BAB II
TINJAUAN PUSTAKA

2.1 Kecerdasan Buatan

Kecerdasan buatan atau *artificial intelligence* sebenarnya sulit untuk didefinisikan secara tepat. Dalam salah satu penelitian Alan Turing berjudul "*Computing Machinery and Intelligence*", beliau berpendapat bahwa sebuah mesin dianggap cerdas jika tidak sama atau tidak dapat dibedakan dengan manusia, bahkan dengan caranya berbicara. Namun, saat ini *artificial intelligence* lebih mengacu kepada kemampuan mesin untuk berkomunikasi, bernalar, dan beroperasi secara mandiri dengan hal yang baru atau lama dengan cara yang mirip dengan manusia [7]

Mengambil pengertian atau definisi dari *amazon*, kecerdasan buatan merupakan bidang dalam ilmu komputer yang mempelajari pengembangan sistem yang dapat meniru proses berpikir dan bertindak manusia untuk menyelesaikan masalah yang umumnya melibatkan aspek kecerdasan manusia, seperti pembelajaran, prediksi, dan pengenalan gambar. Kecerdasan buatan atau *artificial intelligence* bekerja dengan cara mengumpulkan data yang kemudian dianalisis untuk ditemukan polanya. Menganalisis dan menemukan pola dalam data tidak diprogram oleh manusia secara eksplisit, namun hanya perlu memberikan sekali saja perintah dan program akan berjalan dengan sendirinya. Salah satu tujuan kecerdasan buatan adalah untuk menciptakan sistem yang dapat belajar secara mandiri tanpa memberikan kode atau perintah secara eksplisit dengan cara mengolah data yang diberikan.

Artificial intelligence pertama kali diperkenalkan oleh Alan Turing melalui makalahnya yang membahas kemungkinan mesin dapat berpikir layaknya manusia. Kemudian antara tahun 1957 sampai 1974, komputasi awan atau penyimpanan berbasis *cloud* berkembang sehingga memungkinkan komputer untuk menyimpan data dalam jumlah yang lebih besar dan memprosesnya dengan kecepatan yang lebih tinggi. Pada masa ini, ilmuwan juga terus mengembangkan berbagai teknologi lebih lanjut tentang *machine learning* atau pembelajarn mesin. Tujuan awal dari

penelitian yang dilakukan oleh Alan Turing ini adalah untuk melihat apakah komputer dapat menyalin dan menerjemahkan bahasa lisan atau bahasa manusia.

Pada tanggal 17 februari tahun 1996, *grand master* catur dan pecatur terbaik dunia nomor satu bernama Garry Kasparov bertanding melawan *artificial intelligence* catur yang diberi nama *Deep Blue*. Pada usia 13 tahun *Grand Master* Garry Kasparov sudah memenangkan juara catur junior di Uni Soviet dan menjadi juara catur dunia termuda sepanjang sejarah ketika berusia 22 tahun, karenanya beliau dianggap sebagai pecatur terbaik dan terhebat dunia sepanjang sejarah. Pertandingan diadakan selama enam hari dan enam permainan, dari enam permainan tersebut *Grand Master* Garry Kasparov menang empat kali dan kalah dua kali melawan *Deep Blue*.

Kemudian pada tahun 1997, *Deep Blue* lebih disempurnakan dan bertanding ulang melawan Garry Kasparov. Dari 6 kali permainan, Garry Kasparov memenangkan 2 pertandingan, 3 kekalahan dan 1 kali seri yang membuat *Deep Blue* akhirnya memenangkan pertandingan. Dengan kemenangan *Deep Blue* ini, membuat banyak orang terkejut karena catur yang selama ini dianggap sebagai permainan yang membutuhkan intuisi, kreatifitas dan kemampuan berpikir secara strategis, di mana hal ini dianggap hanya dimiliki oleh manusia. Kemenangan *Deep Blue* tersebut juga membuka prespektif baru bahwa mesin dapat mengerjakan tugas seperti manusia atau bahkan mampu menyaingi dan melampaui manusia.

Saat ini *artificial intelligence* sudah menjadi hal yang sangat lumrah dan bahkan digunakan oleh semua orang dalam kegiatan sehari-hari mengingat banyaknya manfaat yang dimiliki oleh *artificial intelligence*. Contoh penerapan *artificial intelligence* yang digunakan dalam sehari-hari adalah penggunaan asisten virtual. Asisten virtual berbasis *artificial intelligence* adalah *software* yang menggunakan *Natural Language Processing* (*NLP*) untuk mengikuti perintah secara interaktif yang mampu mengeluarkan suara, teks atau bahkan *hologram* [8]. *Google Assistant* adalah salah satu aplikasi asisten virtual yang mampu menjawab pertanyaan, mengatur jadwal dan bahkan mengendalikan perangkat rumah yang sudah terintegrasi dengan internet. Selain *Google Assistant*, terdapat juga aplikasi asisten

virtual bernama *ChatGPT* yang saat ini banyak digunakan oleh masyarakat mengingat kemampuannya untuk menjawab soal atau bahkan mengerjakan esai.

Selain digunakan sebagai asisten virtual, *artificial intelligence* juga dapat dikembangkan untuk membantu menganalisis data. Dengan menggunakan *artificial intelligence*, pengolahan data dengan jumlah yang besar dapat dilakukan dengan lebih akurat dan dengan kecepatan yang luar biasa yang bahkan melebihi kemampuan manusia. Selain itu, *artificial intelligence* dapat terus belajar untuk meningkatkan tingkat akurasinya dan mengurangi persentase *error* sehingga menghasilkan *output* yang lebih akurat. Selain untuk menganalisis data, *artificial intelligence* juga dapat dikembangkan untuk mengenali pola dan tren dalam data serta mnyediakan rekomendasi yang berdasarkan pada analisis yang cerdas dan terstruktur dengan menggunakan *computer vision*, *deep learning* dan *natural language processing* [9].

Dengan memanfaatkan data yang tersedia, *artificial intelligence* juga dapat berperan dalam dunia medis, terutama dalam mendiagnosis penyakit dengan mengidentifikasi pola dalam *dataset* [10]. Dalam dunia kesehatan, ketepatan dalam memprediksi suatu penyakit memerlukan keputusan yang efektif dan keakuratan prediksi akan menjadi bagian yang sangat penting dalam menentukan keputusan. Salah satu pemanfaatan *artificial intelligence* dalam dunia medis adalah menganalisis gambar radiologi seperti *CT scan* yang kemudian dapat teridentifikasi sebuah penyakit [11]. Dengan pemanfaatan menggunakan *artificial intelligence* tersebut, maka dapat membantu dokter untuk meningkatkan deteksi dini penyakit serta memungkinkan untuk melakukan tindakan yang cepat dan tepat sehingga penderita dapat berkurang dan harapan hidup meningkat [12].

2.2 *Machine Learning*

Pada abad 20-an awal, sejumlah ilmuwan matematika memperkenalkan dasar-dasar *machine learning* dan konsep-konsep yang mendasarinya. Konsep ini kemudian terus dikembangkan oleh banyak ilmuwan seiring berjalannya waktu. Contoh terkenal dari penerapan *machine learning* adalah *Deep Blue*, yang dikembangkan oleh perusahaan IBM pada tahun 1996. *Deep Blue* dirancang untuk belajar dan

bermain catur, dan telah diuji dalam pertandingan melawan juara catur profesional, di mana *Deep Blue* berhasil meraih kemenangan [13].

Machine learning adalah salah satu dari cabang kecerdasan buatan yang dapat didefinisikan sebagai algoritma atau teknik komputasi yang diprogram untuk belajar dari data sehingga dapat melakukan prediksi dengan tingkat akurasi yang tinggi. Kualitas dan kuantitas data memiliki peran yang penting dalam meningkatkan tingkat akurasi dan menghasilkan prediksi yang optimal. Dari hasil tersebut, maka komputer dapat melakukan tugas-tugas tanpa program eksplisit dari manusia [7]

Machine learning adalah salah satu cabang dari ilmu komputer yang merancang algoritma untuk memungkinkan komputer memperoleh pengetahuan dari data, sehingga sering disebut sebagai *'learn from data'*. Dengan demikian, *machine learning* melibatkan pemrograman komputer yang memanfaatkan data historis untuk melatih model, guna mencapai performa optimal dalam menganalisis informasi dari kumpulan data. Menurut Tom M. Mitchell, *machine learning* adalah program komputer yang mampu memanfaatkan pengalaman sebagai bahan pembelajaran ketika menjalankan tugas tertentu dengan kinerjanya yang dievaluasi berdasarkan hasil dari pengalaman tersebut [14].

Saat ini *machine learning* sangat banyak berperan dalam membantu manusia dalam kehidupan. Fitur keamanan untuk membuka *smartphone* dengan *face unlock* saat ini juga didasari dengan konsep *machine learning* dengan mendeteksi wajah, sehingga memudahkan dan meningkatkan persentase keamanan [15]. Bahkan presensi kelas dapat memanfaatkan *machine learning* dengan melakukan deteksi wajah untuk melakukan presensi [16]. Kemudian, saran penjualan yang diberikan di layanan *e-commerce* juga merupakan hasil pengolahan *machine learning* dengan mengidentifikasi pola perilaku pengguna dan karakteristik produk, yang telah terbukti memperbaiki kualitas rekomendasi produk. Model yang dibuat dapat lebih memahami preferensi pengguna, sehingga memberikan saran yang lebih tepat dan menarik. Hal ini dapat berkontribusi pada peningkatan interaksi pengguna,

memperluas basis pelanggan, serta mendorong peningkatan konversi penjualan di *platform e-commerce* [17].

Pada penelitian kali ini, peneliti akan menggunakan algoritma yang dikategorikan sebagai *supervised learning* dikarenakan *dataset* pada penelitian kali ini lebih berfokus kepada kasus klasifikasi dengan mengukur tingkat akurasi setiap model. *Dataset* penelitian ini juga memiliki sebuah label dan memang dikhususkan untuk digunakan sebagai klasifikasi, sehingga *supervised learning* menjadi pilihan yang tepat. Algoritma yang akan digunakan pada penelitian kali ini antara lain yaitu, *neural network, random forest* dan *support vector machine*.

2.2.1 *Supervised Learning*

Supervised learning adalah teknik melatih mesin menggunakan *dataset* yang diberi label, maksudnya adalah data sudah memiliki jawaban yang benar yang kemudian mesin belajar dari data tersebut, sehingga jika diberikan data yang baru, maka mesin dapat memprediksi jawabannya. Contohnya adalah ketika ingin memprediksi harga mobil dengan memasukkan *variable input* tahun kendaraan, jenis mobil dan bahan bakar. Sementara *variable* harga digunakan sebagai *variable output* yang merupakan label, maka kita dapat membuat model yang bisa memprediksi harga sebuah mobil. *Supervised learning* dibagi lagi menjadi 2, yaitu klasifikasi dan regresi [18].

Regresi merupakan salah satu metode dalam *machine learning* yang diterapkan untuk memperkirakan nilai kontinu yang berarti dapat mengambil dari angka berapapun termasuk pecahan desimal. Regresi mempelajari hubungan antara variabel independen (fitur) dan variabel dependen (target) untuk memprediksi nilai baru. Contoh penggunaan algoritma ini adalah untuk memprediksi harga rumah dengan menggunakan *dataset* yang terdiri dari variabel dependen atau target. Contoh algoritma yang umum digunakan untuk regresi adalah *linear regression, decision tree regression* dan *random forest rggression.*

Klasifikasi adalah metode dalam *machine learning* yang digunakan untuk mengelompokkan data ke dalam kategori atau kelas tertentu [19]. Algoritma

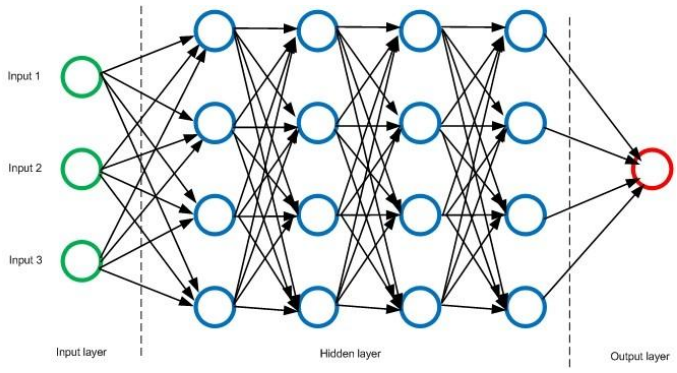
klasifikasi mempelajari pola dari data yang sudah diberi label dan kemudian menggunakan pola tersebut untuk mengklasifikasikan data baru [20]. Klasifikasi dapat digunakan untuk mengklasifikasikan *email spam* dan mengidentifikasi penyakit dengan *dataset* yang memiliki atribut target. Contoh algoritma yang umum digunakan untuk klasifikasi adalah *decision tree, random forest, support vector machine* dan *neural network*.

2.2.2 Algoritma *Neural Network*

Neural network pada dasarnya meniru kemampuan otak manusia dalam melakukan proses kompleks dan menghasilkan *output.* [21]. Contohnya adalah anak-anak yang sedang belajar, sebelumnya anak-anak tidak mengetahui hal apapun, kemudian orang dewasa mengajarkan banyak hal yang membuat otak anak-anak dapat mengetahui banyak hal. *Neural network* mengadopsi kemampuan otak manusia yang kemudian diinterpretasikan dan diterapkan dalam bahasa mesin untuk menyelesaikan proses perhitungan [22].

Jaringan saraf tiruan atau lebih dikenal *neural network* memiliki sejumlah keunggulan yang menjadikannya sangat bermanfaat dalam berbagai aplikasi *artificial intelligence*. Salah satu kelebihan utamanya adalah kemampuannya untuk mengenali pola dan membuat prediksi berdasarkan data yang diberikan. Hal ini memungkinkan *neural network* menyelesaikan tugas-tugas kompleks dan tidak terstruktur dengan lebih efisien. Selain itu, *neural network* juga sangat fleksibel dan dapat digunakan dalam berbagai pengembangan kecerdasan buatan, seperti pengenalan gambar, pengenalan suara dan menganalisis sebuah harga barang. Kemampuan ini didapat dari struktur lapisan tersembunyi dalam *neural network* atau disebut *hidden layer*, yang memungkinkan pemrosesan data pada tingkat kompleksitas yang tinggi. Lapisan-lapisan ini bertindak sebagai *filter* yang mengidentifikasi fitur dan pola yang lebih rumit secara bertahap, sehingga *neural network* mampu menangani masalah yang kompleks dan memprediksi hasil dengan lebih akurat. *Neural network* juga mempunyai kelebihan dalam efisiensi waktu dan akurasi.

Setelah dilatih dengan *dataset* yang berkualitas, *neural network* mampu menganalisis data dan memberikan hasil dengan cepat serta akurat, sehingga mengurangi risiko kesalahan atau *error*. Keunggulan ini sangat penting dalam aplikasi yang memerlukan keputusan cepat dan tepat, seperti sistem pengenalan wajah atau diagnosis medis. Secara keseluruhan, kelebihan *neural network* terletak pada kemampuannya untuk belajar, beradaptasi, fleksibilitas dalam berbagai aplikasi, serta efisiensi dan akurasi dalam pemrosesan data.



Gambar 2. 1 Arsitektur *neural network*

Setiap *neuron* hanya memiliki satu *input* dan satu *output*. Data *input* merupakan data yang belum diproses atau diolah. sementara data *output* merupakan hasil dari proses atau pengolahan data yang sebelumnya sudah dimasukkan. Sementara *hidden layer* adalah lapisan tersembunyi dalam *neural network* karena berada di antara *input* dan *output* yang berfungsi sebagai perantara. Pemanfaatan algoritma *neural network* dalam *deep learning* seperti *convolutional neural network* dan *recurrent neural network* sangat efektif dalam menyelesaikan masalah klasifikasi yang kompleks. Contoh pemanfaatan *neural network* dalam *deep learning* adalah pengenalan objek gambar yang dapat digunakan untuk medis, keamanan dan bahkan industri. Rumus yang digunakan untuk melakukan prediksi berdasarkan independen fitur adalah $y_{pred} = \sigma(z)$. Sementara z didapatkan dari perhitungan $z = w_n \cdot x_n + b$

Dalam dunia medis, pemanfaatan algoritma ini dapat digunakan untuk mendeteksi tumor pada gambar *X-Ray*, *MRI* atau bahkan mendeteksi sel kanker [11]. Dalam dunia keamanan, pemanfaatan *neural network* digunakan untuk proses

otentikasi perangkat ketika akan mengakses *smartphone* atau bahkan *m-banking*. Dengan adanya fitur ini, maka keamanan diharapkan dapat lebih terjaga.

Meskipun memiliki potensi besar, *neural network* juga memiliki beberapa kelemahan, salah satunya adalah kebutuhan akan data yang sangat banyak dan berkualitas tinggi untuk pelatihan yang sering kali sulit diperoleh. Selain itu, model ini membutuhkan komputasi yang intensif, memerlukan perangkat keras yang kuat, serta waktu pelatihan yang cukup lama. *Neural network* juga dikenal sebagai "kotak hitam," karena sulit untuk memahami bagaimana model membuat keputusan yang bisa menjadi masalah dalam aplikasi yang memerlukan transparansi. Selain itu, *neural network* juga rentan terhadap *overfitting*, terutama ketika data pelatihan tidak cukup besar atau bervariasi. Meskipun demikian, *neural network* tetap banyak digunakan karena kemampuannya dalam menyelesaikan tugas-tugas kompleks dan *non-linear*.

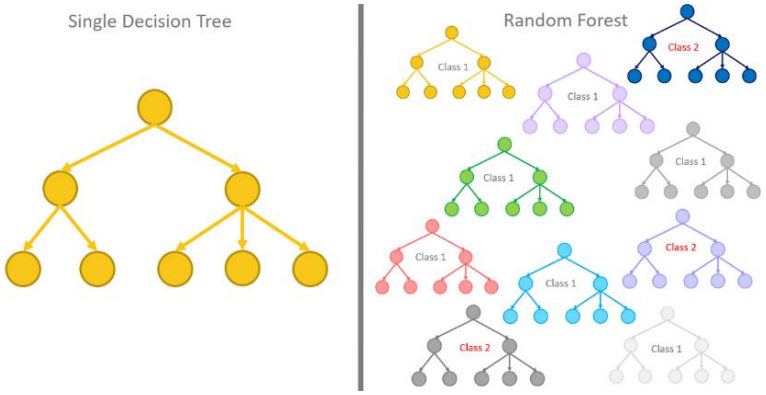
Terlepas dari kelebihan dan kekurangan tersebut, *neural network* cocok digunakan untuk tugas klasifikasi dengan data yang terstruktur atau berupa tabel. Salah satu contoh penggunaanya adalah ketika akan mendeteksi penyakit yang *dataset* nya berupa tabel seperti riwayat penyakit. Dari *dataset* tersebut, algoritma *neural network* akan mengolah data dan memberikan prediksinya sehingga mengeluarkan *output* sesuai dengan data yang ada. Dalam penelitian kali ini, peneliti akan menggunakan *dataset* penderita penyakit diabetes.

2.2.3 Algoritma *Random Forest*

Random forest merupakan sebuah algoritma yang *output* nya didapat dari gabungan hasil dari *decision tree* untuk mendapatkan hasil yang lebih akurat [23]. Jumlah *tree* yang seimbang akan menghasilkan akurasi yang tinggi dan menghindari *overfitting* [24]. Cara kerja *random forest* yaitu, algoritma secara acak akan memilih sejumlah sampel dari data *training*. Untuk setiap sampel, algoritma membangun pohon keputusan dan setiap pohon keputusan merupakan model yang dihasilkan dari sampel data tersebut [25]. Setelah semua pohon keputusan terbentuk, algoritma mengintegrasikan hasil dari masing-masing pohon untuk menentukan prediksi akhir. Jika yang diteliti adalah regresi, maka dihitung berdasarkan nila rata-rata,

sementara jika menggunakan klasifikasi, maka mayoritas suara dari pohon keputusan yang akan menjadi *final score.*

Jika dianalogikan dengan manusia, maka dapat dianalogikan seperti ingin meminta rekomendasi produk dari beberapa orang yang kemudian dari pendapat setiap orang dapat ditarik kesimpulan manakah produk yang terbaik. Rekomendasi dari orang-orang itu adalah hasil dari *decision tree* dan kesimpulan produk yang dihasilkan adalah *output* dari *random forest*.



Gambar 2. 2 Arsitektur *random forest*

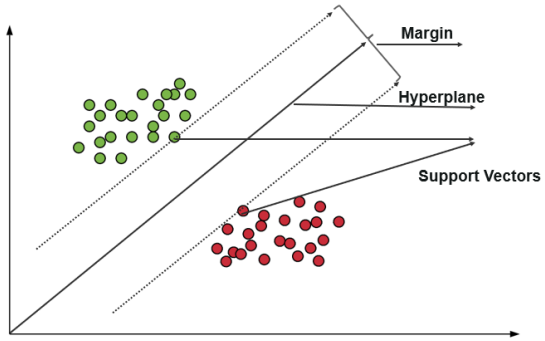
Salah satu kelebihan *random forest* adalah memiliki ketahanan yang baik terhadap *outlier* dan *noise* dalam data. *Outlier* adalah data yang nilai numeriknya jauh berbeda dari nilai-nilai lainnya dalam suatu *dataset*. Data ini bisa sangat tinggi atau sangat rendah dibandingkan dengan data lainnya yang bisa terjadi ketika proses pengumpulan data. *Noise* adalah gangguan acak atau kesalahan dalam data yang menyebabkan data menjadi kurang akurat atau tidak sesuai dengan konteks sebenarnya. *Outlier* dan *Noise* dapat mengakibatkan data menjadi tidak relevan dan mempengaruhi hasil analisis atau prediksi. Perhitungan nilai akhir dari *random forest* adalah $\frac{1}{k} \sum_{i=1}^{k} i$

Algoritma *random forest* mampu memproses dan menganalisis data yang memiliki jumlah variabel atau fitur yang sangat banyak. Algoritma *random forest* juga dapat bekerja tanpa bantuan manusia untuk menentukan variabel mana yang signifikan dan relevan untuk digunakan dalam analisis. Algoritma *random forest*

juga mengurangi kesalahan pada satu pohon keputusan dengan mengandalkan rata-rata hasil dari beberapa pohon. Ini membuat *random forest* menjadi metode yang dapat diandalkan dalam berbagai aplikasi, termasuk klasifikasi dan regresi.

2.2.4 Algoritma *Support Vector Machine*

Support vector machine merupakan salah satu teknik dalam *machine learning* yang sangat populer baik untuk memproses tugas regresi maupun tugas klasifikasi karena cukup efektif untuk menangani data yang berdimensi tinggi [26]. Tujuan utama *support vector machine* adalah untuk menentukan *hyperplane* (bidang pemisah) terbaik yang dapat memisahkan data menjadi dua kelas atau lebih. *Hyperplane* ini harus memaksimalkan jarak ke data terdekat dari setiap kategori yang disebut sebagai *margin* [27].

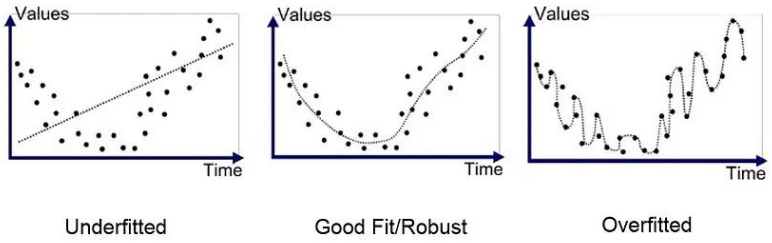


Gambar 2. 3 contoh pemisahan *support vector machine*

Kelebihan dari *support vector machine* terletak pada kesederhanaan dan fleksibilitasnya yang efektif dalam menangani berbagai masalah klasifikasi. *Support vector machine* juga dikenal memberikan performa prediksi yang konsisten, bahkan dalam penelitian dengan ukuran sampel yang terbatas [28]. Salah satu kelebihan lain dari algoritma *support vector machine* adalah *robust* atau tahan terhadap *overfitting*.

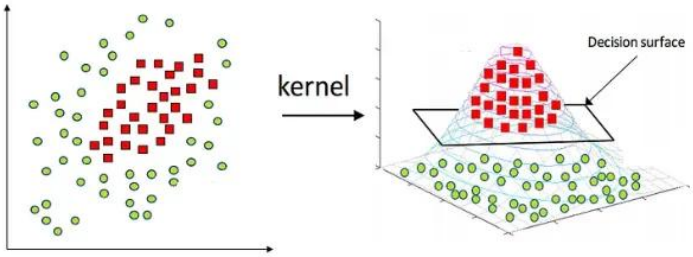
Overfitting adalah kondisi ketika model *machine learning* terlalu fokus mempelajari detail dari data pelatihan, termasuk *noise* atau variasi acak yang tidak relevan, hal ini membuat model menjadi sangat spesifik terhadap data pelatihan, sehingga tidak mampu menggeneralisasi dengan baik ketika diberikan data baru yang belum pernah ditemui sebelumnya. Dengan menggunakan *margin* yang lebih

lebar, *support vector machine* memastikan model tidak terlalu menyesuaikan data pelatihan, sehingga meningkatkan kemampuannya dalam melakukan generalisasi pada data baru.



Gambar 2. 4 Visualisasi *overfitting*

Support vector machine dapat menggunakan *kernel trick*, yang memungkinkan untuk memodelkan hubungan *non-linear* dengan cara memproyeksikan data ke dimensi ruang yang lebih tinggi. Ini sangat membantu ketika data tidak bisa dibagi secara *linear* di ruang aslinya.



Gambar 2. 5 Visualisasi *kernel trick*

Di samping itu semua, ketika menggunakan *support vector machine* sebagai algoritma, maka waktu dan daya yang dibutuhkan ketika *training* model akan cukup lama, hal ini juga dianggap sebagai kekurangan yang dimiliki oleh algoritma ini. *Support vector machine* dapat melakukan klasifikasi data dengan persamaan sebagai berikut $f(x, w, b) = sign(wTxi + b)$.

2.2.5 Akurasi

Akurasi adalah metrik yang digunakan untuk menilai sejauh mana model kita mampu memprediksi kelas yang tepat dari suatu data. Sederhananya, akurasi adalah

persentase dari total prediksi yang benar. Akurasi dapat dihitung dengan rumus sebagai berikut: $akurasi = \frac{jumlah\ prediksi\ benar}{jumlah\ total\ data}$

2.2.6 *Precision*

Precision merupakan metrik yang menghitung seberapa besar jumlah prediksi positif yang tepat dibandingkan dengan seluruh prediksi positif yang dikeluarkan oleh model. [29]. Metrik ini penting dalam klasifikasi, terutama jika data tidak seimbang atau jika kesalahan dalam memprediksi sesuatu sebagai positif, padahal salah (*false positive*), maka akan dapat menimbulkan dampak yang besar.

Jika nilai *precision* yang didapatkan adalah tinggi, maka ini menunjukkan bahwa model jarang sekali melakukan kesalahan terhadap klasifikasi data yang positif sebagai negatif. Sementara jika nilai *precision* nya rendah, maka ini menunjukkan bahwa model sering salah mengklasifikasikan data negatif sebagai data positif. Rumus menghitung *precision* adalah: $Precision = \frac{TP}{TP+FP}$

TP (*True Positive*): total data yang sebenarnya positif dan juga diprediksi positif.
FP (*False Positive*): total data yang sebenarnya negatif namun diprediksi positif.

2.2.7 *Recall*

Recall adalah ukuran yang menunjukkan seberapa baik model sukses menemukan semua data yang benar-benar termasuk kategori positif. Sederhananya, recall mengukur seberapa baik model menangkap semua contoh yang seharusnya diprediksi sebagai positif. [29]. Jika nilai *recall* yang didapatkan adalah tinggi, maka ini menunjukkan bahwa model sangat baik dalam menemukan label atau kasus yang positif. Sementara jika nilai *precision* nya rendah, maka ini menunjukkan bahwa model melewatkan banyak label yang positif. Rumus menghitung *precision* adalah: $Recall = \frac{TP}{TP+FN}$

TP (*True Positive*): total data yang seharusnya positif dan juga diprediksi positif.
FN (*False Negative*): total data yang pada kenyataannya positif, namun diprediksi sebagai negatif.

2.2.8 *F1-Score*

F1-score adalah metrik evaluasi yang mengkombinasikan *precision* dan *recall* dalam satu angka. [29]. Ini berguna ketika kita ingin menyeimbangkan kedua metrik antara *recall* dan *precision*. Jika nilai dari *f1-score* tinggi, maka itu menandakan adanya keseimbangan yang optimal antara *precision* dan *recall* pada model. Sementara jika nila *f1-score* nya rendah, maka ini menunjukkan bahwa mode masih memiliki kekurangan dalam *precision* ataupun *recall*. Untuk menghitung f-1 score, maka kita menggunakan formula sebagai berikut: $F1\ Score = \frac{Precision*recall}{Precision+Recall}$

2.2.9 *K-Fold Cross Validation*

K-Fold cross validation adalah teknik untuk mengevaluasi model yang membagi *dataset* menjadi K bagian (*folds*) yang seimbang. Pada setiap iterasi, satu *fold* digunakan sebagai data pengujian, sementara *K-1 fold* lainnya dimanfaatkan untuk pelatihan. Tahapan ini dilakukan sebanyak K kali, sehingga setiap *fold* berfungsi data uji sekali. Teknik ini memungkinkan pengukuran performa model yang lebih akurat dengan mengurangi varian yang dapat timbul dari pembagian data yang tidak merata, serta memberikan estimasi yang lebih baik terhadap akurasi model [30]. Pada penelitian ini, *fold* yang digunakan berjumlah 10, sehingga bisa disebut *ten-fold cross validation. Ten-fold cross validation* bekerja dengan mengiterasikan dan membagi data untuk pelatihan berjumlah 9 bagian dan 1 bagian lainnya untuk pengujian, kemudian setiap iterasi diulang sebanyak 10 kali.

2.3 *Python*

Python adalah bahasa pemrograman yang memanfaatkan *interpreter* untuk menjalankan kode program secara langsung. Dengan menggunakan *interpreter* ini, *Python* bisa digunakan di berbagai sistem operasi, seperti *windows, linux,* dan lain-lain. *Python* menggabungkan beberapa paradigma pemrograman, seperti pemrograman prosedural yang mirip dengan *C,* pemrograman berorientasi objek seperti *java,* dan pemrograman fungsional yang terinspirasi oleh *Lisp.* Pendekatan ini memungkinkan para pengembang untuk lebih fleksibel dalam mengembangkan berbagai jenis aplikasi dengan *Python.*

Python juga terintegrasi dengan sistem *database* dan memiliki kemampuan untuk membaca serta memodifikasi *file*, menjadikannya pilihan populer untuk pembuatan prototipe atau pengembangan aplikasi perangkat lunak yang cepat dan andal. Di samping itu, *Python* banyak digunakan oleh para peneliti karena keterampilannya dalam mengelola data dalam jumlah besar dan melakukan perhitungan matematis yang rumit [31]. *Python* dapat digunakan dalam berbagai bidang, mulai dari pengembangan *web,* aplikasi *desktop, machine learning,* hingga *telemedicine* [31].

2.4 Penyakit Diabetes

Diabetes melitus terjadi akibat kelainan pada metabolisme yang memengaruhi fungsi pankreas yang mengarah pada peningkatan tingkat gula darah yang tinggi atau *hiperglikemia*, disebabkan oleh penurunan produksi *insulin*. Penyakit ini dapat menyebabkan berbagai komplikasi, baik *makrovaskular* maupun *mikrovaskular*. Selain itu, diabetes melitus juga dapat memicu gangguan *kardiovaskular* yang berpotensi berbahaya jika tidak segera ditangani, sehingga dapat meningkatkan risiko *hipertensi* dan *infark miokard*.

Diabetes melitus adalah kondisi *metabolik* yang ditandai dengan *hiperglikemia* (kadar gula darah tinggi) karena gangguan pada produksi atau fungsi *insulin*. Penyakit ini menjadi perhatian utama dalam dunia kesehatan karena termasuk penyakit tidak menular yang menjadi fokus penanganan global. Jumlah penderita diabetes terus meningkat setiap tahun dan diperkirakan akan bertambah secara signifikan di masa depan. World Health Organization (WHO) sendiri mencatat bahwa pada tahun 2000, diperkirakan ada sekitar 171 juta orang yang menderita diabetes di seluruh dunia, kemudian International Diabetes Federation (IDF) juga memperkirakan angka ini akan melonjak hingga 643 juta pada tahun 2030, hal ini mengarah pada fakta bahwa jumlah penderita diabetes melitus setiap dekadenya naik secara signifikan [32].

Berolahraga memberikan hasil positif yang besar bagi individu yang menderita diabetes melitus, seperti membantu mengurangi kadar glukosa darah, menghindari kegemukan atau obesitas, mengurangi risiko komplikasi, serta memperbaiki

masalah kadar lemak dalam darah dan mengontrol tekanan darah. Diabetes melitus sering dianggap penyakit yang tidak terdeteksi karena banyak penderita yang tidak menyadari kondisi mereka hingga komplikasi muncul. Beberapa beberapa faktor yang dapat memicu resistensi insulin antara lain yaitu, obesitas atau kelebihan berat badan, kelebihan *glukokortikoid*, kelebihan hormon pertumbuhan (*akromegali*), kehamilan dan diabetes *gestasional* [32].

Diabetes tipe 1 terjadi ketika sel beta pankreas terhancurkan karena reaksi autoimun, yang menyebabkan tubuh tidak dapat memproduksi insulin. Walaupun glukosa yang berasal dari makanan tetap ada dalam darah dan menyebabkan *hiperglikemia* pasca makan, tubuh tidak dapat menyimpan glukosa tersebut di hati. Ketika kadar gula darah cukup tinggi, ginjal tidak bisa menyerap seluruh glukosa yang sudah disaring, yang akhirnya menyebabkan glukosa terbuang dalam urin, kondisi yang dikenal sebagai kencing manis. Proses ini juga menyebabkan ekskresi limbah berlebih, termasuk *elektrolit*, yang dikenal dengan *diuresis osmotik*.

Hilangnya cairan secara berlebihan dapat menyebabkan sering buang air kecil (*poliuria*) dan rasa haus yang tidak biasa (*polidipsia*). Selain itu, kurangnya insulin dapat mengacaukan metabolisme protein dan lemak yang pada akhirnya menyebabkan menurunnya berat badan. Dalam keadaan kekurangan insulin, tubuh tidak mampu menyimpan kelebihan protein di jaringan. Akibatnya, metabolisme lemak akan meningkat drastis, khususnya di antara waktu makan [32].

Faktor gaya hidup seseorang serta genetik sering kali memengaruhi terjadinya diabetes. Lingkungan sosial dan akses terhadap layanan kesehatan juga berperan dalam munculnya diabetes serta komplikasinya. Penyakit ini dapat menyebabkan gangguan pada berbagai organ tubuh seiring waktu, yang dikenal dengan istilah komplikasi. Diabetes dapat menyebabkan komplikasi yang terbagi dalam dua kategori, yaitu *mikrovaskuler* dan *makrovaskuler*. Komplikasi *mikrovaskuler* meliputi kerusakan pada saraf (*neuropati*), ginjal (*nefropati*), dan mata (*retinopati*). Penyakit ini umumnya ditemukan pada individu berusia 40 hingga 60 tahun. [32].

2.5 *Google Collaboratory*

Google Collaboratory adalah sebuah *platform* berbasis *cloud* yang memfasilitasi penulisan dan eksekusi kode *Python* secara langsung melalui *browser* tanpa perlu instalasi *software* tambahan. *Google Collaboratory* sangat mirip dengan *jupyter notebook* dan sering digunakan untuk keperluan *data science* dan *machine learning*. Berikut adalah beberapa fitur utama *Google Collaboratory*:

a. Akses gratis: *Google Collaboratory* dapat digunakan secara gratis tanpa perlu membayar biaya untuk berlangganan
b. *Interface* yang sederhana: *Google Collaboratory* memiliki *interface* yang intuitif dan mudah digunakan sehingga sangat cocok untuk orang yang baru mengenal bahasa pemrograman *Python*.
c. Integrasi dengan *Google drive*: *Google Collaboratory* dapat disimpan di *Google Drive* dan proses penyimpanan serta pemanggilannya pun mudah, sehingga mengurangi penyimpanan dalam *device*. *Dataset* yang diperlukan untuk pelatihan model *machine learning* juga dapat disimpan di *Google Drive* dan langsung dapat dipanggil dengan *Google Collaboratory*.
d. Kolaborasi: Pekerjaan yang sedang dikerjakan dengan *Google Collaboratory* juga dapat dikerjakan secara *real-time* oleh partner yang memiliki izin, sehingga memudahkan dalan menyelesaikan project
e. *GPU* dan *TPU*: *Google Collaboratory* menyediakan akses ke *GPU* dan *TPU* untuk mempercepat pelatihan model *machine learning*.

2.6 Kaggle

Kaggle adalah platform komunitas daring yang didirikan pada tahun 2010 oleh Anthony Goldbloom yang kemudian diakuisisi oleh Google pada tahun 2017. Kaggle juga menyediakan berbagai kegiatan, dengan salah satu yang paling populer adalah kompetisi *machine learning*. Selain itu, Kaggle juga memungkinkan anggota komunitas untuk menulis, berbagi kode serta dan berbagai topik yang berhubungan dengan data. *Data scientist* juga dapat memperoleh penghasilan dari proyek-proyek yang ditawarkan di platform ini. Hingga saat ini, Kaggle memiliki lebih dari 1000 *dataset*, 170.000 postingan di forum, dan setidaknya 250 *kernel*

berkualitas tinggi. Meskipun format file *CSV* paling sering digunakan di Kaggle, beberapa *dataset* juga tersedia dalam format *JSON*, *SQLite*, *archive*, dan *BigQuery*.

Kaggle memiliki proses verifikasi dan validasi untuk *dataset* yang diunggah, sehingga membantu memastikan bahwa data tersebut akurat dan dapat dipercaya. Hal Ini penting karena berfungsi untuk menjaga integritas data serta membangun kepercayaan pengguna terhadap *platform*. Kaggle juga menerapkan kebijakan privasi yang ketat untuk melindungi data pengguna, ini termasuk perlindungan terhadap data pribadi yang mungkin ada dalam *dataset*, memastikan bahwa informasi sensitif diproses dengan hati-hati dan sesuai dengan regulasi yang berlaku.

2.7 Penelitian terkait

Penelitian yang dilakukan oleh Dwi Yuni Utami, Elah Nurlelah, Fuad Nur Hasan pada tahun 2021 dengan judul penelitian "*Comparison of Neural Network Algorithms, Naive Bayes and Logistic Regression To Find The Highest Accuracy In Diabetes*" dan menggunakan algoritma *neural network, Naïve Bayes, logistic regression,* didapatkan bahwa masing-masing algoritma memiliki tingkat akurasi yang berbeda. Algoritma *neural network* memiliki tingkat akurasi mencapai 69,27%, *Naïve Bayes* 73,87% dan *logistic regression* memiliki tingkat akurasi yang paling tinggi dengan tingkat akurasi 75,78%. Dari hasil penelitian tersebut dapat disimpulkan bahwa algoritma *logistic regression* dengan *dataset* berjumlah 768 adalah algoritma dengan nilai akurasi tertinggi [6].

Penelitian lainnya dengan judul "Deteksi Penyakit Diabetes Melitus Menggunakan Algoritma *Decision Tree Model Arsitektur C4.5*" dan ditulis oleh Achmad Afifuddin, Lukman Hakim pada tahun 2023 menghasilkan penelitian tingkat akurasi mencapai 96% dengan menggunakan algoritma *c4.5* dengan *dataset* berjumlah 2000 dan 5 variabel. Untuk hasil akurasi data *testing* dan data *validation*, masing-masing mendapatkan akurasi 90%. Sementara hasil *precision*, *recall*, dan f1-score adalah 99%, 95%, dan 97%. Pada penelitian ini pun dapat disimpulkan bahwa semakin banyak *dataset* yang digunakan, maka akan semakin akurat prediksi yang dihasilkan [33].

Penelitian lain dengan judul "Prediksi Penyakit Diabetes Menggunakan Algoritma *Support Vector Machine* (*SVM*)" yang diteliti oleh Hovi Sohibul Wafa, Asep Id Hadiana, Fajri Rakhmat Umbara pada tahun 2022 menghasilkan nilai akurasi 91,2% dengan menggunakan *dataset* berjumlah 1017. Penelitian ini menggunakan algoritma *support vector machine* dan *forward selection* dimana 80% dari data digunakan sebagai data latih dan 20% digunakan untuk data uji. Untuk nilai *precision* pada penelitian ini didapatkan dengan 93%, 94,3% untuk *recall* dan 93,7% untuk *f1-score [27]*.

Penelitian yang dilakukan oleh Muhammad Fahrul Aditya, Andri Pramuntadi, Dhina Puspasari Wijaya, Yanuar Wicaksono pada tahun 2024 dengan judul "Implementasi Metode *Decision Tree* pada Prediksi Penyakit Diabetes Melitus Tipe 2" mendapatkan hasil tingkat akurasi mencapai 92% dengan menggunakan 200 *dataset* dan 5 atribut. Penelitian ini menggunakan algoritma *decision tree* dengan melakukan pembagian data sebesar 80:20 atau 160 data *training* dan 40 data *testing* menggunakan metode *train test split [34]*.

Penelitian terkait perbandingan algoritma klasifikasi yang dilakukan oleh Yunan Fauzi Wijaya, Agung Triayudi pada tahun 2023 dengan judul "Perbandingan Algoritma Klasifikasi *Data Mining* Pada Prediksi Penyakit Diabetes" menghasilkan bahwa algoritma *Naïve Bayes* mendapatkan tingkat akurasi 75%, *K-Nearest Neighbor* 80,6% dan *c4.5* mencapai 91,8%. Dengan hasil itu dapat disimpulkan bahwa algoritma *c4.5* memiliki nilai akurasi tertinggi melalui penelitian ini [35].

Tabel 2. 1 Penelitian terkait

No	Nama, Tahun	Judul Penelitian	Metode	Akurasi	Jumlah *Dataset*
1	Dwi Yuni Utami, Elah Nurlelah, Fuad Nur Hasan (2021)	*Comparison of Neural Network Algorithms, Naive Bayes, and Logistic Regression To Find*	*Neural Network,*	69,27%	768
			Naïve Bayes,	74,87%	
			Logistic Regression	75,78%	

		The Highest Accuracy In Diabetes			
2	Achmad Afifuddin, Lukman Hakim (2023)	Deteksi Penyakit Diabetes Melitus Menggunakan Algoritma *Decision Tree Model Arsitektur C4.5*	Algoritma *C4.5*	96%	2000
3	Hovi Sohibul Wafa1, Asep Id Hadiana, Fajri Rakhmat Umbara (2022)	Prediksi Penyakit Diabetes Menggunakan Algoritma *Support Vector Machine* (SVM)	*Support Vector Machine*	91,2%	1017
4	Muhammad Fahrul Aditya, Andri Pramuntadi, Dhina Puspasari Wijaya, Yanuar Wicaksono (2024)	Implementasi Metode *Decision Tree* pada Prediksi Penyakit Diabetes Melitus Tipe 2	*Decision Tree*	92%	200
5	Yunan Fauzi Wijaya, Agung Triayudi (2023)	Perbandingan Algoritma Klasifikasi *Data Mining* Pada Prediksi Penyakit Diabetes	*Naïve Bayes*	75%	-
			K-Nearest Neighbor	80,6%	
			C4.5	91,8%	
6	Yunada Wiratama, RZ Abdul Aziz	Perbandingan Prediksi Penyakit Stunting Balita Menggunakan Algoritma SVM dan Random Forest	*Support Vector Machine*	99.51%	22855
			Random Forest	99.97%	