

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Terdapat beberapa sumber yang dijadikan referensi oleh penulis dalam melakukan penelitian ini. Berikut referensi yang berkaitan dengan penelitian ini.

Pertama, penelitian yang berjudul “Composing Multi-Instrumental Music with Recurrent Neural Networks” yang dilakukan oleh David Samuel dan Martin Pilat pada tahun 2019. Penelitian tersebut berasal dari permasalahan kompleksitas dalam mengukur kualitas musik secara Otomatis, dikarenakan musik memiliki struktur hierarkis yang sangat rumit. Tujuan penelitian tersebut adalah untuk menciptakan sebuah model generatif yang mampu mentransformasi musik multi-instrumen dengan tingkat kesamaan seperti yang dilakukan oleh manusia. Penelitian tersebut dijalankan melalui serangkaian metode, yang pertama adalah pengumpulan dataset, kemudian dilakukan proses pra-pemrosesan data, pembentukan model generatif, penetapan format untuk input dan output modul, pembangunan arsitektur saraf, dan akhirnya proses pengambilan sampel dari model generatif. Pengumpulan dataset dilakukan dengan mengumpulkan sebanyak 126.506 berkas MIDI dari sumber internet. Dataset ini kemudian dibagi menjadi dua bagian, di mana 1.000 data digunakan sebagai set validasi dan sisa data untuk set pelatihan. Proses pra-pemrosesan data bertujuan untuk mengurangi kompleksitas berkas MIDI, terutama pada aspek ekspresi musikal, dengan menerapkan metode deteksi metrik, deteksi kunci, serta deteksi akor. Selain itu, juga dilakukan pengelompokan instrumen, reduksi rentang nada, penyesuaian dinamika, dan penyetaraan pitch dalam tahap pra-pemrosesan data. Selanjutnya, dalam tahap input dan output modul, terdapat tiga modul utama, yaitu prediktor akor, prediktor nada, dan prediktor volume. Prediktor akor hanya menggunakan 24 akor

yang telah terdeteksi oleh detektor akor sebagai input dan output-nya. Prediktor nada menerima dua input, yaitu ID peristiwa terakhir dan ID akor yang akan dimainkan pada ketukan berikutnya. Prediktor volume menerima input dari peristiwa volume dan menghasilkan output berupa nada yang telah dinormalisasi. Untuk arsitektur jaringan saraf, tahap pembentukannya melibatkan lapisan embedding, k lapisan LSTM berulang dengan normalisasi batch, serta lapisan akhir yang terhubung sepenuhnya ke aktivasi softmax. Pada tahap pengambilan sampel dari model generatif, dilakukan secara iteratif dengan menggunakan peristiwa baru sebagai input pada setiap iterasi berikutnya. Hasil penelitian tersebut terdiri dari tanggapan dari 293 responden. Berdasarkan pertanyaan yang diajukan, sekitar 62,6% responden menyatakan bahwa musik yang dihasilkan tidak selalu mudah dibedakan, namun terkadang mengandung kesalahan. Dari segi kualitas, rata-rata kesalahan dalam lagu buatan manusia adalah 3,126, sedangkan lagu yang dihasilkan oleh model memiliki rata-rata kesalahan sebesar 2,745. Hal ini menunjukkan bahwa model yang dibuat berhasil menciptakan musik yang dapat dinikmati. Lebih dari 60,5% responden menyatakan bahwa musik tersebut terdengar seperti karya instrumen tunggal, dan sekitar 65,0% responden menganggapnya sebagai karya multi-instrumen. Dari total 2,930 jawaban, hanya 40 di antaranya yang mengenali bagian yang terasa familiar. Hanya delapan karya yang berhasil diidentifikasi sebagai hasil karya komputer [26].

Kedua, penelitian yang berjudul "Speech Coding Using Discrete Cosine Transform and Chaotic Map" yang dilaksanakan oleh Marwa Jamal dan Tariq A. Hassan pada tahun 2022. Penelitian tersebut berakar pada permasalahan mengenai tingginya biaya pemrosesan dan besar penyimpanan yang diperlukan dalam pemanfaatan data berskala besar. Penelitian tersebut bertujuan untuk memanfaatkan pendekatan kompresi yang mampu mengurangi kehilangan informasi dalam sinyal suara hingga tingkat tertentu. Penelitian tersebut dijalankan dengan metode sebagai berikut. Pada tahap pertama, sinyal di-segmentasikan ke dalam bingkai

dengan panjang tetap 30 milidetik dan energi dihitung untuk tiap bingkai. Bingkai dengan energi di atas ambang batas diatribusikan nilai satu, sementara yang lainnya nol. Sinyal yang diekstrak dari tahap pertama, diatur dalam vektor baru. Pada tahap ini, sinyal ucapan di-segmentasikan menjadi bingkai 1 detik, masing-masing terdiri dari 8.000 sampel. Transformasi Kosinus Diskret (DCT) diterapkan pada setiap bingkai, dan kemudian 25% dari setiap bingkai dihilangkan dalam langkah kuantisasi. Untuk menerapkan proses enkripsi, dilakukan langkah pemutasi berdasarkan kunci yang dihasilkan dari peta logistik, dan proses XOR dilakukan pada hasilnya dengan kunci yang dihasilkan dari peta Gauss. Hasil XOR diubah menjadi urutan biner dengan satu kolom bit (0,1) untuk persiapan tahap pengkodean Run-Length. Run-Length Encoding (RLE) menghilangkan nilai-nol dan nilai-satu, sambil tetap mempertahankan panjangnya. Nilai pertama dalam urutan tersebut mengacu pada jumlah angka satu, diikuti oleh jumlah angka nol dalam urutan, dan seterusnya. Hasil dari penelitian tersebut menunjukkan bahwa sistem yang diusulkan secara signifikan meningkatkan efisiensi pengkodean dan berhasil mencapai rasio kompresi sebesar 13%, sebuah hasil yang memadai dalam pengkodean suara. Sistem kompresi yang diusulkan, yang menggunakan metode DCT dan RLE, berhasil menunjukkan rasio kompresi yang dapat diterima sambil mempertahankan kualitas sinyal ucapan. Penggunaan RLE berhasil mengurangi ukuran fisik dari nilai yang berulang dalam sinyal [27].

Ketiga, penelitian yang berjudul “K-means clustering analysis of Chinese traditional folk music based on midi music textualization” yang dilakukan oleh Zhang Liumei, Li Jiao, Liu Tianshi, Jiang Fanzhi dan Ma Gang pada tahun 2021. Penelitian tersebut didasarkan pada masalah yaitu masih jarang yang melakukan ekstraksi fitur dan analisis musik secara langsung pada musik simbolik. Penelitian tersebut bertujuan untuk mengamati bahwa data teks relatif terhadap skala pentatonic tradisional. Penelitian tersebut menggunakan metode yaitu pengumpulan dataset dan pemrosesan data. Pada tahap pengumpulan dataset, diperoleh 1.300 musik

rakyat tradisional Tiongkok yang diambil melalui internet dengan panjang rata – rata sekitar 12 detik. Setelah itu, mengubah file MIDI menjadi obyek skor, lalu mengekstrak not di bagian piano instrumen, karena data not memiliki tangga nada yang berbeda dan tangga nada berada dalam kisaran satu kelompok huruf kecil ke kelompok huruf kecil, sehingga tanda tangga nada dibuang dan nada diekstraksi part, lalu tekstualisasi untuk mendapatkan Kumpulan data teks asli berdasarkan simbol musik. Pada tahap pemrosesan data, langkah pertama yang dilakukan adalah mengekstrak nada – nada komplit untuk tekstualisasi. Setelah diproses, jumlah fitur yang diekstrak adalah 19, yang didistribusikan dalam keyboard piano. Kemudian menyederhanakan data dan memfasilitasi statistic pengelompokkan, nada – nada diekstraksi dan skalanya dibuang. Setelah vektorisasi, jumlah fitur yang diekstraksi menjadi Sembilan yaitu C, D, E, F, F#, G, A, B, Bb dan kemudian dikonversi ke perhitungan matriks bobot, didapatkan $87.000 * 9$ tensor. Hasil yang diperoleh dari penelitian tersebut yaitu menurut tujuh indikator evaluasi obyektif dari algoritma pengelompokan yaitu Adjusted_rand_score, V_measure_score, Mutual_info_score, Completeness_score, Homogenety_score, Calinski_harabasz_score dan Silhouette_score, K-Means menghasilkan efek pengelompokkan yang baik, dengan menggabungkan pengetahuan teori musik professional dengan perkembangan sejarah musik rakyat tradisional Tiongkok. Karakteristiknya menunjukkan bahwa kumpulan data ini memiliki karakteristik modal yang berbeda dari musik nasional tradisional Tiongkok dan eksperimen ini telah menghasilkan kombinasi tata bahasa musik yang baik [28].

Keempat, penelitian yang berjudul “An automatic music generation method based on RSCLN_Transformer network” yang dilakukan oleh Yumei Zhang, Xiaojiao Lv, Qi Li, Xiaojun Wu, Yuping Su dan Honghong Yang pada tahun 2022. Penelitian ini bertujuan untuk mengatasi keterbatasan yang ada pada metode-metode seperti RNN, LSTM, Transformer, dan model urutan lainnya dalam menghasilkan musik. Kendala-kendala tersebut termasuk ketidakmampuan dalam menangkap

ketergantungan sekuens musik yang panjang pada waktu sebelum dan sesudahnya, perbedaan yang rendah antara musik yang dihasilkan dan gaya aslinya, serta masalah hilangnya gradien selama tahap pelatihan model. Penelitian ini menggunakan pendekatan untuk memperbaiki representasi peristiwa MIDI dengan menggunakan konsep Revamped MIDI-derived event (REMI) untuk mengubah musik menjadi urutan peristiwa yang dapat diproses. Selain itu, penelitian ini juga memperbaiki proses pengoptimalan jaringan dengan mengimplementasikan modul RSCLN, yang membantu dalam mengatasi masalah hilangnya gradien melalui penggunaan koneksi yang dilewati. Tahapan penelitian meliputi pengumpulan data latihan dari musik MIDI, peningkatan dinamis data secara online pada dataset latihan MIDI, pemilihan bagian data untuk penyempurnaan model, penguraian informasi dari input MIDI menggunakan miditoolkit, konversi informasi musik menjadi urutan peristiwa REMI, serta pelatihan model generasi musik Otomatis RSCLN_Transformer-XL. Hasil dari penelitian ini mengindikasikan bahwa model RSCLN_Transformer-XL mampu menghasilkan musik dengan kualitas yang lebih baik daripada musik yang dihasilkan oleh model Transformer-XL [29].

Kelima, penelitian yang berjudul "Automatic Melody Composition Using Enhanced GAN" yang dilakukan oleh Shuyu Li, Sejun Jang, dan Yunsick Sung pada tahun 2019. Penelitian tersebut berfokus pada permasalahan yang berkaitan dengan waktu yang dibutuhkan untuk menciptakan karya musik baru. Sebagian besar metode komposisi musik mengalami kendala dalam menghasilkan komposisi berdasarkan not-not musik. Tujuan dari penelitian tersebut adalah untuk memastikan keberterimaan dalam pembangunan matriks berdimensi tinggi yang mencerminkan fitur-fitur bar, serta penerapan pendekatan berbasis CNN sebagai pendekatan pembeda. Penelitian tersebut mengikuti beberapa tahapan. Pertama-tama, dalam fase pra-pemrosesan, data pelatihan yang diperlukan untuk model dihasilkan. Perbedaan antara frekuensi tinggi pada melodi dan bagian non-melodi diidentifikasi menggunakan algoritma TF-

IDF. Melodi dalam karya musik diekstraksi sebagai data pelatihan melalui filter yang didasarkan pada deskripsi struktur dangkal. Melodi yang telah diekstraksi kemudian dipecah menjadi sejumlah bar menggunakan informasi tempo yang ada dalam file MIDI. Kemudian, semua not pada setiap bar, yang terdiri dari empat fitur yaitu nada, waktu mulai, durasi, dan intensitas, akan dikodekan ke dalam matriks dengan dimensi yang lebih tinggi. Pada akhirnya, semua matriks bar yang telah dikodekan dinormalisasi agar memiliki dimensi yang seragam. Tahap kedua, pada proses pembuatan file MIDI, matriks melodi baru dihasilkan selama proses pelatihan dengan menggunakan model GAN, dan kemudian hasilnya diperbaiki dengan menggunakan data pelatihan yang telah diproses sebelumnya. Matriks melodi baru ini dapat diubah menjadi sebuah file MIDI melalui transformasi tertentu. Model GAN yang telah diperbaiki terdiri dari satu generator dan dua diskriminator. Dua diskriminator ini mencakup diskriminator berbasis RNN dan diskriminator berbasis CNN. Diskriminator berbasis RNN mampu mempertimbangkan karakteristik melodi dalam bentuk data urutan waktu, sementara diskriminator berbasis CNN mempertimbangkan seluruh struktur melodi secara keseluruhan. Generator menghasilkan matriks melodi baru berdasarkan vektor noise dan mengirimkannya kepada kedua diskriminator. Kedua diskriminator tersebut kemudian menilai apakah matriks melodi masukan merupakan hasil dari generator atau matriks melodi yang telah diekstrak dari file MIDI. Baik diskriminator maupun generator dilatih dengan menggunakan output dari diskriminator. Selanjutnya, kedua diskriminator ini digunakan untuk melatih model GAN yang diajukan. Hasil dari penelitian ini menunjukkan bahwa struktur melodi yang dihasilkan dengan pendekatan GAN memiliki kemiripan yang mencapai 80% dengan struktur melodi asli [30].

Penelitian – penelitian tersebut telah dirangkum dalam Tabel 2.1 sebagai berikut.

Tabel 2.1 Ringkasan Penelitian Sebelumnya

No.	Judul Penelitian (Penulis, Tahun)	Masalah Penelitian	Tujuan Penelitian	Metode / Teknik Penyelesaian	Hasil Penelitian	Perbedaan
1.	Composing Multi-Instrumental Music with Recurrent Neural Networks (David Samuel dan Martin Pilat, 2019)	Kompleksitas dalam mengukur kualitas musik secara Otomatis, dikarenakan musik memiliki struktur hierarkis yang sangat rumit	Untuk menciptakan sebuah model generatif yang mampu mentransformasi musik multi-instrumen dengan tingkat kesamaan seperti yang dilakukan oleh manusia	Pertama adalah pengumpulan dataset, kemudian dilakukan proses pra-pemrosesan data, pembentukan model generatif, penetapan format untuk input dan output modul, pembangunan arsitektur saraf, dan akhirnya proses pengambilan sampel dari model generatif	Meskipun outputnya tidak sempurna dan terbatas pada representasi simbolis, ia dapat menciptakan komposisi yang dapat didengarkan yang sulit dibedakan dari musik buatan manusia	Dataset yang digunakan dalam penelitian tersebut yaitu MIDI file, sedangkan dalam penelitian ini menggunakan suara latar belakang musik tanpa suara vokal
2.	Speech Coding Using Discrete Cosine Transform and Chaotic Map (Marwa Jamal dan Tariq A. Hassan, 2022)	Tingginya biaya pemrosesan dan besar penyimpanan yang diperlukan dalam pemanfaatan data berskala besar	Untuk memanfaatkan pendekatan kompresi yang mampu mengurangi kehilangan informasi dalam sinyal suara hingga tingkat tertentu	Segmentasi sinyal, pengaturan ulang sinyal, transformasi DCT, proses enkripsi dan pengkodean Run-Length (RLE)	Sistem yang diusulkan secara signifikan meningkatkan efisiensi pengkodean dan mencapai rasio kompresi sebesar 13% yang seharusnya merupakan hasil yang baik dalam pengkodean ucapan	Pada penelitian tersebut tahap setelah transformasi DCT yaitu pengkodean Run-Length, sedangkan pada penelitian ini dilakukan clustering setelah transformasi DCT.

No.	Judul Penelitian (Penulis, Tahun)	Masalah Penelitian	Tujuan Penelitian	Metode / Teknik Penyelesaian	Hasil Penelitian	Perbedaan
3.	K-means clustering analysis of Chinese traditional folk music based on midi music textualization (Zhang Liumei, Li Jiao, Liu Tianshi, Jiang Fanzhi dan Ma Gang, 2021)	Masih jarang yang melakukan ekstraksi fitur dan analisis musik secara langsung pada musik simbolik	Untuk mengamati bahwa data teks relatif terhadap skala pentatonic tradisional	Pengumpulan dataset dan pemrosesan data	Menurut tujuh indikator evaluasi obyektif dari algoritma pengelompokan yaitu, K-Means menghasilkan efek pengelompokan yang baik, dengan menggabungkan pengetahuan teori musik profesional dengan perkembangan sejarah musik rakyat tradisional Tiongkok	Clustering pada penelitian tersebut berfokus pada mengelompokkan fitur simbolik pada musik tradisional China, sedangkan penelitian ini clustering digunakan untuk mengelompokkan fitur hasil transformasi DCT.
4.	An automatic music generation method based on RSCLN_Transformer network (Yumei Zhang, Xiaojiao Lv, Qi Li, Xiaojun Wu, Yuping Su dan Honghong Yang, 2022)	Ketidakmampuan model sequence sebelumnya dalam menangkap ketergantungan sekuens musik yang panjang pada waktu sebelum dan sesudahnya, perbedaan yang rendah antara musik yang dihasilkan dan gaya aslinya, serta masalah hilangnya gradien selama tahap pelatihan model	Untuk mengatasi keterbatasan yang ada pada metode-metode seperti RNN, LSTM, Transformer, dan model urutan lainnya dalam menghasilkan musik	Pengumpulan data latihan dari musik MIDI, peningkatan dinamis data secara online pada dataset latihan MIDI, pemilihan bagian data untuk penyempurnaan model, penguraian informasi dari input MIDI menggunakan miditoolkit, konversi informasi musik menjadi urutan peristiwa REMI, serta pelatihan model generasi musik Otomatis RSCLN_Transformer-XL	Model RSCLN_Transformer-XL mampu menghasilkan musik dengan kualitas yang lebih baik daripada musik yang dihasilkan oleh model Transformer-XL	Penelitian tersebut menggunakan algoritma RSCLN_Transformer sedangkan penelitian ini menggunakan algoritma RNN

No.	Judul Penelitian (Penulis, Tahun)	Masalah Penelitian	Tujuan Penelitian	Metode / Teknik Penyelesaian	Hasil Penelitian	Perbedaan
5.	Automatic Melody Composition Using Enhanced GAN (Shuyu Li, Sejun Jang, dan Yunsick Sung, 2019)	Waktu yang dibutuhkan untuk menciptakan karya musik baru	Untuk memastikan keberterimaan dalam pembangunan matriks berdimensi tinggi yang mencerminkan fitur-fitur bar, serta penerapan pendekatan berbasis CNN sebagai pendekatan pembeda	Tahap pra-pemrosesan, pembuatan file MIDI dan model GAN	Struktur melodi yang dihasilkan dengan pendekatan GAN memiliki kemiripan yang mencapai 80% dengan struktur melodi asli	Pada penelitian tersebut RNN dijadikan sebagai salah satu diskriminator, sedangkan pada penelitian ini RNN dijadikan sebagai model utama dalam mengarang musik.

2.2 Landasan Teori

2.2.1 Musik

Musik pada dasarnya merupakan bunyi yang diungkapkan melalui melodi yang indah yang memiliki ritme yang teratur dan tercipta dari berbagai media. Terdapat beberapa unsur yang ada di dalam musik yang diuraikan sebagai berikut.

1. Melodi merupakan naik turunnya atau tinggi rendahnya suatu nada yang terdiri dari durasi, pitch dan tone. Durasi yaitu serangkaian waktu tertentu dari suatu rangkaian not yang berubah menjadi melodi. Pitch atau warna suara yaitu merupakan kunci berupa alfabet yang digunakan untuk mengatur serangkaian not. Tone yaitu pitch yang menghasilkan not dari serangkaian alat musik yang berbeda.
2. Ritme (irama) merupakan ketetapan dalam memunculkan aksen yang mengalirkan gerak teratur.
3. Birama merupakan keteraturan dari suatu ketukan atau ayunan dalam waktu yang sama.
4. Harmoni merupakan bunyi yang selaras.
5. Tempo merupakan kecepatan gerak musik atau lagu.
6. Dinamik merupakan perubahan suatu lagu yang dinilai dari keras dan lembutnya.
7. Tangga Nada merupakan nada yang terurut.
8. Timbre merupakan pitch yang dapat membedakan kesan dari kualitas bunyi.
9. Notasi merupakan unsur yang terbentuk dari suatu lagu yang terdiri dari not angka dan not balok [31].

2.2.2 Otomatisasi

Otomatisasi pertama kali muncul sekitar tahun 1946 dalam industri otomotif dengan tujuan meningkatkan penggunaan kontrol dan perangkat otomatis dalam jalur produksi mekanis. Seiring waktu, istilah "otomatisasi" juga digunakan di luar bidang manufaktur,

mencakup penggunaan teknologi listrik, tindakan mekanis, dan komputerisasi untuk meningkatkan kecerdasan dan usaha manusia. Otomatisasi merujuk pada penerapan teknologi guna mengurangi intervensi manusia, membuatnya lebih mudah untuk dipantau melalui berbagai sistem kontrol. Definisi ini diberikan oleh beberapa ahli yaitu sebagai berikut.

Otomatisasi adalah penerapan teknologi dan kreativitas untuk mengontrol produksi, pemantauan, serta penyampaian layanan dan produk, seperti yang dinyatakan oleh International Society of Automation. Sementara menurut Technopedia, otomatisasi adalah penerapan teknologi dan kreativitas untuk mengirimkan dan memproduksi barang dan jasa dengan meminimalkan campur tangan manusia.

Tujuan utama otomatisasi adalah sebagai berikut.

1. Melakukan lebih banyak dengan sumber daya yang ada.
2. Mengurangi kesalahan yang berasal dari kesalahan manusia.
3. Meningkatkan keselamatan dengan menghindari keterlibatan langsung manusia.
4. Membebaskan tenaga kerja manusia untuk fokus pada tugas yang memerlukan kecermatan manusia.
5. Meningkatkan kualitas produk dengan biaya produksi yang lebih rendah.

Manfaat dari otomatisasi mencakup hal – hal berikut.

1. Mengurangi penggunaan dokumen fisik dengan menyimpan informasi dalam sistem, mengurangi biaya cetak dan penyimpanan dokumen.
2. Mengoptimalkan waktu karyawan dengan memungkinkan mereka untuk fokus pada pekerjaan yang membutuhkan kreativitas dan pengambilan keputusan.

3. Menghemat biaya dengan menggantikan tugas rutin manusia dengan mesin.
4. Mendorong efisiensi kerja karyawan dengan memberi mereka waktu istirahat yang cukup.
5. Memudahkan pengelolaan tim yang tersebar di lokasi dan waktu yang berbeda melalui sistem otomatis.

Dengan demikian, otomatisasi telah membawa perubahan besar dalam berbagai industri dengan tujuan meningkatkan efisiensi, produktivitas, dan kualitas kerja [32].

2.2.3 Deep Learning

Pembelajaran Mendalam (Deep Learning) adalah salah satu cabang dari Machine Learning yang memanfaatkan algoritma untuk menggambarkan abstraksi tingkat tinggi dari data melalui serangkaian fungsi transformasi non-linear yang terstruktur dalam lapisan-lapisan yang mendalam. Pendekatan ini sangat berguna dalam berbagai jenis Pembelajaran Terbimbing, Tanpa Pengawasan, Semi-Terbimbing, dan Pembelajaran Penguatan, serta dapat diterapkan dalam beragam konteks seperti pengenalan citra, suara, klasifikasi teks, dan lainnya.

Model dalam Pembelajaran Mendalam didasarkan pada Jaringan Saraf Tiruan (Neural Network). Meskipun penelitian di bidang ini telah dimulai sejak era 1980-an, minatnya baru-baru ini meningkat berkat kemajuan dalam kemampuan komputasi. Hal ini lebih diperkuat dengan perkembangan Teknologi Big Data, termasuk penggunaan platform seperti Hadoop dan Spark yang menggunakan kluster multi-node, serta penerapan pemrosesan paralel berbasis GPU.

Pembelajaran Mendalam memanfaatkan fitur berbentuk hirarki, yang dapat diatur dalam ukuran yang disesuaikan dengan kasus yang dihadapi. Algoritma Deep Learning mampu secara otomatis mengekstraksi fitur dari data mentah secara rinci, berkat penerapan struktur eksploitasi. Fitur-fitur ini tidak dapat dengan mudah dikenali secara visual, terutama karena distribusi pembeda kelas data sering

kali kompleks. Oleh karena itu, fitur tingkat tinggi harus diubah menjadi fitur paling dasar agar dapat dipahami oleh mesin pembelajaran. Ini mengindikasikan bahwa jika fitur tingkat dasar dapat diidentifikasi dengan mudah oleh Deep Learning, maka fitur tingkat tinggi akan lebih mudah diidentifikasi.

Kombinasi inilah yang membuat Deep Learning mampu menghasilkan representasi fitur yang optimal dan efektif. Pendekatan eksploitasi fitur ini berusaha untuk menggali dan menyaring ruang atau wilayah tertentu hingga kedalaman maksimal. Di sisi lain, pendekatan eksplorasi bertujuan untuk mencakup sebanyak mungkin ruang atau wilayah target, baik dengan teknik yang serupa atau berbeda selama proses eksplorasi. Prinsip kerja algoritma Deep Learning dijelaskan dalam langkah-langkah berikut.

1. Merancang arsitektur yang tepat, yang meliputi jenis metode Deep Learning yang sesuai dengan kompleksitas kasus yang dihadapi. Ini termasuk layer input, hidden (darknet/layer gelap/tersembunyi), dan output.
2. Menyiapkan data atau elemen yang diperlukan selama proses pelatihan dan pengujian Deep Learning.
3. Memilih tempat implementasi (lokal atau menggunakan cloud) dan jenis kode program (dari awal, kombinasi kode dari library, atau hanya menggunakan library tertentu).
4. Melakukan pengujian yang sesuai dengan standar algoritma Deep Learning, mulai dari parameter hingga pengujian arsitektur. Pendekatan pengujian bisa berbentuk waterfall (secara sekuensial tanpa loop) atau recycle (secara sekuensial dengan beberapa loop). Pendekatan recycle ini melibatkan pengujian berulang kali hingga mencapai stabilitas parameter yang memberikan akurasi tertinggi.
5. Memikirkan cara mengoptimalkan lebih lanjut algoritma Deep Learning berdasarkan hasil yang telah diperoleh [33].

2.2.4 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) dirancang untuk memanfaatkan dan belajar dari informasi berurutan. Arsitektur RNN menjalankan tugas yang sama untuk setiap elemen urutan. RNN sangat berguna dalam tugas pemrosesan bahasa alami karena ketergantungan berurutan kata-kata dalam bahasa apa pun. Secara umum, pada setiap langkah urutan waktu, RNN menghitung beberapa memori berdasarkan perhitungannya sejauh ini; yaitu, memori sebelumnya dan input saat ini. Memori yang dihitung ini digunakan untuk membuat prediksi pada langkah waktu saat ini dan diteruskan ke langkah berikutnya sebagai input [34].

2.2.5 Gated Recurrent Unit (GRU)

GRU pertama kali diperkenalkan pada tahun 2014 oleh Chung et al. Tujuan utama pengembangan GRU adalah menciptakan setiap unit berulang yang mampu menangkap ketergantungan dalam rentang waktu yang beragam secara adaptif. Untuk memberikan gambaran, manusia tidak perlu dan terkadang tidak seharusnya mempertimbangkan semua informasi masa lalu ketika membuat keputusan saat ini. Sebagai contoh, saat kita memutuskan untuk makan, informasi tentang jadwal ujian tengah semester dari masa lalu tidak akan memiliki dampak besar pada keputusan tersebut.

Dalam struktur GRU, elemen yang mengendalikan aliran informasi ini disebut sebagai gate, dan GRU memiliki dua jenis gate, yaitu reset gate dan update gate. Jika kita ingin membuat keputusan tentang makanan seperti yang dijelaskan sebelumnya, gate reset dalam GRU akan mengatur bagaimana cara menggabungkan input baru dengan informasi masa lalu, sementara gate update akan menentukan seberapa banyak informasi dari masa lalu yang perlu dipertahankan [35].

2.2.6 Frekuensi

Frekuensi merujuk pada jumlah gelombang yang melewati titik tetap dalam satuan waktu. Ini juga mengacu pada jumlah siklus atau getaran yang terjadi dalam satu satuan waktu oleh suatu benda dalam gerakan periodik. Benda yang bergerak secara berkala dianggap telah melalui satu siklus atau getaran setelah melewati serangkaian posisi atau peristiwa dan kembali ke posisi awal. Periode atau interval waktu yang diperlukan untuk menyelesaikan satu siklus atau getaran berbanding terbalik dengan frekuensi. Sebagai contoh, jika periode suatu fenomena adalah 1/2 detik, maka frekuensinya adalah 2 per detik. Frekuensi juga dapat dihitung sebagai kebalikan dari periode, yaitu $\text{frekuensi} = 1/\text{periode} = 1/(\text{interval waktu})$.

Satuan yang umum digunakan untuk mengukur frekuensi adalah hertz (Hz), yang dinamai berdasarkan fisikawan Heinrich Rudolf Hertz. Satu hertz mengindikasikan satu siklus per detik. Satuan lainnya termasuk kilohertz (kHz), yang setara dengan 1.000 Hz, dan megahertz (MHz), yang setara dengan 1.000.000 Hz. Dalam konteks spektroskopi, kadang-kadang digunakan satuan bilangan gelombang, yang mengukur jumlah gelombang dalam satuan jarak [36].

2.2.7 Nada

Sistem melodi yang umum digunakan saat ini adalah pengaturan berjenjang dari nada terendah ke nada tertinggi, terdiri dari tujuh nada dasar dengan interval setengah nada di antara mereka, serta satu interval nada penuh, yang dikenal sebagai oktaf. Nama untuk setiap nada dalam susunan ini mengikuti tujuh huruf pertama dalam abjad: a, b, c, d, e, f, dan g. Johann Sebastian Bach telah mengembangkan sistem ini menjadi 12 nada dengan interval setengah nada di antara masing-masingnya, yang dikenal sebagai sistem well-tempered. Sistem tangga nada diatonis dengan tujuh nada dasar didasarkan pada variasi jarak antara nada dan setengah nada. Terlepas dari sebutan yang berbeda, nada cis dan des merujuk pada nada-nada dengan

frekuensi identik, dan keduanya disebut sebagai contoh harmonis dalam musik [37].

2.2.8 Discrete Cosine Transform (DCT)

Discrete Cosine Transform (DCT) merupakan sebuah transformasi yang utamanya digunakan dalam algoritma kompresi. Transformasi ini mengubah titik data dalam domain spasial menjadi domain frekuensi, memudahkan identifikasi pengulangan pola. DCT memungkinkan pengembalian data asli dari transformasi yang diberikan [38].

DCT digunakan dalam kompresi citra dengan kehilangan informasi karena memiliki kemampuan pemadatan energi yang kuat. Informasi dalam jumlah besar disimpan dalam komponen frekuensi sinyal yang sangat rendah, sedangkan frekuensi lainnya menyimpan data yang kecil dan memerlukan jumlah bit yang sedikit (biasanya 2 atau 3 bit) untuk penyimpanan [39].

Cara kerja DCT yaitu mengambil data dalam bentuk domain piksel, kemudian mengonversinya ke bentuk lain dengan ketelitian yang cermat. Data keluaran disusun dalam urutan menurun sesuai dengan tingkat kepentingannya. Proses transformasi domain ini memungkinkan kita menggunakan hanya beberapa titik data keluaran sambil tetap menjaga hubungan dengan data asli [40].

2.2.9 K-Means

Algoritma K-Means bertujuan untuk mengelompokkan pengamatan yang serupa berdasarkan jaraknya dari pusat kluster. Proses penugasan dimulai dengan mengambil pengamatan data dan membentuk himpunan awal k-titik rata-rata melalui perhitungan rata-rata dari tiga pengamatan acak dalam data. Dari titik awal ini, setiap pengamatan diberikan ke pusat kluster yang memiliki penugasan dengan jumlah kuadrat terkecil dalam kluster tersebut. Penugasan ini ditentukan oleh norma Euclidean antara rata-rata pengamatan dan rata-rata pusat kluster [41].

2.2.10 Overfitting

Overfitting terjadi saat model statistik tidak mampu memberikan prediksi yang akurat pada data pengujian. Ini terjadi ketika model dilatih menggunakan banyak data, yang mengakibatkan model belajar dari noise dan entri data yang tidak akurat di dalam kumpulan data. Dalam pengujian dengan data yang tidak terlihat sebelumnya, ini menghasilkan varians yang tinggi. Terlalu banyak detail dan noise membuat model kesulitan mengkategorikan data dengan benar. Overfitting biasanya disebabkan oleh metode non-parametrik dan non-linear, di mana algoritma pembelajaran mesin memiliki kebebasan lebih dalam membangun model, namun bisa mengarah pada pembentukan model yang tidak realistis.

Solusi untuk mengatasi overfitting termasuk menggunakan algoritma linier ketika data berhubungan linier atau mengatur parameter seperti kedalaman maksimal pada pohon keputusan. Secara ringkas, overfitting terjadi ketika performa algoritma pembelajaran mesin pada data pelatihan berbeda dengan performanya pada data yang tidak digunakan selama pelatihan.

Penyebab Overfitting adalah sebagai berikut.

1. Varians tinggi dan bias rendah.
2. Model terlalu kompleks.
3. Ukuran data pelatihan terbatas.

Teknik Mengurangi Overfitting adalah sebagai berikut.

1. Meningkatkan jumlah data pelatihan.
2. Mengurangi kompleksitas model.
3. Penghentian awal selama pelatihan (menghentikan pelatihan saat loss mulai meningkat).
4. Menggunakan regularisasi Ridge dan Lasso.
5. Menggunakan dropout dalam jaringan saraf untuk menangani overfitting [42].

2.2.11 Underfitting

Model statistik mengalami underfitting saat tidak mampu menangkap tren mendasar data, sehingga hanya berhasil pada data pelatihan dan gagal pada data pengujian, mirip dengan mengenakan celana berukuran kecil yang merusak kenyamanan. Ketidakcocokan ini merusak akurasi model pembelajaran mesin, menunjukkan bahwa model atau algoritme tidak cocok dengan data. Ini umumnya muncul saat data yang ada sedikit untuk membangun model yang tepat atau saat mencoba membuat model linier untuk data non-linier. Dalam kasus ini, aturan model pembelajaran mesin terlalu sederhana dan fleksibel untuk data yang minim, sehingga menghasilkan banyak prediksi yang salah. Solusinya termasuk penggunaan lebih banyak data dan seleksi fitur. Secara ringkas, underfitting terjadi ketika model tidak dapat berkinerja baik pada data pelatihan atau mampu menggeneralisasi data baru.

Penyebab Underfitting dapat disebabkan oleh hal berikut.

1. Bias yang tinggi dan varian yang rendah.
2. Ukuran dataset pelatihan yang terbatas.
3. Kesederhanaan model.
4. Data pelatihan yang belum diperbaiki dan berisi noise.

Teknik untuk Mengurangi Underfitting meliputi hal berikut.

1. Meningkatkan kompleksitas model.
2. Menambah jumlah fitur dan melakukan rekayasa fitur.
3. Membersihkan data dari kebisingan.
4. Menambah jumlah epochs atau durasi latihan untuk hasil yang lebih baik [42].

2.2.12 Bestfitting

Sebuah model dianggap paling sesuai (best fitting) saat mampu melakukan generalisasi dan menunjukkan perilaku yang serupa pada data pelatihan dan pengujian. Tingkat ketepatan (akurasi) model seharusnya hampir identik antara data pelatihan dan pengujian.

Dengan kata lain, keadaan paling sesuai tidak merujuk pada terjadinya overfitting atau underfitting, melainkan mewakili model yang umum dan konsisten dalam menghadapi berbagai jenis data [43].

2.2.13 Bias

Bias, dalam pengertian sederhana, terjadi ketika model tidak mampu secara akurat memprediksi nilai sebenarnya karena adanya perbedaan atau kesalahan antara prediksi model dan nilai aktual. Kesalahan ini, disebut bias, muncul ketika nilai yang diprediksi berbeda dari nilai yang diharapkan atau aktual. Bias dapat terjadi akibat asumsi yang salah dalam proses pembelajaran mesin.

Misalkan Y adalah nilai sebenarnya dari suatu parameter, dan \hat{Y} adalah estimasi Y berdasarkan sampel data. Dalam hal ini, bias dari estimasi \hat{Y} dapat dihitung menggunakan rumus berikut.

$$Bias(\hat{Y}) = E(\hat{Y}) - Y \quad (2.1)$$

di mana $E(\hat{Y})$ merupakan nilai yang diharapkan dari estimasi \hat{Y} . Ini mengukur seberapa baik model cocok dengan data.

1. Bias Rendah: Bias rendah menunjukkan sedikitnya asumsi yang dibuat dalam membangun fungsi target. Dalam situasi ini, model sangat cocok dengan dataset pelatihan.
2. Bias Tinggi: Bias tinggi menunjukkan banyaknya asumsi yang dibuat dalam membangun fungsi target. Pada kasus ini, model tidak cocok dengan data pelatihan.

Model dengan bias tinggi gagal menggambarkan tren dalam dataset. Ini biasanya disebut underfitting, di mana model terlalu disederhanakan. Cara mengurangi bias tinggi adalah sebagai berikut.

1. Menggunakan Model yang Lebih Kompleks: Bias tinggi sering terjadi pada model yang terlalu sederhana. Menggunakan model yang lebih kompleks, seperti jaringan saraf dalam dengan lapisan tersembunyi tambahan, regresi polinomial untuk data non-linear, atau CNN dan RNN untuk jenis data tertentu, dapat membantu model menangkap kompleksitas data yang lebih baik.

2. Tambahkan Fitur: Menambahkan lebih banyak fitur ke dalam dataset pelatihan dapat meningkatkan kompleksitas model dan kemampuannya dalam menangkap pola-pola yang ada dalam data.
3. Reduksi Regularisasi: Teknik regularisasi, seperti L1 atau L2 regularization, dapat membantu mengurangi overfitting dan meningkatkan generalisasi model. Jika model memiliki bias yang tinggi, mengurangi kekuatan regularisasi atau menghapusnya sepenuhnya bisa membantu meningkatkan kinerja model.
4. Perluas Ukuran Data Pelatihan: Meningkatkan jumlah data pelatihan dapat membantu mengurangi bias dengan memberikan lebih banyak contoh kepada model untuk dipelajari dari dataset [44].

2.2.14 Varians

Varians adalah ukuran penyebaran data dari posisi rata-ratanya. Dalam pembelajaran mesin, varians mengacu pada perubahan kinerja model prediktif saat dilatih pada subkumpulan data pelatihan yang berbeda. Varians mengukur variabilitas model dalam respons terhadap subset lain dari dataset pelatihan, yaitu sejauh mana model dapat menyesuaikan diri dengan subset baru dari dataset pelatihan.

Misalkan Y adalah nilai sebenarnya dari variabel target, dan \hat{Y} adalah nilai prediksi dari variabel target. Variansi suatu model dapat diukur sebagai nilai ekspektasi dari kuadrat selisih antara nilai prediksi dan nilai ekspektasi dari nilai prediksi.

$$Variance = E \left[(\hat{Y} - E[\hat{Y}])^2 \right] \quad (2. 2)$$

di mana $E[\hat{Y}]$ adalah nilai yang diharapkan dari nilai yang diprediksi. Di sini, nilai yang diharapkan dirata-ratakan untuk semua data pelatihan. Kesalahan varians bisa rendah atau tinggi.

Varians rendah berarti bahwa model kurang sensitif terhadap perubahan data pelatihan dan dapat menghasilkan estimasi fungsi target yang konsisten dengan subset data yang berbeda dari distribusi yang sama. Ini mengindikasikan underfitting di mana model gagal menggeneralisasi baik pada data pelatihan maupun pengujian. Sebaliknya, varians tinggi berarti model sangat sensitif terhadap perubahan data pelatihan, yang bisa mengakibatkan perubahan signifikan dalam perkiraan fungsi target saat dilatih pada subset data yang berbeda dari distribusi yang sama. Ini terkait dengan overfitting di mana model berkinerja baik pada data pelatihan tetapi buruk pada data uji yang baru. Untuk mengurangi varians, beberapa pendekatan bisa digunakan sebagai berikut.

1. Validasi Silang: Dengan membagi data menjadi set pelatihan dan pengujian beberapa kali, validasi silang membantu mengidentifikasi apakah model cenderung overfitting atau underfitting. Ini juga membantu dalam menyetel hyperparameter untuk mengurangi varians.
2. Pemilihan Fitur: Memilih fitur yang relevan dapat mengurangi kompleksitas model dan mengurangi kesalahan varians.
3. Regularisasi: Menggunakan regularisasi L1 atau L2 dapat mengurangi variasi dalam model pembelajaran mesin.
4. Metode Ensemble: Teknik ansambel seperti bagging, boosting, dan stacking menggabungkan beberapa model untuk meningkatkan kinerja generalisasi, membantu mengurangi variasi.
5. Sederhanakan Model: Mengurangi kompleksitas model, seperti mengurangi parameter atau lapisan dalam jaringan saraf, membantu mengurangi variasi dan meningkatkan kinerja generalisasi.

6. Penghentian Awal: Teknik penghentian awal mencegah overfitting dengan menghentikan pelatihan model deep learning saat performa pada set validasi berhenti membaik [44].