BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian mengenai pengenalan pola tulisan tangan menggunakan metode *Convolutional Neural Network* (CNN) telah banyak dikembangkan dalam beberapa tahun terakhir. Salah satu kontribusi awal datang dari Elsawy dkk. (2017) yang memanfaatkan arsitektur CNN klasik *LeNet-5* untuk pengenalan huruf hijaiyah tulisan tangan. Mereka menggunakan dataset AHCD, yang terdiri dari 16.800 gambar karakter Arab dalam format *grayscale*. Meskipun model yang digunakan cukup sederhana dan belum mengintegrasikan teknik regularisasi modern seperti *dropout* atau *batch normalization*, hasil yang diperoleh cukup impresif dengan akurasi mencapai 94,9%, menunjukkan efektivitas struktur CNN konvensional untuk klasifikasi karakter Arab.

Setahun setelahnya, Amrouch dkk. (2018) memperkenalkan pendekatan yang menggabungkan CNN untuk ekstrak fitur otomatis dan *Hidden Markov Models* (HMM) untuk klasifikasi. Penelitian ini menggunakan dataset IFN/ENIT, yang berisi tulisan tangan berupa kata-kata Arab. Dengan memanfaatkan CNN untuk ekstraksi fitur dan HMM untuk mengelola urutan karakter, pendekatan ini memberikan akurasi sebesar 89,23%. Meskipun meningkatkan efisiensi dibandingkan metode berbasis fitur manual, pemisahan antara tahap ekstraksi dan klasifikasi menyebabkan sistem menjadi lebih kompleks.

Kontribusi berikutnya berasal dari Akil & Chaidir (2021) yang menggunakan CNN dengan beberapa lapisan konvolusi dan pooling serta fungsi aktivasi *ReLU*. Penelitian ini kembali menggunakan dataset AHCD dan menekankan pada integrasi proses ekstraksi fitur dan klasifikasi dalam satu alur kerja (*pipeline*) CNN. Model diproses melalui lapisan *fully connected* dan aktivasi *Softmax*. Walaupun model ini berhasil mencapai akurasi sebesar 91,13%, ia belum menerapkan mekanisme regularisasi seperti *dropout*, yang penting untuk mengurangi overfitting terutama pada data tulisan tangan yang memiliki banyak variasi.

Pendekatan yang lebih kompleks kemudian dikembangkan oleh Alheraki dkk. (2023) Mereka membangun model CNN yang terdiri dari empat blok konvolusi dengan peningkatan jumlah filter secara bertahap hingga 384. Selain itu, diterapkan *batch normalization* di setiap blok untuk menjaga

kestabilan pelatihan, serta digunakan *LeakyReLU* sebagai fungsi aktivasi dan *dropout* sebagai teknik regularisasi. Penelitian ini menggunakan dua dataset, yaitu Hijja (berisi tulisan tangan anak-anak) dan AHCD untuk menguji model terhadap variasi tulisan dari kelompok usia berbeda. Model akhir mampu mencapai akurasi hingga 96%.

Pada tahun yang sama, Daood dkk. (2023) mengusulkan sistem deteksi dan pengenalan karakter tulisan tangan Arab dalam skala lebih luas. Tidak hanya membangun model CNN untuk klasifikasi, mereka juga merancang proses pengumpulan data dari awal, dengan melibatkan 150 partisipan yang mengisi formulir berisi angka dan huruf Arab. Hasilnya adalah dataset baru berisi lebih dari 18.000 karakter. Model CNN yang digunakan menggunakan pendekatan *transfer learning MobileNetV2*, dan dievaluasi bersama teknik ekstraksi fitur lain seperti *Histogram of Oriented Gradients* (HOG), *Scale-Invariant Feature Transform* (SIFT), dan *Local Binary Patterns Histograms* (LBPH). Meskipun pendekatan ini menunjukkan akurasi hingga 94%, fokus utama penelitian lebih condong pada pengembangan sistem menyeluruh dibanding eksplorasi arsitektur CNN secara spesifik.

Tabel 2.1 Tabel Penelitian Terdahulu

Penulis	Judul	Metode	Dataset	Hasil
Ahmed El-Sawy, Mohamed Loey, dan Hazem El-Bakry (2017)	*Arabic Handwritten Characters Recognition using Convolutional Neural Network*	Menggunakan model *CNN*. Fokus pada klasifikasi huruf hijaiyah	AHCD	Akurasi sebesar 94,9%
Mustapha Amrouch, Mouhcine Rabi, dan Youssef Es-Saady (2018)	*Convolutional Feature Learning and CNN Based HMM for Arabic Handwriting Recognition*	CNN sebagai ekstraktor fitur, klasifikasi dilakukan menggunakan HMM (*Hidden Markov Models*). Fokus pada pengenalan huruf hijaiyah.	IFN/ENIT	Akurasi sebesar 89,23%

Penulis	Judul	Metode	Dataset	Hasil
Akil & Chaidir (2021)	Deteksi Karakter Huruf Arab dengan Menggunakan *Convolutional Neural Network*	Model CNN dengan training menggunakan *Deeplearning4j*. Fokus pada huruf hijaiyah.	AHCD	Akurasi sebesat 91,13%
Mais Alheraki, Rawan Al-Matham, dan Hend Al-Khalif (2023)	*Handwritten Arabic Character Recognition for Children Writing Using Convolutional Neural Network and Stroke Identification*	Model CNN dengan pendekatan *multi-model* berbasis jumlah goresan (*strokes*). Fokus pada tulisan tangan anak-anak.	Hijja dan AHCD	Akurasi sebesar 96%
Amar Daood, Ali Al-Saegh, dan Ahlam Fadhil Mahmood (2023)	*Handwriting Detection and Recognition of Arabic Numbers and Characters Using Deep Learning*	Menggunakan model CNN dan transfer learning (*MobileNetV2*), disertai segmentasi karakter otomatis. Fokus pada huruf dan angka Arab.	Dataset dikembangkan sendiri	Akurasi sebesar 94%

Penelitian ini bertujuan untuk mengembangkan model *Convolutional Neural Network* (CNN) yang dirancang khusus untuk mengenali tulisan tangan huruf hijaiyah dan angka Arab secara bersamaan dengan menggunakan dua dataset publik, yaitu AHCD dan AHDD. Berbeda dengan penelitian sebelumnya yang umumnya hanya fokus pada huruf atau angka secara terpisah dan belum mengeksplorasi tuning parameter secara sistematis, penelitian ini menawarkan kebaruan dalam bentuk eksplorasi hyperparameter

secara menyeluruh menggunakan metode *Grid Search*, sehingga menghasilkan model CNN yang lebih optimal. Dengan demikian, hasil akhir dari penelitian ini adalah sebuah model siap pakai yang dapat dijadikan dasar untuk sistem pembelajaran berbasis tulisan tangan.

2.2 Landasan Teori

2.2.1 Huruf Hijaiyah

Huruf hijaiyah merupakan elemen dasar dalam pembelajaran bahasa Arab dan menjadi komponen fundamental dalam membaca Al-Qur'an. Seperti yang dijelaskan oleh (Kamalia Nabila dkk., 2024), huruf hijaiyah memiliki peran penting karena menjadi pintu masuk untuk mempelajari tajwid, yang merupakan fondasi dalam membaca Al-Qur'an secara benar. Pemahaman yang baik terhadap huruf hijaiyah akan membantu seseorang dalam melafalkan ayat-ayat Al-Qur'an dengan tepat sesuai dengan kaidah yang berlaku.

Menurut (Imroatun, 2017), huruf hijaiyah terdiri dari dua bentuk, yaitu *mufrad* (tunggal) dan *muzdawij* (berangkai), yang ditulis serta dibaca dari kanan ke kiri. Bentuk huruf-huruf ini beragam, dengan beberapa huruf memiliki kemiripan yang dibedakan oleh jumlah serta posisi titik. Huruf hijaiyah dapat memiliki satu, dua, atau tiga titik yang terletak di atas, di dalam, atau di bawah huruf. Secara umum, huruf hijaiyah merujuk pada sistem ejaan dalam bahasa Arab, yang merupakan bahasa asli Al-Qur'an. Dengan kata lain, penguasaan huruf hijaiyah menjadi dasar utama dalam membaca dan memahami Al-Qur'an dengan baik.

2.2.2 *Deep Learning*

Deep Learning merupakan salah satu cabang dari *artificial intelligence* (AI) dan *machine learning* yang mengembangkan jaringan saraf tiruan dengan banyak lapisan (multi-layer). Teknologi ini dirancang menyerupai cara kerja otak manusia, di mana neuron-neuron buatan saling terhubung untuk membentuk jaringan kompleks. Metode ini mampu memproses data melalui transformasi non-linear secara bertingkat (hierarki), yang menjadikannya efektif dalam tugas-tugas seperti deteksi objek, pengenalan suara, dan penerjemahan bahasa (Raup dkk., 2022). Terdapat contoh beberapa jenis algoritma dalam *Deep Learning*, antara lain:

1) *Deep Belief Network* (DBN)

Algoritma *Deep Belief Network* (DBN) pertama kali diperkenalkan oleh Hinton dan koleganya, bersamaan dengan metode pembelajaran *greedy unsupervised* yang membentuk jaringan secara bertahap satu lapisan pada satu waktu. Penambahan lapisan pada DBN dilakukan secara *greedy*, dimana setiap lapisan baru mempelajari representasi dari lapisan sebelumnya untuk mengekstraksi *higher-level dependencies* dari input awal. Pendekatan ini memungkinkan jaringan untuk mengenali pola yang stabil dalam data secara lebih efektif (Ikawahyuni, 2017).

2) *Recurrent Neural Network* (RNN)

Recurrent Neural Networks (RNN) merupakan jenis jaringan saraf tiruan yang secara khusus dirancang untuk menangani data sekuensial atau deret waktu. Tidak seperti jaringan saraf konvensional, RNN memiliki mekanisme umpan balik internal yang memungkinkan model untuk mempertahankan informasi dari satu langkah ke langkah berikutnya dalam sebuah urutan. Fitur ini menjadikan RNN sangat efektif dalam mengenali pola-pola dinamis yang bergantung pada urutan, seperti pada tugas pemrosesan bahasa alami, prediksi deret waktu, dan pengenalan ucapan. Dalam konteks prediksi beban jaringan, kemampuan RNN untuk menangkap ketergantungan temporal menjadikannya algoritma yang efisien dan relevan (Annisah dkk., 2024).

3) *Generative Adversarial Networks* (GAN)

Generative Adversarial Networks (GAN) merupakan salah satu inovasi penting dalam perkembangan kecerdasan buatan yang diperkenalkan oleh Ian Goodfellow dan rekan-rekannya pada tahun 2014. GAN terdiri dari dua jaringan saraf yang saling berkompetisi, yaitu *generator* dan *discriminator*. Jaringan generator berfungsi menghasilkan data atau citra baru yang menyerupai data asli, sedangkan jaringan discriminator bertugas membedakan antara citra asli dan citra hasil buatan generator. Melalui proses pelatihan yang bersifat kompetitif ini, GAN mampu meningkatkan kualitas hasil secara bertahap hingga mencapai tingkat realisme yang tinggi.

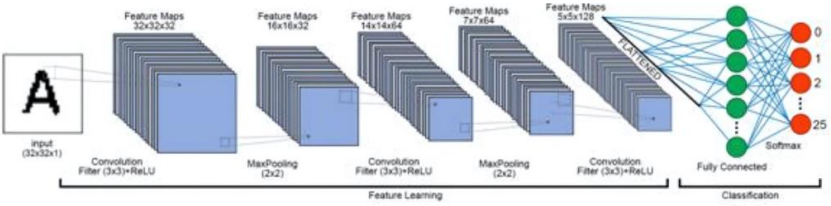
Kemampuan ini menjadikan GAN sangat efektif dalam menghasilkan citra sintetis yang sulit dibedakan dari citra nyata (Syaifudin, 2020).

4) *Deep Reinforcement Learning* (DRL)

Deep Reinforcement Learning (DRL) merupakan cabang dari *machine learning* yang memungkinkan model belajar mengambil tindakan optimal melalui interaksi langsung dengan lingkungan untuk memaksimalkan hasilnya. Metode ini menonjol karena kemampuannya dalam menyelesaikan permasalahan pengambilan keputusan kompleks di berbagai bidang, seperti robotika, permainan digital, dan keuangan. Berbeda dengan metode pembelajaran tradisional, DRL belajar langsung dari data mentah sensorik, menjadikannya efektif untuk menangani ruang keadaan berdimensi tinggi serta dinamika sistem yang kompleks (Atti & Yogi, 2024).

2.2.3 *Convolutional Neural Network*

Convolutional Neural Network (CNN) merupakan salah satu metode dalam machine learning yang dikembangkan dari arsitektur *Multi Layer Perceptron* (MLP) dan dirancang secara khusus untuk mengolah data dalam bentuk dua dimensi. CNN termasuk ke dalam kategori *deep neural network* karena memiliki struktur hierarki jaringan yang dalam dan telah banyak diaplikasikan dalam pengolahan data citra. Dalam implementasinya, CNN terdiri dari dua tahapan utama, yaitu proses klasifikasi yang dilakukan dengan pendekatan *feedforward* dan proses pelatihan jaringan yang dilakukan menggunakan algoritma *backpropagation* (Christiawan dkk., 2023). Untuk lebih jelas dapat dilihat Gambar 2.1



Gambar 2.1 Komponen dari CNN (Sumber: Silalahi dkk., 2020)

Dari penelitian Alzubaidi dkk. (2021), beberapa penjelasan dari komponen-komponen tersebut yaitu:

1) *Convolutional Layers*

Convolutional layer adalah komponen fundamental dari CNN yang berfungsi sebagai ekstraktor fitur otomatis dari data visual. Pada lapisan ini, dilakukan operasi konvolusi antara citra input dengan sejumlah *kernel* atau *filter* yang terlatih. Filter ini dapat mendeteksi pola visual seperti tepi, tekstur, sudut, atau bentuk spesifik tergantung pada bobot yang dipelajari. Proses konvolusi menghasilkan *feature maps* yang mewakili keberadaan fitur-fitur tertentu dalam citra pada posisi spasial tertentu. Dengan menyusun beberapa convolutional layer secara bertingkat, CNN mampu mempelajari fitur-fitur dari yang paling rendah (low-level) hingga tinggi (high-level) secara hierarkis (Alzubaidi dkk., 2021).

2) Fungsi Aktivasi

Setelah proses konvolusi, hasilnya dilewatkan ke fungsi aktivasi untuk memperkenalkan non-linearitas. Fungsi aktivasi membantu jaringan mempelajari hubungan kompleks antar fitur. Fungsi yang paling umum adalah ReLU (Rectified Linear Unit), namun arsitektur modern juga menggunakan Leaky ReLU, ELU, atau Swish untuk menangani kelemahan ReLU seperti *dying neuron*. Pemilihan fungsi aktivasi sangat memengaruhi konvergensi dan akurasi pelatihan (Alzubaidi dkk., 2021).

3) *Pooling Layers*

Pooling digunakan untuk melakukan *downsampling* pada *feature maps*, mengurangi dimensi spasial dan mempercepat proses pelatihan sambil mempertahankan informasi utama. Jenis pooling yang umum adalah max pooling, yang memilih nilai tertinggi dalam area lokal, dan average pooling, yang menghitung nilai rata-rata. Pooling membantu meningkatkan ketahanan terhadap translasi dan distorsi kecil pada citra masukan (Alzubaidi dkk., 2021).

4) *Normalization Layer*

Dalam CNN modern, Batch Normalization digunakan untuk menormalkan output dari setiap lapisan sebelum diteruskan ke lapisan berikutnya. Hal ini membantu menstabilkan dan mempercepat pelatihan jaringan dengan mengurangi masalah internal covariate shift. Normalisasi juga memungkinkan penggunaan learning rate yang lebih tinggi dan mengurangi ketergantungan terhadap inisialisasi parameter (Alzubaidi dkk., 2021).

5) *Dropout Layer*

Dropout adalah teknik regularisasi yang digunakan untuk mencegah *overfitting* dengan cara mematikan sejumlah unit neuron secara acak selama pelatihan. Dengan ini, jaringan tidak terlalu bergantung pada neuron tertentu

dan terdorong untuk belajar representasi yang lebih umum. Dropout sering diterapkan pada lapisan fully connected, tetapi juga dapat digunakan pada lapisan lain tergantung arsitektur (Alzubaidi dkk., 2021).

6) *Fully Connected Layers*

Fully Connected Layers merupakan tahap penting dalam arsitektur CNN, di mana seluruh fitur yang diekstrak oleh lapisan konvolusi sebelumnya diubah terlebih dahulu menjadi vektor satu dimensi melalui proses *flattening*. Setelah itu, semua neuron dalam lapisan ini akan terhubung penuh dengan neuron dari lapisan sebelumnya. Hal ini memungkinkan jaringan untuk mengintegrasikan seluruh informasi yang telah dipelajari dan menggunakannya untuk melakukan klasifikasi akhir. Berbeda dengan lapisan konvolusi yang hanya memiliki koneksi lokal, lapisan ini menciptakan koneksi global terhadap semua fitur yang telah dipelajari, sehingga memperkuat kemampuan klasifikasi sistem secara keseluruhan (Magdalena dkk., 2021).

2.2.4 *Hyperparameter*

Dalam pengembangan model pembelajaran mesin dan *deep learning*, *hyperparameter* adalah komponen yang sangat krusial karena secara langsung mempengaruhi performa dan proses pelatihan model. Beberapa diantaranya yaitu:

1) *Optimizer*

Optimizer adalah algoritma yang digunakan untuk menyesuaikan bobot dan bias pada model selama proses pelatihan, dengan tujuan utama meminimalkan nilai *loss*. Berbagai jenis *optimizer* dalam *deep learning*, seperti *Stochastic Gradient Descent* (SGD), *Adam*, dan *RMSProp*, memiliki karakteristik berbeda. SGD dikenal sederhana dan efisien dalam komputasi, namun umumnya membutuhkan lebih banyak iterasi untuk mencapai konvergensi. Sebaliknya, Adam dan RMSProp memanfaatkan pendekatan adaptif terhadap gradien, sehingga mampu mempercepat proses konvergensi dan meningkatkan efisiensi pelatihan (Kurniadi dkk., 2025).

2) *Kernel Initializer*

Kernel initializer adalah metode untuk menetapkan bobot awal dari layer dalam jaringan saraf. Inisialisasi ini penting untuk memastikan stabilitas dan efektivitas proses pelatihan, karena bobot awal yang buruk dapat

menyebabkan pelatihan lambat atau bahkan tidak konvergen. *Keras* menyediakan berbagai inisialisasi seperti *RandomNormal, GlorotUniform*, dan *HeNormal*, yang dapat digunakan baik secara langsung sebagai objek maupun melalui *identifier string* (Keras Team, n.d.).

3) *Activation Function*

Fungsi aktivasi digunakan dalam jaringan saraf untuk memperkenalkan non-linearitas ke dalam model, sehingga memungkinkan jaringan untuk mempelajari pola data yang kompleks. Keras menyediakan berbagai fungsi aktivasi seperti *relu, sigmoid, tanh*, dan lainnya, yang dapat digunakan langsung di dalam definisi layer atau melalui *Activation layer* tersendiri (Keras Team, n.d.).

4) *Learning Rate*

Seperti dijelaskan oleh Belcic dan Stryker (2024), *Learning rate* merupakan salah satu *hyperparameter* penting dalam machine learning yang menentukan seberapa besar perubahan atau penyesuaian parameter model dilakukan setiap kali algoritma optimasi dijalankan. Tujuan dari proses optimasi ini adalah untuk meminimalkan loss function, yaitu selisih antara prediksi model dan data sebenarnya. Dalam penelitian yang dilakukan oleh Igiri dkk., (2021) menjelaskan bahwa learning rate dapat menjadi masalah ketika jumlah sampel pelatihan terlalu besar atau jumlah neuron dalam jaringan melebihi dimensi input vektor.

2.2.5 *Arabic Handwritten Characters Dataset* (AHCD)

AHCD adalah dataset yang terdiri dari 16.800 gambar karakter tulisan tangan huruf Arab. Dataset ini dibuat melalui kontribusi 60 partisipan dengan rentang usia 19 hingga 40 tahun, di mana 90% dari mereka adalah penulis tangan kanan. Setiap partisipan diminta untuk menulis setiap karakter huruf Arab (dari ’alif’ hingga ’ya’) sebanyak 10 kali.

Karakter tulisan tangan pada dataset ini dipindai dengan resolusi 300 dpi, dan proses segmentasi setiap blok huruf dilakukan secara otomatis menggunakan Matlab 2016a. Dataset AHCD dibagi menjadi dua bagian utama. Training Set, terdiri dari 13.440 gambar (480 gambar per kelas). Dan Test Set, terdiri dari 3.360 gambar (120 gambar per kelas), digunakan untuk menguji performa model.

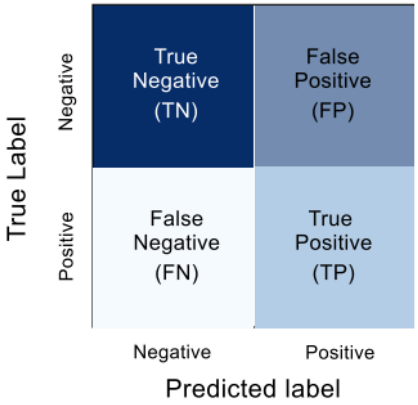
2.2.6 *Arabic Handwritten Digits Dataset* (AHDD)

AHDD adalah dataset yang terdiri dari 70.000 gambar digit angka tulisan tangan huruf Arab. Dataset ini dibuat melalui kontribusi 700 partisipan Untuk memastikan variasi gaya tulisan tangan, dataset ini dikumpulkan dari berbagai institusi, termasuk Fakultas Teknik, Hukum, Kedokteran, Universitas Terbuka, sekolah menengah, serta institusi pemerintah. Setiap partisipan diminta untuk menulis setiap setiap digit angka Arab (dari 0 hingga 9) sebanyak 10 kali.

Dataset AHDD dibagi menjadi dua bagian utama. Training Set, terdiri dari 60.000 gambar (6.000 gambar per kelas). Dan Test Set, terdiri dari 10.000 gambar (1.000 gambar per kelas), digunakan untuk menguji performa model.

2.2.7 Confusion Matrix

Menurut (Raihan dkk., 2021), confusion matrix merupakan sebuah tabel berukuran $N \times N$ (di mana N adalah jumlah kelas atau kategori) yang digunakan untuk menampilkan jumlah prediksi yang benar dan salah dalam suatu model klasifikasi. Matriks ini berfungsi sebagai alat untuk membandingkan nilai aktual dengan nilai yang diprediksi oleh model. Dalam tabel ini, baris merepresentasikan kelas sebenarnya, sementara kolom menunjukkan kelas hasil prediksi. Nilai dalam confusion matrix diklasifikasikan ke dalam empat kategori utama, sebagaimana ditampilkan pada Gambar 2.2.



Gambar 2.2 Confusion Matrix (Sumber: Vysakh dkk., 2023)

1. *True Positive* (TP) : Data aktual positif dan diprediksi sebagai positif.
2. *True Negative* (TN) : Data aktual negatif dan diprediksi sebagai negatif.
3. *False Positive* (FP) : Data aktual negatif dan diprediksi sebagai positif.
4. *False Negative* (FN) : Data aktual positif dan diprediksi sebagai negatif.

Berdasarkan hasil dari *confusion matrix* yang diperoleh, dilakukan tahap evaluasi untuk mengukur efektivitas model klasifikasi:

1) *Accuracy*: Digunakan untuk mengukur efektivitas keseluruhan klasifikasi. Dihitung menggunakan persamaan 1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FI}$$

2) True Positive Rate (Recall): Mengukur seberapa efektif model dalam mengidentifikasi label positif. Dihitung menggunakan persamaan 2.

$$Recall = \frac{TP}{TP + FN}$$

3) True Negative Rate (Specificity): Mengukur seberapa efektif model dalam mengidentifikasi label negatif. Dihitung menggunakan persamaan 3.

$$Recall = \frac{TN}{TN + FP}$$

4) Positive Predictive Value (Precision): Presentase data dengan label positif yang benar-benar positif berdasarkan prediksi model. Dihitung menggunakan persamaan 4.

$$Precision = \frac{TP}{TP + FP}$$

5) AUC (Area Under Curve) digunakan untuk mengevaluasi kinerja model prediksi dengan mempertimbangkan sensitivitas dan spesifisitas. Nilai AUC berkisar antara 0 hingga 1, di mana semakin tinggi nilainya, semakin baik kemampuan model dalam melakukan klasifikasi. Dihitung menggunakan persamaan 5.

$$AUC = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right)$$

6) F1-Score: Mengukur rata-rata tertimbang dari presisi dan recall. Dihitung menggunakan persamaan 6

$$F1 - score = 2\left(\frac{recall * precision}{recall + precision}\right)$$