BAB II
KAJIAN LITERATUR

2.1 Usaha Mikro, Kecil, dan Menengah

Usaha Mikro, Kecil, dan Menengah (UMKM) merupakan unit usaha produktif yang dijalankan oleh perseorangan atau badan usaha yang berdiri sendiri dan bergerak di berbagai sektor ekonomi [14]. UMKM memiliki peran penting dalam perekonomian nasional karena berkontribusi terhadap pertumbuhan ekonomi, penciptaan lapangan kerja, dan peningkatan kesejahteraan masyarakat [15]. Di Indonesia, UMKM didefinisikan dan diklasifikasikan berdasarkan jumlah aset dan omzet tahunan sesuai dengan Undang-Undang Nomor 20 Tahun 2008. Berdasarkan regulasi tersebut, usaha mikro merupakan usaha produktif milik orang perorangan yang memenuhi kriteria tertentu sebagaimana diatur dalam perundang-undangan [16].

Tujuan pengembangan dan pemberdayaan UMKM adalah meningkatkan produktivitas dan daya saing usaha secara berkelanjutan, sehingga mampu meningkatkan pendapatan serta memperkuat struktur perekonomian yang lebih merata. Upaya tersebut mencakup penumbuhan usaha baru, penguatan usaha yang telah berjalan, serta peningkatan kualitas produk agar mampu bersaing di pasar domestik maupun internasional. Dengan demikian, penguatan UMKM tidak hanya berdampak pada pelaku usaha, tetapi juga pada kemandirian ekonomi masyarakat secara luas [17].

Secara umum, UMKM memiliki karakteristik skala usaha yang relatif kecil dilihat dari aset, omzet, dan jumlah tenaga kerja [17]. Fleksibilitas menjadi salah satu ciri utama, karena pelaku usaha dapat menyesuaikan jenis produk dan strategi usaha sesuai dengan kondisi pasar [5]. Dalam aspek pengelolaan, sebagian besar UMKM masih menggunakan sistem pencatatan keuangan yang sederhana dan belum sepenuhnya memisahkan keuangan usaha dari keuangan pribadi. Keterbatasan akses terhadap perbankan dan legalitas formal juga masih ditemukan, terutama pada usaha mikro [16]. Selain itu, penggunaan teknologi sederhana dan pemanfaatan bahan baku lokal menjadi ciri umum yang melekat pada UMKM [18].

2.1.1 Klasifikasi Usaha Mikro, Kecil, dan Menengah

UMKM dapat diklasifikasikan ke dalam beberapa kategori berdasarkan berbagai perspektif, seperti skala usaha, sektor ekonomi, perspektif perkembangan usaha, dan jenis

kegiatan usaha. Pengelompokan ini bertujuan untuk memberikan gambaran yang lebih jelas mengenai karakteristik serta potensi masing-masing kelompok UMKM sehingga dapat membantu pemerintah maupun pelaku usaha dalam merancang strategi pengembangan yang tepat [16].

1. Jenis UMKM Berdasarkan Skala Usaha

   Salah satu pembagian UMKM yang paling umum digunakan adalah berdasarkan skala usaha. Skala usaha dapat diukur melalui beberapa indikator, seperti nilai penjualan tahunan, jumlah tenaga kerja, maupun besarnya modal usaha. Berdasarkan skala ini, UMKM dapat dibedakan menjadi usaha mikro, usaha kecil, dan usaha menengah. Usaha mikro umumnya memiliki modal dan jumlah tenaga kerja yang terbatas serta dikelola secara sederhana oleh individu atau keluarga. Usaha kecil memiliki kapasitas produksi dan jumlah tenaga kerja yang lebih besar dibandingkan usaha mikro serta mulai menerapkan pengelolaan usaha yang lebih terstruktur. Sementara itu, usaha menengah merupakan usaha yang telah memiliki skala operasional lebih besar, tenaga kerja yang lebih banyak, serta manajemen usaha yang lebih profesional.

2. Jenis UMKM Berdasarkan Sektor Ekonomi

   UMKM juga dapat diklasifikasikan berdasarkan sektor ekonomi tempat usaha tersebut beroperasi. Menurut pengelompokan sektor ekonomi, UMKM dapat bergerak dalam berbagai bidang, seperti sektor pertanian, peternakan, kehutanan, dan perikanan yang memanfaatkan sumber daya alam sebagai bahan utama produksi. Selain itu terdapat sektor pertambangan dan penggalian, sektor industri pengolahan yang mengubah bahan mentah menjadi produk bernilai tambah, serta sektor listrik, gas, dan air bersih yang berkaitan dengan penyediaan layanan energi dan air. UMKM juga banyak berkembang pada sektor perdagangan, hotel, dan restoran yang berfokus pada kegiatan jual beli barang maupun jasa kepada konsumen. Selain itu terdapat sektor transportasi dan komunikasi, sektor keuangan dan jasa perusahaan, serta sektor jasa-jasa lainnya yang menyediakan berbagai layanan untuk memenuhi kebutuhan masyarakat dan dunia usaha.

3. Jenis UMKM Berdasarkan Perspektif Perkembangan Usaha

   Berdasarkan perspektif perkembangan usaha, UMKM dapat dibagi menjadi beberapa kelompok. Pertama, sektor informal yang umumnya dijalankan secara sederhana oleh individu, seperti pedagang kaki lima atau usaha rumahan yang belum memiliki legalitas formal. Kedua, usaha mikro yang memiliki keterampilan produksi tertentu namun masih memiliki keterbatasan dalam pengelolaan kewirausahaan dan

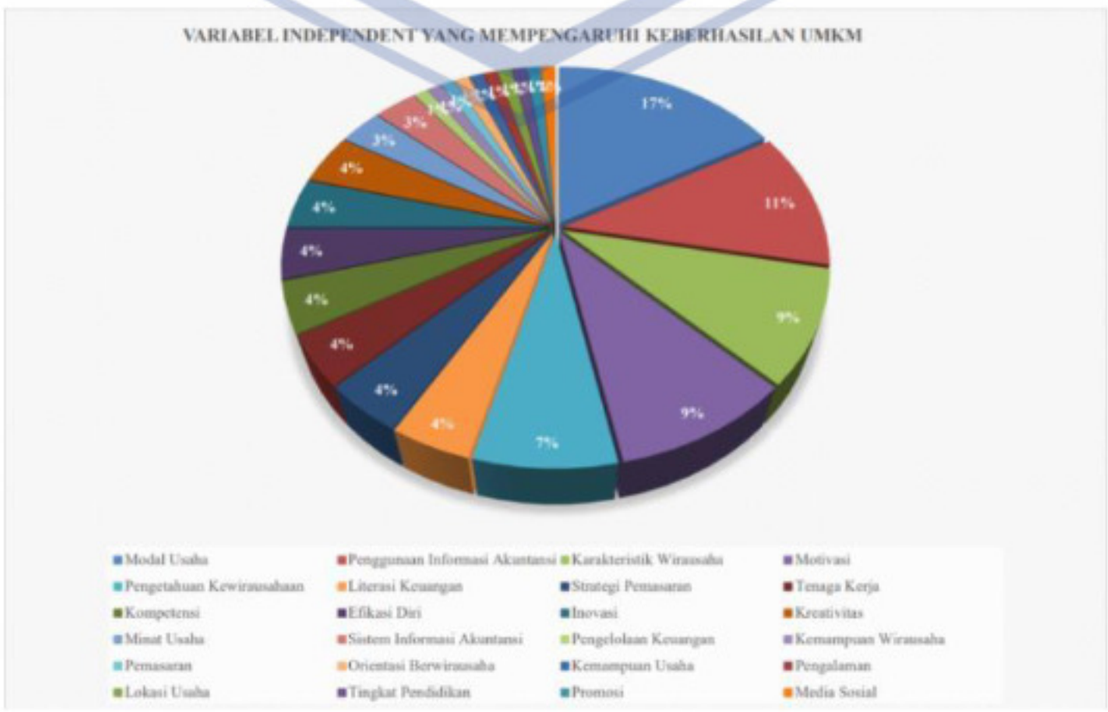
pengembangan usaha. Ketiga, usaha kecil dinamis yang telah mampu menjalin kerja sama usaha, misalnya melalui sistem kemitraan atau subkontrak dengan perusahaan yang lebih besar. Keempat, kelompok fast moving enterprise, yaitu usaha yang memiliki kemampuan kewirausahaan yang kuat dan berpotensi berkembang menjadi usaha berskala besar.

4. Jenis UMKM Berdasarkan Jenis Usaha

   Jika dilihat dari jenis kegiatannya, UMKM di Indonesia dapat bergerak dalam berbagai bidang usaha. Salah satu yang paling umum adalah usaha kuliner yang mencakup produksi dan penjualan makanan serta minuman. Selain itu terdapat usaha fashion yang bergerak dalam produksi maupun perdagangan pakaian, alas kaki, dan aksesori. UMKM juga berkembang dalam bidang agribisnis yang memanfaatkan hasil pertanian sebagai bahan baku utama produksi. Jenis usaha lainnya adalah usaha kerajinan tangan yang menghasilkan berbagai produk berbasis keterampilan seperti anyaman, ukiran, dan kerajinan tradisional. Selain itu terdapat usaha kosmetik dan kecantikan yang memproduksi berbagai produk perawatan tubuh serta usaha jasa yang menyediakan berbagai layanan seperti jasa pengiriman, jasa kebersihan, maupun jasa desain.

2.1.2 Faktor Keberhasilan Usaha Mikro, Kecil, dan Menengah

Keberhasilan UMKM dipengaruhi oleh berbagai faktor yang berkaitan dengan kemampuan pengelolaan usaha, sumber daya yang dimiliki, serta strategi bisnis yang diterapkan oleh pelaku usaha. Keberhasilan usaha umumnya ditandai dengan peningkatan produksi, penjualan, keuntungan, serta keberlanjutan perkembangan usaha dalam jangka panjang [4]. Gambar 2.1 menunjukkan faktor-faktor yang memengaruhi keberhasilan UMKM.



Gambar 2.1 Faktor-Faktor yang Memengaruhi Keberhasilan UMKM [4]

Berdasarkan berbagai penelitian, terdapat beberapa faktor utama yang memengaruhi keberhasilan UMKM, antara lain sebagai berikut:

1. Modal Usaha

   Modal usaha merupakan salah satu faktor penting dalam menunjang keberhasilan UMKM. Ketersediaan modal memungkinkan pelaku usaha untuk menjalankan kegiatan produksi, meningkatkan kapasitas usaha, serta mengembangkan produk yang dihasilkan [19].

2. Pengalaman Usaha

   Pengalaman dalam menjalankan usaha dapat membantu pelaku UMKM dalam menghadapi berbagai tantangan bisnis serta meningkatkan kemampuan dalam mengambil keputusan yang tepat dalam pengelolaan usaha [19].

3. Pengetahuan Usaha

   Pengetahuan mengenai manajemen usaha, pemasaran, dan pengelolaan keuangan sangat berperan dalam meningkatkan keberhasilan UMKM. Pelaku usaha yang memiliki pemahaman yang baik mengenai aspek-aspek tersebut akan lebih mampu mengelola usahanya secara efektif dan efisien [19].

4. Tenaga Kerja

   Tenaga kerja yang memiliki keterampilan dan kompetensi yang baik dapat mendukung kelancaran proses produksi dan meningkatkan kualitas produk atau layanan yang dihasilkan oleh UMKM [19].

5. Inovasi Usaha

   Kemampuan untuk melakukan inovasi dalam produk maupun proses usaha menjadi faktor penting dalam menjaga daya saing UMKM di tengah persaingan pasar yang semakin ketat.

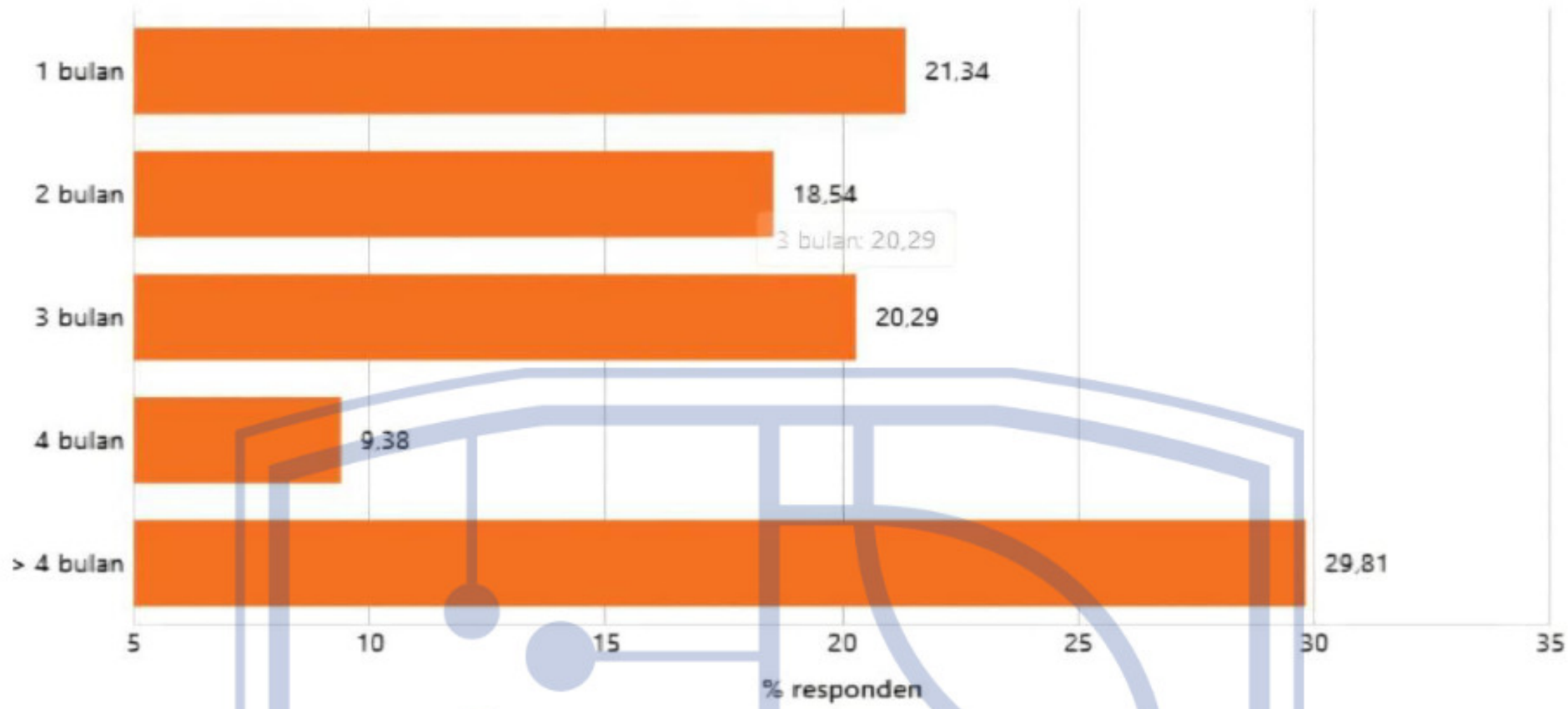
6. Strategi Pemasaran [20].

   Strategi pemasaran yang tepat dapat membantu pelaku UMKM dalam memperluas pasar, meningkatkan penjualan, serta memperkuat posisi usaha dalam persaingan bisnis [21].

2.1.3 Faktor Kegagalan Usaha Mikro, Kecil, dan Menengah

Selain faktor-faktor yang memengaruhi keberhasilan UMKM, terdapat pula berbagai faktor yang dapat menyebabkan kegagalan atau menghambat perkembangan UMKM. Faktor-faktor tersebut dapat berasal dari aspek internal maupun eksternal yang

memengaruhi keberlangsungan usaha [22]. Gambar 2.2 menyajikan grafik kemampuan bertahan UMKM dengan modal pada tahun 2021.



Gambar 2.2 Kemampuan Bertahan UMKM dengan Modal Tahun 2021 [22]

Berdasarkan Gambar 2.2, kemampuan bertahan UMKM dengan modal yang dimiliki pada tahun 2021 menunjukkan variasi waktu operasional yang berbeda-beda. Sebagian besar pelaku UMKM menyatakan bahwa modal yang dimiliki mampu mempertahankan keberlangsungan usaha selama lebih dari empat bulan, yaitu sebesar 29,81% responden. Sementara itu, sebanyak 21,34% responden menyatakan modal usaha dapat bertahan selama satu bulan, 20,29% selama tiga bulan, 18,54% selama dua bulan, dan hanya 9,38% yang menyatakan modalnya bertahan selama empat bulan. Hasil tersebut menunjukkan bahwa kemampuan modal dalam menopang keberlangsungan usaha masih beragam, sehingga ketersediaan modal menjadi salah satu aspek penting yang memengaruhi ketahanan dan keberlangsungan UMKM, terutama ketika menghadapi kondisi ekonomi yang tidak menentu [22].

Berdasarkan berbagai penelitian, terdapat beberapa faktor yang dapat menyebabkan kegagalan atau menghambat perkembangan UMKM, antara lain sebagai berikut:

1. Keterbatasan Modal

   Keterbatasan modal menjadi salah satu hambatan utama bagi pelaku UMKM dalam mengembangkan usahanya. Modal yang terbatas dapat menyebabkan pelaku usaha mengalami kesulitan dalam meningkatkan kapasitas produksi, melakukan inovasi produk, maupun memperluas pasar usaha [22].

2. Kualitas Sumber Daya Manusia

   Rendahnya kualitas sumber daya manusia dapat memengaruhi kemampuan pelaku UMKM dalam mengelola usaha secara efektif. Kurangnya keterampilan, pengetahuan manajerial, serta pengalaman usaha dapat menghambat perkembangan usaha dan meningkatkan risiko kegagalan [22].

3. Keterbatasan Akses Pembiayaan

   Sebagian besar pelaku UMKM masih menghadapi kendala dalam memperoleh akses pembiayaan dari lembaga keuangan formal. Hal ini menyebabkan pelaku usaha kesulitan memperoleh tambahan modal yang dibutuhkan untuk mengembangkan usaha [22].

4. Keterbatasan Infrastruktur dan Teknologi

   Keterbatasan infrastruktur serta rendahnya pemanfaatan teknologi juga menjadi faktor yang dapat menghambat perkembangan UMKM. Kondisi ini dapat mengurangi efisiensi operasional serta membatasi kemampuan pelaku usaha dalam menjangkau pasar yang lebih luas [22][23].

5. Persaingan Usaha

   Persaingan pasar yang semakin ketat dapat meningkatkan risiko kegagalan usaha apabila pelaku UMKM tidak mampu beradaptasi dengan perubahan pasar, melakukan inovasi produk, atau menerapkan strategi pemasaran yang efektif [18].

6. Kondisi Ekonomi dan Faktor Eksternal

   Faktor eksternal seperti kondisi ekonomi, perubahan kebijakan, maupun situasi krisis seperti pandemi juga dapat memengaruhi keberlangsungan usaha UMKM. Peningkatan biaya produksi, biaya distribusi, serta penurunan permintaan pasar dapat menyebabkan menurunnya kinerja usaha [24].

2.2 *Machine Learning*

Machine learning merupakan salah satu cabang dari *artificial intelligence* yang berfokus pada pengembangan sistem komputer yang mampu belajar dari data dan meningkatkan kinerjanya secara otomatis tanpa harus diprogram secara eksplisit. Melalui pendekatan ini, komputer dapat mengenali pola dalam data dan menggunakan pola tersebut untuk melakukan prediksi maupun pengambilan keputusan. Proses pembelajaran pada *machine learning* dilakukan dengan memanfaatkan algoritma yang dapat mempelajari hubungan antara data masukan (*input*) dan keluaran (*output*) berdasarkan data *training* yang diberikan [25].

Secara umum, *machine learning* dapat diklasifikasikan menjadi beberapa jenis utama, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. Pada *supervised learning*, model dilatih menggunakan *dataset* yang telah memiliki *label* sehingga sistem dapat mempelajari hubungan antara variabel *input* dan *output* untuk melakukan prediksi. Sementara itu, *unsupervised learning* digunakan untuk menemukan pola atau struktur tersembunyi dalam data tanpa adanya *label*, sedangkan *reinforcement learning* merupakan metode pembelajaran yang melibatkan interaksi antara agen dan lingkungan melalui mekanisme penghargaan (*reward*) untuk memperoleh kebijakan terbaik [26]. Dalam penerapannya, *machine learning* banyak digunakan dalam berbagai bidang seperti pengenalan pola, analisis data, prediksi, serta pengambilan keputusan berbasis data. Salah satu algoritma yang banyak digunakan dalam pendekatan *machine learning* adalah ANN, yaitu model komputasi yang meniru cara kerja jaringan saraf biologis untuk mempelajari hubungan yang kompleks dan *nonlinier* antarvariabel dalam data [27].

2.3 *Artificial Neural Network*

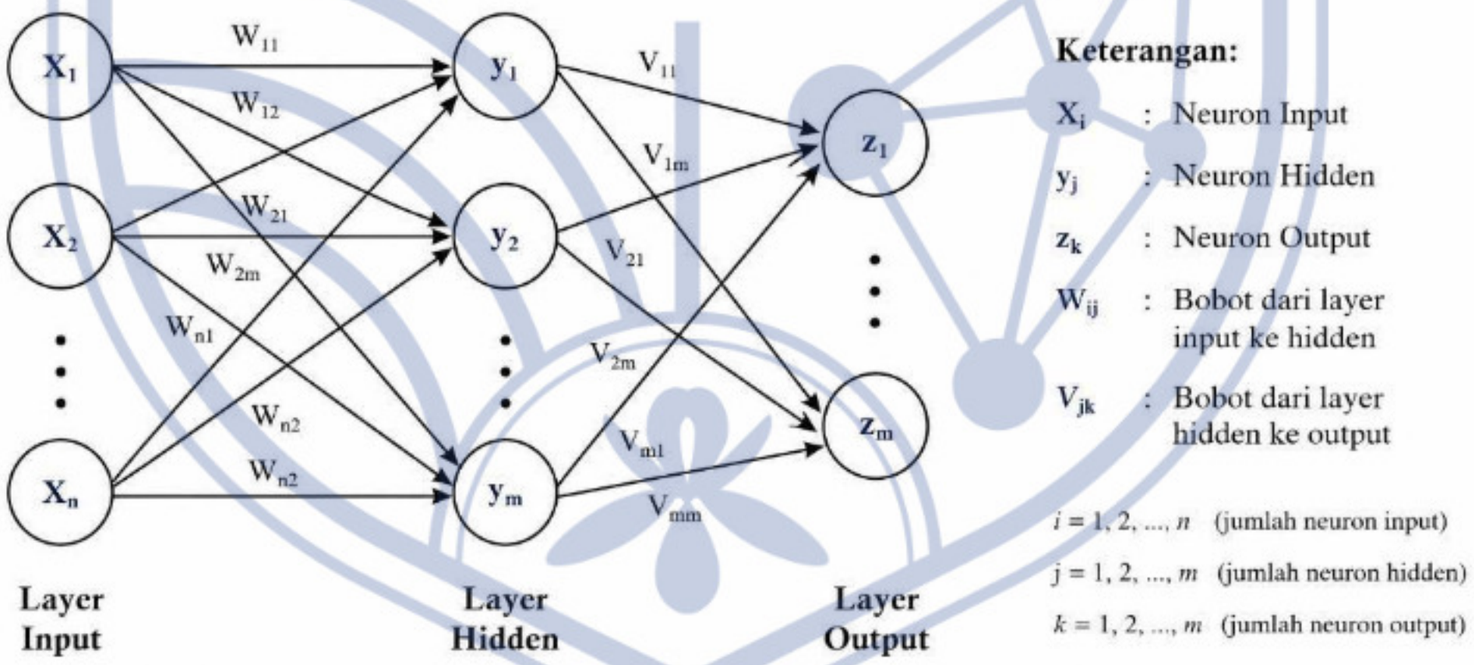
Artificial Neural Network (ANN) atau jaringan saraf buatan merupakan model komputasi yang terinspirasi dari struktur dan mekanisme kerja jaringan saraf biologis pada manusia maupun hewan, yang terdiri atas unit-unit pengolah yang disebut *neuron* dan saling terhubung melalui bobot tertentu [28]. ANN dikembangkan sebagai representasi matematis dari proses pembelajaran pada otak untuk membangun sistem yang mampu menyelesaikan operasi kompleks secara lebih efektif dibandingkan metode komputasi tradisional [29]. ANN dirancang untuk memodelkan hubungan yang kompleks dan nonlinier antarvariabel melalui proses pembelajaran adaptif berbasis data [30]. Setiap *neuron* menerima *input*, mengalikan *input* dengan bobot dan bias, kemudian menerapkan fungsi aktivasi untuk menghasilkan keluaran yang diteruskan ke lapisan berikutnya [31]. Melalui proses pelatihan, ANN menyesuaikan bobot-bobot tersebut menggunakan algoritma *backpropagation* agar kesalahan prediksi terhadap nilai target dapat diminimalkan. Kemampuan ANN dalam mengenali pola (*pattern recognition*) dan memetakan fungsi nonlinier menjadikannya efektif untuk tugas klasifikasi dan prediksi pada data yang kompleks [28]. Oleh karena itu, ANN banyak diterapkan dalam berbagai bidang seperti pengenalan wajah, pengenalan suara, pemrosesan bahasa alami, serta analisis dan peramalan data ekonomi [30]. Dalam penelitian ini, ANN digunakan untuk

mempelajari pola karakteristik UMKM dan memprediksi keberhasilan usaha berdasarkan variabel-variabel yang tersedia dalam *dataset*.

Secara umum, proses kerja ANN dimulai ketika data masukan diterima oleh *input layer* dan diteruskan ke lapisan berikutnya melalui koneksi berbobot. Setiap *neuron* akan menghitung kombinasi linier dari *input* dan bobot, kemudian menambahkan *bias* sebelum diterapkan fungsi aktivasi untuk menghasilkan *output* sementara. Nilai keluaran tersebut selanjutnya diteruskan ke *hidden layer* hingga mencapai *output layer* yang menghasilkan prediksi akhir dari model. Pada tahap pelatihan, hasil prediksi dibandingkan dengan nilai target menggunakan fungsi *loss* untuk menghitung tingkat kesalahan model. Kesalahan tersebut kemudian disebarkan kembali ke seluruh jaringan menggunakan algoritma *backpropagation* untuk memperbarui bobot dan *bias* pada setiap *neuron* sehingga kesalahan dapat diminimalkan pada iterasi berikutnya. Proses ini dilakukan secara berulang hingga model mencapai konvergensi dan mampu mempelajari pola hubungan antarvariabel dalam data secara optimal [27].

2.3.1 Struktur ANN

Gambar 2.3 menyajikan struktur dari *Multilayer Feedforward Network.*



Gambar 2.3 Struktur *Multilayer Feedforward Network* [32]

Secara umum, struktur ANN terdiri atas tiga lapisan utama, yaitu [29][30]:

1. *Input Layer*

   Lapisan ini menerima data masukan tanpa melakukan proses komputasi, kemudian meneruskannya ke lapisan berikutnya.

2. *Hidden Layer*

Lapisan tersembunyi melakukan proses komputasi melalui kombinasi bobot, penambahan bias, dan penerapan fungsi aktivasi nonlinier untuk mengekstraksi pola dari data.

3. *Output Layer*

Lapisan ini menghasilkan nilai akhir yang menjadi prediksi model sesuai dengan tujuan penelitian, baik dalam bentuk klasifikasi maupun regresi.

Aliran data dari *input* menuju *output* disebut sebagai proses *feedforward*, sedangkan proses penyesuaian bobot dilakukan melalui mekanisme *backpropagation* berbasis optimisasi gradien [32]. Pada umumnya, setiap *neuron* dalam satu lapisan terhubung penuh (*fully connected*) dengan *neuron* pada lapisan berikutnya sehingga membentuk struktur yang fleksibel dalam memodelkan hubungan nonlinier [33].

2.3.2 Model Matematis *Neuron*

Secara matematis, keluaran dari satu *neuron* dalam ANN dapat dirumuskan sebagai berikut [27]:

$$y = f\left(\sum_{i=1}^{n} \omega_i x_i + b\right) \quad \text{.....(1)}$$

Di mana:

y = *output neuron*

$\sum_{i=1}^{n} \omega_i x_i + b = z$

$f(z)$ = fungsi aktivasi

ω_i = bobot pada *input* ke-i

x_i = *input* ke-i

b = bias

Persamaan tersebut menunjukkan bahwa setiap *neuron* melakukan operasi penjumlahan berbobot (*weighted sum*) terhadap seluruh *input*, kemudian menerapkan fungsi aktivasi untuk menghasilkan keluaran nonlinier.

2.3.3 Jenis-Jenis ANN

Berdasarkan arsitektur jaringan dan pola aliran informasinya, ANN dapat dikembangkan dalam beberapa bentuk. Secara umum, ANN tersusun atas *input layer*, *hidden layer*, dan *output layer*, di mana setiap lapisan saling terhubung untuk memproses informasi dari data masukan hingga menghasilkan keluaran. Perbedaan jumlah lapisan, arah aliran data, serta mekanisme koneksi antar *neuron* menyebabkan ANN memiliki

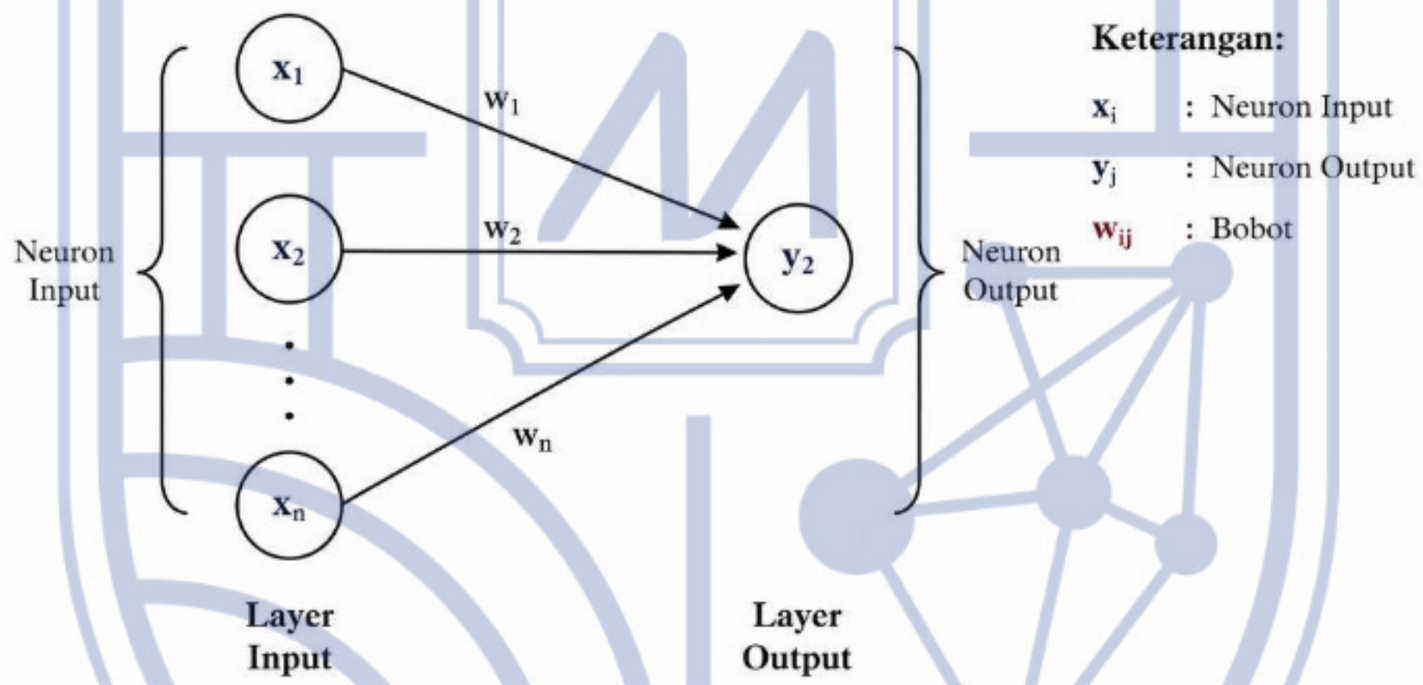
beberapa jenis arsitektur yang digunakan sesuai dengan karakteristik permasalahan, yaitu [27] [29][31]:

1. *Feedforward Neural Network (FFNN)*

   Feedforward neural network merupakan bentuk dasar dari ANN, di mana aliran informasi bergerak satu arah dari *input layer* menuju *output layer* tanpa adanya umpan balik ke lapisan sebelumnya. Pada arsitektur ini, setiap *neuron* menerima *input*, memprosesnya melalui bobot dan fungsi aktivasi, kemudian meneruskannya ke lapisan berikutnya hingga menghasilkan *output*. Struktur ini banyak digunakan pada permasalahan klasifikasi maupun prediksi karena proses komputasinya relatif sederhana dan efektif untuk data tabular.

2. *Single Layer Perceptron*

   Single layer perceptron merupakan bentuk ANN yang paling sederhana. Struktur *single layer perceptron* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Struktur *Single Layer Feedforward* [32]

Model ini terdiri atas *input layer* dan *output layer* dengan satu lapisan pemrosesan, sehingga cocok digunakan untuk permasalahan yang memiliki hubungan linier. Namun, kemampuan model ini masih terbatas dalam menangani pola data yang kompleks.

3. *Multilayer Perceptron* (MLP)

   MLP merupakan pengembangan dari *perceptron* dengan menambahkan satu atau lebih *hidden layer* di antara *input layer* dan *output layer*. Penambahan lapisan tersembunyi ini memungkinkan jaringan mempelajari pola yang lebih kompleks dan hubungan nonlinier antarvariabel. Karena kemampuannya yang lebih baik dalam memodelkan

data dibandingkan *single layer perceptron*, MLP menjadi salah satu arsitektur ANN yang paling banyak digunakan pada berbagai permasalahan klasifikasi dan prediksi.

4. *Reccurent Neural Network* (RNN)

   Berbeda dengan *feedforward network*, RNN merupakan jaringan saraf yang dirancang untuk memproses data yang memiliki urutan atau dependensi waktu. Berbeda dengan FFNN, RNN memiliki mekanisme koneksi berulang (*recurrent connection*) yang memungkinkan informasi dari langkah sebelumnya digunakan kembali pada langkah berikutnya. Oleh karena itu, RNN lebih sesuai digunakan untuk data sekuensial seperti teks, suara, dan deret waktu.

Dalam penelitian ini digunakan arsitektur MLP karena sesuai untuk permasalahan klasifikasi berbasis data tabular. MLP juga cocok digunakan dalam permasalahan yang memiliki hubungan linier dan tidak memerlukan dependensi waktu secara eksplisit.

2.3.4 *Multilayer Perceptron*

Multilayer Perceptron (MLP) merupakan salah satu jenis ANN *feedforward* yang memiliki minimal satu *hidden layer* selain *input layer* dan *output layer* [32]. MLP dikembangkan untuk mengatasi keterbatasan *single perceptron* yang hanya mampu menangani data yang dapat dipisahkan secara linier, sedangkan MLP mampu mempelajari pola yang tidak *linearly separable* [28][33]. MLP menggunakan fungsi aktivasi nonlinier seperti *sigmoid* atau ReLU pada setiap *neuron* di *hidden layer* untuk meningkatkan kemampuan model dalam memetakan hubungan kompleks antarvariabel. Jika seluruh *neuron* menggunakan fungsi linier, maka beberapa lapisan sekalipun dapat direduksi menjadi satu transformasi linier saja, sehingga model tidak memiliki keunggulan dibandingkan model linier biasa [33].

Proses pelatihan MLP dilakukan menggunakan algoritma *backpropagation* untuk meminimalkan fungsi *loss* melalui optimisasi berbasis gradien [31][33]. Pada proses ini, hasil prediksi model dibandingkan dengan nilai target untuk menghitung tingkat kesalahan yang terjadi. Kesalahan tersebut kemudian disebarkan kembali ke seluruh jaringan guna memperbarui bobot dan *bias* pada setiap *neuron* agar model dapat belajar secara lebih optimal. Pembaruan parameter dilakukan dengan bantuan *optimizer* untuk meningkatkan efisiensi konvergensi selama proses pelatihan. Dalam penelitian ini, arsitektur model yang digunakan adalah MLP dengan dua *hidden layer*, fungsi aktivasi ReLU pada *hidden layer*, satu *output layer* dengan fungsi aktivasi *sigmoid.* Arsitektur model ini digunakan untuk klasifikasi biner keberhasilan UMKM.

Pemilihan arsitektur MLP dalam penelitian ini didasarkan pada karakteristik *dataset* yang digunakan, yaitu data tabular dengan sejumlah variabel numerik yang merepresentasikan karakteristik UMKM serta target klasifikasi biner berupa keberhasilan atau kegagalan usaha. MLP dipilih karena mampu mempelajari hubungan nonlinier antarfitur secara efektif tanpa memerlukan struktur data berurutan seperti pada RNN maupun data spasial seperti pada CNN. Selain itu, MLP juga sesuai digunakan untuk kasus prediksi berbasis data terstruktur karena mampu melakukan pemetaan hubungan antara variabel *input* dan *output* secara langsung melalui proses pelatihan berbasis *backpropagation*. Dengan demikian, penggunaan MLP dalam penelitian ini dinilai relevan untuk memodelkan pola keberhasilan UMKM berdasarkan karakteristik usaha yang tersedia dalam *dataset* [27].

2.3.5 Fungsi Aktivasi

Fungsi aktivasi merupakan komponen penting dalam ANN karena memberikan sifat nonlinier pada jaringan sehingga model mampu mempelajari pola yang kompleks. Tanpa fungsi aktivasi, jaringan saraf hanya akan menjadi model linier sederhana [34]. Beberapa fungsi aktivasi yang umum digunakan antara lain *sigmoid* dan *Rectified Linear Unit* (ReLU). Fungsi *sigmoid* menghasilkan *output* dalam rentang 0 hingga 1 sehingga sering digunakan pada *output layer* untuk klasifikasi biner [34]. Fungsi aktivasi ReLU merupakan salah satu fungsi yang banyak digunakan dalam pengembangan model jaringan saraf modern. Fungsi ini dikenal mampu meningkatkan efisiensi proses pelatihan karena mempercepat konvergensi serta mengurangi permasalahan numerik yang sering terjadi pada fungsi aktivasi *sigmoid*. Secara matematis, fungsi ReLU dapat dirumuskan sebagai berikut [25]:

$$f(x) = max(0, x) \quad \text{.....(2)}$$

Di mana:

$f(x)$ = nilai *output* dari fungsi aktivasi ReLU

x = nilai *input* yang masuk ke *neuron*

$max(0, x)$ = fungsi yang memilih nilai terbesar antara 0 dan x

Dalam penelitian ini, fungsi aktivasi ReLU digunakan pada *hidden layer* untuk menangkap pola nonlinier dalam data UMKM, sedangkan fungsi *sigmoid* digunakan pada *output layer* karena sesuai untuk menghasilkan probabilitas pada kasus klasifikasi biner. Secara matematis, fungsi *sigmoid* dirumuskan sebagai berikut [25]:

$$f(x) = \frac{1}{1+e^{-x}} \quad \text{.....(3)}$$

Di mana:

$f(x)$ = *output* fungsi *sigmoid*

x = nilai *input* ke *neuron*

e^{-x} = bilangan eksponensial

2.3.6 Proses Pelatihan Model

Proses pelatihan ANN dilakukan untuk menyesuaikan bobot dan *bias* pada setiap *neuron* agar model mampu menghasilkan prediksi yang semakin mendekati nilai aktual. Pada tahap ini, data masukan akan diproses melalui jaringan secara bertahap mulai dari *input layer*, *hidden layer*, hingga *output layer* melalui mekanisme *forward propagation*. Setiap *neuron* akan menghitung kombinasi linier antara *input*, bobot, dan bias, kemudian hasilnya akan diproses menggunakan fungsi aktivasi [27]. Pada penelitian ini, fungsi aktivasi yang digunakan pada *hidden layer* adalah *Rectified Linear Unit* (ReLU), sedangkan pada *output layer* digunakan fungsi *sigmoid*.

Hasil prediksi yang diperoleh dari proses *forward propagation* kemudian dibandingkan dengan nilai aktual menggunakan fungsi *loss*. Dalam penelitian ini, fungsi *loss* yang digunakan adalah *binary cross-entropy* karena sesuai untuk permasalahan klasifikasi biner [27]. *Binary cross-entropy* (BCE) merupakan salah satu fungsi *loss* yang umum digunakan dalam permasalahan klasifikasi biner pada model *machine learning* dan *deep learning*. Fungsi ini digunakan untuk mengukur perbedaan antara nilai probabilitas hasil prediksi model dengan nilai aktual yang sebenarnya. Dalam proses pelatihan model, nilai *loss* yang dihasilkan akan digunakan sebagai dasar untuk memperbarui bobot melalui mekanisme *backpropagation*. Semakin kecil nilai *loss* yang dihasilkan, maka semakin baik performa model dalam memprediksi kelas yang sesuai dengan data sebenarnya. Berdasarkan penelitian yang dilakukan pada model klasifikasi berbasis *neural network*, *binary cross-entropy* sangat efektif digunakan ketika target *output* hanya terdiri dari dua kelas, yaitu 0 dan 1. Secara matematis, *binary cross-entropy* dirumuskan sebagai berikut [35]:

$$L = -(y \log(\hat{y}) + (1-y) \log(1-\hat{y})) \quad \text{.....(4)}$$

Di mana:

L = nilai *loss*

y = nilai aktual (*label* sebenarnya)

$\hat{y}$ = nilai prediksi (*output* model)

log = fungsi logaritma

Rumus ini bekerja dengan memberikan penalti yang lebih besar ketika prediksi model jauh dari nilai sebenarnya, terutama ketika probabilitas mendekati nol untuk kelas yang seharusnya bernilai satu. Selain itu, fungsi ini menggunakan logaritma sehingga kesalahan kecil menghasilkan nilai *loss* yang kecil, sedangkan kesalahan besar akan meningkatkan nilai *loss* secara signifikan. Dalam implementasinya, *binary cross-entropy* biasanya digunakan bersamaan dengan fungsi aktivasi *sigmoid*, karena *sigmoid* menghasilkan *output* dalam rentang 0 hingga 1 yang sesuai dengan kebutuhan perhitungan probabilitas pada fungsi *loss* ini [35].

Dalam konteks pelatihan ANN, *binary cross-entropy* berperan penting dalam proses optimasi model. Nilai *loss* yang dihasilkan dari fungsi ini akan digunakan untuk menghitung gradien yang kemudian digunakan dalam proses pembaruan bobot pada setiap *neuron*. Proses ini dilakukan secara iteratif hingga model mencapai kondisi konvergen atau nilai *loss* yang minimum. Penggunaan *binary cross-entropy* juga membantu model dalam membedakan kelas karena mempertimbangkan distribusi probabilitas dari setiap prediksi. Oleh karena itu, pemilihan fungsi *loss* ini sangat sesuai untuk meningkatkan performa model dalam tugas klasifikasi biner [35]. Nilai *loss* yang telah dihitung selanjutnya digunakan dalam proses *backpropagation* untuk menentukan arah dan besar perubahan bobot serta *bias* pada jaringan. Secara umum, pembaruan bobot pada metode optimisasi berbasis gradien dilakukan dengan mengurangi nilai bobot lama menggunakan hasil turunan parsial fungsi *loss* terhadap bobot, sebagaimana ditunjukkan pada persamaan berikut [27]:

$$\omega_{new} = \omega_{old} - \eta \frac{\partial L}{\partial \omega} \quad (5)$$

Di mana:

η = *learning rate*

L = fungsi *loss*

$\frac{\partial L}{\partial \omega}$ = gradien terhadap bobot

Dalam penelitian ini, proses optimisasi dilakukan menggunakan *Adaptive Moment Estimation* (Adam *Optimizer*), yaitu algoritma optimisasi berbasis gradien yang banyak digunakan dalam pelatihan jaringan saraf karena mampu menyesuaikan *learning rate* secara adaptif untuk setiap parameter model. Adam *Optimizer* bekerja dengan memanfaatkan estimasi momen pertama (*first moment*) dan momen kedua (*second*

moment) dari gradien, sehingga proses pembaruan bobot menjadi lebih stabil dan efisien selama pelatihan. Dibandingkan metode optimisasi dasar, Adam *Optimizer* memiliki keunggulan dalam mempercepat konvergensi, mengurangi osilasi selama pembelajaran, serta memberikan performa yang baik pada permasalahan dengan ruang parameter yang kompleks. Oleh karena itu, Adam *Optimizer* dipilih dalam penelitian ini karena sesuai untuk pelatihan model MLP yang digunakan dalam klasifikasi biner keberhasilan UMKM. Selain penggunaan Adam *Optimizer*, untuk mengurangi risiko *overfitting* diterapkan teknik *dropout* pada *hidden layer* sehingga model tidak terlalu bergantung pada *neuron* tertentu selama proses pembelajaran. Sebagian data pelatihan juga digunakan sebagai data validasi untuk memantau performa model selama proses pelatihan berlangsung. Pendekatan ini bertujuan agar model tidak hanya memiliki performa yang baik pada data pelatihan, tetapi juga mampu melakukan generalisasi secara lebih baik terhadap data pengujian [36].

2.3.7 Pembagian Data *Training* dan Data *Testing*

Setelah pemisahan fitur dan target, *dataset* kemudian dibagi menjadi dua bagian yaitu data *training* dan data *testing*. Data *training* digunakan untuk melatih model agar dapat mempelajari pola dari data, sedangkan data *testing* digunakan untuk mengevaluasi performa model terhadap data yang belum pernah dilihat sebelumnya [37]. Pembagian data umumnya dilakukan menggunakan metode *train-test split*, yaitu teknik yang membagi *dataset* menjadi dua bagian berdasarkan proporsi tertentu. Dalam penelitian ini digunakan proporsi 80% data *training* dan 20% data *testing*. Selain itu, pembagian data dilakukan menggunakan teknik *stratified sampling* untuk menjaga distribusi kelas pada data *training* dan data *testing* tetap seimbang sehingga model dapat belajar secara lebih representatif terhadap data yang tersedia [27].

Stratified sampling merupakan teknik pembagian data dengan mempertahankan proporsi setiap kelas pada proses *train-test split*. Teknik ini memastikan bahwa distribusi kelas pada data *training* dan data *testing* tetap sesuai dengan distribusi data awal, sehingga model tidak *bias* terhadap kelas tertentu. Pendekatan ini penting dalam permasalahan klasifikasi karena distribusi data yang tidak seimbang dapat memengaruhi kemampuan model dalam mempelajari pola data. Oleh karena itu, penggunaan *stratified sampling* membantu menghasilkan model yang lebih stabil dan representatif dalam proses pelatihan maupun evaluasi [38].

Pembagian data menjadi data *training* dan data *testing* merupakan tahapan penting dalam proses *machine learning* untuk mengevaluasi kemampuan model terhadap data yang

belum pernah dipelajari sebelumnya. Data *training* digunakan dalam proses pelatihan model, sedangkan data *testing* digunakan untuk mengukur performa model secara objektif. Dalam proses pengolahan data, tahap *preprocessing* seperti transformasi data dan penyesuaian skala fitur juga memiliki peran penting dalam meningkatkan kualitas data sebelum digunakan dalam pemodelan. Dalam penelitian ini, pembagian data dilakukan sebelum proses standardisasi untuk menjaga objektivitas evaluasi model. Hal ini karena parameter standardisasi, seperti nilai rata-rata dan standar deviasi, sebaiknya dihitung berdasarkan data *training* agar tidak memengaruhi data *testing*. Dengan demikian, data *testing* tetap merepresentasikan data baru yang belum digunakan dalam proses pelatihan model. Pendekatan ini bertujuan agar hasil evaluasi model lebih representatif dan mendekati kondisi nyata saat model diterapkan pada data baru [38].

2.4 Data *Preprocessing*

Data *preprocessing* merupakan tahap awal dalam proses analisis data yang bertujuan untuk mempersiapkan *dataset* sebelum digunakan dalam proses pelatihan model *machine learning*. Pada tahap ini dilakukan berbagai proses pengolahan data untuk memastikan bahwa *dataset* memiliki kualitas yang baik dan sesuai untuk digunakan dalam proses pembelajaran model. Proses data *preprocessing* sangat penting karena kualitas data yang digunakan akan memengaruhi performa model yang dihasilkan. Data yang tidak bersih, memiliki skala yang berbeda, atau mengandung nilai yang tidak representatif dapat menyebabkan model menghasilkan prediksi yang kurang optimal [25].

Dalam penelitian ini, tahap data *preprocessing* dilakukan melalui beberapa langkah utama, yaitu pengecekan data duplikat, pengecekan *missing value*, pengecekan *outlier*, pemisahan fitur dan target, pembagian data *training* dan data *testing*, serta standardisasi data. Tahapan-tahapan tersebut dilakukan agar data yang digunakan dalam proses pelatihan model ANN berada dalam kondisi yang konsisten, representatif, dan siap diproses oleh model. Dengan melakukan data *preprocessing* yang baik, model yang dibangun diharapkan mampu mempelajari pola hubungan antarvariabel dengan lebih efektif [27].

2.4.1 Data Duplikat

Pengecekan data duplikat dilakukan untuk mengidentifikasi apakah terdapat baris data yang memiliki kesamaan secara keseluruhan dalam *dataset*. Keberadaan data duplikat dapat menyebabkan redundansi informasi dan memengaruhi distribusi data, sehingga berpotensi mengganggu proses pembelajaran model. Dalam konteks *machine learning*,

data yang terduplikasi dapat membuat model lebih sering "melihat" pola tertentu secara berulang, sehingga hasil pelatihan menjadi kurang representatif terhadap kondisi data yang sebenarnya. Oleh karena itu, identifikasi data duplikat penting dilakukan pada tahap awal data *preprocessing* untuk memastikan bahwa setiap data merepresentasikan satu entitas yang unik [25].

2.4.2 *Missing Value*

Missing value merupakan kondisi ketika suatu atribut dalam *dataset* tidak memiliki nilai atau terdapat data yang hilang pada suatu variabel. Kondisi ini dapat terjadi karena berbagai faktor seperti kesalahan dalam proses pengumpulan data, data yang tidak lengkap, kesalahan pencatatan, atau kegagalan sistem dalam menyimpan informasi tertentu. Keberadaan *missing value* dapat memengaruhi kualitas *dataset* karena nilai yang hilang dapat mengganggu proses analisis serta mengurangi keakuratan model yang dibangun. Dalam konteks *machine learning*, data yang tidak lengkap dapat menyebabkan model kesulitan dalam mempelajari pola hubungan antarvariabel secara optimal [25].

Oleh karena itu, proses identifikasi dan penanganan *missing value* menjadi salah satu langkah penting dalam tahap *missing value*. Beberapa metode yang umum digunakan untuk menangani *missing value* antara lain menghapus data yang memiliki nilai hilang (*deletion*), mengganti nilai yang hilang dengan nilai statistik tertentu seperti rata-rata (*mean*) atau nilai tengah (*median*), serta menggunakan metode imputasi lainnya yang lebih kompleks. Pemilihan metode penanganan biasanya disesuaikan dengan karakteristik *dataset* dan jumlah data yang hilang agar tidak mengurangi kualitas informasi yang terkandung dalam data. Penanganan yang tepat terhadap *missing value* dapat membantu meningkatkan kualitas *dataset* sehingga model *machine learning* dapat bekerja secara lebih efektif [27].

2.4.3 *Outlier*

Outlier merupakan data yang memiliki nilai sangat jauh dari pola umum atau distribusi mayoritas data. Kehadiran *outlier* perlu diperhatikan karena dapat memengaruhi analisis statistik, proses transformasi data, serta performa model *machine learning*, terutama pada data numerik. Nilai yang terlalu ekstrem dapat menyebabkan rata-rata dan standar deviasi menjadi tidak representatif, sehingga berdampak pada proses standardisasi maupun pembelajaran model. Oleh karena itu, identifikasi *outlier* menjadi salah satu tahap penting dalam data *preprocessing* [25].

Beberapa metode yang umum digunakan untuk mendeteksi *outlier* antara lain *boxplot* dan *Interquartile Range* (IQR) [27]. Metode IQR merupakan teknik statistik yang digunakan untuk mengidentifikasi nilai ekstrem berdasarkan distribusi kuartil data. IQR dihitung sebagai selisih antara kuartil ketiga (Q3) dan kuartil pertama (Q1), yang kemudian digunakan untuk menentukan batas bawah dan batas atas *outlier*. Data yang berada di luar rentang tersebut dikategorikan sebagai *outlier*. Metode ini banyak digunakan karena sederhana, tidak bergantung pada distribusi data, serta efektif dalam mendeteksi nilai ekstrem [37]. Pada penelitian berbasis data tabular, metode IQR sering digunakan karena mampu mengidentifikasi data yang berada di luar rentang kuartil secara sederhana namun efektif. Dengan melakukan pengecekan *outlier*, peneliti dapat memahami distribusi data dengan lebih baik dan menentukan apakah nilai ekstrem tersebut perlu dipertahankan atau ditangani lebih lanjut sesuai dengan konteks penelitian [27].

2.4.4 Pemisahan Fitur dan Target

Pada proses *machine learning*, *dataset* umumnya dipisahkan menjadi dua bagian utama yaitu variabel fitur (*feature*) dan variabel target (*label*). Variabel fitur merupakan atribut yang digunakan sebagai *input* bagi model untuk mempelajari pola dalam data, sedangkan variabel target merupakan nilai keluaran yang ingin diprediksi oleh model. Pemisahan ini penting dilakukan agar algoritma dapat mempelajari hubungan antara variabel *input* dan *output* secara efektif [26]. Dalam penelitian ini, variabel target yang digunakan adalah ***Success***, yang menunjukkan keberhasilan atau kegagalan UMKM. Sementara itu, variabel lainnya digunakan sebagai fitur *input* bagi model ANN untuk mempelajari pola yang memengaruhi keberhasilan usaha.

2.4.5 Standardisasi Data

Standardisasi data merupakan proses transformasi nilai fitur agar berada pada skala yang seragam sebelum digunakan dalam pelatihan model. Tahap ini penting pada algoritma berbasis jaringan saraf seperti ANN, karena perbedaan rentang nilai antar fitur dapat memengaruhi proses pembelajaran. Fitur dengan skala yang lebih besar cenderung menghasilkan nilai gradien yang lebih dominan, sehingga dapat menyebabkan pembaruan bobot menjadi tidak seimbang. Oleh karena itu, standardisasi dilakukan agar setiap fitur numerik memberikan kontribusi yang proporsional dalam proses pelatihan model [25].

Metode standardisasi yang digunakan dalam penelitian ini adalah *Z-score standardization*, yang mentransformasikan data berdasarkan nilai rata-rata (*mean*) dan

standar deviasi dari masing-masing fitur. Proses ini menghasilkan distribusi data dengan rata-rata mendekati nol dan standar deviasi sebesar satu. Dengan skala yang seragam, proses optimasi berbasis gradien pada model MLP dapat berlangsung secara lebih stabil dan efisien. Pemilihan metode *Z-score* didasarkan pada karakteristik *dataset* UMKM yang terdiri dari variabel numerik dengan rentang nilai yang bervariasi. Tanpa proses standardisasi, perbedaan skala antar fitur berpotensi menyebabkan *bias* dalam pembelajaran model. Oleh karena itu, *Z-score* dipilih untuk memastikan bahwa setiap fitur memiliki pengaruh yang seimbang, sehingga model dapat mempelajari pola hubungan antarvariabel [39].

Secara matematis, standardisasi *Z-score* dapat dirumuskan sebagai berikut [27][39]:

$$x' = \frac{x-\mu}{\sigma} \quad (6)$$

Di mana:

x = nilai data

μ = nilai rata-rata data

σ = standar deviasi data

x' = nilai data setelah standardisasi

Untuk menghitung nilai standardisasi *Z-score*, diperlukan nilai rata-rata (*mean*) dan standar deviasi dari setiap fitur. Nilai rata-rata digunakan untuk menentukan pusat distribusi data. Secara matematis, nilai rata-rata (*mean*) dapat dihitung menggunakan rumus berikut [27]:

$$\mu = \frac{\sum x_i}{n} \quad (7)$$

Di mana:

μ = nilai rata-rata data

$\sum xi$ = jumlah keseluruhan nilai data

n = jumlah seluruh data

Selanjutnya, standar deviasi dihitung untuk mengukur tingkat penyebaran data terhadap rata-ratanya. Semakin besar nilai standar deviasi, maka semakin besar variasi data dalam suatu fitur. Rumus standar deviasi dapat dituliskan sebagai berikut [27]:

$$\sigma = \sqrt{\frac{\sum(x_i-\mu)^2}{n-1}} \quad (8)$$

Di mana:

σ = standar deviasi data

x_i = nilai data ke-i

μ = nilai rata-rata data

n = jumlah seluruh data

$\sum(x_i - \mu)^2$ = jumlah kuadrat selisih antara nilai data dan rata-rata

2.5 *Confusion Matrix*

Setelah proses pelatihan selesai, model dievaluasi untuk mengukur kemampuan generalisasi terhadap data yang belum pernah dilihat sebelumnya. Evaluasi performa model klasifikasi umumnya dilakukan menggunakan *confusion matrix*, yaitu tabel yang menggambarkan perbandingan antara kelas hasil prediksi model dengan kelas sebenarnya [25]. *Confusion matrix* memberikan gambaran detail mengenai jenis kesalahan yang dilakukan model dan tidak hanya menampilkan tingkat akurasi secara keseluruhan. Pada kasus klasifikasi biner, *confusion matrix* terdiri atas empat komponen utama, yaitu [27]:

1. *True Positive* (TP): jumlah data positif yang diprediksi benar sebagai positif.
2. *True Negative* (TN): jumlah data negatif yang diprediksi benar sebagai negatif.
3. *False Positive* (FP): jumlah data negatif yang diprediksi salah sebagai positif (*Type I Error*).
4. *False Negative* (FN): jumlah data positif yang diprediksi salah sebagai negatif (*Type II Error*).

Secara konseptual, *confusion matrix* dapat direpresentasikan dalam bentuk tabel 2 × 2 yang membandingkan kelas aktual dan kelas prediksi. Berdasarkan nilai-nilai tersebut, dihitung beberapa metrik evaluasi untuk menilai performa model, yaitu [27]:

1. *Accuracy*

   Accuracy menunjukkan proporsi jumlah prediksi yang benar terhadap seluruh data yang diuji, dirumuskan sebagai:

   $$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{.....(9)}$$

   Metrik ini digunakan untuk memberikan gambaran umum mengenai performa model klasifikasi secara keseluruhan.

2. *Precision*

   Precision mengukur ketepatan model dalam memprediksi kelas positif, yaitu proporsi prediksi positif yang benar.

   $$Precision = \frac{TP}{TP+FP} \quad \text{.....(10)}$$

Nilai *Precision* yang tinggi menunjukkan bahwa model jarang melakukan kesalahan dalam memprediksi kelas positif.

3. *Recall*

Recall mengukur kemampuan model dalam mendeteksi seluruh data positif yang sebenarnya ada.

$$Recall = \frac{TP}{TP+FN} \quad \text{.....(11)}$$

Recall yang tinggi menunjukkan bahwa model mampu menangkap sebagian besar data positif dan meminimalkan kesalahan *False Negative*.

4. *F1-score*

F1-score merupakan rata-rata harmonik antara *Precision* dan *Recall*, yang digunakan untuk menyeimbangkan kedua metrik tersebut.

$$F1 - Score = 2 \times \frac{Precicion \times Recall}{Precicion+Recall} \quad \text{.....(12)}$$

Metrik ini sangat berguna untuk memberikan penilaian performa model secara lebih komprehensif, karena menggabungkan nilai *Precision* dan *Recall* dalam satu ukuran evaluasi.

Dengan demikian, penggunaan *confusion matrix* dalam penelitian ini tidak hanya berfokus pada nilai *accuracy*, tetapi juga mempertimbangkan metrik *precision*, *recall*, dan *F1-score.* Hal ini dilakukan untuk memberikan evaluasi yang lebih menyeluruh terhadap kinerja model ANN. Hasil evaluasi ini selanjutnya digunakan sebagai dasar dalam menganalisis kemampuan model dalam mengklasifikasikan keberhasilan UMKM serta mendukung penarikan kesimpulan penelitian.

Selain mengevaluasi hasil prediksi model menggunakan data pengujian, proses pelatihan model juga perlu diamati untuk mengetahui bagaimana model mempelajari pola data selama setiap *epoch*. Salah satu indikator yang digunakan untuk mengevaluasi proses pembelajaran tersebut adalah *validation loss*. Berbeda dengan *confusion matrix* yang digunakan setelah pelatihan selesai, *validation loss* diamati selama proses pelatihan berlangsung sehingga dapat memberikan informasi mengenai kemampuan model dalam melakukan generalisasi terhadap data yang belum pernah dipelajari.

Validation loss merupakan nilai fungsi kerugian (*loss function*) yang dihitung menggunakan data validasi selama proses pelatihan model. Berbeda dengan *training loss* yang dihitung menggunakan data pelatihan untuk memperbarui bobot model, *validation loss* digunakan untuk mengukur kemampuan model dalam melakukan prediksi terhadap data yang tidak terlibat dalam proses pembelajaran sehingga dapat menggambarkan

kemampuan generalisasi model terhadap data baru. Nilai *validation loss* yang rendah dan stabil menunjukkan bahwa model mampu mempertahankan performanya pada data yang belum pernah dipelajari sebelumnya.

Selama proses pelatihan, nilai *validation loss* diamati pada setiap *epoch* dan dibandingkan dengan *training loss*. Apabila kedua kurva menunjukkan tren penurunan yang relatif serupa, maka model dianggap mampu mempelajari pola data dengan baik. Sebaliknya, apabila *training loss* terus menurun sementara *validation loss* mulai meningkat, kondisi tersebut mengindikasikan terjadinya *overfitting*, yaitu keadaan ketika model terlalu menyesuaikan diri terhadap data pelatihan sehingga performanya menurun pada data baru. Oleh karena itu, *validation loss* digunakan sebagai salah satu indikator dalam mengevaluasi proses pelatihan model, namun penentuan kualitas model tetap perlu mempertimbangkan metrik evaluasi lain, seperti *accuracy*, *precision*, *recall*, dan *F1-score*, agar diperoleh penilaian yang lebih komprehensif.

2.6 *Permutation Feature Importance*

Permutation Feature Importance merupakan salah satu metode yang digunakan untuk mengetahui tingkat kontribusi setiap variabel terhadap hasil prediksi model. Metode ini termasuk ke dalam pendekatan *model-agnostic*, yaitu metode yang dapat diterapkan pada berbagai jenis model *machine learning* tanpa bergantung pada struktur internal model tersebut. Hal ini membuat *Permutation Feature Importance* dapat digunakan pada model ANN yang memiliki proses pembelajaran kompleks dan sulit dijelaskan hanya melalui bobot antar *neuron*. Dengan metode ini, setiap fitur dapat dianalisis berdasarkan pengaruhnya terhadap performa model sehingga faktor-faktor yang paling berkontribusi terhadap hasil prediksi dapat diidentifikasi [40].

Cara kerja *Permutation Feature Importance* dilakukan dengan membandingkan performa awal (*baseline*) model dengan performa model setelah nilai pada salah satu fitur diacak (*permuted*). Pertama, model yang telah dilatih digunakan untuk menghasilkan nilai performa awal (*baseline*). Selanjutnya, nilai pada setiap fitur diacak secara bergantian sehingga hubungan antara fitur tersebut dengan variabel target terganggu, kemudian model kembali digunakan untuk melakukan prediksi. Pada penelitian ini, performa model diukur menggunakan nilai *F1-score*, sehingga tingkat kepentingan setiap fitur ditentukan berdasarkan besarnya penurunan nilai *F1-score* setelah proses pengacakan dilakukan. Semakin besar penurunan nilai *F1-score* yang terjadi, maka semakin besar pula kontribusi fitur tersebut terhadap hasil prediksi model. Sebaliknya, apabila perubahan nilai *F1-score*

relatif kecil, maka fitur tersebut dianggap memiliki pengaruh yang lebih rendah terhadap proses prediksi [40].