

BAB II
TINJAUAN PUSTAKA

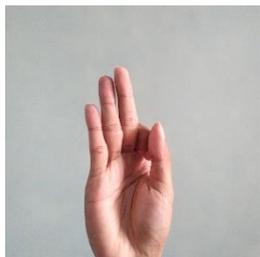
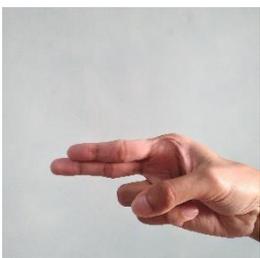
2.1 Sistem Bahasa Isyarat Indonesia (SIBI)

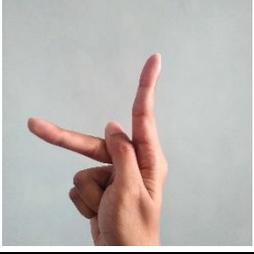
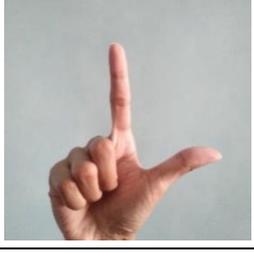
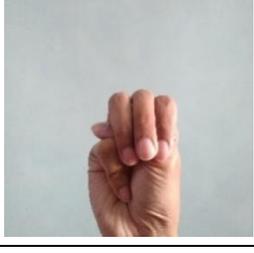
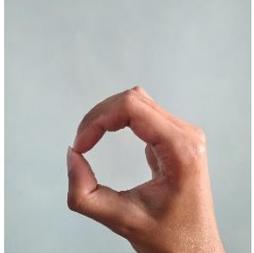
SIBI atau sistem isyarat bahasa isyarat Indonesia merupakan bahasa yang pemerintah sahkan sebagai bahasa isyarat resmi di sekolah. Bahasa isyarat SIBI diciptakan pertama kali oleh Alm. Anton Widyatmoko. Pada tahun 2001 pemerintah menerbitkan kamus SIBI untuk disebarluaskan ke SLB di seluruh Indonesia. Berbeda dengan BISINDO, SIBI disampaikan dengan satu tangan.

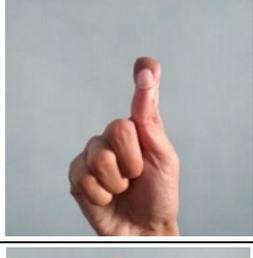
Contoh isyarat abjad menggunakan SIBI dapat dilihat pada bar 2.1. Bahasa Isyarat Indonesia ini berasal dari *Kaggle* yang di unggah oleh Alvin Bintang.

Tabel 2.1 Sistem Bahasa Isyarat Indonesia (SIBI) (Sistem Isyarat Bahasa Indonesia (SIBI), n.d.)

No	Abjad	Gambar Citra
1	A	
2	B	

No	Abjad	Gambar Citra
3	C	 A hand gesture where the thumb and index finger are curled together to form a 'C' shape, with the other three fingers extended.
4	D	 A hand gesture where the index finger is extended upwards, and the other four fingers are curled together.
5	E	 A hand gesture where the index finger is extended upwards, and the other four fingers are curled together.
6	F	 A hand gesture where the index, middle, and ring fingers are extended upwards, and the thumb and pinky are curled.
7	G	 A hand gesture where the index and middle fingers are extended upwards, and the other three fingers are curled.
8	H	 A hand gesture where the index and middle fingers are extended upwards, and the other three fingers are curled.

No	Abjad	Gambar Citra
9	I	 A hand gesture where the index finger is extended upwards, and the other fingers are curled into the palm.
10	K	 A hand gesture where the index and middle fingers are extended upwards, and the other fingers are curled into the palm.
11	L	 A hand gesture where the index finger is extended upwards, and the thumb is extended to the side.
12	M	 A hand gesture where the index and middle fingers are extended upwards, and the other fingers are curled into the palm.
13	N	 A hand gesture where the index and middle fingers are extended upwards, and the other fingers are curled into the palm.
14	O	 A hand gesture where the thumb and index finger are extended upwards, and the other fingers are curled into the palm.

No	Abjad	Gambar Citra
15	P	 A hand gesture where the index and middle fingers are extended downwards, with the thumb tucked in and the ring and pinky fingers curled.
16	Q	 A hand gesture where the index and middle fingers are extended downwards, with the thumb pointing to the right and the ring and pinky fingers curled.
17	R	 A hand gesture where the index and middle fingers are extended upwards, with the thumb tucked in and the ring and pinky fingers curled.
18	S	 A hand gesture where the index and middle fingers are extended upwards, with the thumb pointing to the right and the ring and pinky fingers curled.
19	T	 A hand gesture where the index and middle fingers are extended upwards, with the thumb pointing to the right and the ring and pinky fingers curled.
20	U	 A hand gesture where the index and middle fingers are extended upwards, with the thumb pointing to the right and the ring and pinky fingers curled.

No	Abjad	Gambar Citra
21	V	
22	W	
23	X	
24	Y	

2.2 Citra Digital

Citra biasanya disebut sebagai representasi pada bidang dwimatra (2-D) dan dapat dilihat oleh sistem visual manusia. Citra adalah fungsi dua dimensi yang menggambarkan intensitas cahaya pada suatu bidang dan dapat dinyatakan secara matematis sebagai suatu persamaan (1) (Ratna, 2020)

$$f(x,y) \quad (1)$$

Pada persamaan (1) yaitu (x, y) mewakili koordinat dalam ruang dua dimensi, dan $f(x, y)$ menunjukkan kecerahan atau intensitas cahaya pada titik tertentu (x, y) .

Citra digital adalah representasi 2D yang dapat dilihat pada layar komputer, terdiri dari nilai digital diskrit yang dikenal sebagai *pixel* atau *picture elements* (Libertinus, 2021). Semakin besar resolusi citra, yang diukur dalam jumlah piksel, akan memerlukan ruang penyimpanan yang lebih besar pula (Gunardi, 2023). Citra digital direpresentasikan sebagai matriks berukuran $M \times N$ yang dapat dilihat pada persamaan (2) (Sembiring, 2012)

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \dots & \dots & \dots & \dots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \quad (2)$$

Pada persamaan (2) dimana $M \times N$ menyatakan resolusi citra dan setiap elemen matriks dan $f(x,y)$ menyatakan intensitas piksel pada posisi (x,y) dalam citra.

2.2.1 Pengolahan Citra Digital

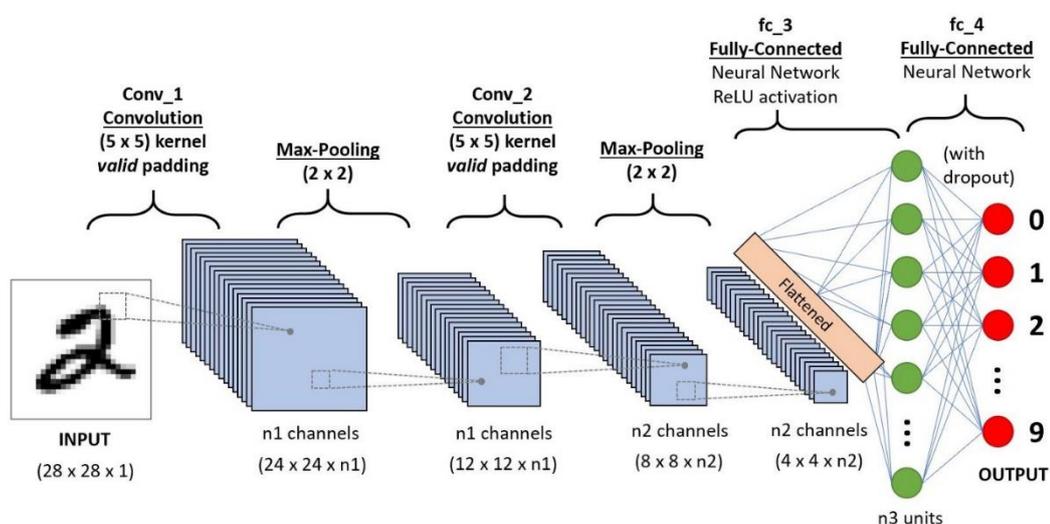
Pengolahan citra mengacu pada metode atau teknik memanipulasi dan memodifikasi citra melalui berbagai cara. Pengolahan citra digital mengacu pada manipulasi dan analisis gambar 2D menggunakan komputer yang melibatkan pemrosesan data dua dimensi di mana citra digital disimpan sebagai *array* nilai-nilai real atau kompleks yang direpresentasikan dalam bit-bit tertentu. Citra digital dianggap sebagai fungsi $f(x, y)$ dengan ukuran M baris dan N kolom, dengan x dan

y adalah koordinat spasial, dan nilai amplitudo f di titik koordinat (x, y) disebut tingkat keabuan, apabila nilai x, y , dan amplitudo f secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital (Salam et al., 2018).

2.3 Convolutional Neural Network

CNN adalah adaptasi dari *Multilayer Perceptron* (MLP) yang secara khusus dirancang untuk memproses data dua dimensi. CNN adalah *deep neural network* yang dicirikan oleh kemampuannya untuk menangani kedalaman jaringan yang tinggi, dan banyak diterapkan untuk memproses data citra (Sriyati et al., 2020). CNN pada awalnya dikenal sebagai *NeoCognitron*, yang diciptakan oleh Kunihiko Fukushima, seorang peneliti di NHK Broadcasting Science Research Laboratories yang berlokasi di Kinuta, Setagaya, Tokyo, Jepang, pada tahun 1980.

2.3.1 Konsep Convolutional Neural Network

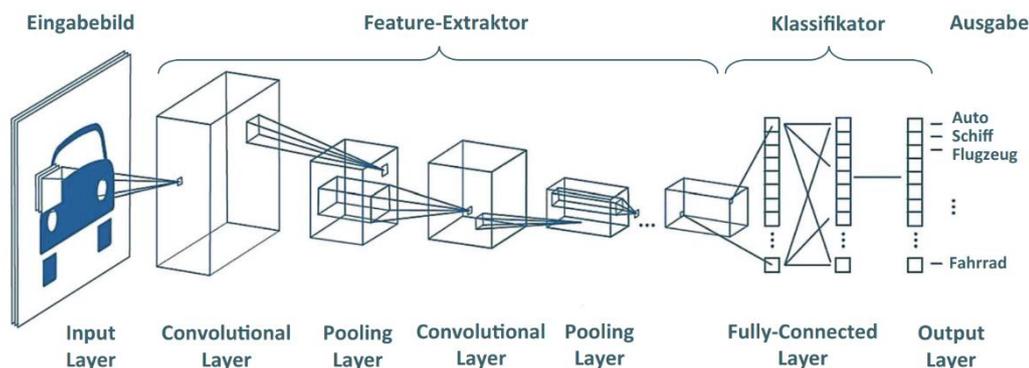


Gambar 2.1 Konsep *Convolutional Neural Network* (Munir, 2022)

Arsitektur *Convolutional Neural Network* (CNN) pada Gambar 2.1 memiliki beberapa komponen penting, dimulai dengan *input* gambar berwarna abu-abu yang menggambarkan angka 2 dalam format grid piksel ($28 \times 28 \times 1$), representasi ini diproses melalui lapisan-lapisan berikutnya untuk mengklasifikasikan gambar ke dalam kategori tertentu. Lapisan konvolusi pertama (*Conv_1*) menggunakan kernel (5×5) dengan *padding valid* untuk mengekstraksi fitur dari gambar, menghasilkan n_1 saluran dengan dimensi yang lebih rendah ($24 \times 24 \times n_1$). Lalu, lapisan *Max-Pooling* pertama menggunakan *filter* (2×2) untuk mengurangi dimensi fitur menjadi matriks dengan ukuran lebih kecil ($12 \times 12 \times n_1$). Dilanjutkan dengan lapisan konvolusi kedua (*Conv_2*) yang menggunakan kernel (5×5) dengan *padding valid* untuk mengekstraksi fitur lebih lanjut, menghasilkan n_2 saluran dengan dimensi yang lebih kompak ($8 \times 8 \times n_2$). Setelah itu, lapisan *Max-Pooling* kedua kembali menggunakan *filter* (2×2) untuk mengurangi dimensi fitur menjadi matriks dengan dimensi lebih kecil lagi ($4 \times 4 \times n_2$). Lalu, lapisan *Fully-Connected* (*fc_3* & *fc_4*) menggunakan fungsi aktivasi *ReLU* dan teknik *dropout* untuk mempelajari pola-pola kompleks dari fitur yang diekstraksi. Terakhir, lapisan *output* terdiri dari n_3 unit yang mewakili klasifikasi dari 0 hingga 9, di mana setiap unit sesuai dengan kategori digit yang mungkin.

2.3.2 **Arsitektur *Convolutional Neural Network***

Neural network adalah model yang terdiri dari *neuron* buatan yang saling terhubung yang meniru karakteristik jaringan saraf biologis otak manusia. Jaringan ini berfungsi dengan menerima rangsangan atau *input*, memprosesnya, dan kemudian menghasilkan *output*. (Mustakim et al., 2021).



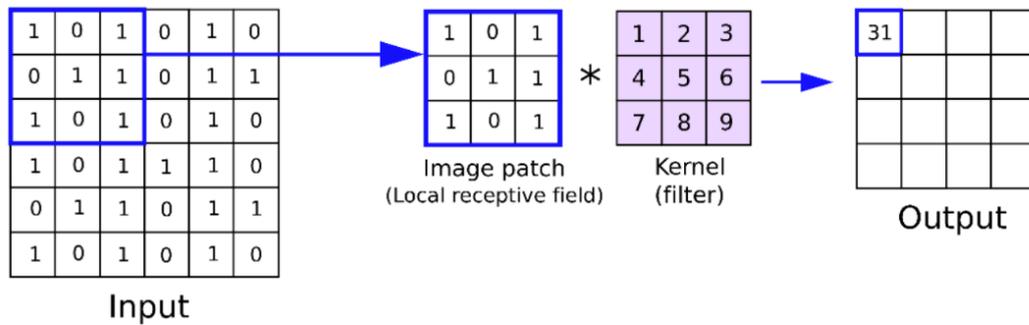
Gambar 2.2 Arsitektur *Convolutional Neural Network* (Zschech et al., 2021)

Berdasarkan Gambar 2.2, arsitektur CNN terbagi menjadi dua bagian besar yaitu: *convolutional layer* (terdiri dari *pooling layer*, *activation function* dan *hyperparameter*) dan *fully connected layer*.

a. *Convolutional Layer*

Lapisan konvolusi adalah komponen mendasar dari arsitektur CNN. Tahap ini melakukan operasi konvolusi pada *output layer* sebelumnya dan merupakan proses inti yang mendasari arsitektur CNN. Konvolusi adalah operasi matematika di mana suatu fungsi diterapkan berulang kali ke *output* fungsi lain. Secara khusus, operasi konvolusi melibatkan dua fungsi argumen bernilai nyata, dengan fungsi keluaran berfungsi sebagai peta fitur dari gambar masukan. *Input* dan *output* ini dapat dipandang sebagai dua argumen yang bernilai nyata. Lapisan konvolusional terdiri dari *neuron-neuron* yang disusun membentuk *filter* dengan dimensi tertentu (panjang dan tinggi dalam piksel). Misalnya, lapisan pertama pada tahap ekstraksi fitur biasanya berupa lapisan konvolusional dengan ukuran 5x5x3, yang berarti panjang 5 piksel, tinggi 5 piksel, dan 3 *filter* yang sesuai dengan saluran gambar. *Filter* ini digeser ke seluruh gambar, dan pada setiap posisi, operasi "titik" dilakukan antara masukan dan nilai *filter* untuk menghasilkan keluaran, yang biasa

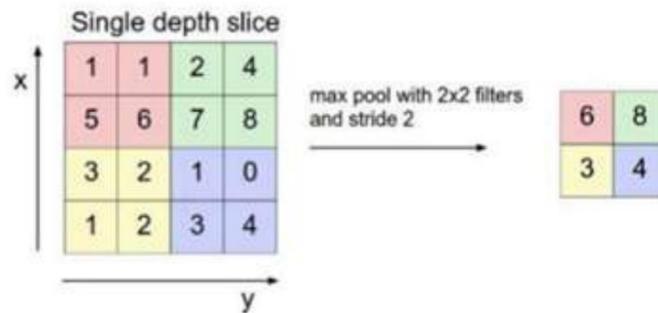
disebut sebagai peta aktivasi atau peta fitur. Ilustrasi dari *feature map* ditampilkan pada Gambar 2.3.



Gambar 2.3 Ilustrasi Feature Map (Fawwaz et al., 2020)

b. *Pooling*

Pooling merupakan pengurangan ukuran matriks menggunakan operasi *pooling*. Biasanya, *pooling layer* ditempatkan setelah lapisan konvolusi dan terdiri dari *filter* dengan ukuran dan *stride* tertentu yang bergerak melintasi area *feature map*. Terdapat dua jenis *pooling* yang umum digunakan pada *layer* ini yaitu *average pooling* dan *max-pooling*. Pada *average pooling*, nilai rata-rata yang diambil, sedangkan pada *max-pooling*, nilai maksimum yang dipilih. Memasukkan lapisan *pooling* di antara lapisan konvolusi yang berurutan dalam arsitektur model CNN secara bertahap mengurangi ukuran volume *output* pada peta *feature map*. Pengurangan ini menurunkan jumlah parameter dan komputasi dalam jaringan, sehingga membantu mengendalikan *overfitting*. Lapisan *pooling* bekerja pada setiap tumpukan *feature map*, mengurangi ukurannya. Umumnya, *pooling layer* menggunakan *filter 2x2* yang diterapkan dalam dua langkah, yang berfungsi pada setiap irisan input. Operasi *max pooling* ditampilkan pada Gambar 2.4.

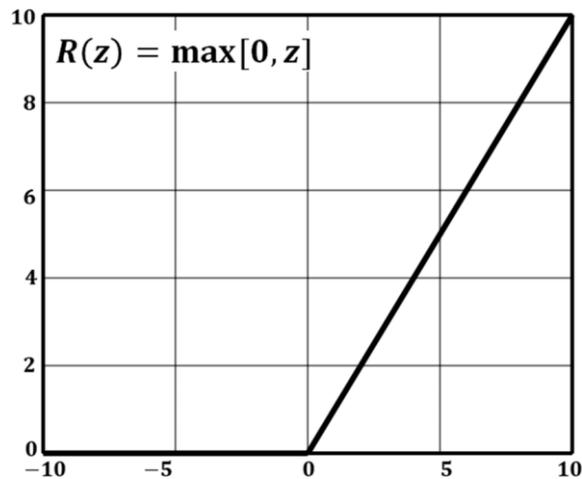


Gambar 2.4 Contoh *Operasi Max Pooling* (Fawwaz et al., 2020)

Gambar 2.4 mengilustrasikan proses *max pooling*. *Output* dari proses *pooling* ini adalah matriks dengan dimensi yang lebih kecil dari gambar asli. Lapisan *pooling* beroperasi secara berurutan pada setiap irisan kedalaman dari volume *input*. Seperti yang ditunjukkan pada gambar diatas, operasi *max-pooling* menggunakan *filter 2x2*. *Input* untuk proses ini adalah 4x4, dan nilai maksimum dari setiap kelompok 4 angka pada *input* dipilih untuk membentuk matriks *output* baru dengan dimensi 2x2.

c. *Activation Function: ReLu (Rectified Linear Units)*

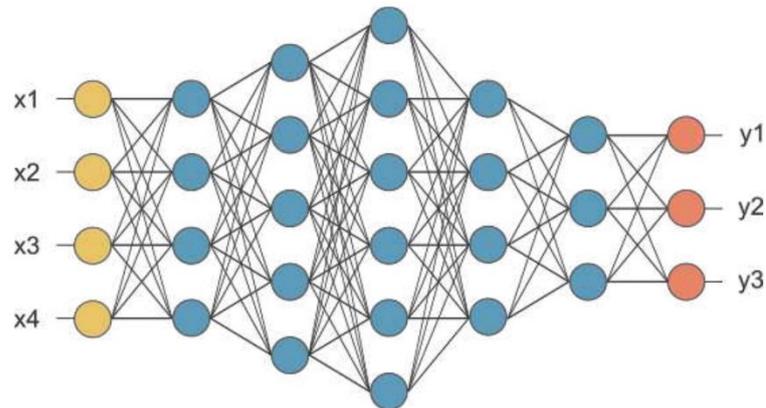
Rectified Linear Units (ReLU), menerapkan fungsi $R(z) = \max(0, z)$ untuk meningkatkan nonlinieritas fungsi keputusan dan seluruh jaringan tanpa mengubah bidang reseptif pada lapisan *convolutional layer*. Contoh operasi *ReLU* seperti pada Gambar 2.5.



Gambar 2.5 Ilustrasi *Rectified Linear Units (ReLU)* (Nurhikmat, 2018)

d. *Fully Connected Layer*

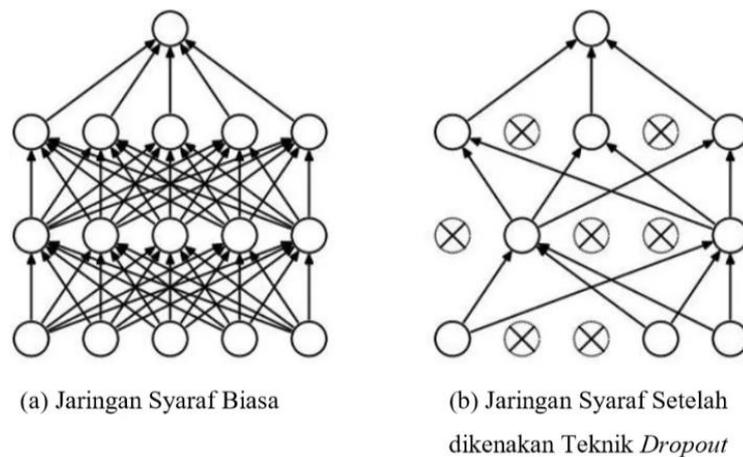
Fully-Connected Layer adalah dimana semua *neuron* aktivasi dari *layer* sebelumnya terhubung ke *neuron* di *layer* berikutnya, seperti dengan *neural network* biasa. *Layer* ini biasanya digunakan dalam *Multi-Layer Perceptron (MLP)* untuk mengubah dimensi data untuk klasifikasi linier. Tidak seperti lapisan *fully-connected* dan lapisan konvolusional biasa, di mana *neuron* hanya terhubung ke daerah tertentu dari input, *neuron* dalam lapisan *fully-connected* terhubung sepenuhnya. Namun, kedua lapisan tersebut masih beroperasi pada produk *dot*, sehingga fungsinya sangat mirip. Gambar 2.7 menunjukkan contoh dari proses *fully-connected*.



Gambar 2.6 Ilustrasi *Fully Connected Layer* (Bakti & Firdaus, 2023)

e. *Dropout*

Dropout adalah teknik regularisasi jaringan saraf di mana beberapa *neuron* dipilih secara acak untuk diabaikan selama proses pelatihan, dan secara efektif membuangnya secara acak. Ini berarti bahwa kontribusi dari *neuron-neuron* yang dibuang ini dihentikan, dan *neuron* tidak diterapkan dengan bobot baru dan jaringan selama *backpropagation*. Gambar 2.7 menunjukkan contoh proses penggunaan teknik tanpa menggunakan *dropout* dan teknik dengan menggunakan *dropout*.



Gambar 2.7 Ilustrasi *Dropout* (Apa Itu Convolutional Neural Network? | by QOLBIYATUL LINA | Medium, n.d.)

Pada Gambar 2.7 (a) adalah jaringan syaraf biasa, semua *neuron* di setiap lapisan terhubung sepenuhnya ke *neuron* di lapisan berikutnya. Selama pelatihan, semua *neuron* aktif dan berkontribusi terhadap *output*. Pada Gambar 2.7 (b) adalah jaringan syaraf setelah dikenakan teknik dropout, beberapa *neuron* secara acak dinonaktifkan (ditandai dengan tanda silang) selama setiap iterasi pelatihan. Koneksi dari *neuron* yang dinonaktifkan juga diabaikan dan tidak berkontribusi terhadap *forward pass* atau *backpropagation* dalam proses iterasi itu.

f. *Softmax Classifier*

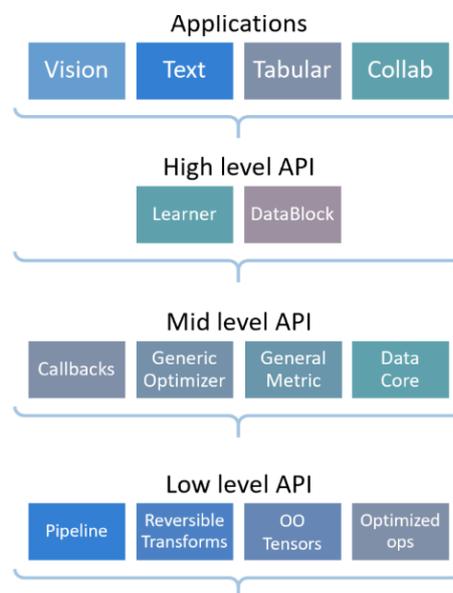
Softmax Classifier diperluas dari fungsi logistik dan menghasilkan *output* yang mewakili distribusi kategori. *Softmax function* ini digunakan dalam berbagai teknik klasifikasi termasuk *naive bayes classifier*, *multinomial logistic regression*, *multiclass linear discriminant analysis*, dan *neural network*. Secara khusus, ini banyak digunakan dalam *multinomial logistic regression* dan *multiclass linear discriminant analysis*. Fungsi yang digunakan dapat dilihat pada persamaan (3) (S. Sun et al., 2021).

$$p_i = s(z)_i = \frac{e^{z_i}}{\sum_{j=1}^d e^{z_j}} \quad (3)$$

Fungsi eksponensial dieksekusi pada setiap elemen z_i dari vektor *input* z dan nilai *outputnya* dinormalisasi dengan membaginya dengan jumlah dari semua eksponensial. Operasi normalisasi memastikan bahwa setiap elemen dalam vektor *output* p_i menjumlahkan menjadi 1.

2.4 *FastAI*

Fastai adalah sebuah perpustakaan (*library*) dalam bidang *deep learning* yang menyediakan komponen-komponen tingkat tinggi (*High level API*) yang dapat dengan cepat dan mudah mendapatkan hasil dalam domain *deep learning* standar, selain itu, *fastai* juga menyediakan komponen-komponen tingkat rendah (*Low level API*) yang dapat disesuaikan untuk membangun pendekatan-pendekatan baru (Howard & Gugger, n.d.).



Gambar 2.8 Lapisan *API FastAI* (Howard & Gugger, n.d.)

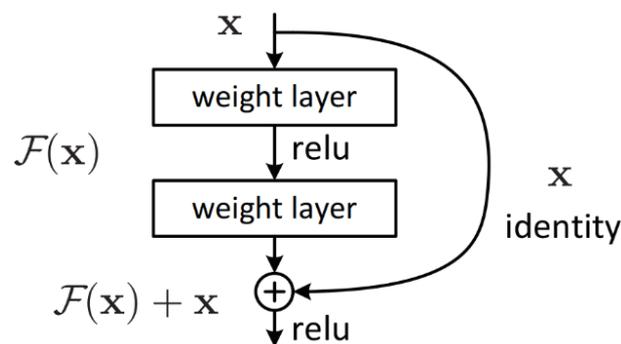
Pada Gambar 2.8 fokus penelitian yang akan dilakukan berada pada tingkatan High level API dengan menggunakan fungsi learner. Dan pada bagian applications menggunakan jenis vision untuk melakukan proses pengenalan objek.

2.5 *Residual Network (ResNet)*

Residual Network adalah jenis arsitektur *Convolution Neural Network (CNN)* yang digunakan dalam *deep learning*. *ResNet* yang biasa dikenal karena

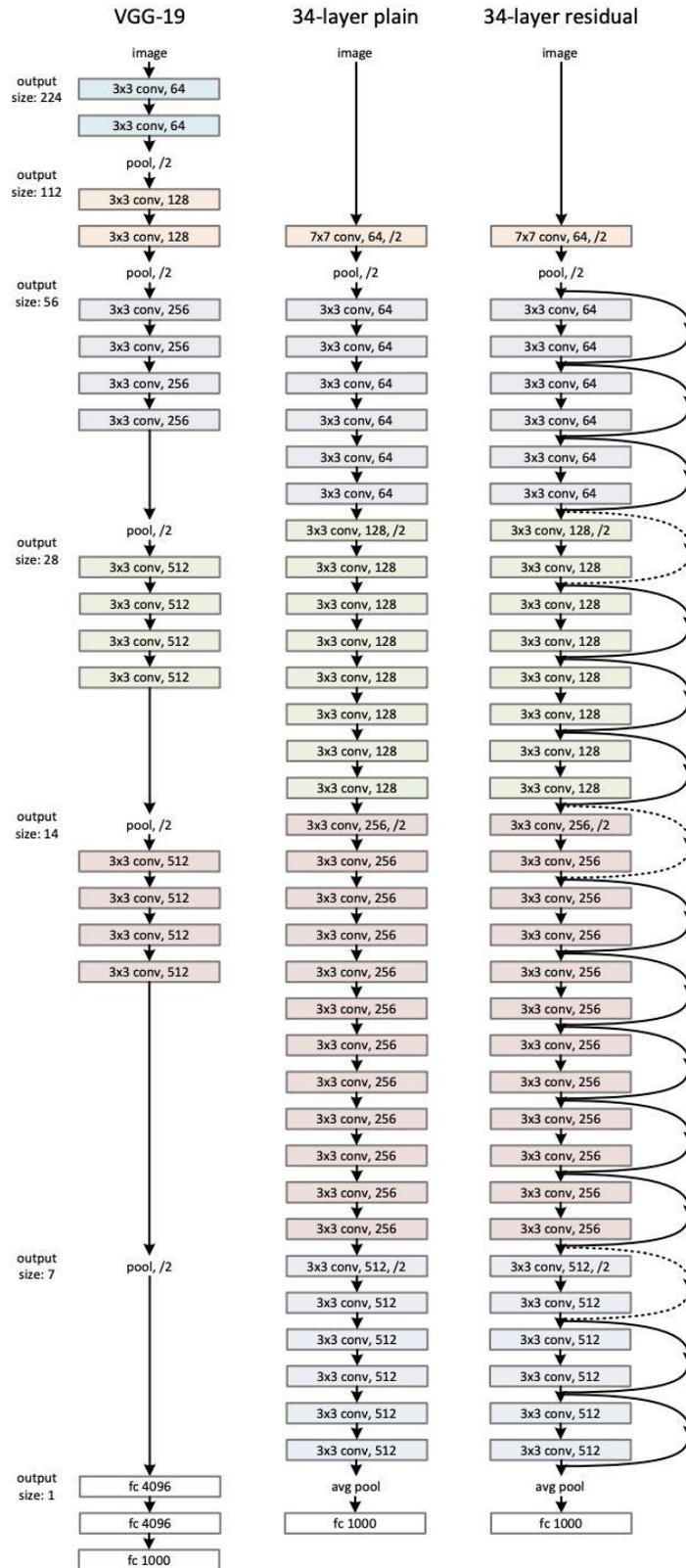
kemampuannya untuk mengatasi masalah *vanishing gradient* yang biasa ditemui dalam jaringan saraf *deep learning*. *ResNet* merupakan *pretrained model* yang sudah terlatih dan sudah dikonfigurasi sebelumnya, sehingga tidak memerlukan pengaturan khusus untuk lapisan-lapisannya (Ridhovan & Suharso, 2022). *ResNet* berfungsi dengan membangun jaringan yang lebih dalam daripada jaringan biasa dan bersamaan mengoptimalkan jumlah lapisan untuk mengurangi masalah hilangnya gradien. (Ridhovan & Suharso, 2022).

Pada sejumlah arsitektur CNN sebelumnya, umumnya akurasi cenderung menurun, menghasilkan tingkat *error* yang lebih tinggi. *ResNet* mengatasi masalah ini dengan memperkenalkan *residual learning*, yang menambahkan cara untuk melewati antar lapisan. (He et al., 2016). Blok *Residual Learning* ditampilkan pada Gambar 2.9.



Gambar 2.9 *Residual Learning Block* (He et al., 2016)

Pada Gambar 2.9 adalah proses menambahkan jalan pintas yang bertindak sebagai fungsi identitas atau *identity mapping*. Pintasan ini secara tidak langsung menggerakkan satu atau lebih kelas dan menciptakan blok *residual*, pada Gambar 2.10 adalah contoh arsitektur *ResNet-34*.



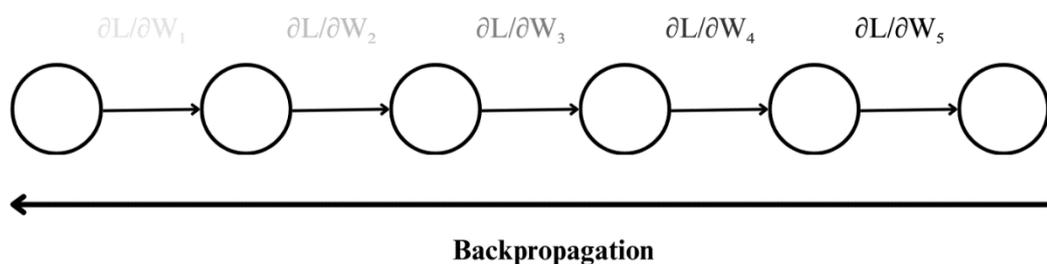
Gambar 2.10 Arsitektur ResNet-34 (He et al., 2016)

Penggunaan *residual block* ditunjukkan pada Gambar 2.10 pada ujung kanan, terdapat arsitektur *residual block*, lalu arsitektur *VGGnet* berada di paling kiri dan arsitektur tanpa *residual block* berada di tengah-tengah antara keduanya.

Arsitektur dengan menggunakan *residual block* melewati lebih banyak lapisan *layer*, sehingga dapat mengatasi masalah *vanishing gradient* pada saat proses *backpropagation* karena gradien dapat mencapai lapisan pertama tanpa perlu *block* yang berlebihan dan jumlah perkalian yang sama dengan arsitektur yang sama.

2.6 *Vanishing Gradient*

Vanishing gradient terjadi ketika gradien hasil *training* gagal mencapai *layer* secara efektif ke lapisan awal karena operasi konvolusi yang berulang-ulang, sehingga lapisan-lapisan ini tidak menerima gradien apapun (Cendekia Vandara et al., n.d.-a). Masalah ini menyebabkan CNN tidak mendapatkan informasi dari *error* yang dihitung. Ilustrasi dari *vanishing gradient* ditampilkan pada Gambar 2.11.



Gambar 2.11 *Vanishing Gradient*

Pada Gambar 2.11 dalam proses *backpropagation*, gradien dari fungsi *loss* dihitung mundur dari lapisan *output* ke lapisan *input*. Saat gradien dihitung dan dibawa mundur melewati beberapa *layer*, gradien ini sering kali menjadi semakin

kecil. Jika fungsi aktivasi yang digunakan menyebabkan gradien menyusut (seperti sigmoid yang memiliki rentang *output* antara 0 dan 1), gradien dapat menjadi sangat kecil setelah melewati beberapa *layer*. Ketika gradien yang sangat kecil digunakan untuk mengupdate bobot pada lapisan-lapisan awal yaitu $\partial L/\partial W_1$, perubahan bobot menjadi sangat kecil yang menyebabkan lapisan-lapisan awal belajar sangat lambat atau bahkan tidak belajar sama sekali.

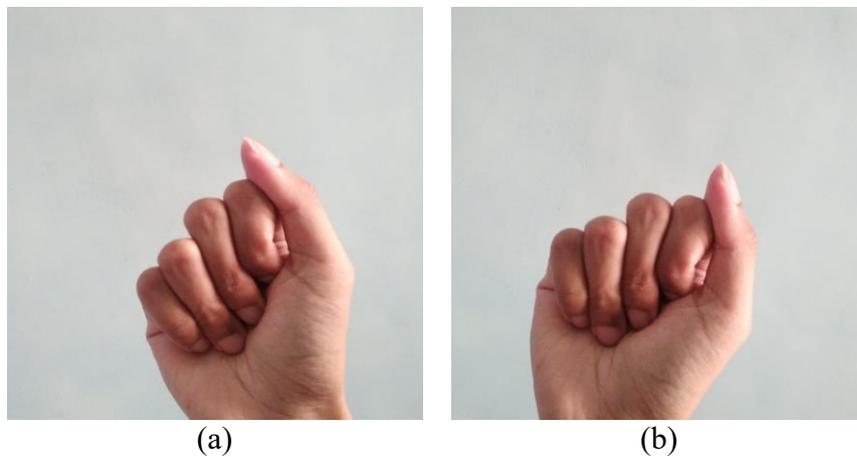
2.7 Augmentasi Citra

Augmentasi citra adalah teknik yang memungkinkan para praktisi untuk meningkatkan variasi data yang digunakan untuk model pelatihan, tanpa perlu mengumpulkan data baru. Teknik seperti *padding*, *cropping*, dan *flipping horizontal* sering digunakan untuk melatih jaringan *neural* besar (Sanjaya & Ayub, 2020). Augmentasi citra melibatkan penerapan deformasi pada sampel pelatihan beranotasi, yang merupakan sampel *noise* yang tidak ditemukan pada data uji, sehingga menciptakan data pelatihan yang baru dan tidak redudan. Teknik ini meningkatkan model dan mengurangi *overfitting* dengan memperluas kumpulan data pelatihan tanpa memerlukan informasi asli tambahan (Sanusi et al., 2023).

Teknik augmentasi yang akan dilakukan dalam penelitian ini adalah teknik *rotation*, *shear horizontal* dan *vertical*, *grayscale*, *saturation*, *brightness* dan *exposure*. Implementasi teknik augmentasi dalam penelitian ini menggunakan bantuan *tools roboflow*.

a. *Rotation*

Teknik *rotation* adalah operasi transformasi pada citra yang berguna untuk menggeser citra pada suatu sudut tertentu. Teknik rotasi yang digunakan adalah sebesar -10° dan $+10^\circ$.

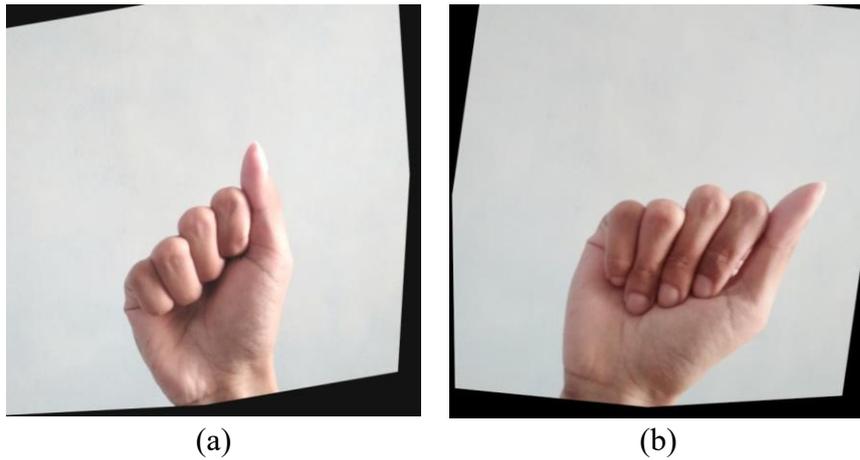


Gambar 2.12 Teknik *Rotation*

Gambar 2.12 (a) adalah gambar dengan menggunakan teknik *rotation* sebesar -10° dan Gambar 2.12 (b) adalah gambar dengan menggunakan teknik *rotation* sebesar $+10^\circ$.

b. *Shear Horizontal* dan *Vertical*

Teknik *shear horizontal* dan *vertical* adalah teknik penggeseran yang mengakibatkan citra akan terdistorsi sepanjang sumbu x dan sumbu y. Digunakan untuk memperbaiki sudut sehingga dapat lebih mudah dipahami oleh *computer*. Teknik *shear horizontal* dan *vertical* yang digunakan adalah sebesar $\pm 10^\circ$ *horizontal* dan $\pm 10^\circ$ *vertical*.



Gambar 2.13 Teknik *Shear Horizontal* dan *Vertical*

Gambar 2.13 (a) adalah gambar dengan menggunakan teknik *shear horizontal* sebesar $\pm 10^\circ$ dan Gambar 2.13 (b) adalah gambar dengan menggunakan teknik *shear vertical* sebesar $\pm 10^\circ$.

c. *Grayscale*

Teknik *grayscale* adalah teknik mengubah gambar berwarna menjadi gambar hitam-putih (monokrom) yang hanya memiliki berbagai tingkatan abu-abu. Teknik *grayscale* yang digunakan adalah sebesar 15% dari total keseluruhan *dataset*.

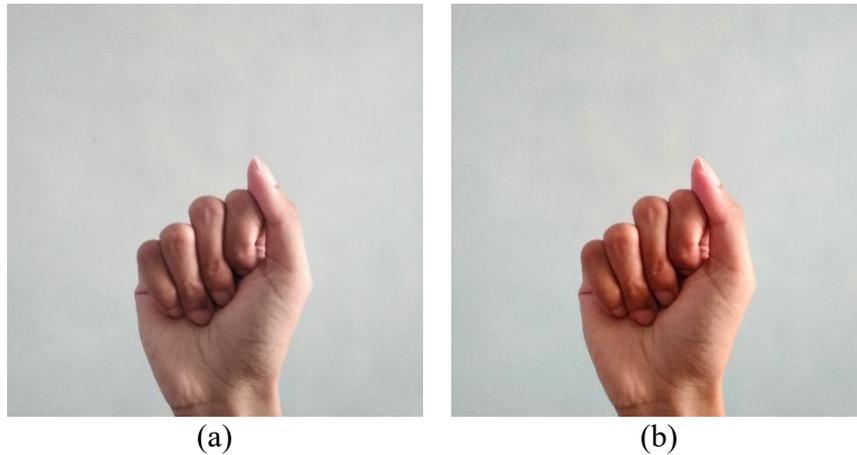


Gambar 2.14 Teknik *Grayscale*

Gambar 2.14 adalah gambar dengan menggunakan teknik *grayscale*.

d. *Saturation*

Teknik *saturation* adalah teknik mengubah intensitas warna dalam gambar. Mengubah saturasi dapat membuat warna dalam gambar lebih hidup atau lebih pudar. Teknik *saturation* yang digunakan adalah sebesar -30% dan +30%.

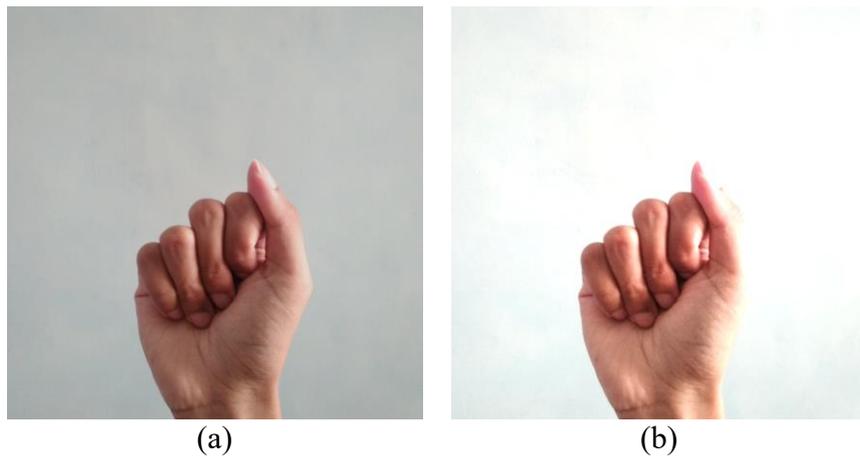


Gambar 2.15 Teknik *Saturation*

Gambar 2.15(a) adalah gambar dengan menggunakan teknik *saturation* sebesar -15% dan Gambar 2.15(b) adalah gambar dengan menggunakan teknik *saturation* sebesar +15%.

e. *Brightness*

Teknik *brightness* adalah teknik penyesuaian kecerahan yang melibatkan penambahan atau pengurangan nilai piksel secara acak dalam citra, dapat mengubah tingkat kecerahan keseluruhan citra. Teknik *brightness* yang digunakan adalah sebesar -15% dan +15%.

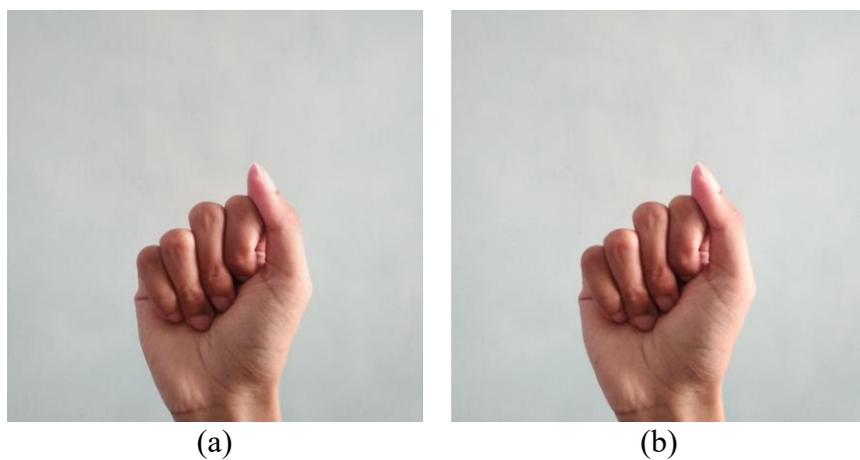


Gambar 2.16 Teknik *Brightness*

Gambar 2.16(a) adalah gambar dengan menggunakan teknik *brightness* sebesar -15% dan Gambar 2.16(b) adalah gambar dengan menggunakan teknik *brightness* sebesar +15%.

f. *Exposure*

Teknik *exposure* adalah teknik mengatur tingkat pencahayaan pada gambar, sehingga gambar bisa menjadi lebih terang atau lebih gelap. Teknik *exposure* yang digunakan adalah sebesar -15% dan +15%.

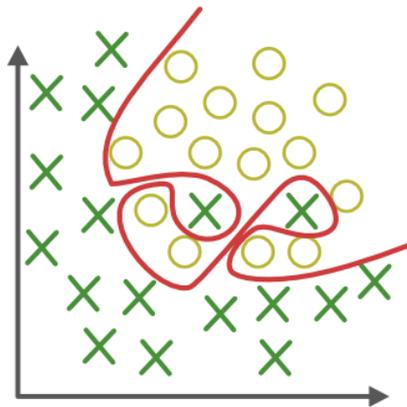


Gambar 2.17 Teknik *Exposure*

Gambar 2.17(a) adalah gambar dengan menggunakan teknik *exposure* sebesar -15% dan Gambar 2.17(b) adalah gambar dengan menggunakan teknik *exposure* sebesar +15%.

2.8 *Overfitting*

Overfitting adalah kondisi di mana model mempelajari data pelatihan dengan sangat baik hingga hampir selalu menghasilkan prediksi yang benar (Mei Sarah et al., 2023). Pada Gambar 2.18 menunjukkan kasus *overfitting* pada model.



Gambar 2.18 *Overfitting* (dewangnautiyal, 2024)

Tanda silang hijau (X) dan lingkaran kuning (O) mewakili dua kelas berbeda dari data pelatihan. Garis merah yang berkelok-kelok menunjukkan keputusan model yang telah *overfit* terhadap data pelatihan. Model pada Gambar 2.18 telah belajar secara mendetail pola dan *noise* dari data pelatihan, sehingga mampu memisahkan setiap contoh dalam data pelatihan dengan sangat tepat. Meskipun model memberikan hasil yang hampir sempurna pada data pelatihan (karena mengikuti setiap titik data dengan tepat), model kemungkinan besar akan berkinerja buruk pada data baru atau data validasi. Hal ini disebabkan bahwa model ini terlalu

kompleks dan disesuaikan dengan data pelatihan, termasuk *noise* dan *outlier*, sehingga tidak dapat mengenali dengan baik pada data yang belum pernah dilihat sebelumnya.

2.9 *Hardware Tools dan Software Tools*

Alat bantu dalam pengerjaan penelitian ini dibagi menjadi dua bagian yaitu *hardware tools* dan *software tools*. Adapun objek yang diuji spesifikasi *minimum hardware* dan *software* yang digunakan dalam penulisan ini ditampilkan pada Tabel 2.2.

Tabel 2.2 Tabel Hardware dan Software

No	<i>Hardware</i>	<i>Software</i>
1	Laptop HP 1000 Notebook PC	Jupyter Notebook versi 7.0.7
2	Xiaomi Redmi Note 8 Pro	Microsoft Visual Studio Code versi 1.89.1
3	-	Roboflow
4	-	Canva
5	-	Draw.io

2.10 Penelitian Terkait

Literatur yang digunakan dalam penelitian ini terdiri dari jurnal ilmiah nasional maupun internasional terkait pengembangan model menggunakan arsitektur *Residual Network*. Referensi ini digunakan sebagai dasar acuan melakukan pengembangan model deteksi menggunakan arsitektur *ResNet* dengan tujuan menghasilkan nilai akurasi yang optimal. Pada Tabel 2.3 merupakan tabel penelitian terkait.

Tabel 2.3 Penelitian Terkait

<i>State of The Art – Convolutional Neural Network dan Augmentasi Citra</i>			
Penulis	Judul	Algoritma dan Teknik	Hasil
Riyadi et al., 2022	Perbandingan Metode <i>ResNet</i> , <i>YoloV3</i> , dan <i>TinyYoloV3</i> pada Deteksi Citra dengan Pemrograman <i>Python</i>	<i>Residual Network</i>	Tingkat akurasi = 86,96%
HARIYANI et al., 2020	Deteksi Penyakit Covid-19 Berdasarkan Citra <i>X-Ray</i> Menggunakan <i>Deep Residual Network</i>	<i>Residual Network-50</i>	Hasil <i>precision</i> = 0,98 Hasil <i>recall</i> = 0,95 Hasil <i>F1-score</i> = 0,97 Hasil akurasi = 99%
Sanjaya & Ayub, 2020	Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup	<i>Residual Network-152</i>	Tingkat akurasi = 82,6714%
Ridhovan & Suharso, 2022	Penerapan Metode <i>Residual Network</i> (<i>Resnet</i>) Dalam Klasifikasi Penyakit Pada Daun Gandum	<i>Residual Network-152V2</i>	Tingkat akurasi = 98%
Ma & Long, 2023	A Face Recognition Method Using <i>ResNet34</i> and <i>RetinaFace</i>	<i>Residual Network-34 & Retina Face</i>	Tingkat akurasi = 98,78%
Zhao et al., 2021	A combination method of stacked autoencoder and 3D deep residual network for hyperspectral image classification	<i>Stacked Autoencoder (SAE) & 3D Deep Residual Network (3DDRN)</i>	Tingkat akurasi = 98,97%, 99.69%, 99,24%
Latif et al., 2023	Deep learning in Transportation: Optimized driven deep residual networks for Arabic traffic sign recognition	<i>Residual Network V1 & Residual Network V2</i>	Tingkat akurasi = 99,18%, 96,14%

<i>State of The Art – Convolutional Neural Network dan Augmentasi Citra</i>			
Penulis	Judul	Algoritma dan Teknik	Hasil
Tasci et al., 2023	InCR: Inception and concatenation residual block-based deep learning network for damaged building detection using remote sensing images	<i>Inception and Concatenation Residual (InCR) Deep Learning Network</i>	Tingkat akurasi = 98%
Han et al., 2022	Metallic product recognition with dual attention and multi-branch residual blocks-based convolutional neural networks	<i>Residual Network</i>	Tingkat akurasi = 94,31%, 95,88%
Pattanaik et al., 2022	Malaria detection using deep residual networks with mobile microscopy	<i>Multi-Magnification Deep Residual Neural Network (MM-ResNet)</i>	Tingkat akurasi = 98.08%
Niu et al., 2024	Automated graptolite identification at high taxonomic resolution using residual networks	<i>Squeeze-and-Excitation Residual Network (SE-ResNet-50)</i>	Tingkat akurasi = 81%
Sahoo et al., 2023	Residual learning for segmentation of the medical images in healthcare	<i>Three-Dimensional Residual Networks (ResUNet++)</i>	Tingkat akurasi = 87,24%
Long et al., 2020	Automatic Classification of Cichlid Behaviors Using 3D Convolutional Residual Networks	<i>Pixel-based Hidden Markov Model Density-based Spatiotemporal Clustering & 3D Residual Networks (3D ResNet)</i>	Tingkat akurasi = >76%
(Wang et al., 2023)	A diagnostic classification of lung nodules using multiple-scale residual network	<i>Multiple-Scale Residual Network (MResNet)</i>	<i>MResNet</i> memiliki akurasi sebesar 85.23%, sensitivitas 92.79%, spesifisitas 72.89%, nilai positif prediktif (PPV) 84.56%, dan nilai negatif prediktif (NPV) 86.34%. AUC adalah 0.9275.

<i>State of The Art – Convolutional Neural Network dan Augmentasi Citra</i>			
Penulis	Judul	Algoritma dan Teknik	Hasil
X. Ma et al., 2024	ERCPC-Net: a channel extension residual structure and adaptive channel attention mechanism for plant leaf disease classification network	<i>Extension Residual Channel Attention Network</i> 1. <i>Channel Extension Residual Block (CER-Block)</i> 2. <i>Adaptive Channel Attention Block (ACA-Block)</i> 3. <i>Bidirectional Information Fusion Block (BIF-Block)</i>	Hasil akurasi = 99,82%
Mekruksavanich et al., 2022	A Hybrid Deep Residual Network for Efficient Transitional Activity Recognition Based on Wearable Sensors	<i>Residual Network (ResNet) & Bidirectional Gated Recurrent Unit (BiGRU)</i>	1. Hasil akurasi untuk dataset HAPT = 98,03% 2. Hasil akurasi untuk dataset MobiAct v2.0 = 98,03%
Wadekar & Singh, 2023	A modified convolutional neural network framework for categorizing lung cell histopathological image based on residual network	<i>Pre-trained Model (VGG19)</i>	Hasil akurasi = 97,73%
Sun et al., 2021	An Improved Deep Residual Network Prediction Model for the Early Diagnosis of Alzheimer's Disease	<i>Deep Residual Network (ResNet) & Spatial Transformer Networks (STN)</i>	Hasil akurasi = 97,1% Hasil akurasi precision = 95,5% Hasil akurasi recall = 95,3% Hasil akurasi F1 = 95,4%
Sarwinda et al., 2021	Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer	Perbandingan <i>ResNet-18 & ResNet-50</i>	Hasil ResNet-18: 1. Akurasi = 85% 2. Sensitivity = 83% 3. Specificity = 87% Hasil ResNet-50: 1. Akurasi = 88% 2. Sensitivity = 93% 3. Specificity = 83%

State of The Art – Convolutional Neural Network dan Augmentasi Citra			
Penulis	Judul	Algoritma dan Teknik	Hasil
Prasetyo et al., 2022	Multi-level residual network VGGNet for fish species classification	<i>Multi-Level Residual (MLR) & VGGNet</i>	Hasil akurasi = 99,69%
Sasongko et al., 2023	Analisis Efek Augmentasi <i>Dataset</i> dan <i>Fine Tune</i> pada algoritma <i>Pre-trained Convolutional Neural Network (CNN)</i>	<i>Mobilenet, Efficientnet, Nasnetmobile</i> dan Teknik Augmentasi	Jumlah peningkatan akurasi setelah dilakukan augmentasi random erase ataupun zoom range yang terjadi sekitar 0.03% hingga 0.1%.
Pradnya D & Kusumaningtyas, 2022	Analisis Pengaruh Data Augmentasi Pada Klasifikasi Bumbu Dapur Menggunakan <i>Convolutional Neural Network</i>	<i>Pre-Trained Convolutional Neural Network (CNN)</i> dan Teknik Augmentasi	Hasil akurasi sebelum menggunakan teknik augmentasi adalah sebesar 54% dan hasil setelah menggunakan teknik augmentasi adalah sebesar 80%.
Putri Ayuni et al., 2023	Augmentasi data pada Implementasi <i>Convolutional Neural Network</i> arsitektur <i>Efficientnet-B3</i> untuk Klasifikasi Penyakit Daun Padi	<i>Efficientnet-B3</i> dan Teknik Augmentasi	Hasil akurasi dengan menggunakan rasio perbandingan data latih dan data uji 90:10 sebelum dilakukan teknik augmentasi adalah sebesar 92,39%, setelah menggunakan teknik augmentasi adalah sebesar 98,91%.
Fadillah et al., 2021	Data Augmentasi Untuk Mengatasi Keterbatasan Data Pada Model Penerjemah Bahasa Isyarat Indonesia (BISINDO)	<i>Convolutional Neural Network (CNN)</i> dan Teknik Augmentasi	Hasil akurasi sebelum menggunakan teknik augmentasi adalah sebesar 64,38% dan hasil setelah menggunakan teknik augmentasi adalah sebesar 94,38%.

<i>State of The Art – Convolutional Neural Network dan Augmentasi Citra</i>			
Penulis	Judul	Algoritma dan Teknik	Hasil
Penelitian	Pengembangan Model Deteksi Sistem Bahasa Isyarat Indonesia Menggunakan Arsitektur Resnet-34 Dan Augmentasi Citra	<i>Residual Network-34 & Augmentasi Citra</i>	?

Pada Tabel 2.3 terdapat beberapa kelemahan dalam penggunaan algoritma *Convolutional Neural Network (CNN)* khususnya dalam keakuratan hasil yang kurang optimal karena permasalahan *vanishing gradient* dan *overfitting* yang timbul dari algoritma CNN. Maka solusi yang dapat diberikan untuk mengatasi masalah *vanishing gradient* adalah dengan menambahkan pemetaan identitas (*identity mapping*) dan lapisan lainnya yang disalin dari model yang lebih dangkal yaitu dengan menggunakan jaringan *residual* atau *ResNet*. Penambahan teknik augmentasi citra juga perlu dilakukan untuk mengatasi permasalahan *overfitting* pada model dan teknik ini dapat meningkatkan kinerja model yang berpengaruh pada hasil akurasi model. Pada Tabel 2.4 menunjukkan perbandingan capaian yang akan didapatkan pada penelitian ini dengan capaian yang telah didapatkan oleh penelitian sebelumnya. Perbandingan ini mengacu pada kinerja model yang menggunakan arsitektur *ResNet-34* dan teknik augmentasi citra dengan model tanpa menggunakan teknik augmentasi citra.

Tabel 2.4 Perbandingan Target Capaian Penelitian

Penelitian	<i>Convolutional Neural Network (CNN)</i>	<i>Residual Network (ResNet)</i>	Augmentasi Citra
HARIYANI et al., 2020	√	√	-
Fadillah et al., 2021	√	-	√
Ridhovan & Suharso, 2022	√	√	-
Putri Ayuni et al., 2023	√	-	√
Han et al., 2022	√	√	-
H. Sun et al., 2021	√	√	-
(Pradnya D & Kusumaningtyas, 2022)	√	-	√
Penelitian ini	√	√	√

Pada Tabel 2.4 terdapat beberapa penelitian terkait mengenai pengembangan model deteksi objek menggunakan arsitektur *ResNet* dan penerapan teknik augmentasi pada *dataset*. Penelitian pertama yang dilakukan oleh (HARIYANI et al., 2020) menggunakan arsitektur *ResNet-50* untuk mendeteksi penyakit Covid-19, memiliki nilai akurasi sebesar 99%. Penelitian selanjutnya yang dilakukan oleh (Ridhovan & Suharso, 2022) menggunakan arsitektur *ResNet-152V2* untuk mengklasifikasikan penyakit pada daun gandum, memiliki nilai akurasi 98%. Penelitian selanjutnya yang dilakukan oleh (Han et al., 2022) menggunakan arsitektur *ResNet* untuk pengenalan produk logam, memiliki nilai akurasi sebesar 95,88%. Dan penelitian yang dilakukan oleh (H. Sun et al., 2021) menggunakan arsitektur *ResNet* untuk melakukan prediksi awal pada penyakit *alzheimer*, memiliki nilai akurasi sebesar 97,1%. Meskipun beberapa penelitian terdahulu menggunakan arsitektur *ResNet* memiliki nilai akurasi yang optimal,

tetapi masih memiliki risiko akan terjadinya *overfitting* pada model. Sehingga diperlukan cara untuk mencegah terjadinya *overfitting* pada model, yaitu dengan menerapkan teknik augmentasi pada *dataset* yang akan digunakan.

Penelitian pertama yang menerapkan teknik augmentasi telah dilakukan oleh (Fadillah et al., 2021) untuk mengatasi keterbatasan *dataset* pada model penerjemah BISINDO, sebelum dilakukannya teknik augmentasi, hasil akurasi adalah 64,38% dan setelah teknik augmentasi hasil akurasi meningkat menjadi 94,38%. Penelitian selanjutnya dilakukan oleh (Putri Ayuni et al., 2023) untuk klasifikasi penyakit pada daun padi, dan sebelum dilakukannya teknik augmentasi, hasil akurasi adalah 92,39% dan setelah teknik augmentasi hasil akurasi meningkat menjadi 98,91%. Penelitian yang dilakukan oleh (Pradnya D & Kusumaningtyas, 2022) untuk mengklasifikasikan bumbu dapur, sebelum dilakukannya teknik augmentasi, hasil akurasi adalah 54% dan setelah teknik augmentasi hasil akurasi meningkat menjadi 80%. Penelitian terdahulu dengan menerapkan teknik augmentasi berhasil mencegah terjadinya *overfitting* pada model, dan berhasil meningkatkan nilai akurasi akhir. Permasalahan yang dihadapi dari penelitian-penelitian tersebut adalah risiko terjadinya *vanishing gradient* pada model.

Penelitian-penelitian terkait tersebut memiliki risiko akan terjadinya *vanishing gradient* dan *overfitting* karena tidak menggunakan arsitektur *ResNet* dan penerapan teknik augmentasi. Selain itu, hanya sedikit penelitian yang menerapkan arsitektur *ResNet* dan teknik augmentasi, bahkan tidak ada sama sekali penelitian yang menggunakan arsitektur *ResNet* dan teknik augmentasi untuk pengenalan objek dari alfabet SIBI. Maka dari itu, penelitian ini akan menghasilkan kebaruan

yang tidak dimiliki oleh penelitian-penelitian terkait pada tabel 2.4. Penelitian ini akan menerapkan arsitektur ResNet-34 dengan beberapa teknik augmentasi seperti *rotation*, *shear horizontal* dan *vertical*, *grayscale*, *saturation*, *brightness* dan *exposure*. Dengan tujuan dapat diadopsi sebagai alat bantu komunikasi bagi penyandang tunarungu dan tunawicara. Selain itu, sistem ini juga dapat membantu masyarakat umum berkomunikasi secara langsung dengan penyandang disabilitas tersebut.