

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Kanker Kulit

Kanker kulit merupakan penyakit yang berkembang dari sel-sel kulit yang tidak terkendali dan bisa terbentuk di berbagai macam lapisan kulit. Ada dua jenis utama kanker kulit yang ditetapkan menjadi kanker kulit yaitu *non-melanoma (benign)* dan kanker kulit *melanoma (malignant)* [18].

A. Kanker kulit *non-melanoma*

Kanker kulit *non-melanoma* (NMSC) atau kanker kulit *keratinosit* (KSC) merupakan kanker kulit yang paling umum. NMSC dipecah menjadi dua jenis utama:

- 1) *Karsinoma Sel Basal (Basal Cell Carcinoma, BCC)* termasuk golongan kanker kulit yang paling umum. BCC biasanya tampak sebagai benjolan atau bercak merah pada kulit yang terkena sinar matahari, seperti wajah dan leher. Paparan sinar ultraviolet (UV) dari matahari merupakan penyebab risiko terbesar. Meskipun BCC memiliki tingkat kematian yang rendah, penanganan yang tepat tetap diperlukan untuk mencegah penyebaran dan kerusakan jaringan di sekitarnya.
- 2) *Karsinoma Sel Skuamosa (Squamous Cell Carcinoma, SCC)* termasuk golongan kanker kulit kedua yang paling umum. SCC terjadi pada sel keratinosit di lapisan luar/atas epidermis dan sering tampak sebagai bercak bersisik berupa benjolan merah yang keras. Paparan sinar UV yang berlebihan juga merupakan penyebab risiko utama untuk SCC.

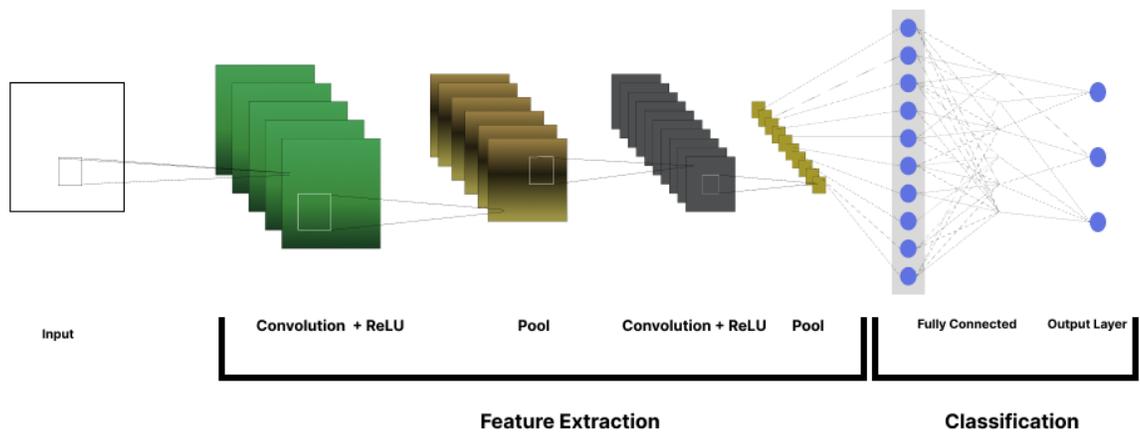
B. Kanker kulit *Melanoma*

Melanoma adalah tumor ganas agresif yang datang dari melanosit. Melanosit terletak di lapisan basal epidermis. Penumpukan perubahan genetik yang

mengaktifkan onkogen, menonaktifkan gen penekan tumor, dan menghambat perbaikan DNA ketika organisme terkena radiasi ultraviolet. Mekanisme ini dapat menyebabkan pertumbuhan melanosit yang tidak terkendali dan berakibat menjadi sangat berbahaya [18].

2.1.2 Convolutional Neural Networks (CNN)

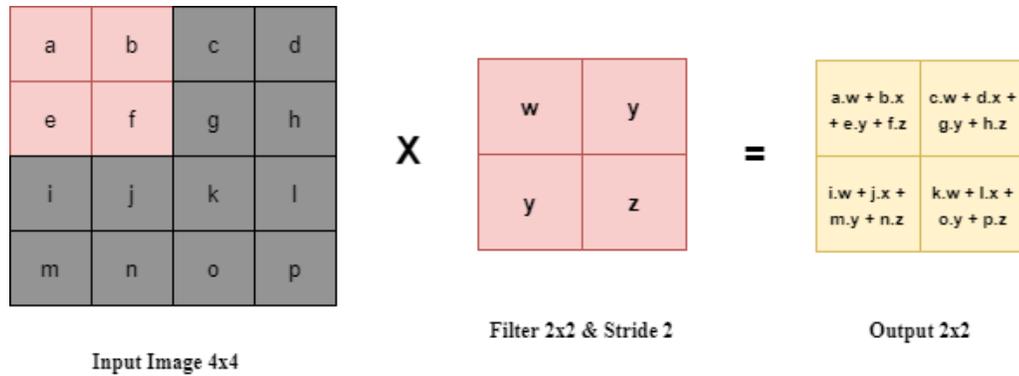
Convolutional Neural Networks (CNN) merupakan bagian dari *deep learning* yang digunakan terutama dalam analisis citra. CNN memiliki arsitektur yang unik dan efisien dalam menangani tugas-tugas yang berkaitan dengan pengenalan pola visual, seperti klasifikasi gambar, deteksi objek, dan segmentasi gambar. Keunggulan utama CNN adalah kemampuannya untuk secara otomatis belajar dan mengekstraksi fitur dari gambar mentah tanpa memerlukan teknik ekstraksi fitur manual yang rumit [19]. CNN terdiri dari dua komponen utama, yaitu *feature extraction layer* dan *classification layer*. Pada *Feature Extraction Layer* ini juga terdapat beberapa bagian lagi, yaitu *Convolutional Layer* dengan aktivasi *ReLU*, dan *Pooling layer* yang berfungsi untuk mengekstraksi ciri-ciri penting dari gambar. Pada *feature extraction layer* terjadi proses *encoding* sebuah input berupa *images* menjadi *features* berupa angka-angka yang merepresentasikan *images* tersebut. Setelah fitur-fitur penting di *encoding*, data ini diteruskan ke *Classification Layer*. *Layer* ini umumnya terdiri atas, *fully connected layer*, dan *output layer*. CNN telah terbukti sangat efektif dalam beragam pengaplikasian pengenalan pola dan telah menjadi standar dalam analisis citra digital, termasuk deteksi kanker kulit [20]. Berikut ilustrasi *layer* pada algoritma CNN dapat dilihat pada gambar 2.1.



Gambar 2.1 *Layers* pada Algoritma CNNs

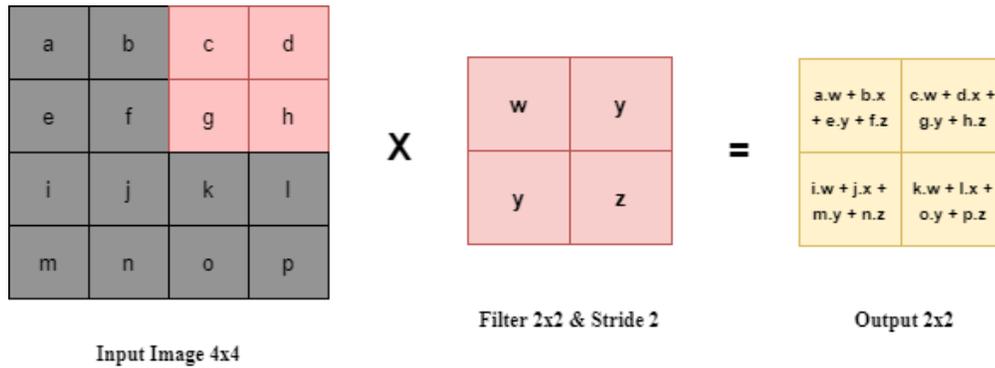
2.1.2.1 *Convolution Layer*

Convolution layer merupakan *layer* utama yang paling penting dalam arsitektur CNN. *Layer* ini bertujuan untuk mengekstraksi fitur-fitur penting dari input citra. Filter dapat digambarkan sebagai nilai matrik, dimana setiap nilai ditetapkan dengan angka acak misalnya antara -1 sampai 1 [21]. *Filter* yang digunakan memiliki panjang, tinggi, dan tebal yang sesuai dengan *channel* pada data input, misalnya *filter* untuk gambar RGB memiliki tiga *channel* dimana setiap nilai kernel ditetapkan dengan angka acak antara -1 sampai 1. *Filter* yang digunakan memiliki panjang, tinggi, dan tebal yang sesuai dengan *channel* pada data input, misalnya *filter* untuk gambar RGB memiliki tiga *channel* [22]. Proses konvolusi melibatkan operasi matematika yang dikenal dengan istilah *dot product* antara nilai piksel citra input dan kernel. Setiap *filter* bergerak melintasi citra input dengan menggunakan *stride* yang telah ditentukan. Pada setiap posisi, dilakukan perhitungan antara nilai-nilai piksel citra dan nilai *filter* menggunakan operasi *dot product*. Hasil dari operasi ini adalah sebuah *feature map* yang menggambarkan fitur yang diekstraksi dari citra input [23]. Berikut ilustrasi sederhana operasi konvolusi satu *channel* (*grayscale*) dengan *stride* 2 pada gambar 2.2.



Gambar 2.2 Ilustrasi Konvolusi Sederhana Stride 2

Konvolusi pertama akan mengambil sub-matriks 2x2 pertama yaitu abef, menghitung *dot product*, dan kemudian *filter* akan melompat 2 langkah (*stride 2*) untuk menghitung *dot product* di submatriks berikutnya yaitu cdgh, berikut ilustrasi pergeseran *stride 2* dapat dilihat pada gambar 2.3.



Gambar 2.3 Ilustrasi Pergeseran Stride 2

Filter kemudian bergerak dua langkah ke bawah untuk perhitungan selanjutnya dan tetap dengan *stride 2* untuk menghitung *dot product* 2 posisi horizontal dan 2 posisi vertikal sehingga menghasilkan *feature map* dengan ukuran dimensi seperti pada persamaan berikut [20].

$$h = \frac{i-k+2p}{s} + 1 \tag{2.1}$$

Dimana:

h adalah dimensi *feature map output*

i adalah dimensi citra input

k adalah ukuran filter

p adalah ukuran padding

s adalah stride

2.1.2.2 Rectified Linear Unit (ReLU)

Rectified Linear Unit (ReLU) merupakan fungsi aktivasi yang paling umum digunakan dalam *Convolutional Neural Networks (CNN)*. Fungsi ini digunakan untuk mengubah semua nilai input menjadi angka positif. Keunggulan *ReLU* ini adalah memerlukan beban komputasi yang sangat minim dibandingkan dengan fungsi aktivasi lainnya. Secara matematis ReLU dapat didefinisikan pada persamaan berikut [21].

$$f(x)_{ReLU} = \max(0, x) \quad (2.2)$$

Dari persamaan tersebut berarti, setiap nilai input x yang lebih besar dari 0 akan tetap dipertahankan, sedangkan yang kurang dari atau sama dengan 0 akan diganti dengan nilai 0.

2.1.2.3 Pooling

Pooling merupakan metode yang digunakan dalam CNN dalam mengurangi dimensi dari peta fitur (*feature map*) yang dihasilkan oleh lapisan konvolusi. *Pooling* menggabungkan serangkaian nilai menjadi nilai yang lebih kecil dengan mempertahankan informasi penting dan menghilangkan informasi yang tidak relevan. *Pooling* juga membantu dalam meningkatkan kemampuan mengenali posisi objek walaupun terdapat pergeseran kecil pada gambar. Metode *pooling* yang ideal diharapkan mampu untuk mengekstrak informasi yang berguna dan membuang detail-detail yang tidak relevan [16].

2.1.2.4 Fully Connected Layer

Fully Connected Layer merupakan bagian terakhir dari arsitektur *convolutional neural networks* yang digunakan untuk klasifikasi. Pada *layer* ini setiap neuron terhubung dengan semua neuron di *layer* sebelumnya. Biasanya, *layer* ini terdiri dari satu atau lebih *fully connected layers* tergantung pada tingkat abstraksi fitur yang diinginkan. Output dari *layer* ini adalah skor probabilitas untuk klasifikasi kedalam kelas-kelas yang tersedia [23]. *Fully connected layers* didefinisikan pada persamaan berikut [24].

$$y_j = b_j + \sum_i w_{ij}x_i \quad (2.3)$$

Dimana:

x adalah input pada *fully connected layer*

w_{ij} adalah bobot matrik berukan $i \times j$ dengan i menunjukkan jumlah fitur dan j menunjukkan jumlah target kelas.

b adalah bias neuron j .

y adalah output dari *fully connected layer*.

Setelah melewati *fully connected layer*, biasanya output tersebut diteruskan ke fungsi aktivasi untuk mengubah nilai output menjadi probabilitas untuk klasifikasi. Salah satu yang paling umum digunakan yaitu *sigmoid* sebagai fungsi aktivasi [25]. Fungsi *sigmoid* digunakan untuk tugas klasifikasi biner, yaitu memetakan skor output menjadi nilai probabilitas dalam rentang $[0,1]$. Fungsi ini didefinisikan dengan persamaan berikut [21].

$$\sigma(y) = \frac{1}{1 + e^{-y}} \quad (2.4)$$

Dimana:

y adalah output dari *fully connected layer* seperti pada persamaan 3.

$\sigma(y)$ adalah probabilitas bahwa data masuk ke kelas tertentu.

2.1.3 Max Pooling

Max pooling merupakan teknik *pooling* yang paling populer dalam CNN, yang digunakan untuk mengurangi dimensi peta fitur sambil mempertahankan informasi penting [16]. Fungsi *max pooling* diberikan dalam persamaan berikut [26].

$$y_j = \max (a_1, a_2, \dots, aR_j) \quad (2.5)$$

Dimana:

y_j : Hasil *max pooling* untuk area *pooling* ke-j.

a_1, a_2, \dots, aR_j : Sekumpulan nilai dalam area *pooling* ke-j pada fitur yang diambil

R_j : Jumlah elemen dari area *pooling*, misalnya 2x2 atau 3x3.

Max pooling ini memilih nilai maksimum dari area tertentu pada peta fitur yang dianggap sebagai fitur paling penting dan mengabaikan nilai-nilai lainnya. Berikut contoh konsep dari *max pooling* [16].



Gambar 2.4 Contoh *Max Pooling*

Pada contoh diatas, proses *max pooling* pada sebuah *feature map* yang awalnya berukuran 4x4 menggunakan ukuran filter 2x2 dengan stride 2 akan mengambil nilai maksimum dari masing-masing blok area, dan mendapatkan *feature map* hasil *max pooling* sebesar 7, 5, 3 dan 4.

2.1.4 Global Average Pooling

Global average pooling merupakan teknik yang digunakan dalam CNN untuk mereduksi dimensi dari peta fitur (*feature map*). Teknik ini menggantikan penggunaan lapisan *fully connected* yang umumnya ditemui dalam arsitektur tradisional CNN. Teknik GAP ini bekerja dengan menghitung rata-rata dari seluruh elemen pada peta fitur. Hal ini menjadikan GAP unggul karena mampu mengurangi risiko *overfitting* dan lebih tangguh terhadap *spatial translations* dalam citra input [27]. Persamaan GAP dapat dirumuskan sebagai berikut [28].

$$F_k = \frac{1}{H \times W} \sum_{x,y} f_k(x,y) \quad (2.6)$$

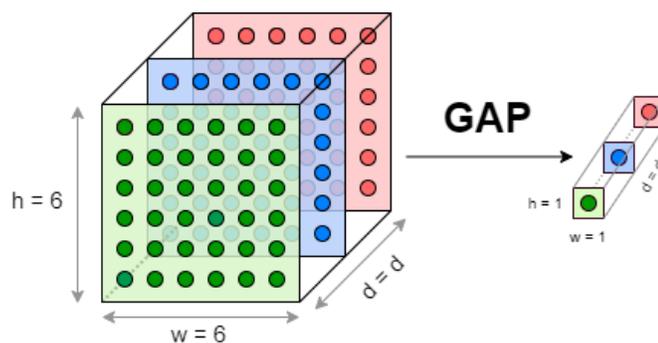
Dimana:

$f_k(x,y)$: Nilai aktivasi atau intensitas pada peta fitur ke-k pada posisi (x,y).

$\sum_{x,y}$: Notasi untuk penjumlahan dari seluruh elemen dalam peta fitur

$H \times W$: Dimensi peta fitur, H yaitu tinggi dan W yaitu lebar dari peta fitur.

Hasil dari GAP itu sendiri adalah sebuah vektor satu dimensi, dengan setiap nilai vektor tersebut mewakili rata-rata aktivasi dari masing-masing peta fitur [28]. Berikut gambar ilustrasi dari GAP.

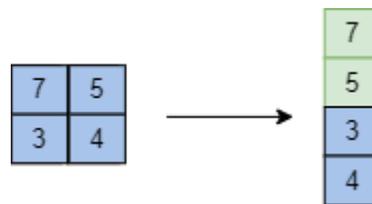


Gambar 2.5 Ilustrasi GAP

Pada ilustrasi gambar tersebut menunjukkan bahwasannya peta fitur awal yang berukuran 6x6 dan kedalaman (d) ada tiga pada contoh gambar tersebut, akan berkurang dari 6x6 menjadi 1x1 dan menghasilkan satu nilai rata-rata pada masing-masing peta fitur.

2.1.5 Flatten

Flatten merupakan salah satu tahap dalam pemrosesan data pada model *convolutional neural network*. Pada tahap ini, informasi spasial dari gambar yang sebelumnya direpresentasikan dalam bentuk matriks dua dimensi akan diubah menjadi satu dimensi sehingga dapat digunakan sebagai input pada lapisan-lapisan Dense dalam model. Implementasi flatten pada CNN sering kali ditempatkan setelah lapisan pooling terakhir sebelum lapisan fully connected [2], [14].



Gambar 2.6 Ilustrasi *Flatten*

2.1.6 Normalisasi

Normalisasi adalah sebuah proses untuk mengubah skala data menjadi 0-1. Proses normalisasi dirumuskan pada persamaan 2.7 [29]. Tujuan dari normalisasi adalah untuk memastikan bahwa semua variabel dalam dataset memiliki skala yang konsisten, sehingga dapat meningkatkan efisiensi dan performa model. Seperti yang dijelaskan oleh Sriyanto et al. dalam artikel tersebut, normalisasi membantu dalam menciptakan satu set nilai dengan properti tertentu yang memudahkan proses analisis dan pemrosesan data [30].

$$x_{norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (2.7)$$

Dimana x_{norm} merupakan data yang telah di normalisasi, x_i merupakan data ke-i, x_{min} merupakan data yang paling kecil dan x_{max} adalah data yang paling besar.

2.1.7 Augmentation

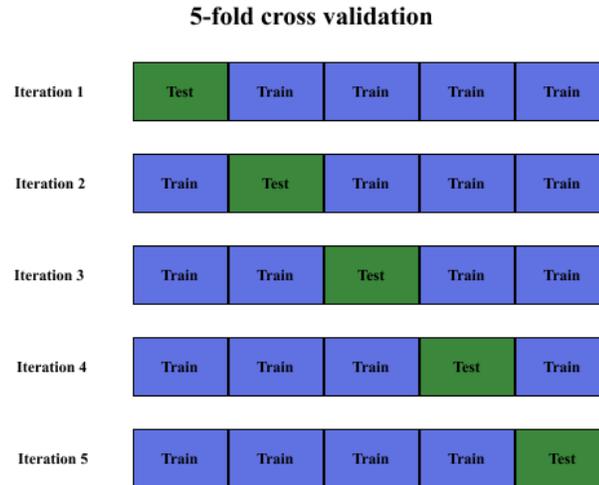
Augmentation data bertujuan untuk memperkaya dataset dengan membuat variasi tambahan dari gambar yang ada [31]. *Augmentation data* dilakukan dengan menerapkan beberapa transformasi acak pada gambar, teknik yang umum digunakan seperti rotasi, perubahan skala, refleksi horizontal, dan pengaturan pencahayaan [6]. Teknik ini digunakan untuk meningkatkan jumlah data pelatihan dan membantu model belajar untuk mengenali pola yang lebih general [17], bukan hanya pola yang ada pada citra asli.

2.1.8 K-Fold Cross Validation

Cross validation merupakan metode statistik yang sering digunakan untuk mengevaluasi performa model *machine learning* dengan cara membagi dataset menjadi beberapa subset untuk pelatihan dan pengujian. Salah satu metode yang paling umum yaitu *k-fold cross validation*, seluruh proses *k-fold cross validation* dapat diringkas dalam empat langkah berikut:

- a. Membagi seluruh dataset menjadi pelatihan S menjadi k subset (*fold*) dengan ukuran yang sama.
- b. Melatih model pada $k - 1$ subset sebagai data pelatihan, sementara satu subset digunakan sebagai data pengujian.
- c. Menghitung rata-rata kesalahan empiris dari seluruh iterasi k .
- d. Menggunakan seluruh dataset untuk pelatihan akhir berdasarkan model dengan rata-rata kesalahan terendah

Pendekatan ini memastikan bahwa model tidak terlalu bergantung pada subset tertentu dan memberikan estimasi performa yang lebih stabil karena memanfaatkan seluruh data secara efektif untuk pelatihan dan pengujian, tanpa ada data yang terbuang [32]. Berikut gambar ilustrasi proses *k-fold cross validation*.



Gambar 2.7 Proses *K-Fold Cross Validation*

2.1.9 Confusion Matrix

Confusion matrix digunakan sebagai alat evaluasi untuk mengukur performa model klasifikasi dengan membandingkan hasil prediksi model terhadap data aktual. *Confusion matrix* memuat informasi yang bersifat prediktif dalam bentuk klasifikasi. Tabel ini mengelompokkan hasil prediksi kedalam empat kategori utama, yaitu *true positive (TP)*, *false positive (FP)*, *false negative (FN)*, dan *true negative (TN)*. Berikut contoh tabel *confusion matrix* [33].

Tabel 2.1 *Confusion Matrix*

		<i>Predicted</i>	
		<i>Positive</i>	<i>Negative</i>
<i>Actual</i>	<i>Positive</i>	TP	FN
	<i>Negative</i>	FP	TN

Keterangan:

- a. *True Positive (TP)*: Jumlah data aktual dengan nilai positif yang diprediksi sebagai positif.

- b. *False Positive (FP)*: Jumlah data aktual dengan nilai negatif yang diprediksi sebagai positif.
- c. *False Negative (FN)*: Jumlah data aktual dengan nilai positif yang diprediksi sebagai negatif.
- d. *True Negative (TN)*: Jumlah data aktual dengan nilai negatif yang diprediksi sebagai negatif.

Confusion matrix digunakan untuk menghitung beberapa *matrix* evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-Score*. Berikut persamaan dari masing-masing *matrix* tersebut [3].

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.8)$$

$$precision = \frac{TP}{TP+FN} \quad (2.9)$$

$$recall = \frac{TP}{TP+FN} \quad (2.10)$$

$$f1 - score = \frac{2*precision*recall}{precision+recall} \quad (2.11)$$

Hasil pengukuran menggunakan *confusion matrix* diatas digunakan untuk mengevaluasi performa model, dan memberikan analisis lebih lanjut terhadap distribusi kesalahan prediksi antara kelas positif dan negatif [34]. Pemahaman mendalam mengenai nilai TP, FP, FP, dan TN sangatlah penting, misalnya, peningkatan nilai TP dan penurunan FN akan menghasilkan nilai recall yang lebih tinggi, yang sangat krusial dalam aplikasi dimana kehilangan deteksi kasus positif dapat memiliki konsekuensi serius [35], [36].

2.1.10 Dropout

Dropout merupakan salah satu teknik regularisasi yang digunakan dalam *neural network* untuk mengatasi masalah *overfitting* selama proses pelatihan. Teknik ini pertama kali diperkenalkan oleh Srivastava et al. (2014) dan telah digunakan secara

luas dalam berbagai arsitektur *deep neural network* [37]. Proses *dropout* bekerja dengan secara acak memilih neuron dari beberapa lapisan menengah dalam jaringan dan mengganti nilai dari neuron-neuron tersebut dengan nol selama langkah pelatihan tertentu. Dengan begitu, neuron-neuron yang nilainya sudah 0 tidak berkontribusi pada tahapan pelatihan. Dalam implementasinya, *dropout* sering dikombinasikan dengan teknik regularisasi lainnya seperti *batch normalization* untuk mencapai model yang lebih stabil saat pelatihan [38].

2.1.11 Batch Normalization

Batch Normalization merupakan teknik normalisasi aktivasi yang populer, dalam menstabilkan proses optimasi dengan menormalkan output pada setiap layer. Proses ini dilakukan dengan mengurangi nilai rata-rata (mean) dan membagi dengan standar deviasi aktivasi pada layer tersebut. Selain meningkatkan stabilitas pelatihan, *batch normalization* juga membantu mengatasi permasalahan nilai *vanishing gradient*, dimana nilai gradien menjadi sangat kecil dan menghambat pembaruan bobot. Kemudian *batch normalization* memiliki efek regularisasi ringan yang dapat membantu dalam mengurangi risiko *overfitting* pada model [21], [39].

2.2 Penelitian Terkait

Penelitian terkait merupakan rangkaian upaya penulis atau peneliti dalam membaca, meninjau, dan mencari perbandingan antara penelitian yang akan dilakukannya dengan sejumlah penelitian yang sudah ada sebelumnya, dimana penelitian-penelitian tersebut memiliki topik, teori, maupun konsep yang berkesinambungan dengan penelitian yang dilakukan peneliti. Berikut ini merupakan penelitian terdahulu yang memiliki topik, teori, maupun konsep yang berkesinambungan dengan penelitian yang dilakukan peneliti.

Penelitian yang dilakukan Luqman Hakim, dkk dalam penelitiannya, Hakim menggunakan CNN dengan metode *pooling* yang mencakup kombinasi *max pooling* dan *average pooling* untuk klasifikasi kanker kulit. Penelitian ini menunjukkan bahwa kombinasi *pooling* dapat mencapai akurasi sebesar 75% dengan pengaturan 100 *epoch*

[6]. Meskipun akurasi belum optimal, penelitian tersebut memberikan wawasan bahwa kombinasi *pooling* memiliki potensi untuk ditingkatkan lebih lanjut. Penelitian ini juga menyarankan untuk eksplorasi lebih lanjut terhadap kombinasi *pooling* yang lebih efektif seperti *global pooling*.

Penelitian yang dilakukan oleh Reynaldi Rio Saputro, dkk dalam penelitiannya Saputro mengembangkan CNN dengan menggunakan metode *pooling* yang paling umum yaitu *Max Pooling*, namun dengan pengaturan *epoch* yaitu 50 *epoch*. Hasilnya menunjukkan peningkatan akurasi hingga 92,64%. Studi ini menyoroiti bahwa pengaturan parameter pelatihan, seperti jumlah *epoch*, dan pemilihan jenis *pooling* berkontribusi signifikan terhadap performa model CNN [1]. Dalam konteks penelitian ini, hasil tersebut *max pooling* memiliki peran penting dalam meningkatkan akurasi model CNN ketika digunakan dengan parameter pelatihan yang optimal. Penelitian ini juga menjadi acuan penting untuk mengeksplorasi dalam mencoba jenis *pooling* lain, seperti *Global Average Pooling (GAP)*, untuk melihat kemampuan kinerja model lebih lanjut jika dikombinasikan.

Penelitian yang dilakukan oleh Teresia R. Savera, dkk dengan tujuan meneliti tingkat akurasi metode K-NN dan CNN dalam mendeteksi kanker kulit. Penelitian tersebut membandingkan nilai akurasi antara metode regresi dan metode CNN dengan menggunakan dataset ISIC yang berjumlah 800 data gambar lesi kulit. Hasil penelitian tersebut mendapatkan nilai akurasi klasifikasi pada metode regresi mencapai 75% dan untuk CNN sebesar 76%. Penelitian ini menunjukkan bahwa CNN lebih efektif dalam klasifikasi kanker kulit dibandingkan metode tradisional seperti KNN, dan penggunaan *max pooling* membantu meningkatkan akurasi dalam identifikasi fitur lokal pada gambar kanker kulit [5]. Namun akurasi tersebut belum terlalu tinggi dan kemungkinan dapat ditingkatkan lagi dengan tahap *preprocessing* atau penambahan data citranya.

Penelitian yang dilakukan Vishal Passricha, dkk pada penelitian ini menganalisis berbagai jenis *pooling* dalam CNN, termasuk *max pooling* untuk pengenalan suara, dan menemukan bahwa *max pooling* bekerja sangat baik dalam kondisi tertentu (misalnya lingkungan bersih tanpa bising), tetapi kurang efektif dalam kondisi dengan variasi atau kebisingan tinggi karena hanya mempertimbangkan fitur

lokal yang dominan, hal tersebut juga dapat menyebabkan *overfitting* [16]. Hasil penelitian ini mendukung penggunaan *Global Average Pooling* sebagai alternatif untuk mengatasi kelemahan tersebut dengan memadukannya sehingga membantu mengurangi *overfitting* dengan memberikan kontribusi yang lebih merata dari seluruh area fitur pada gambar yang bervariasi.

Penelitian yang dilakukan oleh Lokesh Kumar, dkk, dimana penelitian tersebut Kumar meneliti pengaruh *Global Average Pooling* (GAP) dalam klasifikasi tumor otak menggunakan CNN dan menemukan bahwa GAP membantu mengurangi *overfitting* dan meningkatkan generalisasi model, dengan akurasi mencapai 97,48% [17]. Temuan ini mendukung bahwa GAP memiliki efek positif dalam meningkatkan akurasi model, yang mendasari penelitian ini untuk mengkombinasikan GAP dengan *max pooling* untuk klasifikasi kanker kulit.

Berikut tabel penelitian terdahulu yang telah dijabarkan sebelumnya dapat dilihat pada tabel 2.2 adalah sebagai berikut:

Tabel 2.2 Penelitian Terdahulu

No	Peneliti, Tahun	Objek Penelitian	Poling	Metode preprocessing	Model	Validasi	Dataset	Data	Akurasi
1	Luqman Hakim, 2021	[Kanker Kulit] 1. Actinic keratoses 2. Basal cell carcinoma 3. Benign keratosis like lesions 4. Dermatofibroma 5. Vascular lesions 6. Melanoma 7. Melanocytic nevus.	- Max - Average	1. Ekstraksi Dataset 2. Augmentasi Data 3. Resizing Gambar 4. Data Splitting	CNN	Train-Validation-Test Split	Dataset (ISIC) 2018	10015	75%
2	Reynaldi Rio Saputro, 2022	[Kanker Kulit] 1. Melanoma 2. Non-melanoma	Max	1. Data Splitting 2. Resizing	CNN	Hold-out validation	Dataset Kaggle	17.805	92,64%
3	Teresia R. Savera, 2020	[Kanker Kulit] 1. Benign 2. Malignant	Max	1. Image Resizing 2. Image Masking 3. Binarization/Thresholding	K-NN + CNN	-	Dataset (ISIC)	200/800	75% / 76,56%
4	Vishal Passricha, 2019	[Komparasi Jenis Pooling] Pengenalan Suara berbahasa Hindi (Bersih & Noise)	1. Max 2. Average 3. Stochastic 4. Lp 5. Mixed pooling (Max+ Average)	1. Ekstraksi Fitur 2. Cepstral Mean and Variance Normalization (CMVN) 3. Sliding Window	CNN	K-Fold Cross Validation	Hindi Speech Dataset	23.984 Audio File	Max pooling = 22.1% Average pooling = 23.5%

									Stochastic pooling = 19.3% Lp pooling = 21.7% Mixed pooling = 20.5%
5	Lokesh Kumar, 2021	[Tumor Otak] 1. Meningioma 2. Glioma 3. Pituitary	Global Average Pooling	1. Augmentasi Data	CNN	K-Fold Cross-Validation	Dataset Tumor	3064	97,48%
6	Fitra Salam S. Nagalay	[Kanker Kulit] 1. Benign 2. Malignant	- Max - Global Average Pooling	1. Resizing 2. Normalization Citra 3. Augmentation Data	CNN	K-Fold Cross-Validation	Dataset (ISIC)	10.605	???

Berdasarkan hasil penelitian terdahulu tersebut, dapat disimpulkan bahwa setiap metode *pooling* pada CNN memiliki kelebihan dan kekurangannya masing-masing. Penelitian oleh Luqman Hakim dkk, menunjukkan bahwa kombinasi *pooling* memiliki potensi untuk ditingkatkan. Sementara itu, penelitian Reynaldi Rio Saputro dkk, menyoroti bahwa *max pooling* mampu memberikan akurasi yang signifikan dengan pengaturan parameter pelatihan data yang optimal. Penelitian yang dilakukan Teresia R. Savera dkk juga menunjukkan bahwa CNN secara konsisten lebih unggul dibandingkan metode tradisional seperti KNN dalam klasifikasi kulit. Namun *max pooling* ini sendiri mempunyai kelemahan seperti pada penelitian yang dilakukan Vishal Passricha dkk, dimana *pooling* tersebut dapat menyebabkan *overfitting* karena hanya mempertimbangkan fitur lokal yang dominan. Hal tersebut mendorong penggunaan kombinasi *pooling* lain seperti *Global Average Pooling* (GAP) dalam meningkatkan generalisasi model. Penelitian Lokesh Kumar dkk, mendukung manfaat GAP dalam mengurangi *overfitting* dan meningkatkan generalisasi, sehingga menghasilkan akurasi yang lebih tinggi pada tugas klasifikasi. Berdasarkan temuan tersebut, penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan model *Convolutional Neural Networks* (CNN) dengan memanfaatkan kombinasi *max pooling* dan GAP untuk mendeteksi kanker kulit. Dengan menggunakan dataset ISIC (*International Skin Imaging Collaboration*), penelitian ini akan mengeksplorasi pengaruh kombinasi kedua jenis *pooling* terhadap kinerja model CNN. Diharapkan kombinasi ini dapat menghasilkan model yang lebih efisien dalam mendeteksi kanker kulit, baik untuk kategori *malignant* maupun *benign*, sehingga dapat mendukung proses diagnosis yang cepat, akurat, dan terjangkau.