BAB 2
DASAR TEORI

2.1 Kajian Pustaka

Penelitian ini dirancang dengan menerapkan alogaritma CNN dalam mengklasifikasikan penyakit *Acute Lymphoblastic Leukemia* berdasarkan citra ALL. Studi sebelumnya mengenai klasifikasi citra menggunakan CNN dijadikan sebagai dasar, mengingat metode tersebut efektif untuk klasifikasi citra.

Salah satu penelitian terkait pada tahun 2020 [1] mengenai klasifikasi penyakit ALL menggunakan CNN. Pendekatan yang dilakukan pada penelitian ini dapat mengklasifikasi ALL ke dalam tipe L1, L2, dan L3 atau tipe normal. Pada penelitian ini pengumpulan data terdiri dari semua gambar subtipe, yang meiliputi 100 gambar L1, 100 gambar L2, 30 gambar L3 karna sifat L3 yang langka dan 100 gambar sumsum tulang normal. Penedektakan yang diusulkan memiliki kinerja yang dapat diterima, mengambil gambar sumsum tulang sebagai input, melakukan segmentasi dan mengklasifikasikan normal jika sumsum tidak terpengaruh atau menjadi subtipe L1, L2, dan L3. Perbandingan sistem yang disulkan dengan pengklasifikasian dengan ketepatan Naif Baysian 78,34%, KNN 80,42%. Kekurangan dari pendekatan pada penelitian ini masih memerlukan akurasi yang lebih baik untuk menyegmentasikan sel yang masih tumpang tindih.

Penelitian terkait pada tahun 2020 [7] mengenai klasifikasi ALL menggunakan arsitektur R-CNN. pada penelitian ini mengembangkan klasifikasi leukemia limfoblastik akut (ALL) berdasarkan segmentasi. Dataset yang diguanakan pada penelitian ini adalah dataset penelitian terdahulu oleh Naden Siti Fatonah yang diperoleh dari dr. RS Soetomo Surabaya. Jumlah data yang diguanakan adalah 301 citra multi-cell ALL yang terdiri dari 128 citra tipe L1, 63 citra tipe L2 dan 110 citra tipe L3. Pada penelitian ini segmentasi instance lymphoblast menggunakan Mask R-CNN yang bertujuan untuk mendapatkan kelas subtipe ALL dan kemdian direkamn untuk digunakan pada tahap klasifikasi. Dengan menggunakan Metode yang diusulkan didapatkan akurasi 83,72%, presisi 85,17% dan sensitivitas 81,61%. Kekurangan pada penelitian ini adalah masih sulit

untuk menemukan metode terbaik untuk penelitian ini, terutama dalam proses deteksi, segmentasi dan klasifikasi limfoblas. bahkan kondisi dataset juga menjadi menjadi tantangan pada penelitian ini karena kondisi citra asli yang diperoleh dari rumah sakit bisa berbeda dengan kondisi citra yang diperoleh dari dataset open-source diinternet. Selain itu kekurangan dari 7 penelitian ini menggunakan metode peningkatan kontras dengan meningkatkan kontras dapat menunjukkan area yang memiliki eksprosur dapat menunjukkan area yang memiliki eksprosur yang buruk, tetapi juga dapat membuat area yang memiliki eksprosur yang baik menjadi overexposed pada saat bersamaan.

Penelitian terkait pada tahun 2022 [9] mengenai klasifikasi *Leukocytes* menggunakan metode CNN. pada penelitian ini menyajikan sistem untuk klasifikasi otomatis Leukosit. sistem akan menerima gambar mikroskopis sebagai input dan menggunakan arsitektur CNN untuk memprediksi kategorinya. Kemduian sistem akan menggabungkan hasil dari model yang berbeda dan akan mengklasifikasikan sel ke dalam kelas yang sesuai. Pada penelitian ini jumlah dataset yang digunakan sekitar 2000 gambar miksroskopis dari 7 jenis sel darah putih, dataset tersebut disediakan oleh rumah sakit Menara gendering afiliasi sekolah kedokteran Universitas Nanjing. Pada penelitian ini mengusulkan untuk menggunakan arsitektur CNN yakni ResNet50, VGG 19 dan MobileNet yang mampu menghasilkan akurasi sebesar 88,5%.

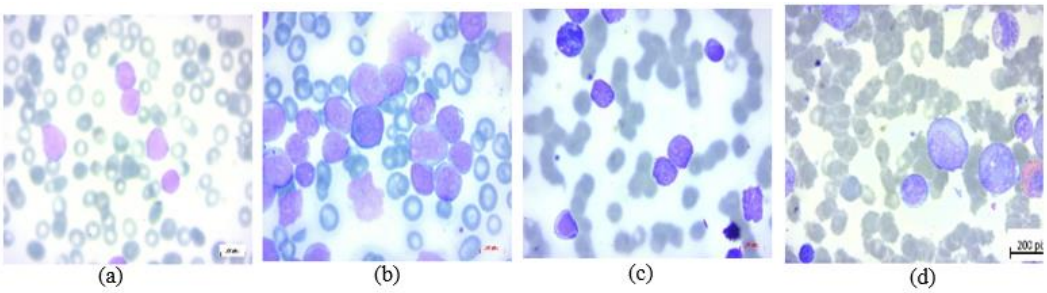
Penelitian berikutnya pada tahun 2021 [2] menegnai klasifikasi kanker sel darah putih menggunakan metode CNN. pada penelitian ini peneliti menggunakan arsitektur CNN yaitu Resnet dan VGG16. Pada penelitian ini didapatkan hasil accuracy traning arsitektur VGG16 dengan menggunakan optimizers Adam dengan menggunakan 100 epoch yakni sebesar 93,80% dan accuracy testing sebesar 87,00%. Pada arsitektur ResNet didapatkan accuracy traning dan accuracy testing berturut-turut 81,81% dengan menggunakan 80 epoch dan menggunakan optimizer RMSProp. Pada penelitian ini mengusulkan untuk melakukan eksplorasi untuk mendapatkan accuracy yang lebih tinggi dengan cara mengubah model arsitektur dan melakukan perubahan hyperparameter didalamnya.

2.2 Sel Darah Putih

Leukemia adalah jenis kanker yang menyerang sel darah putih, yang sebenarnya bertugas melawan infeksi dalam tubuh. Ini adalah kanker yang berasal dari sumsum tulang dan mempengaruhi sel darah putih manusia. Leukemia adalah penyakit serius yang terjadi ketika sel darah putih tumbuh tidak normal [8]. Leukemia menyebabkan tubuh memproduksi sel darah yang tidak normal dalam jumlah besar. Sel-sel darah abnormal ini menyebar cepat dan memerlukan penanganan segera. Hal ini mengganggu fungsi normal darah dan membuat sistem kekebalan tubuh rentan [2]. Dalam leukemia, sel darah yang tidak normal akan menyebar dengan cepat dan memerlukan penanganan yang segera. Hal ini mengakibatkan gangguan pada fungsi normal darah dan membuat sistem kekebalan tubuh menjadi rentan [2]. Sel-sel darah yang terpengaruh oleh leukemia sangat berbeda dengan sel darah normal dan tidak dapat berfungsi seperti biasanya [3].

Berdasarkan jenis sel darah putih yang terlibat, leukemia terbagi menjadi empat jenis. Pertama, *Acute Lymphoblastic Leukemia* (ALL) terjadi ketika sumsum tulang memproduksi terlalu banyak sel darah putih jenis limfosit yang belum matang. Kedua, *Chronic Lymphocytic Leukemia* (CLL) terjadi ketika sumsum tulang menghasilkan terlalu banyak limfosit yang tidak normal secara perlahan dan menyebabkan kanker. Ketiga, *Acute Myeloblastic Leukemia* (AML) terjadi ketika sumsum tulang menghasilkan terlalu banyak sel myeloid yang belum matang atau myeloblast. Terakhir, *Chronic Myelocytic Leukemia* (CML) terjadi ketika sumsum tulang tidak mampu memproduksi sel myeloid yang matang [10]

Dalam penelitian ini, akan dilakukan klasifikasi leukemia jenis ALL. Sel kanker akan dibagi menjadi dua kategori, yaitu *Benign* dan *Malignant*, dengan kategori *Malignant* kemudian dibagi lagi menjadi dua kelompok kelas, yaitu *Early Pre* dan *Pro* [11]. Dataset ini diambil di laboratorium sumsum tulang di Rumah Sakit Taleqani (Tehran, Iran). Dataset tersebut terdiri dari 3256 gambar PBS yang diambil dari 89 pasien yang menderita ALL, di mana sampel darah mereka disiapkan dan diwarnai oleh staf laboratorium yang terampil. Semua gambar diambil menggunakan kamera Zeiss pada mikroskop dengan perbesaran 100x dan disimpan dalam format file JPG. Penentuan *definitif* tentang jenis dan subtipe sel-sel ini dilakukan oleh seorang spesialis menggunakan alat *Flow Cytometry* [12].



Gambar 2. 1 Contoh citra penyakit pada *Acute Lymphoblastic leukemia* yang terdiri dari 4 kelas. (a) *Benign*, (b) *early*, (c) *pre*, (c) *pro* [13]

Berikut adalah penjelasan terkait klasifikasi ALL ke dalam empat kelas yang terapat pada Gambar 2.1.

a) Benign (Noncancerous)

Pada kondisi ini, "*benign*" merujuk pada jenis sel darah putih yang tidak berkembang menjadi kanker atau leukemia. Sel-sel ini normal dan tidak menyebabkan kanker [13].

b) Early (Early Satge Type L1)

Pada tahap ini, *leukemia* masih dalam tahap awal perkembangannya. Sel-sel leukemik masih terlokalisasi dalam sumsum tulang atau darah, dan gejala mungkin belum terlalu jelas atau menyebar ke organ lain. Ini menunjukkan leukemia dalam tahap awal dengan penyebaran yang terbatas [13].

c) Pre (Middle stage Type 2)

Pada tahap ini, leukemia telah berkembang lebih jauh dari tahap awal. Sel-sel leukemik telah menyebar ke berbagai bagian dari sumsum tulang atau aliran darah. Ini menandakan penyebaran yang lebih luas dari leukemia [13].

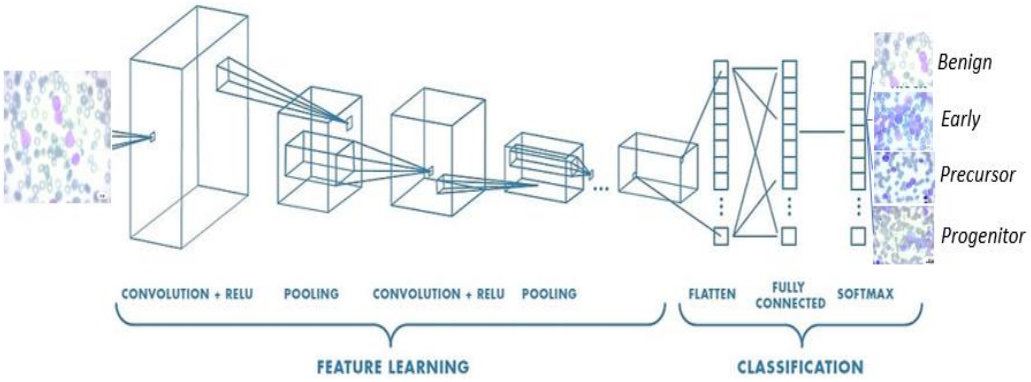
d) Pro (Later stage type 3)

Pada tahap ini, leukemia telah mencapai tahap lanjut dari perkembangannya. Sel-sel leukemik telah menyebar ke organ tubuh lain seperti limpa, hati, atau bahkan ke otak dan sistem saraf pusat. Ini menunjukkan penyebaran yang sangat luas dari leukemia dan bisa menyebabkan komplikasi serius [13].

2.3 *Convolutional Neural Network* (CNN)

Convolutional Neural Network adalah salah satu jenis *neural network* yang biasa digunakan pada data *image.* CNN bisa digunakan untuk mendeteksi dan mengenali object pada sebuah *iamge.* CNN terdiri dari *neuron* yang memiliki *weight,* bias dan *activation function. Convolutional layer* uga terdiri dari *neuron* yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi *(pixels)*

CNN adalah jenis arsitektur jaringan saraf yang secara khusus dirancang untuk mengenali pola dan menganalisis gambar. CNN digunakan untuk mendeteksi dan mengidentifikasi objek dalam gambar. Metode ini telah menjadi salah satu yang paling terkenal dan diminati dalam bidang visi komputer dan pembelajaran mesin [13]. Salah satu keunggulan utama CNN dalam klasifikasi gambar adalah kemampuannya dalam merespons dan memahami fitur-fitur visual pada gambar. Struktur dasar CNN terdiri dari lapisan input, Lapisan Konvolusi, dan lapisan klasifikasi. CNN unggul dalam tugas-tugas seperti klasifikasi gambar, deteksi objek, dan segmentasi karena kemampuannya untuk secara otomatis belajar hierarki spasial fitur dari data. Dengan menerapkan filter konvolusi dan lapisan pooling, CNN dapat secara efektif menangkap pola lokal dan ketergantungan spasial dalam gambar, memungkinkannya untuk generalisasi dengan baik ke data baru yang belum pernah dilihat sebelumnya. Hal ini membuat CNN sangat efektif dalam berbagai tugas terkait gambar, termasuk analisis gambar medis, kendaraan otonom, pengenalan wajah, dan lainnya.



Gambar 2. 2 Arsitektur CNN [14]

Pada Gambar 2.2 menunjukkan arsitektur CNN yang terdiri dari *input layer, Convolution Layer* dan *classification. Input layer* dalam hal ini adalah citra sel *leukemia*. Kemudian citra tersebut akan dikonvolusi dan ReLU membantu mengatasi masalah *vanishing gradient* dan mempercepat proses pelatihan. *Globlas Average Pooling* digunakan untuk mengurangi dimensi dari fitur dalam satu layer. *Dropout* digunakan untuk mengurangi *overfitting* dalam model. *Batch Normalization* digunakan untuk mempercepat dan menstabilkan pelatihan model. *Flatten* mengubah tensor multidimensi menjadi satu dimensi. *Dense Layer* mempelajari pola dan hubungan kompleks dalam data dan *softmax* mengubah nilai dalam output menjadi probabilitas sehingga akan menghasilkan distribusi probabilitas untuk setiap kelas.

2.4 Input Layer

Pada CNN untuk analisis citra medis, seperti dalam deteksi kanker. Data gambar yang dijadikan input pada CNN adalah citra dari sel-sel ini. Tujuan akhir dari CNN adalah untuk mengklasifikasikan ALL ke dalam 4 kelas yaitu *Early, Benign, Pre dan Pro.* Jadi, Dengan menggunakan citra sel kanker sebagai input, model CNN dilatih dan diuji untuk mengidentifikasi dan mengklasifikasikan setiap sel ke dalam salah satu dari empat kelas tersebut, dengan tujuan untuk mencapai tingkat akurasi yang tinggi dalam pengklasifikasian yang dilakukan.

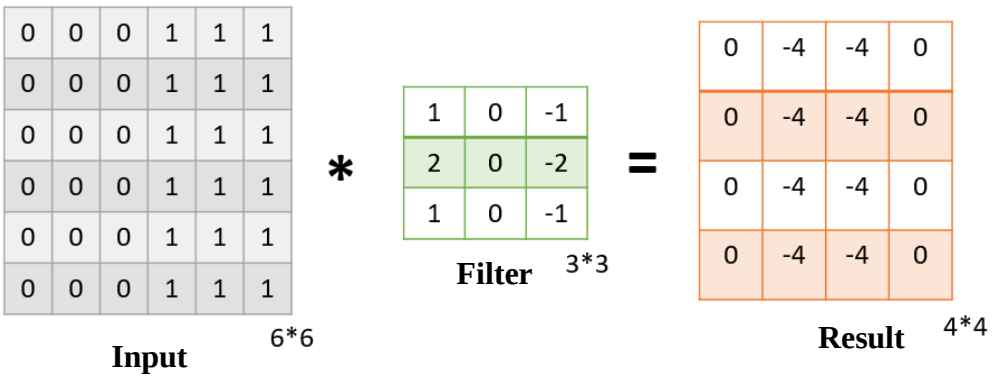
2.5 Feature Extraction

Dalam arsitektur CNN, terdapat bagian utama yang berperan penting dalam menyelesaikan tugas yang diberikan, yaitu proses ekstraksi fitur. Dalam arsitektur CNN terdapat bagian utama yang bekerja untuk menyelesaikan tugas yang diberikan yaitu proses ekstraksi fitur. Tujuan dari tahap ini adalah untuk mengidentifikasi dan mengekstrak fitur-fitur penting dari data input (misalnya gambar) yang akan digunakan untuk melakukan klasifikasi lebih lanjut [15]. Proses ini melibatkan serangkaian operasi konvolusi, aktivasi, dan pooling yang dilakukan oleh lapisan-lapisan konvensional dalam CNN. Dengan cara ini, model dapat secara otomatis mengekstrak fitur-fitur hierarkis dari data input yang kemudian akan digunakan untuk membuat prediksi atau klasifikasi akhir. Proses ekstraksi fitur dilakukan melalui beberapa lapisan khusus dalam CNN, yaitu:

2.5.1 Convolution Layer

CNN terdiri dari beberapa blok penyusun atau lapisan arsitektur. Salah satu blok penyusun dari CNN dapat disebut sebagai lapisan *convolution*. Pada lapisan ini dilakukan operasi konvolusi yang bertujuan untuk mengekstraksi fitur citra input. Terdapat 3 komponen pada *convolution layer* yaitu input data, karnel atau apat juga disebut sebagai filter dan terakhir *featrue map.* Input data adalah proses awal yang terjadi pada *convolution layer*, proses ini terdiri dari piksel matriks dalam tiga dimensi. Proses selanjutnya dilakukan oleh karnel atau filter. Kernel merupakan parameter berbentuk grid dengan bobot yang digunakan untuk mendeteksi pola yang di miliki oleh suatu citra input. Selama awal proses pelatihan model CNN, semua bobot kernel ditetapkan dengan angka acak. Kemudian, dengan sietiap periode pelatihan, bobot disetel dan karnel belajar untuk mengekstraksi fitur[15].

Cara menghitug konvolusi adalah dengan melakukan perkalian matriks antar citra input dan karnel melakukan dot product. Dot product akan melakukan perhitungan antara piksel input dan kernel. Dot product kemudian dimasukkan kedalam output matriks lalu filter akan bergeser sedikit demi sedikit dan mengulangi setiap prosesnya hingga seluruh citra selesai diproses [16]. Hasil akhir dari dot product tersebut biasanya dikenal dengan feature map.



Gambar 2. 3 Proses *Convolutional Layer* [17]

Pada Gambar 2.3 merupakan proses *convolutional layer* terdapat input matrix dan kernel. Ukuran kernel biasanya 3x3 lalu akan dilakukan proses konvolusi. Matrix kernel akan digeser secara perlahan diatas input matrix, pada setiap langkah pergeseran dilakukan operasi konvolusi antara kersel dan

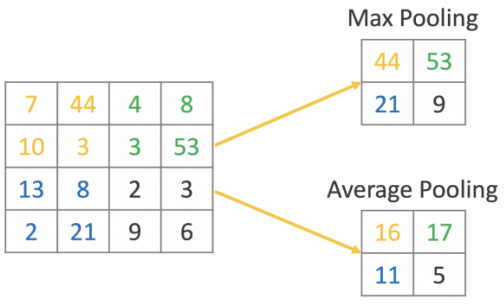
sebagian area input yang sesuai dengan ukuran kernel. Pada setiap langkah elemn-elemn kernel dikalikan dengan elemen-elemen input matrix yang berada dibawahnya, dan hasil perkalian tersebut dijumlahkan untuk menghasilkan nilai pada *feature map*. Setelah selesai proses konvolusi pada area input matrix, hasilnya adalah feature map. *Featre map* merupakan representasi dariinput matrix yang telah diproses menggunakan matrix karnel.

2.5.2 ReLU (*Rectified Linear Unit*)

ReLU merupakan fungsi aktifasi yang umum digunakan dalam jaringan saraf tiruan (*Neural networks*). ReLU didefinisikan sebagai f(x)=max(0,x) dimana x adalah input ke fungsi. Jika input lebih kecil dari 0, output ReLU akan 0, dan jika input lebih besar dari atau sama dengan 0, output ReLU akan sama dengan input itu sendiri [18]. Relu adalah salah satu jenis fungsi aktifasi yang bekerja dengan cara mengubah seluruh nilai input kedalam nilai positif.

2.5.3 *Pooling*

Pooling layer merupakan layer pada *feature extraction* yang berfungsi untuk menyederhanakan dimensi dan mengurangi parameter suatu *feature map* tanpa merubah informasi aslinya. Ada dua jenis lapisan pooling yang umum digunakan, yaitu max pooling dan average pooling. Max pooling mengambil nilai tertinggi dari bagian input yang terpengaruh oleh filter, sementara average pooling mengambil nilai rata-ratanya [19]. Anda dapat melihat ilustrasi max pooling dan average pooling pada Gambar 2.4.



Gambar 2. 4 *Pooling Layer* [14]

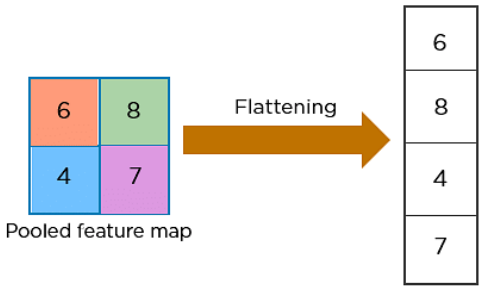
2.6 Classification

Bagian kedua dari *Convolutional Neural Network* (CNN) adalah proses klasifikasi. Setelah fitur-fitur penting diekstrak dari data input melalui tahap

konvolusi dan *pooling*, tahap klasifikasi bertujuan untuk mengenali dan memprediksi kategori atau kelas dari data tersebut [20]. Proses klasifikasi ini melibatkan serangkaian lapisan yang bertujuan untuk mengekstraksi dan menafsirkan fitur-fitur yang relevan yang telah ditemukan sebelumnya. Lapisan-lapisan ini umumnya termasuk lapisan-lapisan konvensional tambahan, diikuti oleh lapisan-lapisan sepenuhnya terhubung (*fully connected*) seperti *dense layer* yang berperan dalam membuat keputusan akhir tentang kelas mana yang paling mungkin sesuai dengan fitur-fitur yang diekstrak sebelumnya. Proses klasifikasi ini melibatkan lapisan-lapisan berikut:

2.6.1 Flattening

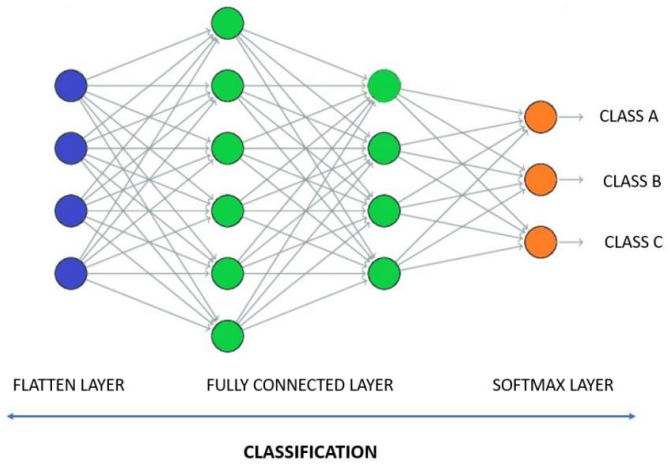
Flatten adalah proses mengubah matrix dari pooling menjadi vektor satu dimensi. Ini dilakukan dengan menggabungkan semua elemen dari *feature map* menjadi satu baris, sehingga menghasilkan vektor tunggal yang menyimpan seluruh informasi fitur dari input [17]. Gambar 2.5 merupakan ilustrasi dari dari *flatten*



Gambar 2. 5 *Flattening Layer* [14]

2.6.2 Fully Connected

Dalam arsitektur CNN terdapat bagian utama yang bekerja untuk menyelesaikan tugas yang diberikan yaitu proses ekstraksi fitur. Tahap ini bertujuan untuk mengenali dan mengekstraksi fitur-fitur penting dari data [19].



Gambar 2. 6 *Fully Connected Layer* [14]

Pada Gambar 2.6 merupakan proses *fully connected layer* dimana vektor tersebut dihubungkan ke laposan *fully connected layer*. *fully connected* akan terhubung dengan vektor tunggal tersebut. *Fully Connected layer* terdiri dari neuron-neuron yang memiliki bobot (*weight*) dan bias yang harus diatur selama proses pelatihan (*training*).

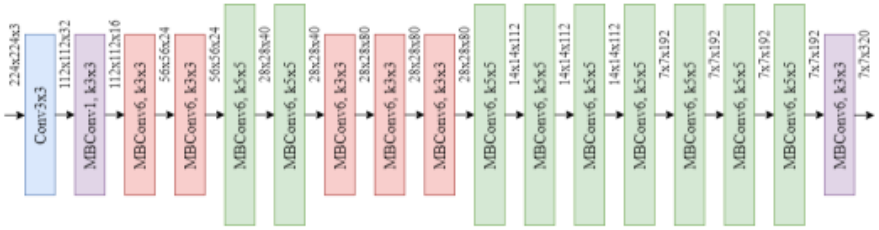
2.6.3 Dense Layer

Penggunaan dense layer pada tugas klasifikasi gambar, Dimana lapisan ini biasanya digunakan sebagai lapisan akhir dari model. Jika model tersebut harus mengenali beberapa kelas (misalnya kucing, anjing dan burung), dense layer akan memiliki beberapa neuron, dimana setiap neuron mewakili salah satu kelas dan menghasilkan probabilitas bahwa input masukan termasuk dalam kelas tersebut. dengan demikian, output dari dense layer akan berupa distribusi probabilitas di antara berbagai kelas yang mungkin, membantu model untuk membuat keputusan akhir tentang kelas mana yang paling mungkin sesuai dengan input gambar yang diberikan.

2.7 Arsitektur *EfficientNet-B0*

EfficientNet-B0 adalah salah satu varian dari arsitektur *EfficientNet*, yang dikembangkan oleh tim peneliti pada *Google* AI dalam upaya untuk menciptakan model *neural network* yang lebih efisien secara komputasi dengan kinerja yang tinggi. Arsitektur *EfficientNet* didasarkan pada konsep scaling yang cerdas, yaitu

meningkatkan kedalaman (*depth*), lebar (*width*), dan resolusi (*resolution*) model secara proporsional untuk mengoptimalkan kinerja dan keefisienan.



Gambar 2. 7 Arsitektur *EfficientNet-B0* [21]

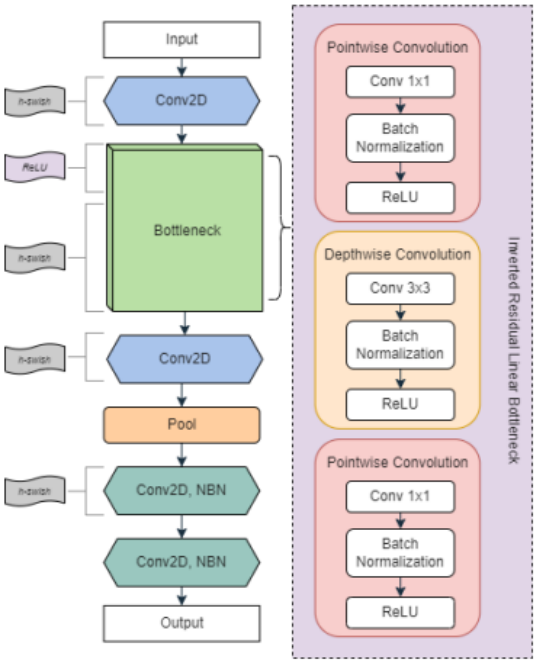
EfficientNet-B0 merupakan model dasar dari keluarga *EfficientNet*, dengan jumlah parameter paling rendah. Meskipun demikian, *EfficientNet-B0* tetap menawarkan kinerja yang baik dalam tugas klasifikasi gambar. Arsitektur ini memiliki beberapa blok berulang (*repeating blocks*) yang terdiri dari konvolusi *Depthwise Separable*, dilanjutkan dengan *Swish nonlinearity* sebagai fungsi aktivasi [21].

Konvolusi *Depthwise Separable* adalah teknik yang memecah konvolusi standar menjadi dua tahap terpisah, yaitu konvolusi *depthwise* dan *pointwise*. Pada tahap *depthwise*, setiap channel input diolah secara terpisah dengan filter yang hanya memiliki satu channel, menghasilkan *feature map* dengan banyak channel namun dengan komputasi yang lebih ringan. Kemudian, pada tahap *pointwise*, konvolusi 1x1 (*pointwise*) digunakan untuk menggabungkan channel-channel tersebut untuk menghasilkan output akhir. Dengan menggunakan teknik ini, jumlah parameter yang dibutuhkan dapat diurangi tanpa mengorbankan kinerja. Selain itu, dalam *EfficientNet-B0*, dilakukan scaling pada kedalaman, lebar, dan resolusi model berdasarkan parameter scaling yang ditentukan. Scaling ini memungkinkan model untuk beradaptasi dengan baik terhadap berbagai tugas dan ukuran dataset tanpa mengalami *overfitting* atau *underfitting* [21].

2.8 Arsitektur *MobileNetV3-Large*

Arsitektur *MobileNetV3 Large* dikembangkan dengan menggunakan teknik Network Architecture Search (NAS) dan algoritma NetAdapt untuk menemukan jumlah kernel yang optimal, sehingga menghasilkan model yang lebih ringan dan

efisien. *MobileNetV3-Large* adalah salah satu model jaringan saraf konvolusi yang dikembangkan untuk tugas pengenalan gambar. Model ini memiliki beberapa keunggulan, terutama dalam hal efisiensi dan kecepatan komputasi. *MobileNetV3-Large* dirancang dengan ukuran dan kompleksitas yang lebih kecil daripada beberapa model lainnya, seperti VGG atau ResNet, sehingga cocok untuk diimplementasikan pada perangkat seluler atau perangkat dengan sumber daya terbatas. *MobileNetV3-Large* juga dikenal karena aqueeze-and-excitation blocks, yang membantu dalam menyoroyi fitur-fitur penting dalam gambar dengan meningkatkan representasi-representasi yang relevan dan mengabaikan yang tidak penting.



Gambar 2. 8 Arsitektur *MobileNetV3-Large* [22]

MobileNetV3 Large terdiri dari blok-blok *bottleneck*, yang terdiri dari tiga layer konvolusi berbeda yaitu konvolusi 1×1, 3×3, dan 1×1. Konvolusi 1×1 digunakan untuk mengurangi dimensi input citra, kemudian diikuti oleh konvolusi 3×3 untuk menghasilkan representasi fitur yang lebih kompleks dari citra tersebut. Selanjutnya, konvolusi terakhir 1×1 digunakan untuk mengembalikan dimensi menjadi seperti input awal. Salah satu fitur khas dari *MobileNetV3-Large* adalah penggunaan *Squeeze and Excite* (SE) dalam setiap blok *bottleneck*. Teknik SE

bertujuan untuk memberikan penekanan pada fitur-fitur yang penting dan mengurangi peran fitur-fitur yang kurang relevan. Ini dilakukan dengan memperkenalkan mekanisme gating yang memungkinkan model untuk menyesuaikan bobot secara adaptif, sehingga fitur-fitur yang paling penting diberi lebih banyak bobot saat proses klasifikasi [23].

Selain itu, dalam *MobileNetV3-Large* fungsi aktivasi ReLU yang biasa digunakan dalam arsitektur sebelumnya diganti dengan *h-swish nonlinearity h-swish* adalah fungsi aktivasi nonlinier yang mengkombinasikan fungsi ReLU dengan fungsi linear, dan telah terbukti dapat meningkatkan akurasi *neural networks* dengan mengurangi risiko *vanishing gradient*. Arsitektur *MobileNetV3 Large* dirancang untuk menghasilkan model dengan ukuran yang lebih ringan dan efisien [24].

2.9 Hyperparameter

Hyperparameter adalah parameter yang nilainya ditentukan sebelum proses pelatihan dimulai. *Hyperparameter* mempengaruhi nagaimana model pembelajaran mesin akan belajar dan berkembang selama proses pelatihan. Berikut beberapa *hyperparameter* yang akan digunakan meliputi *optimizer, learning rate, epoch* dan *batch size.* Setiap *hyperparameter* memiliki dampak yang berbeda terhadap kinerja model dan pemilihan yang tepat dari *hyperparameter* ini dapat mempengaruhi seberapa baik model dapat mempelajari pola dari data yang diberikan.

2.9.1 Optimizer

Optimizer adalah algoritma yang digunakan dalam proses pelatihan (*training*) model dalam *machine learning* dan *deep learning* untuk mencari dan mengoptimalkan nilai-nilai bobot (*weights*) dan bias dalam jaringan saraf (*neural network*) [25]. Tujuannya adalah untuk meminimalkan fungsi agar model dapat belajar dari data pelatihan dengan baik dan menghasilkan prediksi yang akurat pada data baru [7]. Berukut *optimiaer* yang digunakan pada penelitian ini sebagai berikut:

a. Adam (*Adaptive Moment Estimation*)

   Adam merupakan algoritma optimisasi yang digunakan dalam melatih jaringan saraf tiruan. Tujuan utama Adam adalah untuk memperbarui bobot jaringan dengan cara yang efisien dan cepat sehingga jaringan dapat belajar dari data pelatihan dengan baik [17].

b. RMSprop (*Root Mean Square propagation*)

   RMSprop adalah varian khusus dari *Adagrad* yang dikembangkan oleh *Profesor Geoffrey Hinton* untuk jaringan sarafnya. RMSprop memiliki kesamaan dengan Adaprop, yaitu metode optimasi lain yang bertujuan untuk menyelesaikan beberapa masalah yang belum teratasi oleh Adagrad. RMSprop menggunakan besarnya gradien terbaru untuk menormalkan gradient, yang menjaga rata-rata bergerak dari *gradien root mean square*, sehingga disebut RMS. [26].

2.9.2 Learning Rate

Learning rate adalah *hyperparameter* yang sangat penting dalam pelatihan jaringan saraf tiruan. Ini mengontrol seberapa besar perubahan yang dibuat pada bobot jaringan selama setiap iterasi pelatihan. Jika *learning rate* terlalu kecil, pelatihan bisa berjalan lambat karena perubahan pada bobot sangat kecil. Di sisi lain, jika *learning rate* terlalu besar, pelatihan bisa menjadi tidak stabil karena perubahan yang terlalu besar dan melompat-lompati minimum global dari fungsi kerugian. Oleh karena itu, memilih *learning rate* yang sesuai sangat penting untuk memastikan pelatihan jaringan berjalan dengan baik dan model dapat mencapai hasil yang baik Meskipun tidak ada aturan yang baku terkait dengan *learning rate* dalam konteks jaringan saraf tiruan, secara umum, semakin besar nilai *learning rate*, semakin mungkin akurasi jaringan akan menurun. Sebaliknya, ketika nilai *learning rate* semakin kecil, kemungkinan besar akurasi jaringan akan meningkat, meskipun dengan konsekuensi waktu komputasi yang lebih lama [18].

2.9.3 Epoch

Jumlah *epoch* menentukan berapa kali seluruh dataset pelatihan akan diperlihatkan kepada model selama proses pelatihan. Jumlah *epoch* yang

digunakan dalam pelatihan model merupakan salah satu *hyperparameter* yang harus ditentukan. Jumlah *epoch* yang tepat dapat bervariasi tergantung pada komplesitas tugas dan ukuran dataset. Terlalu sedikit *epoh* mungkin tidak memberikan model cukup kesempatan untuk belajar dengan baik, sementara jika terlalu banyak *epoch* akan menyebabkan *overfitting* [18].

2.9.4 Batch Size

Batch Size adalah jumlah sampel data yang digunakan dalam satu iterasi pelatihan model. Saat melatih model menggunakan alogaritma pembelajaran mesin, data serigkali terlalu besar untuk dimuat secara keseluruhan ke dalam memori komputer. Oleh karena itu, data dibagi menjadi beberapa *batch* yang lebih kecil. Setiap *batch* kemudian dimasukkan ke dalam model secara berurutan untuk melakukan pembelajaran. Proses ini diulangi hingga seluruh data telah dimasukkan ke dalam model dan satu iterasi melalui seluruh dataset disebut satu *epoch* [2].

2.10 Confusion Matrix

Confusion matrix adalah teknik yang digunakan untuk mengevaluasi kinerja suatu metode klasifikasi. Ini memberikan informasi tentang sejauh mana hasil klasifikasi dari sistem tersebut sesuai dengan hasil yang seharusnya. Dari *confusion matrix* dapat mengevaluasi performa model dalam mengklasifikasikan data ke dalam masing-masing kelas. Tabel *confusion matrix*, seperti yang terlihat dalam Tabel 2.1, menggambarkan perbandingan antara nilai aktual dan prediksi dari klasifikasi. [17]. Tabel 2.1 menggambarkan perbandingan antara nilai aktual dan nilai prediksi dari klasifikasi kelas *benign*.

Tabel 2. 1 Tabel Tingkat Klasifikasi

		Nilai Prediksi	
		Positive (pp)	**Negative (PN)**
Nilai Aktual	**Positive (P)**	True Positive (TP) Benign-Benign	False Negative (FN) Benign-Bukan Benign
	Positive (N)	False Positive (FP) Bukan Benign-Benign	True Negative (TN) Bukan Benign-Bukan Benign

Confusion matrix adalah alat evaluasi yang digunakan dalam pemodelan statistik dan pembelajaran mesin untuk mengevaluasi kinerja model klasifikasi. Ini merupakan tabel yang menggambarkan performa model dengan membandingkan prediksi model dengan nilai sebenarnya dari data yang diamati. *Confusion matrix* memberikan gambaran yang jelas tentang seberapa baik model dapat memprediksi data ke dalam kelas yang benar dan seberapa sering kesalahan prediksi terjadi untuk setiap kelas. Dalam penggunaan confusion matrix untuk mengukur performa klasifikasi, terdapat empat istilah yang mewakili hasil dari proses klasifikasi, yaitu TP (*True Positive)*, TN (*True Negative*), FP (*False Positive*), dan FN (*False Negative*), yang dijelaskan dalam Tabel 2.1. Berikut adalah definisi dari masing-masing istilah tersebut:

a. TP *(True Positif)*

   Menunjukkan jumlah data yang kelasnya aktual positif dan model juga memprediksi positif.

b. TN *(True Negatif)*

   Merupakan jumlah data yang kelasnya aktual negatif dan model juga memprediksi negatif.

c. FP *(Flase Positf)*

   Mengindikasikan jumlah data yang kelasnya aktual negatif, tetapi model memprediksi positif.

d. *FN (Flase Negatif*)

   Menunjukkan jumlah data yang kelasnya aktual positif, namun model memprediksi negatif.

Dengan menggunakan *confusion matrix* dapat menghitung berbagai metrik evaluasi kinerja model seperti *presisi, recall, F1-score* serta menvisualisasikan pola kesalahan yang dilakukan oleh model dalam klasifikasi. Beberapa parameter digunakan untuk mengukur kinerja sistem yang akan dievaluasi dalam penelitian ini, diantaranya:

1. Akurasi

Akurasi merupakan ukuran seberapa tepat sistem dalam mengklasifikasi masukan sehingga menghasilkan keluaran yang sesuai dengan kelasnya. Rumus untuk menghitung akurasi dapat ditemukan pada persamaan (2.1) sebagai berikut:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

2. Presisi

   Presisi merupakan ukuran ketepatan prediksi positif yang benar terhadap total prediksi positif yang dibuat oleh sistem. Perhitungan presisi dapat dilakukan dengan menggunakan persamaan (2.2) berikut:

$$Presisi = \frac{TP}{TP + FP} \tag{2.2}$$

3. *Recall*

   Recall atau sensitivitas adalah ukuran ketepatan suatu sistem dalam memprediksi nilai positif yang benar terhadap total nilai aktual yang seharusnya positif. Persamaan matematis untuk *recall* dapat dilihat pada persamaan (2.3) berikut:

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

4. F1-Score

   F1-Score adalah metrik yang menggabungkan nilai presisi dan *recall* dengan mempertimbangkan *false positive* dan *false negative*. Persamaan untuk menghitung *F1-Score* dapat ditemukan pada persamaan (2.4) berikut:

$$F1 - Score = 2x\frac{presisi\ x\ recall}{presisi + recall} \tag{2.4}$$

5. *Loss*

   Parameter *loss* mengartikan ketidaktepatan sistem dalam melakukan prediksi untuk mengklasifikasikan sutau masukan.