BAB 2
LANDASAN TEORI

Dalam deteksi penyakit kulit, terdapat beberapa teori yang dipahami seperti algoritma CNN, VGG19, dan teori-teori lainnya.

2.1 Penyakit Kulit

Penyakit kulit merupakan kondisi multifaktorial yang etiologinya melibatkan infeksi oleh mikroorganisme seperti bakteri, virus, jamur, reaksi alergi, dan gangguan autoimun, sehingga menghasilkan berbagai manifestasi klinis seperti ruam, gatal, dan inflamasi yang dapat menurunkan kualitas hidup penderitanya. Selain faktor patogen, variabel lingkungan serta predisposisi genetik juga turut berkontribusi pada kerentanan individu terhadap gangguan kulit. Penelitian terkini menekankan peran penting interaksi antara *mikrobioma* kulit dan sistem imun dalam menjaga integritas barier kulit, di mana *disbiosis mikrobioma* dapat memicu peradangan kronis dan memperburuk kondisi seperti akne dan eksim [15, 16].

2.1.1 Monkeypox

Disebabkan oleh virus *monkeypox* (MPXV), genus *orthopoxvirus*. Berbeda dengan variola (cacar), MPXV memiliki inang alami di hewan pengerat dan bersifat zoonosis. Penularan antar manusia terjadi melalui kontak langsung dengan lesi kulit, *droplet* pernapasan, atau benda yang terkontaminasi [17]. Wabah 2022 menunjukkan transmisi non-endemik melalui kontak intim, berbeda dengan pola historis di Afrika [18].

Ruam pada infeksi MPXV menunjukkan pola persebaran *centrifugal*, dimulai dari wajah dan meluas ke ekstremitas seperti tangan dan kaki. Perkembangan lesi berlangsung secara sinkron, melalui tahapan papula, *vesikel, pustul*, hingga akhirnya membentuk *krusta*:

- Papula: Tahap awal lesi kulit muncul sebagai benjolan kecil, biasanya berwarna merah atau coklat. Papula ini tidak berisi cairan dan terasa keras saat disentuh. Pada tahap ini, lesi mulai muncul di area wajah dan kemudian menyebar ke bagian tubuh lainnya [19, 17].

- *Vesikel*: Setelah beberapa hari, papula berkembang menjadi *vesikel*, yaitu lesi yang berisi cairan jernih. *Vesikel* ini tampak seperti gelembung kecil di permukaan kulit dan dapat berukuran bervariasi. Pada tahap ini, lesi mulai terlihat lebih menonjol dan dapat terasa gatal atau nyeri [19, 17].
- *Pustul*: *Vesikel* kemudian berubah menjadi *pustul*, yaitu lesi yang berisi nanah. *Pustul* biasanya berwarna putih atau kuning dan memiliki dasar yang kemerahan. Pada tahap ini, lesi menjadi lebih besar dan lebih jelas terlihat, serta dapat menyebabkan ketidaknyamanan [19, 17].
- *Krusta*: Setelah beberapa hari, *pustul* akan pecah dan mengering, membentuk kerak atau *krusta*. *Krusta* ini adalah lapisan keras yang menutupi area yang sebelumnya terinfeksi. Proses penyembuhan dimulai pada tahap ini, dan *krusta* akan jatuh dengan sendirinya setelah beberapa waktu, meninggalkan kulit yang mungkin masih kemerahan atau sedikit teriritasi [19, 17].

Lesi sering muncul di area genital atau *perianal* pada wabah 2022, berbeda dengan pola klasik [19]. *Limfadenopati* atau pembengkakan kelenjar getah bening adalah pembeda utama dari cacar air dan campak [20].

2.1.2 Chickenpox

Disebabkan oleh virus varicella-zoster (VZV), genus Herpesvirus. Menular melalui aerosol atau kontak langsung dengan lesi. Masa inkubasi 10–21 hari. Reaktivasi VZV laten menyebabkan herpes zoster (shingles) [21].

Ruam vesikular gatal muncul dalam gelombang, dengan lesi berada pada tahap makula, *vesikel*, dan *krusta*. Pola distribusinya bersifat sentripetal, dimulai dari badan. Lesi pada kulit kepala merupakan pembeda khas dari infeksi MPXV [22].

2.1.3 Measles

Virus campak (Morbillivirus) memiliki tingkat penularan tinggi (R0=12–18) melalui aerosol. R0, atau angka reproduksi dasar, adalah ukuran yang digunakan dalam epidemiologi untuk menggambarkan seberapa menular suatu penyakit. Angka ini menunjukkan rata-rata jumlah orang yang dapat terinfeksi oleh satu orang yang terinfeksi dalam populasi yang sepenuhnya rentan (belum terinfeksi dan belum divaksinasi). Ketika R0 bernilai antara 12 hingga 18, untuk virus

campak (Morbillivirus), ini menunjukkan bahwa satu orang yang terinfeksi dapat menularkan virus tersebut kepada 12 hingga 18 orang lainnya. Ini adalah angka yang sangat tinggi, yang menjelaskan mengapa campak sangat menular. Satu kasus dapat menginfeksi 90% individu non-imun dalam kontak dekat [23]. Wabah sering terjadi di populasi dengan cakupan vaksinasi kurang dari 95% [24].

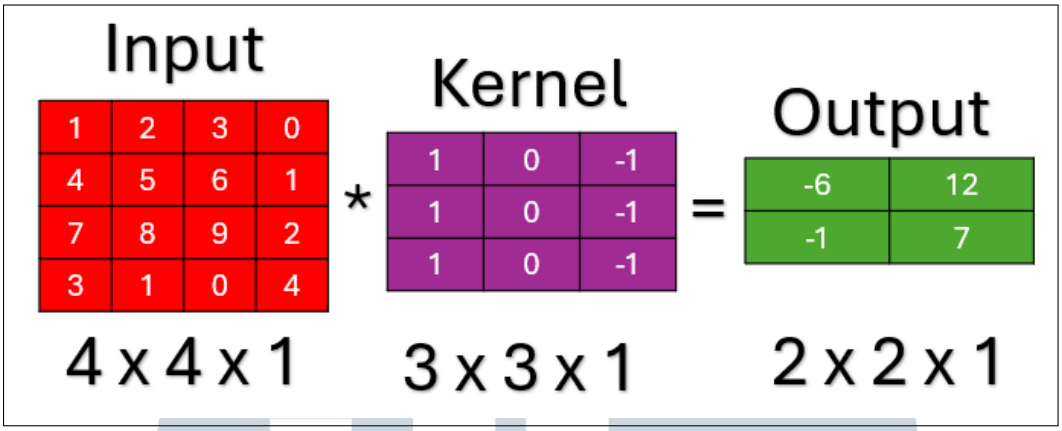
Ruam makulopapular biasanya muncul pertama kali di belakang telinga, lalu menyebar ke seluruh tubuh dengan pola menyebar keluar (sentrifugal). Ruam ini disertai gejala awal seperti demam tinggi, batuk, dan mata merah (konjungtivitis). Ciri khas yang membedakan ruam ini dari yang lain adalah adanya bintik putih kecil di bagian dalam pipi (mukosa bukal) yang dikenal sebagai *koplik spots*, yang merupakan tanda khas penyakit ini [25].

2.2 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salat satu algoritma yang ada di dalam *deep learning*. CNN merupakan suatu lapisan yang memiliki susunan neuron 3D (lebar, tinggi, dan kedalaman). Lebar dan tinggi merupakan ukuran lapisan sedangkan kedalaman mengacu pada jumlah lapisan [26]. CNN digunakan untuk mengklasifikasikan video ataupun gambar [27].

2.2.1 Convolution Layer

Convolution Layer merupakan komponen dasar dari struktur algoritma CNN yang berfungsi untuk melakukan ekstraksi fitur pada data *input* [28]. Dalam proses ekstraksi fitur data *input* akan dibagi menjadi beberapa bagian kecil yang terdiri dari 3×3 atau 5×5 piksel. *Convolution Layer* memproses data yang telah dipecah dengan menggeser lapisan konvolusi di atas data tersebut dan akan menghasilkan peta fitur [29]. Berikut adalah ilustrasi proses konvolusi.

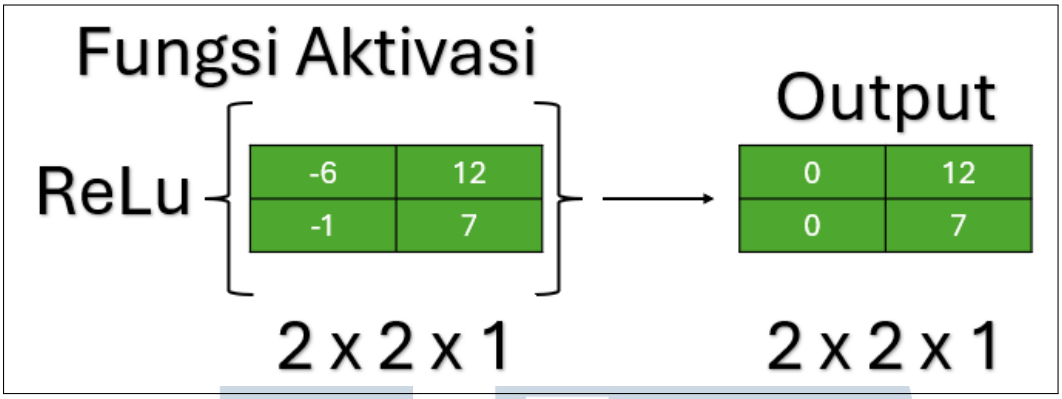


Gambar 2.1. Proses operasi konvolusi

Gambar 2.1 menunjukkan bagaimana operasi konvolusi bekerja. Matriks *input* berukuran 4×4 (merah) dikenai sebuah *filter* atau *kernel* berukuran 3×3 (ungu). *Kernel* ini akan bergeser (melakukan *stride*) ke seluruh bagian matriks *input*. Pada setiap pergeseran, dilakukan operasi perkalian *element-wise* antara nilai pada *kernel* dan bagian dari matriks *input* yang tumpang tindih dengannya. Hasil perkalian tersebut kemudian dijumlahkan untuk menghasilkan satu nilai tunggal pada matriks *output* atau peta fitur (*feature map*) yang berukuran 2×2 (hijau).

Sebagai contoh, nilai pertama pada peta fitur (-6) dihasilkan dari perhitungan $(1 \times 1) + (2 \times 0) + (3 \times -1) + (4 \times 1) + (5 \times 0) + (6 \times -1) + (7 \times 1) + (8 \times 0) + (9 \times -1) = -6$. Proses ini diulangi dengan menggeser *kernel* ke seluruh area matriks *input* untuk melengkapi semua nilai pada peta fitur.

Setelah peta fitur terbentuk, setiap elemen di dalamnya akan dilewatkan melalui fungsi aktivasi non-linear. Fungsi ini bertujuan untuk menambahkan non-linearitas ke dalam model, sehingga pola yang lebih kompleks dapat dipelajari. Salah satu fungsi aktivasi yang populer adalah *Rectified Linear Unit* (ReLU).



Gambar 2.2. Aplikasi fungsi aktivasi ReLU

Seperti yang diilustrasikan pada Gambar 2.2, fungsi ReLU bekerja dengan mengubah semua nilai negatif pada peta fitur menjadi nol dan mempertahankan nilai non-negatif (positif dan nol). Secara matematis, fungsi ReLU didefinisikan sebagai $f(x) = \max(0,x)$. Dalam contoh ini, nilai -6 dan -1 menjadi 0, sementara nilai lainnya tetap karena tidak negatif. Peta fitur yang telah diaktivasi ini kemudian menjadi *input* untuk *layer* selanjutnya dalam arsitektur CNN.

2.2.2 Pooling Layer

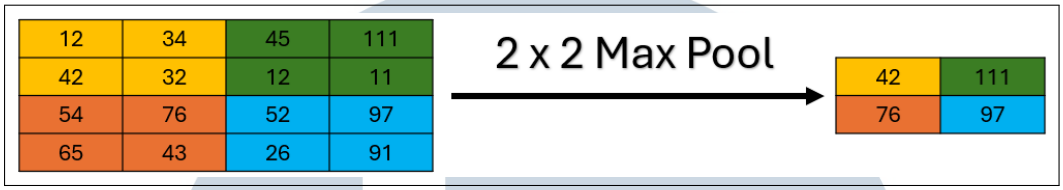
Pooling layer adalah salah satu komponen dari struktur algoritma CNN yang berfungsi untuk mengurangi ukuran data dan mengurangi kompleksitas komputasi. *Pooling layer* bertujuan untuk menurunkan sampel peta fitur yang dihasilkan oleh *convolution layer* namun tetap menyimpan informasi yang penting sehingga dapat meningkatkan efisiensi dari komputasi [29]. Terdapat beberapa jenis *pooling layer*, yaitu *max pooling* dan *average pooling*.

A Max Pooling

Max pooling bekerja dengan cara membagi *feature map* ke dalam beberapa wilayah kecil. *Max pooling* memilih nilai maksimum dari setiap wilayah yang telah dibagi. Dalam setiap blok, dilakukan pencarian terhadap elemen dengan nilai tertinggi, yang kemudian dipilih sebagai representasi dari blok tersebut.

Nilai maksimum yang diperoleh dari setiap wilayah tersebut disusun kembali untuk membentuk *feature map* baru dengan ukuran yang lebih kecil. Dengan demikian, proses ini memastikan bahwa setiap wilayah pada *feature map* hanya diwakili oleh nilai tertingginya, yang kemudian digunakan dalam proses

selanjutnya pada jaringan [30]. Gambar 2.3 merupakan ilustrasi cara kerja *max pooling*.

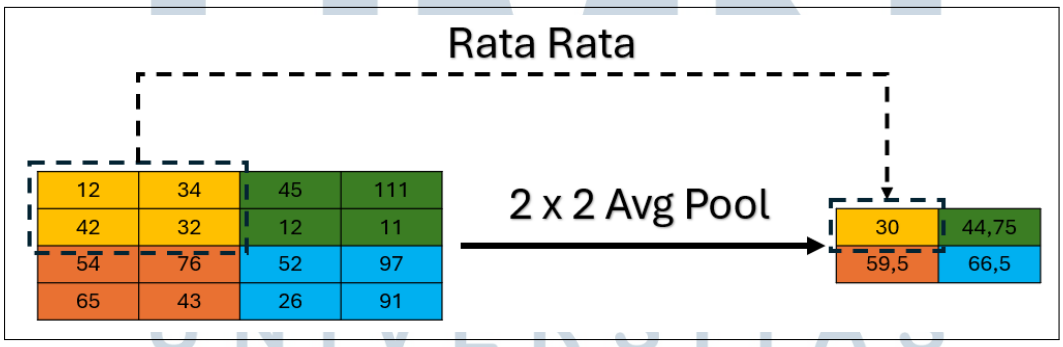


Gambar 2.3. *Max pooling*

B Average Pooling

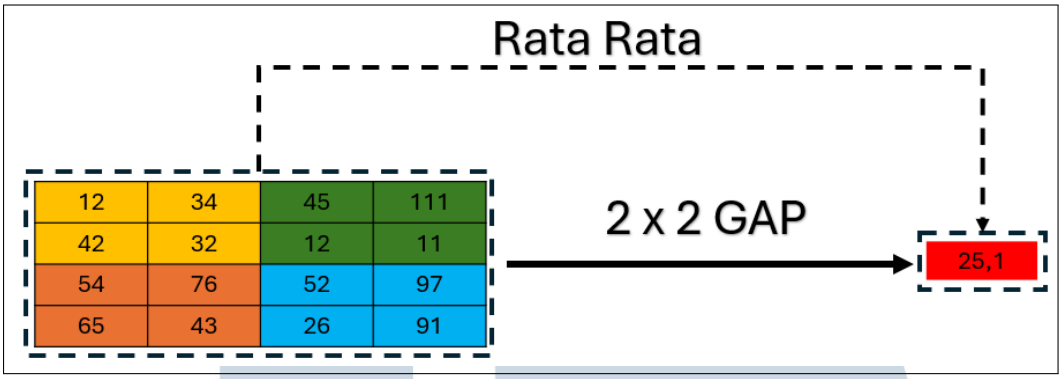
Average pooling bekerja dengan cara membagi *feature map* menjadi beberapa wilayah kecil atau blok, kemudian menghitung nilai rata-rata dari semua elemen yang ada di masing-masing wilayah tersebut. Setiap blok pada *feature map* diolah secara independen, dan hasil perhitungan rata-rata ini menggantikan blok asli untuk membentuk representasi data yang baru.

Selanjutnya, nilai rata-rata dari masing-masing wilayah tersebut disusun kembali untuk menghasilkan *feature map* dengan dimensi yang lebih kecil. Proses ini memungkinkan jaringan untuk mengurangi jumlah data yang diproses di lapisan-lapisan selanjutnya tanpa menghilangkan informasi dasar yang terkandung dalam setiap wilayah [30]. Gambar 2.4 merupakan ilustrasi cara kerja *average pooling*.



Gambar 2.4. *Average pooling*

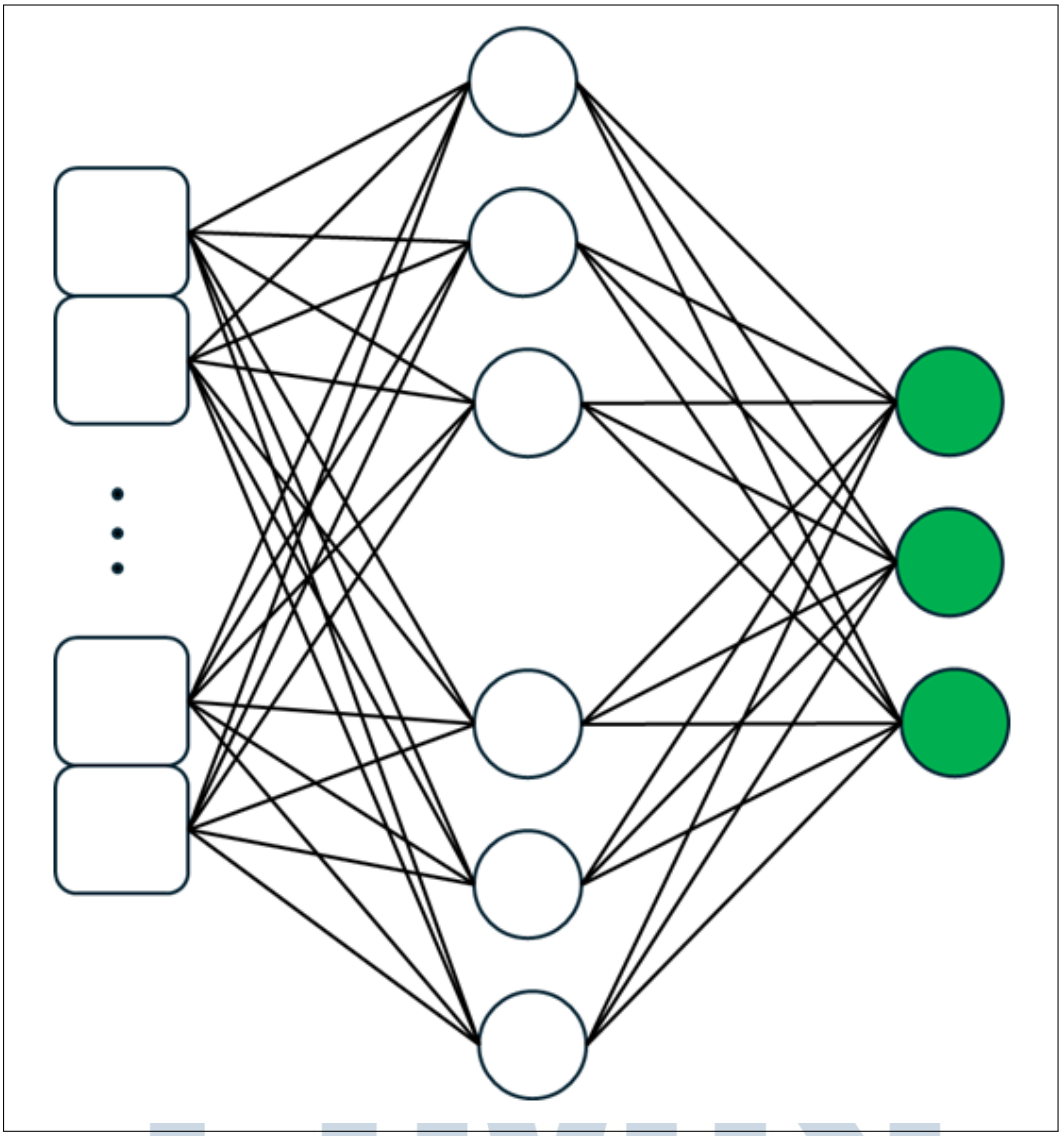
Sedangkan, *average pooling* yang mengambil rata-rata dari seluruh elemen di sebuah *feature map* untuk setiap *channel* disebut dengan *Global Average Pooling* (GAP). Gambar 2.5 merupakan ilustrasi cara kerja *global average pooling*.



Gambar 2.5. *Global average pooling*

2.2.3 Fully Connected Layer

Setelah melalui beberapa lapisan konvolusi dan *pooling*, tahap ini menghubungkan setiap neuron dengan semua neuron pada lapisan sebelumnya dan biasanya dipakai untuk tugas klasifikasi. Proses ini diletakkan di akhir jaringan karena semua neuron harus diubah menjadi data satu dimensi sehingga tidak dapat dikembalikan ke bentuk data awal. Gambar 2.6 menggambarkan ilustrasi *fully connected layer*.



Gambar 2.6. *Fully connected layer*

Kotak-kotak di sebelah kiri mewakili input ke jaringan, di mana masing-masing kotak merepresentasikan satu fitur dari data masukan. Lingkaran di tengah menunjukkan neuron pada lapisan tersembunyi, yang menerima informasi dari seluruh neuron di lapisan input. Di sini, setiap neuron memproses data dengan menerapkan bobot dan bias, lalu mengirimkan hasilnya melalui fungsi aktivasi. Lingkaran hijau di sebelah kanan mewakili neuron *output*, yang menghasilkan keluaran akhir dari jaringan saraf. Hasil yang dihasilkan oleh setiap neuron *output* bergantung pada nilai-nilai yang dihitung oleh neuron-neuron di lapisan tersembunyi. Sementara itu, garis-garis penghubung antar neuron adalah koneksi yang disebut bobot, yang menunjukkan bahwa setiap neuron di lapisan sebelumnya

terhubung ke neuron di lapisan berikutnya dan berfungsi untuk menyesuaikan pengaruh masing-masing input.

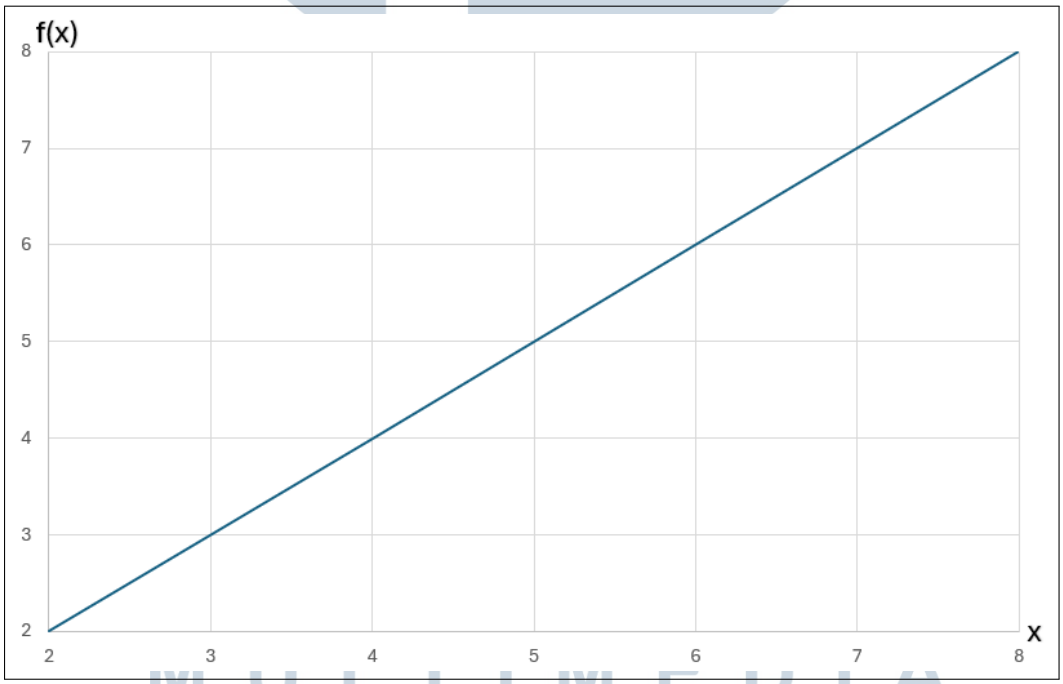
2.3 Activation Function

Terdapat beberapa jenis fungsi aktivasi yaitu, Linear type, Sigmoid, *Hyperbolic Tangent* (Tanh), *Rectified Linear Units* (ReLu), *Thresholded ReLu* (T-ReLu), Swish, Mish, dan Softmax.

Fungsi aktivasi akhir dari lapisan terakhir hanyalah fungsi linear dari lapisan pertama input, dan dapat digunakan di lapisan *output* [31]. Aktivasi *linear* dapat dihitung menggunakan Rumus 2.1.

$$Y = x; -\infty \; sampai \; +\infty \tag{2.1}$$

Gambar 2.7 merupakan ilustrasi plot *linear activation*.



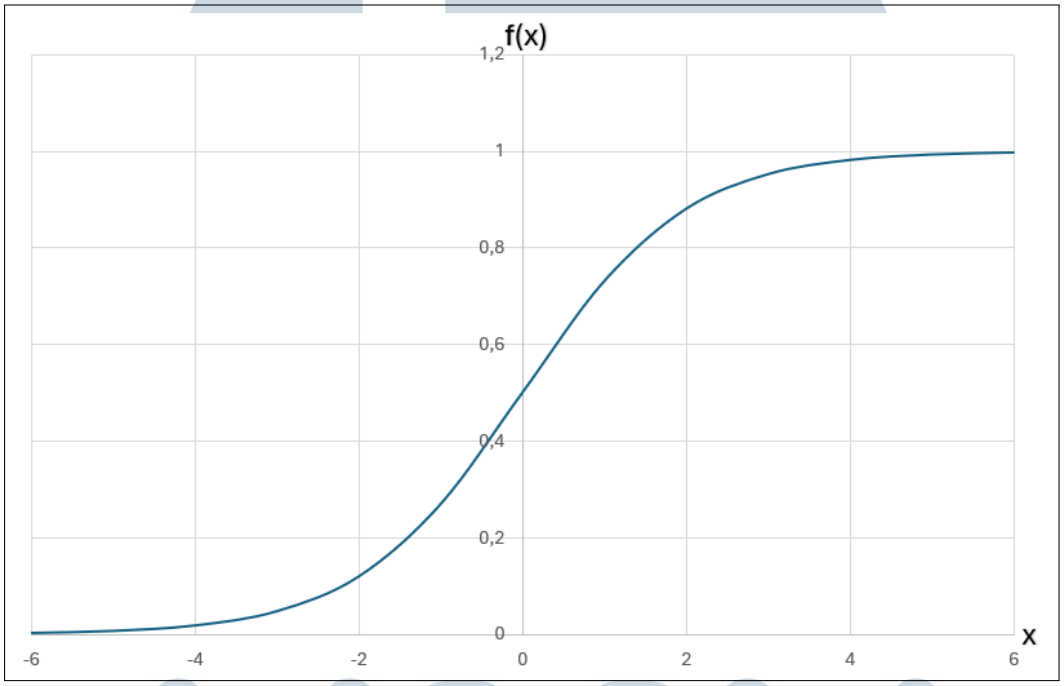
Gambar 2.7. *Linear activation* plot

Aktivasi sigmoid merupakan fungsi yang memetakan input ke rentang (0, 1). Perubahan kecil pada input akan menghasilkan perubahan besar pada *output*. Untuk mengubah *output* menjadi skor yang dapat diprediksi, lapisan ini ditempatkan di

akhir model [31]. Aktivasi sigmoid dapat dihitung menggunakan Rumus 2.2.

$$f(x) = \frac{1}{1+e^{-x}} \tag{2.2}$$

Gambar 2.8 merupakan ilustrasi plot sigmoid.

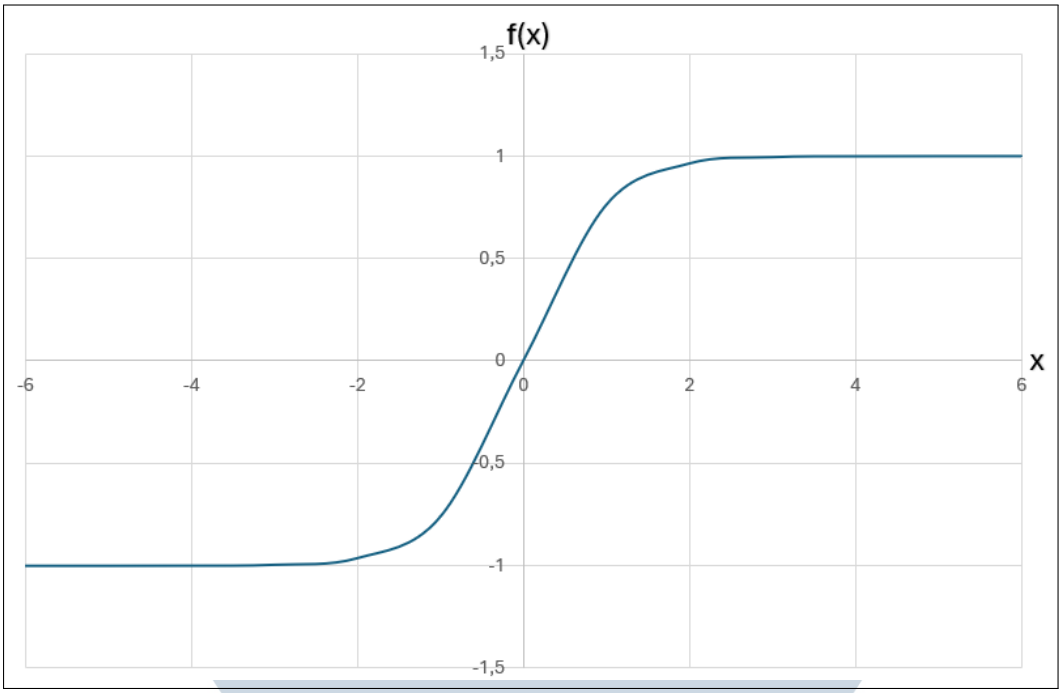


Gambar 2.8. Sigmoid plot

Aktivasi Tanh merupakan fungsi yang memetakan input ke rentang (-1, 1) dan sering digunakan di lapisan tersembunyi. Digunakan sebagai alternatif fungsi sigmoid jika *output* bukan nol dan satu. Jika jumlah bobot input sangat besar, maka gradien fungsi menjadi sangat kecil dan mendekati nol. Menggunakan aktivasi tanh dalam kondisi seperti ini memungkinkan memiliki masalah gradien yang hilang [31]. Aktivasi tanh dapat dihitung menggunakan Rumus 2.3.

$$f(x) = Tanh(x = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}) \tag{2.3}$$

Gambar 2.9 merupakan ilustrasi plot tanh.

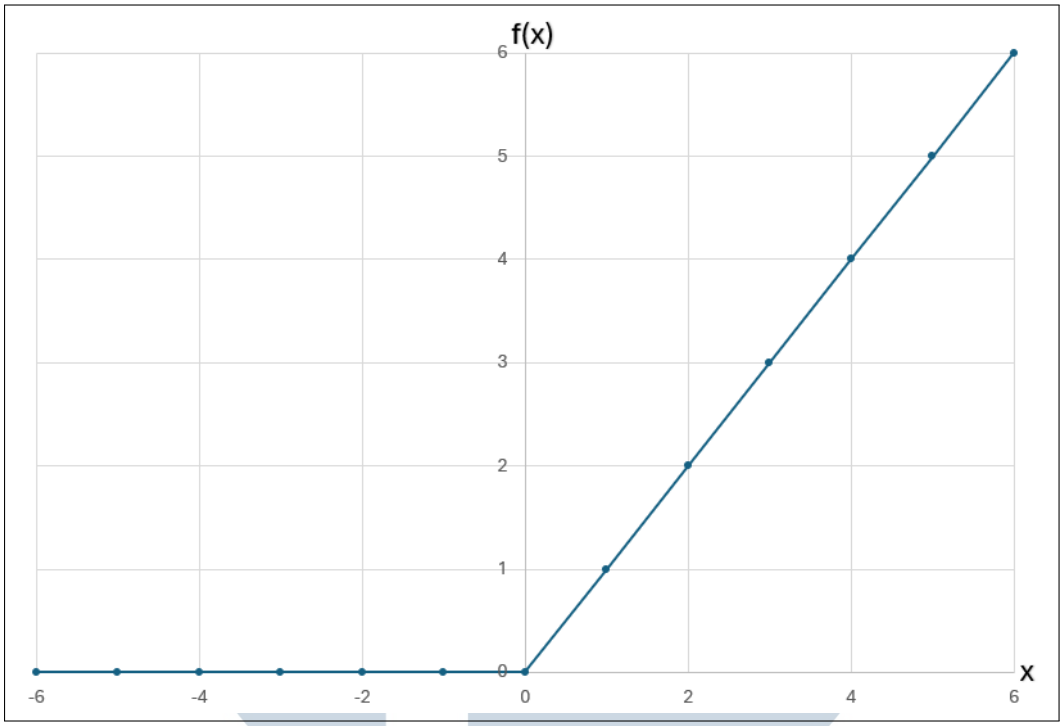


Gambar 2.9. Tanh plot

Aktivasi ReLu merupakan fungsi yang memetakan input ke rentang (0, + ∞), aktivasi ReLu diimplementasikan dalam lapisan tersembunyi pada model. Secara komputasi jauh lebih cepat daripada tanh dan sigmoid dan memecahkan masalah gradien yang menghilang. Namun aktivasi ReLu tidak menghitung eksponensial dan pembagian [31]. Aktivasi ReLu dapat dihitung menggunakan Rumus 2.4.

$$f(x) = max(0,x) \tag{2.4}$$

Gambar 2.10 merupakan ilustrasi plot ReLu.

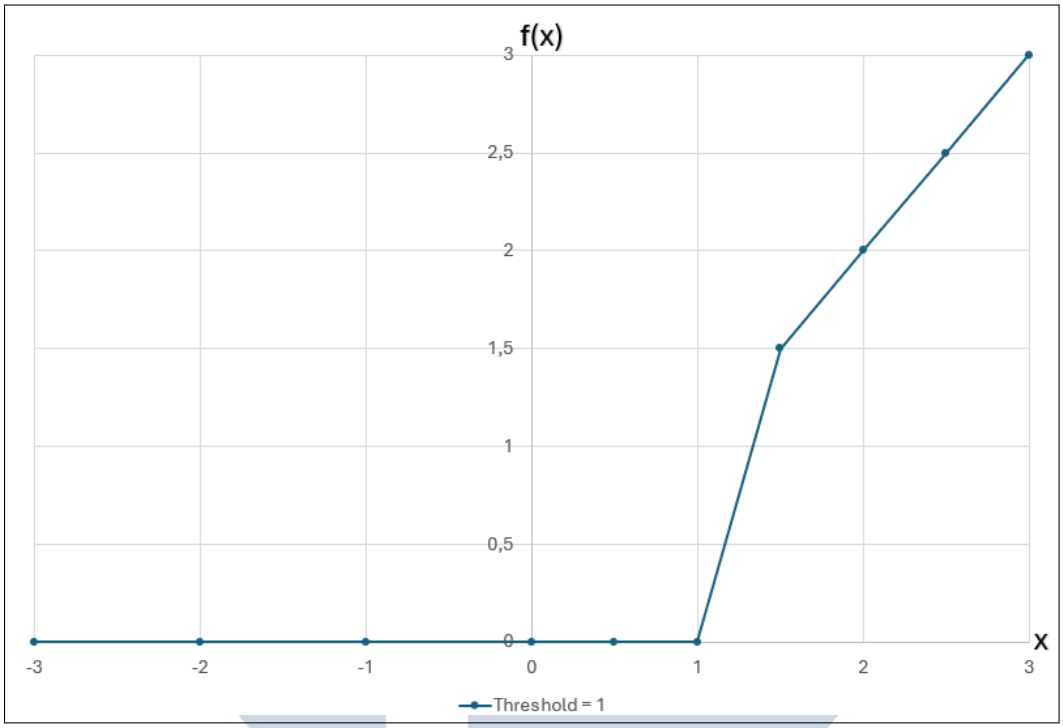


Gambar 2.10. ReLu plot

Aktivasi T-ReLu merupakan fungsi varian dari ReLU yang memperkenalkan ambang batas (*threshold*). Nilai input yang lebih kecil dari *threshold* akan menjadi nol. Aktivasi T-ReLu dihitung menggunakan Rumus 2.5.

$$f(x) = \begin{cases} x, & \text{jika } x > \text{threshold} \\ 0, & \text{jika } x \leq \text{threshold} \end{cases} \tag{2.5}$$

Gambar 2.11 merupakan ilustrasi plot T-ReLu.

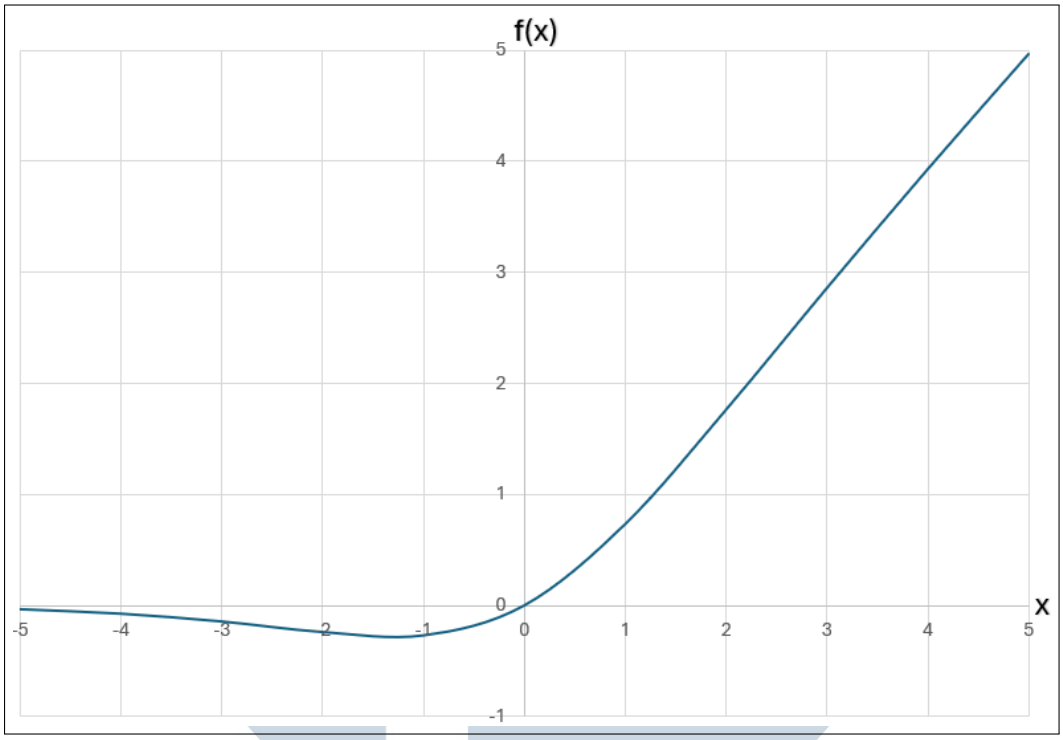


Gambar 2.11. T-ReLu plot

Aktivasi swish merupakan fungsi yang memetakan input ke rentang (- ∞ , + ∞). Aktivasi swish menangani masalah gradien yang menghilang dan membantu dalam menormalkan *output*. Namun secara komputasi, aktivasi swish lebih mahal daripada Sigmoid [31]. Aktivasi swish dapat dihitung menggunakan Rumus 2.6.

$$f(x) = x \cdot \frac{1}{1+e^{-x}} \tag{2.6}$$

Gambar 2.12 merupakan ilustrasi plot swish.

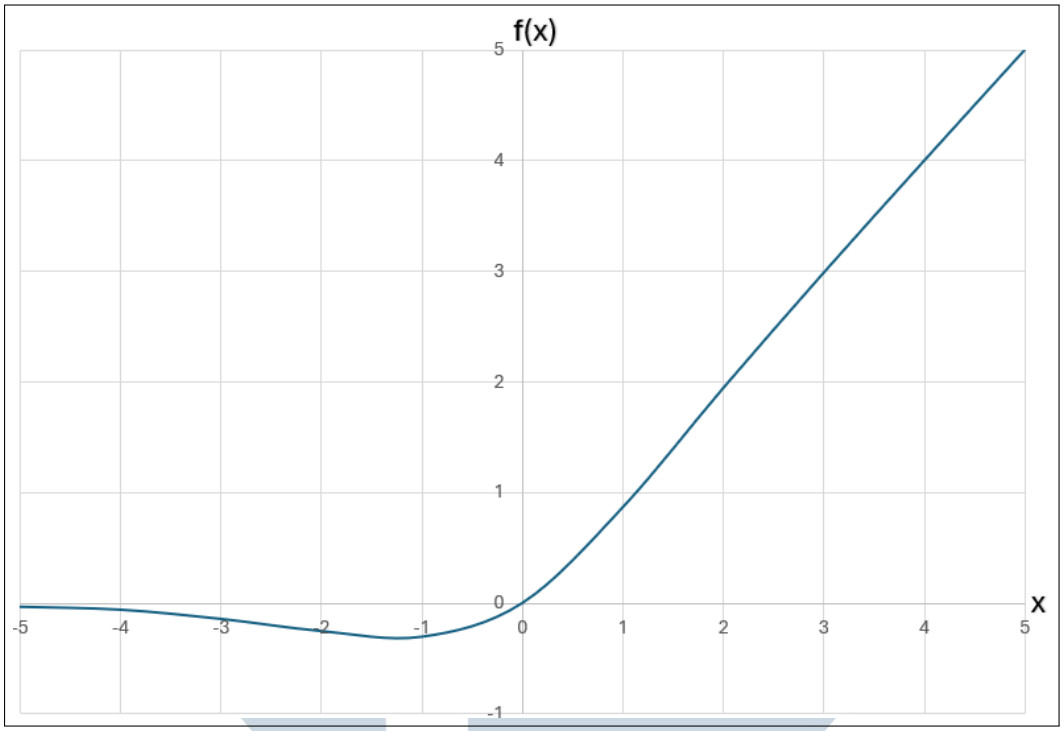


Gambar 2.12. Swish plot

Mish adalah fungsi aktivasi yang *continuously differentiable* (fungsi yang memiliki turunan di semua titik dan turunannya juga kontinu) dan *nonmonotonic* (fungsi yang tidak selalu naik atau turun, tetapi bisa naik lalu turun atau sebaliknya). Namun secara komputasi, aktivasi mish lebih mahal daripada ReLu [31]. Aktivasi mish memetakan input ke rentang (- ∞ , + ∞) dan dapat dihitung menggunakan Rumus 2.7.

$$f(x) = x \cdot tanh(ln(1 + e^x)) \tag{2.7}$$

Gambar 2.13 merupakan ilustrasi plot mish.

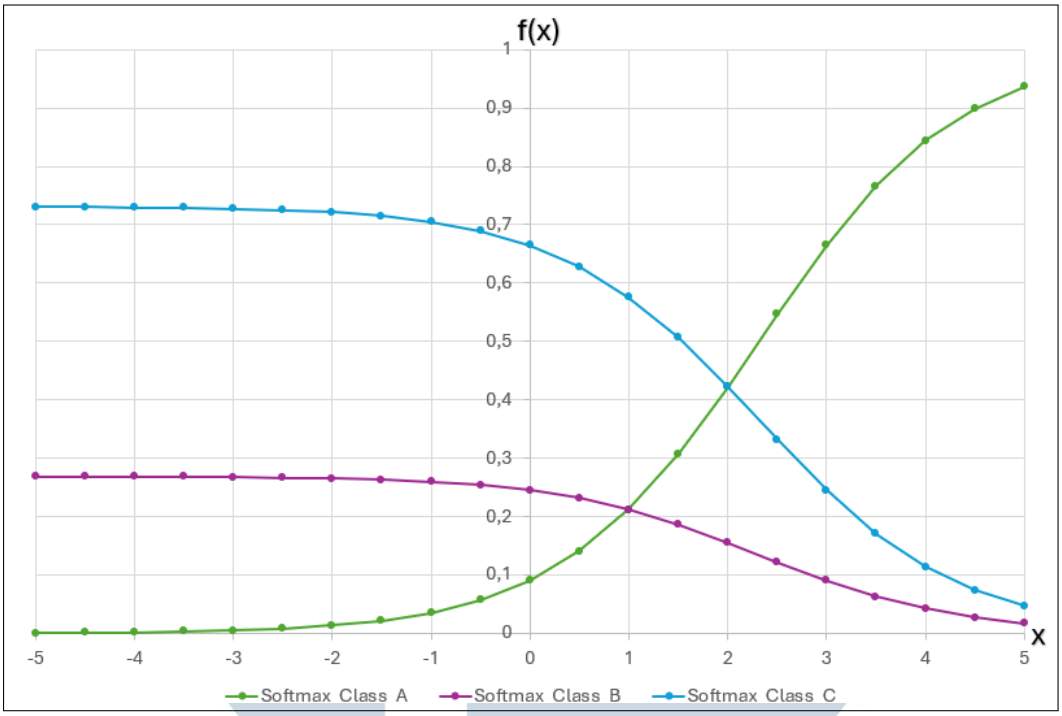


Gambar 2.13. Mish plot

Aktivasi softmax adalah bentuk lain dari algoritma regresi logistik yang dapat digunakan untuk mengklasifikasikan lebih dari dua kelas. Softmax memberikan hasil yang lebih intuitif dan memiliki interpretasi probabilistik yang lebih baik daripada algoritma klasifikasi lainnya. Softmax memungkinkan untuk menghitung probabilitas untuk semua label. Dari label yang ada, sebuah nilai nyata diambil dan mengubahnya menjadi vektor dengan nilai antara nol dan satu yang jumlahnya akan bernilai satu [32]. Aktivasi softmax dapat dihitung menggunakan Rumus 2.8.

$$S_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \tag{2.8}$$

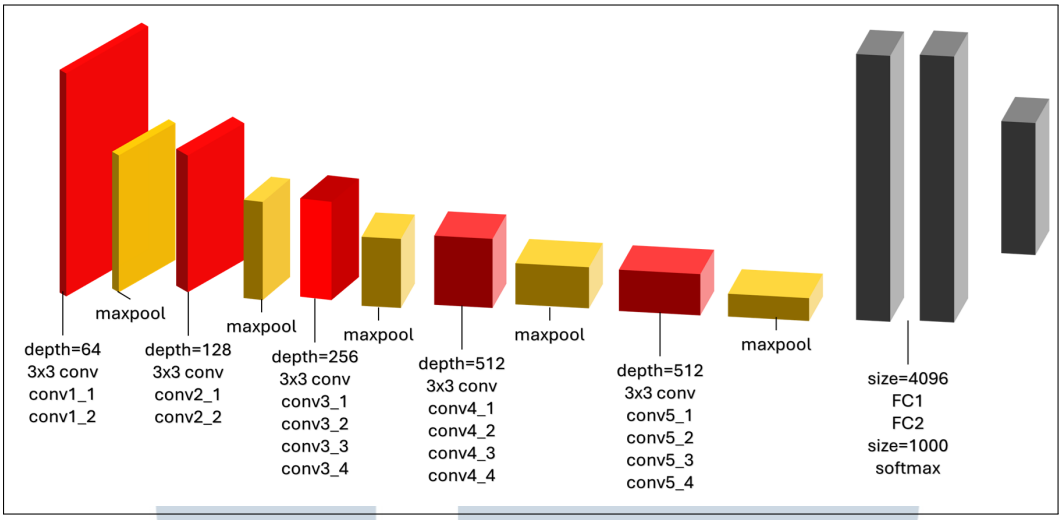
Gambar 2.14 merupakan ilustrasi plot softmax.



Gambar 2.14. Softmax plot

2.4 VGG19

VGG merupakan arsitektur jaringan CNN, dan VGG19 adalah salah satu variannya. Jaringan *deep learning* ini terdiri dari 19 lapisan, yang mencakup 16 *convolution layer* serta 3 *fully connected layer*. *Convolution layer* berperan dalam mengekstraksi fitur dari gambar yang dimasukkan, sementara *fully connected layer* bertugas mengklasifikasikan gambar berdasarkan fitur yang telah diperoleh. Selain itu, lapisan *pooling* digunakan untuk mengurangi jumlah fitur guna mencegah *overfitting* [33], seperti yang ditunjukkan dalam Gambar 2.15.



Gambar 2.15. Visualisasi arsitektur VGG19
Sumber: [34]

Karakteristik utama dari arsitektur VGG adalah penggunaan filter konvolusi berukuran sangat kecil, yaitu 3x3, pada seluruh lapisannya. Penggunaan filter kecil yang ditumpuk secara mendalam dapat menghasilkan medan reseptif (receptive field) yang setara dengan filter yang lebih besar, namun dengan jumlah parameter yang lebih sedikit dan mampu menangkap lebih banyak fitur non-linear. Setiap lapisan konvolusional diikuti oleh fungsi aktivasi *Rectified Linear Unit* (ReLU) untuk memperkenalkan non-linearitas ke dalam model.

Secara rinci, alur pemrosesan data pada arsitektur VGG19 dapat diuraikan menjadi dua bagian utama: bagian ekstraksi fitur dan bagian klasifikasi.

1. Bagian ekstraksi fitur: Bagian ini terdiri dari lima blok konvolusional, di mana setiap blok diakhiri dengan sebuah lapisan *max-pooling* (dengan filter 2x2 dan *stride* 2) untuk mereduksi dimensi spasial.

    - Blok 1 dan 2: Masing-masing terdiri dari dua lapisan konvolusional. Blok pertama menggunakan 64 filter, sedangkan blok kedua menggunakan 128 filter.
    - Blok 3, 4, dan 5: Masing-masing terdiri dari empat lapisan konvolusional. Blok ketiga menggunakan 256 filter, sedangkan blok keempat dan kelima menggunakan 512 filter. Seiring dengan bertambahnya kedalaman jaringan, jumlah filter digandakan untuk menangkap fitur yang lebih kompleks, sementara dimensi spasial dari peta fitur terus berkurang.

2. Bagian Klasifikasi: Setelah melewati blok konvolusional kelima, peta fitur yang dihasilkan akan diratakan (*flattened*) menjadi sebuah vektor tunggal. Vektor ini kemudian menjadi masukan bagi tiga lapisan *fully-connected*.

    - Dua lapisan *fully-connected* pertama masing-masing memiliki 4096 neuron.
    - Lapisan *fully-connected* ketiga (lapisan output) memiliki jumlah neuron yang sesuai dengan jumlah kelas target, misalnya 1000 neuron untuk klasifikasi pada dataset ImageNet.
    - Fungsi aktivasi Softmax diterapkan pada lapisan terakhir untuk menghasilkan distribusi probabilitas dari setiap kelas, sehingga model dapat memberikan prediksi akhir.

2.5 Evaluasi

Evaluasi dilakukan untuk mengukur tingkat keberhasilan model yang telah dikembangkan. Pengujian model dilakukan dengan membandingkan data latih dan data uji menggunakan beberapa parameter, yaitu akurasi, presisi, *recall*, dan *F1-score*, yang dihitung berdasarkan *confusion matrix*.

Confusion matrix banyak digunakan dalam pembelajaran mesin untuk tugas *supervised classification* atau untuk menganalisis kinerja model klasifikasi. Matriks ini berbentuk persegi dengan baris yang mewakili kelas sebenarnya dari data, sementara kolom menunjukkan kelas yang diprediksi. Dalam klasifikasi biner, *confusion matrix* memiliki ukuran 2×2 dan terdiri dari empat komponen utama yaitu *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN). Untuk klasifikasi *multiclass, confusion matrix* dengan k kelas akan memiliki *confusion matrix* $k \times k$ [35].

Tabel 2.1. *Confusion matrix* 2×2

Actual	*Predict*	
	Positive	*Negative*
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

Nilai presisi (P) adalah rasio observasi positif yang diprediksi secara akurat

terhadap total observasi positif yang diprediksi [36]. Nilai presisi dapat dihitung menggunakan persamaan 3.2.

$$P = \frac{TP}{TP+FP} \times 100\% \tag{2.9}$$

Nilai *recall* (R) adalah rasio pengamatan positif yang diprediksi secara akurat terhadap jumlah semua sampel yang relevan [36]. Nilai *recall* dapat dihitung menggunakan persamaan 2.10.

$$R = \frac{TP}{TP+FN} \times 100\% \tag{2.10}$$

F1-score adalah rata-rata tertimbang dari presisi dan *recall* dan metode yang digunakan untuk mengukur kinerja model [36]. *F1-score* dapat dihitung menggunakan persamaan 2.11.

$$F_1 score = 2 \times \frac{P \times R}{P+R} \tag{2.11}$$

Nilai akurasi adalah persentase dari data yang diklasifikasikan dengan benar. Nilai akurasi merupakan ukuran kinerja yang paling intuitif dan merupakan rasio dari observasi yang diprediksi dengan benar terhadap total pengamatan [36]. Akurasi dapat dihitung menggunakan persamaan 2.12.

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \tag{2.12}$$