

## BAB 2 LANDASAN TEORI

Landasan Teori ini menjabarkan teori-teori yang mendasari penelitian terkait *Electric Vehicle*, *Lithium Ion Battery*, *State of Charge*, *State of Health*, algoritma *Random Forest*.

### 2.1 Electric Vehicle

*Electric Vehicle (EV)* adalah Kendaraan bermotor berbasis listrik, termasuk *Battery Electric Vehicle (BEV)*, *Plug-in Hybrid Electric Vehicle (PHEV)*, dan *Hybrid Electric Vehicle (HEV)* yang keseluruhan memanfaatkan listrik untuk meningkatkan efisiensi kendaraan. *BEV* beroperasi hanya dengan listrik dan biasanya memiliki jarak tempuh 150 hingga 400 mil, sehingga cocok untuk berbagai kebutuhan berkendara sehari-hari [7]. Departemen Energi AS telah mendanai beberapa proyek kendaraan listrik di beberapa negara bagian untuk mendukung adopsi kendaraan listrik dan infrastruktur pengisian daya.

Berdasarkan Kepentingan Publik dan Infrastruktur, sekitar 38% orang Amerika sangat atau agak mungkin mempertimbangkan kendaraan listrik untuk pembelian berikutnya [8]. Pemerintahan Biden telah mengalokasikan 5 miliar dolar untuk membangun jaringan stasiun pengisian daya kendaraan listrik, yang bertujuan untuk mengatasi terbatasnya ketersediaan stasiun pengisian daya publik [9]. Terdapat kepercayaan publik yang terbatas pada kemampuan negara untuk membangun infrastruktur yang diperlukan untuk mendukung sejumlah besar kendaraan listrik di jalan raya, dengan 53% menyatakan bimbang. [8].

Pandangan tentang penghapusan mobil bensin secara bertahap, agenda iklim Presiden Joe Biden melibatkan percepatan transisi ke *EV*, termasuk batas emisi yang diusulkan untuk produsen mobil dan rencana untuk menghentikan produksi kendaraan bertenaga gas baru pada tahun 2035 [10]. Namun, gagasan untuk menghapus mobil dan truk berbahan bakar bensin pada tahun 2035 menghadapi lebih banyak tentangan dari masyarakat daripada dukungan, dengan 59% menentang dan 40% mendukung [8].

## 2.2 Lithium Ion Battery

Baterai *lithium ion* sering disebut sebagai baterai *Li-ion*, adalah jenis baterai isi ulang yang memanfaatkan interkalasi bersifat reversibel ion  $Li^+$  menjadi padatan menghantar elektron. Baterai ini telah menjadi sumber energi yang dominan, terutama di pangsa elektronik, karena kepadatan energinya yang tinggi, siklus dan masa simpan yang lebih lama [11]. Baterai ini juga telah mendapatkan perhatian yang signifikan dalam kendaraan penggerak listrik, yang berkontribusi pada meningkatnya minat terhadap energi bersih dan kelestarian lingkungan.

Berdasarkan beberapa literatur terkait memberikan gambaran umum yang komprehensif tentang konsep dasar, kemajuan saat ini, dan tantangan yang terkait dengan baterai *Li-ion*. Literatur tersebut memperkenalkan prinsip-prinsip dasar baterai *Li-ion* dan membahas kemajuan dan hambatan di lapangan. Selain itu, ulasan terkait bahan baterai *Li-ion* oleh beberapa penelitian mencakup perkembangan teknologi utama dan tantangan ilmiah untuk berbagai macam elektrode baterai *Li-ion*, yang menawarkan wawasan tentang masa kini dan masa depan teknologi baterai *Li-ion* [12][13].

Kisaran suhu optimal untuk baterai *lithium ion* tergantung pada jenis baterai tertentu dan tujuan penggunaannya. Namun demikian, beberapa panduan umum dapat diberikan:

1. Suhu pengisian daya.

Baterai *lithium ion* harus diisi pada suhu antara  $-10^{\circ}\text{C}$  dan  $45^{\circ}\text{C}$  ( $14^{\circ}\text{F}$  dan  $113^{\circ}\text{F}$ ) [14].

2. Suhu pemakaian.

Baterai *lithium ion* dapat beroperasi dalam kisaran suhu yang lebih luas, dengan beberapa sumber menyarankan kisaran dari  $-4^{\circ}\text{F}$  hingga  $130^{\circ}\text{F}$  untuk pengosongan [12].

3. Suhu penyimpanan.

Baterai *lithium ion* harus disimpan dalam kondisi sejuk dan kering pada suhu  $15^{\circ}\text{C}$  ( $59^{\circ}\text{F}$ ) [15].

Baterai *lithium ion* dioperasikan di luar kisaran suhu optimalnya dapat menimbulkan beberapa dampak negatif terhadap performa dan keamanannya:

1. Kapasitas berkurang.

Baterai yang dioperasikan pada suhu ekstrem, baik tinggi maupun rendah,

dapat mengalami penurunan kapasitas, yang menyebabkan masa pakai yang lebih pendek dan kinerja yang berkurang [11][14].

2. Peningkatan resistensi internal.

Temperatur yang tinggi dapat menyebabkan resistensi internal baterai meningkat, yang mengurangi laju reaksi kimia dan dapat menyebabkan pengurangan kapasitas baterai secara permanen [15].

3. *Overheat*.

Baterai *lithium ion* yang dioperasikan pada suhu tinggi dapat meningkatkan risiko *overheat*, kondisi berbahaya di mana baterai menjadi terlalu panas dan dapat menyebabkan kebakaran, ledakan, dan masalah keselamatan lainnya [11][14].

4. Reaksi kimia.

Suhu yang ekstrem dapat menyebabkan reaksi kimia di dalam baterai yang dapat menyebabkan kerusakan baterai, serta kebakaran dan ledakan di tempat kerja [14].

Untuk memastikan kinerja dan keamanan yang optimal, maka penting untuk mengoperasikan baterai *lithium ion* dalam rentang suhu yang direkomendasikan, yaitu berkisar antara  $-10^{\circ}\text{C}$  hingga  $45^{\circ}\text{C}$  untuk pengisian daya dan  $-4^{\circ}\text{F}$  hingga  $130^{\circ}\text{F}$  untuk pengosongan daya.

Dengan demikian, baterai *lithium ion* adalah teknologi penyimpanan energi yang banyak digunakan dan memiliki peranan penting dengan aplikasi yang signifikan dalam elektronik konsumen maupun kendaraan listrik. Baterai ini telah mengalami perkembangan substansial dan menjadi subjek penelitian yang sedang berlangsung untuk mengatasi berbagai tantangan dan lebih meningkatkan kinerja dan keamanan kedepannya.

### 2.3 State of Charge

*State of Charge* (SoC) baterai *lithium ion* adalah parameter penting yang mengindikasikan energi yang tersisa di dalam baterai. Biasanya dinyatakan dalam bentuk persentase, dengan nilai mulai dari 0 hingga 100%. Beberapa metode dan model telah diusulkan untuk memperkirakan *SoC* baterai *lithium ion*, termasuk model *Susceptible, Infected, dan Recovered* (SIR) kompetitif dan penggunaan

sistem manajemen baterai (BMS). Estimasi *SoC* sangat penting untuk mengelola dan memanfaatkan energi yang tersimpan di dalam baterai secara efektif.

Metode estimasi status pengisian daya untuk baterai *lithium ion* yang disarankan berdasarkan model SIR kompetitif, yang menggunakan model matematika untuk memperkirakan *SoC* baterai. Selain itu, sebuah buku putih tentang *Li-Ion BMS* menunjukkan bahwa menggabungkan penghitungan *Coulomb* dan terjemahan tegangan dapat memberikan perkiraan *SoC* yang masuk akal, di mana penghitungan *Coulomb* melibatkan pengintegrasian arus ke dalam atau ke luar baterai, dan terjemahan tegangan menggunakan perubahan tegangan pada tingkat *SoC* yang berbeda untuk memperkirakan status pengisian daya. Selain itu, video tentang menghitung status pengisian daya sistem baterai *lithium ion* menggunakan Sistem Manajemen Baterai memberikan wawasan praktis tentang estimasi *SoC* dalam sistem baterai. Metode dan model ini memainkan peran penting dalam memastikan pemanfaatan baterai *lithium ion* yang efisien dan aman dalam berbagai aplikasi.

Kesimpulan untuk *SoC* baterai *lithium ion* adalah parameter kunci yang menunjukkan sisa energi dalam baterai. Berbagai metode estimasi, seperti model SIR yang kompetitif dan penggunaan sistem manajemen baterai, digunakan untuk menentukan *SoC* secara akurat, sehingga memungkinkan manajemen dan pemanfaatan energi yang efektif.

## 2.4 State of Health

*State of Health* (*SoH*) baterai *lithium ion* adalah parameter penting yang mencerminkan tingkat degradasi baterai dan masa pakai yang tersisa. Estimasi *SoH* yang akurat sangat penting untuk memastikan keselamatan operasional, mencegah kegagalan, dan mengoptimalkan manajemen baterai. Beberapa metode dan teknologi telah dikembangkan untuk estimasi *SoH*, termasuk pendekatan berbasis pembelajaran mesin, kerangka kerja pembelajaran mendalam, dan analisis berbasis data tentang resistansi internal baterai. Kemajuan ini bertujuan untuk memungkinkan estimasi *SoH* baterai tanpa memerlukan eksperimen degradasi tambahan, sehingga mengatasi sifat metode estimasi *SoH* tradisional yang memakan waktu dan sumber daya.

*SoH* baterai *lithium ion* dipengaruhi oleh berbagai faktor, termasuk kerusakan bagian dari keseluruhan elektrokimia, peningkatan resistansi internal, serta kapasitas dan pelemahan daya. Penuaan baterai dapat dipengaruhi oleh

reaksi elektrokimia, tekanan mekanis, dan suhu pengoperasian. Oleh karena itu, pengetahuan yang komprehensif tentang SoH baterai memainkan peran penting dalam manajemen kinerja tinggi, terutama dalam aplikasi kendaraan listrik.

Prediksi *SoH* yang akurat sangat penting untuk menjamin keamanan operasional dan menghindari kegagalan laten baterai *lithium ion*. Hal ini juga penting untuk mengoptimalkan manajemen baterai di berbagai aplikasi, termasuk kendaraan listrik. Pengembangan metode dan teknologi estimasi SoH yang canggih berkontribusi pada pemanfaatan baterai *lithium ion* yang aman dan efisien [16],[8],[17].

*State of Health* (SoH) baterai *lithium ion* dipengaruhi oleh berbagai faktor, termasuk:

1. Suhu.

Suhu merupakan faktor penting yang memengaruhi *SoH* baterai *lithium ion*. Suhu yang ekstrem, baik tinggi maupun rendah, mempercepat degradasi bahan baterai, yang menyebabkan penurunan kapasitas dan kinerja baterai secara keseluruhan. Kontrol dan pemantauan suhu yang akurat sangat penting untuk memastikan pengoperasian baterai *lithium ion* yang aman dan efisien.

2. Besaran arus.

Besarnya arus selama pengisian maupun pengosongan baterai memengaruhi *SoH* dari baterai.

3. Siklus pengisian dan pengosongan.

Jumlah siklus pengisian dan pengosongan, serta kedalaman pengosongan, dapat mengurangi *SoH* baterai.

4. Interval siklus.

Waktu antara siklus pengisian dan pengosongan dapat memengaruhi *SoH* baterai.

5. Tegangan pemutusan pengisian-pengosongan.

Tegangan saat baterai diisi dan dikosongkan dapat memengaruhi *SoH* baterai.

Faktor-faktor ini saling terkait sehingga menjadi satu kesatuan. Penelitian tentang faktor pengaruh status kesehatan baterai sedang dalam tahap penelitian kualitatif, dan analisis kuantitatif penuaan baterai dan penggabungan berbagai faktor adalah titik panas penelitian kesehatan baterai dan masa pakai baterai di masa depan [16][8][18].

## 2.5 Penelitian terkait

Pada bagian ini akan menyajikan informasi terkait akurasi prediksi dengan menggunakan dataset yang sama beserta penelitian terkait yang sejenis dan atau satu jenis topik mengenai *State of Charge* (SoC) dan *State of Health* (SoH).

Tabel 2.1. Accuracy using various algorithm with different author

Author	Algoritma	Evaluasi	Akurasi
Kenggle	Random Forest (RF)	MAE	1.50%
		R2 Square	0.97
	XGB	MAE	1.54%
		R2 Square	0.97
	LGBM	MAE	1.55%
		R2 Square	0.97
Pengliang Qin, Linhui Zhao, Zhiyuan Liu	Gradient Boosting (GB)	Accuracy	78.87%
Sungwoo Jo, Sunkyu Jung, Taemoon Roh	FNN	MAPE	2.03%
	CNN		3.10%
	LSTM		2.12%
Geng Wang, Zhiqiang Lyu, Xiaoyu Li	Random Forest (RF)	RMSE	1,0976%
		MAE	2,5925%

### 2.5.1 Random Forest, XGB, LGBM Algorithm

Berdasarkan hasil pembentukan model hingga akurasi model pada dataset sama, yakni *kaggle* terkait dengan kriteria *charge* didapatkan bahwa hasil akurasi seperti pada Tabel 2.2 untuk *State of Charge* (SoC) dengan persentase data *testing* 20% dan data *training* 80% [19]. Pada hasil yang tertera di Tabel 2.2, maka algoritma yang memiliki nilai akurasi paling kecil untuk *Mean Absolute Error* (MAE) adalah algoritma *Random Forest* dengan akurasi 1.50%.

Tabel 2.2. Accuracy of random forest, XGB, LGBM Algorithm

Algoritma	Evaluasi	Akurasi
Random Forest	Mean Absolute Error (MAE)	1.50%
	R2 Square	0.97
XGB	Mean Absolute Error (MAE)	1.54%
	R2 Square	0.97
LGBM	Mean Absolute Error (MAE)	1.55%
	R2 Square	0.97

### 2.5.2 Gradient Boosting Algorithm

Berdasarkan penelitian yang telah dilakukan menggunakan *gradient boosting algorithm*, penelitian ini mengekstraksi fitur *aging* berdasarkan bentuk *voltage curve*. Algoritma *gradient boosting* mampu mengukur fungsi *loss* dan mengatur *learning rate*. Metode Algoritma ini dapat berubah dan mengurangi kesalahan berdasarkan hasil prediksi dari Algoritma berbasis data yang sudah ada. Sehingga untuk membuat metode berbasis data yang berbasis *gradient boosting* ini memenuhi kebutuhan *incremental learning*, sebuah strategi pembelajaran peningkatan berbasis pohon Hoeffding dirancang. Hasil eksperimen menunjukkan bahwa fitur *aging* yang dipilih efektif. Pada Tabel 2.3, metode berbasis data yang berbasis *gradient boosting algorithm* yang diusulkan dapat meningkatkan akurasi prediksi Algoritma berbasis data yang dapat ditingkatkan hingga 78,87%. Dengan menggunakan 80% data untuk training, model memiliki cukup data untuk belajar dan mengurangi bias. Sedangkan penggunaan 20% data untuk testing membantu memastikan bahwa model tidak terlalu *overfitting* [5].

Tabel 2.3. State of health accuracy using gradient boosting algorithm

Algoritma	Fold	Akurasi
Gradient Boosting	10	78.87%

### 2.5.3 Feedforward Neural Networks (FNN), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) Algorithm

Berdasarkan penelitian yang telah dilakukan, hasil korelasi menunjukkan bahwa dataset berbasis *State of Charge (SoC)* yang diproses menggunakan metode

yang diusulkan memiliki korelasi yang lebih tinggi dengan kapasitas yang dapat digunakan dibandingkan dataset berbasis waktu pada umumnya. Selanjutnya, berbagai Algoritma *machine learning* seperti FNN, CNN, dan LSTM menggunakan kumpulan data berbasis SoC dan berbasis waktu. Hasilnya dapat terlihat di Tabel 2.4 menunjukkan bahwa model FNN paling sederhana menggunakan metode yang diusulkan memprediksi baterai SoH dengan akurasi yang sama dengan model kompleks seperti CNN dan LSTM. Nilai *Mean Absolute Percentage Error (MAPE)* model FNN dengan dataset berbasis SoC adalah 2,03 lebih rendah dari model CNN sebesar 3,10 dan model LSTM sebesar 2,12 dengan dataset berbasis waktu yang diuji pada data pelatihan kecil. Pada penelitian ini, terlihat bahwa *data training* dan *data testing* memiliki persentase yang berbeda mengikuti jumlah dataset yang ada pada tiap masing-masing baterai [6].

Tabel 2.4. MAPE accuracy using FNN, CNN, LSTM

Algoritma	Akurasi MAPE
Feedforward Neural Networks (FNN)	2.03%
Convolutional Neural Networks (CNN)	3.10%
Long Short Term Memory (LSTM)	2.12%

#### 2.5.4 Random Forest (RF), Extreme Learning Machine (ELM) Algorithm

Berdasarkan referensi yang menggunakan perbandingan algoritma *Random Forest (RF)* dan *Extreme Learning Machine (ELM)*, model *RF* dan *ELM* diteliti untuk estimasi *State of Charge (SoC)* pada baterai *BMW i3 EV*. Model *RF* mencapai kinerja yang lebih unggul dibandingkan metode *Extreme Learning Machine (ELM)* pada Tabel 2.5, dengan *Root Mean Squared Error (RMSE)* yang lebih rendah yaitu 5,9028% dibandingkan dengan 6,3127% untuk *ELM*, dan *Mean Absolute Error (MAE)* yang lebih rendah yaitu 4,4321% dibandingkan dengan 5,1112% untuk *ELM* pada pengujian validasi silang k lipatan sebanyak lima kali. Hal ini menunjukkan akurasi dan presisi yang lebih tinggi dari pendekatan *RF* yang diusulkan [20].

Tabel 2.5. State of charge with random forest and extreme learning machine

Algoritma	CV	RMSE	MAE
Random Forest (RF)	5	5.9028%	4.4321%
Extreme Learning Machine (ELM)		6.3127%	5.1112%

### 2.5.5 Random Forest Algorithm

Berdasarkan referensi yang didapatkan dengan menggunakan algoritma *Random Forest* dari perspektif kesalahan estimasi SoH pada Tabel 2.6, *MAE* dan *RMSE* maksimum untuk kedua baterai masing-masing adalah 2,5925 dan 1,0976. Selain itu, rata-rata *MAE* dan *RMSE* masing-masing adalah 1,8152 dan 0,7581. Pada penelitian ini dilakukan pembagian antara *data training* dan *data testing* [21].

Tabel 2.6. State of health with random forest

Algoritma	Evaluasi	Akurasi
Random Forest (RF)	Root Mean Squared Error (RMSE)	1,0976%
	Mean Absolute Error (MAE)	2,5925%

Berdasarkan referensi prediksi berbasis data dikembangkan dengan menggunakan data yang dikumpulkan dan solusi pembelajaran *Random Forest*. Pada Tabel 2.7, *estimator* yang dilakukan kajiannya menunjukkan akurasi *MAE SoH* sebesar 1,27% [22]. Penggunaan *decision trees* sangat membantu untuk investigasi mekanisme prediksi *SoH* dan mudah diterapkan. Algoritma *Random Forest* digunakan karena memberikan hasil yang sangat baik ketika menggunakan data yang luas, mudah diterapkan, dan menangani data yang heterogen dan hilang dengan baik.

Tabel 2.7. State of health with MAE random forest

Algoritma	Evaluasi	Akurasi
Random Forest	Mean Absolute Error (MAE)	1,27%

## 2.6 Decision Tree

Algoritma *Decision Tree* adalah metode pembuatan model prediktif dalam bentuk pohon keputusan. Pohon ini menggambarkan struktur keputusan hierarki, di mana setiap simpul dalam pohon merepresentasikan keputusan atau tes pada atribut tertentu, setiap cabang merepresentasikan hasil dari tes tersebut, dan setiap daun merepresentasikan nilai prediksi atau kelas target. Beberapa Algoritma populer untuk membangun *Decision Tree* melibatkan pembelajaran supervisi, yaitu *CART* (*Classification and Regression Trees*), *ID3* (*Iterative Dichotomiser 3*), dan *C4.5*.

Berikut adalah langkah-langkah umum dalam Algoritma *Decision Tree*, yaitu:

1. **Pemilihan Atribut (Feature Selection)**

Lakukan pemilihan atribut terbaik yang akan digunakan untuk membagi data menjadi *subset*. Atribut yang terbaik dipilih berdasarkan kriteria seperti *Information Gain*, *Gain Ratio*, *Gini Index* yang bergantung pada Algoritma yang digunakan.

2. **Pembagian Data (Data Splitting)**

Gunakan atribut yang dipilih untuk membagi data menjadi *subset* yang lebih kecil. Setiap cabang dalam pohon keputusan mewakili nilai yang berbeda dari atribut tersebut.

3. **Rekursif (Recursion)**

Lakukan langkah 1 dan 2 secara *rekursif* untuk setiap *subset* data yang dihasilkan. Proses ini terus berlanjut hingga kriteria berhenti *rekursif* terpenuhi, seperti mencapai batasan kedalaman maksimum, mencapai jumlah sampel minimum di sebuah simpul, atau tidak ada peningkatan signifikan dalam kriteria pemilihan atribut.

4. **Penetapan Label (Label Assignment)**

Setiap daun dalam pohon keputusan menetapkan label atau nilai prediksi untuk instansi data yang mencapai daun tersebut. Misalnya, jika membangun pohon keputusan untuk klasifikasi, label di daun mewakili kelas yang diprediksi.

5. **Pruning (Pemangkasan)**

Pemangkasan dilakukan untuk mengurangi kompleksitas pohon dan mencegah *overfitting*. Jenis pemangkasan ada dua, yaitu *pre-pruning* (menghentikan pertumbuhan pohon lebih awal berdasarkan kriteria tertentu) dan *post-pruning* (menghapus atau menggabungkan simpul setelah pohon selesai dibangun).

Algoritma ini bekerja dengan mencari pemisahan yang paling informatif pada setiap langkah, sehingga meminimalkan ketidakpastian atau meningkatkan homogenitas setiap *subset* data. Pemilihan atribut yang baik dan *pruning* yang tepat, penting untuk membuat model *Decision Tree* yang *general* dan tidak *overfit*.

## 2.7 Random Forest (RF)

*Random Forest* adalah proses *ensemble machine learning* yang membuat banyak pohon pengambilan keputusan selama pelatihan dan menghasilkan kelas yang di rata-ratakan atau dipilih oleh setiap pohon. *Random Forest* digunakan dalam pemilihan variabel, klasifikasi dan regresi. Hasil dari *Random Forest* adalah membuat pohon keputusan secara acak dari kumpulan data pelatihan. Selama fase pelatihan, pohon keputusan akan dievaluasi. Pemilihan fitur memiliki peran penting karena *Random Forest* mencoba memilih fitur yang paling penting saat mengambil keputusan [23]. *Random Forest* memiliki beberapa parameter, salah satu parameter nya berfokus pada jumlah keputusan untuk kumpulan data pelatihan. Parameter lainnya berfokus pada jumlah fitur. Secara internal, *Random Forest* menerapkan validasi silang dan memiliki konsekuensi prediktif yang sesuai untuk data yang rumit dan non-linier.

### Prediksi regresi

Setiap pohon *Decision Tree* memberikan prediksi untuk data baru. Prediksi dari semua pohon *Decision Tree* kemudian digabungkan untuk menghasilkan prediksi akhir.

$$\text{Prediksiakhir} = \text{Rata-rata}(\text{Prediksi}pohon1, \text{Prediksi}pohon2, \dots, \text{Prediksi}pohonN) \quad (2.1)$$

### Hyperparameter Tuning

*Random Forest* memiliki beberapa *hyperparameter* yang dapat di *tuning* untuk meningkatkan performa model, seperti jumlah pohon *Decision Tree*, kedalaman maksimum pohon *Decision Tree*, ukuran *subset* data, ukuran *subset* fitur. Pada penelitian ini, *hyperparameter* yang digunakan untuk *tuning* model adalah jumlah pohon (*n estimator*), dan *random state*.

### Penerapan Algoritma

*Dataset* dari baterai *lithium ion* pada kendaraan listrik yang terdapat pada *kaggle*, terukur selama proses pengisian dan pengosongan. Data ini mencakup tegangan, arus, hingga suhu dari baterai.

#### 1. Persiapan data

Menyortir data, menangani *outlier*, dan membagi data menjadi set pelatihan dan pengujian.

2. Pemilihan fitur  
Dapat memilih fitur yang relevan seperti tegangan terminal, arus, suhu, dan waktu pengisian/pengosongan.
3. Pemilihan dan pengaturan Algoritma  
Dapat menggunakan *scikit learn* dan mengatur parameter seperti jumlah pohon dan kedalaman pohon.
4. Pelatihan model  
Dapat melatih model *Random Forest* dengan set pelatihan.
5. Evaluasi Model  
Dapat menggunakan set pengujian untuk mengevaluasi keakuratan *prediksi State of Charge* dan *State of Health*.
6. Prediksi  
Dapat memasukkan data dari pengisian/pengosongan baru ke model untuk memprediksi *prediksi State of Charge* dan *State of Health* baterai secara *real time*.

### **Penggunaan Algoritma**

Proses pembentukan setiap pohon keputusan dalam *Random Forest* adalah sebagai berikut:

1. Pemilihan Sampel Acak: Dari dataset pelatihan yang tersedia, sampel data diambil secara acak dengan penggantian (*bootstrap*) untuk membentuk dataset yang berukuran sama dengan dataset pelatihan asli.
2. Pemilihan Fitur Acak: Dari jumlah fitur yang tersedia, *subset* fitur diambil secara acak untuk digunakan dalam membangun pohon keputusan. Biasanya, jumlah fitur yang diambil setiap kali jauh lebih kecil dari jumlah total fitur yang tersedia.
3. Membangun Pohon Keputusan: Dengan menggunakan sampel data dan *subset* fitur yang telah dipilih, pohon keputusan dibangun dengan menggunakan Algoritma seperti *ID3*, *C4.5*, *CART*, atau *Random Forest*. Pohon ini dibentuk dengan melakukan pemisahan data berdasarkan fitur-fitur yang paling informatif, dengan tujuan untuk meminimalkan keragaman (*variance*) dalam setiap simpul pohon.

4. Pembentukan *ensemble*:

Langkah-langkah 1 hingga 3 diulang beberapa kali untuk membentuk kumpulan pohon keputusan. Setiap pohon keputusan dalam *Random Forest* memberikan suara atau memberikan prediksi terhadap kelas atau nilai regresi yang diinginkan.

5. Keputusan Mayoritas:

Prediksi akhir dalam *Random Forest* diperoleh dengan mengambil mayoritas suara atau rata-rata prediksi dari semua pohon keputusan dalam *ensemble*.

