316078

**PROGRAM STUDI TEKNIK FISIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI BANDUNG
2020**

ABSTRAK

PEMODELAN DEGRADASI SISTEM BATERAI PADA KENDARAAN LISTRIK RODA DUA BERBASIS PEMBELAJARAN MESIN

Oleh

Ammar Akila Azhar: 13316048

Faishal Rafi Elian : 13316078

(Program Studi Teknik Fisika)

Saat ini dunia sudah mulai mengalami transformasi besar dalam bidang transportasi dengan meningkatnya jumlah kendaraan listrik yang terjual. Baterai sebagai komponen penyimpanan energi memainkan peran vital dalam hal keamanan hingga efisiensi kendaraan listrik. Namun, baterai tidak dapat digunakan secara terus menerus sehingga pemodelan degradasi baterai menjadi hal penting untuk diketahui. Untuk itu, Sistem Manajemen Baterai (SMB) yang baik harus dikembangkan agar degradasi baterai dapat dimodelkan.

Penelitian ini akan membahas integrasi Sistem Manajemen Baterai yang dapat memberikan cukup data agar Kondisi Kesehatan (KK) dari modul baterai pada kendaraan listrik dapat dimodelkan degradasinya. Pada penelitian ini dibuat SMB berbasis *Internet of Things* (IoT) pada kendaraan listrik roda dua dengan baterai pak Li-Ion jenis katoda LiNiCoMnO_2 (*Nickel-Manganese-Cobalt*, NMC) yang memiliki 5 *level assist* sebagai pembatas daya motor dengan *level 5* sebagai *level* tertinggi, menggunakan ESP32 sebagai modul komunikasi dan mikrokontroler. Pengambilan data dilakukan dengan 3 skenario secara *real-time* dengan skenario 1 dan 2 dilakukan di dalam kampus ITB dengan *level assist* masing-masing 5 dan 3, dan skenario 3 dilakukan di rute menanjak ITB—Jl. Ir. H. Djuanda menuju Bukit Dago dengan *level assist 5*. Data ini akan dilatih bersama dengan data baterai Li-Ion jenis katoda LiFePO_4 (*Lithium-Ferro-Phosphat*, LFP) dan LiMn_2O_4 (*Lithium-Manganese Oxide*, LMO) yang berasal dari referensi dengan beberapa model pembelajaran mesin dan hasil permodelan terbaik yang telah dioptimasi akan dibandingkan dengan model degradasi baterai NMC.

Pada Penelitian ini, SMB yang telah dibuat berhasil menghasilkan data tegangan dan arus selama berkendara yang selanjutnya diolah untuk menghasilkan data kapasitas beberapa siklus awal dengan metode *Approximate Weighted Total Least Square* (AWTLS). Data ini berhasil dimodelkan dengan pembelajaran mesin Regresi Vektor Pendukung (RVP) menggunakan tambahan data baterai jenis LFP untuk skenario 1 dan 2, dan tambahan data LMO untuk skenario 3. Model Skenario 1 dan 2 tidak berhasil mencapai kapasitas degradasi sedangkan skenario 3 berhasil mencapai kapasitas degradasi pada siklus ke-490. Sehingga model skenario 3 dapat merepresentasikan model degradasi baterai NMC dengan $R^2=0,911$.

Kata kunci: Kendaraan listrik, Sistem Manajemen Baterai, Degradasi Baterai, Pembelajaran Mesin, *Internet of Things*

ABSTRACT

BATTERY SYSTEM DEGRADATION MODELLING FOR TWO-WHEEL ELECTRIC VEHICLE BASED ON MACHINE LEARNING

By

Ammar Akila Azhar : 13316048

Faishal Rafi Elian : 13316078

(Engineering Physics Study Program)

Today the world has begun to experience a major transformation in transportation, with the increasing number of electric vehicles sold. Batteries as energy storage, play a vital role in terms of electric vehicle efficiency. But, the battery cannot be used continuously so its degradation model becomes important to be known.. In order to model battery degradation, a suitable Battery Management System (BMS) needs to be developed.

This study will discuss the integration of a Battery Management System that can provide sufficient data so that the battery State-of-Health (SOH) degradation of the battery pack in electric vehicles can be modeled. In this study, an Internet of Things (IoT) based BMS will be made on a two-wheeled electric vehicle with a Li-Ion battery pack of the LiNiCoMnO₂ (Nickel-Manganese-Cobalt,NMC) cathode. This electric vehicle has 5 levels of assist as the motor power limit with level 5 as the highest level, using ESP32 as a communication module. and microcontroller. Data was collected using 3 scenarios in real-time, with scenarios 1 and 2 carried out on the ITB campus with assist levels 5 and 3 respectively, and scenario 3 carried out on the uphill route ITB-Jl. Ir. H. Djuanda to Bukit Dago with assist level 5, this data will be trained together with Li-Ion battery data of the LiFePO₄ (Lithium-Ferro-Phosphat,LFP) and LiMn₂O₄ (Lithium-Manganese Oxide,LMO) cathode types which come from references with several machine learning models and the best-optimized modeling results will be compared to the NMC battery degradation model.

In this study, the BMS that has been made has succeeded in producing voltage and current data during driving which is then processed to produce capacity data for several initial cycles using the Approximate Weighted Total Least Square (AWTLS) method. This data is successfully modeled by machine learning Supporting Vector Regression (SVR) using additional LFP type battery data for scenarios 1 and 2, and additional LMO battery type data for scenario 3. Scenario 1 and 2 models fail to achieve degradation capacity while scenario 3 succeeds in achieving degradation capacity on the 490th cycle. So the scenario 3 model can represent the NMC battery degradation model with $R^2=0.911$.

Keywords: *Electric Vehicle, Battery Management System, Battery Degradation, Machine Learning, Internet of Things*

**PEMODELAN DEGRADASI SISTEM BATERAI PADA
KENDARAAN LISTRIK RODA DUA BERBASIS
PEMBELAJARAN MESIN**

HALAMAN PENGESAHAN

Oleh

Ammar Akila Azhar : 13316048

Faishal Rafi Elian : 13316078

(Program Studi Teknik Fisika)

Institut Teknologi Bandung

Menyetujui

Tim Pembimbing

21 September 2020

Pembimbing 1



Ir. Edi Leksono M.Eng., Ph.D.
NIP. 195912281984031002

Pembimbing 2



Dr. Irsyad Nashirul Haq S.T., M.T.
NIP. 119110107

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir ini. Tugas akhir yang berjudul “Pemodelan Degradasi Sistem Baterai pada Kendaraan Listrik Roda Dua dengan Pembelajaran Mesin” telah selesai disusun untuk memenuhi syarat menyelesaikan studi program sarjana di Program Studi Teknik Fisika, Fakultas Teknologi Industri, Institut Teknologi Bandung.

Tugas Akhir ini berisi tentang proses pembuatan model akuisisi data untuk pak baterai dari kendaraan listrik, proses pengambilan data secara *real-time* di jalanan kampus ITB dan sekitarnya, serta proses pemodelan degradasi sistem baterai menggunakan model pembelajaran mesin. Dalam pengerjaan tugas akhir ini, penulis banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu, izinkan penulis menyampaikan rasa terima kasih dan penghargaan setinggi-tingginya kepada :

1. Orang tua kedua penulis yang selalu memberikan dukungan moral dan material.
2. Dr. Ir. Edi Leksono, M.Eng. dan Dr. Irsyad Nashirul Haq, S.T., M.T. selaku dosen pembimbing yang telah membantu dengan bimbingan, saran dan masukan untuk pengerjaan tugas akhir ini.
3. Ir. F.X. Nugroho Soelami M.BEnv.,Ph.D. dan Dr. Vebi Nadhira, S.T., M.T.selaku dosen wali atas bimbingan dan nasihat selama penulis menempuh studi di Prodi Teknik Fisika - ITB.
4. Staf dan pegawai Prodi Teknik Fisika - ITB atas segala bantuan dan kemudahan yang diberikan selama penulis menempuh studi di Prodi Teknik Fisika - ITB.
5. Bapak Frans Edison, S.T., M.T., Bapak Kristiawan Ariwibawa, S.T dan segenap anggota Laboratorium Manajemen Energi Teknik Fisika - ITB atas ilmu dan bantuan yang telah diberikan untuk menyelesaikan penelitian ini.
6. Diana Vitonia dan Meilisa Dewi Kharisma yang telah membantu dalam pembuatan modul akuisisi data serta pengambilan data menggunakan sepeda.

7. Dimas Apeco Putra dan Prasetyo Wibowo Laksono Sanjaya, yang telah membantu penulis dalam hal pembuatan sistem komunikasi data antara modul akuisisi dengan *database* Lab ME.
8. Teman-teman Teknik Fisika 2016 yang selalu menemani dan memberikan semangat selama penulis menempuh studi di Prodi Teknik Fisika - ITB.
9. Pihak-pihak yang tidak bisa disebutkan satu per satu yang telah memberikan dukungan bagi penulis.

Penulis sangat mengharapkan kritik dan saran yang membangun terkait pengerjaan tugas akhir ini sehingga tugas akhir ini dapat bermanfaat bagi semua pihak. Penulis berharap Tugas Akhir ini dapat bermanfaat bagi mereka yang mungkin hanya sekedar memiliki rasa keingintahuan tentang keilmuan tentang sistem manajemen baterai, maupun mereka yang berencana untuk mengembangkan lebih jauh tentang apa yang tertulis di sini.

Bandung, 1 September 2020

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI	vi
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	xi
DAFTAR SINGKATAN DAN LAMBANG	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Permasalahan.....	2
1.3 Tujuan dan Sasaran	3
1.4 Pembatasan Masalah	3
1.5 Manfaat Penelitian	4
1.6 Metodologi Penelitian	4
BAB II TINJAUAN PUSTAKA.....	6
2.1 Baterai	6
2.1.1 Terminologi Umum Baterai	6
2.1.2 Komponen Sel Elektrokimia	6
2.1.3 Mekanisme Kerja Baterai	7
2.1.4 Mekanisme Degradasi Baterai.....	8
2.2 Sistem Manajemen Baterai	9
2.2.1 <i>Battery Pack Sensing</i>	9
2.2.2 Pengelolaan Performa.....	10
2.2.3 Diagnostik	11
2.3 Sepeda Listrik.....	12
2.4 Pembelajaran Mesin	12
2.4.1 Mesin Vektor Pendukung (MVP) / <i>Support Vector Machine</i> (SVM).....	13
2.4.2 <i>Long-Short Term Memory (Recurrent Neural Network)</i>	15
2.4.3 <i>Decision Tree</i> (DT)	16
2.4.4 <i>Random Forest</i> (RF).....	16
2.5 <i>Approximate Weighted Total Least Square</i>	17

BAB III METODOLOGI PENELITIAN	21
3.1 Permasalahan.....	21
3.2 Spesifikasi Kendaraan Listrik Roda Dua	21
3.2.1 Sepeda Listrik.....	21
3.2.2 Sistem Baterai.....	22
3.2.3 Komponen Pemantau Data.....	24
3.2.4 Komponen Komunikasi.....	25
3.3 Komponen Perangkat Lunak Pendukung.....	26
3.3.1 Arduino IDE.....	26
3.3.2 PuTTY	26
3.3.3 Strava.....	26
3.4 Rancangan Sistem/Ide Solusi	27
3.5 Skenario Data Eksperimen	31
3.6 Data Latih Tambahan	31
3.7 <i>Time Series Forecasting</i>	34
3.7.1 Metode Pelatihan Model Pembelajaran Mesin.....	37
3.7.2 GridSearchCV	38
BAB IV ANALISIS DAN HASIL	39
4.1 Data Eksperimen	39
4.1.1 Rute ITB Kampus Ganesha	39
4.1.2 Rute ITB-Bukit Dago	41
4.2 Hasil Estimasi Kapasitas untuk Beberapa Siklus Awal	43
4.3 <i>Dickey-Fuller Test</i> sebagai <i>Feature Selection</i>	45
4.4 Analisis Model Pembelajaran Mesin	46
4.4.1 <i>Random Forest Regressor, Decision Tree & XGBoost</i>	46
4.4.2 <i>Recurrent Neural Network</i>	49
4.5 Analisis Aplikasi berbagai Kernel Regresi Vektor Pendukung	51
4.6 Optimasi Variasi Terpilih.....	56
BAB V KESIMPULAN	58
5.1 Kesimpulan	58
5.2 Saran.....	58
DAFTAR PUSTAKA	59

LAMPIRAN A KODE SUMBER.....	62
A.1 Kode Sumber AWTLs	62
A.2 Kode Sumber Proses Prediksi, <i>tuning</i> dan <i>ploting</i>	63

DAFTAR GAMBAR

Gambar 1.1 Data Penjualan Mobil Listrik di Beberapa Negara Tahun 2013-2018.1	
Gambar 1.2 Diagram <i>fishbone</i> untuk Kasus Sepeda Listrik <i>e-cleave</i>	3
Gambar 2.1 Grafit, material yang umum digunakan sebagai elektroda negatif (kiri) dan Lithium Manganese Oxide (LMO), material penyusun elektroda positif (kanan).....	8
Gambar 2.2 Skema umum SMB	9
Gambar 2.3 Skema <i>Shunt-Current</i> sensor.....	10
Gambar 2.4 Skema estimasi energi dan daya.....	11
Gambar 2.5 RVP Linear.....	14
Gambar 2.6 Unit Sel LSTM.....	15
Gambar 2.7 Proses Belajar <i>Decision Tree</i>	16
Gambar 2.8 Proses Belajar <i>Random Forest</i>	17
Gambar 2.9 Perbedaan pada WTLS (Kiri), TLS (Tengah), dan AWTLS (Kanan) dalam Mengalkulasi Error Pengukuran	19
Gambar 3.1 Sepeda Gunung Thrill	22
Gambar 3.2 Sel baterai LG-HG2 [battery-bro,2020]	23
Gambar 3.3 Susunan pak baterai sepeda listrik	23
Gambar 3.4 Penempatan pak baterai yang sudah diberi wadah pada sepeda listrik	24
Gambar 3.5 Papan ADS11x dan <i>pin heading</i>	24
Gambar 3.6 Mikrokontroller DOIT ESP32 DevKit V1	25
Gambar 3.7 Antarmuka awal Putty yang terhubung dengan server Lab ME	26
Gambar 3.8 Skema Sistem Pengambilan Data.....	28
Gambar 3.9 Tampilan <i>Database</i> Server Lab ME.....	29
Gambar 3.10 Diagram Alir Algoritma Pengambilan Data Uji	30
Gambar 3.11 Ide Solusi Pemodelan Degradasi Baterai	30
Gambar 3.12 Data Kapasitas dan Regresi Degradasi pada Baterai LFP [18].	31
Gambar 3.13 Kurva Degradasi Baterai Li-Ion NMC [20]	33
Gambar 3.14 Contoh Grafik Degradasi Baterai Li-ion [21]	35
Gambar 3.15 Metode ACF dan PACF dalam Menentukan Orde <i>Lag</i>	37
Gambar 3.16 Pemisahan Data Latih dan Data Uji pada Algoritma <i>Machine Learning</i>	38
Gambar 4.1 Peta rute dalam kampus ITB Ganesha	39
Gambar 4.2 Tegangan Baterai terhadap waktu rute dalam kampus ITB (Assist 5)	40
Gambar 4.3 Arus Baterai terhadap waktu untuk rute dalam kampus ITB (assist 5)	40
Gambar 4.4 Tegangan Baterai terhadap waktu rute dalam kampus ITB (Assist 3)	41
Gambar 4.5 Arus Baterai terhadap waktu rute dalam kampus ITB (Assist 3).....	41
Gambar 4.6 Rute ITB-Bukit Dago	42
Gambar 4.7 Tegangan Baterai terhadap waktu rute ITB-Bukit Dago	42

Gambar 4.8 Arus Baterai terhadap waktu Rute ITB-Bukit Dago	43
Gambar 4.9 Hasil Simulasi Kapasitas Baterai dengan AWTLs pada 50 Siklus Pertama	44
Gambar 4.10 Distribusi Arus Selama Berkendara untuk Tiap Skenario	45
Gambar 4.11 Hasil Prediksi Kapasitas Baterai dengan Model Pembelajaran Mesin <i>Random Forest Regressor</i>	47
Gambar 4.12 Visualisasi <i>Decision Tree</i> pada Prediksi Kapasitas Degradasi	48
Gambar 4.13 Hasil Prediksi Kapasitas Baterai dengan Model Pembelajaran Mesin <i>XGBoost Regressor</i>	49
Gambar 4.14 Hasil Prediksi Kapasitas Baterai dengan Model Pembelajaran Mesin LSTM	50
Gambar 4.15 Fungsi <i>Loss</i> LSTM	51
Gambar 4.16 Hasil Prediksi dengan Model RVP menggunakan kernel (a) linear (b) RBF (Gaussian) dan (c) Polinomial	52
Gambar 4.17 Hasil Prediksi dengan Model RVP menggunakan kernel RBF	53
Gambar 4.18 Beberapa Hasil Prediksi dengan Nilai Indikator Performa yang Optimal	54
Gambar 4.19 Perbandingan Model Optimum dengan kurva degradasi baterai NMC	57

DAFTAR TABEL

Tabel 3.1 Spesifikasi Baterai LG-HG2	22
Tabel 3.2 Variasi Tiap Skenario Pengujian.....	31
Tabel 3.3 Data Kapasitas Baterai LMO pada <i>Discharge Rate</i> Berbeda.[19].....	32
Tabel 3.4 Konstanta-konstanta Hasil Regresi Baterai LMO.....	32
Tabel 3.5 Parameter Model Degradasi Baterai NMC	34
Tabel 4.1 Perbandingan Hasil Model Regresi yang Berbeda pada <i>ADF Test</i>	45
Tabel 4.2 Hasil Indikator Performa dari Tiga Skenario Berkendara dengan Variasi pada Pemilihan Data Referensi Baterai, Kernel RVP, dan Jumlah Siklus	55
Tabel 4.3 Perbandingan R^2 estimasi RVP Sebelum dan Sesudah Tuning	56
Tabel 4.4 Variasi Terpilih Untuk Tiap Skenario Beserta Indikator Performansinya	56

DAFTAR SINGKATAN DAN LAMBANG

SINGKATAN	Nama	Pemakaian pertama kali pada halaman
ACF	<i>Auto Correlation Function</i>	36
ADF	<i>Augmented Dickey-Fuller</i>	35
AWTLS	<i>Approximate Weighted Total Least Square</i>	17
DC	<i>Direct Current</i>	12
DT	<i>Decision Tree</i>	15
GPS	<i>Global Positioning System</i>	26
KK	Kondisi Kesehatan	9
KM	Keadaan Muatan	8
KP	Kedalaman Pengosongan	8
LFP	<i>Lithium Ferro Phosphate</i>	29
Li-ion	Lithium-Ion	2
LMO	<i>Lithium-Manganese-Oxide</i>	8
LSTM	<i>Long-Short Term Memory</i>	14
MAE	<i>Mean Absolute Error</i>	32
MQTT	<i>Message Queuing Telemetry Transport</i>	26
MVP	Mesin Vektor Pendukung	13
NiCd	<i>Nickel-Cadmium</i>	2
Ni-Mh	<i>Nickel-Metal Hydride</i>	2
OLS	<i>Ordinary Least Square</i>	18
PACF	<i>Partial Auto Correlation Function</i>	36
PGA	<i>Programmable Gain Amplifier</i>	25
RBF	<i>Radial Basis Function</i>	51
RMSE	<i>Root Mean Square Error</i>	32
RNN	<i>Recurrent Neural Network</i>	14
RVP	Regresi Vektor Pendukung	13
SEI	<i>Solid Electrolyte Interphase</i>	8
SMB	Sistem Manajemen Baterai	9
SSH	<i>Secure Shell</i>	26
SVM	<i>Support Vector Machine</i>	13
TLS	<i>Total Least Square</i>	18
WLS	<i>Weighted Least Square</i>	18
WTLS	<i>Weighted Total Least Square</i>	18

LAMBANG

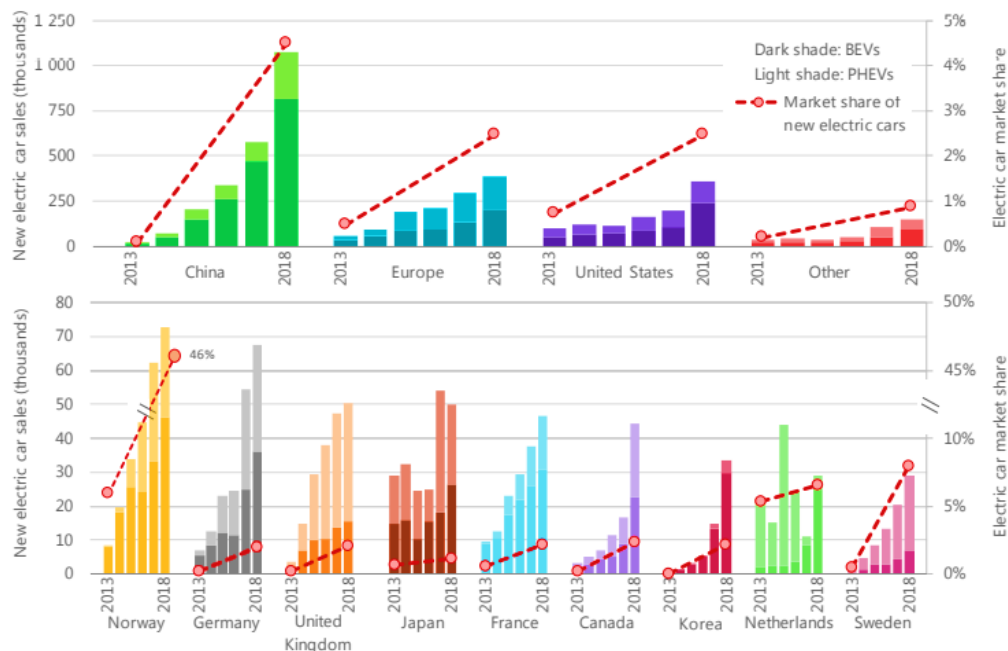
$E(t)$	Energi	11
$E(s)$	Entropi (<i>Decision Tree</i>)	16
k	Nomor siklus	30
OCV	<i>Open Current Voltage</i>	11
$P_{1,2,3,\dots,n}$	Konstanta Polinomial Regresi Baterai LMO	31
P_i	Probabilitas (<i>Decision Tree</i>)	16
Q	Muatan	11
\hat{Q}	Estimasi Muatan	18
V_{nom}	Tegangan Nominal	11
X	Absis hasil regresi	18
x	absis data poin	18
Y	Ordinat hasil regresi	18
y	ordinat data poin	18
α	konstanta regresi	33
α, α^*	Pengali Lagrange (MVP)	14
β	kemiringan garis regresi	36
γ	<i>unit root</i>	36
δ	pembobotan pada komponen lag	36
ε	Kesalahan Maksimum (RVP)	13
λ	Pengali Lagrange (AWTLS)	18
ξ	<i>Slack Variable</i> (RVP)	13
σ	Variansi	18
χ	<i>Cost Function</i>	18
z	Keadaan Muatan	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dua tahun belakangan ini, penjualan kendaraan listrik di dunia mengalami kenaikan dibandingkan dengan tahun-tahun sebelumnya. Berdasarkan data dari *International Energy Agency (IEA)* pada laporan berjudul *Global EV Outlook 2019: Scaling-up the transition to electric mobility*, pangsa pasar kendaraan listrik (pada hal ini mobil listrik) mengalami kenaikan di seluruh dunia.



Gambar 1.1 Data Penjualan Mobil Listrik di Beberapa Negara Tahun 2013-2018

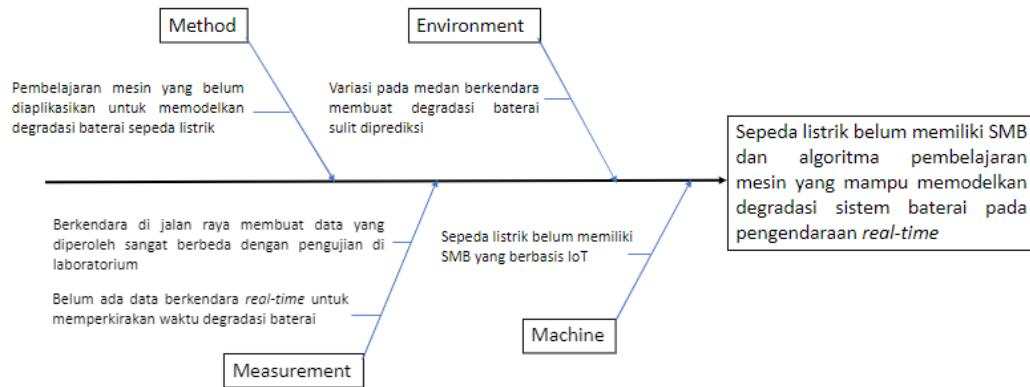
Perkembangan kendaraan listrik di Indonesia pun selaras tren yang terjadi di dunia saat ini. Pada PP No. 22 Tahun 2017 tentang Rancangan Umum Energi Nasional, Indonesia menargetkan untuk mengembangkan kendaraan listrik roda empat sebanyak 2200 unit dan kendaraan listrik roda dua sebanyak 2,1 juta unit pada tahun 2025. Berdasarkan data ini, Indonesia serius ingin mengurangi dominasi kendaraan dengan motor bakar terutama kendaraan roda dua. Kendaraan listrik memiliki komponen yang lebih sederhana dibandingkan dengan kendaraan motor bakar, sehingga Indonesia dapat bersaing sebagai produsen kendaraan listrik terutama di pasar dalam negeri. Komponen yang sudah mulai diproduksi di

Indonesia adalah baterai. Indonesia dinilai memiliki potensi sebagai pusat pengembangan baterai di Asia. Hal ini dibuktikan oleh besarnya laju investasi dan pembangunan riset dan pengembangan, dan pabrik baterai di Indonesia oleh beberapa perusahaan kendaraan bermotor dari negara Asia terutama Tiongkok dan Korea Selatan.

Baterai memang memegang andil yang sangat penting dalam sistem energi kendaraan listrik, yaitu berupa perangkat penyimpan energi. Peran ini menjadikan baterai salah satu bagian dari kendaraan listrik yang harus digunakan secara baik dan benar[1]. Baterai menyimpan energi listrik dalam sel elektrokimia. Bahan yang digunakan dalam sel ini bermacam-macam beberapa di antaranya adalah Ni-Mh, Ni-Cd, Lead-Acid dan Li-Ion. Perkembangan teknologi terutama pada rapat energi per satuan volume menjadikan banyaknya jenis baterai tersebut. Namun, semua baterai memiliki masa pakai yang terbatas yang dapat menjadi lebih singkat apabila diberi beban dan perlakuan yang tidak terkontrol. Peristiwa berkurangnya performansi baterai disebut sebagai degradasi baterai.

1.2 Permasalahan

Degradasi pada sistem baterai kendaraan listrik merupakan suatu hal yang tidak dapat dihindari. Keberadaan degradasi pada sistem baterai membuat pengguna kendaraan listrik harus menyediakan kapasitas baterai yang lebih besar dari kebutuhan energi nominal kendaraan. Oleh karena itu, diperlukan suatu sistem yang dapat memodelkan degradasi baterai pada kendaraan listrik agar pemakai kendaraan dapat memperkirakan kapasitas baterai yang perlu disediakan serta menentukan kapan baterai berada dalam keadaan di mana degradasi baterai sudah tidak dapat ditoleransi. Permasalahan yang berusaha kami pecahkan pada penelitian ini terbentuk dari beberapa permasalahan lain yang lebih kecil, yang dijelaskan oleh diagram *fishbone* berikut.



Gambar 1.2 Diagram *fishbone* untuk Kasus Sepeda Listrik *e-cleave*

1.3 Tujuan dan Sasaran

Tujuan dari penelitian ini adalah untuk memodelkan degradasi sistem baterai pada kendaraan listrik roda dua berbasis pembelajaran mesin. Adapun beberapa sasaran dari penelitian tugas akhir ini adalah

1. Mengembangkan sistem data *logging* yang dapat membaca parameter operasional kendaraan listrik roda dua.
2. Membuat algoritma pembelajaran mesin yang dapat memodelkan waktu degradasi baterai kendaraan listrik roda dua berdasarkan parameter berkendara kendaraan listrik roda dua.
3. Menentukan optimasi hasil model prakiraan degradasi kapasitas baterai pada kendaraan listrik roda dua.

1.4 Pembatasan Masalah

Ruang lingkup penelitian tugas akhir yang dilakukan selanjutnya diperjelas dengan melakukan pembatasan masalah sebagai berikut:

- a. Objek penelitian berupa baterai Li-Ion LG HG2 yang dipakai pada sepeda listrik roda dua rakitan dengan motor listrik berdaya 1000 Watt yang berada di Laboratorium Manajemen Energi.
- b. Data untuk estimasi kondisi muatan baterai berupa data pengukuran aktual berdasarkan eksperimen langsung pada unit.
- c. Sistem manajemen termal pada kabinet baterai tidak dikontrol.

- d. Pengambilan data dilakukan dengan kondisi berkendara normal yaitu, tarikan gas fluktuatif dan sesekali berhenti.
- e. Efek degradasi dari siklus pengisian diabaikan.

1.5 Manfaat Penelitian

Hasil dari penelitian tugas akhir ini diharapkan dapat memiliki manfaat sebagai berikut:

- a. Pustaka pembelajaran mesin yang digunakan akan menjadi tambahan pustaka pembelajaran mesin di Laboratorium Manajemen Energi TF - ITB.
- b. Implementasi yang mudah dari metode estimasi kapasitas membuat model ini dapat langsung diaplikasikan pada *embedded system* kendaraan listrik.
- c. Data hasil analisis degradasi baterai dapat digunakan sebagai acuan bagi penelitian dan/atau tugas akhir selanjutnya.

1.6 Metodologi Penelitian

Pada awal penelitian, kami melakukan studi literatur yang bertujuan untuk mencari informasi dan pengetahuan mengenai baterai secara umum. Literatur yang kami gunakan umumnya berupa karya ilmiah yang telah dipublikasikan pada suatu jurnal tertentu, atau laporan tugas akhir dari orang lain yang memiliki tema seputar baterai isi-ulang dan parameter-parameter pendukung lainnya. Kami juga menggunakan beberapa media pembelajaran video interaktif sebagai cara untuk mendapat pengetahuan fundamental mengenai algoritma sistem manajemen baterai. Keseluruhan proses studi literatur ini merupakan kesempatan bagi kami untuk dapat membentuk pengetahuan yang kokoh terhadap baterai isi-ulang.

Untuk dapat menentukan degradasi sistem baterai, kami harus dapat memonitor beberapa parameter Kondisi Muatan saat baterai sedang digunakan. Beberapa parameter tersebut antara lain tegangan sirkuit terbuka tiap sel dan arus yang ditarik keluar oleh motor listrik. Untuk itu, kami perlu membuat suatu sistem *monitoring* yang dapat mengakuisisi data secara *real-time* agar didapat nilai parameter yang sesuai dengan beban yang diterima oleh sepeda.

Setelah membuat sistem *monitoring*, kami melakukan prosedur pengujian baterai secara langsung dengan menggunakan sepeda listrik pada kurun waktu tertentu. Pada sepeda listrik yang kami gunakan, sistem manajemen baterai tersusun atas 65 baterai Li-ion LG HG2 dengan suatu susunan yang membuat modul baterai memiliki kapasitas energi sekitar 702Wh. Motor listrik yang digunakan pada sepeda membutuhkan daya sekitar 1000W. Dengan demikian satu siklus baterai yang kami lakukan kira-kira memerlukan waktu sekitar 45 menit.

Terakhir, data yang kami dapatkan akan kami olah dengan model pembelajaran mesin. Terdapat banyak metode yang tersedia dalam mengolah data dengan pembelajaran mesin, namun untuk penelitian ini, kami akan menggunakan metode regresi. Metode regresi sendiri bukanlah suatu cara spesifik dalam mengolah data, sebab metode ini masih dapat dibagi-bagi menjadi beberapa cara.

BAB II

TINJAUAN PUSTAKA

2.1 Baterai

2.1.1 Terminologi Umum Baterai

Dalam mempelajari baterai, terdapat beberapa terminologi yang mungkin terdengar cukup umum namun memiliki definisi yang sedikit berbeda.

- Sel, Modul, dan Pak
- Tegangan dan Kapasitas Nominal
- *C Rate*
- Energi dan Daya
- Energi Spesifik dan Densitas Energi

Pada laporan tugas akhir ini, sel baterai yang akan dibahas adalah baterai Li-ion, yang merupakan baterai dengan energi spesifik dan densitas energi yang jauh lebih besar dibanding seluruh baterai elektrokimia yang pernah dibuat[2].

2.1.2 Komponen Sel Elektrokimia

Seperti yang telah dijelaskan sebelumnya, seluruh penjelasan mengenai baterai pada laporan ini akan merujuk pada baterai Li-ion yang umum digunakan pada banyak kendaraan listrik. Layaknya baterai elektrokimia pada umumnya, baterai Li-ion memiliki komponen yang tidak jauh berbeda dengan yang lain. Terdapat lima komponen dalam suatu baterai elektrokimia, kelima komponen ini adalah elektroda negatif, elektroda positif, elektrolit, separator, dan pengumpul arus. Kedua elektroda yang disebutkan sebelumnya, bertindak untuk melepas dan menerima elektron dan kation-anion. Elektroda negatif akan melepas elektron ke sirkuit elektrik dan kation ke elektroda positif ketika proses pengisian. Pada saat yang sama, elektroda positif akan menerima elektron melepas anion yang akan bergerak menuju ke elektroda negatif. Proses yang berbalikan akan terjadi pada saat proses pengosongan berlangsung. Komponen berikutnya adalah elektrolit, yang merupakan medium perpindahan kation dan anion saat proses pengisian dan

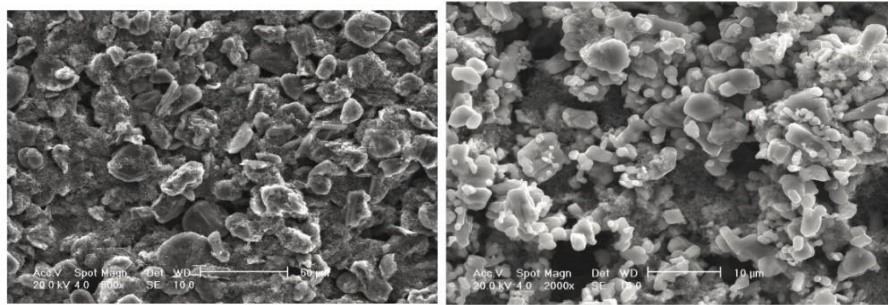
pengosongan sedang berlangsung. Elektrolit umumnya merupakan larutan yang ditambahkan dengan larutan kimia tertentu sehingga memiliki konduktivitas ionik. Elektrolit juga harus bertindak sebagai insulator elektrik agar dapat menghindari fenomena *self discharge* pada bagian internal baterai. Selanjutnya adalah separator, yang harus bertindak sebagai pemisah antara elektroda positif dan negatif sehingga tidak terjadi rangkaian arus pendek. Separator umumnya terbuat dari material plastik seperti *polyethylene* dan *polyester*. Terakhir adalah pengumpul arus, yang merupakan tempat di mana elektroda positif dan negatif melekat. Pengumpul arus ini mengantarkan arus listrik dari terminal sel ke rangkaian listrik eksternal.

2.1.3 Mekanisme Kerja Baterai

Secara umum, prinsip kerja baterai Li-ion dengan baterai elektrokimia lainnya, cukup berbeda. Pada baterai Li-ion terdapat mekanisme interkalasi di mana kation dan anion akan masuk ke dalam elektroda seperti air yang masuk ke dalam spons. Ion-ion ini akan masuk ke celah-celah kristalin elektroda tanpa mengubah struktur sama sekali[3]. Hal ini cukup berbeda dengan baterai elektrokimia lain, di mana terjadi reaksi redoks yang akan membuat ion mengikis dan mengendap pada elektroda, sehingga dapat mengurangi usia baterai. Terdapat dua ketentuan yang harus dimiliki oleh elektroda baterai Li-ion. Pertama elektroda harus memiliki struktur kristal terbuka yang dapat menerima sisipan ion. Kedua, elektroda harus memiliki kemampuan untuk menerima elektron dari sirkuit eksternal.

Selama proses pengosongan, ion Li^+ akan keluar dari elektroda negatif menuju ke elektroda positif melewati larutan elektrolit. Di saat yang sama elektron akan keluar ke sirkuit eksternal, hal ini menetralkan muatan pada elektroda negatif. Pergerakan Li^+ menuju elektroda positif bertujuan untuk menetralkan muatan pada elektroda positif yang telah menerima elektron dari sirkuit eksternal. Hal yang berbalikan terjadi saat proses pengisian, yaitu saat elektron dan ion Li^+ bergerak dari elektroda positif menuju elektroda negatif[3]. Perlu diingat dalam semua proses ini, elektroda bukanlah suatu blok homogen, namun terdiri atas jutaan partikel yang sangat kecil. Semakin kecil partikel elektroda, semakin besar luas permukaan

partikel, dan semakin kecil pula resistansi sel. Resistansi yang kecil ini akan meningkatkan daya baterai.



Gambar 2.1 Grafit, material yang umum digunakan sebagai elektroda negatif (kiri) dan Lithium Manganese Oxide (LMO), material penyusun elektroda positif (kanan)

2.1.4 Mekanisme Degradasi Baterai

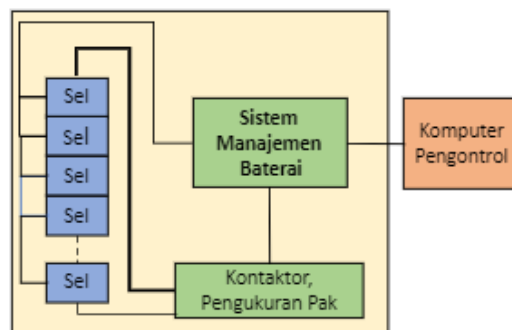
Saat sel baterai melakukan proses interkalasi, terdapat beberapa dampak yang ditimbulkan terhadap bagian internal dari baterai sehingga baterai mengalami degradasi. Salah satu faktor paling mempengaruhi degradasi adalah pembentukan Lapisan *Solid Electrolyte Interphase* (SEI) berlebihan[4]. Lapisan SEI merupakan lapisan tipis yang membatasi elektroda dan elektrolit. Ketika proses interkalasi berlangsung, akan terbentuk lapisan SEI tambahan sehingga formasi lapisan SEI menjadi tidak seragam dan menjadi berlebihan[5]. Lapisan SEI yang berlebihan ini dapat mengakibatkan pertumbuhan dendrit, keretakan, dan penurunan jumlah lithium yang mengalami proses interkalasi ke elektroda. Beberapa faktor lain yang menyebabkan degradasi sel adalah korosi *current collector* dan kegagalan mekanik baterai [6].

Pembentukan lapisan SEI merupakan mekanisme degradasi baterai yang terjadi pada level molekuler. Namun, sebelum terjadinya degradasi pada level molekuler, terdapat empat hal yang menyebabkan sel baterai mengalami degradasi. Temperatur, Keadaan Muatan (KM), *C-rate*, dan Kedalaman Pengosongan (KP) menjadi empat faktor utama degradasi baterai. Temperatur kerja baterai dapat menyebabkan banyak reaksi molekuler yang membuat baterai terdegradasi. Perubahan fasa permukaan elektroda, terjadinya fase presipitasi, hilangnya material aktif, dan pembentukan lapisan SEI menjadi beberapa hal yang membuat baterai

terdegradasi apabila dioperasikan pada temperatur tinggi. Kedua, Kondisi Muatan pada nilai yang tinggi dapat menyebabkan pertumbuhan lapisan SEI dan korosi pada *current collector*. Ketiga adalah *C-rate* yang tinggi, yang dapat menyebabkan terbentuknya lapisan metalik lithium pada permukaan anode. Keempat, adalah Kedalaman Pengosongan, yaitu penurunan Kondisi Muatan baterai pada rentang yang cukup besar. Kedalaman Pengosongan yang besar dapat menyebabkan baterai mengalami kegagalan secara mekanik [3].

2.2 Sistem Manajemen Baterai

Secara umum, Sistem Manajemen Baterai (SMB) merupakan suatu sistem tertanam yang terdiri atas rangkaian elektronika untuk menjalankan suatu aplikasi yang spesifik. SMB memiliki beberapa tujuan, yakni melindungi sel baterai dan pengguna dari risiko kecelakaan, mengoptimalkan usia baterai, mempertahankan baterai agar selalu pada kondisi yang dapat memenuhi ketentuan desain, dan mengkomunikasikan beberapa parameter baterai saat sedang digunakan [7].

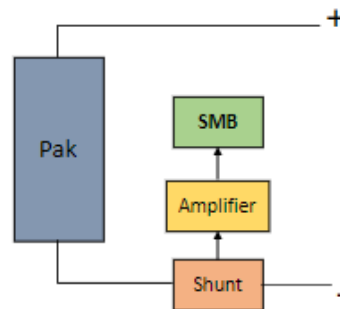


Gambar 2.2 Skema umum SMB

2.2.1 Battery Pack Sensing

Fungsi pertama dari suatu SMB adalah untuk mengukur nilai dari beberapa parameter baterai [7]. Parameter pertama yang sangat penting untuk diukur oleh suatu SMB adalah tegangan. Dalam mengukur tegangan, kita harus mengukur semua tegangan sel baterai. Hal ini karena tegangan memberikan informasi yang membuat kita dapat mengestimasi Keadaan Muatan (KM), dan Keadaan Kesehatan (KK) dari baterai.

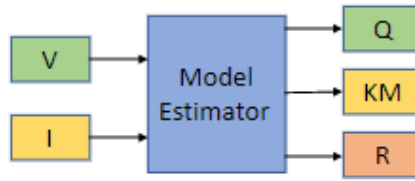
Satu parameter yang cukup penting untuk diperhatikan, namun bukan pada tingkat sel baterai melainkan pada pak baterai, adalah arus listrik. Pengukuran arus pada pak baterai diperlukan untuk dapat memonitor keamanan baterai, estimasi KM, dan estimasi KK. Sama seperti temperatur, untuk dapat mengukur arus, kita harus mengukur tegangan. Dalam penelitian ini kami menggunakan metode *relative shunt*. *Relative shunt*, merupakan metode yang mengukur arus listrik langsung pada rangkaian pak baterai secara seri. Agar tidak menghasilkan kehilangan daya yang besar, maka resistansi sensor harus cukup kecil. Resistansi yang kecil ini menyebabkan tegangan yang terukur akan kecil, sehingga nilainya harus diperbesar. Satu kelebihan dari *relative shunt* adalah, sensor ini tidak memiliki *offset* pada pengukuran 0 Ampere. Namun kekurangannya adalah, sensor ini tersambung langsung dengan rangkaian SMB, sehingga perlu mekanisme tambahan agar keamanan dari sensor ini dapat terjaga.



Gambar 2.3 Skema *Shunt-Current* sensor

2.2.2 Pengelolaan Performa

Performa yang akan dikelola pada SMB meliputi estimasi KM, komputasi batas-daya, dan penyeimbangan tegangan sel baterai. Untuk dapat mengelola performa kendaraan listrik, dua kuantitas harus diketahui, energi dan daya. Untuk dapat mengetahui dua kuantitas tersebut, diperlukan estimasi atas beberapa parameter seperti jumlah muatan, KM baterai, resistansi baterai. Namun sekali lagi, ketiga parameter di atas juga tidak dapat diestimasi secara langsung, diperlukan pengukuran atas tiga besaran utama, yaitu tegangan, arus, dan temperatur, yang pada penelitian ini hanya ada tegangan dan arus.



Gambar 2.4 Skema estimasi energi dan daya

Dalam melakukan estimasi ini, kita harus dapat memastikan bahwa estimasi yang kita lakukan sudah tepat, dengan kata lain pengukuran tegangan dan arus. Adapun dampak dari estimasi yang buruk adalah, koreksi nilai yang mendadak saat tegangan atau arus telah mencapai nilai maksimum, yang tentunya akan mempengaruhi persepsi pengguna kendaraan dan berimbas pada pengendaraan kereta yang buruk. Data yang buruk juga dapat merusak sel karena mengakibatkan kelebihan muatan atau kekurangan muatan.

Selanjutnya, akan ditawarkan satu metode konvensional untuk mengestimasi total energi sel dan total daya sel. Pada metode konvensional, estimasi energi pada sel baterai dilakukan dengan menjumlahkan tegangan sirkuit terbuka pada rentang KM tertentu. Perhitungan ini dijelaskan dengan persamaan,

$$E(t) = Q \int_{z_{min}}^{z(t)} OCV(\xi) d\xi \approx QV_{nom} \Delta z \quad (2.1)$$

2.2.3 Diagnostik

Pada fungsi yang terakhir ini, SMB akan mendiagnostik dua parameter, yaitu kondisi kesehatan dan kondisi hidup baterai[8]. Untuk kondisi kesehatan, indikator yang dapat memberikan informasi penting adalah degradasi kapasitas dan pemudaran daya. Umumnya, baterai sudah tidak dapat digunakan ketika kapasitas telah memudar sekitar 20% dan resistansi telah bertambah 50% - 100%. Degradasi pada baterai li-ion sangat tergantung pada cara penggunaan oleh pengguna, dan hal ini akan berdampak pada efek lebih lanjut yang menyebabkan degradasi menjadi tidak linear[9]. Parameter kedua adalah kondisi hidup, yang akan memprediksi seberapa lama usia baterai dalam sistem penanggalan waktu[8]. Diagnostik yang dilakukan SMB juga harus dapat mendeteksi kegagalan yang umum terjadi seperti kegagalan sensor tegangan, arus, dan temperatur kegagalan sistem penyeimbang

tegangan, kegagalan kontaktor, kegagalan sistem pendinginan, dan kegagalan komunikasi ke pusat data.

2.3 Sepeda Listrik

Selain kendaraan listrik roda empat, salah satu kendaraan yang umum menggunakan energi listrik adalah sepeda. Sepeda listrik merupakan satu transportasi yang konsepnya hampir sama dengan motor atau skuter listrik, di mana terdapat motor DC yang akan membantu kayuhan pedal sepeda sehingga pengguna tidak perlu menggunakan energi cukup besar dalam mengayuh. Sepeda listrik umumnya digunakan untuk memudahkan mobilisasi dengan jarak tempuh yang tidak terlalu jauh, berbeda dengan motor, skuter, ataupun mobil listrik. Berdasarkan riset yang dilakukan di Swiss, penggunaan sepeda listrik mampu menurunkan konsumsi gas rumah kaca hingga 10%. Angka ini akan menjadi lebih besar menjadi 17,51% apabila mobil listrik diikutsertakan ke dalam perhitungan[10].

Meskipun memiliki banyak keterbatasan, sepeda listrik tetap memiliki banyak manfaat bagi penggunanya. Dampak positif untuk kesehatan, transportasi yang lebih ekonomis, peningkatan ruang terbuka di kota-kota besar, hingga pengurangan kemacetan di jalanan menjadi beberapa faktor yang memberikan keunggulan terhadap sepeda[10].

2.4 Pembelajaran Mesin

Pembelajaran Mesin (*Machine Learning*) merupakan salah satu perkembangan ilmu komputasi yang algoritmanya memiliki kemampuan untuk berkembang berdasarkan pengalaman secara otomatis. Untuk dapat melakukan hal ini pada pembelajaran mesin dibuat suatu model matematis dari suatu set data bernama data latih (*training data*). Kemampuan pembelajaran mesin untuk menentukan algoritmanya sendiri mempermudah penyelesaian kasus yang tidak mudah diselesaikan dengan metode penyusunan algoritma konvensional.

Pembelajaran mesin memiliki beragam tipe yang berbeda berdasarkan baik model pendekatan algoritmanya, pengaplikasiannya maupun bagaimana

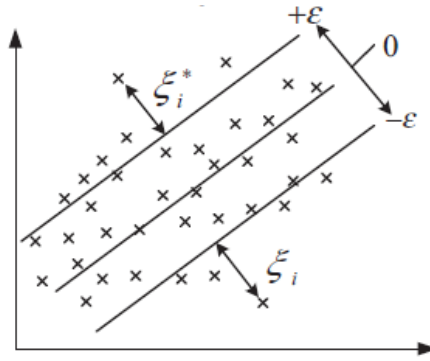
pembelajaran mesin ini mengolah data. Namun, secara garis besar pembelajaran mesin dibagi menjadi beberapa tipe, dua tipe yang paling populer adalah *supervised learning* dan *unsupervised learning*. Pada *supervised learning*, mesin dilatih dengan menggunakan data yang telah dibuat berdasarkan karakteristik atau pelabelan tertentu, berdasarkan data ini mesin akan belajar untuk mendapatkan bagaimana perilaku dari data tersebut baik sebagai kelompok data maupun tren dari data. Berbeda dengan *supervised learning*, pada *unsupervised learning*, pengguna tidak memasukan karakteristik atau pelabelan tertentu. Pada *unsupervised learning*, mesin lah yang akan mempelajari bagaimana data kelakuan dari data tersebut.

Perkembangan pembelajaran mesin menciptakan berbagai model yang digunakan untuk melatih data. Beberapa model dari pembelajaran mesin adalah sebagai berikut.

2.4.1 Mesin Vektor Pendukung (MVP) / *Support Vector Machine* (SVM)

Mesin Vektor Pendukung merupakan salah satu model pembelajaran mesin dengan tipe *Supervised Learning* yang dikembangkan oleh Vapnik dan rekan-rekannya pada tahun 1992-1997, untuk menjadi alat penyelesaian masalah terkait klasifikasi dan regresi. MVP dikenal akan keandalannya melakukan prediksi dengan akurasi tinggi walau dengan data sedikit [11][12].

Pada penelitian ini akan digunakan Regresi Vektor Pendukung (RVP), yang tidak memiliki perbedaan signifikan dengan MVP. Pada dasarnya, algoritma RVP bekerja dengan tujuan untuk memproyeksikan data latih ke dimensi yang lebih tinggi sehingga akan dihasilkan bidang linear yang menggeneralisasikan data dengan lebih baik. Pada bentuk yang paling sederhana, RVP hanya dapat melakukan regresi secara linear.



Gambar 2.5 RVP Linear

Pada gambar di atas, dapat dilihat bahwa garis regresi diikuti oleh dua garis margin. RVP dapat dikatakan sebagai algoritma yang dalam kinerjanya memiliki *constrained optimization*. RVP harus berusaha untuk mendapatkan sebanyak mungkin data dalam proses pelatihan dan di sisi lain harus berusaha untuk memaksimalkan margin. Apabila margin semakin besar, yang juga berarti epsilon semakin besar, tidak akan dihasilkan prediksi yang baik karena model belajar dari data yang besar. Di sisi lain, apabila epsilon bernilai terlalu kecil, model akan mengalami *overfitting*, dan data yang diprediksi akan jauh dari yang diharapkan. Berikut adalah persamaan yang berusaha untuk dioptimalkan oleh algoritma RVP.

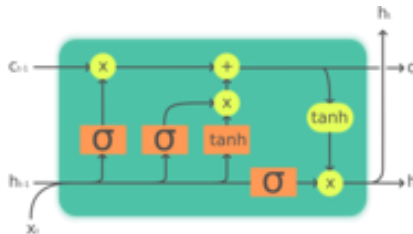
$$\min_{\alpha, \alpha^*} \frac{1}{2} \sum_{i,j=1}^l (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)k(x_i, x_j) + \epsilon \sum_{i=1}^l (\alpha_i^* + \alpha_i) - \sum_{i=1}^l y_i(\alpha_i^* - \alpha_i) \quad (2.2)$$

Pada persamaan di atas, alpha merupakan vektor pengali Lagrange yang telah disubstitusi untuk menggantikan variabel lain agar proses optimisasi dapat dilakukan dengan variabel yang lebih sederhana. Fungsi $k(x_i, x_j)$ merupakan kernel, suatu fungsi nonlinear yang memproyeksikan data ke dimensi yang lebih tinggi. Beberapa kernel yang akan digunakan dalam penelitian ini adalah kernel linear, polinomial, dan gaussian.

Pada pembelajaran mesin dikenal pula istilah *hyperparameter*. *Hyperparameter* ini adalah suatu nilai yang mengontrol jalannya algoritma pembelajaran mesin. Setiap model dari pembelajaran mesin memiliki *hyperparameter*-nya sendiri, misal pada MVP menggunakan *hyperparameter* C, dan gamma.

2.4.2 Long-Short Term Memory (Recurrent Neural Network)

Recurrent Neural Network (RNN) merupakan suatu metode dari *Neural Network* yang memanfaatkan sifat alami data yaitu sekuensial [13]. Bentuk RNN yang sederhana hanya dapat mengolah satu sekuens data di waktu yang bersamaan. Dalam RNN yang sederhana, satu unit RNN menerima dua buah input, yaitu data dari unit sebelumnya dan data hasil keluaran unit yang sama. Dengan memasukkan kembali keluaran dari unit, RNN memiliki semacam memori yang dapat mengingat sekuens data. Namun, apabila RNN terdiri atas lapisan yang banyak, beban antara unit pada lapisan yang lebih dalam akan semakin rendah, menyebabkan terjadinya *vanishing gradient*. Untuk menyelesaikan permasalahan ini, LSTM mulai dikenalkan.

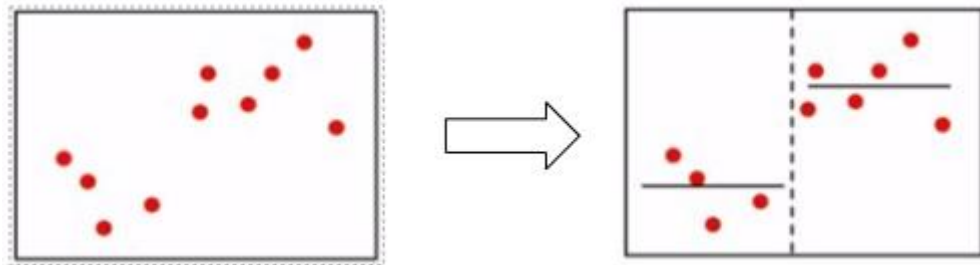


Gambar 2.6 Unit Sel LSTM

Gambar di atas merupakan satu unit sel dari LSTM. Terdapat dua macam fungsi aktivasi yang membangun unit tersebut. Kedua fungsi aktivasi ini adalah sigmoid dan tangen hiperbolik. Pada LSTM terdapat dua fitur pengulangan, yang ditandai dengan 'h' dan 'c'. 'h' merupakan *hidden state*, dan 'c' dinamakan *cell state*. Kedua fitur ini akan dimasukkan ke dalam unit sel seperti yang terlihat pada gambar. Kemudian, fungsi aktivasi akan memisahkan data ke dalam kategori 0 dan 1, yang berarti apakah data yang diinput bersesuaian atau tidak dengan sekuens yang sedang dijalankan. Dengan unit sel ini, LSTM dapat mempelajari pembobotan antara tiap input berdasarkan kesesuaiannya dengan sekuens yang sedang dilatih, sekaligus menyelesaikan permasalahan *vanishing gradient* pada RNN konvensional.

2.4.3 Decision Tree (DT)

Seperti namanya *Decision Tree* adalah model pembelajaran mesin yang menggunakan struktur seperti pohon keputusan sebagai metode pembentukan algoritma yang cocok untuk sebuah data. DT dapat digunakan sebagai metode klasifikasi dan regresi. Pada awalnya, algoritma DT digunakan dalam klasifikasi, lalu pengenalan *regression tree* membuat algoritma ini mampu menangani analisis regresi dan klasifikasi. Pohon regresi dibangun dengan memisah data ke dalam beberapa sampel berdasarkan nilai rata-rata data pada satu wilayah pemisahan. Semakin besar percabangan yang dihasilkan pada DT, semakin kecil pula error antara keluaran yang diestimasi dengan keluaran sebenarnya [14].



Gambar 2.7 Proses Belajar *Decision Tree*

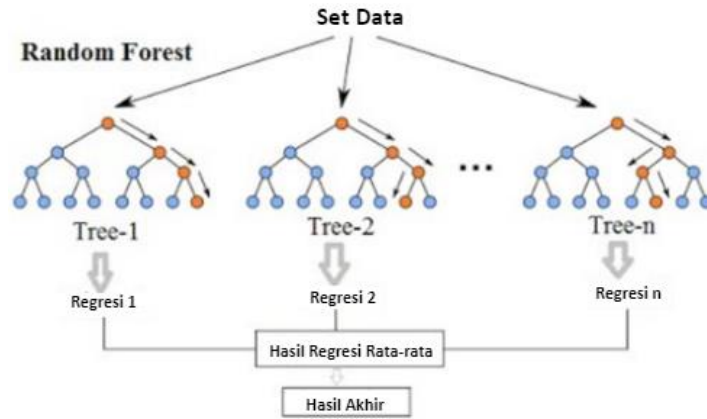
Pada regresi, DT akan bekerja seperti gambar di atas. DT akan membagi data ke dalam beberapa segmen, yaitu garis putus-putus vertikal pada gambar kanan. Segmen ini ditentukan oleh entropi dan indeks gini. Apabila data semakin rumit, maka entropi akan semakin besar, dan akan lebih banyak segmen yang dihasilkan. Setelah dibagi ke dalam beberapa segmen, DT akan mencari garis horizontal yang memiliki varians terkecil pada data pada setiap data dalam segmen. Dengan demikian varians dari seluruh data total akan tereduksi, dan prediksi yang dihasilkan akan memiliki error yang rendah. Berikut adalah persamaan entropi pada DT

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i \quad (2.3)$$

2.4.4 Random Forest (RF)

Random Forest merupakan model pembelajaran mesin yang memanfaatkan banyak DT dalam satu waktu pelatihan data. Penggunaan banyak DT ini akan

mengurangi kemungkinan *overfitting* yang sering terjadi pada model DT individual. Dengan RF, hubungan yang kompleks antara variabel masukan dapat dimodelkan dan relatif lebih *robust* dari pengaruh *outlier* [15]. RF juga sangat berguna untuk digunakan dalam mengestimasi data yang hilang pada suatu set data.



Gambar 2.8 Proses Belajar *Random Forest*

Istilah pada model pembelajaran mesin yang menggunakan banyak model lain untuk dapat belajar adalah *bagging*. Metode ini akan membuat set data masukan dibagi secara acak ke dalam beberapa DT dan dilatih. Pembagian secara acak dan jumlah pembagian yang tidak merata inilah yang membuat model ini random. Hasil akhir dari pelatihan berbagai DT ini akan diambil nilai rata-ratanya dan menjadi solusi regresi akhir. Selain beberapa kelebihan yang telah disebutkan sebelumnya, model RF sayangnya tidak dapat mengingat data dengan cukup baik. Hal ini karena setiap pengulangan akan menggunakan data yang acak, meskipun pada akhirnya menghasilkan nilai rata-rata yang tidak jauh berbeda. Selain itu, solusi akhir dari metode ini akan memiliki varians yang cukup tinggi karena faktor keteracakan di awal pelatihan setiap DT.

2.5 *Approximate Weighted Total Least Square*

Approximate Weighted Total Least Square adalah metode untuk mengestimasi kapasitas dari pak baterai setelah sepeda listrik melakukan beberapa siklus berkendara. AWTLS akan menerima input berupa ΔKM dari siklus berkendara. Dalam penelitian ini, kami menggunakan ΔKM setiap satu menit.

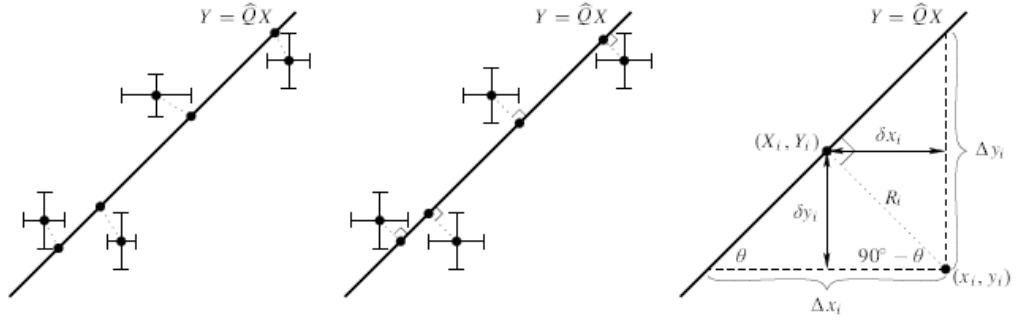
Fungsi AWTLs kami juga menerima input berupa jumlah siklus berkendara yang kapasitasnya ingin diestimasi. Penjelasan lengkap AWTLs akan dimulai dari penjelasan mengenai metode *Least Square* secara umum.

Metode *Least Square*, pada dasarnya merupakan metode regresi linear sederhana untuk membuat sebuah garis lurus yang merepresentasikan persebaran data pada domain x dan y [16]. Metode regresi yang paling sederhana adalah *Ordinary Least Square* (OLS), yang hanya mempertimbangkan *noise* dari variabel di sumbu y . Regresi yang dilakukan oleh OLS bertujuan untuk mengestimasi kapasitas dari sebuah sel dengan meminimalkan galat kuadrat total Δy . Berbeda halnya dengan *Total Least Square* (TLS), metode ini memperhitungkan *noise* pengukuran pada variabel di sumbu x . Pengembangan dari kedua metode ini yang memberikan hasil estimasi lebih baik adalah *Weighted Least Square* (WLS) atau dinamakan WTLS untuk apabila diaplikasikan pada TLS. Metode OLS dan TLS mempunyai asumsi bahwa *variance* dalam *error* pengukuran bernilai konstan. Dengan WLS, kita akan memberikan ‘bobot’ pada hasil pengukuran. Dengan WLS, estimasi yang akan dilakukan bukan lagi untuk meminimalkan galat kuadrat total Δy , namun menjadi meminimalkan *cost function* yang memiliki komponen pembobotan semakin besar apabila *variance* dalam *error* pengukuran bernilai semakin kecil. Kedua *cost function* WLS dan WTLS adalah sebagai berikut. Dengan ‘bobot’ adalah $1/\sigma^2$ dan λ adalah *Lagrange multiplier*.

$$\chi_{\text{WLS}}^2 = \sum_{i=1}^N \frac{(y_i - Y_i)^2}{\sigma_{y_i}^2} = \sum_{i=1}^N \frac{(y_i - \hat{Q}x_i)^2}{\sigma_{y_i}^2} \quad (2.4)$$

$$\chi_{\text{WTLS},a}^2 = \sum_{i=1}^N \frac{(x_i - X_i)^2}{\sigma_{x_i}^2} + \frac{(y_i - Y_i)^2}{\sigma_{y_i}^2} - \lambda_i (Y_i - \hat{Q}X_i) \quad (2.5)$$

Persamaan *cost function* inilah yang dapat diturunkan menjadi persamaan yang dapat mengestimasi kapasitas baterai dalam beberapa siklus awal.



Gambar 2.9 Perbedaan pada WTLS (Kiri), TLS (Tengah), dan AWTLS (Kanan) dalam Mengalkulasi Error Pengukuran

Gambar di atas akan membantu dalam memberikan penjelasan terhadap metode AWTLS yang akan kami gunakan pada pengerjaan tugas akhir ini. Pertama, gambar di sebelah kiri merupakan WTLS yang telah kita kembangkan sebelumnya. Pada AWTLS, *error bar* pada setiap titik data memiliki panjang yang berbeda untuk komponen x dan y. *Error bar* ini proporsional dengan standar deviasi, yang kita perlakukan secara berbeda antara satu sama lain. Kedua, gambar di tengah merupakan satu kasus spesifik TLS di mana standar deviasi x dan y memiliki korelasi. Terlihat bahwa *error bar* x dan y bernilai sama, membuat titik data (x,y) tegak lurus dengan garis regresi $Y = QX$. Ketiga, gambar di sebelah kanan merupakan AWTLS yang akan kita buat. AWTLS akan menggabungkan solusi rekursif yang dimiliki oleh TLS dengan cara membuat titik data tegak lurus dengan garis regresi, sekaligus memberikan bobot terhadap standar deviasi kedua variabel [16]. AWTLS akan dibangun dengan observasi ortogonalitas untuk mendapatkan *cost function* yang mendukung kedua properti WTLS dan TLS [17].

Apabila melihat kembali gambar di sebelah kanan, komponen x dan y yang memberikan jarak tegak lurus antara titik dengan garis adalah δx dan δy . *Cost function* AWTLS akan dibentuk berdasarkan pembobotan kedua jarak tersebut.

$$\chi_{\text{AWTLS}}^2 = \sum_{i=1}^N \frac{\delta x_i^2}{\sigma_{x_i}^2} + \frac{\delta y_i^2}{\sigma_{y_i}^2} \quad (2.6)$$

Selanjutnya δx dan δy akan ditulis sehingga memiliki komponen Q di dalamnya, yang nantinya akan dioptimasi. Hasil *cost function* AWTLS akhirnya menjadi

$$\chi_{\text{AWTLS}}^2 = \sum_{i=1}^N \frac{(y_i - \hat{Q}x_i)^2}{(1 + \hat{Q}^2)^2} \left(\frac{\hat{Q}^2}{\sigma_{x_i}^2} + \frac{1}{\sigma_{y_i}^2} \right) \quad (2.7)$$

Fungsi di atas dapat langsung diturunkan secara parsial terhadap Q tanpa harus mengalami penurunan terhadap X, Y, dan λ seperti yang terjadi pada WTLS.

$$\begin{aligned} \frac{\partial \chi_{\text{AWTLS}}^2}{\partial \hat{Q}} = & \frac{2}{(\hat{Q}^2 + 1)^3} \sum_{i=1}^N \hat{Q}^4 \left(\frac{x_i y_i}{\sigma_{x_i}^2} \right) + \hat{Q}^3 \left(\frac{2x_i^2}{\sigma_{x_i}^2} - \frac{x_i^2}{\sigma_{y_i}^2} - \frac{y_i^2}{\sigma_{x_i}^2} \right) \\ & + \hat{Q}^2 \left(\frac{3x_i y_i}{\sigma_{y_i}^2} - \frac{3x_i y_i}{\sigma_{x_i}^2} \right) + \hat{Q} \left(\frac{x_i^2 - 2y_i^2}{\sigma_{y_i}^2} + \frac{y_i^2}{\sigma_{x_i}^2} \right) + \left(\frac{-x_i y_i}{\sigma_{y_i}^2} \right) \end{aligned} \quad (2.8)$$

$$\begin{aligned} \frac{\partial \chi_{\text{AWTLS}}^2}{\partial \hat{Q}} = & \frac{2}{(\hat{Q}^2 + 1)^3} \left(c_5 \hat{Q}^4 + (2c_4 - c_1 - c_6) \hat{Q}^3 \right. \\ & \left. + (3c_2 - 3c_5) \hat{Q}^2 + (c_1 - 2c_3 + c_6) \hat{Q} - c_2 \right) \end{aligned} \quad (2.9)$$

Dengan penyederhanaan variabel c di atas, solusi atas Q dapat dihitung secara rekursif. Terakhir, persamaan kuartik untuk mencari kandidat nilai Q adalah,

$$c_5 \hat{Q}^4 + (2c_4 - c_1 - c_6) \hat{Q}^3 + (3c_2 - 3c_5) \hat{Q}^2 + (c_1 - 2c_3 + c_6) \hat{Q} - c_2 = 0 \quad (2.10)$$

BAB III

METODOLOGI PENELITIAN

3.1 Permasalahan

Untuk menentukan daerah yang dapat digunakan untuk pengambilan data pada penelitian ini, ditentukan beberapa tahapan survei yang digunakan. Survei awal adalah menentukan rute di Bandung Raya yang memiliki sudut kemiringan yang beragam. Parameter ini didapat baik secara survei lapangan, wawancara maupun survei menggunakan piranti lunak *Google Earth Pro*. Setelah parameter pertama ini memenuhi untuk dilakukan penelitian, dilakukan pengambilan data. Penelitian ini menggunakan data berupa rute perjalanan (lokasi, perubahan elevasi, waktu tempuh) dan data kelistrikan awal dan akhir penggunaan serta saat kondisi baterai penuh dan hampir kosong akibat pemakaian.

3.2 Spesifikasi Kendaraan Listrik Roda Dua

Dalam membuat sistem manajemen baterai untuk kendaraan listrik roda dua, dalam penelitian ini adalah sepeda listrik, kami membagi penjelasan spesifikasi ini ke dalam tiga subbab, yaitu sepeda listrik, komponen pengambil data sistem manajemen baterai, dan komponen komunikasi data sistem manajemen baterai.

3.2.1 Sepeda Listrik

Dalam pelaksanaan tugas besar ini, kami akan menggunakan sepeda dengan merek Thrill Cleave 2.0 sebagai kendaraan uji coba. Thrill merupakan salah satu jenis merek sepeda gunung yang diproduksi di Jepang dan cukup banyak dijual di Indonesia. Rangka sepeda ini terbuat dari material Alloy 6061, membuat sepeda ini cukup ringan dibanding sepeda lain pada umumnya. Untuk keperluan penelitian Lab ME, sepeda ini telah dilengkapi dengan motor listrik pada roda belakang dan 5 mode *pedal assist* yang dapat membantu pengendara dalam mengendarai sepeda.



Gambar 3.1 Sepeda Gunung Thrill

3.2.2 Sistem Baterai

Sepeda listrik menggunakan pak baterai dengan sel LG-HG2 yang merupakan baterai Lithium-Ion dengan jenis katoda LiNiMnCoO_2 atau disebut juga baterai NMC. Adapun spesifikasi dari sel LG-HG2 adalah sebagai berikut.

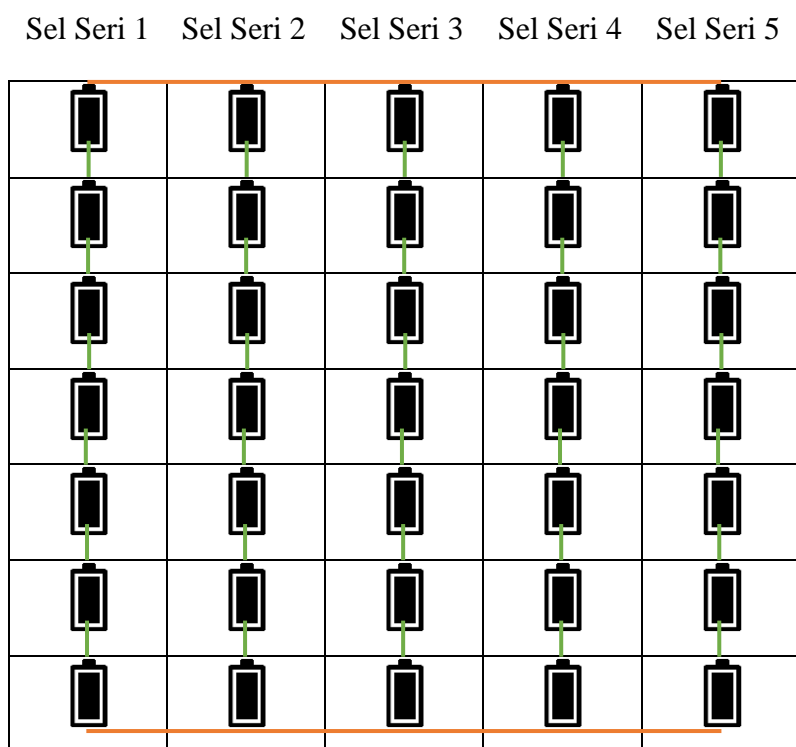
Tabel 3.1 Spesifikasi Baterai LG-HG2

Variabel	Keterangan	Spesifikasi
Kapasitas Nominal	Pengisian/ pengosongan standar	3 Ah
Tegangan Nominal	Rata-rata untuk pengosongan standar	3,60 V
Pengisian Standar	Arus Konstan	1.500mA
	Tegangan Konstan	4,2 V
	Arus Akhir (<i>Cut-Off</i>)	50 mA
Pengosongan Standar	Arus Konstan	600mA
	Tegangan Akhir (<i>Cut-Off</i>)	2,5 V
Arus Pengosongan Maksimum	Untuk pengosongan kontinu	20 A
Berat	Maksimum	47 gram
Suhu Operasi	Pengisian	0 – 50 °C
	Pengosongan	-20 – 75 °C



Gambar 3.2 Sel baterai LG-HG2 [battery-bro,2020]

Baterai sebanyak 65 buah disusun 13 seri dan 5 paralel sehingga dapat menyimpan energi sebesar 702 Wh.



Gambar 3.3 Susunan pak baterai sepeda listrik

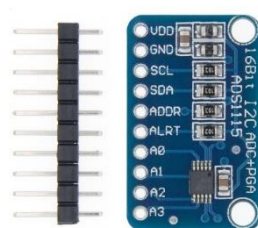


Gambar 3.4 Penempatan pak baterai yang sudah diberi wadah pada sepeda listrik

3.2.3 Komponen Pemantau Data

Pada pengerjaan Tugas Akhir ini, komponen yang cukup vital adalah sensor tegangan dan arus. Dalam membuat sistem manajemen baterai ini kami menggunakan sensor ADS1115, suatu sensor yang dikembangkan oleh Texas Instrument, untuk melakukan pengawasan terhadap tegangan dan arus pada baterai. Beberapa fitur yang dimiliki oleh sensor ini adalah,

- Temperatur operasi antara -40°C sampai 120°C .
- Rentang tegangan operasi antara 2 – 5,5V.
- Konsumsi arus yang rendah, 150uA.
- Sistem komunikasi I2C, di mana tersedia 4 buah pin pada komponen yang mendukung protokol komunikasi ini.



Gambar 3.5 Papan ADS11x dan *pin heading*

ADS dengan seri 111x mampu melakukan konversi data dengan kecepatan hingga 860 sampel per detik. *Programmable Gain Amplifier* (PGA) komponen ini menawarkan rentang input yang sangat luas yaitu dari ± 256 mV hingga $\pm 6,144$ mV. Rentang yang sangat luas ini mendukung sinyal pengukuran yang sangat kecil, seperti nilai arus pada pak baterai yang telah dikonversi menjadi tegangan karena dilewatkan pada *shunt resistor*.

3.2.4 Komponen Komunikasi

Komponen komunikasi yang kami gunakan pada penelitian ini, yang sekaligus berfungsi sebagai mikrokontroler, adalah ESP 32. ESP 32 merupakan suatu *chip* mikrokontroler yang dapat digunakan dalam pengaplikasian *Internet-of-Things*. Mikrokontroler ini memiliki fitur seperti penggunaan daya yang rendah, berbagai mode pengisian daya, dan mendukung berbagai protokol komunikasi seperti *bluetooth*. ESP32 juga dapat diintegrasikan dengan Wi-Fi, sehingga mendukung pemakaian protokol komunikasi dalam jaringan.



Gambar 3.6 Mikrokontroler DOIT ESP32 DevKit V1

- Microcontroller: Tensilica 32-bit Single-/Dual-core CPU Xtensa LX6
- Operating Voltage: 3,3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 25
- 6 Analog Input Pins (ADC), 2 Analog Outputs Pins (DAC)

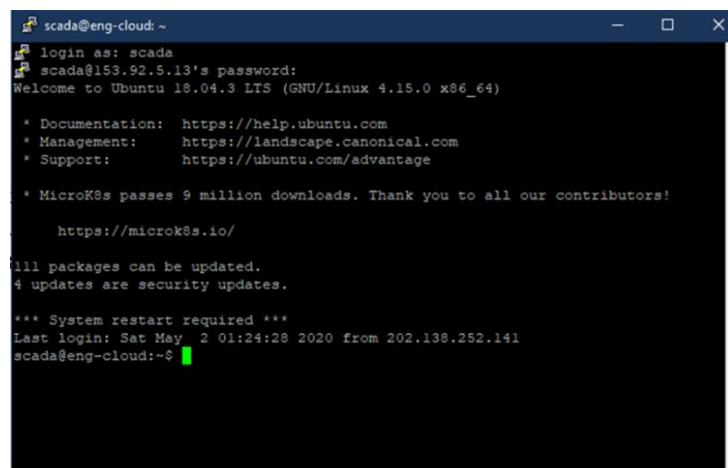
3.3 Komponen Perangkat Lunak Pendukung

3.3.1 Arduino IDE

Pada penelitian ini, kami menggunakan ESP 32 alih-alih *board* Arduino yang didesain untuk perangkat lunak ini. Penggunaan Arduino IDE untuk ESP 32 memerlukan suatu *package* tertentu yang dapat diunduh pada *Boards Manager* Arduino IDE. *Package* tambahan ini telah disiapkan oleh Espressif Systems selaku perusahaan manufaktur ESP32 agar *board* ESP32 menjadi kompatibel dengan program pada Arduino IDE.

3.3.2 PuTTY

PuTTY adalah perangkat lunak sumber terbuka yang berfungsi sebagai emulator terminal, konsol serial, dan transfer berkas berbasis jaringan. PuTTY memiliki dukungan protokol jaringan SSH, TelNet, Rlogin dan SUPDUP. Protokol ini digunakan untuk menjalankan sistem operasi jarak jauh berbasis UNIX.



Gambar 3.7 Antarmuka awal Putty yang terhubung dengan server Lab ME

Kami menggunakan PuTTY sebagai klien SSH lab ME. Sistem Operasi daring digunakan untuk menyimpan data sensor yang dikirim menggunakan MQTT ke basis data Lab ME. Pemrograman program ini dibuat dalam bahasa Python.

3.3.3 Strava

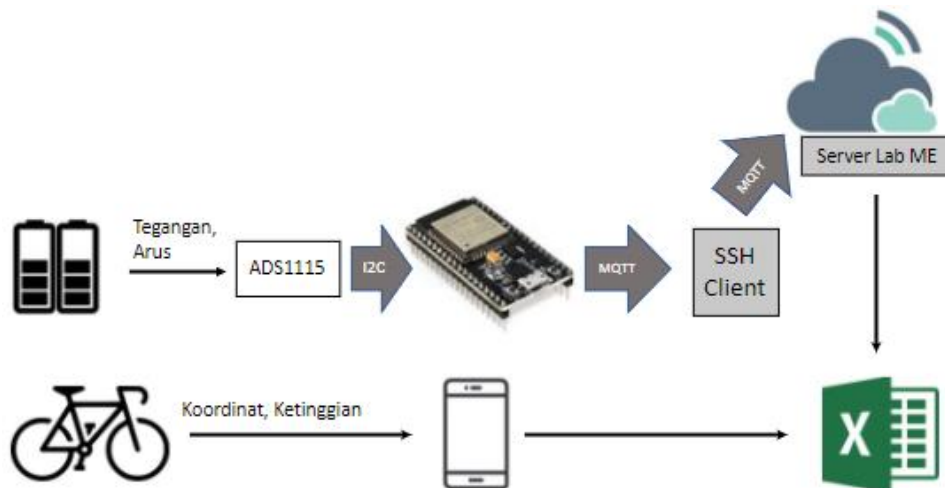
Strava merupakan aplikasi berbasis internet yang digunakan untuk menyimpan data perjalanan dari seseorang yang melakukan olahraga lari, dan bersepeda menggunakan data GPS. Pada setiap aktivitas, Strava akan menampilkan

rute perjalanan yang ditempuh, elevasi, kecepatan pergerakan (rata-rata, minimum, maksimum), dan waktu.

Pada penelitian, Strava kami gunakan untuk memperoleh data berkendara sepeda listrik ketika melalui berbagai medan berkendara. Dengan menggunakan Strava, kami tidak perlu menggunakan sensor GPS dan *Altitude* tambahan pada SMB. Keputusan penggunaan Strava ini kami ambil setelah melihat performa modul GPS Neo-6M yang sangat fluktuatif dan tidak dapat diandalkan. Modul GPS Neo-6M yang sebelumnya kami gunakan tidak dapat mengambil data pada kondisi di dalam ruangan, serta pembacaan satelit yang tidak akurat sama sekali. Dengan Strava, data yang diperoleh setelah berkendara dapat langsung diunduh dalam format .xls yang tentu saja sangat membantu.

3.4 Rancangan Sistem/Ide Solusi

Dalam merancang sistem manajemen baterai sekaligus sistem komunikasi dan pengambilan data, satu hal yang harus dipertimbangkan dengan matang adalah konsumsi daya setiap komponen yang digunakan. Hal ini menjadi sangat penting karena konsumsi daya harus dibuat seminim mungkin, sehingga penggunaan daya listrik hanya diperuntukkan bagi motor listrik pada sepeda. Dalam sistem manajemen baterai ini, akan ada beberapa proses pengambilan data, seperti data tegangan dan arus listrik pada pak baterai. Mekanisme komunikasi dengan modul server Lab ME juga harus dipertimbangkan dengan cukup baik sehingga menghasilkan transmisi data yang baik. Berikut adalah skema sistem manajemen baterai, transmisi data, dan penyimpanan data yang kami ajukan pada pengerjaan TA ini.

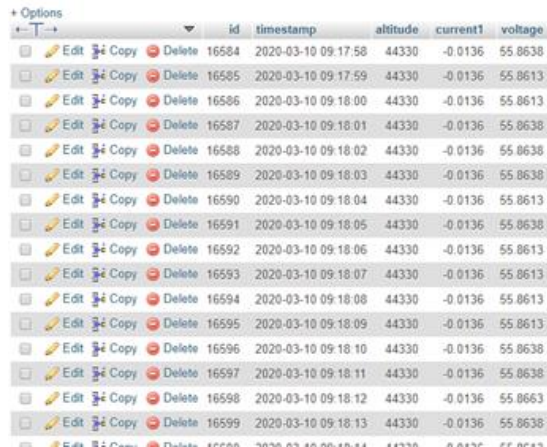


Gambar 3.8 Skema Sistem Pengambilan Data

Secara keseluruhan, terdapat tiga data utama yang harus diperoleh dari pengambilan data dengan skema di atas. Pertama, yang berasal dari pak baterai, adalah tegangan dan arus listrik. Data tegangan yang diambil merupakan tegangan terminal pak baterai, sedangkan data arus yang diambil merupakan nilai arus ketika melewati *shunt resistor*, yang nantinya harus dikonversi ke nilai sebenarnya. Data tegangan dan arus ini akan dibaca oleh sensor ADS1115, yang merupakan suatu *analog-to-digital converter*. ADS1115 ini terhubung dengan ESP32, yang akan melakukan pembacaan data dengan frekuensi 1 Hz, dilanjutkan dengan konversi data ke variabel JSON. Untuk melakukan pengiriman data, diperlukan Wi-Fi dan broker MQTT. Wi-Fi diperoleh dari *portable hotspot smartphone*, dan broker MQTT yang digunakan, yaitu lokasi pengiriman data, merupakan server Lab ME dalam *software* phpMyAdmin. Setelah konfigurasi Wi-Fi dan *broker* MQTT berhasil, data berformat JSON dikirim dalam protokol MQTT. Sebelum masuk ke server Lab ME, data akan ‘ditangkap’ oleh *SSH Client*, yang akan melakukan penyesuaian variabel dengan lokasi penyimpanan data pada server. Penyesuaian ini dilakukan agar satu *datapoints* dapat masuk ke kolom tabel yang bersesuaian pada *database*. Data yang dikirim dari *SSH Client* umumnya disebut sebagai *payloads*. *SSH Client* juga akan menambah satu jenis data, yaitu waktu pengiriman dalam satuan detik, menit, dan jam sebagai variabel *datetime*.

Setelah berhasil dikirim, server akan menerima data dan memasukkannya ke kolom *database* yang sesuai. Terdapat tiga data utama pada *database* yang akan

digunakan, yaitu tegangan, arus, dan *datetime*. Tampilan *database* Lab ME dalam *software* phpMyAdmin adalah sebagai berikut



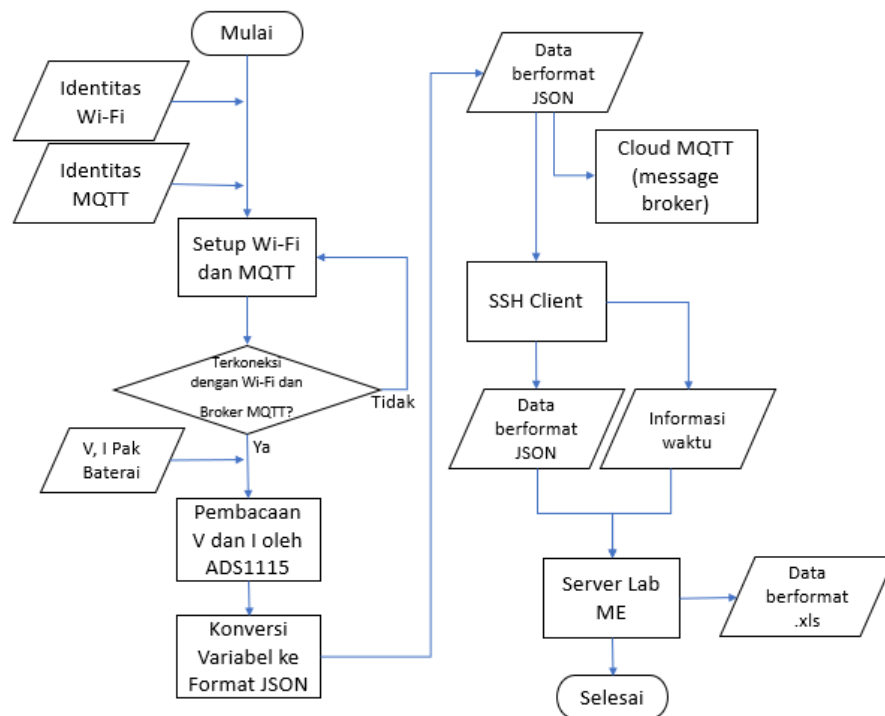
	id	timestamp	altitude	current1	voltage
	16584	2020-03-10 09:17:58	44330	-0.0136	55.8638
	16585	2020-03-10 09:17:59	44330	-0.0136	55.8613
	16586	2020-03-10 09:18:00	44330	-0.0136	55.8613
	16587	2020-03-10 09:18:01	44330	-0.0136	55.8638
	16588	2020-03-10 09:18:02	44330	-0.0136	55.8638
	16589	2020-03-10 09:18:03	44330	-0.0136	55.8638
	16590	2020-03-10 09:18:04	44330	-0.0136	55.8613
	16591	2020-03-10 09:18:05	44330	-0.0136	55.8638
	16592	2020-03-10 09:18:06	44330	-0.0136	55.8613
	16593	2020-03-10 09:18:07	44330	-0.0136	55.8613
	16594	2020-03-10 09:18:08	44330	-0.0136	55.8613
	16595	2020-03-10 09:18:09	44330	-0.0136	55.8613
	16596	2020-03-10 09:18:10	44330	-0.0136	55.8638
	16597	2020-03-10 09:18:11	44330	-0.0136	55.8638
	16598	2020-03-10 09:18:12	44330	-0.0136	55.8663
	16599	2020-03-10 09:18:13	44330	-0.0136	55.8638

Gambar 3.9 Tampilan *Database* Server Lab ME

Tabel ini dapat diunduh dalam format *.xls*, yaitu format *file* pada Microsoft Excel, yang selanjutnya akan digunakan dalam pengolahan data.

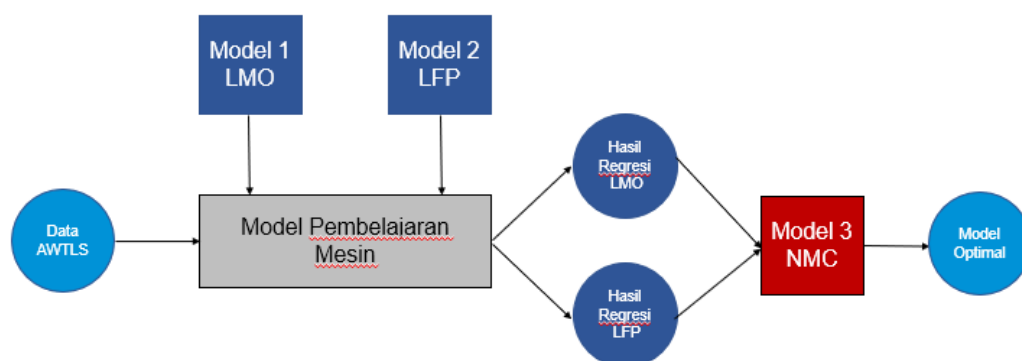
Selanjutnya terdapat data yang berasal dari suatu aplikasi pada *smartphone* yang akan menerima informasi ketinggian, *latitude*, dan *longitude*. Pengambilan data ini dilakukan dengan menggunakan aplikasi bernama Strava. Strava Dapat membaca ketiga variabel tersebut dengan periode *sampling* sekitar 10s. Data dari Strava dapat langsung diunduh dalam format *.xls* yang bisa digabung secara langsung dengan data dari *database* server Lab ME.

Berikut adalah *flowchart* algoritma pengambilan data latih. *Flowchart* ini dibuat berdasarkan algoritma pada *software* Arduino IDE dan PuTTY.



Gambar 3.10 Diagram Alir Algoritma Pengambilan Data Uji

Pada penelitian ini, akan digunakan dua sumber data, yaitu data hasil eksperimen yang akan diestimasi beberapa siklus pengosongan awalnya menggunakan metode AWTLS, dan data degradasi baterai LFP dan LMO yang berasal dari *paper*. Kedua data ini akan menjadi data masukan model pembelajaran mesin. Model hasil dari pembelajaran mesin ini akan dibandingkan dengan model degradasi baterai NMC yang juga berasal dari *paper*.



Gambar 3.11 Ide Solusi Pemodelan Degradasi Baterai

3.5 Skenario Data Eksperimen

Pada penelitian ini kami mendefinisikan tiga skenario dengan variasi rute-*assist level*. Ketiga skenario ini dapat dilihat pada tabel di bawah.

Tabel 3.2 Variasi Tiap Skenario Pengujian

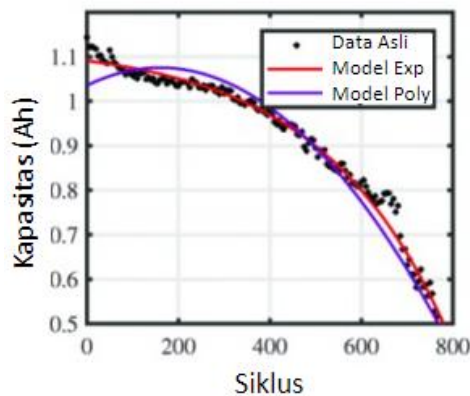
	Rute	Elevasi Rute	Level Assist
Skenario 1	Kampus ITB	Datar	5
Skenario 2	Kampus ITB	Datar	3
Skenario 3	ITB-Bukit Dago	Menanjak	5

3.6 Data Latih Tambahan

Penelitian ini menggunakan data degradasi dari tiga referensi berbeda sebagai data latih tambahan. Pada referensi pertama merupakan data degradasi dari baterai LMO dengan dataset CALCE Battery Research Group yang data kapasitas baterai setiap siklus. Referensi ini memberikan persamaan model degradasi baterai dengan menggunakan persamaan polinomial dan eksponensial. Persamaan eksponensial diketahui lebih mewakili model pada referensi dibandingkan dengan polinomial dengan $R^2 = 0,9925$ [18]. Model Persamaan dari referensi ini adalah sebagai berikut.

$$Q = -3,78 \times 10^{-2} \times \exp(3,61 \times 10^{-3} \times k) + 1,13 \times \exp(-2.99 \times 10^{-8} \times k) \quad (3.1)$$

Dengan k adalah nomor siklus.



Gambar 3.12 Data Kapasitas dan Regresi Degradasi pada Baterai LFP [18].

Data referensi kedua, merupakan data kapasitas baterai LFP yang di-*discharge* pada 1C dan 2C. Data diambil pada siklus 1, 50, 100 dan kelipatannya.

Tabel 3.3 Data Kapasitas Baterai LMO pada *Discharge Rate* Berbeda.[19]

Siklus	Kapasitas (Ah)	
	1C	2C
1	803,72	808,86
50	780,00	732,95
100	775,33	700,70
150	759,58	670,48
200	740,72	648,61
250	727,58	625,66
300	707,58	600,72

Untuk mendapatkan nilai kapasitas pada setiap siklus, kami lakukan regresi polinomial dengan persamaan dasar sebagai berikut.

$$Q = p_1 k^2 + p_2 k + p_3 \quad (3.2)$$

Dengan k adalah nomor siklus.

Persamaan yang dihasilkan adalah sebagai berikut.

Tabel 3.4 Konstanta-konstanta Hasil Regresi Baterai LMO

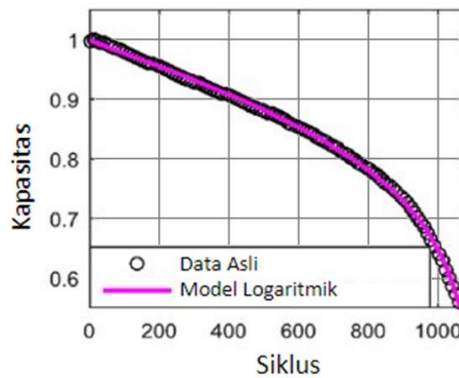
<i>Discharge Rate</i>	p ₁	p ₂	p ₃	R ²
1C	5 x 10 ⁻⁵	-0,3096	801,75	0,9796
2C	5 x 10 ⁻³	-1,099	799,14	0,9834

Berdasarkan ketiga persamaan di atas, dibuat ulang data kapasitas baterai tiap siklus dari masing-masing model. Data ini lalu di normalisasi sehingga nilai maksimal selalu di 100%.

Data referensi baterai ini akan digunakan sebagai *feature* dalam proses pembelajaran mesin untuk dapat menghasilkan prediksi degradasi kapasitas baterai pada beberapa siklus yang terjadi di masa depan hingga nilai kapasitasnya menjadi 80% dari nilai awal. Pada penelitian ini kami menggunakan satu siklus berkendara

untuk tiga skenario situasi. Berikutnya, akan diestimasi nilai penurunan kapasitas dalam beberapa *cycle* pertama berdasarkan data satu siklus berkendara yang didapatkan secara *real-time*.

Data degradasi ketiga merupakan data degradasi baterai Li-ion dengan material katoda berupa Li(NiMnCo)O₂, atau biasa disebut sebagai NMC. Data degradasi NMC ini digunakan untuk memvalidasi model degradasi yang didapat dari baterai LMO dan LFP sebelumnya. Hal ini dilakukan karena NMC juga merupakan material katoda dari baterai yang berada pada pak baterai sepeda listrik, sehingga validasi yang dilakukan adalah untuk menguji kesamaan antara estimasi degradasi pak baterai NMC dengan degradasi baterai NMC pada kondisi Lab. Metrik yang akan digunakan adalah *Root Mean Square Error* (RMSE), *Mean Absolute Error* (MAE), dan R^2 , dan kami akan berusaha untuk mengoptimalkan ketiga metrik prediksi ini.



Gambar 3.13 Kurva Degradasi Baterai Li-Ion NMC [20]

Pada referensi ini, digunakan model persamaan ‘*two-term logarithmic*’. Hal ini dilakukan setelah melihat bentuk dari kurva degradasi NMC yang memiliki dua fase penurunan. Fase pertama yaitu ketika penurunan kapasitas berlangsung dengan nilai yang konstan pada hampir sebagian besar waktu degradasi. Kemudian nilai kapasitas akan turun dengan cepat pada siklus-siklus terakhir. Karena hal ini, penulis *paper* terinspirasi dengan bentuk kurva logaritma natural yang memiliki bentuk konkaf dan memiliki dua fase perubahan nilai. Persamaan ini dituliskan sebagai berikut.

$$C_k = \alpha_1 + \alpha_2 \cdot \ln(k + m) + \alpha_3 \cdot \ln(1 - \alpha_4 \cdot k), \quad (3.3)$$

Pada persamaan di atas, k merupakan jumlah siklus, C_k merupakan kapasitas baterai pada siklus k , α merupakan parameter model yang harus diestimasi, dan m merupakan suatu nilai konstan yang harus dinyatakan di awal. Pada penelitian ini, dihasilkan nilai m yang paling baik adalah 200, dengan nilai α cukup bervariasi.

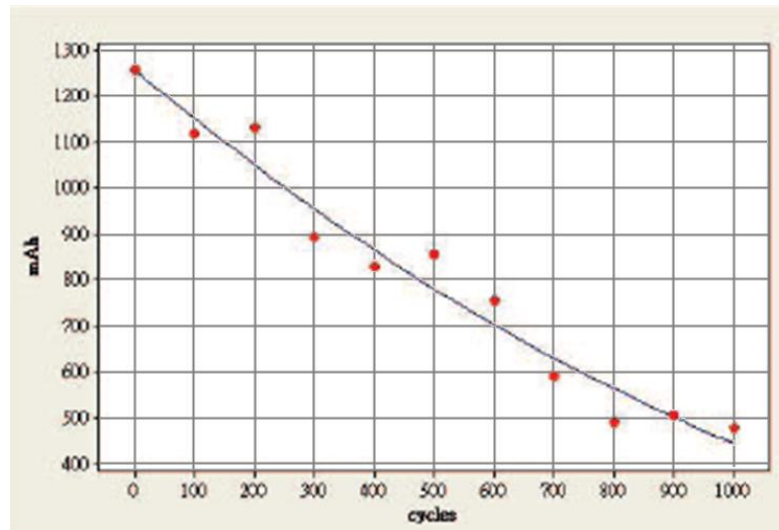
Tabel 3.5 Parameter Model Degradasi Baterai NMC

Battery ID	α_1	α_2	α_3	α_4	RMSE	R_{adj}^2
Cell #01	1.1978	-0.0365	0.1309	$8.93e-4$	0.002	0.9997
Cell #02	1.1646	-0.0295	0.2115	$7.42e-4$	0.0027	0.9993
Cell #03	1.1214	-0.0232	0.1192	$8.65e-4$	0.0053	0.9968
Cell #04	1.2217	-0.0405	0.139	$6.25e-4$	0.0025	0.9993

Data di atas merupakan data dari percobaan 4 sel baterai NMC, di mana kami pada akhirnya menggunakan nilai pada sel 1 karena memiliki performa yang paling baik.

3.7 Time Series Forecasting

Dalam pengerjaan tugas akhir ini, data yang akan kami analisis merupakan degradasi baterai Lithium-ion yang menjadi sumber energi bagi sepeda listrik. Pada penelitian ini, kami memiliki tiga skenario berkendara yang masing-masing skenario memiliki karakteristik arus *discharge*. Setelah kami memperoleh grafik penurunan kapasitas baterai dengan metode AWTLS yang telah dijelaskan sebelumnya, kami akan melakukan prediksi terhadap degradasi yang terjadi di masa mendatang dengan model berupa grafik degradasi yang memiliki karakteristik konsumsi arus yang hampir sama dengan karakteristik konsumsi arus dari tiap skenario.



Gambar 3.14 Contoh Grafik Degradasi Baterai Li-ion [21]

Pada gambar di atas, terlihat kurva degradasi kapasitas baterai secara umum. Terlihat bahwa penurunan kapasitas terjadi secara linear terhadap siklus *charging discharging*. Untuk dapat melakukan prediksi, data degradasi kapasitas yang kami miliki, dengan *datapoints* sebanyak 1/10 dari data degradasi, akan diperlakukan sebagai data *time series*. Alasan dibalik perlakuan data sebagai *time series* adalah karena data degradasi kapasitas merupakan data yang bersifat *univariate*. Hanya ada satu variabel yang mempengaruhi nilai degradasi, yaitu siklus *discharging*. Pengolahan data *univariate* dalam *machine learning* tidak dapat dilakukan dengan mudah. Hal ini dikarenakan dalam data set, kami hanya memiliki *label*, yaitu kapasitas baterai. Ketiadaan *feature* dalam data set membuat proses pembelajaran mesin tidak dilakukan dengan optimal, dan hasil prediksi pun menjadi buruk. Dengan memperlakukan data degradasi kapasitas sebagai *time series*, kami dapat melakukan *feature engineering* sehingga data set kami memiliki sejumlah variabel yang akan digunakan untuk menghasilkan hasil prediksi yang lebih baik.

Satu hal yang perlu diperhatikan adalah, karena data degradasi yang dijadikan model berbentuk linear seperti gambar di atas, maka akan digunakan salah satu model *time series forecasting*, yaitu *Linear Trend Forecasting*. *Linear Trend Forecasting* akan menghasilkan kurva degradasi data *training* yang memiliki kecocokan dengan kurva model. Karena akan data akan diperlakukan sebagai *time series*, maka perlu dilakukan beberapa penyesuaian terhadap fitur yang akan dimasukkan ke dalam *machine learning*, seperti yang telah dijelaskan sebelumnya.

Dalam melakukan *Time Series Forecasting*, terdapat beberapa kriteria yang harus terpenuhi. Pertama, data yang diolah harus bersifat stasioner, dengan kata lain nilai *mean* dan *variance* dari data harus konstan [22]. Lalu, data harus memiliki suatu tren tertentu, misalnya data memiliki tren yang deterministik ataupun stokastik. Dalam kasus degradasi kapasitas baterai, kita dapat langsung mengetahui bahwa data memiliki tren yang deterministik, dengan perubahan nilai kapasitas dapat dinyatakan sebagai persamaan garis lurus yang memiliki nilai awal 100% (di mana baterai belum terdegradasi) dan gradien garis yang bernilai negatif. Karena memiliki bentuk seperti ini, dapat disimpulkan juga bahwa degradasi kapasitas tidak bersifat stasioner, karena memiliki nilai *mean* yang berubah seiring bertambahnya siklus baterai.

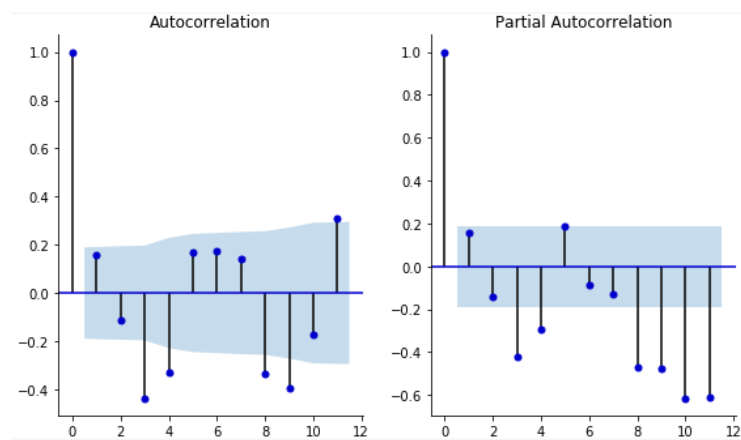
Data yang bersifat tidak stasioner ini perlu diubah dengan mencari selisih nilai antara satu *datapoint* dengan *datapoint* sebelumnya. Metode ini dinamakan *differencing*, yang akan menghasilkan data stasioner dengan nilai *mean* dan *variance* yang konstan [23]. *Differencing* yang dilakukan dengan mencari selisih antara satu periode *datapoint* dinamakan *differencing* dengan *lag* 1. Ada kemungkinan bahwa *lag* 1 belum menghasilkan data yang bersifat stasioner, sehingga kita harus mencari *lag* dengan orde yang lebih tinggi. Satu metode untuk mencari orde yang tepat dalam melakukan *differencing* adalah *Augmented Dickey-Fuller (ADF) Test* .

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \delta_2 \Delta y_{t-2} + \dots \quad (3.4)$$

Pada dasarnya *ADF Test* berfungsi untuk menguji *null hypothesis* (H0) bahwa *dataset* yang kami miliki bersifat stasioner. Dalam *ADF Test*, H0 diterima apabila *p-value* dari hasil pengujian bernilai kurang dari 5%. *ADF* juga menentukan secara otomatis berapa orde *lag* yang harus digunakan sehingga data dapat dikatakan stasioner.

Sampai sekarang, *dataset* degradasi kapasitas yang sebelumnya hanya terdiri atas kolom nilai kapasitas yang menurun, akan mengalami penambahan *feature* yaitu kolom *lag* yang berjumlah sebanyak orde yang direkomendasikan oleh *ADF Test*. Sebenarnya terdapat cara lain untuk melihat korelasi antara setiap orde

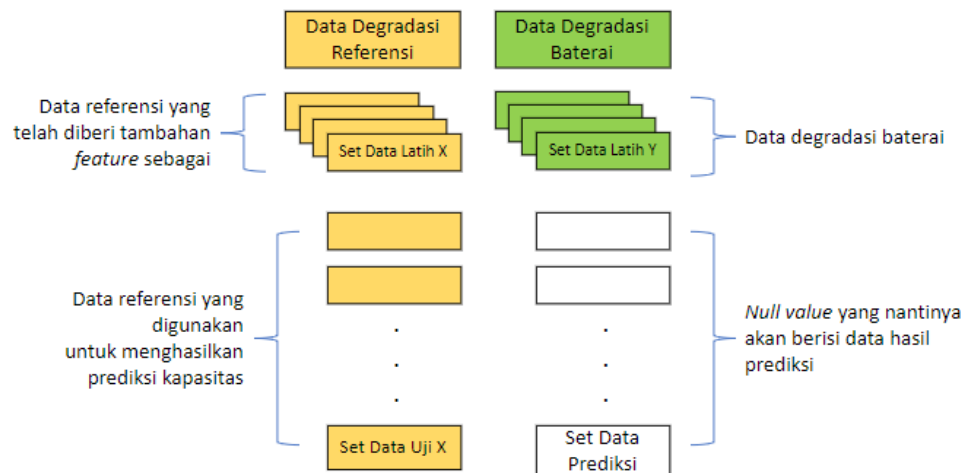
lag dengan degradasi kapasitas, cara ini adalah dengan menggunakan *Autocorrelation Function* (ACF) dan *Partial Autocorrelation Function* (PACF). Kedua cara ini akan menghasilkan grafik yang memperlihatkan orde-orde tertentu dengan korelasi yang cukup tinggi sehingga berpotensi untuk menghasilkan prediksi yang cukup baik tanpa mengakibatkan *dataset* mengalami *overfitting* atau *underfitting* [22]. Namun, karena baik ADF, ACF, dan PACF sama-sama beroperasi pada proses *autoregressive*, dan penggunaan ADF dapat langsung memberikan nilai orde *lag* secara eksak kepada pengguna, maka kami memilih untuk menggunakan *ADF Test* sebagai metode penentuan orde *lag*.



Gambar 3.15 Metode ACF dan PACF dalam Menentukan Orde *Lag*

3.7.1 Metode Pelatihan Model Pembelajaran Mesin

Pada penelitian ini, kami menggunakan data latih dari dua sumber berbeda yaitu data degradasi baterai riil, dan data degradasi referensi. Data degradasi baterai riil memiliki data degradasi 50 dan/atau 80 siklus awal, sedangkan referensi berisi data degradasi hingga 80% kapasitas baterai. Kedua data dilatih secara simultan. Data referensi akan menghasilkan data tes yang akan digunakan untuk memprediksi (*forecast*) data degradasi setelah 50 dan/atau 80 siklus.



Gambar 3.16 Pemisahan Data Latih dan Data Uji pada Algoritma *Machine Learning*

3.7.2 GridSearchCV

Salah satu metode yang kami gunakan untuk melakukan optimasi pada model *machine learning* yang dipilih adalah menggunakan *GridSearchCV*. Pada *support vector machine*, kita dapat mengatur angka pada *hyperparameter* C dan γ .

Apabila dilakukan secara manual hal tersebut akan memakan waktu yang cukup banyak, karena yang akan ditentukan tidak hanya C dan γ , melainkan juga kernel yang digunakan, dan beberapa model *machine learning* lain yang dijadikan sebagai pembandingan.

Dalam pengaplikasiannya, kami akan menggunakan *library GridSearchCV*, yang bekerja berdasarkan prinsip *Cross-Validation*. Ketika menentukan model dan beberapa *hyperparameter* yang akan dioptimasi, akan dimasukkan beberapa nilai *hyperparameter* yang memiliki kemungkinan menghasilkan skor lebih baik. *GridSearchCV* akan melakukan komputasi untuk setiap kombinasi *hyperparameter* yang telah dideklarasikan. Proses ini terkadang dapat berlangsung dengan sangat lama dan membutuhkan kemampuan komputasi yang tinggi. Oleh karena itu, *library* lain yang biasanya dipilih selain *GridSearchCV* adalah *RandomizedSearchCV*. *RandomizedSearchCV* tetap bekerja dengan prinsip *Cross-Validation*, namun *library* ini tidak menguji seluruh kombinasi yang ada, sehingga beban komputasional dapat dikurangi.

BAB IV

ANALISIS DAN HASIL

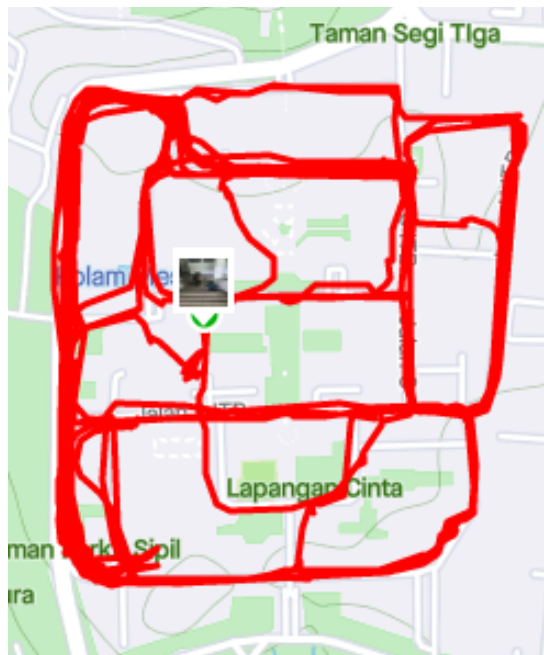
4.1 Data Eksperimen

Data eksperimen diambil dengan menggunakan sepeda listrik secara langsung pada rute yang ditentukan. Sepeda listrik digunakan secara nonstop mulai pada keadaan muatan 100% hingga baterai sepeda listrik habis (sistem tidak dapat dinyalakan lagi). Selain itu pengambilan data juga dilakukan dengan pengaturan sepeda listrik yang berbeda yaitu *assist level 3* dan *assist level 5*.

Sebelum dilakukan sesi pengambilan data, dilakukan pengecekan pada sepeda listrik. Pemeriksaan ini meliputi pemeriksaan sambungan listrik, pemeriksaan perangkat pengambil data serta pemeriksaan sistem IoT. Pemeriksaan ini memastikan pada pengambilan data berlangsung dengan hambatan yang sedikit.

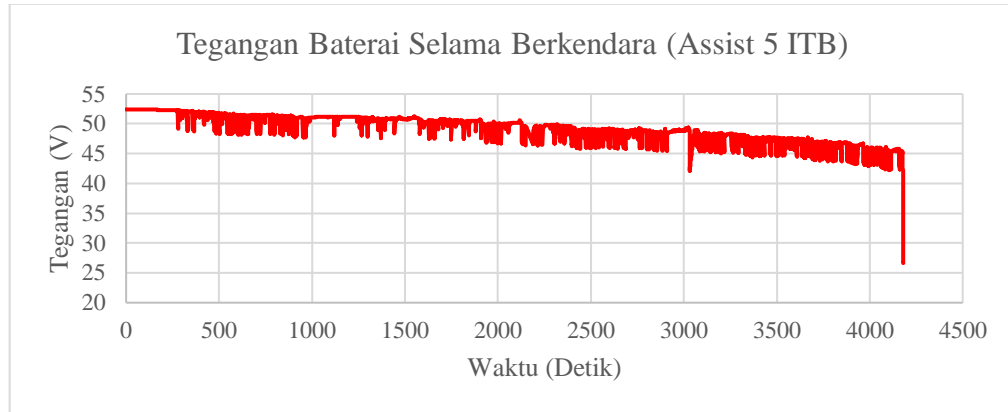
4.1.1 Rute ITB Kampus Ganesha

Pada rute ini diambil dalam dua sesi berbeda yaitu sesi dengan *assist level 5* dan *3*. Rute ini mencakup hampir seluruh jalan di dalam kampus ITB Ganesha secara acak. Rute ini memiliki perubahan ketinggian maksimal 19 meter.

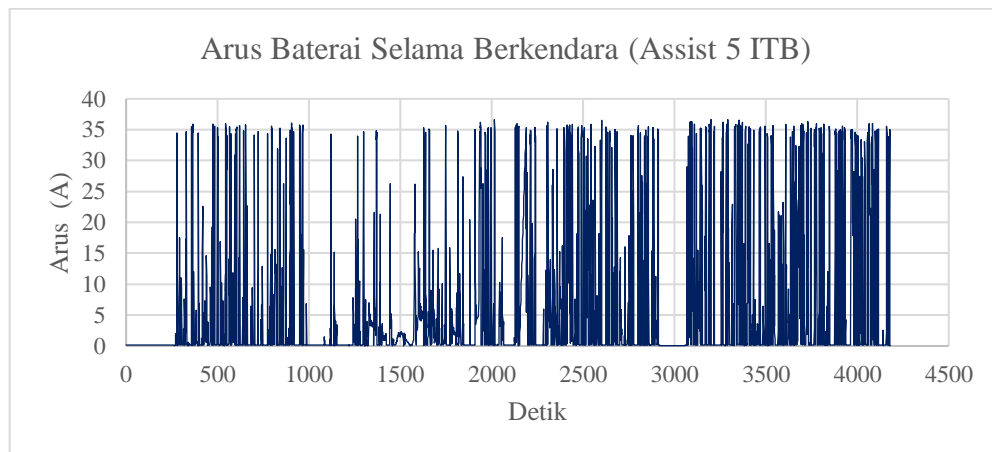


Gambar 4.1 Peta rute dalam kampus ITB Ganesha

Pada sesi pengambilan data dengan *assist level 5* yang dilakukan di dalam lingkungan ITB kampus Ganesha yang berlangsung selama 1 jam 15 menit 21 detik menempuh jarak 21,95 km, didapat data tegangan dan Arus sebagai Berikut.



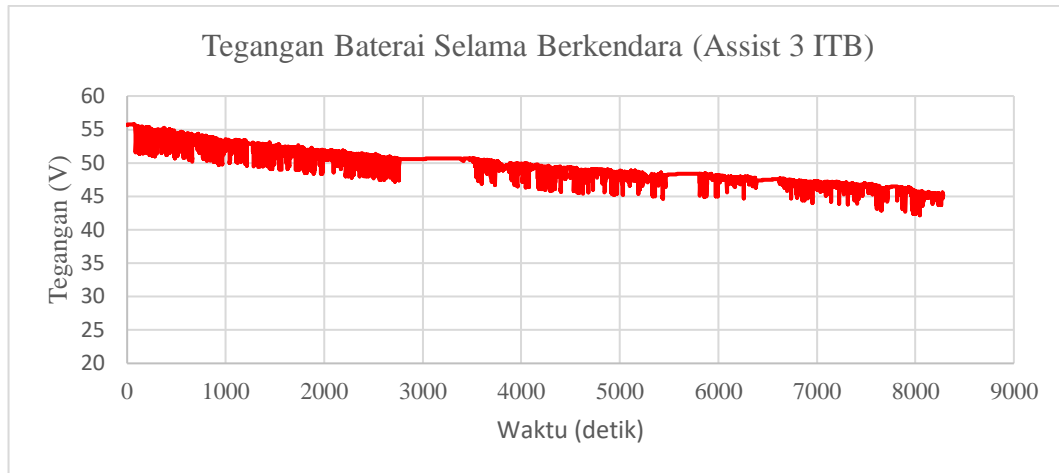
Gambar 4.2 Tegangan Baterai terhadap waktu rute dalam kampus ITB (Assist 5)



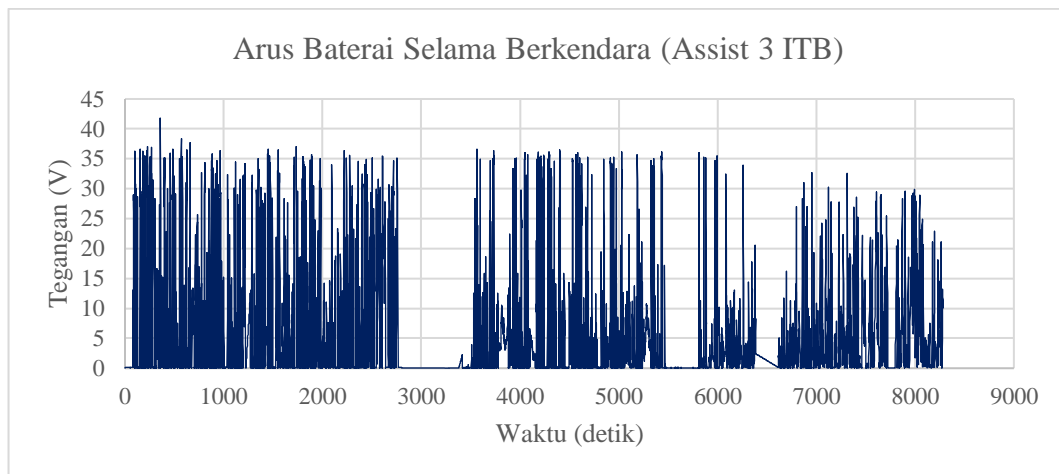
Gambar 4.3 Arus Baterai terhadap waktu untuk rute dalam kampus ITB (assist 5)

Pada skenario ini seperti yang terlihat pada **Gambar 4.2**, baterai memiliki tegangan 52,4 V pada saat keadaan baterai penuh dan berkurang menjadi 26,6 V pada saat baterai telah habis. Sedangkan terlihat pada **Gambar 4.3** Arus terlihat naik turun dengan curam, hal ini dikarenakan sinyal pengontrolan arus motor ini memiliki jenis sinyal impuls.

Sedangkan pada sesi pengambilan data dengan assist level 3 yang memakan waktu 2 jam 5 menit 59 detik dengan jarak 42,12 Km, memberikan data sebagai berikut.



Gambar 4.4 Tegangan Baterai terhadap waktu rute dalam kampus ITB (Assist 3)



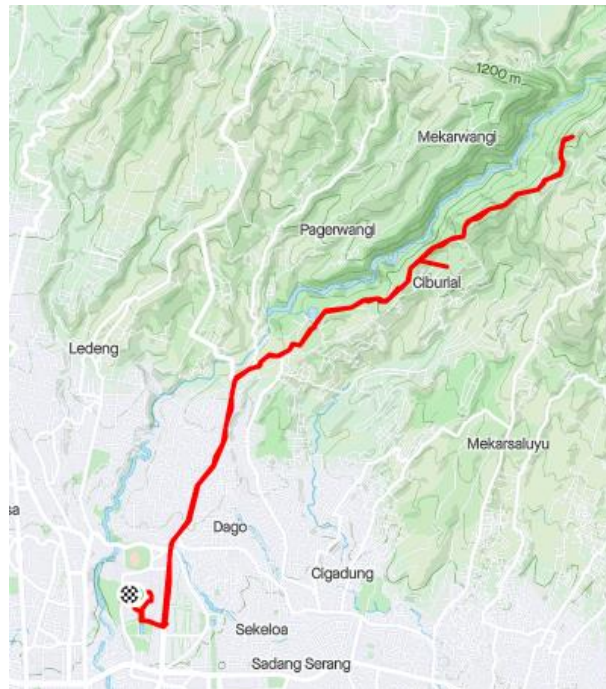
Gambar 4.5 Arus Baterai terhadap waktu rute dalam kampus ITB (Assist 3)

Gambar 4.4 memperlihatkan perubahan tegangan pada saat pengosongan baterai jika sepeda listrik dijalankan pada setelan *Assist 3*. Tegangan pada sesi ini 55.7 V pada saat baterai penuh dan 42.2 V pada saat baterai habis. Sedangkan **Gambar 4.5** memperlihatkan perubahan arus baterai. Lonjakan yang curam pada grafik ini dikarenakan oleh sinyal impuls yang digunakan pengontrol untuk mengontrol arus sepeda.

4.1.2 Rute ITB-Bukit Dago

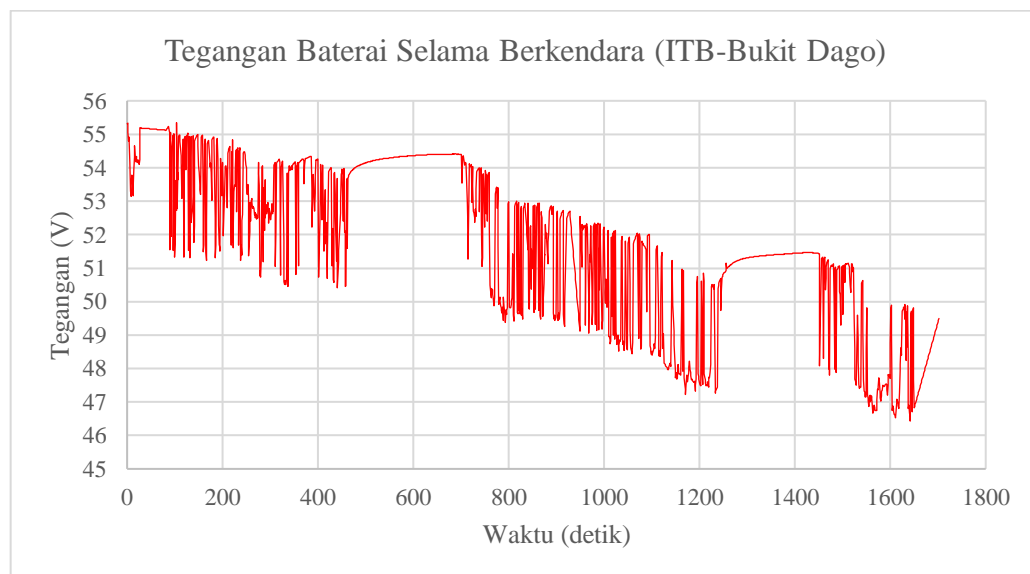
Pada pilihan rute ini, sepeda listrik diset dengan *assist level 5*. Pengambilan data dilakukan dengan menyusuri Jalan Ir. H. Djuanda mendaki Bukit Dago hingga

baterai habis. Peta rute yang dilalui pada sesi pengujian ini adalah seperti pada **Gambar 4.6** di bawah ini.

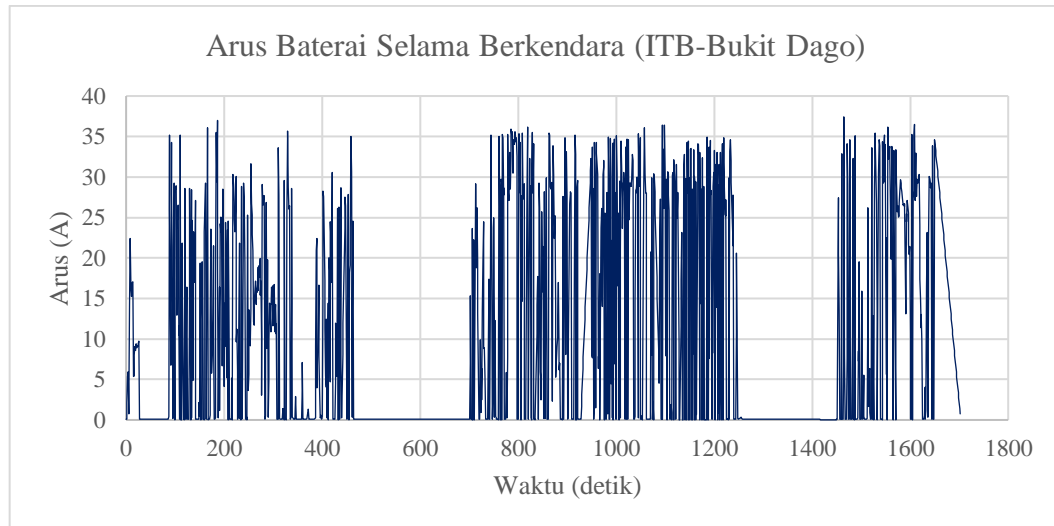


Gambar 4.6 Rute ITB-Bukit Dago

Pengambilan data pada rute ini menghabiskan waktu 28 menit 22 detik dengan jarak tempuh 8,4 Km. Rute ini memiliki perubahan elevasi sebesar 494,2 meter. Hasil dari pengambilan data ini adalah sebagai berikut.



Gambar 4.7 Tegangan Baterai terhadap waktu rute ITB-Bukit Dago



Gambar 4.8 Arus Baterai terhadap waktu Rute ITB-Bukit Dago

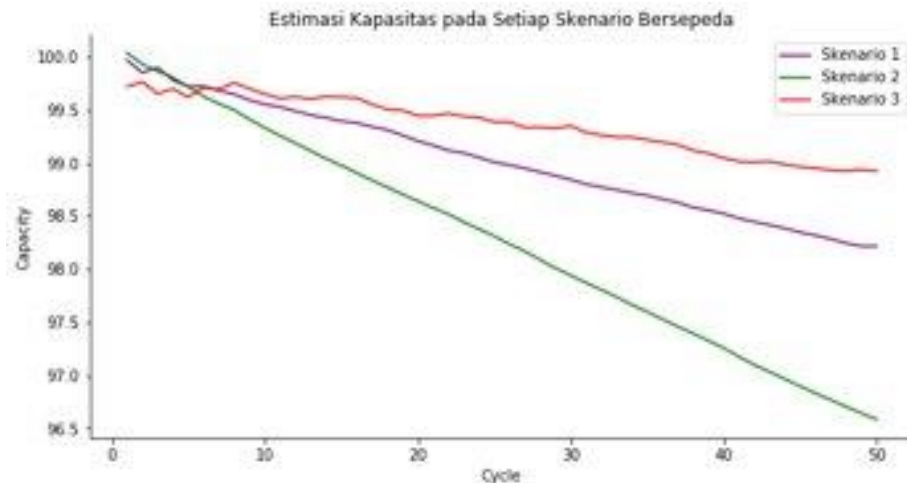
Gambar 4.7 menunjukkan perubahan tegangan baterai pada saat sepeda listrik dengan nilai tegangan maksimum 55,3 V dan minimum 46,42 V.

Gambar 4.8 Menunjukkan dinamika arus baterai pada saat digunakan dengan arus maksimal mencapai 37,4 A dan minimum 0,0136 A.

Ketiga data skenario inilah yang nanti akan diolah lebih lanjut menggunakan AWTLS untuk menghasilkan data kapasitas pada siklus awal.

4.2 Hasil Estimasi Kapasitas untuk Beberapa Siklus Awal

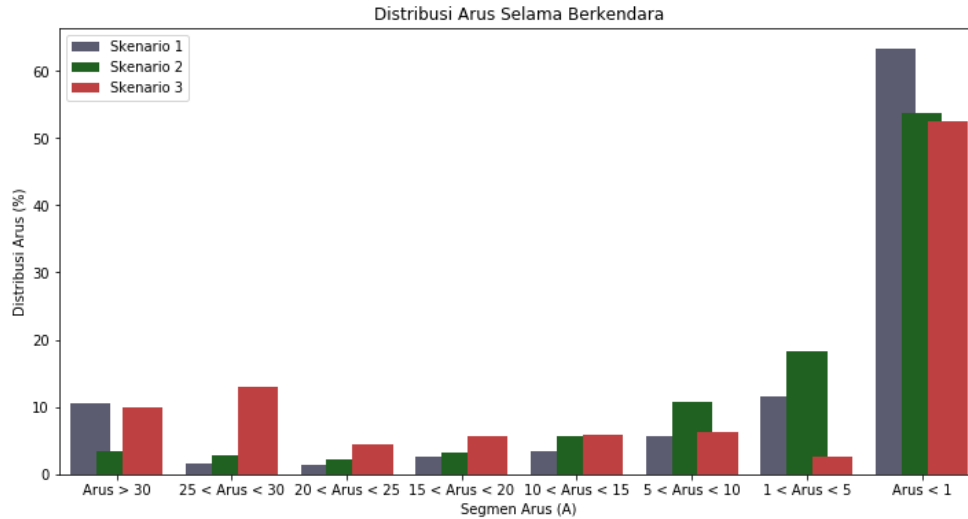
Pada Bab 3 telah diperlihatkan data berkendara dari ketiga skenario yang kami rencanakan, yaitu Skenario 1 Jalan Mendatar dengan *Assist* 5, Skenario 2 Jalan Mendatar *Assist* 3, dan Skenario 3 Jalan Menanjak *Assist* 5. Karena sangat sulit untuk bisa mengetahui penurunan degradasi kapasitas dari satu kali siklus berkendara, maka kami menggunakan AWTLS sebagai cara untuk mengestimasi kapasitas dengan menggandakan data ΔKM setiap menit dari setiap skenario sebanyak jumlah tertentu. Dalam pengolahan data, kami menggandakan data ini sebanyak 50 kali, dengan demikian kami mendapatkan data kapasitas pak baterai setelah 50 *cycle*.



Gambar 4.9 Hasil Simulasi Kapasitas Baterai dengan AWTLS pada 50 Siklus Pertama

Pada gambar di atas, dapat dilihat bahwa nilai kapasitas tidak tepat berada di 100% setelah *cycle* pertama. Selain itu, dalam 10 *cycle* pertama terlihat bahwa Skenario 1 dan Skenario 3 masih memiliki *ripple* pada penurunan kapasitas. Ketidakstabilan ini terjadi karena AWTLS belum menemukan solusi kapasitas yang tepat yang dapat meminimalisir *cost function*. Pada saat inisialisasi, nilai kapasitas di asumsikan sebesar 99% dari kapasitas awal, dan AWTLS belum memiliki *memory* untuk nilai kapasitas yang akan diestimasi.

Dengan melihat grafik penurunan degradasi di atas, terlihat bahwa Skenario 2 akan memberikan penurunan degradasi yang paling cepat. Apabila melihat distribusi arus per skenario berkendara, Skenario 2 memiliki distribusi arus yang relatif cukup tinggi pada arus rendah (1 – 10 A). Dengan operasinya yang sedikit mengonsumsi arus tinggi, membuat penggunaannya dapat lebih lama, sehingga degradasi semakin lebih besar. Pada Skenario 1, penggunaan Assist 5 pada sepeda listrik membuat distribusi pada arus tinggi (> 30 A) dan arus rendah (< 5 A) menjadi cukup tinggi. Hal ini dikarenakan pada pengendaraan sepeda, throttle maksimum telah menghasilkan kecepatan yang cukup tinggi, sehingga sepeda sudah dapat melaju cukup jauh. Sedangkan pada skenario 3, distribusi pada arus yang tinggi menjadi cukup dominan, hal ini karena pada jalanan menanjak, beban yang diterima sepeda listrik menjadi besar, mengakibatkan putaran motor menurun dan arus yang dikonsumsi meningkat.



Gambar 4.10 Distribusi Arus Selama Berkendara untuk Tiap Skenario

4.3 Dickey-Fuller Test sebagai Feature Selection

Seperti yang telah dijelaskan pada Bab 3, data yang digunakan sebagai data latih hanyalah degradasi kapasitas antara *cycle* pertama sampai *cycle* ke-*n* di mana nilai kapasitasnya telah mencapai 80%. Agar model *machine learning* dapat dilatih dengan baik, diperlukan *feature* tambahan sehingga model tidak mengalami *underfitting*. Telah dijelaskan pula sebelumnya bahwa data ini akan diperlakukan sebagai *time series*, oleh karena itu digunakan *Augmented Dickey-Fuller (ADF) test* untuk menguji apakah data degradasi yang digunakan stasioner atau tidak. Di bawah ini merupakan hasil *ADF test* dari data degradasi baterai LFP dengan arus *discharging* pada *rate* 1C.

Tabel 4.1 Perbandingan Hasil Model Regresi yang Berbeda pada *ADF Test*

Model Regresi	Critical value	Probabilitas Null Hypothesis	Jumlah 'lag' Digunakan	Jumlah Data Observasi	t-value 1%	t-value 5%	t-value 10%
Constant	-0,096	0,949	0	598	-3,441	-2,866	-2,569
Constant & Trend	$-7,75 \times 10^9$	0	5	593	-3,974	-3,417	-3,131

Pada dasarnya ADF bertujuan untuk menguji apakah data dapat digunakan untuk prediksi *time series* atau tidak dengan melihat stasioneritas-nya. Data dikatakan lulus tes ini (stasioner) apabila *critical value* lebih kecil dibanding *t-*

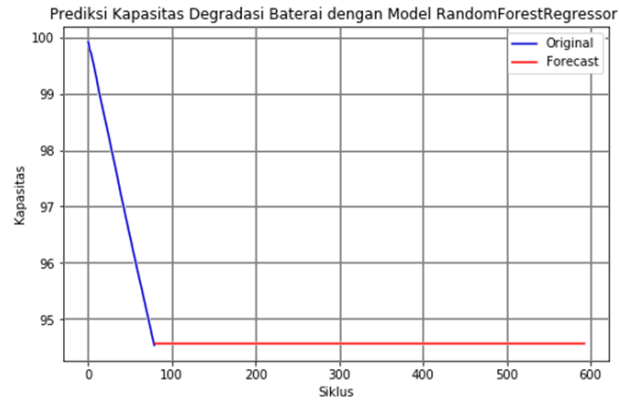
values pada *confidence interval* 1%, 5%, dan 10%. Pada tabel di atas, diperlihatkan dua model regresi, Constant dan Constant & Trend. Model regresi ini merupakan salah satu parameter yang harus dinyatakan ketika menginisiasi ADF *test*. Pada model Constant, *critical value* lebih besar dibanding t-value pada 1%, 5%, dan 5%. Terlihat pula probabilitas *null hypothesis* sebesar 94,97%, yang tentunya sesuai dengan nilai *critical value* yang lebih besar. Untuk model regresi ini, data tidak dapat diklasifikasikan sebagai data stasioner. Pada model Constant & Trend, *critical value* jauh lebih kecil dibanding seluruh t-value. Probabilitas *null hypothesis* diterima pun 0%, di mana hal ini berarti *null hypothesis* ditolak dan data dapat diklasifikasikan sebagai stasioner. Hal ini sesuai dengan bentuk dari degradasi baterai LFP, yaitu memiliki penurunan degradasi yang konstan antara tiap *cycle* dan memiliki tren yang negatif-linear.

Satu lagi hasil ADF *test* yang membantu dalam menentukan *feature* tambahan pada data latih adalah Jumlah *lag* yang Digunakan. Model regresi Constant & Trend, yang berhasil membuktikan bahwa data degradasi bersifat stasioner, menggunakan 5 *lag* dalam kalkulasi ADF *test*. Kelima *lag* ini memiliki korelasi yang tinggi terhadap prediksi data, dan akan digunakan sebagai tambahan *feature* pada data latih

4.4 Analisis Model Pembelajaran Mesin

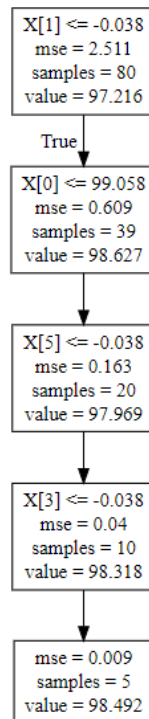
4.4.1 Random Forest Regressor, Decision Tree & XGBoost

Pada *Random Forest* (RF), model pembelajaran mesin lain yang menyokong proses *ensemble learning* merupakan *Decision Tree* (DT). Teknik *ensemble learning* sendiri terbagi atas dua metode, yaitu *boosting* dan *bootstrap aggregation (bagging)*. RF menggunakan metode *bagging*, yang artinya data uji dibagi ke dalam beberapa subset secara random dan setiap subset digunakan untuk melatih DT. RF akan mengambil nilai rata-rata dari seluruh DT yang ada, dengan demikian, hasil prediksi akan menjadi lebih akurat karena hasil prediksi dari DT yang tidak begitu sempurna akan dinetralisir oleh DT yang lain. Berikut adalah hasil prediksi RF untuk nilai degradasi kapasitas baterai pada beberapa siklus di masa depan.



Gambar 4.11 Hasil Prediksi Kapasitas Baterai dengan Model Pembelajaran Mesin *Random Forest Regressor*

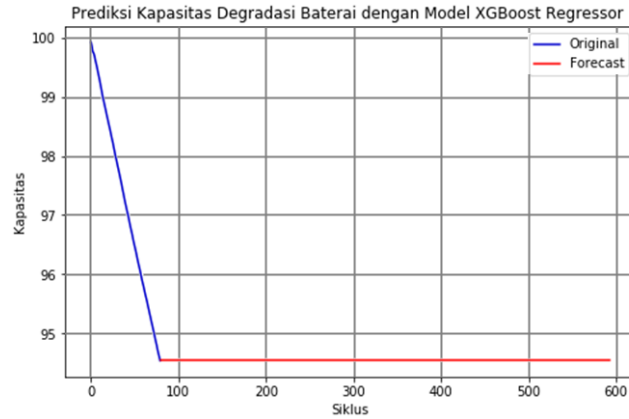
Terlihat bahwa RF tidak dapat melakukan prediksi yang baik. Untuk itu, akan dilakukan analisis terhadap model DT yang menyokong proses prediksi RF. Alasan mengapa seluruh DT pada model RF tidak dapat melakukan prediksi yang tepat adalah karena DT hanya bekerja untuk data yang nonlinear. DT bekerja dengan cara membagi dataset yang di-input ke dalam beberapa subtree (cabang). Proses percabangan ini dilakukan dengan mencari nilai rata-rata pada dataset. Nilai rata-rata seluruh dataset akan menjadi node utama dan bercabang menjadi *decision node* yang lain. Proses ini berlangsung terus menerus dengan mencari nilai rata-rata di setiap cluster percabangan. Pada kasus regresi, proses percabangan akan terjadi secara satu dimensional, yaitu pada variabel x (dalam penelitian ini, kapasitas). Karena data bersifat linear, tidak ada kategori yang dapat membuat percabangan pada DT, sehingga hanya dihasilkan prediksi berupa garis lurus dengan nilai yang konstan. Berikut ditampilkan visualisasi DT pada proses fitting model.



Gambar 4.12 Visualisasi *Decision Tree* pada Prediksi Kapasitas Degradasi

Pada visualisasi DT di atas, tidak ada percabangan yang dihasilkan, dan model hanya dapat membaca nilai rata-rata dari data degradasi estimasi hasil AWTLS. Karena satu DT gagal untuk melakukan prediksi, maka setiap DT yang berada pada Random Forest juga tidak dapat menghasilkan prediksi apapun.

Selain DT dan RF, kami juga menggunakan model Xtreme Gradient Boosting, suatu model pembelajaran mesin yang bekerja berdasarkan teknik *gradient boosting*. Teknik ini bekerja hampir sama dengan DT, namun jika *decision tree* hanya membuat satu pohon dengan kedalaman cabang yang dapat diatur, *gradient boosting* membuat beberapa pohon yang akan memperkecil error prediksi. Prinsip dalam percabangan masih sama, yaitu nilai rata-rata dari sejumlah data yang berada pada rentang nilai tertentu. Sayangnya, XGBoost juga tidak dapat mempelajari penurunan tren degradasi berdasarkan data referensi, berikut adalah hasilnya.



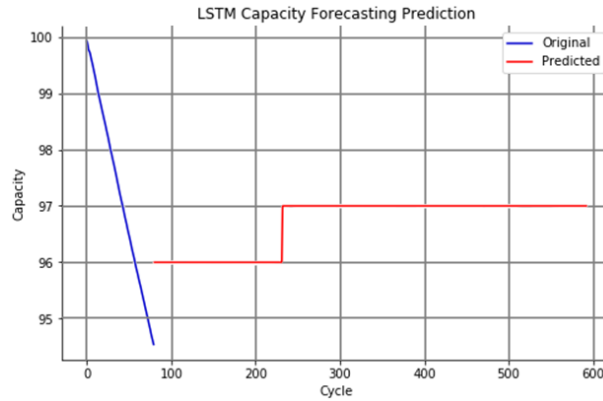
Gambar 4.13 Hasil Prediksi Kapasitas Baterai dengan Model Pembelajaran Mesin *XGBoost Regressor*

Dengan hasil XGBoost ini, kami mengambil kesimpulan bahwa penggunaan model pembelajaran mesin yang menggunakan prinsip percabangan pohon tidak akan berhasil untuk mempelajari tren penurunan degradasi yang linear.

4.4.2 *Recurrent Neural Network*

Metode kedua yang kami gunakan dan akan dianalisis kegagalannya adalah *Recurrent Neural Network* (RNN). RNN merupakan salah satu jenis neural network yang memiliki *hidden state* yaitu semacam memori sehingga prediksi yang dihasilkan oleh RNN mengacu pada informasi sebelumnya [24]. Pada tugas akhir ini, kami menggunakan *Long-Short Term Memory* (LSTM), yaitu suatu jenis khusus dari RNN yang dapat menyimpan lebih banyak memori dan menangani rantang data yang lebih luas. Sama seperti RF, LSTM akan kami gunakan untuk memprediksi nilai degradasi.

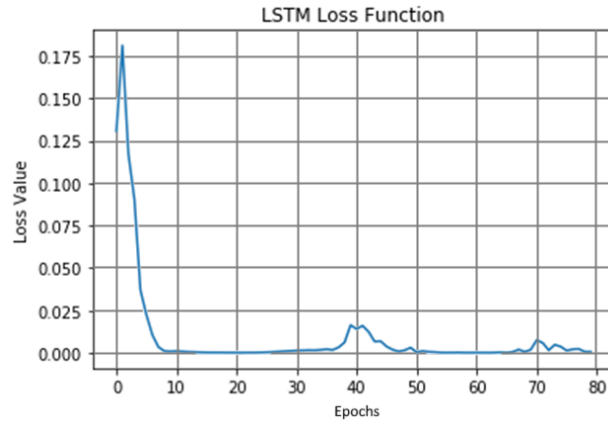
Berikut kami tampilkan hasil prediksi kapasitas baterai hingga baterai terdegradasi hingga 80% menggunakan model LSTM.



Gambar 4.14 Hasil Prediksi Kapasitas Baterai dengan Model Pembelajaran Mesin LSTM

Gambar di atas memperlihatkan hasil prediksi yang tidak merepresentasikan penurunan degradasi kapasitas baterai. Alih-alih menghasilkan kurva penurunan yang linear dan memiliki tren menurun, kurva yang dihasilkan memiliki bentuk sigmoid, yang sangat tidak diharapkan karena apa yang kami coba lakukan adalah prediksi, bukan klasifikasi. Hal ini dapat terjadi karena LSTM merupakan metode yang pada umumnya digunakan untuk melakukan prediksi pada data yang nonlinear. Data yang kami miliki sangat sederhana, tidak memiliki keterulangan (*seasonality*) maupun, dan memiliki tren menurun yang konstan. Alasan mengapa dihasilkan bentuk seperti kurva sigmoid dapat dijelaskan dengan melihat unit aktivasi pada sel LSTM. Berikut adalah ilustrasinya,

Satu unit sel LSTM dapat terdiri atas tiga fungsi aktivasi sigmoid dan satu fungsi aktivasi tanh. Seluruh fungsi aktivasi ini biasanya digunakan pada data nonlinear, dan memiliki bentuk aktivasi seperti huruf ‘S’ yang terdapat pada hasil prediksi. Karena data terlalu sederhana, unit aktivasi ini menghasilkan *output* berupa bentuk fungsi aktivasi yang dimiliki oleh unit ini. Untuk dapat melihat apakah model LSTM telah melakukan performa dengan baik, akan ditampilkan fungsi *loss* dari model data latih.

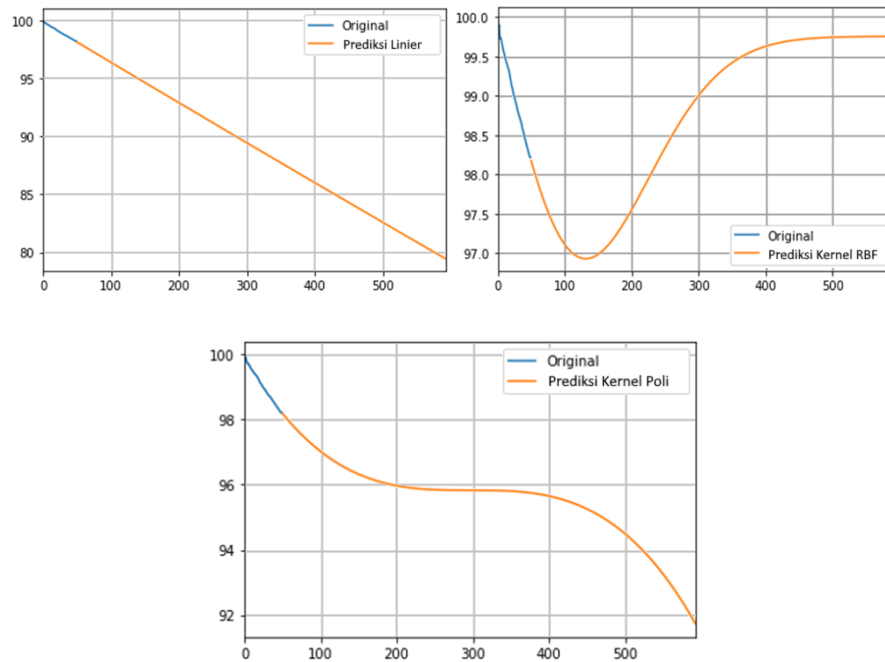


Gambar 4.15 Fungsi *Loss* LSTM

Dapat dilihat bahwa LSTM sudah melakukan performa yang cukup baik, yaitu dengan konvergen ke suatu nilai yang rendah dengan cepat, meskipun terdapat sedikit kenaikan nilai *loss* di beberapa titik data. Fungsi *loss* yang kami gunakan pada model LSTM ini adalah fungsi *mean squared error*, yang menghitung kuadrat dari perbedaan nilai antara nilai terprediksi dan nilai seharusnya.

4.5 Analisis Aplikasi berbagai Kernel Regresi Vektor Pendukung

Dalam penggunaan model *machine learning Support Vector Regression*, terdapat empat kernel yang umumnya digunakan. Keempat kernel tersebut adalah Linear, Polynomial, Rbf (Gaussian), dan Sigmoid. Pada penelitian ini kami telah mencoba untuk menggunakan keempat kernel tersebut untuk dapat memprediksi degradasi kapasitas dari pak baterai Li-ion pada sepeda listrik berdasarkan data degradasi referensi. Seperti yang terlihat pada gambar di atas, kernel Linear dapat mempelajari degradasi dengan cukup baik, dibuktikan dengan nilai *Mean Absolute Error* (MAE) yang diperoleh sebesar 0,37. Kernel polinomial terlihat berhasil mempelajari tren penurunan nilai kapasitas namun tidak berhasil mencapai nilai degradasi 80%, hal ini didukung dengan nilai MAE yang cukup besar, 7.17. Kedua kernel yang lain, rbf dan Sigmoid tidak berhasil menunjukkan tren degradasi sama sekali, didukung oleh nilai MAE untuk kedua hasil regresi kernel ini sebesar 10, 35 dan 49.85.

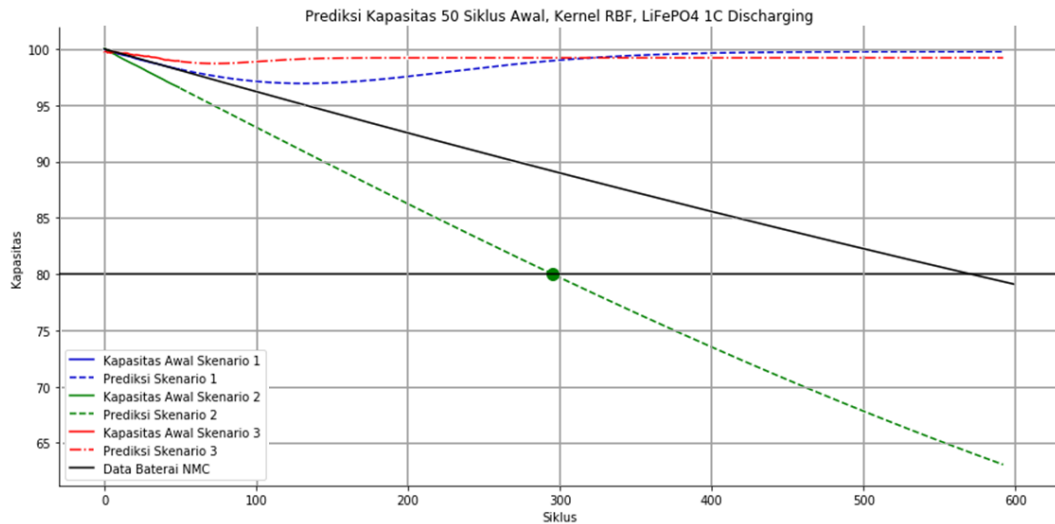


Gambar 4.16 Hasil Prediksi dengan Model RVP menggunakan kernel (a) linear (b) RBF (Gaussian) dan (c) Polinomial

Hasil ini tentu saja dipengaruhi bagaimana setiap kernel memperlakukan data agar dapat memproyeksikan data supaya dapat didekati dengan bidang linear. Penjelasan akan dimulai dari kernel sigmoid yang menghasilkan nilai MAE paling tinggi, mengindikasikan performa algoritma yang paling buruk. Kernel sigmoid pada RVP digunakan untuk mempelajari data yang memiliki *seasonality*, atau biasa diperlihatkan dengan pengulangan pola antara rentang waktu tertentu. Karena data degradasi yang diperoleh dari *paper* memiliki pola yang linear, penggunaan kernel sigmoid tidak akan mngeluarkan hasil yang baik. Penjelasan kedua adalah tentang kernel polinomial dan rbf. Kedua kernel ini umumnya digunakan pada data yang bersifat nonlinear [25], yang tentunya tidak akan berhasil apabila diaplikasikan pada degradasi baterai. Namun, terlihat bahwa polinomial masih dapat mempelajari tren penurunan, sedangkan Rbf tidak. Alasan kernel polinomial masih dapat mempelajari tren penurunan degradasi adalah karena kernel polinomial merupakan kernel global. Kernel global memungkinkan data dengan jarak yang jauh untuk mempengaruhi proses regresi pada dimensi yang lebih tinggi [26]. Selain itu, kernel polinomial yang mencari bidang linear dengan cara mengangkat datapoin dengan suatu nilai tertentu akan menghasilkan nilai regresi yang menjauh dari nilai

sebenarnya, mengakibatkan MAE yang besar. Lalu pada kernel rbf, kernel ini merupakan kernel lokal, yang sangat terpengaruh pada datapoint yang berdekatan dalam menghasilkan prediksi [26]. Pada data tes, terdapat jarak yang cukup jauh antara nilai degradasi pak baterai dan baterai referensi. Jarak yang jauh ini membuat kernel rbf tidak dapat menciptakan bidang linear yang diinginkan dan pengaruh dari nilai degradasi data latih menjadi dominan. Alhasil, hasil prediksi justru naik ke angka ~99% dan konstan di nilai tersebut.

Namun, ketidakberhasilan kernel RBF karena merupakan kernel lokal tidak dapat langsung diterima begitu saja. Setelah kami melakukan pemodelan menggunakan kernel RBF dengan menggunakan model baterai LFP 1C, didapatkan hasil yang cukup berbeda untuk setiap skenario.

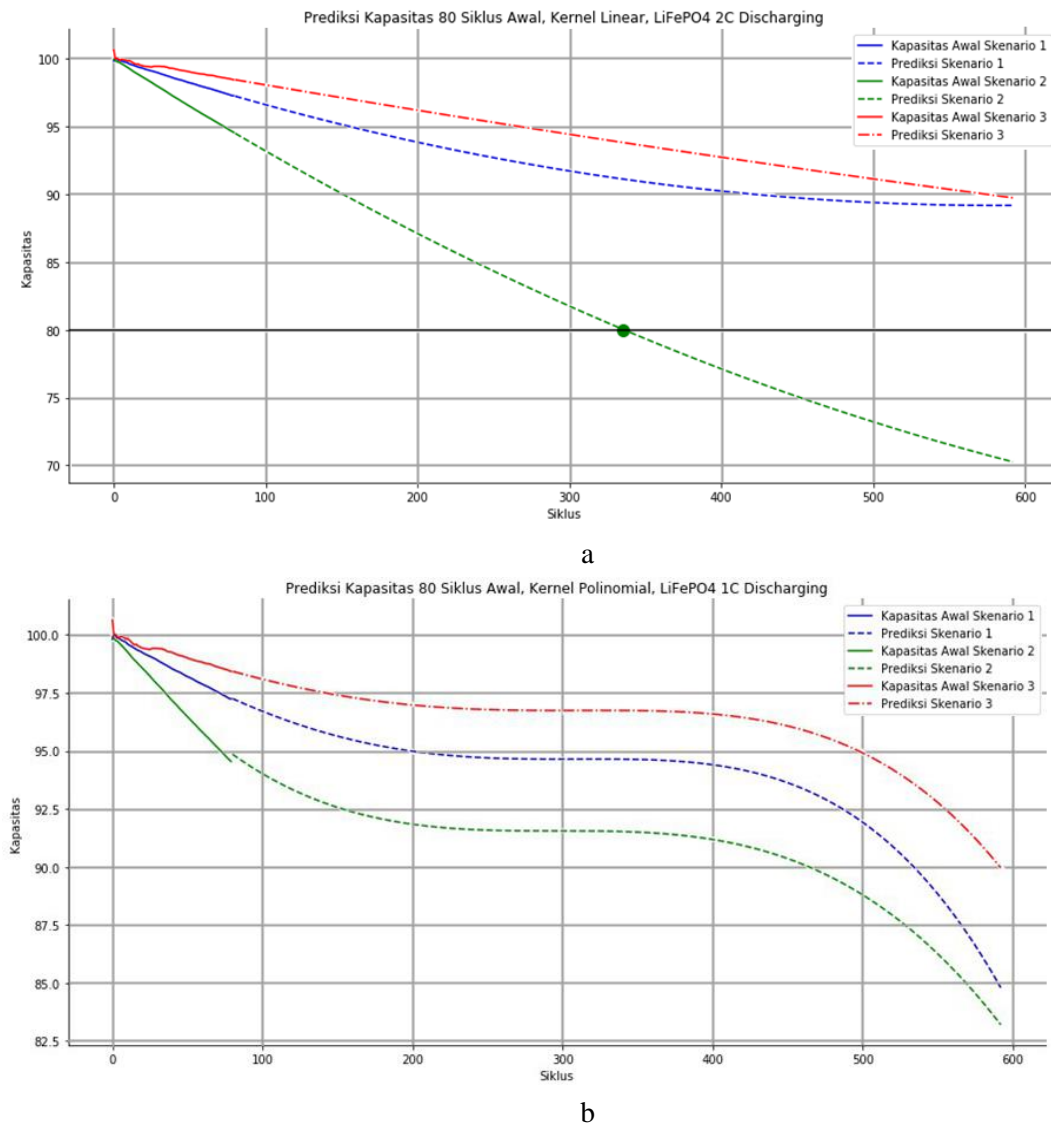


Gambar 4.17 Hasil Prediksi dengan Model RVP menggunakan kernel RBF

Dengan menggunakan data estimasi 50 siklus dari AWTLs sebagai data latih bersama dengan 50 siklus baterai LFP 1C, didapatkan bahwa hanya skenario 2 yang berhasil menunjukkan tren penurunan. Skenario 1 dan 3 tidak berhasil mempelajari tren tersebut, dan didapatkan hasil R^2 pada metrik prediksi bernilai $-58,5$ untuk skenario 1 dan $-3542,1$ untuk skenario 3. Pada skenario 2, diperoleh nilai R^2 sebesar $-1,21$. Skenario 1 dan 3 menghasilkan kurva yang konvergen ke atas disebabkan karena kedua skenario ini tidak menghasilkan penurunan degradasi yang besar dalam 50 siklus awal yang diestimasi. Akibatnya, kedua data ini memiliki bentuk kurva yang hampir landai, sehingga model RVP dengan kernel RBF tidak dapat melakukan regresi. Perlu diingat kembali bahwa RBF merupakan

kernal yang bekerja pada data nonlinear. Sedangkan untuk data skenario 2, karena telah menghasilkan degradasi yang cukup besar, maka data ini, meskipun linear, masih dapat mempelajari kurva penurunan degradasi yang dimiliki oleh model baterai LFP.

Berikut kami tampilkan beberapa visualisasi degradasi kapasitas baterai menggunakan kernel linear dan polinomial.



Gambar 4.18 Beberapa Hasil Prediksi dengan Nilai Indikator Performa yang Optimal Hasil prediksi ini lalu diseleksi dengan juga mempertimbangkan nilai RMSE, MAE dan R^2 . dengan hasil sebagai berikut.

Tabel 4.2 Hasil Indikator Performa dari Tiga Skenario Berkendara dengan Variasi pada Pemilihan Data Referensi Baterai, Kernel RVP, dan Jumlah Siklus

Data Latih Referensi	Data Eksperimen (Jumlah Siklus Awal)	Kernel	Skenario 1			Skenario 2		
			RMSE	MAE	R ²	RMSE	MAE	R ³
LFP 2C	80	Linear	1,424	1,102	0,652	7,880	7,707	-1,692
		Poly	-	-	-	2,186	1,956	-0,796
	50	Linear	1,687	1,177	0,538	6,981	6,756	-0,891
		Poly	-	-	-	3,266	2,772	-10,834
LFP 1C	80	Linear	3,101	2,821	0,593	15,059	13,792	-1,451
		Poly	4,256	3,525	-10,495	2,178	1,757	0,249
	50	Linear	2,143	1,934	0,792	14,471	13,006	-1,042
		Poly	4,718	3,862	-16,104	1,932	1,723	0,326
LMO	80	Linear	8,527	6,776	-0,092	24,909	21,042	-1,328
		Poly	3,777	2,255	0,539	12,063	7,921	-0,157
	50	Linear	7,135	5,473	0,154	17,076	14,009	-0,737
		Poly	3,238	2,088	0,583	10,327	6,552	-0,017
Data Latih Referensi		Data Eksperimen (Jumlah Siklus Awal)	Kernel	Skenario 3				
				RMSE	MAE	R ⁴		
LFP 2C		80	Linear	4,399	3,733	-9,894		
			Poly	-	-	-		
		50	Linear	4,847	4,097	-13,733		
			Poly	-	-	-		
LFP 1C		80	Linear	2,019	1,862	0,480		
			Poly	-	-	-		
		50	Linear	30,688	2,743	-0,619		
			Poly	-	-	-		
LMO		80	Linear	1,391	1,115	0,896		
			Poly	3,793	3,346	-0,569		
		50	Linear	1,112	1,025	0,912		
			Poly	4,133	3,602	-1,812		

Berdasarkan tabel diatas, variasi terpilih dipilih dengan memperhatikan nilai RMSE dan MAE yang paling rendah serta R² yang paling mendekati nilai 1. Pada bagian nilai R² terlihat bahwa beberapa variasi memiliki nilai yang negatif, bahkan memiliki nilai negatif ratusan sehingga variasi tersebut diberi tanda garis “-“. Nilai R² negatif dimungkinkan secara matematis dengan interpretasi secara sederhana bahwa data memiliki performa yang sangat buruk terhadap pembanding [27], yang pada tabel ini dibandingkan dengan model degradasi baterai NMC referensi.

4.6 Optimasi Variasi Terpilih

Ketiga variasi terpilih lalu dioptimasi dengan *tuning hyperparameter* lanjutan agar didapat kurva degradasi yang memiliki *goodness-of-fit* terhadap model referensi paling baik. Tuning dilakukan dengan melakukan gridsearchCV pada *hyperparameter* C. Dua dari tiga variasi terpilih menunjukkan bahwa data lebih cocok menggunakan kernel linear dan satu cocok dengan kernel polinomial. Pada penggunaan kernel linear dan polinomial, C merupakan *hyperparameter* yang paling berpengaruh dibanding *hyperparameter* lain (epsilon dan gamma) [28]. Pada keadaan tanpa *tuning*, pustaka RVP sebenarnya telah menetapkan nilai *hyperparameter* C=1 [29].

Proses *tuning* ini diharapkan menghilangkan faktor kesalahan yang disebabkan oleh pemakaian *hyperparameter* yang kurang sesuai. Hasil sebelum dan sesudah di-*tuning* akan dibandingkan dan dipilih yang memiliki R^2 estimasi RVP lebih tinggi. Nilai R^2 estimasi RVP merupakan nilai koefisien determinasi R^2 yang membandingkan nilai sebenarnya dan prediksi yang dihasilkan dalam proses pembelajaran mesin RVP.

Tabel 4.3 Perbandingan R^2 estimasi RVP Sebelum dan Sesudah Tuning

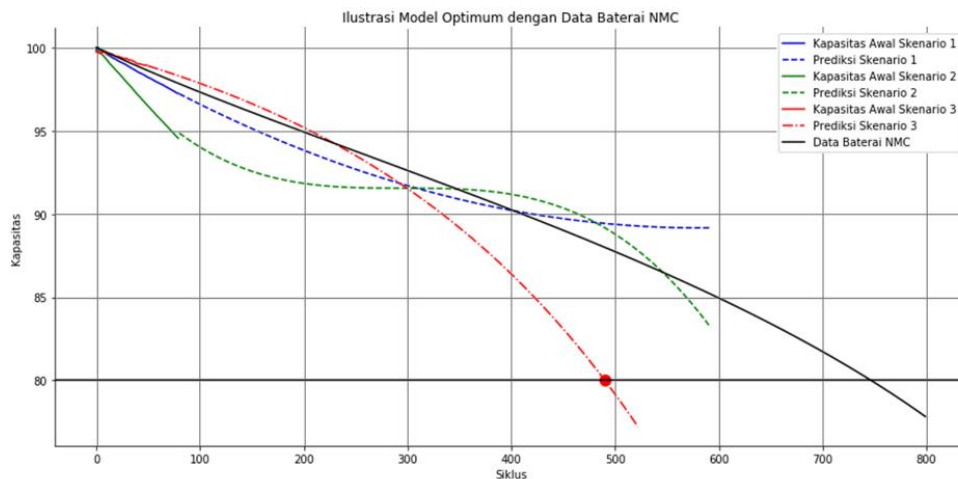
Skenario	R^2 estimasi RVP	
	Sebelum Tuning	Setelah Tuning
Skenario 1	0,971	0,964
Skenario 2	0,990	0,991
Skenario 3	0,907	0,959

Tabel di atas menunjukkan bahwa pada dasarnya data sudah dipelajari dengan baik oleh pembelajaran mesin. Untuk skenario 1 dipilih nilai sebelum *tuning* sedangkan 2 dan 3 dipilih hasil setelah *tuning*. Sehingga indikator performansi untuk masing-masing skenario adalah sebagai berikut.

Tabel 4.4 Variasi Terpilih Untuk Tiap Skenario Beserta Indikator Performansinya

Skenario	Data Eksperimen	Data Latih Referensi	Kernel	Indikator Performansi		
				RMSE	MAE	R^2
Skenario 1	50 Siklus awal	LFP 1C	Linear	2,143	1,934	0,792
Skenario 2	80 Siklus awal	LFP 1C	Polinomial	1,932	1,715	0,338
Skenario 3	50 Siklus awal	LMO 1C	Linear	1,312	1,025	0,911

Tabel 4.4 menunjukkan informasi variasi terpilih dari setiap skenario beserta indikator performansinya. Indikator performansi ini merupakan nilai dari hasil prediksi yang dibandingkan dengan model degradasi baterai Li-Ion tipe NMC sehingga nilai R^2 ini bukanlah nilai yang sama seperti pada Tabel 4.6. Kombinasi pada skenario 3 memiliki nilai error RMSE dan MAE paling kecil dibanding dengan kedua variasi lain walau nilai ini hanya berkisar 1,312 sampai 2,143 untuk RMSE dan 1,025 sampai 1,934 untuk MAE yang apabila dibandingkan dengan data kapasitas baterai yang berkisar dari 100 hingga 80 pada saat degradasinya tidak terlalu besar. Namun untuk nilai R^2 hanya skenario 1 dan 3 memiliki nilai dengan ikatan kuat terhadap model degradasi baterai NMC, sedangkan pada skenario 2 didapat nilai dengan ikatan lemah dengan model degradasi baterai NMC [30].



Gambar 4.19 Perbandingan Model Optimum dengan kurva degradasi baterai NMC.

Berdasarkan hasil ini didapat bahwa skenario 1 dapat diprediksi dengan menggunakan data simulasi 50 siklus dan RVP dengan kernel linear, namun prediksi tidak dapat membuat kurva degradasi dengan bentuk konkaf ke arah bawah sehingga tidak dapat di prediksi nilai degradasinya.

Berdasarkan hasil ini didapat bahwa skenario 2 dapat diprediksi dengan menggunakan data simulasi 80 siklus dan RVP dengan kernel polinomial, skenario ini tidak menyentuh kapasitas 80% dengan rentang data yang tersedia.

Berdasarkan hasil ini didapat bahwa skenario 3 dapat diprediksi dengan menggunakan data simulasi 50 siklus dan RVP dengan kernel linear, skenario ini menyentuh kapasitas 80% pada siklus ke-490.

BAB V

KESIMPULAN

5.1 Kesimpulan

Pada penelitian ini, kami telah berhasil mengembangkan algoritma degradasi sistem baterai dari beberapa skenario berkendara sepeda listrik. Kami juga telah menyelesaikan seluruh sasaran yang ditargetkan pada penelitian ini, yaitu,

1. Pemodelan telah berhasil dilakukan dengan menggunakan data hasil modul akuisisi data yang menggunakan ESP32 sebagai modul komunikasi. Data berkendara tiap skenario terdiri atas satu siklus pemakaian yang diolah dengan AWTLS agar rentang siklus dapat ditambah menjadi 50 atau 80 siklus sebelum masuk ke pembelajaran mesin.
2. Dari pengujian antara beberapa model pembelajaran mesin antara lain Decision Tree, Random Forest, XGBoost, dan LSTM, diperoleh RVP sebagai model yang dapat memperlihatkan tren penurunan dengan nilai R^2 berkisar antara 0,991–0,995 dengan data latih baterai LMO dan LFP.
3. Hasil Skenario 3 yang dimodelkan dengan 50 siklus awal data percobaan, data referensi LFP 1C, dengan menggunakan model pembelajaran mesin RVP dengan kernel linier memiliki nilai performansi paling baik dan berhasil mencapai kapasitas degradasi pada 490 siklus dengan nilai R^2 0,911, sehingga variasi optimal model ini dapat merepresentasikan degradasi baterai NMC

5.2 Saran

Adapun saran yang dapat dilakukan untuk penelitian selanjutnya adalah sebagai berikut :

1. Mempertimbangkan faktor *charging* pada siklus baterai dan pengaruh temperatur terhadap performa degradasi sistem baterai.
2. Mengimplementasikan algoritma AWTLS dan model pembelajaran mesin di sistem tertanam pada sepeda listrik untuk prediksi degradasi.

DAFTAR PUSTAKA

- [1] Liu, W., Song, M. S., Kong, B., & Cui, Y. (2017). *Flexible and stretchable energy storage: recent advances and future perspectives*. *Advanced materials*, 29(1), 1603436.
- [2] X., Ai, W., Marlow, M. N., Patel, Y., & Wu, B. (2019). The effect of cell-to-cell variations and thermal gradients on the performance and degradation of lithium-ion battery packs. *Applied Energy*, 248, 489–499. doi: 10.1016/j.apenergy.2019.04.108
- [3] Rodríguez, A., Plett, G. L., & Trimboli, M. S. (2018). *Improved transfer functions modeling linearized lithium-ion battery-cell internal electrochemical variables*. *Journal of Energy Storage*, 20, 560–575. doi: 10.1016/j.est.2018.06.015
- [4] Broussely, M., Biensan, P., Bonhomme, F., Blanchard, P., Herreyre, S., Nechev, K., & Staniewicz, R. J. (2005). *Main aging mechanisms in Li ion batteries*. *Journal of Power Sources*, 146(1-2), 90–96. doi:10.1016/j.jpowsour.2005.03.172
- [5] Han, X., Lu, L., Zheng, Y., Feng, X., Li, Z., Li, J., & Ouyang, M. (2019). A review on the key issues of the lithium ion battery degradation among the whole life cycle. *ETransportation*, 1, 100005. <https://doi.org/10.1016/j.etrans.2019.100005>
- [6] Birkel, C. R., Roberts, M. R., McTurk, E., Bruce, P. G., & Howey, D. A. (2017). *Degradation diagnostics for lithium ion cells*. *Journal of Power Sources*, 341, 373–386. doi:10.1016/j.jpowsour.2016.12.011
- [7] Vezzini, A. (2014). *Lithium-Ion Battery Management*. *Lithium-Ion Batteries*, 345–360. doi:10.1016/b978-0-444-59513-3.00015-7
- [8] Pastor-Fernández, C., Bruen, T., Widanage, W. D., Gama-Valdez, M. A., & Marco, J. (2016). *A Study of Cell-to-Cell Interactions and Degradation in Parallel Strings: Implications for the Battery Management System*. *Journal of Power Sources*, 329, 574–585. doi:10.1016/j.jpowsour.2016.07.121
- [9] Crawford, A. J., Huang, Q., Kintner-Meyer, M. C., Zhang, J.-G., Reed, D. M., Sprenkle, V. L., ... Choi, D. (2018). Lifecycle comparison of selected Li-ion battery chemistries under grid and electric vehicle duty cycle combinations. *Journal of Power Sources*, 380, 185–193. doi: 10.1016/j.jpowsour.2018.01.080
- [10] Bucher, D., Buffat, R., Froemelt, A., & Raubal, M. (2019). Energy and greenhouse gas emission reduction potentials resulting from different

- commuter electric bicycle adoption scenarios in Switzerland. *Renewable and Sustainable Energy Reviews*, 114, 109298. doi: 10.1016/j.rser.2019.109298
- [11] Cortez, C., & Vapnik, V. (1995). Support-Vector Networks. *Support-Vector Network*.
- [12] Li, L. L., Zhao, X., Tseng, M.-L., & Tan, R. R. (2019). Short-term wind power forecasting based on support vector machine with improved dragonfly algorithm. *Journal of Cleaner Production*, 242.
- [13] Elsworth, S., Guttel, S. (2020). *Time Series Forecasting Using LSTM Networks: A Symbolic Approach*. *arXiv:2003.05672 [cs.LG]*
- [14] Kim, K., & Hong, J. (2017). A hybrid decision tree algorithm for mixed numeric and categorical data in regression analysis. *Pattern Recognition Letters*, 98, 39–45. <https://doi.org/10.1016/j.patrec.2017.08.011>
- [15] Iskenderoglu FC et al., Comparison of support vector regression and random forest algorithms for estimating the SOFC output voltage by considering hydrogen flow rates, *International Journal of Hydrogen Energy*, <https://doi.org/10.1016/j.ijhydene.2020.07.265>
- [16] Plett, G. L. (2011). Recursive approximate weighted total least squares estimation of battery cell total capacity. *Journal of Power Sources*, 196(4), 2319–2331. <https://doi.org/10.1016/j.jpowsour.2010.09.048>
- [17] Plett, L. G. Battery State of Health Estimation. Coursera Online Course. University of Colorado Boulder.
- [18] V. Sangwan, R. Kumar and A. K. Rathore, "An Empirical Capacity Degradation Modeling and Prognostics of Remaining Useful Life of Li-ion Battery using Unscented Kalman Filter," 2018 8th IEEE India International Conference on Power Electronics (IICPE), JAIPUR, India, 2018, pp. 1-6, doi: 10.1109/IICPE.2018.8709470.
- [19] Li, J., Wang, D., Deng, L., Cui, Z., Lyu, C., Wang, L., Pecht, M. (2020). Aging modes analysis and physical parameter identification based on a simplified electrochemical model for lithium-ion batteries. *Journal of Energy Storage*
- [20] Yang, F., Wang, D., Xing, Y., & Tsui, K.-L. (2017). Prognostics of Li(NiMnCo)O₂-based lithium-ion batteries using a novel battery degradation model. *Microelectronics Reliability*, 70, 70–78. <https://doi.org/10.1016/j.microrel.2017.02.002>
- [21] Chung, K.-J., & Hsiao, C.-C. (2012). Accelerated Degradation Assessment of 18650 Lithium-Ion Batteries. *2012 International Symposium on Computer, Consumer and Control*, 930–933. <https://doi.org/10.1109/is3c.2012.242>
- [22] Pexeiro, M. (2019). The Complete Guide to Time Series Analysis and Forecasting. Towards Data Science. Disadur dari :

<https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775>

- [23] Chen, X., Huang, J., Han, Z., Gao, H., Liu, M., Li, Z., ... Huang, Y. (2020). *The importance of short lag-time in the runoff forecasting model based on long short-term memory*. *Journal of Hydrology*, 125359. doi:10.1016/j.jhydrol.2020.12535
- [24] Olah. J. (2015). Understanding LSTM Networks. Colah's blog. Disadur dari : <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [25] Azimi, H., Bonakdari, H., & Ebtehaj, I. (2019). Design of radial basis function-based support vector regression in predicting the discharge coefficient of a side weir in a trapezoidal channel. *Applied Water Science*, 9(4), 1–12. <https://doi.org/10.1007/s13201-019-0961-5>
- [26] Cheng, K., Lu, Z., Wei, Y., Shi, Y., & Zhou, Y. (2017). Mixed kernel function support vector regression for global sensitivity analysis. *Mechanical Systems and Signal Processing*, 96, 201–214. <https://doi.org/10.1016/j.ymssp.2017.04.014>
- [27] J Chem Inf Model. 2015 July 27; 55(7): 1316–1322. doi:10.1021/acs.jcim.5b00206.
- [28] Yildirim, S. (2020). *Hyperparameter Tuning for Support Vector Machines — C and Gamma Parameters*. *Towards Data Science*. Disadur dari : <https://towardsdatascience.com/hyperparameter-tuning-for-support-vector-machines-c-and-gamma-parameters-6a5097416167>
- [29] *Sklearn.svm.SVR*. (n.d.), Disadur dari : <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html>
- [29] Moore, D. S., Notz, W. I., & Flinger, M. A. (2013). *The basic practice of statistics (6th ed.)*. New York, NY: W. H. Freeman and Company. pp 138.

LAMPIRAN A KODE SUMBER

A.1 Kode Sumber AWTLS

```
1. function [Qhat] = parameterAWTLS(x, n_cycle)
2.
3. % {
4. function x = accum_cycle(x, n_cycle)
5. matrix = ones(n_cycle);
6. matrix = matrix(:,1);
7. matrix = matrix';
8. x = x*matrix;
9. x = x(:);
10. % }
11.
12. x = accum_cycle(x, n_cycle);
13.
14. Q0 = 100;          % actual new - cell capacity of cell
15. maxI = 5* Q0;      % must be able to measure current up to +/- maxI
16. precisionI = 1024; % 10- bit precision on current sensor
17. slope = -0.001;
18. Qnom = 0.99* Q0;   % ** nominal capacity , used for init . of recursive methods
19. xmax = 1; xmin = x(end) ; % ** range of the x(i) variables
20. theCase = 2;        % ** random interval between updates
21. mode = 0.5; sigma = 0.6; % ** needed for case 2
22. socnoise = std(x);  % ** standard deviation of x(i)
23. gamma = 1;         % forgetting factor
24. plotTitle = 'Estimasi Degradasi Baterai';
25.
26. n = length(x);      % number of data points collected
27. Q = (Q0+ slope *(1: n)); % evolution of true capacity over time
28. Q = Q';
29. x = (( xmax - xmin ) * rand ( n ,1) + xmin ); % true x(i), without noise
30. y = Q.*x;          % true y(i), without noise
31. binsize = 2* maxI / precisionI ; % resolution of current sensor
32. rn1 = ones ( n ,1); % init std . dev . for each measurement
33. SigmaX = socnoise * rn1 ; % scale Gaussian std. dev .
34.
35. if theCase == 1      % the typical case
36.     rn2 = rn1 ;      % same scale on y(i) as x(i) noise
37.     mu = log( mode )+ sigma ^2;
38.     m = 3600* lognrnd (mu ,sigma ,n ,1);
39.     SigmaY = binsize * sqrt (m /12) /3600* rn2; % std. dev . for y(i)
40. else                % this case will be discussed for BEV scenario 3
41.     mu = log( mode )+ sigma ^2;
42.     m = 3600* lognrnd (mu ,sigma ,n ,1);
43.     SigmaY = binsize * sqrt (m /12) /3600; % std. dev . for y(i)
44. end
45.
46. measX = x + SigmaX .* randn ( n ,1) ; % measured x(i) data , including noise
47. measY = y + SigmaY .* randn ( n ,1) ; % measured y(i) data , including noise
48.
49. % AWTLS Function
50.
51. measX = measX(:); measY = measY(:); SigmaX = SigmaX(:); SigmaY = SigmaY(:);
52.
53. % Reserve some memory
54. zero = 1:length(measX);
55. zero = (0 * zero)';
56. Qhat = zero;
57. % Qhat = zeros(length(measX));
```

```

58. SigmaQ = Qhat;
59. K = sqrt(SigmaX(1)/SigmaY(1));
60.
61. % Initialize some variables used for the recursive methods
62. c1 = 0; c2 = 0; c3 = 0; C1 = 0; C2 = 0; C3 = 0; C4 = 0; C5 = 0; C6 = 0;
63.
64. if Qnom ~= 0
65.
66.     SigmaY0=SigmaY(1);
67.
68.     c1 = 1/SigmaY0; c2 = Qnom/SigmaY0; c3 = Qnom^2/SigmaY0;
69.     C1 = 1/(K^2*SigmaY0); C2 = K*Qnom/(K^2*SigmaY0); C3 = K^2*Qnom^2/(K^2*SigmaY0);
70.     % Init C4...C6 assuming SigmaX0 = K^2*SigmaY0 to match TLS
71.     C4 = 1/(K^2*SigmaY0); C5 = K*Qnom/(K^2*SigmaY0); C6 = K^2*Qnom^2/(K^2*SigmaY0);
72. end
73.
74. for iter = 1:length(measX)
75.     % Compute some variables used for the recursive methods
76.     c1 = gamma*c1 + measX(iter)^2/SigmaY(iter);
77.     c2 = gamma*c2 + measX(iter)*measY(iter)/SigmaY(iter);
78.     c3 = gamma*c3 + measY(iter)^2/SigmaY(iter);
79.
80.     C1 = gamma*C1 + measX(iter)^2/(K^2*SigmaY(iter));
81.     C2 = gamma*C2 + K*measX(iter)*measY(iter)/(K^2*SigmaY(iter));
82.     C3 = gamma*C3 + K^2*measY(iter)^2/(K^2*SigmaY(iter));
83.     C4 = gamma*C4 + measX(iter)^2/SigmaX(iter);
84.     C5 = gamma*C5 + K*measX(iter)*measY(iter)/SigmaX(iter);
85.     C6 = gamma*C6 + K^2*measY(iter)^2/SigmaX(iter);
86.
87.     % AWTLs with pre-scaling
88.     r = roots([C5 (-C1+2*C4-C6) (3*C2-3*C5) (C1-2*C3+C6) -C2]);
89.     r = r(r==conj(r)); % discard complex-conjugate roots
90.     r = r(r>0); % discard negative roots
91.     Jr = ((1/(r.^2+1).^2).*(r.^4*C4-2*C5*r.^3+(C1+C6)*r.^2-2*C2*r+C3));
92.     J = min(Jr);
93.     Q = r(Jr==J); % keep Q that minimizes cost function
94.     H = (2/(Q.^2+1).^4)*(-2*C5*Q.^5+(3*C1-6*C4+3*C6)*Q.^4+(-12*C2+16*C5)*Q.^3 ...
95.         +(-8*C1+10*C3+6*C4-8*C6)*Q.^2+(12*C2-6*C5)*Q+(C1-2*C3+C6));
96.     Qhat(iter) = Q/K;
97.     SigmaQ(iter) = 2/H/K^2;
98. end
99.
100. hold on
101.
102. plot(Qhat(:,1),'g','linewidth',3); % AWTLs
103.
104. xlabel('Algorithm update index');
105. ylabel('Capacity estimate (Ah)');
106. title(sprintf('%s: Capacity estimates, bounds',plotTitle));
107.
108. % Plot 3-sigma bounds
109. plot(Qhat(:,1)+3*sqrt(SigmaQ(:,1)), 'g--','linewidth',0.5);
110. plot(Qhat(:,1)-3*sqrt(SigmaQ(:,1)), 'g--','linewidth',0.5);
111.
112. % Plot over top to make sure estimate is on top of bounds
113. plot(Qhat(:,1),'g','linewidth',3); % AWTLs
114.
115. % Plot true capacity
116. plot(1:length(x),Q,'k--','linewidth',1);

```

A.2 Kode Sumber Proses Prediksi, *tuning* dan *ploting*

```

1. import pandas as pd
2. import numpy as np
3. import matplotlib.pyplot as plt
4. import seaborn as sns
5. import math
6.
7. from sklearn import preprocessing, svm
8. from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
9. from sklearn.model_selection import RandomizedSearchCV
10. from sklearn.model_selection import GridSearchCV
11. from sklearn.model_selection import KFold, cross_val_score, train_test_split
12.
13. import statsmodels.tsa.api as smt
14. import pickle
15.
16. series = pd.read_excel('LFP 1C.xlsx')
17. komparator = pd.read_excel('NMC.xlsx')
18. data = series
19.
20. jumlah_cycle = 50
21.
22. data['capacity_diff'] = data['Capacity'].diff()
23. data = data.dropna()
24.
25. #-----
26. #Lakukan ADF Test
27.
28. dt_data = data.set_index('Cycle').drop('Capacity', axis=1)
29. dt_data.dropna(axis=0)
30. smt.stattools.adfuller(dt_data['capacity_diff'], regression = 'ct')
31.
32. #-----
33. #Define semua fungsi yg akan dibutuhkan
34.
35. def generate_supervised(data):
36.     supervised_df = data.copy()
37.
38.     #create column for each lag
39.     for i in range(1,7):
40.         col_name = 'lag_' + str(i)
41.         supervised_df[col_name] = supervised_df['capacity_diff'].shift(i)
42.
43.     #drop null values
44.     supervised_df = supervised_df.dropna().reset_index(drop=True)
45.
46.     #supervised_df.to_csv('./data/model_df.csv', index=False)
47.
48.     return supervised_df
49.
50. def inisiasi(cyclesum, skenario):
51.
52.     real = pd.read_excel(f'Data Degradasi {cyclesum} Cycle Skenario {skenario}.xlsx')
53.     real = real.drop('Cycle', axis = 1)
54.     model_df['Real'] = real
55.
56. def grafik_raw(cycle_amount):
57.
58.     fig, ax = plt.subplots(figsize=(15,7))
59.
60.     sns.lineplot(alldata1.index[:cycle_amount], alldata1['Data Awal'][:cycle_amount], data=alldata1, ax=ax,
61.                 label='Kapasitas Awal Skenario 1', color='mediumblue')
62.     frc1 = sns.lineplot(alldata1.index[cycle_amount:], alldata1['Prediksi Skenario 1'][cycle_amount:], data=alldata1, ax=ax,

```

```

63.         label='Prediksi Skenario 1', color='mediumblue')
64.     frc1.lines[1].set_linestyle("--")
65.
66.     sns.lineplot(alldata2.index[:cycle_amount], alldata2['Data Awal'][:cycle_amount], data=alldata2, ax=ax,
67.         label='Kapasitas Awal Skenario 2', color='green')
68.     frc2 = sns.lineplot(alldata2.index[cycle_amount:], alldata2['Prediksi Skenario 2'][:cycle_amount:], data=alldata2, ax=ax,
69.         label='Prediksi Skenario 2', color='green', dashes = False)
70.     frc2.lines[n9].set_linestyle("--")
71.
72.     sns.lineplot(alldata3.index[:cycle_amount], alldata3['Data Awal'][:cycle_amount], data=alldata3, ax=ax,
73.         label='Kapasitas Awal Skenario 3', color='red')
74.     frc3 = sns.lineplot(alldata3.index[cycle_amount:], alldata3['Prediksi Skenario 3'][:cycle_amount:], data=alldata3, ax=ax,
75.         label='Prediksi Skenario 3', color='red', dashes = False )
76.     frc3.lines[n7].set_linestyle("-.")
77.
78.     sns.lineplot(series.index, series['Capacity'], data=series, ax=ax,
79.         label='Data Baterai LFP', color='black')
80.
81.     ax.set(xlabel = "Siklus",
82.         ylabel = "Kapasitas",
83.         title = f"Prediksi Kapasitas {jumlah_cycle} Siklus Awal, Kernel RBF, LiFePO4 2C Discharging")
84.
85.     ax.legend()
86.     sns.despine()
87.
88.     #-----
89.     #Algoritma di bawah di compile untuk membuat SATU plot degradasi skenario
90.
91.     model_df = generate_supervised(data)
92.
93.     #Masukin skenario yang ingin diprediksi (1,2,3)
94.     inisiasi(jumlah_cycle, 3)
95.
96.     X = np.array(model_df.drop(columns = ['Real'], axis = 1))
97.     X = preprocessing.scale(X)
98.     X_lately = X[jumlah_cycle:]
99.     X = X[:jumlah_cycle]
100.     model_df.dropna(inplace=True)
101.     y = np.array(model_df['Real'])
102.     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
103.
104.     rs = GridSearchCV(svm.SVR(),
105.         {
106.             'C': [0.1, 1, 100, 1000],
107.             'epsilon': [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10],
108.             'gamma': [0.0001, 0.001, 0.005, 0.1, 1, 3, 5]
109.         },
110.         cv=5, return_train_score=False)
111.     rres_svr = rs.fit(X, y)
112.     best_params_svr = rres_svr.best_params_
113.
114.     #Tentukan kernel
115.     regm = svm.SVR(kernel='rbf',
116.         C=best_params_svr['C'],
117.         epsilon=best_params_svr['epsilon'],
118.         gamma=best_params_svr['gamma'])
119.
120.     regm.fit(X_train, y_train)
121.
122.     forecast_set = regm.predict(X_lately)
123.

```

```

124. mse = mean_squared_error(forecast_set, komparator['capacity'][jumlah_cycle:(jumlah_cycle+forecast_set.shape
    [0]))
125. rmse = np.sqrt(mse)
126. mae = mean_absolute_error(forecast_set, komparator['capacity'][jumlah_cycle:(jumlah_cycle+forecast_set.shap
    e[0]))
127. r2 = r2_score(forecast_set, komparator['capacity'][jumlah_cycle:(jumlah_cycle+forecast_set.shape[0]))
128.
129.
130. #-----
131. #Setelah membuat satu plot degradasi, buat Dataframe untuk hasil prediksinya
132. #Misalnya, apabila telah selesai meng-compile prediksi untuk skenario 1,
133. #compile baris forecast_linear_1 sampai alldata1.rename.
134. #Lalu sebelum mengcompile baris forecast_linear_1 sampai alldata1.rename,
135. #compile lagi algoritma sebelumnya, namun untuk skenario 2.
136.
137. forecast_linear_1 = pd.DataFrame(forecast_set, columns = ['Prediksi Skenario 1'])
138. alldata1 = pd.concat([model_df['Real'].dropna(), forecast_linear_1, ignore_index=True)
139. alldata1.rename(columns = {0:'Data Awal'}, inplace = True)
140. forecast_linear_2 = pd.DataFrame(forecast_set, columns = ['Prediksi Skenario 2'])
141. alldata2 = pd.concat([model_df['Real'].dropna(), forecast_linear_2, ignore_index=True)
142. alldata2.rename(columns = {0:'Data Awal'}, inplace = True)
143.
144. forecast_linear_3 = pd.DataFrame(forecast_set, columns = ['Prediksi Skenario 3'])
145. alldata3 = pd.concat([model_df['Real'].dropna(), forecast_linear_3, ignore_index=True)
146. alldata3.rename(columns = {0:'Data Awal'}, inplace = True)
147.
148. #-----
149. #Perlihatkan grafik
150.
151. grafik_raw(jumlah_cycle)
152.
153. #-----
154. #Algoritma di bawah untuk menampilkan titik perpotongan kurva dengan garis 80%
155.
156. x = np.array(alldata1.index[jumlah_cycle:])
157.
158. # Grafik yang berhasil memotong garis degradasi JANGAN di-comment
159. # f1 berarti skenario 1, f2 skenario 2, f3 skenario 3
160. # g merupakan garis batas degradasi di nilai 80%
161.
162. f1 = np.array(alldata1['Prediksi Skenario 1'][jumlah_cycle:])
163. f2 = np.array(alldata2['Prediksi Skenario 2'][jumlah_cycle:])
164. f3 = np.array(alldata3['Prediksi Skenario 3'][jumlah_cycle:])
165. g = np.full((len(alldata3.index[jumlah_cycle:])), 80)
166.
167. plt.plot(x, f1, '-')
168. plt.plot(x, f2, '-')
169. plt.plot(x, f3, '-')
170. plt.plot(x, g, '-')
171.
172. #idx mengindikasikan titik perpotongan antara kurva degradasi dengan garis g
173. #idx1 untuk skenario 1, etc.. apabila skenario TIDAK memotong garis g, berikan '#'
174. idx1 = np.argwhere(np.diff(np.sign(f1 - g))).flatten()
175. idx2 = np.argwhere(np.diff(np.sign(f2 - g))).flatten()
176. idx3 = np.argwhere(np.diff(np.sign(f3 - g))).flatten()
177.
178. plt.plot(x[idx2], f2[idx2], 'ro')
179. plt.show()
180.
181. #-----
182. #Compile untuk menampilkan hasil akhir
183.
184. cycle_amount = jumlah_cycle

```



```

185. fig, ax = plt.subplots(figsize=(15,7))
186.
187. sns.lineplot(alldata1.index[:cycle_amount], alldata1['Data Awal'][:cycle_amount], data=alldata1, ax=ax,
188.             label='Kapasitas Awal Skenario 1', color='mediumblue')
189. frc1 = sns.lineplot(alldata1.index[cycle_amount:], alldata1['Prediksi Skenario 1'][:cycle_amount:], data=alldata1,
190.                   ax=ax,
191.                   label='Prediksi Skenario 1', color='mediumblue')
192. frc1.lines[1].set_linestyle("--")
193. sns.lineplot(alldata2.index[:cycle_amount], alldata2['Data Awal'][:cycle_amount], data=alldata2, ax=ax,
194.             label='Kapasitas Awal Skenario 2', color='green')
195. frc2 = sns.lineplot(alldata2.index[cycle_amount:], alldata2['Prediksi Skenario 2'][:cycle_amount:], data=alldata2,
196.                   ax=ax,
197.                   label='Prediksi Skenario 2', color='green', dashes = False)
198. frc2.lines[n9].set_linestyle("--")
199. sns.lineplot(alldata3.index[:cycle_amount], alldata3['Data Awal'][:cycle_amount], data=alldata3, ax=ax,
200.             label='Kapasitas Awal Skenario 3', color='red')
201. frc3 = sns.lineplot(alldata3.index[cycle_amount:], alldata3['Prediksi Skenario 3'][:cycle_amount:], data=alldata3,
202.                   ax=ax,
203.                   label='Prediksi Skenario 3', color='red', dashes = False )
204. frc3.lines[n7].set_linestyle("-.")
205. #Atur batas degradasinya
206. frc1.axhline(80, ls='-', color = 'black')
207.
208. #Pilih grafik skenario berapa yang ingin ditampilkan titik perpotongannya
209. plt.scatter(idx1+jumlah_cycle, 80, marker='o', s=100, c = 'blue')
210. plt.scatter(idx2+jumlah_cycle, 80, marker='o', s=100, c = 'green')
211. plt.scatter(idx3+50, 80, marker='o', s=100, c = 'red')
212.
213. ax.set(xlabel = "Siklus",
214.       ylabel = "Kapasitas",
215.       title = f"Prediksi Kapasitas {jumlah_cycle} Siklus Awal, Kernel Linear, LiFePO4 2C Discharging")
216.
217. ax.legend()
218. sns.despine()

```