

BAB II DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian telah dilakukan pada Deteksi Penyakit Daun pada Tanaman Padi Menggunakan Algoritma *Decision Tree*, *Random Forest*, *Naïve Bayes*, SVM dan KNN 2020. Pengenalan jenis-jenis hama invasif merupakan langkah awal yang sangat penting dalam membantu keberhasilan upaya pengendalian. Oleh karena itu, dengan menggunakan metode KNN mendapatkan nilai akurasi 87%, karena model tersebut berkinerja baik secara konsisten pada kedua evaluasi. KNN belum terbukti memiliki masalah *overfitting* karena secara konsisten bekerja dengan baik pada data latih dan uji [8].

Dalam penelitian sebelumnya menggunakan objek yang berbeda yaitu doodle. Implementasi metode *convolutional neural network* menggunakan arsitektur LeNet-5 untuk Pengenalan *doodle*. Sampai saat ini, pengenalan objek telah banyak digunakan dalam berbagai bidang, seperti pengenalan tulisan tangan. Penelitian ini memanfaatkan kemampuan CNN dalam menggunakan arsitektur LeNet 5 untuk menampilkan jenis doodle pada lima objek gambar, yaitu baju, celana, kursi, kupu-kupu, dan sepeda. Setiap item *Doodle* terdiri dari 30 gambar dengan total 150 gambar. Hasil pengujian menunjukkan bahwa akurasinya adalah 93% hingga 98% [9].

Beberapa penelitian tentang deteksi penyakit daun tanaman telah dilakukan sebelumnya. Salah satunya Deteksi Penyakit Pada Daun Tanaman Padi Menggunakan Metode *Convolutional Neural Network* 2021. Deteksi penyakit dilakukan pada penelitian ini menggunakan CNN. Citra melewati proses *resizing*, *cropping* dan ekstraksi fitur. Dataset yang digunakan adalah dataset *RiceLeafs* yang berisi total 900 citra sampel daun padi, 780 citra sebagai data latih dan 180 citra sebagai data validasi. Persentase akurasi rata-rata dari hasil pengujian model yang dibuat adalah 92% dari data pelatihan [2].

Penelitian selanjutnya juga melakukan Identifikasi jenis tumbuhan Berdasarkan Citra Daun Menggunakan *Convolutional Neural Network* 2018, Metode yang digunakan dalam penelitian ini yaitu *Alexnet*, Arsitektur CNN

Krizhevsky dengan delapan layer ekstraksi fitur. Akurasi rata-rata hasil klasifikasi adalah 85%. Pada saat yang sama, akurasi dari identifikasi mencapai 90% yang ditentukan dengan menguji 40 gambar [9].

Penelitian lain menggunakan metode CNN untuk Pengelompokan Jenis Candi Berdasarkan Bentuk Dan Modelnya PADA YOLO v3 2021. Metode algoritma *Convolutional Neural Network* (CNN) dapat diterapkan untuk mengidentifikasi tipe candi, dimana candi tersebut memiliki cirinya sendiri. Algoritma melewati fase klasifikasi menggunakan *backpropagation* dan tahap pembelajaran yaitu setiap *neuron* dipresentasikan dalam dua dimensi. Metode pengembangan sistem berdasarkan kecerdasan buatan ini akan belajar dari data yang diinputkan sehingga semakin banyak data input yang diberikan maka semakin cerdas dalam menarik kesimpulan. Hasil penelitian ini menunjukkan akurasi lebih dari 70%, sehingga CNN yang ada pada YOLO v3 ini baik untuk mendeteksi bentuk candi [10].

Tabel 2. 1 Rangkuman Keterkaitan dengan Penelitian Sebelumnya

| Judul Penelitian | Metode | Akurasi |
|---|--|---------|
| Klasifikasi Penyakit Tanaman Padi Menggunakan Model <i>Deep Learning Efficientnet B3</i> Dengan <i>Transfer Learning</i> (2021) | Dalam penelitian ini, metode <i>Deep Learning</i> khususnya <i>Convolutional Neural Network</i> dengan arsitektur <i>EfficientNet B3</i> | 79.53% |
| Deteksi Penyakit Pada Daun Tanaman Padi Menggunakan Metode <i>Convolutional Neural Network</i> (2021) | <i>Convolutional Neural Network</i> (CNN) | 92% |
| <i>Fine-Tuning Model Transfer Learning VGG16</i> Untuk Klasifikasi Citra Penyakit Tanaman Padi (2022) | <i>Convolution neural network</i> dengan arsitektur VGG16 | 62.21% |

| Judul Penelitian | Metode | Akurasi |
|---|---|---------|
| Pengelan Jenis Candi Berdasarkan Bentuk Dan Modelnya Menggunakan Metode <i>Convolutional Neural Network</i> (CNN) Pada YOLLO v3 | <i>Convolutional Neural Network</i> (CNN) | 70% |
| Identifikasi Penyakit Tanaman Padi Melalui Citra Daun Menggunakan DenseNet 201 | Arsitektur <i>Densely Connected Convolutional Networks</i> (DenseNet)-201 | 83% |
| Klasifikasi Penyakit Tanaman Padi Menggunakan Metode <i>Convolutional Neural Network</i> Melalui Citra Daun (<i>Multilayer Perceptron</i>) (2022) | <i>Convolutional Neural Network</i> (CNN) | 91,7% |
| Klasifikasi Citra Buah Menggunakan <i>Convolutional Neural Network</i> | Algoritma CNN terdiri dari 3 layer utama yaitu <i>Convolutional Layer</i> , <i>Pooling Layer</i> , dan <i>Fully Connected Layer</i> . | 91.24% |
| Klasifikasi Kualitas Kayu Kelapa Menggunakan Arsitektur CNN (2021) | <i>Convolutional Neural Network</i> (CNN), arsitektur CNN yaitu, <i>AlexNet</i> , <i>GoogLeNet</i> , <i>ResNet101</i> , <i>ResNet18</i> , dan <i>ResNet50</i> | 77,18% |
| Deteksi Penyakit Daun pada Tanaman Padi | Algoritma <i>Decision Tree</i> , <i>Forest Random</i> , | 87% |

| Judul Penelitian | Metode | Akurasi |
|---|--|---------|
| Menggunakan Algoritma <i>Decision Tree, Forest Random, Bayes Naïve, SVM dan KNN</i> (2020). | <i>Bayes Naïve, SVM dan KNN</i> | |
| Implementasi metode <i>convolutional neural network</i> menggunakan arsitektur LeNet-5 untuk Pengenalan <i>doodle</i> . | <i>Convolutional Neural Network</i> , arsitektur LeNet-5 | 93% |
| Segmentasi Kerusakan Daun Padi pada Citra Digital | segmentasi otomatis | 0,20 |
| CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK MENENTUKAN GAGRAK WAYANG KULIT | <i>Convolutional Neural Network (CNN)</i> | 92,27% |
| Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network | <i>Convolutional Neural Network (CNN)</i> | 90% |
| Klasifikasi Citra Menggunakan | <i>Convolutional Neural Network</i> dan <i>K Fold Cross Validation</i> | 80,36% |

| Judul Penelitian | Metode | Akurasi |
|---|---|---------|
| Convolutional Neural Network dan K Fold Cross Validation | | |
| <i>Convolution Neural Network (CNN) Untuk Pengklasifikasian Citra Makanan Tradisional</i> | <i>Convolution Neural Network (CNN)</i> | 73% |

Dengan memahami penjelasan table 2.1 dan ingat juga pentingnya perkembangan teknologi pertanian di Indonesia khususnya dalam budidaya padi di Indonesia. Pada penelitian ini menggunakan metode *Convolutional Neural Network* dengan arsitektur *convolutional 7 layer*. Penerapan metode Algoritma *Convolutional Neural Network (CNN)* untuk membuat model deteksi dan klasifikasi penyakit daun tanaman padi. Penelitian ini mengklasifikasikan tiga jenis penyakit yaitu *brown, leafscald* dan *narrow*.

2.2 DASAR TEORI

2.2.1 Tanaman Padi

Tanaman padi merupakan tanaman yang istimewa karena tanaman padi dapat beradaptasi pada hampir semua lingkungan, dari dataran rendah hingga dataran tinggi (2000 m dpl), dari tropis hingga subtropis kecuali Antartika (daerah kutub) dan dari daerah lembab sampai daerah kering, daerah pinggiran yang subur (stres garam, aluminium, besi, asam organik, kekeringan, dll.). Tanaman padi merupakan jenis rumput-rumputan dengan tandan yang kuat dan banyak tumbuh anakan dari ruas-ruas yang berakar. Menurut tempat tumbuhnya, tanaman padi dapat dibedakan menjadi padi sawah, padi ladang dan padi rawa. Sistem pertanian padi sawah lebih dulu dikenal dari pada pertanian padi ladang. Budidaya padi dalam sistem sawah sudah ada sejak 6280 tahun SM, diketahui sistem ladang baru dikenal sekitar 2000 tahun SM.

Sistem penanaman padi pertama kali dikembangkan di Tiongkok sedangkan sistem padi pertama kali diidentifikasi di semenanjung Korea. Di Indonesia, sistem budidaya padi ladang berkembang lebih awal dibandingkan dengan sistem budidaya padi sawah. Hal ini mengacu pada pola budaya nenek moyang kita yang hidup secara nomaden (bergerak) sehingga ketika kesuburan tanah pertanian menurun, mereka membuka lahan baru untuk bercocok tanam berbagai tanaman, termasuk padi ladang [12]. Masyarakat mengenal ribuan tanaman padi di alam, namun tidak semuanya memiliki nilai ekonomis. Spesies yang dibudidayakan oleh petani biasanya adalah *Oryza sativa* L. *Oryza sativa* L adalah tanaman pangan dan sumber karbohidrat yang penting.

Tanaman padi memiliki satu daun, dua baris, terkadang terlihat beberapa baris. Pelepah daun berkembang dengan baik, seringkali terdapat lidah di tepi pelepah daun dan helaian daun. Tangkai daun sessile, hampir selalu lanset atau lurik di kedua sisi pelepah, dengan beberapa urat sejajar. Permukaan daunnya kasar dan menyempit di ujungnya. Panjang pelat daun sangat bervariasi, kebanyakan 100-150 cm. Warna daunnya hijau tua dan berubah menjadi kuning keemasan setelah tanaman dipanen. Penyakit padi merupakan cekaman biotis yang dapat mengurangi hasil dan bahkan menyebabkan gagal panen, berikut ciri dari penyakit daun padi:

1. *Brown Spot*

Penyakit beras bintik coklat yang sempit disebabkan oleh jamur *Cercospora oryzae* dan garis-garis pendek berwarna Bercak coklat muda, agak kemerahan dan memanjang ke urat daun tanaman *sensitive* rentan terhadap infeksi serius. Bercak nekrotik terjadi pada tanaman dewasa, berbentuk melingkar berwarna coklat dengan pusat berwarna abu-abu hingga putih dan tepian kemerahan. Penyakit ini dapat menyebabkan kerusakan pada daun dan menurunkan hasil panen seperti terlihat pada gambar 2.1 [13].



Gambar 2. 1 penyakit Brown Spot [13]

2. *Leaf Scald*

Leaf scald atau lepuh daun padi adalah penyakit yang disebabkan oleh bakteri *Xanthomonas oryzae* pv. *Oryzae*. Gejala dari penyakit ini adalah adanya kerusakan berair ringan pada daun, mulai dari ujung daun. Kerusakan kemudian melebar dan membentuk lepuhan pada helai daun. Daun yang terinfeksi kemudian menjadi layu. Penyakit ini biasanya terjadi pada ujung daun tua, namun dapat juga terjadi pada sepanjang pinggir dan bagian tengah daun seperti terlihat pada gambar 2.2 [13].



Gambar 2. 2 Penyakit Leaf Scald [13]

3. *Narrow Brown Spot*

Narrow brown spot atau bercak coklat sempit pada daun padi adalah penyakit yang disebabkan oleh jamur *Cochliobolus miyabeanus*. Gejala dari penyakit ini adalah adanya bercak-bercak coklat bundar atau oval dengan lingkaran kuning pada tahap anakan. Ketika bercak-bercak tersebut membesar, pusat abu-abu muncul di tengah-tengah dan tepian coklat kemerahan terlihat. Perubahan warna batang juga dapat menjadi gejala karakteristik lainnya. Infeksi pada kuntum menyebabkan pengisian bulir tidak sempurna atau terganggu dan menurunkan kualitas gabah seperti terlihat pada gambar 2.3 [13].



Gambar 2. 3 Penyakit Narrow Brown Spot [13]

2.2.2 Artificial Intelligence

Menurut John McCarthy, kecerdasan buatan adalah usaha memodelkan proses berpikir manusia dan mendesain mesin itu agar dapat menirukan perilaku manusia. Dilansir dari Techopedia, AI adalah area pengetahuan komputer yang menekankan pada pembuatan “mesin pintar” yang mampu bekerja dan bereaksi seperti manusia. “mesin“ tersebut bisa memahami bahasa manusia secara natural, mengenali wajah, dan lain sebagainya. Sistem AI dapat berbasis aturan, didorong oleh data, atau campuran keduanya. Sistem ini *supervised*, *unsupervised*, atau *semi-supervised*. Algoritme AI yang relevan dari algoritme tradisional, seperti regresi dan clustering, hingga teknik ML modern, seperti natural language *processing*, *deep learning* (DL), *convolutional networks*, dan *reinforcement learning*. Aplikasi AI di bidang pengenalan suara, terjemahan bahasa, pemrosesan video, dan pengenalan wajah dan objek [14].

AI Biasanya digunakan untuk pengambilan keputusan yang sangat sulit dan terstruktur, mempunyai basis pengetahuan ratusan atau ribuan aturan. *Machine learning* adalah bagaimana program komputer meningkatkan kinerja tanpa pemrograman eksplisit, mengenal pola, pengalaman, pembelajaran sebelumnya (database), dan pembelajaran yang diawasi dan yang tidak diawasi. Di mana sistem komputer belajar bagaimana melakukan tugas dari pada diprogram bagaimana melakukannya. Kualitas dan ukuran kumpulan data ini penting untuk membangun

sistem yang mampu melaksanakan tugas yang ditentukan secara akurat. Contohnya adalah *google search* dan rekomendasi youtube, *netflix*, dan aplikasi *e-commerce* seperti amazon. Sedangkan untuk neural networks adalah serangkaian algoritma yang berupaya mengenali hubungan mendasar dalam sekumpulan data melalui proses yang meniru cara otak manusia beroperasi.

Neural networks didesain persis seperti otak manusia agar dapat berpikir seperti otak manusia. Kunci keberhasilan mesin adalah *neural networks*. *Neural networks* dikerjakan untuk mengumpulkan data yang mengajarkan apa yang harus disajikan dengan data tertentu selama pelatihan dan meniru cara neuron biologis memberi sinyal satu sama lain. *Neural networks* bagus untuk pengenalan pola, klasifikasi, dan optimasi. Hal ini termasuk seperti pengenalan tulisan tangan, pengenalan wajah, pengenalan suara, terjemahan teks, deteksi penipuan kartu kredit, diagnosis medis, dan solusi untuk sejumlah besar data.

Kemungkinan sistem AI yang dapat menggantikan banyak tenaga kerja manual modern dan kemungkinan yang lebih kredibel di masa depan. Memasuki era digital yang semakin meningkat, berkarir di bidang AI adalah sesuatu yang sangat canggih yang dijanjikan. Banyak perusahaan yang nantinya akan merekrut orang dengan keahlian AI di masa depan. AI akan mampu menciptakan, mengubah, dan meningkatkan banyak aspek kehidupan manusia [15].

Visi komputer adalah bidang ilmu komputer, matematika, dan teknik elektro. Ini mencakup cara-cara untuk memperoleh, memproses, menganalisis, dan memahami gambar dan video dari dunia nyata untuk meniru penglihatan manusia. Juga, tidak seperti visi manusia, visi komputer juga dapat digunakan untuk menganalisis dan memproses kedalaman dan gambar inframerah. Sistem visi komputer dapat menerima berbagai bentuk data sebagai masukan, termasuk, namun tidak terbatas pada, gambar, urutan gambar, dan video yang dapat dialirkan dari berbagai sumber untuk diproses lebih lanjut dan mengekstrak informasi yang berguna untuk pengambilan keputusan [16].

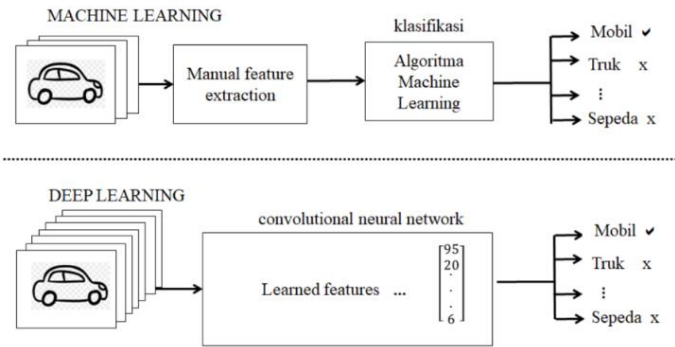
2.2.3 Deep Learning

Deep Learning adalah cabang ilmu machine learning berbasis Jaringan Saraf Tiruan (JST) atau bisa dikatakan sebagai perkembangan dari JST. Dalam *deep*

learning, sebuah komputer belajar mengklasifikasi secara langsung dari gambar atau suara [10]. *Deep learning* melakukan pembelajaran yang terdiri dari banyak layer. Layer awal menghasilkan fitur sederhana hingga layer akhir menghasilkan fitur-fitur yang lebih kompleks. *Deep Learning* diaplikasikan pada teknik klasifikasi, *clustering*, segmentasi ataupun *recognition*. *Deep Learning* cocok untuk data yang tidak teratur seperti teks, suara dan citra [17].

Deep Learning dikembangkan pada tahun 1950 namun baru tahun 1990 dapat di aplikasikan dengan sukses. *Learning* algoritma yang digunakan sekarang pada task yang kompleks hampir sama seperti *learning* algoritma yang digunakan untuk menyelesaikan masalah permainan pada tahun 1980, meskipun model algoritma yang digunakan berubah menjadi training yang sederhana dari arsitektur *deep learning*. Hal yang penting pada pengembangan model yang sekarang adalah kita dapat mendukung dengan *resource* yang dibutuhkan agar menjadi sukses [7]. Pengembangan data set yang semakin meningkat menyebabkan data set tersentralisasi yang memudahkan dalam pengelolaannya.

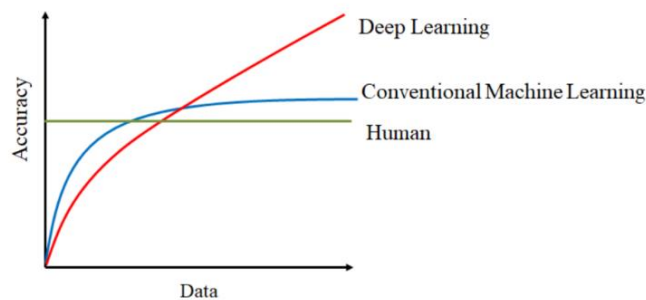
Salah satu implementasi dari *deep learning* yaitu pada data citra . Klasifikasi citra bertujuan untuk mengkategorikan citra pada kelas-kelas tertentu. Proses klasifikasi membutuhkan metode dari *machine learning* untuk dapat melakukan tugasnya. Selain itu, dibutuhkan proses ekstraksi fitur untuk dapat melakukan pengenalan citra. Perbedaan fitur yang dimiliki citra dapat menunjukkan perbedaan kategori. Pada *machine learning* konvensional, tahapan ekstraksi fitur dilakukan secara manual dengan memilih fitur-fitur yang akan digunakan secara mandiri. Fitur-fitur yang umum digunakan adalah fitur mon logi, warna, dan tekstur. Setiap fitur dari morfologi, warna dan tekstur memiliki jenis fitur yang beragam. Tidak semua fitur dipilih , hanya fitur fitur paling berpengaruh saja yang digunakan untuk proses selanjutnya. Teknik seleksi fitur yang tepat berpengaruh pada hasil klasifikasi citra. Saat ini telah berkembang luas tentang deep learning [17].



Gambar 2. 4 Machine Learning vs Deep Learning [17]

Perbedaan antara *machine learning* dan *deep learning* pada ekstraksi fitur dapat dilihat pada gambar 2.4 pada *machine learning* konvensional ekstraksi fitur dilakukan secara manual dengan menentukan terlebih dahulu jenis fitur yang akan digunakan. Setelah fitur-fitur berpengaruh didapatkan, selanjutnya diproses dengan algoritma *machine learning*. Pada *deep learning*, ekstraksi fitur dan klasifikasi dilakukan secara otomatis.

Deep learning merupakan teknik *machine learning* yang mengajarkan komputer untuk belajar dari data. *Deep learning* melakukan pembelajaran mendalam dengan data besar dan *network* dengan jumlah layer hingga ratusan. Semakin banyak sistem melakukan pembelajaran terhadap data, semakin baik sistem melakukan pengenalan. Data dapat dikenali dari fitur-fitur yang dimilikinya. Data dapat terdiri dari citra (*image*), teks, dan suara (*signal*). *Deep learning* dapat melakukan pengenalan dengan akurasi tinggi bahkan dapat melebihi pengenalan yang dilakukan oleh manusia. *Deep learning* membutuhkan dua syarat utama yaitu data besar dan diproses pada komputer dengan *High Performance Graphical Processing Unit* (GPU).



Gambar 2. 5 Perbandingan akurasi pengenalan dan data training [17]

Gambar 2.5 menunjukkan perbandingan data yang digunakan, akurasi pengenalan yang dapat dicapai oleh manusia, *machine learning* konvensional, dan *deep learning*. *Deep learning* memiliki tingkat akurasi lebih tinggi dibandingkan lainnya. Pada *deep learning*, semakin banyak data training yang digunakan, semakin tinggi akurasi pengenalan yang didapatkan. Salah satu metode *deep learning* adalah *Convolutional Neural Network* (CNN). Pada dasarnya, CNN merupakan pengembangan dari *Artificial Neural Network* (ANN) konvensional [17].

2.2.4 Supervised Learning

Supervised learning adalah salah satu metode dalam pembelajaran mesin (*machine learning*) di mana model mempelajari hubungan antara *input* dan *output* yang ditentukan sebelumnya. Pada *supervised learning*, data latih terdiri dari pasangan *input-output*, yang disebut sebagai pasangan "fitur-label" atau "*input-output*". Tujuan utama dari pembelajaran yang diawasi adalah untuk membuat model yang dapat memetakan masukan yang tidak diketahui ke keluaran yang sesuai berdasarkan pola yang ditemukan dalam data pelatihan [17].

1. Persiapan Data

Data pelatihan yang digunakan harus siap untuk diproses. Ini termasuk operasi seperti membersihkan data untuk nilai yang hilang atau tidak valid, mengonversi data kategorikal menjadi bentuk yang dapat diproses model, dan membagi data menjadi set pelatihan dan set validasi untuk menguji keefektifan model.

2. Pemilihan Model

Memilih model diperlukan dalam tahap ini karena akan digunakan untuk mempelajari pola pada data pelatihan. Contoh model yang sering digunakan dalam supervised learning adalah regresi linier, pohon keputusan, hutan acak, naif Bayes, dan jaringan saraf.

3. Pembelajaran Model

Model diuji terhadap data pelatihan menggunakan algoritma pembelajaran yang sesuai untuk jenis model yang dipilih. Algoritme ini menyesuaikan parameter model dengan data pelatihan dan meminimalkan kesalahan prediksi.

4. Evaluasi Model

Setelah model dilatih, kinerjanya dievaluasi terhadap validasi atau data uji yang sebelumnya tidak diketahui. Metrik evaluasi yang umum digunakan adalah presisi, akurasi, recall, F1-score, atau mean squared error (MSE), bergantung pada jenis masalah yang dipecahkan (klasifikasi atau regresi).

5. Penyempurnaan Model

Jika model tidak berjalan dengan baik, langkah ini mencakup penyesuaian parameter model atau perbaikan teknik pembelajaran untuk meningkatkan kinerja model. Ini mungkin melibatkan pemilihan fitur yang lebih tepat, pengoptimalan hyperparameter, atau penggunaan teknik seperti regularisasi atau ensemble learning.

6. Penggunaan Model

Setelah model dianggap cukup, dapat digunakan untuk membuat prediksi tentang informasi baru yang sebelumnya tidak terlihat. Input baru diberikan ke model dan model menghasilkan prediksi berdasarkan informasi yang dipelajarinya selama proses pelatihan.

Ensemble learning adalah teknik pembelajaran mesin yang menggunakan beberapa model atau algoritme untuk menghasilkan prediksi yang lebih akurat daripada model atau algoritme yang dibangun secara mandiri. Pembelajaran *ensemble* dapat digunakan untuk meningkatkan kinerja model dan mengurangi *overfitting*. Ada beberapa jenis pembelajaran ensemble antara lain *bumping*, *boosting*, *stacking*, *voting*, dan *random forest*.

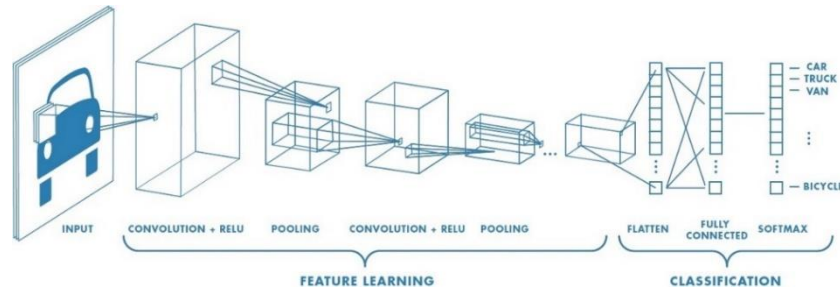
Supervised learning sangat berguna dalam masalah klasifikasi, di mana *output* yang diinginkan adalah dalam bentuk kategori diskrit, dan juga dalam masalah regresi, di mana *output* yang diinginkan adalah dalam bentuk nilai kontinu. Dengan menggunakan data training yang teridentifikasi, model dapat mempelajari pola tentang input dan *output* terkait dan kemudian menggeneralisasikannya untuk membuat prediksi dari data baru. Untuk *supervised learning*, ruang berdimensi rendah yang disesuaikan dengan informasi kelas sering kali lebih unggul daripada ruang dimensi rendah intrinsik. Hal ini perlu dicatat bahwa pelestarian lingkungan dalam bermacam-macam pembelajaran membutuhkan sampel yang padat, dan ini memang merupakan penghalang utama dalam situasi *high-dimensional*. Oleh karena itu, dalam praktiknya kinerja pengurangan dimensi dari pembelajaran

berjenis sering tidak sebaik yang diharapkan. Namun demikian, gagasan tentang pelestarian lingkungan telah secara signifikan mempengaruhi yang lain cabang penelitian pembelajaran mesin lainnya, seperti yang terkenal asumsi manifold yang terkenal dan regularisasi manifold dalam semisupervised pembelajaran menyediakan kerangka kerja terpadu untuk pengurangan dimensi dari sudut pandang penyematan graf [17].

2.2.5 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu metode *Deep Learning* yang digunakan untuk mendeteksi dan mengenali sebuah objek pada citra digital. *Convolutional networks* adalah contoh pertama dari arsitektur dalam yang telah berhasil mencapai generalisasi yang baik pada visual input. Mereka adalah metode yang paling dikenal untuk *digit recognition*. Kemampuan CNN di klaim sebagai model terbaik untuk memecahkan permasalahan *Object Detection* dan *Object Recognition*. Pada tahun 2012, penelitian tentang CNN dapat melakukan pengenalan citra digital dengan akurasi yang menyaingi manusia pada dataset tertentu [18].

CNN merupakan pengembangan dari ANN konvensional. Jika pada ANN terdiri dari tiga hingga empat layer, CNN merupakan *network* dengan puluhan hingga ratusan layer. CNN merupakan metode deep learning yang lebih baik penerapannya untuk data *supervised* [17]. CNN memiliki kemampuan *feature learning* lebih baik dibandingkan dengan metode deep learning lain. Selain itu, CNN memiliki kemampuan yang baik untuk data augmentasi diantaranya rotasi dan translasi citra. CNN memproses citra melalui *network* layer dan menghasilkan output pada kelas tertentu. Setiap layer melakukan pembelajaran. Output dari setiap layer digunakan sebagai input untuk layer selanjutnya. Layer pada CNN dibagi menjadi dua bagian sesuai dengan fungsinya yaitu layer ekstraksi fitur dan layer klasifikasi. Layer ekstraksi fitur merupakan layer yang berfungsi untuk melakukan ekstraksi fitur dari citra input, sedangkan layer klasifikasi melakukan klasifikasi citra berdasarkan fitur - fitur yang telah dihasilkan pada layer ekstraksi fitur. Layer ekstraksi fitur terdiri dari *convolutional* layer, *pooling* layer dan *Rectified Linear Unit* (RELU) seperti yang terlihat pada gambar 2.6 [19].



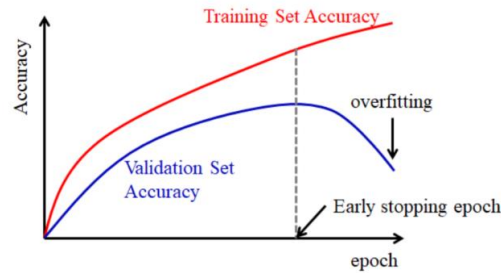
Gambar 2.6 CNN Architecture [20]

Pada awal *network*, layer menghasilkan fitur - fitur sederhana seperti warna, kecerahan (*brightness*) dan tepi (*edges*). Selanjutnya *network* akan menghasilkan fitur yang lebih kompleks. Optimasi CNN dilakukan secara menggunakan alur *backpropagation* yang terdiri dari *forward pass* dan *backward pass*.

Optimasi CNN dilakukan secara menggunakan alur *backpropagation* yang terdiri dari *forward pass* dan *backward pass*. Evaluasi model digunakan untuk mengukur performa model CNN yang sudah dirancang dan dilakukan proses training. *Underfit*, *just fit*, dan *overfit* adalah istilah yang digunakan untuk menggambarkan performa model *machine learning*. *Underfit* dapat terjadi ketika sebuah model terlalu sederhana dan tidak dapat menangkap tren yang mendasari data. Model yang kurang cocok memiliki bias yang tinggi dan varians yang rendah, serta berkinerja buruk pada data pelatihan dan pengujian.

Just fit terjadi ketika sebuah model telah menemukan keseimbangan yang tepat antara bias dan varians, dan model tersebut menggeneralisasi dengan baik pada data baru. Model yang pas adalah model ideal yang ingin kita capai. *Overfitting* terjadi ketika model terlalu kompleks dan mencoba memasukkan semua atau lebih titik data yang diperlukan ke dalam kumpulan data tertentu. Model yang *overfit* memiliki bias yang rendah dan varians yang tinggi, dan berkinerja baik pada data pelatihan tetapi buruk pada data pengujian. Untuk menghindari *underfitting* dan *overfitting*, kita perlu menemukan keseimbangan yang tepat antara bias dan varians. Model yang pas adalah model ideal yang ingin kita capai, karena model ini menggeneralisasi dengan baik pada data baru. Untuk mencapai model yang pas, kita dapat menggunakan teknik seperti validasi silang, regularisasi, dan ensembling.

Saat *forward pass*, CNN menghasilkan *error* atau *loss function*. *loss function* merupakan selisih antara prediksi dan target. *Loss function* diminimalkan menggunakan algoritma optimasi *gradient descent*. Prosedur untuk memperbaiki *loss function* menggunakan *backward pass*. Hal tersebut terjadi secara iteratif hingga tercapai salah satu dari dua kondisi yaitu *epoch* maksimal telah tercapai atau validasi data mengalami *overfitting*.



Gambar 2. 7 Kondisi *early stop* pada saat terjadi *overfitting*[17]

Pada Gambar 2.7 menunjukkan proses terjadinya *early stop* karena *overfitting*. *Overfitting* terjadi pada saat akurasi pada training data meningkat dan akurasi validation data mengalami penurunan. Pada kondisi ini proses training akan dihentikan lebih awal dari *epoch* yang telah ditetapkan [17].

a. Convolutional Layer

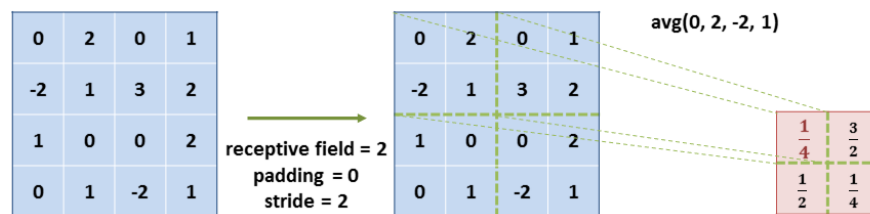
Convolutional layer merupakan layer utama dari CNN. CNN dapat memiliki lebih dari satu *convolution layer* yang menghasilkan fitur-fitur sederhana pada awal network hingga fitur-fitur kompleks pada *deeper convolutional layer*. Pada *conventional layer* terdapat operasi konvolusi antara citra input dan filter. Terlihat dari gambar 2.8 operasi konvolusi merupakan penjumlahan hasil perkalian antara matriks pada citra input dengan matriks pada filter.

| Input Citra | Filter | Hasil Konvolusi |
|-------------|--------|-----------------|
| 4 3 3 2 2 1 | 1 0 1 | 17 |
| 2 3 4 2 3 4 | 0 1 0 | |
| 5 2 2 1 3 5 | 1 0 1 | |
| 1 2 3 2 2 1 | | |
| 2 1 3 4 2 4 | | |
| 2 3 1 1 1 2 | | |

Gambar 2. 8 Konvolusi antara matriks citra dan filter untuk baris 1 kolom[17]

b. Pooling Layer

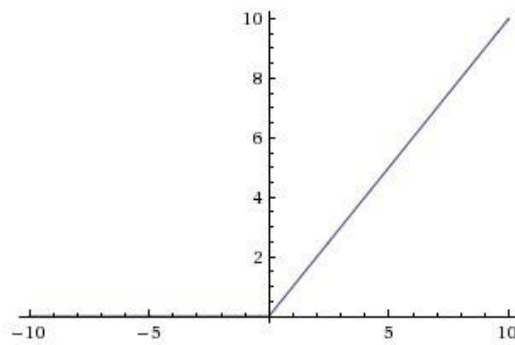
Pooling Layer adalah layer yang menggunakan fitur sebagai *Feature Map* sebagai masukan dan memprosesnya menggunakan berbagai macam operasi statistik berdasarkan nilai piksel terdekat. Pada model CNN, lapisan *Pooling* biasanya disisipkan secara teratur setelah beberapa lapisan konvolusi. Lapisan *Pooling* yang dimasukkan diantara lapisan konvolusi secara berturut-turut dalam arsitektur model CNN dapat secara progresif mengurangi ukuran volume *output* pada *Feature Map*, sehingga mengurangi jumlah parameter dan perhitungan di jaringan, dan untuk mengendalikan *Overfitting*. Terlihat dari gambar 2.9 lapisan *Pooling* bekerja di setiap tumpukan *Feature Map* dan mengurangi ukurannya. Bentuk lapisan Pooling yang paling umum adalah dengan menggunakan filter berukuran 2x2 yang diaplikasikan dengan langkah sebanyak 2 dan kemudian beroperasi pada setiap irisan dari input. Bentuk seperti ini akan mengurangi *Feature Map* hingga 75% dari ukuran aslinya [10].



Gambar 2. 9 Contoh Operasi Maxpooling [10]

c. Aktivasi ReLu

Aktivasi ReLU (*Rectified Linear Unit*) merupakan lapisan aktivasi pada model CNN yang mengaplikasikan fungsi $f(x)=\max(0,x)$ yang artinya fungsi ini melakukan thresholding dengan nilai nol terhadap nilai piksel pada input citra. Aktivasi ini membuat seluruh nilai piksel yang bernilai kurang dari nol pada suatu citra akan dijadikan 0 seperti terlihat pada gambar 2.10.



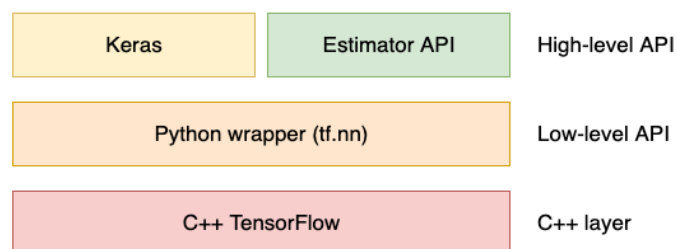
Gambar 2. 10 Aktivasi ReLU [10]

2.2.6 Tensorflow

TensorFlow menggunakan library yang dikembangkan oleh NVIDIA yang disebut CUDA. *Tensorflow* adalah *library open source* untuk komputasi numerik dan *machine learning* skala besar. *Tensorflow* menggabungkan banyak model, algoritma *machine learning* dan algoritma *deep learning* (jaringan syaraf). *Tensorflow* menggunakan *Python* untuk menyediakan *API front-end* untuk membangun aplikasi dengan *framework*, sekaligus menjalankan aplikasi tersebut dengan performa tinggi [21].

Tensorflow dapat melatih dan menjalankan jaringan syaraf tiruan untuk klasifikasi tulisan tangan, pengenalan gambar, penyematan kata, *recurrent neural network*, *sequence-to-secuence models* untuk terjemahan, pemrosesan *natural language*, dan simulasi berbasis PDE (*Partial Differential Equation*). Wrapper layer ini memungkinkan pengguna bekerja lebih cepat karena *Python* dianggap lebih mudah digunakan dari pada *C++* dan tidak memerlukan kompilasi [22].

Terlihat dari gambar 2.11 arsitektur *TensorFlow* memiliki beberapa tingkat abstraksi. Pertama-tama, mari kita perkenalkan layer paling bawah dan temukan jalan menuju layer paling atas:



Gambar 2. 11 Diagram arsitektur TensorFlow [22]

2.3 DROPOUT

Dropout adalah teknik regularisasi saraf yang digunakan untuk mengurangi *overfitting*. Penghapusan bekerja dengan menghapus beberapa *neuron* secara acak dan koneksi terkaitnya selama pelatihan model. Ini mencegah *neuron* dan koneksi terkaitnya menjadi terlalu bergantung satu sama lain dan memaksa model untuk mempelajari lebih banyak fitur umum. Berikut adalah beberapa poin penting dari kelalaian:

1. *Dropout* bekerja dengan menghapus beberapa neuron secara acak dan koneksi terkaitnya selama pelatihan model.
2. neuron dan koneksi terkaitnya dicegah agar tidak terlalu bergantung satu sama lain dan memaksa model untuk mempelajari lebih banyak fitur umum.
3. *Dropout* dapat diterapkan ke lapisan tersembunyi dan lapisan input jaringan saraf tiruan.
4. *Dropout* dapat digunakan bersamaan dengan teknik legalisasi lainnya seperti penurunan berat badan dan penghentian dini.

Dalam penggunaannya, *dropout* terbukti efektif dalam mengurangi *overfitting* dan meningkatkan performa model dalam berbagai tugas termasuk klasifikasi gambar dan pengenalan suara.

2.4 OPTIMASI ADAM

Optimalisasi Adam adalah algoritma optimasi *stochastic gradient descent* (SGD) yang biasa digunakan dalam pembelajaran mendalam. Berikut adalah beberapa poin utama untuk mengoptimalkan Adam:

1. Optimasi Adam menggabungkan keunggulan dari dua algoritma optimasi lainnya diantaranya *AdaGrad* dan *RMSProp*.
2. Optimalisasi Adam mempertahankan pembelajaran terpisah untuk setiap bobot dalam model dan memperbarui pembelajaran secara adaptif selama pelatihan.
3. Optimalisasi Adam menggunakan momentum untuk mempercepat konvergensi dan memperbaiki bias dalam estimasi momen gradien pertama dan kedua.

4. Pengoptimalan Adam efisien secara komputasi dan telah terbukti menyatu lebih cepat daripada algoritme pengoptimalan lainnya untuk banyak tugas pembelajaran mendalam.

Singkatnya, optimasi Adam adalah algoritma optimasi pembelajaran mendalam yang populer yang menyesuaikan pembelajaran sesuai dengan bobot masing-masing model, menggabungkan keunggulan *AdaGrad* dan *RMSProp*. *Algoritme* ini efisien secara komputasi dan telah terbukti konvergen lebih cepat daripada *algoritme* pengoptimalan lainnya untuk banyak tugas pembelajaran mendalam.