

BAB II

LANDASAN TEORI

2.1 Dasar Teori

2.1.1 *Capacity Factor* (Faktor Kapasitas)

Capacity Factor (CF) sebuah pembangkit listrik adalah perbandingan antara kapasitas rata-rata dalam *megawatt* (MW) produksi selama periode tertentu terhadap kapasitas terpasang (Kementerian ESDM, 2017). Dalam pengoperasian sistem tenaga listrik Jawa Bali, terminologi CF ini lebih dikhususkan pada *Net Capacity Factor*. *Net Capacity factor* (NCF) adalah rasio antara total produksi netto dengan daya mampu netto unit pembangkit dikali dengan jam periode tertentu (umumnya periode 1 tahun, 8760 atau 8784 jam) (PT PLN (Persero), 2017). Rumus untuk menghitung CF ditunjukkan pada persamaan (2.1).

$$\text{Net Capacity Factor} = \frac{\text{Produksi Netto}}{\text{Period Hours} \times \text{Daya Mampu Netto}} \times 100\% \quad (2.1)$$

dengan *Period Hours* adalah interval waktu yang akan diukur, produksi netto adalah energi netto yang disalurkan sebuah pembangkit, dan Daya Mampu Netto adalah Daya mampu maksimal sebuah pembangkit yang sudah dikurangi pemakaian sendiri dan susut trafo. Contoh perhitungan sederhana CF sebuah pembangkit dapat dijelaskan sebagai berikut.

Sebuah pembangkit mempunyai Daya Mampu Netto 100 MW. Jumlah energi yang disalurkan oleh pembangkit tersebut selama bulan Desember 2023 adalah sebesar 70.000 MWh. Maka NCF pembangkit tersebut adalah sebagai berikut.

$$NCF = \frac{70.000}{31 \times 24 \times 100} \times 100\% = \frac{70.000}{74.400} \times 100\% = \mathbf{94,086\%}$$

Terdapat beberapa faktor yang berpotensi mempengaruhi CF sebuah pembangkit listrik. Faktor tersebut dapat berupa faktor internal maupun faktor eksternal. Faktor internal berasal dari internal pembangkit, sedangkan faktor eksternal berasal dari luar pembangkit. Faktor internal yang mempengaruhi CF di antaranya adaah *Scheduled Outage Factor* (SOF) dan *Equivalent Forced Outage Factor* (EFOR). Adapun faktor eksternal yang mempengaruhi CF adalah beban

(permintaan) listrik suatu area, daya mampu netto pembangkit suatu area, *merit order*, dan *incremental fuel cost*. Berikut ini adalah penjelasan dari masing-masing faktor internal dan faktor eksternal tersebut.

a. *Scheduled Outage Factor (SOF)*

Scheduled Outage Factor (SOF) adalah rasio dari jumlah jam unit pembangkit keluar terencana (*planned outage* dan *maintenance outage*) terhadap jumlah jam dalam satu periode. Besaran ini menunjukkan prosentase kondisi unit pembangkit akibat pelaksanaan pemeliharaan, inspeksi dan *overhaul* pada suatu periode tertentu (PT PLN (Persero), 2017). Rumus untuk menghitung SOF ditunjukkan pada persamaan (2.2).

$$SOF = \frac{POH + MOH}{PH} \quad (2.2)$$

dimana POH adalah *planned outage hour*, MOH adalah *maintenance outage hour*, dan PH adalah *period hour*. Contoh perhitungan SOF adalah sebagai berikut.

Sebuah Pembangkit Listrik pada bulan Desember 2023 melaksanakan pekerjaan pemeliharaan tahunan selama 200 jam. Bulan Desember memiliki period hours sebesar 24 x 31 jam, yaitu 744 jam. Pekerjaan tersebut dilakukan dengan keluar dari jaringan (*grid*) dan sudah direncanakan setahun sebelumnya (*planned outage*). Pada bulan tersebut, tidak ada pekerjaan pemeliharaan terencananya yang direncanakan sebulan atau seminggu sebelumnya (*maintenance outage*). Perhitungan SOF nya adalah sebagai berikut

$$SOF = \frac{POH + MOH}{PH}$$

$$SOF = \frac{200 + 0}{744}$$

$$SOF = \frac{200}{744} = 0,2688 = \mathbf{26,88\%}$$

b. Equivalent Forced Outage Rate (EFOR)

Equivalent Forced Outage Rate (EFOR) adalah *Forced Outage Rate* (FOR) yang telah memperhitungkan dampak dari derating pembangkit. FOR didefinisikan sebagai jumlah jam unit pembangkit dikeluarkan dari sistem (keluar paksa) dibagi jumlah jam unit pembangkit dikeluarkan dari sistem ditambah jumlah jam unit pembangkit beroperasi, yang dinyatakan dalam persen (PT PLN (Persero), 2017). Rumus untuk menghitung EFOR ditunjukkan pada persamaan (2.3).

$$EFOR = \frac{FOH + EFDH}{FOH + SH + Synchr. Hrs. + EFDHRS} \times 100\% \quad (2.3)$$

dimana FOH adalah *Forced Outage Hour*, EFDH adalah *Equivalent Forced Outage Hour*, SH adalah *Service Hour*, dan EFDHRS adalah *Equivalent Forced Derated Hours during Reserve Shutdown*. Contoh perhitungan EFOR sebuah pembangkit adalah sebagai berikut.

Sebuah pembangkit berkapasitas 300 MW pada bulan Desember 2023 mengalami gangguan yang membuat pembangkit tersebut harus keluar jaringan selama 10 jam. Selama bulan tersebut, pembangkit tersebut juga mengalami penurunan kapasitas 100 MW sehingga menjadi 200 MW selama 30 jam. Pembangkit ini diminta terus beroperasi oleh *dispatcher* (tidak mengalami *Reserve Shutdown*). Pembangkit ini tidak berperan sebagai kondensator sinkron sehingga *synchronous hours*-nya 0. Perhitungan EFOR-nya adalah sebagai berikut

$$FOH = 10 \text{ jam}$$

$$Service Hour (SH) = (31 \times 24) - 10 = 744 - 10 = 734 \text{ jam}$$

$$EFDH = \frac{300 - 200}{300} \times 30 \text{ jam} = 10 \text{ jam}$$

$$EFOR = \frac{FOH + EFDH}{FOH + SH + Synchr. Hrs. + EFDHRS} \times 100\%$$

$$EFOR = \frac{10 + 10}{10 + 734 + 0 + 0} \times 100\% = 2,688 \%$$

c. Beban (Permintaan) Listrik

Beban (permintaan) listrik adalah jumlah permintaan tenaga listrik pada suatu area atau wilayah. Beban listrik ini dinyatakan dengan satuan MW (Megawatt). Beban listrik dibedakan menjadi dua, yaitu beban listrik dasar dan beban listrik puncak (Rooij, 2016). Dalam penelitian ini, beban listrik yang dimaksudkan sebagai variabel independen adalah beban listrik puncak. Beban listrik puncak atau beban puncak adalah beban listrik tertinggi yang harus dipenuhi oleh sistem tenaga listrik dalam suatu periode tertentu (Gonen, 2015). Beban puncak biasanya dibagi menjadi tahunan, harian, dan musiman (Torriti, 2015). Bagi sebuah perusahaan penyedia listrik, titik beban puncak adalah jam tertentu di mana terjadi konsumsi listrik tertinggi. Pada periode beban puncak, biasanya terjadi kombinasi antara konsumsi listrik perkantoran dan rumah tangga (Landsberg & Stewart, 1980).

d. Daya Mampu Netto Jawa-Bali

Daya Mampu Netto (DMN) adalah daya mampu netto Pembangkit sesuai kontrak PJBTL antara perusahaan pembangkit dengan PLN dan dapat direvisi dengan surat Deklarasi dari Divisi Operasi Regional PLN (PT PLN (Persero), 2017). DMN area artinya Jumlah DMN semua pembangkit yang terhubung dalam suatu area atau wilayah. DMN Jawa Bali artinya adalah jumlah DMN semua pembangkit yang terhubung dalam sistem Jawa Bali.

e. Merit order

Merit order adalah metode paling sederhana dalam *economic dispatch* yang dilakukan dengan mengurutkan pembangkit dari yang paling rendah sampai yang paling tinggi biaya operasinya. Pembangkit yang paling rendah biaya operasinya (Rp/kWh) dianggap sebagai pembangkit yang paling ekonomis (Mariani *et al.*, 2021).

f. Incremental fuel cost

Incremental fuel cost (IFC) didefinisikan sebagai perubahan biaya bahan bakar yang terjadi bila terjadi perubahan daya listrik yang dibangkitkan. Perubahan jumlah bahan bakar yang terjadi karena perubahan keluaran

didefinisikan sebagai *Incremental Fuel Rate* (IFR) (Delima & Syafii, 2016).
 .Persamaan IFR ditunjukkan pada persamaan (2.4).

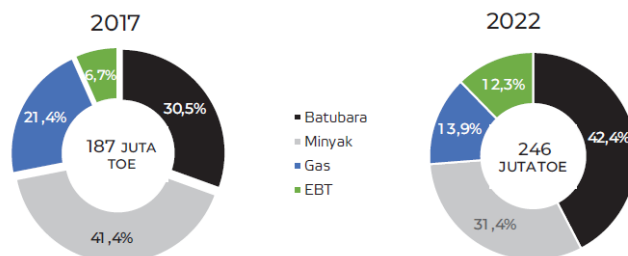
$$IFR = \frac{\Delta F}{\Delta P} \quad (2.4)$$

dimana ΔF adalah perubahan jumlah bakar dan ΔP adalah perubahan jumlah daya pembangkitan. IFC dapat dihitung sebagai perkalian antara IFR dan biaya bahan bakar (*fuel cost*) sesuai dengan persamaan (2.5).

$$IFC = IFR \times \text{fuel cost} \quad (2.5)$$

2.1.2 Energi Primer

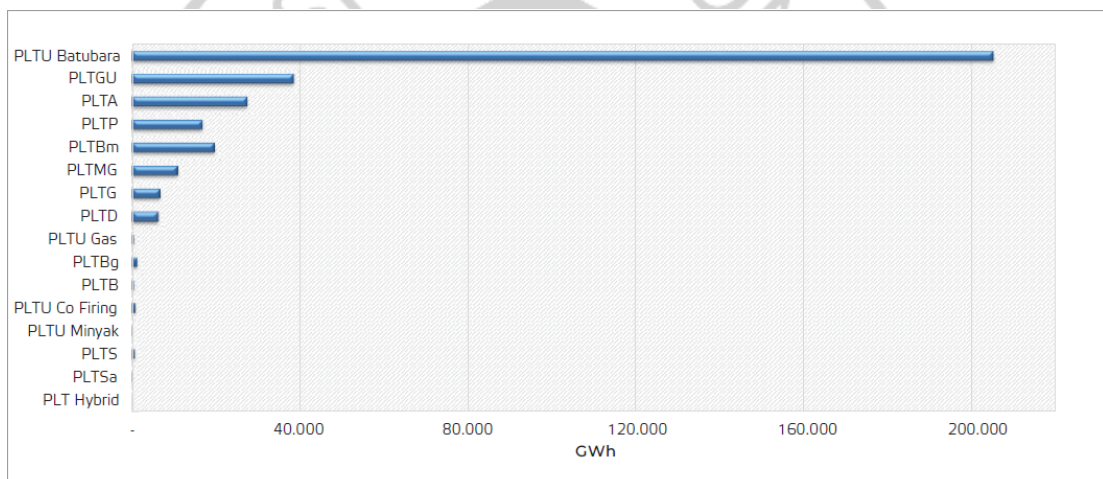
Energi merupakan faktor produksi yang esensial dalam proses produksi. Semua produksi melibatkan transformasi atau pergerakan material melalui beberapa tahapan yang keseluruhan proses tersebut memerlukan energi. Energi tidak hanya dipandang sebagai barang konsumsi semata, namun juga sebagai input yang penting bagi pengembangan serta kemajuan teknologi yang berperan signifikan bagi pembangunan ekonomi (Kartiasih *et al.*, 2012). Bentuk energi ada dua macam, yaitu energi primer dan energi final. Energi primer adalah energi yang diberikan oleh alam dan belum mengalami proses pengolahan lebih lanjut (Pemerintah RI, 2014). Energi primer memiliki peran sebagai bahan bakar awal untuk kemudian diolah menjadi bentuk energi akhir. Yang termasuk dalam kategori sumber energi primer adalah minyak bumi, gas bumi, batu bara, tenaga air, dan panas bumi. Grafik bauran energi primer di Indonesia dapat ditunjukkan pada gambar 2.1



Gambar 2.1 Bauran Energi Primer Indonesia Tahun 2017 dan Tahun 2022
 Sumber : (Kementerian ESDM, 2023)

Berdasarkan gambar 2.1 terjadi pergeseran peringkat pertama bauran energi primer di Indonesia antara tahun 2017 dan tahun 2022. Pada tahun 2017, minyak bumi masih mendominasi bauran energi primer di Indonesia yaitu sebesar 41,4%. Dominasi ini kemudian digeser oleh batubara pada tahun 2022 dengan bauran sebesar 42,4%. Bauran EBT mengalami peningkatan dari 6,7% pada tahun 2017 menjadi 12,3% pada tahun 2022.

Energi primer yang bersumber dari batubara saat ini masih mendominasi dalam pembangkit listrik di Indonesia. Produksi listrik per per jenis pembangkit tahun 2022 dapat dilihat pada Gambar 2.2



Gambar 2.2 Produksi Listrik per Jenis Pembangkit Tahun 2022

Sumber : (Kementerian ESDM, 2023)

Berdasarkan gambar 2.2, produksi listrik yang bersumber dari batubara pada tahun 2022 adalah sebesar 205,3 GWh atau sebesar 62% (Dewan Energi Nasional, 2024). Produksi listrik dari PLTU batubara jauh lebih tinggi dari produksi listrik dari PLTGU yang hanya sebesar 38,5 GWh atau 12%.

Untuk memastikan pasokan listrik dari pembangkit listrik berbasis batubara tidak terhambat, maka diperlukan perencanaan energi primer yang akurat. Dalam Sistem Jawa-Bali, proses perencanaan batubara dilakukan tahunan dan bulanan. Perencanaan tahunan dilakukan dalam forum Rapat ROT (Rencana Operasi Tahunan). ROT merupakan perencanaan operasi dengan jangka waktu satu tahun meliputi rencana pemeliharaan unit-unit pembangkit yang memerlukan persiapan sejak tahun sebelumnya karena pengadaan suku cadangnya memerlukan waktu

lama. ROT mencakup estimasi alokasi bulanan produksi neto pembangkit, prakiraan beban sistem, rencana *outage* unit pembangkit, rencana *outage* transmisi, prediksi *fuel mix*, CF pembangkit, dan susut transmisi, serta data-data lainnya (Kementerian ESDM, 2020).

Perencanaan energi bulanan dilakukan dalam forum RAE (Rapat Alokasi Energi) dan Rapat Koordinasi Batubara. Forum RAE dihadiri oleh PT PLN (Persero) dan perusahaan pembangkit listrik. Dalam RAE disampaikan ROB (Rencana Operasi Bulanan). Rencana operasi bulanan merupakan rencana operasi seluruh unit pembangkit dan transmisi termasuk rencana keluaran pembangkit (Kementerian ESDM, 2020). Rencana keluaran pembangkit ini berupa rencana CF pembangkit bulan depan. Rencana pembangkitan harus dapat memenuhi prakiraan beban sistem dengan biaya variabel minimal dengan tetap memperhatikan kriteria keandalan, ekonomis, dan kualitas Sistem Tenaga Listrik. Forum RAE ini masih belum secara spesifik membahas alokasi kebutuhan batubara unit pembangkit. Forum yang lebih spesifik membahas alokasi batubara pembangkit di lingkungan PLN *Group* adalah forum Rakor (Rapat Koordinasi) Batubara. Forum ini dihadiri oleh PLN Energi Primer Indonesia (PLN EPI) yang merupakan *subholding* PLN yang mengelola masalah energi primer dan pembangkit-pembangkit listrik di lingkungan PLN *Group*. Dalam forum ini, tiap pembangkit diminta untuk menyampaikan alokasi jumlah kedatangan tongkang batubara pada bulan depan.

2.1.3 Prediksi

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahan (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil (Wantono, 2014). Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha mencari jawaban sedekat mungkin yang akan terjadi (Herdianto, 2013). Pengertian prediksi sama dengan ramalan atau perkiraan (KBBI, 2014).

Prediksi bisa berdasarkan metode ilmiah ataupun subjektif belaka. Prediksi cuaca selalu berdasarkan data dan informasi terbaru yang didasarkan pengamatan

termasuk oleh satelit. Begitupun prediksi gempa, gunung meletus ataupun bencana secara umum. Namun, prediksi seperti pertandingan sepakbola, olahraga, dan lain sebagainya umumnya berdasarkan pandangan subyektif dengan sudut pandang sendiri yang memprediksinya (Wantono, 2014)

2.1.4 Regresi

Regresi adalah studi tentang keterkaitan. Regresi digunakan untuk menjawab pertanyaan tentang bagaimana satu atau beberapa variabel mempengaruhi sebuah variabel tertentu (Weisberg, 2014). Analisis regresi adalah kajian terhadap hubungan satu variabel yang disebut sebagai variabel yang diterangkan (*dependent variable*) dengan satu atau dua variabel yang menerangkan (*explanatory variable*) (Gujarati, 2013). Variabel dependen adalah variabel yang dijelaskan atau dipengaruhi oleh variabel independen. Variabel dependen disebut juga variabel yang diduga sebagai akibat (*presumed effect variable*). Variabel dependen juga dapat disebut sebagai variabel konsekuensi (Supomo & Indriantoro, 2014). Variabel independen adalah variabel yang menjelaskan atau mempengaruhi variabel yang lain. Variabel independen disebut pula variabel yang diduga sebagai sebab (*presumed cause variable*). Variabel independen juga dapat disebut sebagai variabel yang mendahului (Supomo & Indriantoro, 2014).

Dalam mengevaluasi sebuah regresi, terdapat beberapa nilai parameter yang harus diperhatikan, yaitu koefisien determinasi (R^2), hasil uji F, dan hasil uji T. Hasil uji F dan hasil uji T dapat digunakan sebagai kriteria untuk pemilihan variabel independen dalam sebuah regresi (Kutner *et al.*, 2004)

Koefisien Determinasi (R^2) merupakan ukuran statistik yang menunjukkan pengaruh yang diberikan variabel bebas / independent (X) terhadap variabel terikat atau dependent (Y). Nilai dari koefisien determinasi adalah $0 \leq R^2 \leq 1$. Semakin besar nilai R^2 atau semakin mendekati nilai 1 maka semakin kuat hubungan antara variabel bebas (X) dengan variabel terikat (Y). Sebaliknya apabila koefisien determinasinya bernilai nol atau mendekati 0, maka semakin lemah atau tidak terdapat hubungan antara variabel bebas (X) dengan variabel terikat (Y) (Weisberg, 2014). Persamaan koefisien determinasi ditunjukkan pada persamaan (2.6).

$$R^2 = 1 - \frac{RSS}{\sum(y_i - \bar{y})^2} = 1 - \frac{\sum(y_i - f(x_i))^2}{\sum(y_i - \bar{y})^2} \quad (2.6)$$

dengan RSS adalah jumlah kuadrat *residual* (*residual sum squares*), y_i adalah nilai aktual ke - i , \bar{y} adalah rerata nilai aktual, dan $f(x_i)$ adalah nilai prediksi ke- i

Proses pemilihan variabel independen dalam sebuah regresi dapat menggunakan *forward selection* maupun *backward elimination* (Kutner *et al.*, 2004). Proses *forward selection* diawali dengan model yang hanya ada *intercept* tanpa ada variabel independen. Proses selanjutnya adalah menambahkan beberapa variabel independen dan setiap penambahan variabel independen dihitung jumlah galat kuadrat nya. Model yang memiliki jumlah galat kuadrat terkecil dipilih sebagai model terbaik (James *et al.*, 2023). Sebaliknya, pada metode *backward elimination*, proses dimulai dengan memasukkan semua variabel independen dalam model. Tahapan selanjutnya adalah dilakukan uji F untuk menghitung F-statistik. Hasil pengujian F dapat digunakan sebagai dasar bahwa terdapat minimal 1 variabel independen yang berpengaruh terhadap variabel dependen. Selanjutnya dilakukan pengujian t untuk memperoleh nilai *p-value* masing-masing variabel. Variabel dengan *p-value* melebihi suatu batasan akan dikeluarkan dari model (James *et al.*, 2023).

Pengujian F merupakan salah satu uji hipotesis yang bertujuan untuk apakah ada hubungan antara variabel independen dan variabel dependen (Kutner *et al.*, 2004). Pada pengujian ini terdapat dua hipotesis, yaitu H_0 (*null hypothesis*) dan H_a (*alternative hypothesis*). H_0 menyatakan bahwa tidak ada hubungan antara variabel independen dan variabel dependen, sedangkan H_a menyatakan bahwa terdapat minimal satu variabel independen yang mempengaruhi variabel dependen (James *et al.*, 2023). Uji F dilakukan dengan membandingkan nilai F_{rasio} dengan nilai F_{tabel} (Sparks *et al.*, 1987). Jika $F_{rasio} > F_{tabel}$ maka H_0 ditolak atau artinya terdapat pengaruh minimal 1 variabel independen terhadap variabel dependen. sedangkan jika $F_{rasio} < F_{tabel}$, maka H_0 diterima, artinya tidak terdapat pengaruh variabel independen terhadap variabel dependen. Nilai F_{tabel} dapat dilihat pada tabel

distribusi F untuk tingkat kepercayaan, jumlah variabel independen, dan jumlah observasi data tertentu (Sparks *et al.*, 1987). Rumusan F_{rasio} ditunjukkan pada persamaan (2.7)

$$F = \left(\frac{R^2}{1 - R^2} \right) \left(\frac{N - k - 1}{k} \right) \quad (2.7)$$

dengan F adalah F_{rasio} , R^2 adalah koefisien determinasi, N adalah jumlah observasi data, dan k adalah jumlah variabel independen

Pengujian-t bertujuan untuk mengetahui ada atau tidaknya pengaruh parsial (sendiri) yang diberikan variabel independen secara individu terhadap variabel dependen (Sudjana, 2005). Uji t dilakukan dengan membandingkan nilai $t_{statistik}$ dengan nilai t_{tabel} . Jika $t_{statistik} > t_{tabel}$ maka H_0 ditolak atau artinya terdapat pengaruh variabel independen terhadap variabel dependen. sedangkan jika $t_{statistik} < t_{tabel}$, maka H_0 diterima, artinya tidak terdapat pengaruh variabel independen terhadap variabel dependen. Nilai t_{tabel} dapat dilihat pada tabel distribusi t untuk tingkat kepercayaan dan jumlah observasi data tertentu (James *et al.*, 2023). Persamaan $t_{statistik}$ ditunjukkan pada persamaan (2.8).

$$t = \frac{\widehat{\beta}_1}{SE(\widehat{\beta}_1)} \quad (2.8)$$

dengan t adalah $t_{statistik}$, $\widehat{\beta}_1$ adalah estimasi koefisien variabel independen, dan $SE(\widehat{\beta}_1)$ adalah *standard error* untuk estimasi koefisien variabel independen.

2.1.5 Machine Learning

Machine Learning adalah ilmu pengembangan algoritma dan model secara statistik yang digunakan sistem komputer untuk menjalankan tugas tanpa instruksi eksplisit, mengandalkan pola serta inferensi sebagai gantinya (Sen, 2021). *Machine Learning* dapat didefinisikan sebagai bidang ilmu komputer yang memberikan komputer kemampuan untuk belajar tanpa deprogram secara ekspilisit (Samuel, 1959). Metode machine learning secara umum dibagi menjadi tiga kategori, yaitu

supervised learning, *unsupervised learning*, dan *reinforcement learning* (Somvanshi & Chavan, 2016).

Metode *supervised learning* menganalisis dan mempelajari data latih dan membangkitkan sebuah fungsi yang dapat digunakan untuk memetakan data yang baru. Algoritma yang termasuk kedalam teknik *supervised learning* diantaranya *Decision Tree*, *K-Nearest Neighbor* (KNN), *Naive Bayes*, Regresi, dan *Super Vector Machine* (Pamungkas *et al.*, 2020). Metode *unsupervised learning* menggunakan teknik menarik kesimpulan berdasarkan dataset yang merupakan input dari *labeled response* (Nurhayati *et al.*, 2019). Beberapa algoritma dalam *unsupervised learning* diantaranya DBSCAN, *Fuzzy C-Means*, *K-Means*, dan *Self Organizing Map* (Retnoningsih & Pramudita, 2020)

Metode *reinforcement learning* berasal dari teori belajar hewan. Pembelajaran ini tidak memerlukan pengetahuan sebelumnya, dapat secara mandiri mendapatkan kebijakan opsional dengan pengetahuan yang diperoleh melalui coba-coba dan terus berinteraksi dengan lingkungan yang dinamis. Beberapa algoritma *reinforcement learning* di antaranya adalah *sarsa*, *temporal difference*, *Q-learning* dan *function approximation* (Wang & Zhan, 2011)

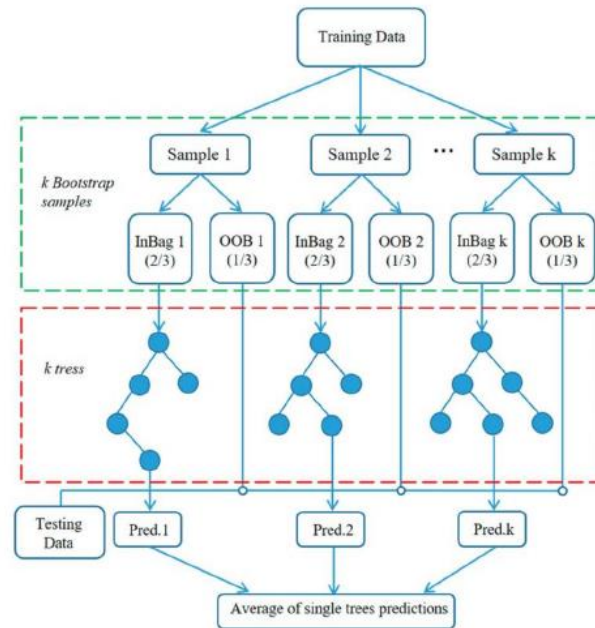
2.1.6 *Random forest regression*

Random forest adalah algoritma *machine learning* yang termasuk dalam teknik *supervised learning* yang menggabungkan keluaran dari beberapa *decision tree* untuk mencapai suatu hasil. *Random forest* dapat digunakan untuk regresi dan klasifikasi. *Random forest regression* dibentuk dari banyak *tree* (pohon) yang bergantung pada vector acak sehingga *predictor tree* mengambil nilai numerik sebagai lawan dari kelas label (Jin *et al.*, 2020).

Random forest merupakan metode yang diperkenalkan oleh Breiman sebagai pengembangan dan kombinasi dari banyak *decision tree*. Jika pada *Decision Tree* merupakan pohon klasifikasi tunggal, maka pada *random forest* dibuat banyak pohon untuk menentukan hasil prediksinya. Kombinasi dari *bootstrap aggregating* dan *random feature selection* pada *random forest* dapat digunakan untuk mengurangi masalah *overfitting* pada data *train* yang kecil (Breiman, 2001).

Algoritma *Random forest regression* (RFR) merupakan sebuah metode pembelajaran regresi bersifat *ensemble* yang diusulkan oleh Leo Breiman (Breiman, 2001). *Ensemble* sendiri merupakan model yang terdiri dari beberapa algoritma, sehingga dalam *Random Forest* ini terkandung beberapa *decision tree* dengan distribusi yang sama dan dengan kondisi tidak berkorelasi satu sama lain untuk membangun *forest* atau "hutan" dengan tujuan melatih dan memprediksi data sampel (Zhang *et al.*, 2021). *Random forest regression* dapat mengandung ratusan atau bahkan ribuan *Decision Tree* yang bertindak sebagai fungsi regresi sendiri. Setiap *tree* di RFR ditanam dengan subset prediktor acak maka dari itu algoritma ini disebut hutan 'acak' (Zhang *et al.*, 2021).

Decision Tree atau yang disebut juga sebagai *Classification and Regression Tree* (CART) adalah sebuah model statistik nonparametrik yang dapat menggambarkan hubungan antara variabel. Variabel di sini merupakan variabel respon (dependen) dengan satu atau lebih variabel prediktor (independen). Setiap pohon keputusan terdiri dari simpul keputusan atau internal node dan simpul daun atau *leaf*. Setiap node keputusan mengevaluasi setiap sampel variabel atau atribut dan setiap cabangnya merupakan hasil dari pengujian tersebut, sementara itu node terluar yaitu daun menjadi labelnya. Berbeda dari hasil akhir *Random Forest Classifier* yang mengambil *Majority Votes* dari berbagai *Decision Tree* sebagai hasilnya, hasil akhir dari *Random forest regression* ini merupakan nilai rata-rata dari keluaran semua *Decision Tree* yang telah dibangun (Li *et al.*, 2018). Gambar 2.3 akan menunjukkan cara kerja dari *Random forest regression*.



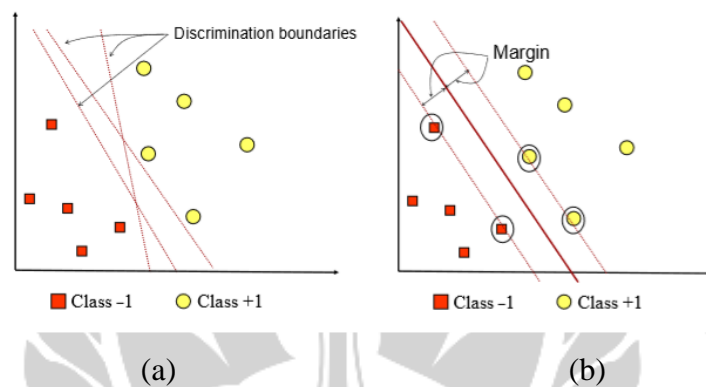
Gambar 2.3 Diagram Alur *Random forest regression*
 Sumber : (Rodriguez-Galiano *et al.*, 2015)

Seperti yang terlihat pada Gambar 2.3, untuk mendapatkan model *ensemble* dengan kemampuan generalisasi yang kuat, *base learner* atau pohon regresi di dalam model harus dibuat seunik mungkin. Oleh karena itu, digunakan sebuah metode bernama *Bagging (bootstrap aggregating)* untuk melakukan *resampling* pada data. Untuk kumpulan data yang berisi k sampel, sampel secara acak diambil ke dalam kumpulan sampel. Setelah itu, sampel dikembalikan ke data awal, yang memungkinkan sampel untuk dipilih pada pengambilan sampel berikutnya. Jadi, setelah k operasi pengambilan sampel acak, diperoleh satu set yang berisi k sampel yang nantinya akan digunakan untuk melatih setiap *Decision Tree* berdasarkan setiap set sampel. Terakhir, akan diambil nilai rata-rata dari setiap regresi *Decision Tree*.

2.1.7 *Support vector regression*

Support vector regression (SVR) adalah penerapan dari *Support Vector Machines (SVM)* dalam analisis regresi (Bi *et al.*, 2011). Ide dasar SVM adalah memetakan data pelatihan dari ruang input ke ruang fitur berdimensi lebih tinggi melalui fungsi dan kemudian membangun *hyperplane* pemisah dengan margin

maksimum di ruang fitur (Wu *et al.*, 2004). Secara teoritik *Support Vector Machine* dikembangkan untuk menyelesaikan permasalahan klasifikasi pada dua kelas dengan mencari *hyperplane* terbaik. *Hyperplane* merupakan fungsi untuk memisahkan antara dua kelas pada *input space*, sehingga dari data yang tersebar dapat dilakukan klasifikasi dan analisis regresi. Vapnik menerangkan bahwa setiap permasalahan dapat dimodelkan dengan menggunakan SVM (Naufal, 2017) . Ilustrasi SVM dapat dilihat pada Gambar 2.4.



Gambar 2.4 Ilustrasi *Support Vector Machine*
 Sumber: (Naufal, 2017)

Gambar 2.2 (a) menunjukkan alternatif garis pemisah (*discrimination boundaries*) dimana *pattern* yang tergabung pada class -1 disimbolkan dengan warna merah (kotak), sedangkan *pattern* pada class +1, disimbolkan dengan warna kuning (lingkaran). Sedangkan pada Gambar 3.1 (b) diperlihatkan bahwa terdapat garis *hyperplane* yang tepat berada diantara dua buah kelas. Prinsip dasar dari analisis ini adalah menemukan *hyperplane* terbaik yakni dengan meminimalkan kesalahan klasifikasi dan memaksimalkan margin geometrinya seperti pada Gambar 3.1 (b). Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM (Nugroho *et al.*, 2003).

Hyperplane pemisah terbaik antara kedua *class* dapat ditemukan dengan mengukur *margin hyperplane* tsb. dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing *class*. *Pattern* yang paling dekat ini disebut sebagai *support vector*. Garis solid pada

gambar 2.2(b) menunjukkan *hyperplane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua kelas, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah *support vector*.

2.1.8 *Multiple polynomial regression*

Sebuah regresi bisa dinamakan sebagai regresi polinomial jika hubungan antara variabel dependen dan variabel independen dapat digambarkan dengan kurva (Akhlaghi *et al.*, 2019). *Multiple polynomial regression* adalah regresi polinomial dengan lebih dari satu variabel independen. Dengan lebih dari satu variabel independen, persamaan regresi merupakan kombinasi dari derajat *predictor* dan perkalian antara *predictor* (Weisberg, 2014). Sebagai contoh, untuk kasus regresi polinomial berganda dengan dua prediktor, maka persamaan regresi polinomial pangkat dua-nya ditunjukkan pada persamaan (2.9).

$$E(Y|X_1 = x_1, X_2 = x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 \quad (2.9)$$

Persamaan regresi di atas adalah interaksi antara $x_1 x_2$. Dengan k predictor, model persamaan regresi terdiri dari *intercept*, k regresor linier, k regresor kuadratik, dan $k(k-1)/2$ regresor interaksi. Jika $k = 5$, persamaan regresi mempunyai 21 regresor dan jika $k = 10$, persamaan regresi memiliki 66 regresor.

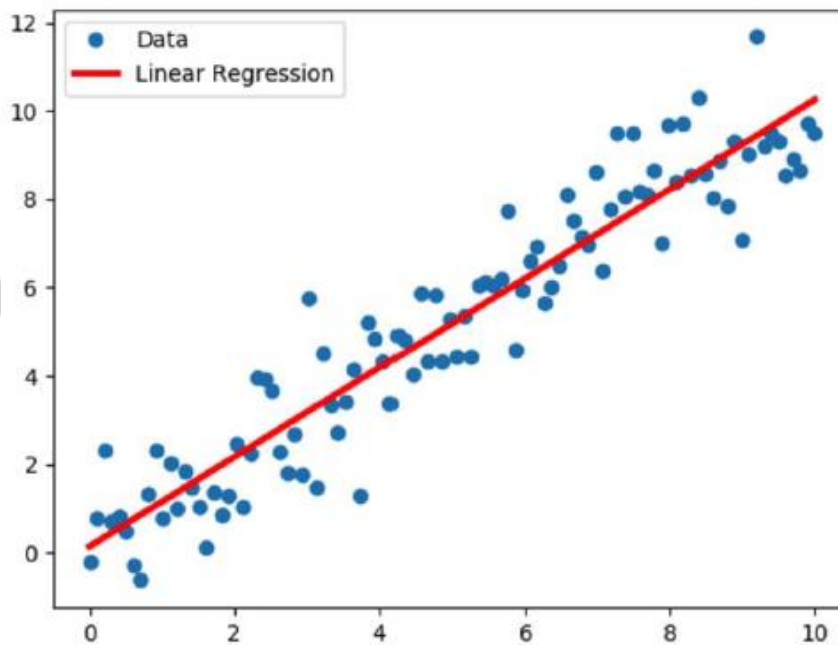
2.1.9 *Multiple linear regression*

Multiple linear regression pada dasarnya adalah pengembangan dari *simple linear regression* yang memiliki lebih dari satu variabel prediktor (Marill, 2004). *Linear Regression* merupakan salah satu jenis regresi yang melibatkan hubungan antara satu variabel dependen atau variabel tak bebas atau label (Y) dan variabel independen atau variabel bebas atau fitur (X). Besarnya nilai variabel dependen akan dipengaruhi oleh besar kecilnya variabel bebas. Algoritma *linear regression* didasarkan pada pola hubungan data terkait masa lalu (Hamdanah & Fitriana, 2021). Umumnya, algoritma *linear regression* dibagi menjadi dua jenis, yaitu *simple linear regression* dan *multiple linear regression*. *Simple linear regression*

merupakan hubungan antara satu variabel dependen dengan satu variabel independen, sedangkan *multiple linear regression* merupakan hubungan antara satu variabel dependen dengan dua atau lebih variabel independen (Herwanto *et al.*, 2019). *Multiple linear regression* yang diekspresikan melalui persamaan.

$$Y = a + a_1X_1 + b_2X_2 + \dots + b_nX_n \quad (2.10)$$

Terdapat beberapa komponen pada persamaan 2.10, di antaranya adalah Y yang merupakan variabel dependen atau nilai yang diprediksikan, a adalah konstanta, X_n adalah variabel independen, dan b_n adalah koefisien regresi. Dari persamaan ini, dapat ditarik sebuah garis yang mampu memprediksi variabel dependen berdasarkan variabel independen. Algoritma *Multiple linear regression* akan berusaha untuk menemukan garis prediksi terbaik (Gupta *et al.*, 2021). Kualitas garis prediksi dapat ditentukan dari seberapa dekatnya garis prediksi dengan poin-poin data nilai variabel dependen. Gambar 2.5 berikut menggambarkan bagaimana bentuk garis *linear regression*.



Gambar 2.5 Ilustrasi *Linear Regression*

Sumber : (Ramadhansyah, 2022)

Pada Gambar 2.5, garis yang berwarna merah merupakan garis *Linear Regression* dan poin-poin berbentuk bulat berwarna biru merupakan poin data dari

variabel independen. Sebuah garis *linear regression* dapat dikatakan bagus apabila garis mempunyai jarak yang dekat dengan keseluruhan poin-poin data variabel independen. Semakin dekat garis *linear regression* dengan poin-poin data variabel independen, maka akan semakin bagus juga prediksi yang dihasilkan.

2.1.10 *Root Mean Square Error*

Root Mean Square Error (RMSE) atau *Root Mean Square Deviation* merupakan salah satu metode yang digunakan untuk menghitung tingkat *error* pada hasil estimasi (Eka & Juarna, 2021). *Error* tersebut menunjukkan seberapa besar perbedaan atau deviasi hasil estimasi dengan nilai yang akan diestimasi. Tujuan dari *Root Mean Square Error* digunakan adalah untuk mengukur tingkat *error* dari hasil perhitungan analisis yang menggunakan metode tertentu seperti data training dan data testing. Persamaan (2.11) merupakan bentuk umum *Root Mean Square Error*.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y - \hat{y}_i)^2}{n}} \quad (2.11)$$

dengan n merupakan jumlah data atau sampel, i merupakan variabel bebas, y merupakan nilai aktual, dan \hat{y}_i adalah nilai prediksi.

2.1.11 *Mean Absolute Percentage Error*

Mean Absolute Percentage Error (MAPE) adalah nilai rata – rata perbedaan absolut yang ada di antara nilai dari prediksi dan nilai realisasi yang disebutkan sebagai hasil persentase dari nilai realisasi. Penggunaan MAPE pada evaluasi dari hasil peramalan dapat melihat tingkat akurasi terhadap angka peramalan dan angka realisasi (Nabillah & Rangadara, 2020). Nilai MAPE dapat dihitung menggunakan persamaan (2.12) (Putro *et al.*, 2018).

$$MAPE = \frac{100}{n} \sum \frac{|A_t - F_t|}{A_t} \quad (2.12)$$

dengan A_t adalah nilai aktual pada data t, F_t adalah nilai prediksi pada data t dan n adalah jumlah periode data.

2.1.11 Peramalan Deret Waktu

Peramalan deret waktu adalah teknik untuk memprediksi peristiwa melalui urutan waktu. Prediksi peristiwa masa depan dilakukan dengan menganalisis tren masa lalu menggunakan asumsi bahwa tren masa depan akan terus mirip dengan tren historis (Yudistira *et al.*, 2023). Peramalan deret waktu dapat dilakukan apabila ada waktu historis. Data historis dapat dianalisis melalui pola dekomposisi waktu, seperti tren, pola musiman, pola siklus dan tingkat keteraturan.

Teknik peramalan deret waktu terbagi menjadi dua, yaitu model peramalan berbasis model matematika statistik dan peramalan berbasis kecerdasan buatan (Sukerti, 2015). Contoh peramalan berbasis model matematika statistik adalah *moving average*, *exponential smoothing*, ARIMA (*Auto Regressive Integrated Moving Average*), SARIMA (*Seasonal Auto Regressive Integrated Moving Average*). Contoh peramalan berbasis kecerdasan buatan adalah LSTM (*Long Short Term Memory*), RNN (*Recurrent Neural Network*), MLP (*Multi Layer Perceptron*), BP (*Back Propagation*), dan lain sebagainya

a. ARIMA (*Auto Regressive Integrated Moving Average*)

ARIMA (*Auto Regressive Integrated Moving Average*) adalah suatu metode analisis dan peramalan data deret waktu. Model ARIMA menggabungkan elemen autoregressive (AR), yang menggambarkan hubungan antara nilai sekarang dengan nilai sebelumnya, dengan elemen moving average (MA), yang mencerminkan pola pergerakan rata-rata *residual* dari suatu deret waktu. "Integrated" menunjukkan bahwa deret waktu asli telah diubah menjadi deret waktu stasioner melalui proses diferensiasi. Untuk dapat melakukan pemodelan ARIMA, data harus diubah menjadi stasioner (Alfaki & Masih, 2015). Cara membuat stasioner. Proses untuk membuat data menjadi stasioner dinamakan proses diferensiasi. Proses diferensiasi membantu menstabilkan rata-rata dengan menghilangkan selisih antar data, sehingga menghilangkan atau mengurangi *trend* dan musiman (Hyndman &

Athanasopoulos, 2018). Model persamaan ARIMA ditunjukkan pada persamaan (2.13) (Wei, 1991)

$$\phi_p(B)(1 - B)^d Z_t = \theta_0 + \theta_q(B)a_t \quad (2.13)$$

dengan:

- $\phi_p(B)$: komponen autoregressive non musiman
- $(1 - B)^d$: differencing non musiman
- $\theta_q(B)$: komponen moving average non musiman
- Z_t : nilai peramalan pada waktu ke-t
- a_t : white error noise pada waktu ke-t
- B : operator backshift
- θ_0 : koefisien moving average pada waktu saat ini,

b. SARIMA (*Seasonal Auto Regressive Integrated Moving Average*)

SARIMA (Seasonal ARIMA) adalah perluasan dari ARIMA (Hyndman & Athanasopoulos, 2018). Model SARIMA dibentuk dengan menyertakan elemen musiman dalam model. Model seasonal ARIMA merupakan bentuk khusus dari model ARIMA jika terdapat unsur musiman yang jelas pada hasil observasi $\{Z_t\}$. Hal ini berarti data memiliki pola berulang – ulang dalam selang waktu yang tetap (Ainy, 2015). Notasi SARIMA disusun sebagai SARIMA (p, d, q)(P, D, Q) $_s$, di mana p, d, q adalah orde untuk komponen non-musiman, dan P, D, Q adalah orde untuk komponen musiman dengan s sebagai periode musiman.. Adapun persamaan umum dari SARIMA (p, d, q)(P, D, Q) $_s$ ditunjukkan pada persamaan (2.14) (Wei, 1991).

$$\phi_p B^s \phi_p(B)(1 - B)^d (1 - B^s)^D Z_t = \theta_q(B) \theta_q(B^s) a_t \quad (2.14)$$

dengan:

- $\phi_p(B)$: komponen *autoregressive* non musiman

$\Phi_p B^s$: komponen <i>autoregressive</i> musiman
$(1 - B)^d$: differencing non musiman
$(1 - B^s)^D$: differencing musiman
$\theta_q(B)$: komponen moving average non musiman
$\Theta_q(B^s)$: komponen moving average musiman
Z_t	: nilai peramalan pada waktu ke-t
a_t	: white error noise pada waktu ke-t
B	: operator backshift

c. RNN (Recurrent Neural Network)

RNN (*Recurrent Neural Network*) memainkan peranan dalam jaringan syaraf tiruan yang terdiri dari lapisan input, lapisan output, dan lapisan tersembunyi. RNN mampu memproses dan menyimpan informasi pada waktu yang sama, sehingga informasi dapat diteruskan ke periode selanjutnya (Henrique *et al.*, 2018). RNN mempertimbangkan data historis untuk membuat prediksi, tetapi RNN tidak mampu menyimpan data historis jangka panjang dan ada kemungkinan ledakan gradien (Jiang *et al.*, 2022). RNN dirumuskan dengan persamaan sebagai berikut (Guo *et al.*, 2023).

$$h_t = f_h(u_t x_t + W_{t-1} h_{t-1}) \quad (2.15)$$

$$y_{t+T} = f_y(v_t h_t + b_y) \quad (2.16)$$

dengan h_t merupakan vektor *hidden layer*, x_t adalah vektor *input layer*, y_{t+T} adalah vektor *output layer*, u_t adalah dari *input state* pada waktu t ke *hidden state* pada waktu t, v_t adalah bobot dari *hidden state* pada waktu t ke *output state* pada waktu t+T, dan W_{t-1} adalah bobot dari *output state* pada waktu t-1+T ke *hidden state* pada waktu t

d. LSTM (*Long Short Term Memory*)

LSTM dikembangkan dari RNN yang terdiri dari *forgetting gate*, *input gate*, dan *output gate*. LSTM mewarisi keunggulan dari RNN dan mengatasi kekurangannya (Guo *et al.*, 2023), sehingga LSTM cocok untuk peramalan deret waktu. Persamaan proses transfer LSTM ditunjukkan pada persamaan (2.15) sampai (2.19) (Jiang *et al.*, 2022).

$$F_t = \rho(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \quad (2.15)$$

$$I_t = \rho(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \quad (2.16)$$

$$O_t = \rho(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \quad (2.17)$$

$$C_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \quad (2.18)$$

$$h_t = o_t \odot \tanh(c_t) \quad (2.19)$$

dengan F_t adalah *forget gate* pada waktu t , W adalah matriks bobot, x_t adalah vektor input pada waktu t , b adalah parameter bias, h_t adalah vektor *hidden state* pada waktu t , I_t adalah *input gate* pada waktu t , O_t adalah *output gate* pada waktu t , ρ dan \tanh adalah fungsi aktivasi, dan C_t adalah set kandidat.

2.2 Tinjauan Pustaka

Penelitian mengenai prediksi *capacity factor* suatu pembangkit listrik sudah dilakukan sebelumnya oleh beberapa peneliti terdahulu. Penelitian oleh Babatunde *et al.* (2020) berfokus pada pemodelan output daya dari sebuah pembangkit listrik tenaga uap yang berlokasi di Nigeria. Penelitian ini walupun tidak langsung terkait CF, namun output daya sangat berkaitan dengan CF. Mereka menggunakan metode K-Fold Cross Validation, yang merupakan teknik validasi yang membagi data menjadi beberapa subset dan memastikan bahwa setiap subset digunakan sebagai data uji setidaknya sekali. Dalam penelitian ini, parameter output yang diukur adalah Net Power Output, yaitu daya bersih yang dihasilkan oleh pembangkit listrik. Untuk mencapai hasil ini, mereka mempertimbangkan beberapa parameter input penting, termasuk laju bahan bakar, *Air-Fuel Ratio*, laju aliran uap, temperatur

superheater, dan tekanan *superheater*. Meskipun penelitian ini berhasil memodelkan output daya, ditemukan bahwa parameter input yang digunakan belum mencakup data eksternal yang mungkin berpengaruh. Selain itu, model yang dikembangkan hanya mampu meramalkan output generator dalam jangka pendek, menunjukkan keterbatasan dalam aplikasi jangka panjang atau dalam kondisi yang berubah.

Gufron *et al.* (2022) melakukan penelitian menggunakan algoritma *Support Vector Machine* (SVM) untuk memprediksi produksi energi dari Pembangkit Listrik Tenaga Surya (PLTS) berkapasitas 1 kWp. Penelitian ini menggunakan data historis untuk meramalkan produksi energi di masa depan. Data deret waktu dibagi menjadi empat bagian, di mana tiga bagian digunakan sebagai fitur dan satu bagian sebagai target. Fitur-fitur ini mewakili kondisi historis yang dianggap relevan untuk prediksi masa depan. Meskipun SVM terbukti efektif dalam beberapa kasus, penelitian ini belum mempertimbangkan penggunaan model regresi lainnya seperti *Random Forest* atau *Multiple polynomial regression* yang mungkin memberikan hasil yang lebih baik. Selain itu, penelitian ini juga belum memasukkan parameter eksternal yang dapat mempengaruhi produksi listrik, seperti kondisi cuaca atau faktor lingkungan lainnya.

Rashid *et al.* (2020) mengeksplorasi penggunaan *Random forest regression* untuk memprediksi output power dari turbin angin. Berdasarkan data historis, mereka mengidentifikasi variabel independen yang mencakup arah angin, kecepatan angin, dan kondisi luar ruangan. *Random Forest* adalah metode pembelajaran mesin yang membangun banyak pohon keputusan dan menggabungkan hasilnya untuk mendapatkan prediksi yang lebih akurat dan mengurangi risiko *overfitting*. Penelitian ini menunjukkan hasil yang menjanjikan dalam memprediksi output power dari turbin angin. Namun, cakupan prediksi terbatas hanya pada pembangkit listrik energi terbarukan, yaitu Pembangkit Listrik Tenaga Bayu (PLTB) dan belum diterapkan pada pembangkit listrik yang terhubung ke jaringan listrik (interkoneksi). Selain itu, penelitian ini belum mempertimbangkan variabel internal seperti gangguan operasional yang dapat mempengaruhi produksi listrik PLTB secara signifikan.

Penelitian yang dilakukan oleh Tüfekci (2014) memprediksi output power dari Pembangkit Listrik Tenaga Gas dan Uap (PLTGU) yang beroperasi pada kondisi base load. Tüfekci (2014) menggunakan 15 metode regresi yang berbeda untuk menentukan metode mana yang paling efektif. Variabel independen yang digunakan dalam model ini meliputi temperatur ambien, tekanan atmosferik, kelembaban relatif, dan tekanan uap exhaust. Metode yang menghasilkan error terkecil dalam penelitian ini adalah *Bagging REPTree*, sebuah teknik *ensemble* yang menggabungkan hasil dari beberapa pohon keputusan untuk meningkatkan akurasi prediksi. Meskipun hasilnya menjanjikan, penelitian ini hanya fokus pada kondisi base load yang memiliki variasi pembebanan kecil. Oleh karena itu, belum mempertimbangkan variabel eksternal yang mungkin mempengaruhi produksi listrik, khususnya dari grid atau jaringan listrik yang lebih luas.

Purwanto *et al.* (2021) melakukan studi menggunakan lima model regresi untuk memprediksi output daya setiap setengah jam dari Pembangkit Listrik Tenaga Uap (PLTU) Batubara Paiton 1-2. Penelitian ini melibatkan beberapa variabel independen, termasuk aliran uap utama, aliran batubara total, aliran udara primer, aliran udara sekunder, dan tekanan vakum kondensor. Hasil penelitian menunjukkan bahwa model regresi ini dapat memprediksi output dengan tingkat akurasi tertentu. Namun, ditemukan bahwa parameter input yang digunakan belum mencakup data eksternal yang mungkin penting untuk meningkatkan akurasi prediksi. Selain itu, prediksi yang dilakukan hanya berlaku untuk jangka pendek, sehingga kurang bermanfaat untuk perencanaan jangka panjang atau untuk kondisi operasional yang berubah.

Penelitian-penelitian terdahulu menunjukkan bahwa meskipun berbagai metode telah digunakan untuk memprediksi output daya pada berbagai jenis pembangkit listrik, masih terdapat beberapa kesenjangan yang perlu diatasi. Secara umum, keterbatasan utama yang ditemukan meliputi kurangnya data eksternal yang dipertimbangkan dalam model, fokus yang terbatas pada prediksi jangka pendek, dan penerapan model yang belum mencakup berbagai kondisi operasional. Penelitian ini bertujuan untuk mengatasi beberapa kesenjangan tersebut dengan mengembangkan model prediksi Capacity Factor (CF) yang lebih komprehensif

dan mempertimbangkan variabel independen yang lebih luas, termasuk variabel eksternal yang mungkin mempengaruhi CF.

Dalam penelitian ini, proses untuk menentukan variabel independen yang mempengaruhi CF dan pemodelan prediksi CF pada PLTU Rembang menggunakan perangkat lunak Python dalam platform Jupyter Anaconda telah dilakukan. Analisis variabel independen yang mempengaruhi CF dilakukan dengan menggunakan metode backward elimination berbasis uji hipotesis, yaitu uji F dan uji t. Selanjutnya, pemodelan prediksi CF PLTU Rembang dilakukan dengan variabel independen yang terpilih. Model yang dikembangkan dalam penelitian ini mencakup empat metode regresi, yaitu *Random forest regression*, *Support vector regression*, *Multiple polynomial regression*, dan *Multiple linear regression*. Model prediksi yang dipilih adalah model yang memiliki Mean Absolute Percentage Error (MAPE) terkecil dan nilai *R-squared* terbesar.

