

BAB II

LANDASAN TEORI

2.1 Kedelai

Menurut (Ginting, E., & Tastra, I. K. 2016), kedelai merupakan sumber protein nabati yang relatif murah bagi sebagian besar masyarakat Indonesia, dibanding protein hewani. Kedelai dalam bentuk bungkil juga digunakan sebagai sumber protein bagi ternak. Konsumsi kedelai sebagai bahan pangan diperkirakan meningkat 2-3% dalam kurun waktu 1995-2010 dan 5-7% sebagai pakan ternak. Kadar protein kedelai lebih tinggi dari jenis kacang-kacangan lainnya dan bervariasi (34-48%), bergantung pada varietas. Protein biji menentukan gizi produk berbahan baku kedelai, seperti tahu, kecap, susu kedelai, tepung, dan protein isolat/konsentrat.

Standar mutu kedelai mempunyai peranan penting dalam pemasaran produksi. Pencapaian standar mutu kedelai yang sesuai dengan yang telah disepakati oleh pihak yang terlibat dalam sistem produksi, seperti tercermin dalam SNI 01-3922-1995. Hal ini dapat dijadikan acuan dalam mengoptimalkan kinerja komponen teknologi pascapanen yang berdampak langsung terhadap pencapaian standar mutu kedelai. Komponen standar mutu yang perlu dipertimbangkan untuk pengembangan standar mutu biji dan produk olahan kedelai adalah aspek kesehatan dan keamanan pangan.

Pemerintah melalui Dewan Standardisasi Nasional (DSN) telah menetapkan Standar Nasional Indonesia untuk mutu fisik biji kedelai (SNI 01-3922- 1995) yang masih berlaku sampai saat ini. Sebelumnya, digunakan standar mutu yang ditetapkan/disepakati secara bersama oleh Departemen Pertanian, Departemen

Koperasi, dan BULOG pada tahun 1988. Standar mutu biji kedelai, baik untuk jenis kuning, hitam, dan hijau maupun campuran ditetapkan dalam SNI 01-3922-1995 yang mengklasifikasikan mutu kedelai dalam empat tingkatan, yakni mutu I, II, III, dan IV.

Tabel 2.1. Klasifikasi Mutu Kedelai

Jenis Uji	Satuan	Persyaratan Mutu			
		I	II	III	IV
Kadar air (maksimum)	%	13	14	14	16
Butir belah (maksimum)	%	1	2	3	5
Butir rusak (maksimum)	%	1	2	3	5
Butir keriput (maksimum)	%	0	1	3	5
Kotoran (maksimum)	%	0	1	2	3

Syarat umum (kualitatif) meliputi bebas hama dan penyakit (kutu, ulat, telur, kepompong), bebas bau busuk, asam atau bau asing lainnya, bebas dari bahan kimia seperti insektisida dan fungisida, memiliki suhu normal, syarat khusus atau kuantitatif. Pengertian definisi masing-masing komponen mutu kuantitatif tersebut adalah sebagai berikut:

- Kadar air, adalah jumlah kandungan air dalam biji kedelai yang dinyatakan dalam persentase berat basah.
- Kedelai kuning, adalah kedelai yang kulit bijinya berwarna kuning, putih atau hijau dan bila dipotong melintang memperlihatkan warna kuning pada bidang irisan keping bijinya dan tidak tercampur lebih dari 10% kedelai jenis lain.

- c. Kedelai hitam, adalah kedelai yang kulit bijinya berwarna hitam dan tidak tercampur lebih dari 10% kedelai jenis lain
- d. Kedelai hijau, adalah kedelai yang kulit bijinya berwarna hijau dan bila dipotong melintang memperlihatkan warna hijau pada bidang irisan keping bijinya dan tidak tercampur lebih dari 10% kedelai jenis lain.
- e. Butir belah, adalah biji kedelai yang kulit bijinya terlepas dan keping-keping bijinya terlepas atau tergeser.
- f. Butir rusak, adalah biji kedelai yang berlubang bekas serangan hama, pecah karena mekanis, biologis, fisis, dan enzimatik seperti berkecambah, busuk, timbul bau yang tidak disukai, dan berubah warna ataupun bentuk.
- g. Kotoran, adalah benda asing seperti pasir, tanah, potongan sisa batang, daun, kulit polong, biji-bijian lain yang bukan kedelai.
- h. Butir keriput, adalah biji kedelai yang berubah bentuknya dan keriput, termasuk biji sangat muda atau tidak sempurna pertumbuhannya.

Sifat fisik dan kimia biji kedelai juga turut menentukan pemanfaatan kedelai sebagai bahan pangan. Berdasarkan warna bijinya, kedelai dapat digolongkan menjadi kedelai kuning, hijau, dan hitam. Warna biji ini sangat mempengaruhi penggunaan kedelai sebagai bahan makanan, misalnya untuk produk kecap lebih disukai kedelai hitam, biji kuning sampai hijau untuk tahu, dan biji kuning untuk tempe.

2.2 Pengolahan Citra

Menurut (Shafira, T. 2018), pengolahan citra merupakan pemrosesan citra atau bisa disebut dengan *image processing*, lebih khususnya dengan menggunakan komputer menjadi citra dengan kualitas yang lebih baik. Ini berarti bahwa pengolahan citra merupakan kegiatan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin. Langkah-langkah dalam pengolahan citra digital antara lain:

- a. Akuisisi citra, yaitu merupakan proses untuk menangkap atau mengambil citra yang dibutuhkan dengan menggunakan sensor pencitraan berupa kamera, *scanner*, dan lainnya.
- b. *Preprocessing* citra, yaitu dilakukan untuk menyiapkan citra untuk diproses lebih lanjut, bisa berupa ekstraksi fitur maupun kebutuhan klasifikasi. Teknik *preprocessing* citra yang umum digunakan antara lain adalah *cropping* dan perubahan ukuran citra.
- c. Segmentasi, yaitu membagi sebuah citra menjadi beberapa bagian penyusunnya. Proses segmentasi dilakukan sampai objek yang diinginkan dalam suatu aplikasi terpisah dari objek aslinya. Tingkat kesuksesan dari sebuah sistem pengenalan citra juga dipengaruhi oleh segmentasi yang akurat.
- d. Representasi dan deskripsi. Representasi yaitu menyatakan data piksel ke dalam bentuk data yang mampu diolah oleh komputer. Sementara proses deksripsi dilakukan untuk mengekstrak atribut yang menghasilkan informasi kuantitatif yang diinginkan atau yang merupakan fitur untuk membedakan citra antar kelas.
- e. Pengenalan, yaitu proses pemberian label pada objek sesuai dengan fitur yang dimiliki objek.

2.3 Citra Digital

Menurut (Nurhikmat, T. 2018), citra digital adalah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi yang *continue* menjadi gambar diskrit melalui proses sampling. Proses perubahan citra menjadi citra digital dinamakan dengan digitasi. Digitasi merupakan proses mengubah sebuah gambar, teks, atau suara dari benda yang dapat dilihat ke dalam data elektronik dan dapat disimpan serta diproses untuk keperluan lainnya. Dalam konteks yang lebih luas, pengolahan citra digital lebih mengacu pada pemrosesan setiap dua data dimensi. Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari tentang bagaimana teknik pengolahan sebuah citra. Citra yang dimaksud disini adalah sebuah gambar diam (foto) maupun gambar bergerak (Video). Sedangkan digital disini mempunyai maksud penting bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer. Dalam citra digital terdapat sebuah larik (*array*) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu.

Dalam sebuah komputer, citra digital dipetakan menjadi bentuk grid dan elemen piksel berbentuk matriks 2 dimensi. Setiap piksel-piksel tersebut memiliki angka yang mempresentasikan *channel* warna. Angka pada setiap piksel disimpan secara berurutan oleh sebuah komputer dan sering dikurangi untuk keperluan kompresi maupun pengolahan tertentu. Sebuah citra digital dapat mewakili oleh sebuah matriks yang terdiri dari M kolom N baris, dimana perpotongan antara kolom dan baris disebut piksel. ($\text{pixel} = \text{picture element}$), yaitu elemen terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai

yang terdapat pada koordinat (x,y) adalah $f(x,y)$, yaitu besar intensitas atau warna dari piksel di titik itu. Oleh karena itu, citra dapat dituliskan kedalam sebuah matriks :

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

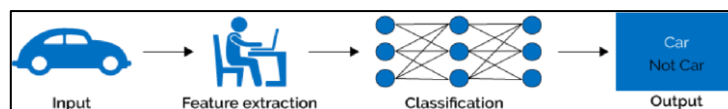
Gambar 2.1 Rumus Matriks

(Sumber: Nurhikmat, T. 2018)

2.4 Machine Learning

Menurut (Kusumaningrum, T. F., 2018), *machine learning* atau pembelajaran mesin adalah pendekatan dalam *Artificial Intelligence* yang banyak digunakan untuk menggantikan atau menirukan perilaku manusia untuk menyelesaikan masalah atau melakukan otomatisasi. Sesuai namanya, *machine learning* mencoba menirukan bagaimana proses manusia atau makhluk cerdas belajar dan menggeneralisasi. Setidaknya ada dua aplikasi utama dalam *machine learning* yaitu, klasifikasi dan prediksi.

Machine learning membantu menangani dan memprediksi data yang sangat besar dengan cara merepresentasikan data-data tersebut dengan algoritma pembelajaran. *Machine learning* dapat membantu komputer memprogram diri mereka sendiri. Jika pemrograman adalah pekerjaan untuk membuat otomatis, maka *machine learning* mengotomatisasi proses otomatis. Pada dasarnya *machine learning* membiarkan data melakukan pekerjaan.



Gambar 2.2. Machine Learning

(Sumber: Kusumaningrum, T. F., 2018)

Ciri khas dari *machine learning* adalah adanya proses pelatihan, pembelajaran atau *training*. Oleh karena itu, agar bisa mengaplikasikan teknik-teknik *machine learning* maka dibutuhkan data. Tanpa data maka algoritma *machine learning* tidak dapat bekerja. Data yang digunakan biasanya dibagi 32 menjadi dua, yaitu data *training* dan data *testing*. Data *training* digunakan untuk melatih algoritma, sedangkan data *testing* digunakan untuk mengetahui performa algoritma yang telah dilatih sebelumnya ketika menemukan data baru yang belum pernah dilihat.

Klasifikasi adalah metode dalam *machine learning* yang digunakan oleh mesin untuk memilah atau mengklasifikasikan objek berdasarkan ciri tertentu sebagaimana manusia mencoba membedakan benda satu dengan yang lain, sedangkan prediksi atau regresi digunakan oleh mesin untuk menerka keluaran dari suatu data masukan berdasarkan data yang sudah dipelajari dalam *training*. Ada empat jenis cara pembelajaran pada *machine learning*, yaitu:

1. *Supervised Learning*, data pembelajaran mencakup keluaran yang sudah ditentukan.
2. *Unsupervised Learning*, data pembelajaran tidak mencakup keluaran yang ditentukan.
3. *Semi-supervised Learning*, data pembelajaran mencakup beberapa keluaran yang ditentukan.
4. *Reinforcement Learning*, pemberian hadiah dari setiap serangkaian tindakan yang dilakukan.

2.5 *Artificial Intelligence* (Kecerdasan Buatan)

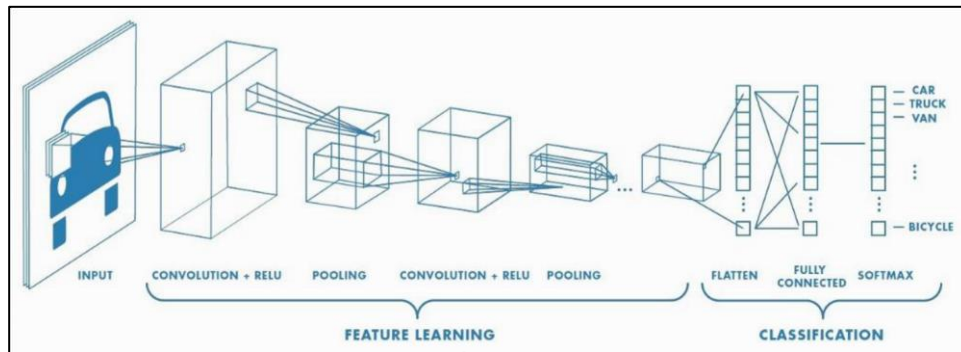
Menurut (Siti Rahmah, 2020), *artificial intelligence* atau kecerdasan buatan adalah suatu rancangan kecerdasan yang ditambahkan kedalam suatu sistem yang dapat diatur dalam konteks ilmiah atau *Intelegensi Artifisial* yang didefinisikan sebagai kecerdasan ilmiah. Kecerdasan buatan dirancang dan diciptakan serta dimasukkan kedalam mesin (komputer) agar dapat melakukan pekerjaan dan bertindak seperti manusia dengan dibekali pengetahuan serta kemampuan menalar yang baik. AI ditujukan untuk mengetahui atau memodelkan proses berpikir seperti manusia dan mendesain mesin sehingga dapat menirukan perilaku manusia. Bagian utama yang paling penting dan dibutuhkan dalam AI adalah basis pengetahuan (*knowledge base*) serta motor inferensi (*inference engine*) dimana basis pengetahuan berisikan pemikiran, fakta beserta teori-teori sedangkan motor inferensi adalah sebuah kemampuan dalam melakukan penarikan kesimpulan dengan berdasarkan pengetahuan. *Artificial Intelligence* atau kecerdasan buatan adalah suatu rancangan kecerdasan yang ditambahkan kedalam suatu sistem yang dapat diatur dalam konteks ilmiah atau *Intelegensi Artifisial* yang didefinisikan sebagai kecerdasan ilmiah. Kecerdasan buatan dirancang dan diciptakan serta dimasukkan kedalam mesin (komputer) agar dapat melakukan pekerjaan dan bertindak seperti manusia dengan dibekali pengetahuan serta kemampuan menalar yang baik. AI ditujukan untuk mengetahui atau memodelkan proses berpikir seperti manusia dan mendesain mesin sehingga dapat menirukan perilaku manusia.

Bagian utama yang paling penting dan dibutuhkan dalam AI adalah basis pengetahuan (*knowledge base*) serta motor inferensi (*inference engine*) dimana basis

pengetahuan berisikan pemikiran, fakta beserta teori-teori sedangkan motor inferensi adalah sebuah kemampuan dalam melakukan penarikan kesimpulan dengan berdasarkan pengetahuan. AI merupakan sebuah proses berpikir manusia dan mempresentasikan proses tersebut kedalam mesin yang digunakan. AI memiliki kemampuan menyelesaikan sebuah masalah yang merupakan tolak ukur sebuah kecerdasan. mesin yang dapat menyelesaikan masalah secara rutin adalah sebuah mesin yang cerdas.

2.6 Convolutional Neural Network

Menurut (Felix, F., Faisal, S., Butarbutar, T. F., & Sirait, P., 2019), penemuan CNN (*Convolutional Neural Network*) pertama kali dilakukan oleh Hubel dan Wiesel mengenai *virtual cortex* pada indera penglihatan kucing. Secara teknis, CNN (*Convolutional Neural Network*) adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap, masukan (*input*) dan keluaran (*output*) . Secara teknis, CNN (*Convolutional Neural Network*) melatih dan menguji setiap gambar masukan melalui serangkaian proses, yaitu *convolutional layer* diikuti oleh *pooling* untuk mengekstraksi fitur dari gambar masukan berturut-turut. Setelah operasi *pooling*, citra di-*flatten* dan kemudian dimasukkan ke proses *fully connected-layer* untuk melaksanakan tugas pengklasifikasian. Arsitektur yang menggambarkan proses-proses yang terjadi pada metode CNN (*Convolutional Neural Network*).

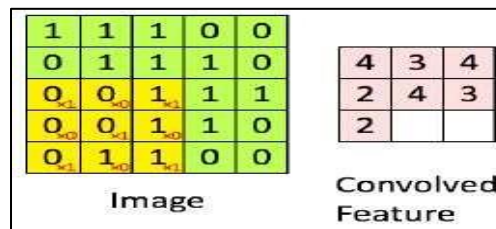


Gambar 2.3. Arsitektur *Convolutional Neural Network*

(Sumber: Felix, F., Faisal, S., Butarbutar, T. F., & Sirait, P., 2019)

2.6.1 *Convolutional Layer*

Menurut (Peryanto, A., Yudhana, A., & Umar, R. 2020), *convolution layer* adalah lapisan utama yang paling penting di metode CNN [10]. Hasil dari *convolution layer* adalah citra baru yang menunjukkan fitur dari citra masukan. Setiap citra yang menjadi masukan, menggunakan filter *convolution layer* dalam proses tersebut. *Convolutional layer* terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (pixel). Sebagai contoh, layer pertama pada *feature extraction layer* adalah *convolutional layer* dengan ukuran 6x6x3. Panjang 6 pixel, tinggi 6 pixel, dan tebal/jumlah 3 buah sesuai dengan channel dari gambar tersebut. Ketiga filter ini akan digeser keseluruhan bagian dari gambar. Setiap pergeseran akan dilakukan operasi “dot” antara *input* dan nilai dari filter tersebut sehingga menghasilkan sebuah *output* atau biasa disebut sebagai *activation map* atau *feature map*.



Gambar 2.4. Operasi Konvolusi

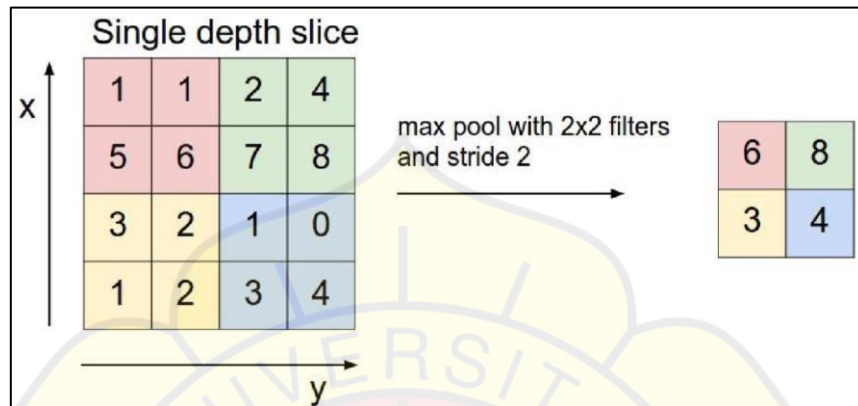
(Sumber: Peryanto, A., Yudhana, A., & Umar, R. 2020)

Kotak hijau secara keseluruhan adalah citra yang akan dilakukan konvolusi. Kernel bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat dari gambar di sebelah kanan. Tujuannya dilakukan konvolusi pada data citra yaitu untuk mengekstraksi fitur dari citra *input*. Konvolusi akan menghasilkan transformasi linear dari data *input* sesuai informasi spasial pada data. Bobot pada lapisan tersebut meng spesifikasi kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan *input* pada CNN.

2.6.2 Operasi *Pooling*

Menurut (Nadira, M. 2019), lapisan *pooling* (*pooling layer*) adalah lapisan fungsi untuk peta fitur sebagai masukan dan diolah dengan berbagai operasi statistik dengan berdasarkan nilai piksel terdekat. Pada model CNN, lapisan *pooling* berada tepat setelah lapisan konvolusi. Lapisan *pooling* terletak di antara lapisan konvolusi sesuai dengan susunan arsitektur model CNN, sehingga dapat mengurangi jumlah parameter dan perhitungan di jaringan, dan dapat mengendalikan *overfitting*. Lapisan *pooling* menghasilkan citra keluaran dengan dimensi yang lebih kecil dibandingkan dengan citra masukan, sehingga diperoleh citra keluaran dengan fitur yang penting

saja. Terdapat dua jenis *pooling* yang umum digunakan yaitu *average pooling* dan *max pooling*. *Average pooling* mengambil nilai rata-rata, sedangkan *max pooling* mengambil nilai terbesar.



Gambar 2.5. Operasi pada *Max Pooling*
(Sumber: Nadira, M. 2019)

2.6.3 Aktivasi ReLU

Menurut (Hardyanti, E. R., Jayanta, J., & Ernawati, I. 2020), Dengan konvolusi, operasi transformasi linier dilakukan semua perubahan *output* sebanding dengan perubahan *input*. Relu adalah transformasi yang menambah non-linearitas. Dengan non-linearitas yang ada akan meningkatkan efektivitas jaringan saraf. Relu menggantikan nilai negatif dengan nol. Itu hanya menyimpan nilai-nilai positif (dan membiarkannya tidak berubah). Fungsi ReLU adalah $f(x) = \max(0, x)$. Biasanya ini diterapkan untuk *output* dari beberapa fungsi lain, seperti produk vektor matriks. Salah satu cara ReLU meningkatkan jaringan saraf adalah dengan mempercepat pelatihan. Komputasi gradien sangat sederhana (0 atau 1 tergantung pada tanda x). Juga, langkah komputasi ReLU mudah: setiap elemen negatif diatur ke 0,0 - tanpa eksponensial, tidak ada operasi perkalian atau pembagian.

2.6.4 Fully-Connected Layer

Menurut (Shafira, T. 2018), hasil dari proses konvolusi menjadi *input* pada *fully-connected layer*. *Fully-connected layer* adalah lapisan dimana semua neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya seperti halnya jaringan saraf tiruan biasa. Setiap aktivasi dari lapisan sebelumnya perlu diubah menjadi data satu dimensi sebelum dapat dihubungkan ke semua neuron di *fully connected layer*. Lapisan ini biasanya digunakan pada metode *Multi-Layer Perceptron* (MLP) dan bertujuan untuk mengolah data sehingga bisa diklasifikasikan. Perbedaan antara *fully-connected layer* dengan lapisan konvolusi biasa adalah neuron di lapisan konvolusi terhubung hanya ke daerah tertentu pada *input*, sementara pada *fully-connected layer* memiliki neuron yang secara keseluruhan terhubung. Namun, kedua lapisan tersebut masih mengoperasikan produk dot, sehingga fungsinya tidak begitu berbeda.

2.6.5 Softmax Classifier

Menurut (Danukusumo, K. P. 2017), *Softmax classifier* merupakan bentuk lain dari algoritma regresi logistik yang dapat digunakan untuk melakukan klasifikasi objek yang memiliki lebih dari dua kelas. Klasifikasi yang biasa dilakukan oleh algoritma regresi logistik adalah klasifikasi kelas biner. Pada klasifikasi *softmax* bentuk persamaan yang muncul adalah sebagai berikut.

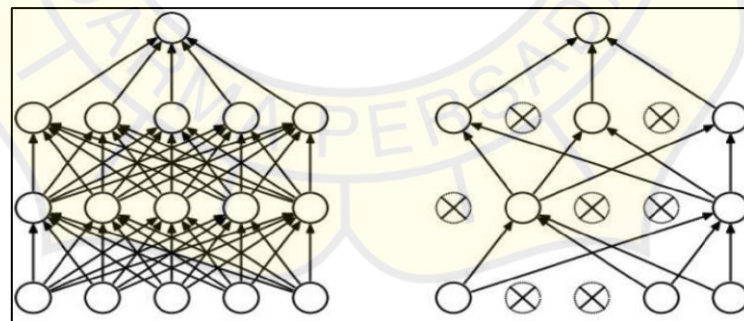
$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad 2.1$$

Pada persamaan 2.1, Notasi f_j menunjukkan hasil fungsi untuk setiap elemen ke- j pada vektor keluaran kelas. Argumen z adalah hipotesis yang diberikan oleh model pelatihan agar dapat diklasifikasi oleh fungsi *Softmax*.

Softmax juga memberikan hasil yang lebih intuitif dan juga memiliki interpretasi probabilistik yang lebih baik dibanding algoritma klasifikasi lainnya. *Softmax* memungkinkan kita untuk menghitung probabilitas untuk semua label. Dari label yang ada akan diambil sebuah vektor nilai bernilai riil dan merubahnya menjadi vektor dengan nilai antara nol dan satu yang bila semua dijumlah akan bernilai satu.

2.6.6 Dropout Regularization

Menurut (Shafira, T. 2018), *Dropout* adalah teknik regularisasi jaringan syaraf dimana beberapa neuron akan dipilih secara acak dan tidak dipakai selama pelatihan. Neuron-neuron ini dapat dibuang secara acak. Hal ini berarti bahwa kontribusi neuron yang dibuang akan diberhentikan sementara jaringan dan bobot baru juga tidak diterapkan pada neuron pada saat melakukan *backpropagation*.



(a) Jaringan Syaraf Biasa

(b) Jaringan Syaraf Setelah
dikenakan Teknik *Dropout*

Gambar 2.6. Contoh Implementasi *Dropout Regularization*

(Sumber: Shafira, T. 2018)

Pada gambar 2.6 diatas, jaringan syaraf (a) merupakan jaringan syaraf biasa dengan 2 lapisan tersembunyi. Sedangkan pada bagian (b) jaringan syaraf sudah diaplikasikan teknik regularisasi *dropout* dimana ada beberapa neuron aktivasi yang tidak dipakai lagi. Teknik ini sangat mudah diimplementasikan pada model CNN dan akan berdampak pada performa model dalam melatih serta mengurangi *overfitting*.

2.7 Alat dan Bahasa Pemrograman yang Digunakan

2.7.1 Web

Menurut (Istiqomah, Q. 2020), website adalah kumpulan halaman-halaman web yang diletakkan dalam satu tempat atau situs. Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau bergerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

2.7.2 PHP

Menurut (Alatas, F. B. 2021), PHP merupakan singkatan dari PHP *Hypertext Preprocessor*, yakni bahasa pemrograman yang berbasis web dan dijalankan pada sisi server. PHP sendiri merupakan singkatan dari *Hypertext Preprocessor* yang merupakan *server-side programming*, yakni bahasa pemrograman yang nantinya diproses pada sisi server. Fungsi PHP sendiri adalah mengolah data pada *database*. Kemudian data yang berasal dari website akan dimasukan kedalam *database*, diedit, dihapus, dan ditampilkan pada website.

2.7.3 CSS

Menurut (Josi, A. 2017), CSS singkatan dari *cascading style sheets*, yaitu skrip yang digunakan untuk mengatur desain website. Walaupun HTML mempunyai kemampuan untuk mengatur tampilan website, namun kemampuannya sangat terbatas. Fungsi CSS adalah memberikan pengaturan yang lebih lengkap agar struktur website yang dibuat dengan HTML terlihat lebih rapi dan indah.

2.7.4 HTML

Menurut (Alatas, F. B. 2021), HTML merupakan singkatan dari *Hypertext Markup Language*. Merupakan skrip yang berupa tag-tag untuk membuat dan mengatur struktur website. Contohnya seperti kita bisa memberitahu *browser* bahwa string atau tulisan tersebut harus ditampilkan sebagai header dengan huruf bold atau *Text* tersebut harus berada di tengah-tengah halaman tersebut.

2.7.5 MySQL

Menurut (Rifai, R. R. 2017), *MySQL* adalah salah satu program yang dapat digunakan sebagai database dan merupakan salah satu *software* untuk database server yang banyak digunakan. *MySQL* bersifat *Open Source* dan menggunakan *SQL*. *MySQL* AB membuat *MySQL* tersedia sebagai perangkat lunak gratis dibawah lisensi *GNU General Public License (GPL)*, tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus yang bersifat khusus. Kekuatan *MySQL* tidak ditopang oleh sebuah komunitas, seperti *apache*, yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh pemilik masing masing, tetapi *MySQL* didukung penuh oleh perusahaan komersial yang profesional, yakni *MySQL* AB dari swedia. *MySQL* adalah *Relational Database Management System (RDBMS)*

yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk mengakses *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat *Closed Source* untuk komersial.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah konsep pengoprasian Database, terutama untuk pemulihan atau seleksi dan pemasukan data, yang memungkinkan pengoprasian data dikerjakan dengan mudah secara otomatis.

2.7.6 Database

Menurut (Josi, A. (2017), *Database* adalah sekumpulan data yang terorganisir untuk mendukung banyak aplikasi secara efisien dengan memusatkan data dan mengontrol data *redundant*. Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.

2.7.7 Bootstrap

Menurut (Christian, A., Hesinto, S., & Agustina, A. 2018), *Bootstrap* adalah paket aplikasi siap pakai untuk membuat *front-end* sebuah website. Bisa dikatakan, *Bootstrap* adalah template desain web dengan fitur plus. *Bootstrap* diciptakan untuk mempermudah proses desain web bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman. Cukup bermodalkan pengetahuan dasar mengenai HTML dan CSS, anda pun siap menggunakan *Bootstrap*.

2.7.8 JavaScript

Menurut (Novita, R. 2021), *JavaScript* merupakan bahasa pemrograman komputer yang dinamis. Biasanya sering digunakan pada web browser untuk

menciptakan halaman web yang menarik, interaktif, serta menerapkan berbagai fungsi pada halaman web. *JavaScript* adalah bahasa *scripting* kecil, ringan, berorientasi objek yang ditempelkan pada kode HTML dan di proses di sisi *client*. *JavaScript* digunakan dalam pembuatan website agar lebih interaktif dengan memberikan kemampuan tambahan terhadap HTML melalui eksekusi perintah di sisi *browser*. *JavaScript* dapat merespon perintah *user* dengan cepat dan menjadikan halaman web menjadi responsif. *JavaScript* memiliki struktur sederhana, kodenya dapat disisipkan pada dokumen HTML atau berdiri sebagai satu kesatuan aplikasi.

JavaScript ialah suatu bahasa *scripting* yang digunakan sebagai fungsionalitas dalam membuat suatu web”. Contoh penggunaan *JavaScript* adalah sebagai berikut :

```
<html>
<head>
<title> JavaScript Contoh 1 </title>
<script type="Text/JavaScript JavaScript pertama");
</script>
</head>
<body>
<br> <!--anda bisa menulis HTML dan PHP disini --><br>
</body>
</html>
```

2.7.9 Xampp

Menurut (Josi, A. 2017), Xampp adalah sebuah aplikasi yang dapat menjadikan komputer kita menjadi sebuah server. Kegunaan Xampp ini untuk membuat jaringan

local sendiri dalam artian kita dapat membuat website secara *offline* untuk masa coba-coba di komputer sendiri. Jadi fungsi dari Xampp server itu sendiri merupakan server website kita untuk cara memakainya. Disebut server karena dalam hal ini komputer yang akan kita pakai harus memberikan pelayanan untuk mengakseskan web, untuk itu komputer kita harus menjadi server. Dapat disimpulkan xampp adalah aplikasi *tools* untuk menyediakan paket lunak yang berisi konfigurasi Web Server, *Apache*, *PHP*, *MySQL* untuk membantu kita dalam proses pembuatan aplikasi web yang menyatu menjadi satu sehingga memudahkan kita dalam membuat program web.

2.7.10 Pemodelan UML

Menurut (Rifai, R. R. 2017), “*unified modeling language*” (UML) adalah adalah suatu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi. UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh *Booch*, *Object Modelling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Dengan UML, metode *Booch*, OMT dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambah dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam daripada metode lainnya.

2.7.10.1 Use Case Diagram

Diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam suatu sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.” (Yunahar, 2018). Berikut adalah simbol-simbol yang ada pada diagram *use case*:

Tabel 2.2. *Use Case Diagram*

Simbol	Deskripsi
<i>Use Case</i> 	Fungsional yang disediakan <i>system</i> sebagai unit-unit yang saling bertukar antar unit atau <i>actor</i> biasanya dinyatakan dengan menggunakan kata kerja diawal <i>frase</i> nama <i>use case</i> .
<i>Actor</i> 	Orang, proses, atau <i>system</i> lain yang berinteraksi dengan <i>system</i> informasi yang akan dibuat diluar <i>system</i> informasi yang akan dibuat itu sendiri, jadi walaupun <i>symbol</i> dari <i>actor</i> adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal <i>frase</i> nama aktor

Asosiasi 	Komunikasi antar actor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Extensi 	Relasi <i>use case</i> tambahkan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat terdiri sendiri walau tanpa <i>use case</i> tambahan
Generalisasi 	Hubungan generalisasi dan sepealisasi (umum-khusu) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya
Include 	Relasi <i>use case</i> tambahkan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini

2.7.10.2 Activity Diagram

Activity Diagram menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor,

jadi aktivitas dapat dilakukan oleh sistem” (Yunahar, 2018). Simbol-simbol yang digunakan dalam *activity diagram* sebagai berikut :

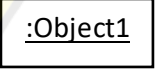
Tabel 2.3. Simbol *Activity Diagram*

Simbol	Deskripsi
	Simbol <i>start</i> untuk menyatakan awal dari suatu proses
	Simbol <i>stop</i> untuk menyatakan akhir dari suatu proses
	Simbol <i>desticion</i> digunakan untuk menyatakan kondisi dari suatu proses
	Simbol <i>action</i> menyatakan aksi yang dilakukan dalam suatu arsitektur sistem

2.7.10.3 *Sequence Diagram*

“*Sequence Diagram* adalah tool yang sangat populer dalam pengembangan sistem informasi secara *object-oriented* untuk menampilkan interaksi antar objek. Berdasarkan definisi tersebut, dapat disimpulkan bahwa *Sequence Diagram* adalah *tools* yang digunakan dalam pengembangan system” (Yunahar, 2018).

Tabel 2.4. *Tools Sequence Diagram*

Nama Komponen	Keterangan	Simbol
<i>Lifeline</i>	Mengindikasikan keberadaan sebuah objek dalam basis waktu. Notasi untuk lifeline adalah garis putus-putus vertikal yang ditarik dari sebuah objek	
<i>Activation</i>	Dinotasikan sebagai sebuah kotak segi empat digambar pada sebuah <i>lifeline</i> mengidentifikasi sebuah objek yang akan melakukan sebuah aksi	
<i>Message</i>	Digambarkan dengan anak panah horizontal antara <i>activation</i> . <i>Message</i> mengindikasikan komunikasi antara objek-objek	
<i>Object</i>	Merupakan <i>instance</i> dari sebuah class dan dituliskan tersusun secara horizontal	
<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi	

2.7.11 Python

Menurut (Pulungan, A. B., & Nafis, Z. 2021), Python adalah bahasa pemrograman interpretatif multiguna yang mudah untuk dibaca dan dipahami, python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks. Dengan kode yang simpel dan mudah diimplementasikan membuat Python sangat mudah dipelajari baik untuk pemula maupun untuk yang sudah menguasai bahasa pemrograman.

2.7.12 Keras

Menurut (Shafira, 2018), Keras adalah salah satu *package* yang digunakan dalam penyelesaian masalah tentang jaringan syaraf. Keras mendukung hampir semua dari model jaringan saraf dengan fokus mempercepat eksperimen dalam proses konvolusi dan *recurrent* pada *neural networks* maupun antara kombinasi keduanya. Untuk mengekspresikan perhitungan matematisnya dalam pembuatan model jaringan syaraf dengan menggunakan keras tidak perlu untuk menuliskan kode nya satu persatu. Hal tersebut dikarenakan keras telah menyediakan beberapa model dasar untuk CNN yang telah dioptimasi untuk mempermudah penelitian mengenai *deep learning*. Proses komputasi dengan menggunakan *package* keras dapat berjalan dengan baik menggunakan CPU ataupun GPU.

2.7.13 Tensorflow

Menurut (Pangestu, M. A., & Bunyamin, H. 2018), *TensorFlow* adalah *framework machine learning* yang bekerja dalam skala besar dan dalam *environment* yang *heterogeneous*. *TensorFlow* digunakan untuk melakukan eksperimen model

deep learning, melatih model pada dataset yang berukuran besar, dan membuatnya layak diproduksi.

Selain itu *TensorFlow* juga mendukung *training* dan *inference* berskala besar dengan menggunakan ratusan server yang menggunakan *Graphic Processing Unit* (GPU) untuk *training* secara efisien. Penelitian ini menggunakan *TensorFlow* sebagai *backend framework* untuk Keras.

2.7.14 Raspberry Pi

Menurut (Sulistiyanti, S. R., Setyawan, F. X., & Purwiyanti, S. 2019), *Raspberry Pi* adalah sebuah mikrokomputer atau *Single Board Computer* (SBC) berukuran sebesar kartu kredit, yang dikenal dengan sebutan RasPi yang dikembangkan oleh yayasan *Raspberry Pi* di Inggris (UK). *Raspberry Pi* mampu membuat dokumen, mengolah data dengan *spreadsheet*, bermain game, mendengar musik dan tentu saja coding seperti layaknya komputer desktop.