

## BAB II

### LANDASAN TEORI

#### 2.1. Tinjauan Pustaka

Penelitian ini dilakukan mengacu pada beberapa referensi dari penelitian terdahulu. Referensi diambil dari topik mengenai identifikasi kerusakan produk menggunakan objek citra. Berikut ini beberapa penelitian terdahulu yang relevan dengan penelitian ini.

Penelitian terdahulu tentang deteksi cacat produk dilakukan pada objek botol *wine* menggunakan *Gaussian filter* dan *Hough transform* yang memanfaatkan garis tepi citra. Deteksi tepi digunakan untuk menentukan ROI dari objek botol *wine*. Selanjutnya hasil ROI citra diklasifikasi menggunakan algoritma CNN. Jumlah data yang digunakan berjumlah 332 citra. Hasil klasifikasi diukur menggunakan akurasi yaitu sebesar 99,6% (Wang, dkk, 2019). Namun, ROI yang di hasilkan dari algoritma *Hough Transform* hanya sebatas untuk mencari tepian dan melakukan pemotongan pada objek citra. Dengan demikian, ROI yang didapatkan masih ada unsur *background* foto dari objek penelitian.

Penelitian sebelumnya tentang deteksi kerusakan pada objek ubin keramik dilakukan dengan pendekatan morfologi dan *fuzzy logic*. Luas kerusakan dihitung berdasarkan citra masukan yang telah disegmentasi dan diklasifikasikan ke dalam 4 kelas. Hasil dari penelitian berupa perhitungan luas kerusakan yang terjadi pada ubin keramik (Ahamad dan Rao, 2015).

Penelitian serupa mengenai deteksi kecacatan pada keramik dilakukan melalui pemrosesan citra digital menggunakan metode *Fuzzy C-Means* dan fitur morfologi. Cacat yang dideteksi adalah cacat goresan pada keramik. Hasil dari penelitian tersebut berupa pelabelan daerah keramik yang terdapat goresan dengan akurasi hingga 100% dengan *false alarm rate* sebesar 5,63% (Tan, dkk, 2020). Pada penelitian tersebut, kecacatan yang dideteksi hanya pada jenis goresan saja.

Penelitian mengenai deteksi kerusakan dilakukan pada objek *photo-voltaic* (PV) atau sel surya. Kerusakan modul PV secara otomatis dideteksi dari format citra

*electroluminescence* (EL). Jumlah modul PV yang menjadi sampel dalam penelitian berjumlah 44 citra. Selanjutnya, citra diaugmentasi dengan beberapa teknik rotasi, *flip*, *crop*, *contrast*, dan *blur* dan menghasilkan 2.624 citra. Citra EL dalam bentuk *grayscale* dan diproses menggunakan metode CNN. Hasil menunjukkan akurasi 93,02% dan dapat bekerja pada komputer non-GPU biasa. Satu citra dapat dideteksi dengan waktu 8,07 ms, dimana membuktikan bahwa sistem dapat memproses secara *real time* (Akram, dkk, 2020).

Penelitian tentang deteksi kerusakan juga dilakukan pada objek kain hasil dari produksi pabrik konveksi. Metode yang digunakan yaitu kombinasi dari *Long-short-term Memory* (LSTM) dan CNN. Terdapat tiga buah *dataset* yang digunakan pada penelitian ini yang berjumlah 500, 1000 dan 10500 data citra. Hasil yang ditunjukkan berupa akurasi pengujian, dimana pada *dataset* berjumlah 500 memberikan akurasi terbaik yaitu sebesar 99,4% (Zhao, dkk, 2020). Namun, tidak dijelaskan mengenai waktu proses deteksi pada sistem. Penggunaan LSTM yang dikombinasikan dengan CNN membuat arsitektur pemodelan menjadi lebih kompleks yang memungkinkan kebutuhan waktu proses yang lebih lama pula.

Pengembangan penelitian terkait deteksi kerusakan juga dilakukan pada citra *aluminium alloy*. Kerusakan dideteksi melalui proses deteksi tepi dan *background removal* pada citra objek. Ekstraksi fitur dilakukan dengan menggunakan 6 jenis spektrum dan menjadi *input* pada tahap klasifikasi. Klasifikasi dilakukan dengan membandingkan tiga metode klasifikasi yaitu *Random Forest*, *Radial Basis Function Neural Network*, dan *Backpropagation Neural Network*. Hasil klasifikasi diukur menggunakan akurasi dan *Random Forest* memberikan hasil akurasi terbaik sebesar 97,75% (Zhang, dkk, 2019).

Penelitian tentang deteksi cacat juga dilakukan pada citra produk permukaan hasil las. Deteksi dilakukan menggunakan *sobel edge detection* dan *contras limited adaptive histogram equalization* (CLAHE) untuk memperbaiki kualitas citra. Pendekatan morfologi digunakan untuk mendapatkan hasil cacat pada citra yang lebih jelas. Penelitian ini bertujuan untuk menemukan pada bagian mana terdapat cacat pada

permukaan hasil las sehingga teknik yang digunakan adalah menggunakan segmentasi citra (Ranjan, dkk, 2016).

Penelitian ini dilakukan dengan mengacu pada penelitian sebelumnya yang memiliki relevansi terhadap objek maupun metode. Pada penelitian ini, citra keramik *tableware* akan dideteksi kecacatannya menggunakan metode *morphological enhancement* dan CNN. Hasil yang diharapkan berupa informasi letak serta persentase kecacatan dan jenis kecacatan pada keramik *tableware*.

## **2.2. Dasar Teori**

### **2.2.1. Kualitas Produk**

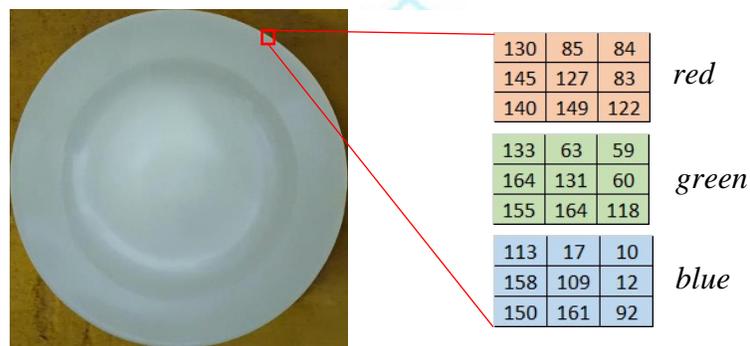
Kualitas produk secara fisik dapat dinilai dari keadaan dari produk tersebut, layak atau tidak. Produk yang memiliki cacat pada fisiknya akan mengurangi penilaian terhadap kepuasan konsumen (Khanna, dkk, 2017). Sehingga, perlu dilakukan pemeriksaan pada keadaan produk sebelum dilakukan pemasaran (Carrillo-Gutierrez, dkk, 2016). Misalkan pada produk keramik, proses pengecekan dilakukan dengan pemeriksaan terhadap fisik untuk dinilai kesempurnaan hasilnya apakah memiliki keretakan atau cacat yang lainnya. Kualitas produk dapat mempengaruhi reputasi dari citra perusahaan, sehingga pengembangan terhadap peningkatan kualitas produk harus dilakukan (Hall, dkk, 2016).

### **2.2.2. Pengolahan Citra Digital**

Pengolahan citra digital adalah teknik untuk memanipulasi citra yang dilakukan dengan memanfaatkan media komputer. Pengolahan citra digital dapat digunakan untuk membuat sistem dan aplikasi seperti pengenalan pola, *computer vision*, dan lain-lain. Pemanfaatan pengolahan citra digital untuk melakukan pengenalan pola dapat digunakan untuk klasifikasi suatu objek atau berupa sistem untuk mendeteksi berdasarkan *input* citra (Andono dan Sutojo, 2015).

Terdapat tiga jenis citra yang dapat diolah menggunakan komputer, yaitu citra warna (RGB), citra keabuan (*grayscale*), dan citra biner. Citra warna (RGB) memiliki

kedalaman piksel sebesar 8 bit yang berarti memiliki variasi warna sebanyak  $2^8$  pada masing-masing unsur warna *red*, *green* dan *blue*. Citra keabuan (*grayscale*) memiliki kedalaman piksel sebesar 8 bit yang berarti memiliki variasi warna sebanyak  $2^8$  dimana rentang intensitas cahayanya 0 hingga 255. Sedangkan, citra biner merupakan citra yang memiliki kedalaman piksel paling rendah yaitu 2 bit dimana rentang intensitas cahayanya 0 dan 1 (Andono dan Sutojo, 2015). Gambar 2.1 menunjukkan representasi citra RGB beserta piksel penyusunannya.



Gambar 2.1 Representasi Citra RGB

Pada Gambar 2.1, setiap piksel pada citra RGB, memiliki intensitas warna yang merupakan kombinasi dari tiga nilai intensitas pada kanal *red*, *green*, dan *blue*. Masing-masing kanal memiliki rentang nilai 0 hingga 255. Citra *grayscale* bisa didapatkan berdasarkan nilai-nilai pada kanal RGB. Perubahan citra RGB menjadi *grayscale* dapat dilakukan dengan menggunakan rumus (2.1) (Andono dan Sutojo, 2015).

$$p(x, y) = \frac{R(x, y) + G(x, y) + B(x, y)}{3} \quad (2.1)$$

Dimana,

$p(x, y)$  adalah nilai piksel pada kanal *grayscale*.

$R(x, y)$  adalah nilai piksel pada kanal *red*.

$G(x, y)$  adalah nilai piksel pada kanal *green*.

$B(x, y)$  adalah nilai piksel pada kanal *blue*.

Citra biner bisa didapatkan berdasarkan nilai-nilai berdasarkan nilai kanal *grayscale*. Perubahan citra *grayscale* menjadi citra biner dapat dilakukan

memanfaatkan nilai ambang (*threshold*) untuk menentukan suatu piksel termasuk kedalam nilai 0 atau 1. Rumus (2.2) merupakan perhitungan perubahan citra grayscale menjadi citra biner (Adi, dkk, 2018).

$$g(x,y) = \begin{cases} 1, & \text{jika } p(x,y) \geq T \\ 0, & \text{jika } p(x,y) < T \end{cases} \quad (2.2)$$

Dimana,

$g(x,y)$  adalah nilai piksel pada kanal *biner*.

$p(x,y)$  adalah nilai piksel pada kanal *grayscale*.

$T$  adalah nilai ambang (*threshold*).

Secara umum, terdapat lima tahapan dalam pengolahan citra (Andono dan Sutojo, 2015), yaitu:

1. Akuisisi Citra

Akuisisi citra adalah metode atau cara untuk mendapatkan data citra. Alat yang biasa dimanfaatkan untuk akuisisi citra adalah kamera, dimana hasil akuisisi dapat berupa gambar atau video.

2. *Preprocessing*

*Preprocessing* adalah metode yang dilakukan untuk memanipulasi citra asli agar sesuai dengan hasil yang diharapkan. *Preprocessing* dilakukan berupa perbaikan kualitas citra seperti peningkatan peningkatkan kualitas citra, menghilangkan *noise*, transformasi citra, dan lain lain.

3. Segmentasi

Segmentasi digunakan untuk mendapatkan informasi penting yang terdapat dalam citra dengan membagi citra menjadi beberapa bagian. Informasi yang didapatkan nantinya akan dapat dimanfaatkan pada tahap pengenalan pola.

4. Representasi dan deskripsi

Representasi dan deskripsi merupakan kumpulan ciri yang dibaca oleh komputer dalam bentuk angka yang mencerminkan keunikan dari citra. Pada tahap ini terdapat fitur ekstraksi dan fitur seleksi. Fitur ekstraksi merupakan kumpulan angka yang mencerminkan ciri pada citra. Fitur seleksi dapat

dimanfaatkan sebagai selektor untuk informasi citra yang penting dan memiliki pola yang unik.

#### 5. Pengenalan dan Interpretasi

Hasil dari fitur ekstraksi akan menjadi acuan pada proses pengenalan pola pada citra. Pengenalan pola dan interpretasi dapat diterapkan untuk melakukan klasifikasi, klustering, dan segmentasi sesuai dengan kebutuhan pengenalan pola.

#### **2.2.3. Morphological Enhancement**

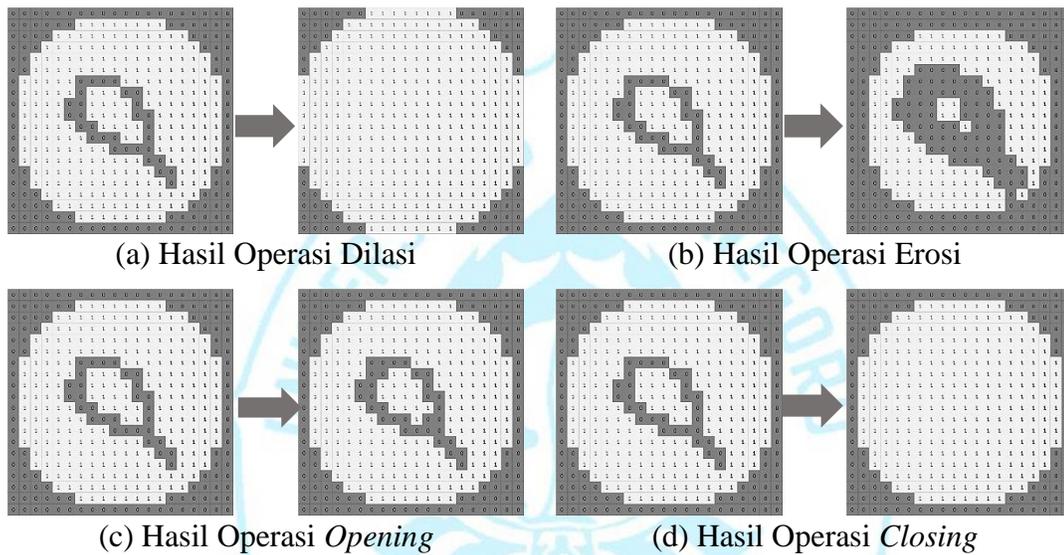
Segmentasi citra (Andono dan Sutojo, 2015) merupakan teknik untuk melakukan pembagian pada citra berdasarkan intensitas citra. Pembagian dilakukan untuk memisahkan antara objek utama dengan *background* dari citra tersebut. Terdapat beberapa jenis metode dalam segmentasi citra, diantaranya deteksi tepi, deteksi bentuk dan morfologi (Yang, dkk, 2021).

Segmentasi citra menggunakan deteksi tepi dilakukan dengan mengidentifikasi berdasarkan tepi objek. Tepi objek dapat ditemukan berdasarkan piksel nilai keabuan yang berbeda cukup besar dari piksel tetangganya. Sehingga, ketika terjadi tumpang tindih antara piksel yang saling bertetangga, maka akan ada titik tepi yang muncul. Morfologi merupakan teknik segmentasi yang ditentukan dengan mengikuti bentuk citra dimana tingkat keabuan yang hampir sama antara piksel citra yang saling bertetangga akan menjadi blok warna yang sama.

Pendekatan morfologi dikembangkan dari bentuk citra (Batchelor dan Waltz, 2012). Pendekatan morfologi memanfaatkan intensitas keabuan pada citra pada piksel yang saling bertetangga untuk menjadi blok warna yang sama. *Morphological enhancement* dapat dilakukan pada format citra *grayscale* maupun biner.

*Morphological enhancement* memiliki beberapa operasi umum yang digunakan untuk melakukan penambahan atau pengurangan pada ketebalan piksel pada citra yaitu operasi dilasi, erosi, *opening* dan *closing* (Ranjan, dkk, 2016). Hasil dari *morphological enhancement* yaitu berupa citra biner (Setyawan, dkk, 2018). Dilasi merupakan operasi morfologi yang dilakukan dengan menambah piksel pada batas objek citra (Adi dkk,

2018). Hasil dari operasi dilasi akan membuat citra menjadi lebih tebal dari bentuk sebenarnya. Erosi adalah operasi morfologi yang dilakukan dengan mengurangi piksel pada batas objek citra. Berbanding terbalik dengan operasi dilasi, hasil pada operasi erosi akan membuat citra terlihat lebih tipis daripada bentuk sebenarnya. Gambar 2.2(a), 2.2(b), 2.2(c) dan 2.2(d) menunjukkan secara lebih jelas perbedaan antara operasi dilasi, erosi, *opening* dan *closing*.



Gambar 2.2 (a) Hasil Operasi Dilasi; (b) Hasil Operasi Erosi

Berdasarkan Gambar 2.2(a), operasi dilasi melakukan perbesaran pada segmen objek (citra biner) dengan menambah lapisan di sekeliling objek. Secara rinci, terjadi perubahan pada titik latar dengan intensitas keabuan 0 yang bertetangga dengan titik objek yang memiliki intensitas keabuan 1 akan berubah menjadi menjadi titik objek dengan intensitas keabuan 1 juga. Gambar 2.2(b) menunjukkan operasi erosi yang merupakan kebalikan dari dilasi. Secara rinci, titik objek yang memiliki tingkat keabuan 1 yang bertetangga dengan titik latar dengan intensitas keabuan 0 akan berubah menjadi titik latar dengan intensitas keabuan 0 juga. Gambar 2.2(c) menunjukkan operasi *opening* yang mengombinasikan antara operasi erosi dan dilasi yang dilakukan secara berurutan. Gambar 2.2(d) menunjukkan operasi *closing* yang mengombinasikan antara operasi dilasi dan erosi yang dilakukan secara berurutan.

#### 2.2.4. Analisis Pengukuran Bentuk Objek

Bentuk objek pada citra dapat diukur menggunakan sifat geometri. Pada pengukuran bentuk objek citra, terdapat beberapa parameter yang dapat dihitung, yaitu luas, keliling, *compactness*, *roundness* dan lain lain (Andono dan Sutiojo, 2015). Pengukuran bentuk objek dilakukan pada citra biner. Luas merupakan jumlah dari kumpulan piksel yang menyusun objek. Pada penelitian ini luasan objek di hitung untuk mendapatkan persentase kerusakan pada objek citra. Luas objek citra dapat di hitung menggunakan rumus (2.3) (Gusa, 2013).

$$Luas = \sum_{i=1}^m \sum_{j=1}^n h(i,j) \quad (2.3)$$

Dimana:

$m$  merupakan jumlah baris piksel citra.

$n$  merupakan jumlah kolom piksel citra.

$h(i,j)$  merupakan piksel objek dalam citra.

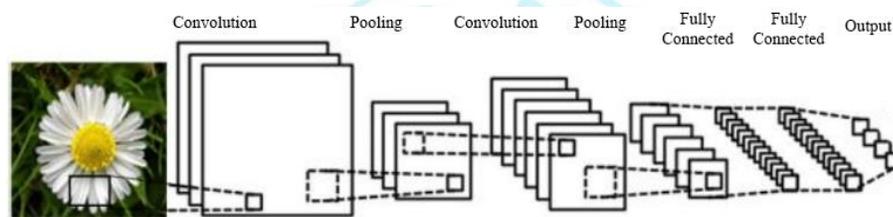
#### 2.2.5. Convolutional Neural Network

*Convolutional Neural Network* (CNN) merupakan salah satu algoritma dari pembelajaran mendalam yang memanfaatkan aritektur dari jaringan syaraf tiruan. CNN cocok digunakan untuk melakukan pemrosesan dan pengenalan pola pada citra (Wang, dkk, 2019). CNN merupakan pembelajaran mendalam hasil dari pengembangan *multi-layer perceptron* (MLP) (Bouwman dkk, 2019). Terdapat kesamaan antara cara kerja CNN dan MLP. Namun, perbedaannya CNN merepresentasikan *neuron* dalam bentuk dua dimensi, sedangkan *neuron* pada MLP direpresentasikan hanya satu dimensi (Kusumaningrum, dkk, 2019).

Di antara metode pembelajaran mendalam, manfaat utama jaringan CNN untuk klasifikasi citra meliputi: (1) CNN memiliki parameter yang relatif lebih sedikit sehingga pelatihan dapat dilakukan dengan cepat, pengurangan risiko *overfitting*, dan kebutuhan data latih yang lebih sedikit, (2) dapat mendeteksi fitur tertentu yang terletak di mana saja pada citra jika pernah mempelajari *filter* yang dapat mendeteksi fitur itu. Karena citra memiliki fitur yang berulang, CNN dapat digeneralisasi dengan relatif

lebih baik, (3) lapisan bawah CNN mendeteksi fitur tingkat bawah dan lapisan atas menggabungkan fitur ini menjadi fitur tingkat tinggi yang lebih besar (Geron, 2017).

Arsitektur CNN dapat bekerja dengan baik pada masukan citra dibandingkan dengan metode pembelajaran mendalam lainnya (Akram, dkk, 2019). Arsitektur CNN memiliki perbedaan pada tahap sebelum dilakukannya proses pelatihan. Bentuk awal *input* dari CNN adalah berupa matriks citra yang dikonversi menjadi lebih kecil dan memiliki nilai unik untuk selanjutnya dapat masuk ke proses pelatihan. Gambar 2.3 menunjukkan arsitektur CNN secara umum.



Gambar 2.3 Arsitektur CNN

Sumber : (Gogul dan Kumar, 2017)

Arsitektur CNN memiliki beberapa komponen (Habib dan Qureshi, 2020), diantaranya:

1. Lapisan Konvolusi

Lapisan konvolusi merupakan lapisan yang digunakan untuk memperkecil ukuran citra yang dilakukan melalui *filtering*. *Filter* akan membagi citra masukan menjadi beberapa kotak kecil (misal 3x3 atau 5x5) dan menggabungkan kisi-kisi kecil ini menjadi keluaran berupa *feature map*.

2. Fungsi Aktivasi

Fungsi aktivasi digunakan sebagai pengambilan keputusan pada jaringan syaraf dan membantu mempelajari fitur kompleks dari gambar masukan. Pada penelitian ini, fungsi aktivasi yang digunakan yaitu *Rectified Linear Units* (ReLU). Proses ReLUs diperlukan untuk menghindari kesalahan dalam pembacaan citra karena intensitas pada citra pada dasarnya berada pada angka 0 hingga 255. Fungsi ReLUs dirumuskan pada rumus (2.4).

$$F(x) = \begin{cases} x, & x > 0 \\ 0, & x < 0 \end{cases} \quad (2.4)$$

### 3. *Pooling*

*Pooling* merupakan lapisan yang berfungsi untuk mengurangi masalah sensitivitas pada keluaran *feature map*. Lapisan *pooling* mengambil nilai yang mewakili setiap *grid* untuk menyusun matriks citra yang telah direduksi. Tujuan dari *pooling* adalah untuk mendapatkan nilai paling unik yang mewakili citra *input*. Dua teknik *pooling* yang sering digunakan yaitu *max pooling* dan *average pooling*. *Max pooling* yaitu mengambil nilai tertinggi dari *grid* yang telah ditentukan. Sedangkan *average pooling* yaitu melakukan perhitungan rata-rata dari nilai-nilai yang berada pada *grid* yang telah ditentukan.

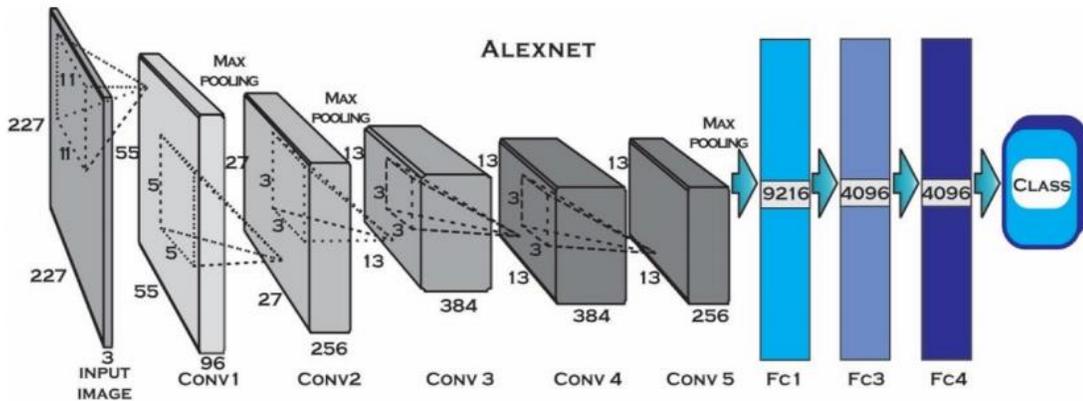
### 4. *Fully Connected Layer*

*Fully connected layer* bekerja dengan melakukan transformasi matriks ke dalam bentuk *flatten* / 1 dimensi. Hasil 1 dimensi tersebut digunakan sebagai *input* awal untuk dilatih menggunakan teknik klasifikasi (misal jaringan syaraf tiruan *backpropagation* atau *feed forward*) dalam mencari *output* klasifikasi.

#### 2.2.5.1. AlexNet

AlexNet merupakan salah satu jenis arsitektur CNN untuk *transfer learning* (Habib dan Qureshi, 2020). AlexNet pertama kali diterapkan pada tahun 2012. Arsitektur AlexNet menggunakan data yang telah diaugmentasi sebagai cara untuk meningkatkan efektivitas klasifikasi pada citra. Selain itu, *dropout layer* diterapkan pada arsitektur AlexNet untuk mengurangi *overfitting* pada citra (Llamas, dkk, 2017). Arsitektur AlexNet dapat dilihat pada Gambar 2.4.

SEKOLAH PASCASARJANA



Gambar 2.4 Arsitektur AlexNet

Sumber: (Ranjan, dkk, 2018)

Tabel 2.1 menunjukkan komposisi dari arsitektur AlexNet.

Tabel 2.1 Komposisi Arsitektur AlexNet

Lapisan	Aktivasi	Learnable
Masukan Citra	227 x 227 x 3	-
Konvolusi 1	55 x 55 x 96	<i>Weights</i> 11 x 11 x 3 x 96 <i>Bias</i> 1 x 1 x 96
ReLU 1	55 x 55 x 96	<i>Bias</i>
Normalisasi 1	55 x 55 x 96	-
Pooling 1	27 x 27 x 96	-
Konvolusi 2	27 x 27 x 256	<i>Weights</i> 5 x 5 x 48 x 128 x 2 <i>Bias</i> 1 x 1 x 128 x 2
ReLU 2	27 x 27 x 256	-
Normalisasi 2	27 x 27 x 256	-
Pooling 2	27 x 27 x 256	-
Konvolusi 3	13 x 13 x 384	<i>Weights</i> 3 x 3 x 256 x 384 <i>Bias</i> 1 x 1 x 384
ReLU 3	13 x 13 x 384	-
Konvolusi 4	13 x 13 x 384	<i>Weights</i> 3 x 3 x 192 x 192 x 2 <i>Bias</i> 1 x 1 x 192 x 2
ReLU 4	13 x 13 x 384	-
Konvolusi 5	13 x 13 x 256	<i>Weights</i> 3 x 3 x 192 x 128 x 2 <i>Bias</i> 1 x 1 x 128 x 2
ReLU 5	13 x 13 x 256	-
Pooling 5	6 x 6 x 256	-

<i>Fully Connected</i>	1 x 1 x 4096	<i>Weights</i> 4096 x 4096 <i>Bias</i> 4096 x 1
ReLU	1 x 1 x 4096	-
<i>Dropout</i>	1 x 1 x 4096	-
<i>Fully Connected</i>	1 x 1 x 4096	<i>Weights</i> 4096 x 4096 <i>Bias</i> 4096 x 1
ReLU	1 x 1 x 4096	-
<i>Dropout</i>	1 x 1 x 4096	-
<i>Fully Connected</i>	1 x 1 x 1000	<i>Weights</i> 1000 x 4096 <i>Bias</i> 1000 x 1
<i>Softmax</i>	1 x 1 x 1000	-
Keluaran Klasifikasi	1 x 1 x 1000	-

Arsitektur AlexNet terdiri dari 8 lapisan, dimana 5 lapisan merupakan lapisan konvolusi dan 3 lapisan lainnya merupakan lapisan *fully connected*. *Filter* 11x11 digunakan pada lapisan konvolusi pertama, 5x5 pada lapisan konvolusi kedua, dan 3x3 pada tiga lapisan konvolusi berikutnya. Setiap lapisan konvolusi menggunakan ReLU sebagai fungsi aktivasi. Tiga lapisan berikutnya yaitu berupa lapisan *fully connected* yang berperan untuk mengklasifikasikan data masukan. Hasil keluaran dari arsitektur AlexNet secara umum adalah 1000 kelas.

#### 2.2.6. *Optimizer*

*Optimizer* merupakan salah satu teknik untuk mengoptimalkan kinerja pembelajaran mendalam secara efektif dan efisien (Yang, dkk, 2020). Tujuan penggunaan *optimizer* yaitu untuk mengurangi waktu dan biaya pelatihan. Selain itu, penggunaan *optimizer* juga mampu meningkatkan kemampuan generalisasi model, yaitu untuk memastikan bahwa model DNN yang dipelajari dapat membuat prediksi yang akurat pada data pengujian (Yang, dkk, 2020). Beberapa metode *optimizer* yang umum digunakan yaitu *Stochastic Gradient Descent with Momentum* (SGDM) dan *Adaptive Moment Estimation* (ADAM).

##### 2.2.6.1. *Stochastic Gradient Descent with Momentum Optimizer*

SGDM *optimizer* merupakan salah satu *optimizer* hasil pengembangan dari *stochastic gradient descent* (SGD). SGDM memperkenalkan momentum orde pertama berdasarkan SGD (Jiang, dkk, 2020). Momentum tradisional yang dibuat oleh Polyak

pada tahun 1964 dianggap sebagai teknik klasik untuk percepatan penurunan gradien. Untuk memperbarui parameter bobot penurunan gradien dengan momentum tidak hanya mempertimbangkan gradien saat ini namun juga mengganti gradien saat ini dengan vektor kecepatan, rata-rata pergerakan eksponensial dari gradien saat ini dan sebelumnya untuk meminimalkan fungsi tujuan. Rumus (2.5) menunjukkan rumus dari SGDM (Habib dan Qureshi, 2020):

$$\begin{aligned}v_{t+1} &= \mu_t - \varepsilon \nabla z(\theta) \\ \theta_{t+1} &= \theta_t - v_{t+1}\end{aligned}\tag{2.5}$$

Dimana,  $\varepsilon$  mewakili laju pembelajaran dan lebih besar dari nol.  $\mu \in (0,1)$  merupakan koefisien momentum yang mengontrol kecepatan dan membantu mencegah overshooting sambil membuat langkah turun lebih cepat, dan  $\nabla z(\theta)$  mewakili gradien pada  $\theta_t$ . Polyak (1964) menunjukkan bahwa momentum klasik memerlukan jumlah iterasi yang lebih sedikit untuk mempercepat konvergensi ke minimum lokal seperti yang dipersyaratkan oleh penurunan paling curam tanpa penurunan akurasi. Keuntungan terbesar dari SGD adalah meskipun memperkenalkan perubahan yang sangat kecil ke SGD tetapi meningkatkan kecepatan pembelajarannya (Habib dan Qureshi, 2020).

#### **2.2.6.2. ADAM Optimizer**

ADAM *optimizer* merupakan salah satu jenis teknik optimasi dengan pembelajaran adaptif yang dirancang khusus untuk melakukan pelatihan pada model jaringan syaraf mendalam. ADAM merupakan kombinasi antara *optimizer* RMSprop dan SGDM. ADAM bekerja dengan lebih sumber daya sehingga prosesnya mampu selesai lebih cepat (Jiang, dkk, 2020). ADAM adalah algoritma optimasi orde pertama yang mampu menggantikan proses SGD. Pada optimizer ADAM, learning rate setiap parameter secara dinamis disesuaikan dengan menggunakan estimasi momen pertama dan kedua dari gradien. Koreksi bias juga ditambahkan, yang membuat parameter relatif stabil. Rumus iteratif ADAM optimizer (Jiang, dkk, 2020), ditunjukkan pada rumus (2.6).

$$\begin{aligned}
g &= (h_{\theta}(x^i - y^i))x^i \\
m_t &= \beta_1 m_{t-1} + (1 - \beta_1) * g \\
v_t &= \beta_2 v_{t-1} + (1 - \beta_2) * g^2 \\
m_t^{\sim} &= \frac{m_t}{1 - \beta_1^t} \\
v_t^{\sim} &= \frac{v_t}{1 - \beta_2^t} \\
\theta_j &= \theta_{j-1} - m_t^{\sim} * \frac{\alpha}{\sqrt{v_t^{\sim}} + \epsilon}
\end{aligned}
\tag{2.6}$$

Dimana,  $g$  adalah gradien yang dihitung,  $m_t$  menunjukkan momen pertama gradien  $g$ , yang juga merupakan ekspektasi gradien  $g$ ,  $v_t$  menunjukkan momen kedua gradien  $g$ , 1 mewakili koefisien redaman momen orde pertama, 2 mewakili redaman momen kedua koefisien, singkatan dari parameter yang perlu dipecahkan (atau diperbarui), dan  $m_t^{\sim}$  dan  $v_t^{\sim}$  menunjukkan koreksi offset  $m_t$  dan  $v_t$ .

### 2.2.7. Confusion Matrix

*Confusion matrix* merupakan teknik evaluasi visual yang digunakan dalam *machine learning* (Xu, dkk, 2020). *Confusion matrix* bekerja dengan cara membandingkan suatu hasil klasifikasi terhadap nilai sebenarnya (Kosasih, dkk, 2019). *Confusion matrix* memberikan informasi mengenai perbandingan hasil klasifikasi dari model yang telah di bentuk oleh sistem. Gambar 2.5 menunjukkan visualisasi perbandingan menggunakan *confusion matrix*.

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

Gambar 2.5 *Confusion Matrix*

*Confusion matrix* terdiri atas nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). TP menunjukkan suatu data yang bernilai positif yang berhasil terklasifikasi dalam kelas positif. TN menunjukkan suatu data yang bernilai negatif yang berhasil terklasifikasi dalam kelas negatif. FP menunjukkan suatu data yang bernilai negatif yang terklasifikasi salah ke dalam kelas positif. Sedangkan, FN menunjukkan suatu data yang bernilai positif yang terklasifikasi salah ke dalam kelas negatif.

Berdasarkan nilai TP, TN, FP dan FN, perhitungan tingkat kinerja model dapat diukur melalui nilai akurasi, presisi dan *recall*. Akurasi menunjukkan nilai keakuratan suatu hasil klasifikasi pada data yang memiliki nilai positif maupun negatif. Semakin tinggi nilai akurasi maka semakin baik model melakukan klasifikasi. Presisi merupakan nilai yang menunjukkan banyak data yang bernilai positif yang terklasifikasi ke dalam kelas negatif. Sedangkan, *recall* merupakan nilai yang menunjukkan banyak data yang bernilai positif mampu terklasifikasi ke dalam kelas positif. Pengukuran tingkat akurasi, presisi dan *recall* yang ditunjukkan pada formula (3), (4), dan (5).

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

$$Presisi = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

SEKOLAH PASCASARJANA