

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia kesehatan, penyakit memiliki berbagai macam dan jenis, mulai dari yang tidak berbahaya sampai yang dapat menyebabkan kematian. Salah satu penyakit umum yang sering terjadi di masyarakat adalah penyakit lambung. Penyakit lambung merupakan penyakit yang mengganggu kesehatan dan fungsi sistem pencernaan yang melibatkan salah satu organ manusia yaitu lambung. Adapun beberapa contoh jenis penyakit lambung pada umumnya, yaitu *Gastritis*, *Gastroesophageal Reflux Disease (GERD)*, *Ulkus Gaster* (Tukak Lambung), Tumor Lambung, *Carcinoma Gaster* (Kanker Lambung), dan masih banyak yang lainnya. Saat ini penyakit lambung menjadi penyakit umum yang terdapat di masyarakat luas yang tidak boleh dianggap biasa, karena jika dibiarkan terus menerus dapat mengakibatkan penyakit yang lain muncul dan juga dapat menyebabkan kanker, bahkan kematian jika tidak segera mendapatkan penanganan yang baik. Penyakit lambung dapat dipengaruhi oleh beberapa faktor, diantaranya seperti pola makan yang tidak teratur, beban pikiran, gangguan kecemasan, dan juga infeksi pada lambung yang disebabkan oleh bakteri.

Menurut Organisasi Kesehatan Dunia (WHO) kanker lambung adalah kanker ke-5 yang paling umum di diagnosa dan ke-4 yang paling umum menyebabkan kematian akibat kanker di seluruh dunia pada tahun 2020, dan diperkirakan ada 1 juta kasus baru kanker lambung dan 769.000 kematian akibat kanker lambung. Selain kanker lambung, salah satu contoh penyakit lambung yang banyak diderita oleh masyarakat di dunia yaitu *Gastritis*. Secara global, insiden *Gastritis* diperkirakan mencapai 1,8 - 2,1 juta kasus per tahun. Di Indonesia sendiri, menurut Kementerian Kesehatan Republik Indonesia, prevalensi *Gastritis* mencapai 40,8% dan tercatat bahwa penyakit lambung termasuk ke dalam 10 penyakit terbanyak yang diderita masyarakat di Indonesia, dimana masing-masing angka kejadian di beberapa daerah sangat tinggi. Penyakit seperti *gastritis*, tukak lambung, dan pendarahan lambung telah diidentifikasi sebagai lesi prakanker yang dapat menyebabkan kanker lambung, karena lesi lambung ini tidak menunjukkan gejala hingga berkembang menjadi kanker, diagnosis dan pengobatan dini adalah cara terbaik untuk mengurangi kejadian tersebut, dan endoskopi adalah salah satu metode pilihan untuk mendapatkan diagnosis yang akurat agar lesi dapat diobati ataupun dihilangkan (Lee dkk., 2021).

Dalam mendiagnosis penyakit lambung, endoskopi merupakan salah satu metode yang memanfaatkan alat khusus berupa selang tipis yang di ujungnya terdapat kamera kecil yang dimasukkan ke dalam perut melalui mulut atau anus untuk melihat langsung kondisi organ dalam tubuh, seperti lambung. Selain itu, adanya beberapa jenis penyakit lambung yang bervariasi juga menjadi salah satu hal yang penting dalam melakukan diagnosis maupun klasifikasi yang tepat dan

akurat karena tiap penyakit ataupun lesi pada lambung memiliki beberapa karakteristik tersendiri (Hatami dkk., 2020).

Untuk mengatasi masalah ini, maka *Artificial Intelligence* (AI) dapat memainkan peran penting sebagai alat bantu yang sangat berharga. AI telah membuktikan kemajuan signifikan dalam prosedur diagnostik di berbagai bidang medis. Dalam bidang endoskopi, AI dapat dimanfaatkan sebagai alat pendukung untuk meringankan beban para ahli endoskopi, sehingga memungkinkan mereka menganalisis dan menafsirkan hasil dengan lebih efektif (Okagawa dkk., 2022).

Selain itu, pendekatan *Machine Learning* seperti *Deep Learning* juga dapat mengekstraksi data tekstur, warna, bentuk, dan tepi dengan akurasi tinggi serta mengklasifikasikan semua citra endoskopi menurut jenis penyakit yang diwakilinya (Ahmad dkk., 2023). *Deep Learning* telah mencapai performa tinggi untuk berbagai jenis penyakit endoskopi ketika jumlah citra dalam kumpulan data cukup tinggi selama waktu pelatihan (Mukhtorov dkk., 2023). Salah satu cabang utama dari *Deep Learning* adalah *Computer Vision*, yang didukung oleh klasifikasi, deteksi objek, dan analisis citra mikroskopis dalam pencitraan medis yang biasanya menggunakan teknik dasar seperti pengolahan citra (*Image Processing*).

Salah satu metode klasifikasi citra atau objek yang biasa diterapkan adalah *Convolutional Neural Network* (CNN). CNN merupakan salah satu algoritma dari deep learning yang mengekstraksi fitur secara bertahap dari citra melalui lapisan convolutional, kemudian mengurangi dimensi data melalui pooling yang bersifat opsional, dan mengintegrasikan informasi dari fitur yang terdeteksi untuk menghasilkan klasifikasi atau prediksi akhir (Hatami dkk., 2020). Penelitian yang dilakukan oleh (Hatami dkk., 2020) telah mengusulkan metode CNN untuk mengklasifikasikan citra polip menjadi 3 kelas dan hasil yang ditunjukkan pada penelitian menghasilkan peningkatan kinerja sistem hingga 89%. Penelitian lain yang mengusulkan metode CNN untuk klasifikasi kanker lambung dan penyakit lambung lainnya mencapai hasil kurva ROC sebesar 0,96 dan menunjukkan peningkatan dengan melakukan metode augmentasi (Lee dkk., 2021). Pada penelitian ini ditemukan bahwa metode augmentasi dan segmentasi memungkinkan model klasifikasi bekerja lebih baik karena kebijakan augmentasi data yang diterapkan mencegah overfitting. Penelitian lainnya yang dilakukan oleh (Escobar dkk., 2021) yang menerapkan metode transfer learning pada CNN berupa VGG-16 dapat mengklasifikasikan delapan kategori penyakit dan kelainan pada saluran pencernaan dengan akurasi yang cukup tinggi, yaitu 98% dengan waktu training selama 45 menit. Selain VGG16, dalam penelitian (Fitriasari & Rizkinia, 2021) yang membandingkan dan mengusulkan kombinasi metode Xception dan ResNet50V2 untuk mendeteksi COVID-19 melalui citra x-ray menghasilkan akurasi sebesar 93,412%. Akurasi dari model tersebut terbilang menjanjikan, karena kumpulan data yang disiapkan dan diolah dalam jumlah yang sedikit.

Di era mobilitas seperti saat ini, efektivitas, efisiensi, dan kemudahan penggunaan menjadi faktor yang sangat penting. Maka dari itu, dengan melihat permasalahan tersebut penulis mengusulkan penelitian dengan judul "Implementasi

Convolutional Neural Network Untuk Klasifikasi Penyakit Lambung Melalui Citra Endoskopi" dalam pembuatan suatu sistem yang dapat melakukan klasifikasi penyakit lambung dengan memanfaatkan metode *deep learning* menggunakan *CNN*, yang dapat menjadi salah satu roadmap penelitian mengenai pengujian beberapa metode yang berbeda untuk mengukur tingkat akurasi sistem dan juga nantinya dapat diterapkan ketika teknologi robot/autonomous system telah digunakan untuk membantu melakukan endoskopi di masa yang akan mendatang, sebagai sistem pendukung bagi para dokter dalam menganalisis maupun mendiagnosis penyakit lambung.

1.2 Teori

1.2.1 Endoskopi lambung

Endoskopi lambung merupakan alat penting dalam diagnosis dan terapi di bidang gastroenterohepatologi. Prosedur ini menggunakan alat yang disebut endoskop, yaitu sebuah alat berbentuk selang tipis dengan kamera kecil di ujungnya, yang dimasukkan melalui mulut atau anus untuk melihat bagian dalam tubuh. Perkembangan endoskopi dimulai sejak abad ke-19 dan terus mengalami peningkatan dalam hal kecanggihan teknologinya (Nelson, 2020). Perkembangan teknologi, seperti pemanfaatan video endoskopi dan sistem pencitraan yang canggih, telah meningkatkan kualitas visualisasi dalam prosedur medis. Inovasi teknologi, seperti endoskopi kapsul, juga memberikan opsi yang menarik untuk melihat saluran pencernaan tanpa perlu tindakan invasif (Kim & Chun, 2021). Untuk pemeriksaan lambung, endoskopi kini menjadi metode standar dalam mendiagnosis dan mengobati berbagai masalah saluran pencernaan seperti esofagus, lambung, usus, dan rektum. Prosedur ini sering dilakukan untuk mendeteksi kondisi seperti gastritis, ulkus peptikum, penyakit refluks *gastroesofagus* (GERD), tumor, atau kanker lambung (Oka dkk., 2022).

1.2.2 Lambung

Lambung merupakan organ penting dalam sistem pencernaan (Meilani dkk., 2022). Organ ini memiliki bentuk seperti kantong berongga dan terletak di antara esofagus serta usus halus. Secara anatomi, lambung terdiri dari beberapa bagian, yaitu kardia (bagian atas yang terhubung dengan esofagus), fundus (bagian lengkung atas), korpus (bagian utama), dan pilorus (bagian yang menghubungkan dengan usus halus). Dinding lambung terdiri dari lapisan otot yang memungkinkan terjadinya kontraksi untuk mencampur makanan dengan asam lambung dan enzim pencernaan. Lambung berperan penting dalam proses pencernaan makanan baik secara mekanis maupun kimiawi.

Fungsi utama lambung adalah sebagai tempat penyimpanan sementara makanan yang telah dikonsumsi, sebelum dicerna lebih lanjut dan disalurkan ke usus halus untuk proses penyerapan nutrisi. Pencernaan dimulai dengan kontraksi otot lambung yang memecah makanan menjadi bagian-bagian kecil, disertai

dengan sekresi asam lambung dan enzim pencernaan, terutama pepsin, yang berperan dalam memecah protein. Selain itu, lambung dilapisi oleh mukosa yang mengandung kelenjar gastrik yang menghasilkan lendir untuk melindungi dinding lambung dari kerusakan akibat asam yang bersifat korosif.

Penyakit lambung gastritis. Gastritis adalah kondisi di mana terjadi peradangan pada lapisan mukosa lambung, yang bisa bersifat akut atau kronis. Peradangan ini biasanya disebabkan oleh berbagai faktor, seperti iritasi akibat produksi asam lambung yang berlebihan, infeksi bakteri *H. pylori* yang hidup di lambung (Yacob dkk., 2023), serta konsumsi alkohol atau obat-obatan tertentu, seperti *NSAID (Non-Steroidal Anti-Inflammatory Drugs)*. Kondisi ini dapat mengganggu pencernaan dan menimbulkan gejala seperti nyeri perut, mual, muntah, serta rasa tidak nyaman di bagian atas perut (Azer dkk., 2024). Gastritis akut muncul secara tiba-tiba dan biasanya berlangsung dalam jangka pendek, sedangkan gastritis kronis berkembang perlahan dan dapat bertahan lama jika tidak diobati, sebagaimana ditunjukkan pada Gambar 1.



Gambar 1. Penyakit gastritis (RSUP Dr. Wahidin Sudirohusodo, 2021)

Berdasarkan Gambar 1, dapat dilihat bahwa penanganan gastritis harus dilakukan dengan segera, terutama jika penyebabnya adalah infeksi *H. pylori* atau penggunaan obat-obatan yang dapat merusak lapisan mukosa lambung. Apabila tidak ditangani dengan baik, gastritis dapat berkembang menjadi masalah yang lebih serius, seperti tukak lambung atau bahkan kanker lambung. Oleh karena itu, sangat penting bagi seseorang yang mengalami gejala gastritis untuk segera memeriksakan diri dan memperoleh penanganan yang tepat dari tenaga medis.

Penyakit kanker lambung. Kanker lambung adalah kondisi di mana sel-sel ganas tumbuh di dalam lapisan mukosa lambung dan berpotensi menyebar ke bagian tubuh lainnya jika tidak segera diatasi. Penyakit ini cenderung berkembang secara perlahan dan pada tahap awal sering kali tidak menunjukkan gejala yang spesifik. Hal ini menyebabkan banyak pasien baru terdiagnosis ketika kanker telah mencapai stadium lanjut (Rifa'i dkk., 2023). Insiden dan mortalitas kanker lambung menduduki peringkat kedua di antara semua kanker, dan keempat di dunia di antara semua penyakit (L. Li dkk., 2020).

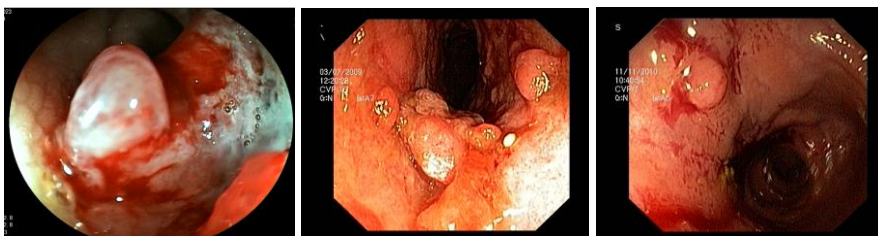
Faktor risiko utama kanker lambung terkait dengan beberapa penyebab, termasuk infeksi kronis oleh bakteri *H. pylori*, yang dianggap sebagai salah satu faktor utama dalam perkembangan kanker lambung (Chudri, 2020). Selain itu, kebiasaan seperti merokok, konsumsi alkohol, serta pola makan yang kaya akan garam dan makanan yang diasinkan, juga dapat meningkatkan risiko terkena kanker lambung. Faktor genetik juga berperan penting, di mana adanya riwayat keluarga yang mengidap kanker lambung dapat meningkatkan kerentanan seseorang terhadap penyakit ini (Chudri, 2020).

Kanker lambung masih menjadi salah satu tantangan utama dalam bidang onkologi, terutama karena sering kali baru terdeteksi pada tahap lanjut. Oleh sebab itu, meningkatkan kesadaran mengenai faktor risiko dan gejala awal, serta melakukan pemeriksaan rutin bagi individu yang berisiko tinggi, sangat penting untuk mengurangi angka kejadian dan kematian akibat penyakit ini. Contoh kasus kanker lambung dapat dilihat pada Gambar 2.



Gambar 2. Penyakit kanker lambung (RSUP Dr. Wahidin Sudirohusodo, 2021)

Penyakit polip lambung. Polip lambung merupakan pertumbuhan jaringan yang tidak normal pada lapisan mukosa lambung. Biasanya, polip ini berukuran kecil dan seringkali tidak menyebabkan gejala apapun. Meski begitu, dalam beberapa kasus, terutama jika ukurannya bertambah besar atau jumlahnya meningkat, polip lambung dapat menimbulkan masalah kesehatan. Penyebab terjadinya polip ini bervariasi, termasuk peradangan kronis, kelainan genetik, atau infeksi bakteri seperti *Helicobacter pylori*, yang pada beberapa orang dapat memicu perkembangan polip. Penggunaan jangka panjang obat penekan asam lambung (seperti penghambat pompa proton) juga bisa menjadi faktor penyebab. Mayoritas polip lambung ditemukan secara kebetulan melalui prosedur endoskopi karena tidak menunjukkan gejala. Polip lambung biasanya dapat dilihat pada Gambar 3.



Gambar 3. Penyakit polip lambung (RSUP Dr. Wahidin Sudirohusodo, 2021)

Berdasarkan Gambar 3. Polip lambung umumnya dideteksi melalui prosedur endoskopi gastrointestinal. Prosedur ini memungkinkan dokter untuk melihat polip secara langsung dan mengambil sampel jaringan (*biopsi*) guna dianalisis secara histopatologi. *Biopsi* diperlukan untuk mengidentifikasi jenis polip dan mengevaluasi risiko keganasannya, karena biopsi merupakan metode standar emas untuk diagnosis kanker lambung (Gullo dkk., 2020). *Adenoma* lambung sendiri dapat berkembang menjadi adenokarsinoma lambung apabila tidak ditangani dengan tepat.

1.2.3 Augmentasi

Augmentasi adalah metode yang digunakan dalam pembelajaran mesin dan visi komputer untuk meningkatkan jumlah serta variasi data pelatihan tanpa harus mengumpulkan data baru. Metode ini melibatkan penerapan transformasi berbeda pada data yang sudah ada, seperti *cropping*, *rotasi*, *flipping*, *scaling*, *zoom*, dan *resize*, untuk menciptakan contoh-contoh data baru yang berbeda namun tetap relevan dengan konteks data asli (Amrulloh dkk., 2024). Teknik augmentasi sangat penting, terutama dalam deep learning, karena algoritma memerlukan data yang besar untuk pelatihan yang optimal. Augmentasi data membantu mengatasi overfitting atau bias terutama pada dataset yang sedikit, dengan meningkatkan variasi data sehingga model lebih robust dan mampu menggeneralisasi pada data baru (Amrulloh dkk., 2024). Teknik ini juga dapat meningkatkan akurasi dan performa model selama pelatihan. Dalam bidang visi komputer, teknik augmentasi data berperan penting dalam meningkatkan kinerja model, khususnya untuk tugas-tugas seperti klasifikasi gambar, deteksi objek, dan segmentasi.

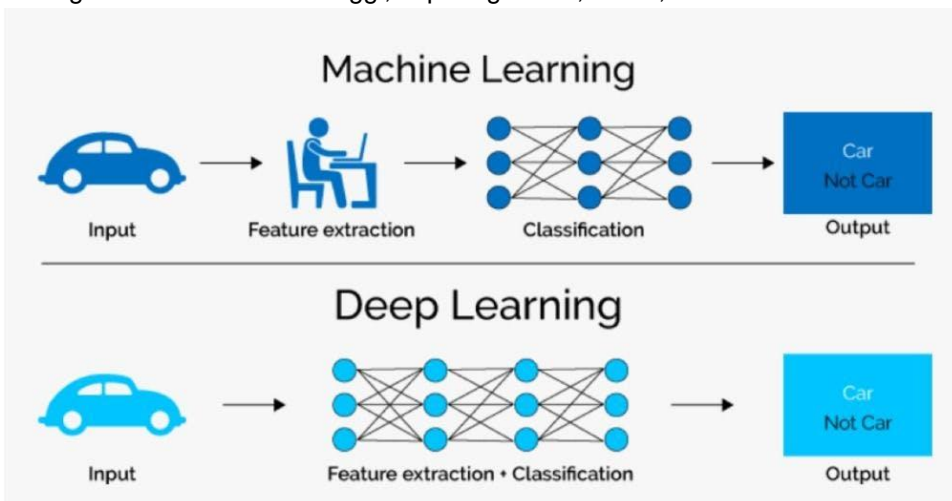
Synthetic Minority Over-sampling Technique (SMOTE). *SMOTE (Synthetic Minority Over-sampling Technique)* adalah sebuah metode yang sering digunakan dalam machine learning untuk menangani masalah ketidakseimbangan data (*imbalanced data*) (Nurdian dkk., 2022). Dalam banyak kasus, dataset yang digunakan untuk pelatihan model memiliki distribusi kelas yang tidak seimbang, di mana jumlah sampel di satu kelas jauh lebih banyak dibandingkan kelas lain. Hal ini bisa menyebabkan model cenderung mengabaikan kelas yang minoritas dan menghasilkan prediksi yang bias.

SMOTE adalah teknik oversampling yang bekerja dengan cara menghasilkan sampel sintesis baru dari kelas minoritas daripada sekadar menduplikasi sampel yang ada. Metode ini menunjukkan secara signifikan dapat meningkatkan kinerja model klasifikasi ketika dihadapkan pada dataset yang tidak seimbang (Nurdian dkk., 2022). Hal ini terutama karena *SMOTE* meningkatkan representasi kelas minoritas dalam pelatihan model tanpa menyebabkan masalah overfitting yang sering muncul dan memungkinkan model untuk belajar pola yang lebih baik terkait kelas tersebut.

SMOTE banyak digunakan dalam berbagai aplikasi seperti prediksi penipuan (fraud detection), diagnosis medis, dan masalah klasifikasi lainnya di mana ketidakseimbangan kelas sering terjadi. Meskipun sangat bermanfaat, *SMOTE* juga memiliki beberapa keterbatasan, seperti kemungkinan menghasilkan data sintesis yang tidak representatif dari distribusi data asli atau tidak mempertimbangkan hubungan antar fitur yang lebih kompleks.

1.2.4 Deep learning

Deep learning adalah bagian dari bidang kecerdasan buatan dan sub bidang dari *machine learning* yang berfokus pada penggunaan jaringan syaraf tiruan (*ANN*) dengan banyak lapisan (*deep neural networks*) untuk memproses dan menganalisis data. Teknik ini terinspirasi oleh cara kerja otak manusia dan dirancang untuk mengekstrak fitur dari data secara otomatis (Sanjaya & Budi, 2020). Deep learning dibangun berdasarkan prinsip pembelajaran lapis demi lapis, di mana lapisan pertama mempelajari fitur sederhana (seperti tepi atau garis dalam pengenalan gambar), sementara lapisan-lapisan selanjutnya mengekstraksi informasi yang lebih abstrak. Kemampuan ini membuat deep learning sangat efektif dalam menangani data berstruktur tinggi, seperti gambar, suara, dan teks.



Gambar 4. Perbandingan *machine learning* dan *deep learning* (Zhang dkk., 2022)

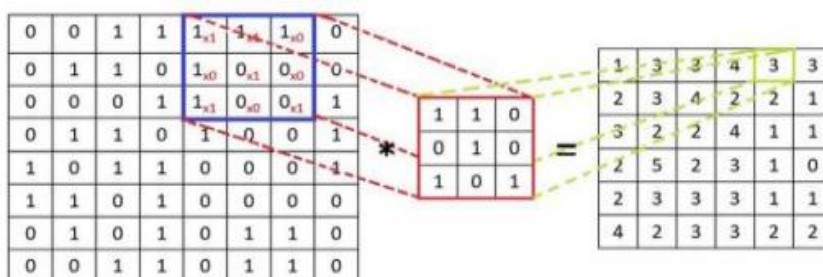
Berdasarkan Gambar 4, dapat kita lihat bahwa Machine learning dan deep learning menerima input yang serupa, namun berbeda dalam cara memperoleh fitur. Pada machine learning, manusia perlu menentukan fitur yang relevan untuk digunakan, sementara deep learning secara otomatis melakukan feature engineering yang mendalam dan kompleks, dan menghasilkan berbagai variasi fitur tanpa intervensi manusia. Dalam konteks machine learning, deep learning telah menciptakan peluang baru dalam pengolahan data, terutama dalam analisis gambar dan data berurutan. Contoh metode yang berhasil dalam bidang ini adalah *CNN* dan *RNN*. Namun, deep learning juga menghadapi beberapa tantangan, seperti kesulitan dalam menginterpretasikan keputusan model, kebutuhan akan data besar untuk pelatihan, serta kompleksitas dalam penyesuaian parameter model. Oleh karena itu, deep learning dapat dipahami sebagai "proses membuat komputer belajar seperti manusia", di mana model-model ini dapat belajar secara mandiri dari data besar untuk menghasilkan prediksi yang lebih cerdas dan akurat. Deep learning telah menghasilkan kemajuan signifikan dalam berbagai bidang, termasuk pengenalan wajah, mobil otonom, penelitian medis, dan masih banyak lainnya.

1.2.5 Convolutional neural network (CNN)

Convolutional neural network (CNN) adalah tipe jaringan saraf yang umum digunakan untuk memproses data berupa gambar dan video. *CNN* pertama kali diperkenalkan oleh Yann LeCun pada tahun 1980-an dan dipopulerkan dengan penggunaan dalam pengenalan tulisan tangan pada dataset *MNIST* (Budi dkk., 2024). *CNN* digunakan untuk mendeteksi dan mengenali objek dalam gambar, terinspirasi dari cara manusia memproses informasi visual. Secara umum, *CNN* memiliki struktur mirip seperti jaringan saraf biasa dengan neuron yang memiliki bobot, bias, dan fungsi aktivasi. Namun, pembeda *CNN* dengan lainnya adalah lapisan konvolusi, terdiri dari neuron-neuron yang tersusun dalam filter dengan dimensi panjang dan tinggi (piksel).

Dengan arsitektur *CNN* kita dimudahkan untuk melakukan tugas-tugas Computer Vision seperti pengenalan objek, segmentasi gambar, deteksi objek, maupun klasifikasi dikarenakan *CNN* memiliki *Convolutional Network* yang dapat mengekstraksi fitur dari gambar inputnya dan juga memiliki *Fully Connected Layer* yang bisa mengklasifikasikan output dari Convolutional Network berupa *feature map* dari gambar yang diinput seperti yang dapat dilihat pada Gambar 5.

3x3, dan seterusnya, yang digeser atau di convolved untuk mencari pola tertentu. Lapisan *convolutional* menggunakan filter kernel untuk menghitung *convolution* dari gambar input, dan mengekstraksi fitur-fitur dasar. Filter kernel memiliki ukuran dimensi yang sama tetapi dengan nilai parameter konstan yang lebih kecil daripada gambar input (Singh dkk., 2020). Sebagai contoh, panjang yang dapat diterima dari filter kernel untuk sebuah skelogram 2D dengan ukuran 35x35x35 adalah $f \times f \times 2$, di mana $f = 3, 5, 7$, dan seterusnya. Namun, ukuran filter harus lebih kecil dari ukuran gambar input. Masker filter bergerak melintasi gambar input langkah demi langkah dan menghitung titik produk antara bobot filter kernel dan nilai piksel dari gambar input. Proses ini menghasilkan peta aktivasi 2D. *CNN* kemudian akan mempelajari fitur visual dari gambar tersebut.



Gambar 7. Ilustrasi proses konvolusi (Purwono dkk., 2022)

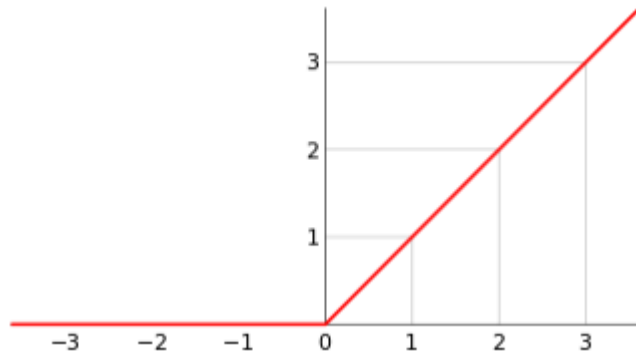
Seperti yang dapat dilihat pada Gambar 7 menunjukkan ilustrasi sederhana dari proses komputasi pada *CNN* yang menghasilkan peta aktivasi. Sebuah *convolutional layer* ditentukan oleh ukuran kernel, panjang langkah, dan padding (Suha & Sanam, 2022). Ukuran kernel merupakan ukuran filter kernel atau kernel geser (Ding dkk., 2018). Panjang langkah adalah jumlah kernel yang bergeser sebelum menghasilkan titik produk dan menciptakan piksel output (Riad dkk., 2022). Padding adalah ukuran bingkai 0-th yang dipasang di sekitar peta fitur input (Nguyen dkk., 2019).

Activation function. Fungsi aktivasi adalah fungsi yang diterapkan dalam jaringan saraf untuk menentukan apakah suatu neuron akan diaktifkan atau tidak. Fungsi ini digunakan untuk memperkenalkan non-linearitas ke dalam jaringan, sehingga variasi data yang dihasilkan dapat menjadi lebih rinci dan kompleks. Adapun beberapa fungsi aktivasi yang sering digunakan yaitu fungsi aktivasi *ReLU*, *Softmax*, dan *Sigmoid*.

a. Fungsi aktivasi *ReLU (Rectified Linear Unit)*

Fungsi aktivasi *ReLU* adalah fungsi aktivasi linear yang mengatasi masalah hilangnya gradien dan mengembalikan nilai input secara langsung jika input tersebut positif, tetapi jika input negatif maka fungsi ini mengembalikan nilai 0. Dengan kata lain, feature map yang dihasilkan dari proses konvolusi,

kemudian akan diterapkan fungsi aktivasi ReLu yang akan mengubah semua nilai yang lebih kecil atau sama dengan nol menjadi 0 dan mempertahankan nilai yang lebih besar dari 0. Adapun kelebihan dari fungsi ini yaitu memiliki efisiensi tinggi. Fungsi ini biasanya didefinisikan seperti Gambar 8.



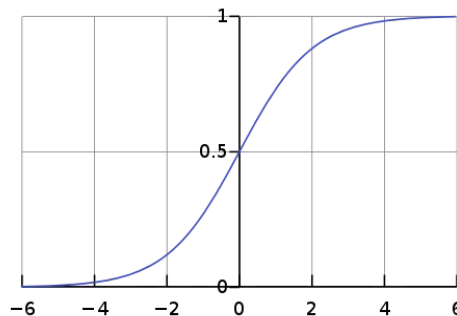
Gambar 8. Grafik fungsi aktivasi ReLu

Adapun persamaan dari fungsi aktivasi ini bisa kita lihat pada persamaan 1.

$$ReLU(x) = \max(0, x) \quad (1)$$

b. Fungsi aktivasi *Softmax*

Fungsi aktivasi *Softmax* merupakan suatu fungsi aktivasi yang mengubah vektor nilai mentah menjadi vektor probabilitas kategorikal, di mana setiap elemen berada dalam rentang (0, 1) dan jumlahnya adalah 1. Fungsi ini sering digunakan sebagai aktivasi pada lapisan terakhir jaringan klasifikasi, dikarenakan hasilnya tersebut dapat diinterpretasikan sebagai distribusi probabilitas, terutama dalam tugas klasifikasi multi-kelas. Grafiknya dapat dilihat pada Gambar 9.



Gambar 9. Grafik fungsi aktivasi *Softmax* (S, 2021)

Adapun persamaan dari fungsi aktivasi ini yaitu :

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \quad (2)$$

for $i = 1, 2, \dots, k$ dan $z = z_1, z_2, \dots, z_k$

Ket :

$\sigma(z_i)$: hasil probabilitas

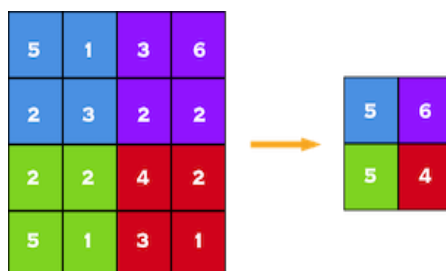
e^{z_i} : eksponensial dari neuron label ke-i

$\sum_{j=1}^k e^{z_j}$: jumlah dari e^{z_j} dari $j = 1$ sampai k

Fungsi eksponensial diterapkan pada setiap elemen z_i dari vektor input z dan menormalkan nilai-nilai tersebut dengan membaginya dengan total dari semua nilai eksponensial. Proses normalisasi ini memastikan bahwa total komponen dalam vektor output adalah 1. Untuk setiap kelas i , di mana z adalah output dari neuron sebelum fungsi aktivasi diterapkan.

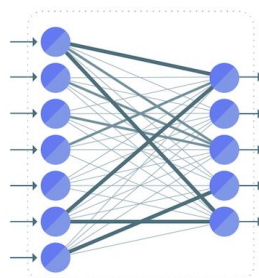
Global Average Pooling. *Global Average Pooling* adalah salah satu layer yang berperan penting dalam arsitektur *CNN* terkait kompleksitas dan jumlah parameter yang tinggi dalam lapisan jaringan. (Gong dkk., 2020). *GAP* bekerja dengan menghitung rata-rata seluruh nilai dalam setiap peta fitur dalam lapisan berikutnya dan sangat membantu untuk mengurangi *overfitting* model. *GAP* sangat membantu dalam meminimalkan kebutuhan parameter dalam layer *CNN* yang mereduksi dimensi seluruh pada peta fitur menjadi satu nilai agar membuat model yang dilatih siap untuk lapisan klasifikasi akhir. Dengan demikian, *GAP* membantu untuk stabilisasi akurasi validasi, mengurangi *overfitting* model, dan waktu komputasi keseluruhan pada model *CNN* (Habib & Qureshi, 2022).

Pooling Layer. *Pooling Layer* merupakan layer yang berfungsi untuk mengurangi dimensi data, baik panjang maupun lebar dari feature map, dengan tujuan mengurangi jumlah parameter, sehingga dapat mencegah *overfitting* dan juga mempercepat komputasi. Dengan kata lain, layer ini sering digunakan untuk menyederhanakan komputasi tanpa menghilangkan informasi penting. Lapisan *pooling* ini menggabungkan nilai piksel untuk mengurangi kompleksitas. Meskipun demikian, tidak berdampak negatif pada performa model. Pemilihan *pooling* dapat didasarkan pada nilai maksimum, rata-rata, atau total dari nilai piksel. Ada dua metode yang biasa diterapkan dalam lapisan *pooling*, yaitu *max-pooling* dan *average-pooling*. Perbedaannya dapat dilihat pada nilai entitas yang diambil, dimana *max-pooling* mengambil nilai entitas yang paling besar dengan ukuran kernel tertentu sedangkan, *average-pooling* menghitung rata-rata dari nilai entitas yang tersebar dengan ukuran kernel tertentu. Tetapi, metode *pooling* yang paling umum digunakan adalah *max-pooling*, contohnya dapat dilihat pada Gambar 10.



Gambar 10. Proses Max-Pooling (Yalçın, 2021)

Fully Connected Layer. *Fully Connected Layer* adalah suatu lapisan yang berfungsi untuk menghubungkan setiap neuron pada satu lapisan ke lapisan berikutnya. Lapisan ini menggabungkan setiap informasi yang didapatkan pada lapisan yang ada sebelumnya (*feature map*). Seperti yang diperlihatkan pada Gambar 11, layer ini dapat ditemukan di berbagai jenis jaringan saraf tiruan, termasuk Neural Nets biasa hingga *CNN*. Layer ini sering digunakan dalam eksperimen machine learning dan umumnya berperan sebagai pengklasifikasi gambar dalam *CNN*.



Gambar 11. *Fully Connected Layer* (Isaksson, 2020)

Pada *Fully Connected Layer*, terdapat sebuah fungsi yang sering digunakan beserta hyperparameter yang dapat diatur, yaitu *Dense Layer* atau sering disebut juga *Linear Layer*. *Dense/Linear layer* merupakan salah satu layer yang paling umum digunakan dalam *Fully Connected Layer* karena merupakan layer yang benar-benar terhubung dalam jaringan neural. Kita juga bisa mengatur hyperparameter dari fungsi ini, seperti menentukan jumlah neuron dalam satu layer dan jumlah hidden layer yang akan dibuat.

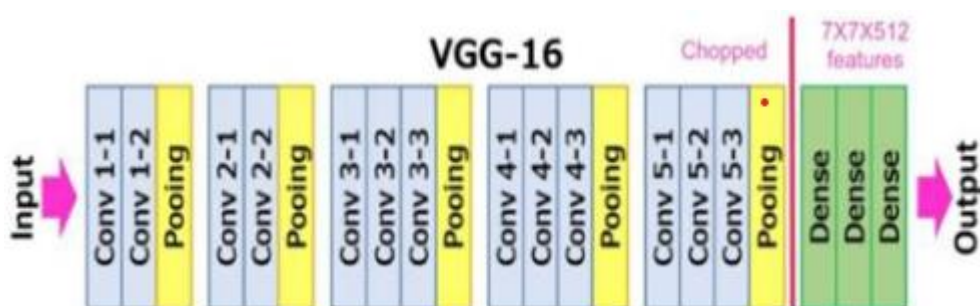
1.2.6 *Transfer learning*

Transfer learning adalah metode pembelajaran mesin di mana model yang telah dilatih untuk satu tugas digunakan kembali (atau ditransfer) ke tugas lain yang serupa, terutama dalam kasus di mana data pelatihan untuk tugas baru terbatas. Dengan menggunakan pengetahuan dari model yang sudah dilatih, transfer learning dapat mempercepat proses pelatihan dan meningkatkan kinerja pada

domain yang memiliki data terbatas (Siregar dkk., 2024). Transfer learning memiliki aplikasi luas di berbagai bidang, terutama ketika data pelatihan pada domain baru tidak mencukupi. Salah satu penggunaan utamanya adalah dalam visi komputer, di mana model-model besar seperti *CNN* yang telah dilatih pada dataset besar seperti ImageNet dapat digunakan kembali pada tugas-tugas lain seperti deteksi objek atau klasifikasi citra medis, karena melalui pendekatan ini, dapat meningkatkan akurasi untuk mendeteksi penyakit dengan memanfaatkan fitur-fitur mendalam yang diekstrak dari citra medis (Siregar dkk., 2024). Karena alasan tersebut, transfer learning sangat menguntungkan di bidang medis.

Dalam beberapa tahun terakhir, penelitian lebih lanjut tentang transfer learning telah mengeksplorasi penerapan di berbagai domain baru seperti klasifikasi dan deteksi penyakit lambung melalui citra medis ataupun prediksi penyakit melalui pemrosesan sinyal biomedis (Escobar dkk., 2021). Penggunaan *transfer learning* dalam domain-domain ini memperlihatkan kemampuannya dalam mempercepat perkembangan teknologi dalam bidang kesehatan.

1.2.7 VGG16

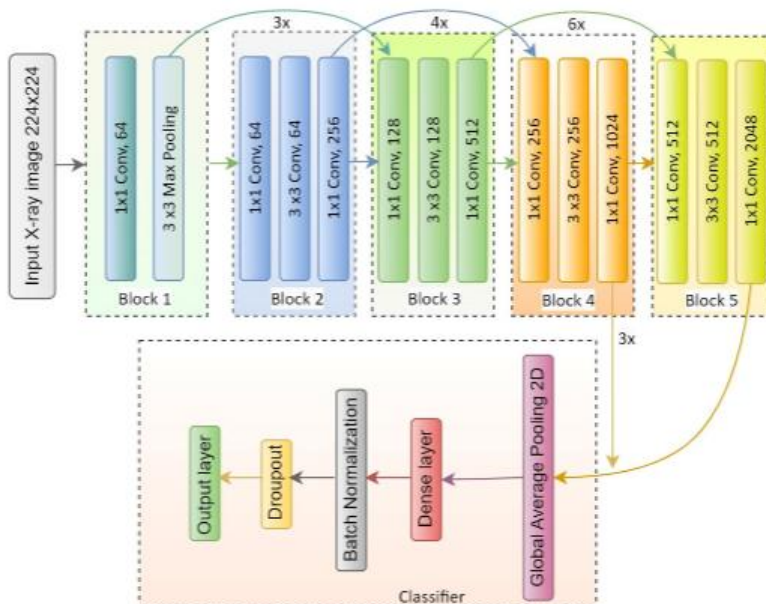


Gambar 12. Arsitektur umum VGG16 (Sulistiyowati dkk., 2023)

VGG16 adalah salah satu arsitektur deep learning yang sangat populer dan digunakan secara luas dalam pengenalan citra. Dikembangkan oleh *Visual Geometry Group* dari University of Oxford, arsitektur ini diperkenalkan dalam kompetisi ImageNet pada tahun 2014 (Muhammad dkk., 2023). *VGG16* dikenal dengan struktur *CNN* yang mendalam namun sederhana, dengan 16 lapisan yang dapat dilatih, termasuk 13 lapisan konvolusi dan 3 lapisan *fully connected*. *VGG16* telah terbukti unggul dalam berbagai tugas pengenalan citra, seperti klasifikasi dan deteksi objek. Meskipun arsitekturnya sederhana, *VGG16* memiliki kinerja yang sangat baik berkat penggunaan lapisan konvolusi kecil dengan filter berukuran 3x3. Ini memungkinkan jaringan untuk menangkap fitur citra dengan lebih baik dibandingkan jaringan yang menggunakan filter besar. Arsitektur ini dikenal karena desainnya yang sederhana namun sangat efektif, dengan fokus pada penggunaan lapisan konvolusi kecil secara berulang untuk meningkatkan kedalaman dan kapasitas jaringan. Namun, kekurangan utama *VGG16* adalah ukuran model yang

besar dan waktu pelatihan yang lama karena memiliki banyak parameter, yang menuntut daya komputasi tinggi (Hutamaputra dkk., 2022).

1.2.8 ResNet50V2



Gambar 13. Arsitektur umum ResNet50V2 (Chakraborty dkk., 2023)

Dalam bidang visi komputer, *ResNet50V2* merupakan salah satu arsitektur *CNN* yang paling populer. *ResNet50V2* adalah arsitektur jaringan konvolusi modern yang lebih mudah dilatih dibandingkan jaringan konvolusi dalam lainnya, menghasilkan akurasi yang lebih tinggi, dan mencapai hasil akhir lebih cepat. (Fitriasari & Rizkinia, 2021). Angka 50 merujuk pada jumlah lapisan dalam arsitektur ini, yang mencakup lapisan konvolusi, normalisasi batch, dan fungsi aktivasi *ReLU*. Blok residual, sebagai elemen penting dari *ResNet50V2*, memungkinkan jaringan mengatasi masalah penurunan kinerja yang sering terjadi pada jaringan yang lebih dalam. Setiap blok residual terdiri dari jalur utama yang melibatkan beberapa lapisan konvolusi, serta koneksi langsung antara input dan output melalui blok samping. Koneksi ini mendukung "pembelajaran residual," memungkinkan jaringan untuk fokus pada perbedaan atau atribut tambahan dalam gambar.

ResNet50V2 mengadopsi arsitektur dengan "*shortcut connections*" atau "*skip connections*", yang memperlancar propagasi gradien selama pelatihan dan mengurangi masalah gradien yang menghilang atau meledak dalam jaringan saraf dalam. Selain itu, normalisasi batch yang diterapkan dalam *ResNet50V2* membantu mempercepat konvergensi dan mengurangi risiko *overfitting*. Dengan menerapkan normalisasi data pada setiap batch, variansnya dapat diminimalkan, yang mendukung pembelajaran yang lebih stabil bagi jaringan. Kombinasi normalisasi

batch, blok residual, dan *shortcut* connections membuat *ResNet50V2* menjadi arsitektur yang sangat efektif untuk pengenalan gambar. Arsitektur ini banyak digunakan dalam berbagai aplikasi, seperti klasifikasi gambar, deteksi objek, dan segmentasi, karena kemampuannya dalam menghasilkan representasi yang kompleks dan mendalam dari data visual.

1.2.9 Confusion matrix

Confusion matrix merupakan suatu metode yang umum digunakan untuk mengevaluasi kinerja dari suatu model, khususnya model klasifikasi. Matriks tersebut terdiri dari susunan tabel khusus yang memperlihatkan hubungan antara prediksi yang dibuat oleh model dan hasil sebenarnya dari data, dengan menampilkan jumlah instance di setiap kategori berdasarkan prediksi yang benar atau salah. Setiap baris pada matriks mewakili kelas yang diprediksi, sedangkan kolom pada matriks mewakili kelas aktual (*ground-truth*). Matriks ini terbagi menjadi empat komponen utama yang bisa kita lihat pada Gambar 15. Yaitu, *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*, yang memungkinkan analisis kinerja model secara menyeluruh (Puspitasari dkk., 2023). Matriks ini sangat berguna untuk mengukur *precision*, *recall*, akurasi dan skor F1, apabila label klasifikasi yang digunakan lebih dari dua maka confusion matrix akan berukuran sesuai dengan jumlah labelnya juga seperti pada Gambar 14.

		True Class			
		A	B	C	D
Predicted Class	A	TP_A	E_{BA}	E_{CA}	E_{DA}
	B	E_{AB}	TP_B	E_{CB}	E_{DB}
	C	E_{AC}	E_{BC}	TP_C	E_{DC}
	D	E_{AD}	E_{BD}	E_{CD}	TP_D

Gambar 14. *Confusion Matrix Multiclass*

Agar bisa menghitung nilai akurasi, presisi, recall dan f1-score maka confusion matrix multiclass untuk setiap kelasnya akan dibentuk confusion matrix 2x2 seperti pada Gambar 15. Kemudian melakukan perhitungan pada parameter *TP*, *FP*, *FN* dan *TN* dimana parameter ini adalah hubungan antara nilai prediksi dan nilai aktual.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Gambar 15. Struktur *Confusion Matrix* (Nugroho, 2020)

Dalam matriks tersebut terdapat empat komponen utama meliputi:

1. *True Positives* (TP) : Jumlah kasus yang labelnya positif dan diprediksi sebagai positif.
2. *False Positives* (FP) : Jumlah kasus yang label sebenarnya adalah negatif tetapi diprediksi sebagai positif.
3. *True Negatives* (TN) : Jumlah kasus yang labelnya negatif dan diprediksi sebagai negatif.
4. *False Negatives* (FN) : Jumlah kasus yang label sebenarnya adalah positif tetapi diprediksi sebagai negatif.

1.2.9 Akurasi, presisi, *recall*, dan *F1-Score*

Dalam evaluasi model *machine learning*, khususnya klasifikasi, metrik evaluasi sangat penting untuk menilai kinerja model. Metrik yang umum digunakan termasuk akurasi, presisi, *recall*, dan *F1-Score* (Rahayu dkk., 2021), yang masing-masing memiliki fungsi berbeda dalam mengukur hasil prediksi terhadap data aktual.

- a. Akurasi (*accuracy*): mengukur sejauh mana model benar-benar melakukan prediksi yang sesuai yang ditandai dengan hasil klasifikasi data benar. Persamaan 1 untuk menghitung akurasi adalah sebagai berikut:

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

- b. Presisi (*precision*): mengukur sejauh mana kebenaran hasil label positif yang diprediksi oleh model yakni menilai berapa persen prediksi positif yang benar-benar akurat. Persamaan 2 untuk menghitung presisi adalah sebagai berikut:

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

- c. *Recall* (Sensitivitas): mengukur sejauh mana model dapat mengidentifikasi semua kasus positif yang sebenarnya terjadi. Persamaan 3 untuk menghitung sensitivitas adalah sebagai berikut:

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

- d. *F1-Score*: merupakan gabungan antara *precision* dan *recall* menjadi satu angka tunggal. Persamaan 4 untuk menghitung *F1-score* adalah sebagai berikut:

$$F1 - Score = \frac{2 \times Presisi \times recall}{Presisi + Recall} \quad (6)$$

1.3 Rumusan Masalah

Berdasarkan latar belakang, maka rumusan masalah yang timbul adalah berikut :

1. Bagaimana cara mengimplementasikan *Deep Learning* dalam melakukan klasifikasi penyakit lambung melalui citra endoskopi menggunakan CNN?
2. Bagaimana tingkat akurasi yang dihasilkan sistem dalam melakukan klasifikasi penyakit lambung melalui citra endoskopi menggunakan CNN?

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Mengimplementasikan *Deep Learning* dalam melakukan klasifikasi penyakit lambung dengan menggunakan CNN.
2. Mengetahui tingkat akurasi sistem dalam klasifikasi penyakit lambung dengan menggunakan CNN.

1.5 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat sebagai berikut:

1. Menjadi salah satu *roadmap* penelitian mengenai pengujian beberapa metode yang berbeda untuk mengukur tingkat akurasi sistem.
2. Membantu para tenaga medis seperti dokter dalam melakukan analisis citra endoskopi lambung melalui hasil klasifikasi pada sistem di masa yang akan datang.
3. Menghasilkan sebuah sistem yang dapat mendukung hasil diagnosa para dokter dan memberikan informasi sebagai pendukung pengambilan keputusan atau tindakan untuk pengendalian maupun pencegahan penyakit lambung di masa yang akan datang.

1.6 Ruang Lingkup

Penelitian ini memiliki beberapa batasan, yaitu:

1. Parameter yang digunakan yaitu citra endoskopi lambung yang normal dan terkena lesi non-kanker, prakanker maupun kanker.
2. Metode yang digunakan yaitu *Convolutional Neural Network* (CNN).
3. Jenis penyakit lambung yang dideteksi yaitu gastritis, polip dan kanker untuk dilakukan klasifikasi.

BAB II METODE PENELITIAN

2.1 Lokasi dan Waktu Penelitian

Penelitian ini dilakukan sejak disetujuinya proposal pada tanggal 13 Mei 2024 hingga 15 November 2024. Penelitian ini dilakukan di Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin. Adapun tempat pengumpulan data primer yaitu di Rumah Sakit Umum Pusat Dr. Wahidin Sudirohusodo, Poli Gastroenterohepatologi yang dimulai dari tanggal 29 Mei hingga 29 Agustus 2024.

2.2 Instrumen Penelitian

Pada penelitian ini digunakan dua jenis instrumen yaitu instrumen perangkat lunak dan instrumen perangkat keras. Berikut rincian mengenai setiap instrumen penelitian yang digunakan.

2.3.1 Instrumen perangkat lunak

Perangkat lunak yang digunakan pada penelitian ini adalah sebagai berikut :

1. Kaggle Notebook.
2. Microsoft Office (Word dan PowerPoint).
3. Windows 10 Pro 64-bit.
4. Windows 11 64-bit.
5. Python.
6. Beberapa library pendukung seperti TensorFlow.

2.3.2 Instrumen perangkat keras

Perangkat keras yang digunakan selama penelitian adalah sebagai berikut :

1. Laptop Asus K46CM Intel® Core™ i5-3317U CPU 1.70 GHz NVIDIA® GeForce GT 635M, RAM 4 GB serta menggunakan penyimpanan HDD 500 GB.
2. Laptop Acer Nitro-AN515-57 Intel® Core™ i7-11800H CPU 2.30GHz NVIDIA® GeForce RTX3050, RAM 16 GB.

2.3 Tahapan Penelitian

Penelitian ini terdiri dari beberapa tahapan yang dapat dilihat pada Gambar 16.



Gambar 16. Alur tahapan penelitian

Berdasarkan Gambar 16 dapat dilihat bahwa penelitian ini memiliki enam tahap, dimana tahapan penelitian yang dilaksanakan adalah sebagai berikut.

1. Studi Literatur, Pada tahapan ini, peneliti melakukan pencarian sumber-sumber literatur mengenai teori-teori terkait dengan topik penelitian mengenai penyakit lambung atau aplikasi lain yang relevan dalam bidang visi komputer dan pembelajaran mesin, dengan penekanan khusus pada aplikasi yang terkait dengan data penyakit lambung.
2. Menentukan Metode, Pada tahapan ini, peneliti akan menentukan metode yang akan digunakan untuk menyelesaikan masalah yang ada. Berdasarkan literatur yang telah dicari, peneliti memilih metode *Convolutional Neural Network from scratch*. Selain itu, peneliti juga akan menggunakan *Transfer Learning* yaitu *VGG16* dan *ResNet50V2* yang kemudian akan dibandingkan dengan metode sebelumnya untuk melihat akurasi yang paling bagus. Kemudian model tersebut akan dikonversi menjadi model *H5*.
3. Pengumpulan Dataset, Peneliti akan melakukan pengumpulan dataset berupa citra endoskopi lambung yang akan digunakan untuk proses membangun sebuah model. Dataset ini, diperoleh langsung di Rumah Sakit Umum Pusat Dr. Wahidin Sudirohusodo, Kota Makassar sebagai data primer dan juga di Platform Kaggle sebagai data sekunder.
4. Perancangan Sistem, Pada tahapan ini, peneliti akan melakukan penyusunan konsep dan alur program, setelah itu akan diimplementasikan sesuai konsep dan alur yang telah dibuat sebelumnya. Disini, Model yang dibuat menggunakan bahasa pemrograman *python* dengan bantuan *library TensorFlow*.
5. Uji Coba Sistem dan Analisis Hasil Penelitian, Pada tahapan ini, yang akan dilakukan oleh peneliti adalah uji coba terhadap model yang dibuat dengan menggunakan *confusion matrix* dan juga mengimplementasikan model pada website. Kemudian akan dilakukan analisis terhadap performa sistem yang telah dibuat dan menyimpulkan hasil dari penelitian yang telah dilakukan.

6. Penyusunan Laporan, Penyusunan laporan akan dilakukan dalam bentuk penulisan seluruh proses penelitian yang telah dilakukan dalam bentuk skripsi sebagai bahan publikasi yang terfokus pada pembelajaran mesin untuk klasifikasi penyakit lambung.

2.4 Teknik Pengambilan Data

Data yang digunakan dalam penelitian ini adalah data citra endoskopi gastrointestinal yang berfokus pada bagian lambung. Data primer diambil pada tanggal 29 Mei 2024 – 29 Agustus 2024, yang diperoleh dari Rumah Sakit Umum Pusat Dr. Wahidin Sudirohusodo, Poli Gastroenterohepatologi, Kota Makassar.

Adapun tahapan pengambilan data yang dilakukan di RSUP Dr. Wahidin Sudirohusodo yaitu:

- a. Pengurusan izin
Untuk proses pengambilan data di Rumah Sakit Umum Pusat Dr. Wahidin Sudirohusodo, peneliti memerlukan beberapa izin dari pihak rumah sakit seperti, persetujuan etik dari Komisi Etik Fakultas Kedokteran, dan izin dari DIKLIT rumah sakit yang berupa surat izin.
- b. Pengambilan Data Rekam Medis
Dalam pengambilan citra endoskopi lambung memerlukan informasi kondisi lambung pasien, sehingga peneliti membuka data rekam medis pasien sebagai validasi agar mengetahui kondisi lambung pasien.
- c. Pengambilan Data Citra Endoskopi Lambung
Peneliti dapat mengumpulkan 373 citra endoskopi lambung yang masing - masing mencakup 93 kasus kanker, 152 kasus gastritis, dan 128 kasus polip, yang dibantu oleh staf poli gastroenterohepatologi RSUP Dr. Wahidin Sudirohusodo. Adapun data yang diperoleh dalam format .PNG.

2.4.1 Karakteristik data

Berdasarkan penjelasan pada bab 2 terkait penyakit lambung yang meliputi kanker, gastritis, dan polip serta hasil observasi di lapangan berupa wawancara dengan dokter ahli terkhusus penyakit lambung, karakteristik utama dari data yang telah diperoleh yaitu:

- a. Lambung Normal
Lambung yang normal memiliki permukaan mukosa yang tampak bersih tanpa kelainan seperti luka ataupun kondisi yang lainnya.
- b. Lambung Kanker
Secara umum, lambung yang terkena kanker ditandai dengan adanya kelainan berupa hipervaskularisasi atau pendarahan yang banyak disertai dengan adanya nanah. Pada mukosa lambung yang terkena kanker juga terdapat massa atau benjolan, serta luka serius pada mukosa lambung atau bagian tertentu pada lambung.
- c. Lambung Gastritis

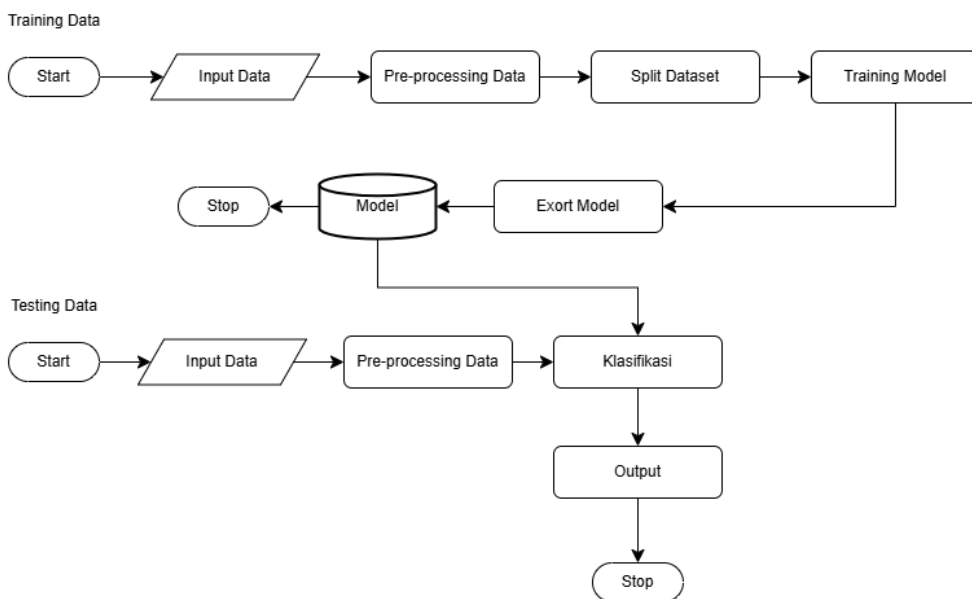
Secara umum, gastritis ditandai oleh adanya inflamasi atau peradangan dan juga erosi atau terjadinya pengikisan pada mukosa lambung yang dapat mengakibatkan luka.

d. Lambung Polip

Secara umum, polip ditandai dengan adanya massa atau benjolan baik yang masih kecil (jinak) ataupun besar (ganas). Penyakit ini juga ditandai dengan adanya pendarahan pada beberapa kasus.

2.5 Perancangan Sistem

Perancangan sistem ini bertujuan untuk menggambarkan proses sistem yang akan dibuat dan dikembangkan, sekaligus untuk memperjelas secara detail tahapan prosesnya. Adapun diagram nya dapat dilihat pada gambar 17.



Gambar 17. Flowchart perancangan sistem

Berdasarkan Gambar 17 perancangan sistem terbagi menjadi dua tahap utama, yaitu proses *training* dan *testing* pada model.

2.5.1 Input data

Data yang digunakan pada penelitian ini terbagi menjadi dua, yaitu data primer dan data sekunder. Adapun skenario dataset yang digunakan disini yaitu memakai 4 kelas label dataset dan 3 kelas label dataset sebagai input. Adapun 1 kelas label dataset yang dihilangkan yaitu kelas polip, yang bertujuan untuk mengurangi

adanya keadaan overfitting pada proses training. Hal ini karena citra yang ditunjukkan oleh kelas polip memiliki banyak noise atau elemen tambahan pada citra serta banyak kesamaan antara dataset dari kelas kanker.



Gambar 18. Citra polip yang memiliki noise

Data primer diperoleh secara langsung di RSUP Dr. Wahidin Sudirohusodo sebanyak 373 citra untuk 3 kelas, yaitu masing – masing 93 citra kanker, 128 citra polip, dan 152 untuk citra gastritis. Contoh data citra endoskopi lambung untuk setiap kelas penyakit dapat dilihat pada Gambar 19.

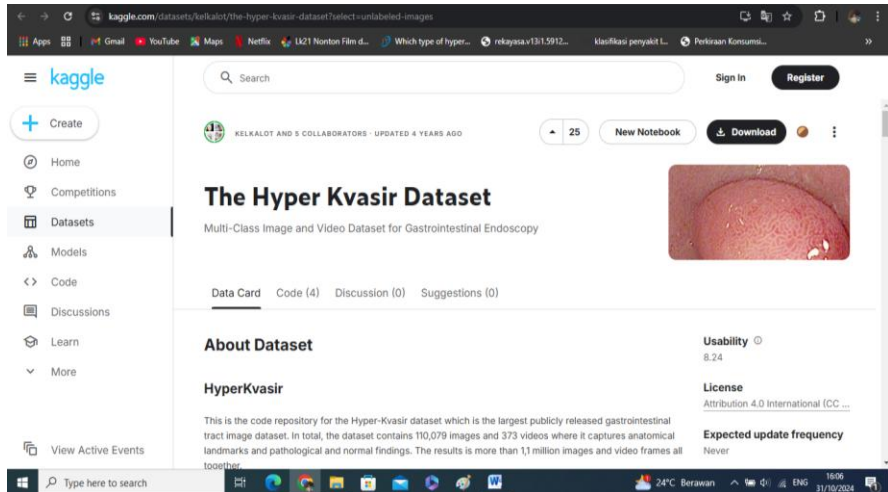


(a) Kelas Gastritis

a. Lambung Normal

Sumber data: <https://www.kaggle.com/datasets/kelkalot/the-hyper-kvasir-dataset?select=unlabeled-images>

Jumlah data yang tersedia adalah sebanyak 1.763 citra. Adapun data yang diambil untuk kemudian diolah adalah sebanyak 1.191 citra

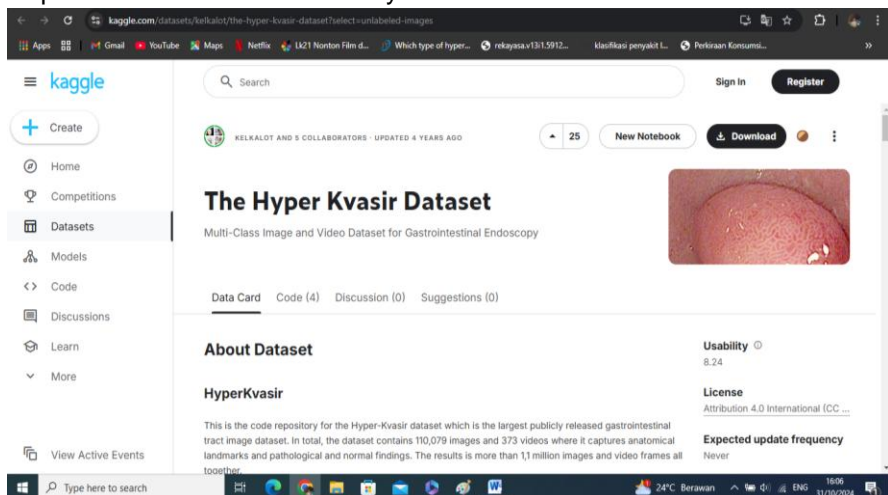


Gambar 20. Sumber data sekunder lambung normal

b. Lambung gastritis

Sumber data: <https://www.kaggle.com/datasets/kelkalot/the-hyper-kvasir-dataset?select=unlabeled-images>

Jumlah data yang tersedia adalah sebanyak 99.400 citra. Adapun data yang dipilih dan diolah adalah sebanyak 165 citra.

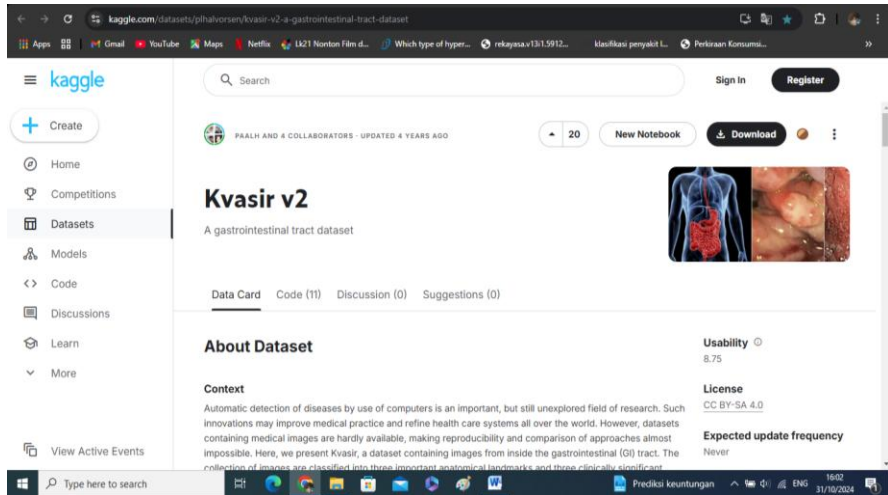


Gambar 21. Sumber data sekunder lambung gastritis

c. Lambung Polip

Sumber data: <https://www.kaggle.com/datasets/plhalvorsen/kvasir-v2-a-gastrointestinal-tract-dataset>

Jumlah data: 1191 Citra



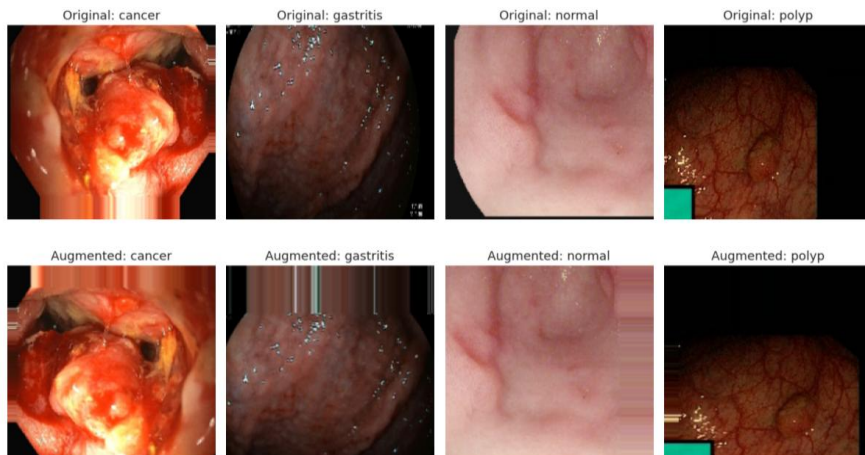
Gambar 22. Sumber data sekunder lambung polip

2.5.2 Preprocessing data

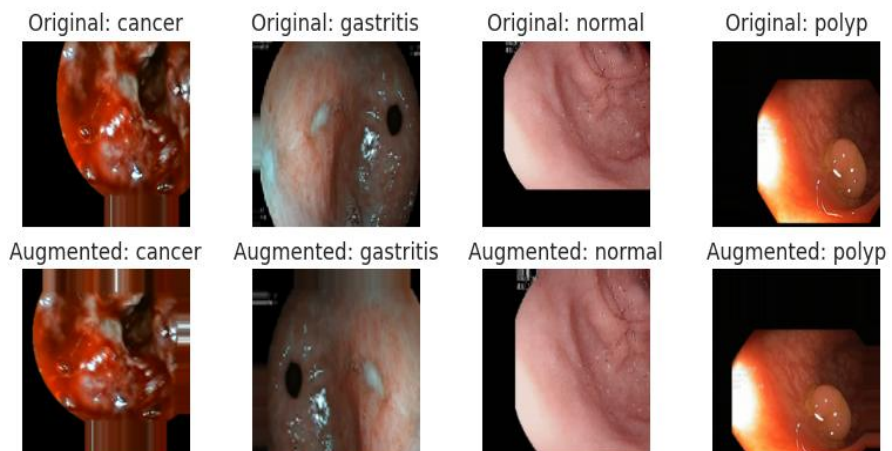
Preprocessing data yang dilakukan pada penelitian ini menggunakan inputan gambar 224 x 224 piksel untuk memastikan dimensi input yang seragam dan konsisten, dimana hal tersebut dilakukan untuk memudahkan model dalam mengenali pola serta mengurangi beban komputasi, sehingga proses pelatihan menjadi lebih cepat dan efisien.

a. Augmentasi

Adapun tahapan augmentasi citra yang dilakukan pada penelitian ini meliputi rescaling, hal ini dilakukan untuk membagi setiap nilai piksel gambar yang biasanya berada dalam rentang 0-255 akan diubah rentangnya menjadi 0-1. Selain rescale, adapun augmentasi yang dilakukan seperti zoom in dan zoom out 20% dari ukuran aslinya, flip secara horizontal, menggeser gambar secara horizontal dan juga vertikal hingga 20% dari lebar gambar secara acak. Hal tersebut dilakukan untuk memperbanyak variasi dataset yang membantu meningkatkan proses pelatihan model untuk mengenali citra dalam berbagai orientasi dan posisi. Selain itu, hal ini juga dapat mengurangi potensi terjadinya overfitting atau kesalahan dalam menghasilkan prediksi yang akurat pada data yang baru.



Gambar 23. Data setelah augmentasi



Gambar 24. Data lainnya setelah augmentasi

b. Labeling Data

Labeling data yang dilakukan pada penelitian ini dilakukan secara manual dengan membuat folder untuk setiap kelas yang akan dijadikan label untuk dataset yang akan diklasifikasikan. Proses selanjutnya, untuk membaca label dilakukan dengan memproses seluruh data gambar dan label dari generator `train_data`, yang menghasilkan data dalam bentuk batch. Pertama, disiapkan dua array kosong untuk menyimpan data gambar dan label. Kemudian, setiap batch diambil secara bertahap, lalu isinya ditambahkan ke array tersebut. Setelah data terkumpul, keduanya dikonversi ke array NumPy agar lebih efisien dan siap digunakan dalam pelatihan model machine learning.

2.5.3 Synthetic Minority Over-sampling Technique (SMOTE)

Dalam penelitian ini, teknik *SMOTE* digunakan untuk mengatasi adanya ketidakseimbangan kelas yang terjadi dalam dataset. Dimana, *SMOTE* disini bekerja dengan membuat sampel baru dari kelas yang lebih sedikit dibandingkan dengan kelas yang lainnya. Setelah dihasilkan data yang lebih seimbang, data gambar yang sebelumnya diubah ke ukuran dan bentuk aslinya dan selanjutnya akan digunakan dalam klasifikasi.

2.5.4 Split dataset

Pada penelitian akan dilakukan *splitting data* dengan dua skenario. Dimana skenario pertama, akan menggunakan empat kelas. Data yang telah dibaca akan dibagi (*splitting data*) menjadi dua bagian, yaitu data latih dan data uji dimana perbandingan 80% untuk data latih yang berjumlah 2186 citra dan 20% untuk data uji yang berjumlah 546 citra, dan dari data uji diambil keseluruhan citra yaitu 2732 citra untuk data validasi seperti yang dilihat pada tabel 1.

Tabel 1. Jumlah dataset 4 kelas

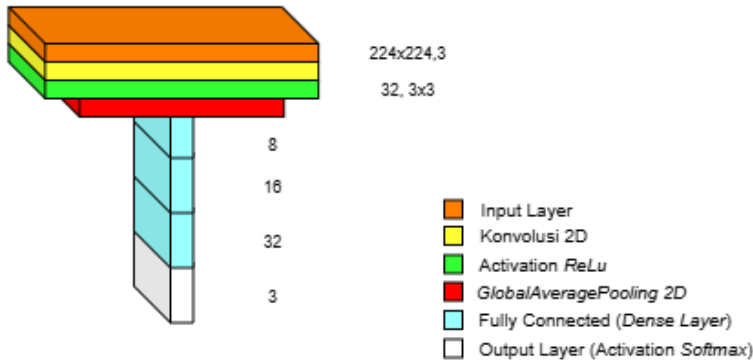
Kelas	Citra Tersedia	Citra Diolah
Normal	1.763	1.191
Gastritis	99.400	165
Kanker	220	185
Polip	1.191	1.191
Total	102.574	2.732

Selanjutnya, untuk skenario kedua yang menggunakan 3 kelas. Data yang telah dibaca akan dibagi (*splitting data*) menjadi dua bagian, yaitu data latih dan data uji dimana perbandingan 80% untuk data latih yang berjumlah 300 citra dan 20% untuk data uji yang berjumlah 73 citra, dan dari data uji diambil keseluruhan citra yaitu sebanyak 373 citra untuk data validasi seperti yang dilihat pada tabel 2.

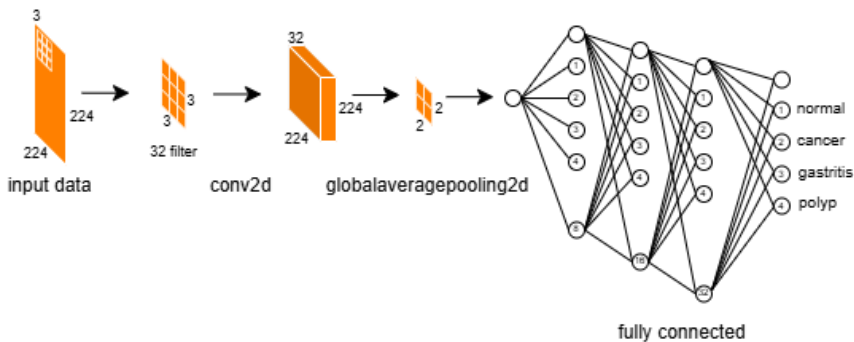
Tabel 2. Jumlah dataset 3 kelas

Kelas	Citra Tersedia	Citra Diolah
Normal	1.763	128
Gastritis	220	152
Kanker	220	93
Total	2.203	373

2.5.5 Training dan testing model



Gambar 25. Arsitektur CNN yang digunakan



Gambar 26. Visualisasi proses *training* arsitektur CNN

Pada lapisan pertama akan menerima inputan gambar yang telah di augmentasi menggunakan *rescaling*, *horizontal flip*, *zoom*, dan pergeseran gambar secara horizontal maupun vertikal dengan ukuran panjang dan lebar 224 dengan 3 channel *RGB*. Proses konvolusi dilakukan dengan menggunakan 32 filter dengan kernel yang berukuran 3×3 dan menggunakan fungsi aktivasi *ReLU*, seperti pada gambar 25.

Kemudian feature map akan masuk ke lapisan pooling 2D untuk mengurangi dimensi citra dengan ukuran kernel 2×2 . Jenis pooling yang dipakai adalah *GlobalAveragePooling2D* yang akan menghasilkan ukuran citra lebih kecil dari ukuran awalnya. *GAP2D* mengambil nilai rata-rata seluruh nilai dalam setiap peta fitur dalam lapisan berikutnya.

Setelah beberapa proses yang telah dilakukan sebelumnya, hasil feature map 2 dimensi akan dijadikan sebagai input pada lapisan *fully connected layer*. Pada lapisan fully connected terdapat 4 lapisan *dense* dengan jumlah neuron masing-masing yaitu 8, 16, dan 32 untuk hidden layer serta 4 untuk *output layer* yang

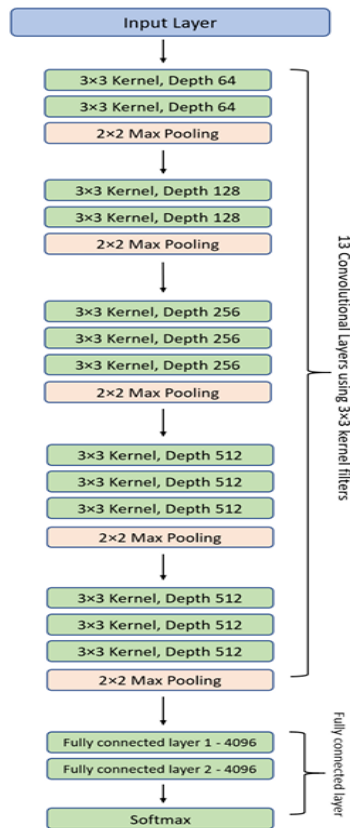
menerapkan fungsi aktivasi *softmax* yang menandai bahwa ada 4 kelas label yang akan di klasifikasi yang dapat dilihat pada gambar 26. Pada tahapan ini, sistem yang dibangun menggunakan *CNN from scratch*, *CNN* dengan *VGG16*, dan *CNN* dengan *ResNet50V2*. Adapun *Epoch* yang digunakan pada sistem ini yaitu sebanyak 10 kali. Parameter lainnya dapat dilihat pada Tabel 3.

Tabel 3. Pengaturan parameter pada proses training model

Parameter	CNN	VGG16	ResNet50V2
Epoch	10	10	10
Image Size	224x224	224x224	224x224
Batch Size	32	32	32
Filter Size	32	32	32
Optimizer	Adam	Adam	Adam

Tabel 3 menunjukkan parameter yang diterapkan, dimana tabel tersebut menunjukkan pengaturan parameter selama proses pelatihan model *CNN*, *VGG16*, dan *ResNet50V2*.

2.5.6 Model VGG16



Gambar 27. Arsitektur VGG16

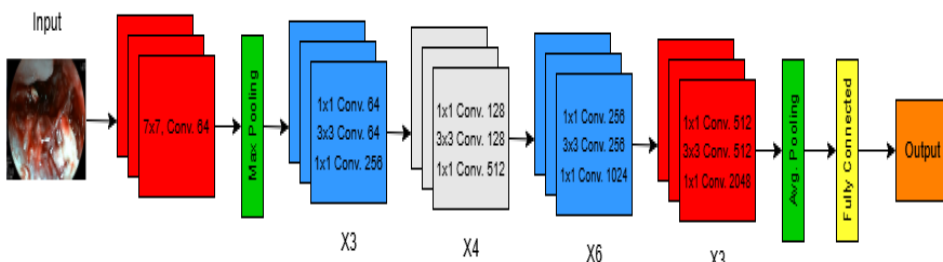
Berikut adalah penjelasan arsitektur yang ditampilkan pada gambar 27.

1. Input Layer:

Gambar inputan akan diproses menggunakan serangkaian lapisan konvolusi dengan kernel berukuran 3×3 , yang memiliki peran untuk mengekstraksi fitur dari gambar. Kedalaman (depth) dari setiap lapisan konvolusi bertambah secara bertahap seiring dengan bertambahnya kompleksitas fitur yang dipelajari. Pada tahap pertama, terdapat dua lapisan konvolusi dengan kernel 3×3 dan kedalaman 64, Proses ini berlanjut ke tahap berikutnya dengan dua lapisan konvolusi lainnya yang memiliki kedalaman 128, dan tahap berikutnya menggunakan tiga lapisan konvolusi dengan kedalaman 256. Pada tahap selanjutnya, tiga lapisan konvolusi dengan kedalaman 512 diterapkan. Langkah terakhir dari bagian konvolusi adalah tiga lapisan konvolusi lainnya dengan kedalaman 512.

2. Max Pooling:
Setelah lapisan konvolusi ditlakukan, proses max pooling diterapkan untuk mengurangi dimensi spasial dengan ukuran kernel 2x2.
3. Fully Connected Layer:
Setelah 13 lapisan konvolusi yang menggunakan kernel 3×3 , selanjutnya akan ke lapisan Fully Connected. Dua lapisan fully connected digunakan dengan jumlah neuron masing-masing 4096, yang bertugas memproses fitur akhir dari bagian konvolusi.
4. Output Layer:
Lapisan terakhir adalah lapisan Output menggunakan fungsi aktivasi *softmax*, yang memberikan probabilitas untuk setiap kelas output yang memungkinkan jaringan membuat prediksi akhir berdasarkan gambar yang diproses.

2.5.7 Model ResNet50V2



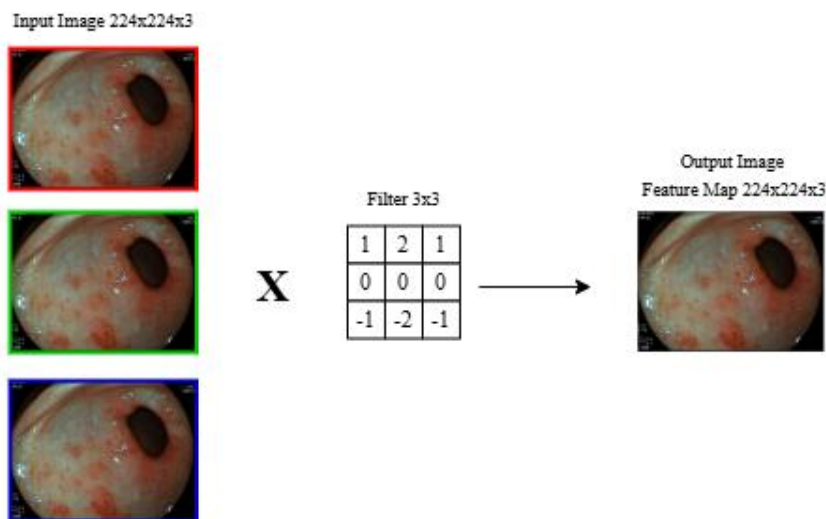
Gambar 28. Arsitektur ResNet50V2

Berikut adalah penjelasan arsitektur yang ditampilkan pada gambar 28.

1. Input Layer:
Gambar dimasukkan ke dalam jaringan, yang langsung diproses oleh lapisan konvolusi pertama dengan ukuran kernel 7×7 dan menggunakan 64 filter. Hasilnya adalah fitur awal dari gambar input.
2. Max-Pooling:
Setelah lapisan konvolusi awal, dilakukan proses max-pooling untuk mengurangi dimensi spasial dari data fitur. Hal ini bertujuan untuk mempercepat komputasi dan mengurangi ukuran data yang diproses.
3. Residual Blocks:
Arsitektur ini terdiri dari beberapa blok residual:
 - a. Blok x3: Terdiri dari 3 unit residual. Setiap unit menggunakan lapisan konvolusi dengan ukuran 1×1 Conv, 64 filter untuk mengurangi dimensi, diikuti oleh 3×3 Conv, 64 filter untuk ekstraksi fitur, dan ditutup dengan 1×1 Conv, 256 filter untuk mengembalikan dimensi.

- b. Blok x4: Terdiri dari 4 unit residual. Pada blok ini, jumlah filter bertambah
 - c. menjadi 128, 128, dan 512 di setiap unit residual.
 - d. Blok x6: Terdiri dari 6 unit residual. Pada tahap ini, jumlah filter meningkat lagi menjadi 256, 256, dan 1024 di setiap unit residual.
 - e. Blok x3: Terdiri dari 3 unit residual. Jumlah filter mencapai maksimum dengan 512, 512, dan 2048.
 - f. Blok residual ini menggunakan shortcut connections untuk membantu pelatihan jaringan yang sangat dalam dengan cara mengatasi masalah degradasi gradien.
4. Average Pooling:
Selanjutnya, dilakukan proses average pooling. Langkah ini meringkas informasi yang telah diekstraksi ke dimensi fitur yang lebih kecil.
 5. Fully Connected Layer:
Lapisan terakhir adalah lapisan fully connected yang memiliki 1000 output. Lapisan ini dirancang untuk menghasilkan prediksi klasifikasi pada dataset ImageNet, yang memiliki 1000 kelas.
 6. Output Layer:
Lapisan output memberikan hasil akhir berupa probabilitas kelas untuk setiap gambar yang dimasukkan ke dalam jaringan.

2.5.8 Convolutional layer



Gambar 29. Ilustrasi proses konvolusi pada data citra lambung

Untuk mendefinisikan citra lambung, digunakan *convolutional layer*. *Convolutional layer* bekerja dengan melakukan operasi konvolusi pada data input berupa data array dengan matriks *filter* atau biasa disebut dengan kernel. Proses ini melibatkan perkalian elemen-elemen data input dengan nilai -nilai pada kernel dan

menjumlahkan hasilnya untuk menghasilkan sebuah nilai Tunggal dalam output yang disebut matriks konvolusi. Kernel ini bergerak melintasi seluruh input data sesuai dengan langkah (*stride*) yang telah ditentukan, sehingga menghasilkan fitur-fitur penting dari data input. Fitur-fitur ini kemudian digunakan oleh jaringan sarraf tiruan untuk mengenali pola-pola yang akan di klasifikasikan. Input citra lambung pada penelitian ini memiliki ukuran 224x224x3, dimana memiliki 3 channel yaitu *red*, *green* dan *blue* (*RGB*). Setiap channel memiliki nilai matriks yang berbeda-beda, dan juga memiliki total 50.176 piksel yang akan di konvolusi dengan menggunakan *filter* berukuran 3x3 dengan perpindahan atau *stride* 1. Secara detail, proses konvolusi pada citra lambung sebagai berikut.

a. Proses konvolusi *Channel Red*

Proses ini merupakan langkah awal dari lima lapisan dalam arsitektur *CNN*, di mana feature map yang dihasilkan memiliki ukuran 224x224 dengan tiga channel, yaitu *red*, *green*, dan *blue*. Gambar input yang terdiri dari tiga channel tersebut akan melalui tahap konvolusi. Pada tahap ini, citra input berukuran 224x224 diproses menggunakan kernel berukuran 3x3 dengan *stride* sebesar 1. Proses ini menghasilkan output baru berupa hasil konvolusi.

Gambar 30. Matriks intensitas piksel citra lambung 224x224

Karena matriks 224x224 mempunyai proses konvolusi yang sangat panjang, maka kali ini akan disederhanakan dengan konvolusi citra lambung berukuran 9x9.

RED	1	2	3	4	5	6	7	8	9
1	79	88	82	86	86	84	85	85	87
2	74	81	81	79	79	85	84	84	84
3	74	81	82	80	80	84	83	83	83
4	74	82	82	81	80	83	82	82	82
5	74	81	82	82	82	83	83	84	84
6	73	80	81	82	82	85	85	86	86
7	71	78	80	81	82	85	87	87	87
8	70	77	79	80	82	84	86	87	86
9	69	75	78	80	81	83	85	86	85

Gambar 31. Nilai matriks intensitas piksel citra lambung 9x9

Proses konvolusi yang terjadi dengan *filter* 3x3 dengan *stride* 1.

RED	1	2	3	4	5	6	7	8	9		
1	79	88	82	86	86	84	85	85	87	X	Filter 3x3 $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
2	74	81	81	79	79	85	84	84	84		
3	74	81	82	80	80	84	83	83	83		
4	74	82	82	81	80	83	82	82	82		
5	74	81	82	82	82	83	83	84	84		
6	73	80	81	82	82	85	85	86	86		
7	71	78	80	81	82	85	87	87	87		
8	70	77	79	80	82	84	86	87	86		
9	69	75	78	80	81	83	85	86	85		
RED	1	2	3	4	5	6	7	8	9		
1	79	88	82	86	86	84	85	85	87	X	Filter 3x3 $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
2	74	81	81	79	79	85	84	84	84		
3	74	81	82	80	80	84	83	83	83		
4	74	82	82	81	80	83	82	82	82		
5	74	81	82	82	82	83	83	84	84		
6	73	80	81	82	82	85	85	86	86		
7	71	78	80	81	82	85	87	87	87		
8	70	77	79	80	82	84	86	87	86		
9	69	75	78	80	81	83	85	86	85		
RED	1	2	3	4	5	6	7	8	9		
1	79	88	82	86	86	84	85	85	87	X	Filter 3x3 $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
2	74	81	81	79	79	85	84	84	84		
3	74	81	82	80	80	84	83	83	83		
4	74	82	82	81	80	83	82	82	82		
5	74	81	82	82	82	83	83	84	84		
6	73	80	81	82	82	85	85	86	86		
7	71	78	80	81	82	85	87	87	87		
8	70	77	79	80	82	84	86	87	86		
9	69	75	78	80	81	83	85	86	85		
RED	1	2	3	4	5	6	7	8	9		
1	79	88	82	86	86	84	85	85	87	X	Filter 3x3 $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
2	74	81	81	79	79	85	84	84	84		
3	74	81	82	80	80	84	83	83	83		
4	74	82	82	81	80	83	82	82	82		
5	74	81	82	82	82	83	83	84	84		
6	73	80	81	82	82	85	85	86	86		
7	71	78	80	81	82	85	87	87	87		
8	70	77	79	80	82	84	86	87	86		
9	69	75	78	80	81	83	85	86	85		
RED	1	2	3	4	5	6	7	8	9		
1	79	88	82	86	86	84	85	85	87	=	Filter 3x3 $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
2	74	81	81	79	79	85	84	84	84		
3	74	81	82	80	80	84	83	83	83		
4	74	82	82	81	80	83	82	82	82		
5	74	81	82	82	82	83	83	84	84		
6	73	80	81	82	82	85	85	86	86		
7	71	78	80	81	82	85	87	87	87		
8	70	77	79	80	82	84	86	87	86		
9	69	75	78	80	81	83	85	86	85		
RED	1	2	3	4	5	6	7	8	9		
1	79	88	82	86	86	84	85	85	87	=	Filter 3x3 $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
2	74	81	81	79	79	85	84	84	84		
3	74	81	82	80	80	84	83	83	83		
4	74	82	82	81	80	83	82	82	82		
5	74	81	82	82	82	83	83	84	84		
6	73	80	81	82	82	85	85	86	86		
7	71	78	80	81	82	85	87	87	87		
8	70	77	79	80	82	84	86	87	86		
9	69	75	78	80	81	83	85	86	85		

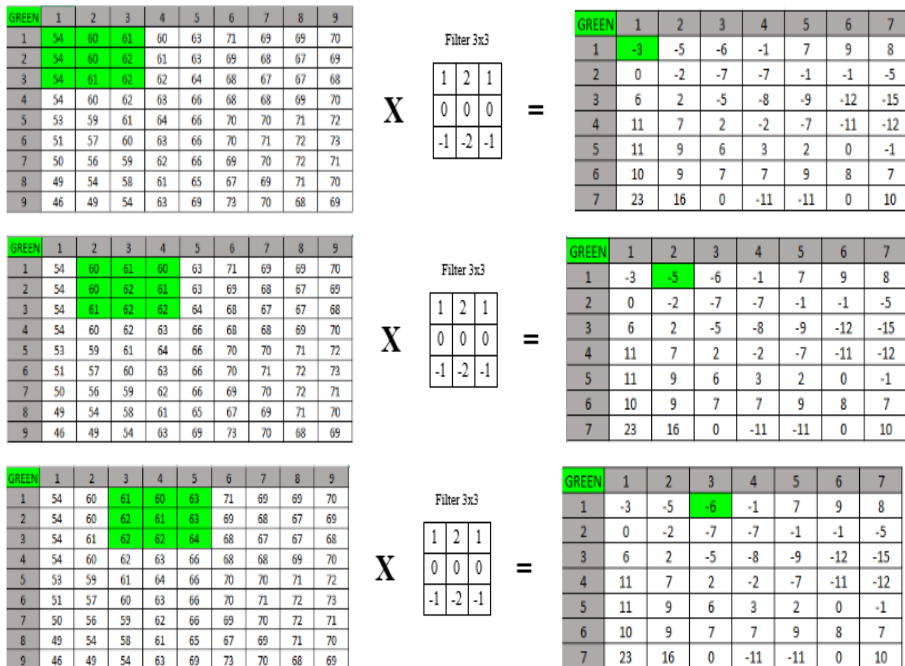
Gambar 32. Proses konvolusi *channel red*

Proses konvolusi yang terjadi menggunakan filter berukuran 3x3 dengan *stride* 1 secara matematis adalah sebagai berikut:

- $[1,1] = (79 \times 1) + (88 \times 2) + (82 \times 1) + (74 \times 0) + (81 \times 0) + (81 \times 0) + (74 \times (-1)) + (81 \times (-2)) + (82 \times (-1))$
- $[1,1] = 19$
- $[1,2] = (88 \times 1) + (82 \times 2) + (86 \times 1) + (81 \times 0) + (81 \times 0) + (79 \times 0) + (81 \times (-1)) + (82 \times (-2)) + (80 \times (-1))$
- $[1,2] = 13$

- $[1,3] = (82 \times 1) + (86 \times 2) + (86 \times 1) + (81 \times 0) + (79 \times 0) + (79 \times 0) + (82 \times (-1)) + (80 \times (-2)) + (80 \times (-1))$
 - $[1,3] = 18$
- Proses ini akan terus berlangsung hingga citra telah di konvolusi membentuk hasil konvolusi berupa feature map dengan ukuran 7x7.

b. Proses konvolusi *Channel Green*

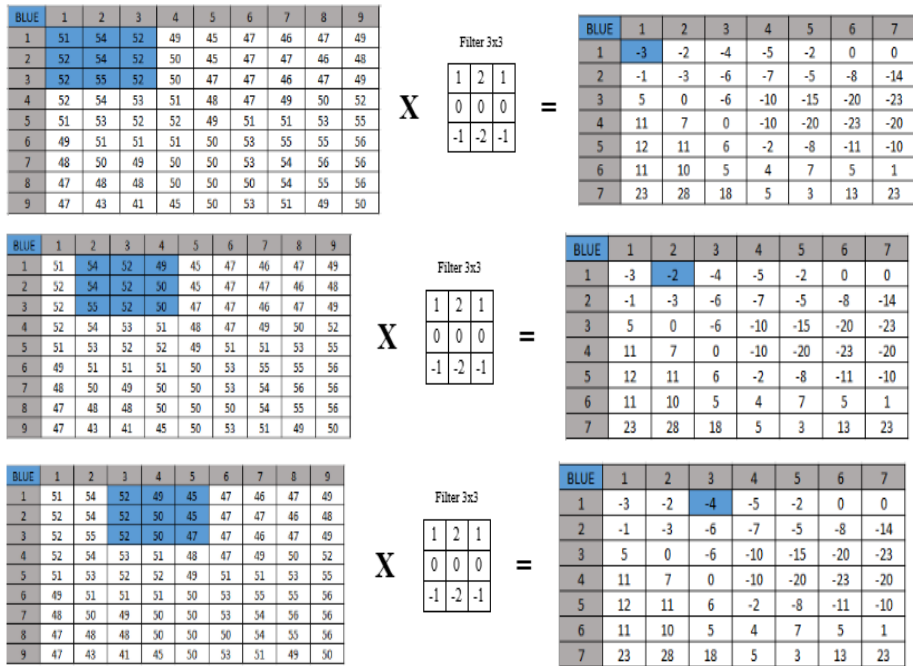


Gambar 33. Proses konvolusi *channel green*

- $[1,1] = (54 \times 1) + (60 \times 2) + (61 \times 1) + (54 \times 0) + (60 \times 0) + (62 \times 0) + (54 \times (-1)) + (61 \times (-2)) + (62 \times (-1))$
- $[1,1] = -3$
- $[1,2] = (60 \times 1) + (61 \times 2) + (60 \times 1) + (60 \times 0) + (62 \times 0) + (61 \times 0) + (61 \times (-1)) + (62 \times (-2)) + (62 \times (-1))$
- $[1,2] = -5$
- $[1,3] = (61 \times 1) + (60 \times 2) + (63 \times 1) + (62 \times 0) + (61 \times 0) + (63 \times 0) + (62 \times (-1)) + (62 \times (-2)) + (64 \times (-1))$
- $[1,3] = -6$

Proses ini akan terus berlangsung hingga citra telah di konvolusi membentuk hasil konvolusi berupa feature map dengan ukuran 7x7.

c. Proses konvolusi *Channel Blue*



Gambar 34. Proses konvolusi *channel blue*

- $[1,1] = (51 \times 1) + (54 \times 2) + (52 \times 1) + (52 \times 0) + (54 \times 0) + (52 \times 0) + (52 \times (-1)) + (55 \times (-2)) + (52 \times (-1))$
- $[1,1] = -3$
- $[1,2] = (54 \times 1) + (52 \times 2) + (49 \times 1) + (54 \times 0) + (52 \times 0) + (50 \times 0) + (55 \times (-1)) + (52 \times (-2)) + (50 \times (-1))$
- $[1,2] = -2$
- $[1,3] = (52 \times 1) + (49 \times 2) + (45 \times 1) + (52 \times 0) + (50 \times 0) + (45 \times 0) + (52 \times (-1)) + (50 \times (-2)) + (47 \times (-1))$
- $[1,3] = -4$

Proses ini akan terus berlangsung hingga citra telah di konvolusi membentuk hasil konvolusi berupa feature map dengan ukuran 7x7.

2.5.9 Activation function

Pada setiap *convolutional layer* diikuti oleh fungsi aktivasi. Fungsi aktivasi yang digunakan dalam penelitian ini adalah *ReLU (Rectified Linear Unit)* dan *Softmax*. Fungsi ini menggantikan semua nilai negatif dalam matriks konvolusi dengan nol yang bertujuan untuk membantu jaringan belajar representasi non-linear. *ReLU* digunakan sebagai fungsi aktivasi setelah setiap *convolutional* dan *dense*. *Softmax* digunakan pada layer output untuk mengubah output menjadi distribusi probabilitas untuk klasifikasi multi-kelas. Adapun proses setiap channel nya, sebagai berikut.

RED	1	2	3	4	5	6	7
1	19	13	18	18	8	6	10
2	-3	-5	-6	-2	5	8	8
3	0	-2	-6	-5	0	0	-3
4	6	3	-3	-7	-9	-12	-15
5	11	8	4	-1	-8	-13	-13
6	11	9	6	3	1	-2	-3
7	10	8	5	5	7	7	6

ReLu

RED	1	2	3	4	5	6	7
1	19	13	18	18	8	6	10
2	0	0	0	0	0	5	8
3	0	0	0	0	0	0	0
4	6	3	0	0	0	0	0
5	11	8	4	0	0	0	0
6	11	9	6	3	1	0	0
7	10	8	5	5	7	7	6

Gambar 35. Proses fungsi aktivasi menggunakan ReLu *channel red*

GREEN	1	2	3	4	5	6	7
1	-3	-5	-6	-1	7	9	8
2	0	-2	-7	-1	-1	-1	-5
3	6	2	-5	-8	-9	-12	-15
4	11	7	2	-2	-7	-11	-12
5	11	9	6	3	2	0	-1
6	10	9	7	7	9	8	7
7	23	16	0	-11	-11	0	10

ReLu

GREEN	1	2	3	4	5	6	7
1	0	0	0	0	0	7	9
2	0	0	0	0	0	0	0
3	6	2	0	0	0	0	0
4	11	7	2	0	0	0	0
5	11	9	6	3	2	0	0
6	10	9	7	7	9	8	7
7	23	16	0	0	0	0	10

Gambar 36. Proses fungsi aktivasi menggunakan ReLu *channel green*

BLUE	1	2	3	4	5	6	7
1	-3	-2	-4	-5	-2	0	0
2	-1	-3	-6	-7	-5	-8	-14
3	5	0	-6	-10	-15	-20	-23
4	11	7	0	-10	-20	-23	-20
5	12	11	6	-2	-8	-11	-10
6	11	10	5	4	7	5	1
7	23	28	18	5	3	13	23

ReLu

BLUE	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	5	0	0	0	0	0	0
4	11	7	0	0	0	0	0
5	12	11	6	0	0	0	0
6	11	10	5	4	7	5	1
7	23	28	18	5	3	13	23

Gambar 37. Proses fungsi aktivasi menggunakan ReLu *channel blue*

2.5.10 Pooling layer

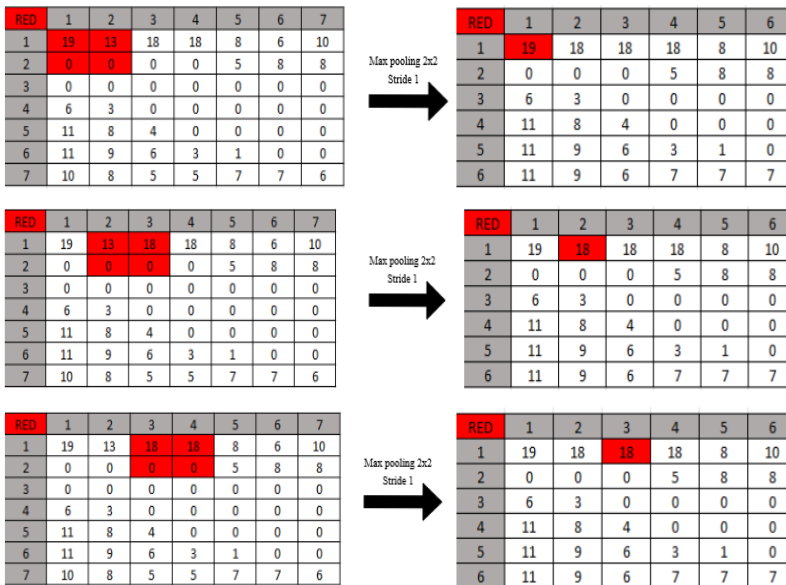
Pooling layer berguna untuk mengurangi dimensi spasial dari representasi gambar yang dihasilkan oleh *layer convolutional* sebelumnya. *Pooling layer* dalam penelitian ini menggunakan *Max pooling* dan *Global Average Pooling*.

a. *Max Pooling*

Max pooling memiliki fungsi untuk mengekstrak fitur penting dari sebuah gambar dengan memilih nilai yang paling tinggi di setiap area *pooling* atau *window*. Teknik ini sering digunakan untuk mengurangi dimensi gambar sekaligus mempertahankan fitur-fitur penting yang relevan. Adapun ilustrasi peroses max pooling dari setiap *channel*, sebagai berikut.

1) *Max Pooling Channel Red*

Max pooling dilakukan dengan memilih nilai tertinggi dari elemen matriks yang dilalui oleh filter. Dalam penelitian ini, filter yang digunakan memiliki ukuran 2x2 dengan *stride* sebesar 1

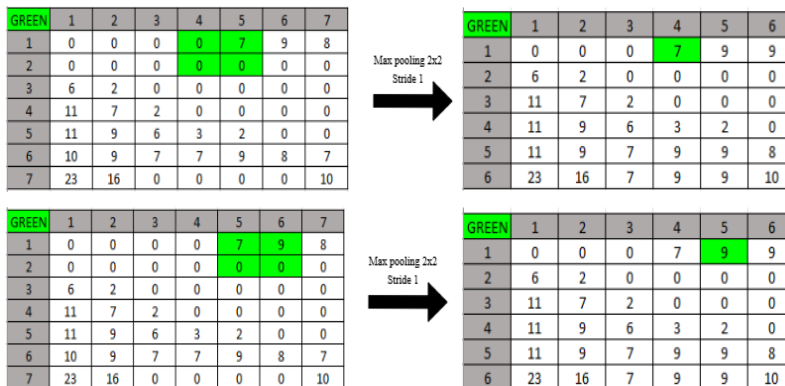


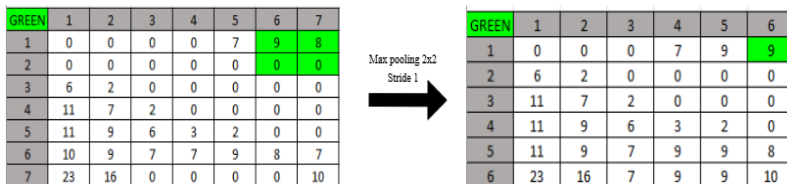
Gambar 38. Proses *max pooling* pada *channel red*

- Stride ke-1 nilai tertinggi = 19
- Stride ke-2 nilai tertinggi = 18
- Stride ke-3 nilai tertinggi = 18
- Stride ke-4 nilai tertinggi = 18
- Stride ke-5 nilai tertinggi = 8
- Stride ke-6 nilai tertinggi = 10

Dan seterusnya sampai didapatkan matriks baru hasil *max pooling* dengan dimensi 6x6.

2) *Max Pooling Channel Green*



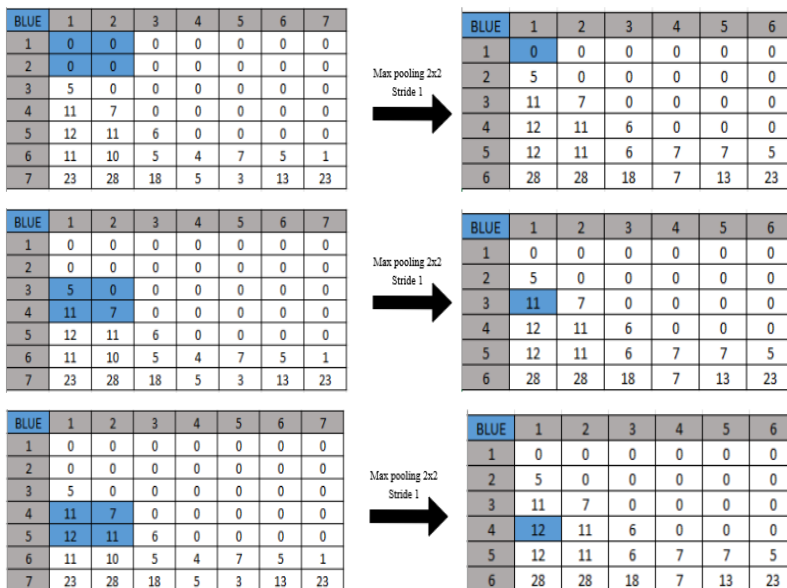


Gambar 39. Proses *max pooling* pada *channel green*

- Stride ke-1 nilai tertinggi = 0
- Stride ke-2 nilai tertinggi = 0
- Stride ke-3 nilai tertinggi = 0
- Stride ke-4 nilai tertinggi = 7
- Stride ke-5 nilai tertinggi = 9
- Stride ke-6 nilai tertinggi = 9

Dan seterusnya sampai didapatkan matriks baru hasil *max pooling* dengan dimensi 6x6.

3) *Max Pooling Channel Blue*



Gambar 40. Proses *max pooling* pada *channel blue*

- Stride ke-1 nilai tertinggi = 0
- Stride ke-13 nilai tertinggi = 11
- Stride ke-14 nilai tertinggi = 7
- Stride ke-19 nilai tertinggi = 12
- Stride ke-20 nilai tertinggi = 11

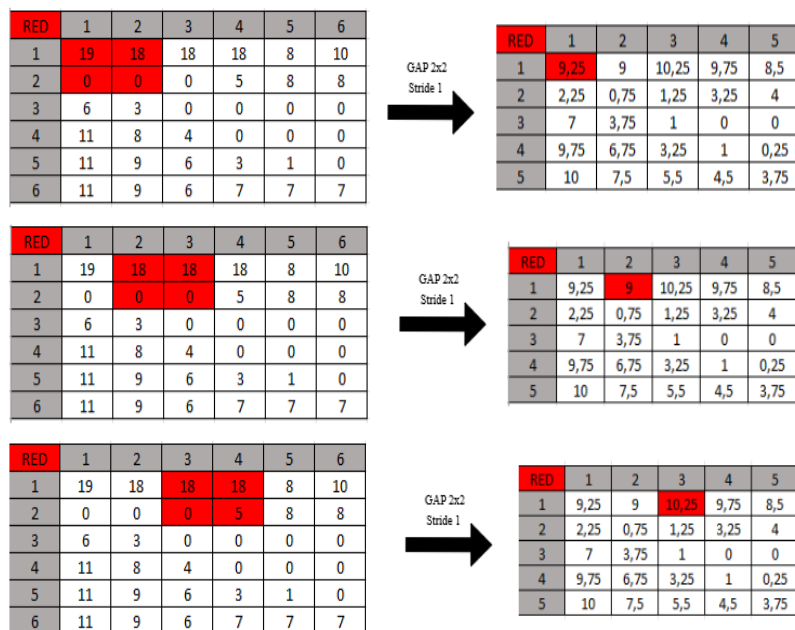
Dan seterusnya sampai didapatkan matriks baru hasil *max pooling* dengan dimensi 6x6.

b. *Global Average Pooling 2D*

Global Average Pooling 2D (GAP2D) merupakan jenis pooling yang menghitung nilai rata-rata dari setiap fitur di seluruh area gambar pada layer sebelumnya. Proses ini menghasilkan vektor fitur berukuran tetap yang selanjutnya dihubungkan ke layer berikutnya untuk klasifikasi. Dalam proses ini, seluruh nilai piksel dalam setiap fitur dijumlahkan dan dibagi dengan total jumlah pikselnya. Adapun proses *Global Average Pooling 2D* sebagai berikut.

1) *Global Average Pooling Channel Red*

Global Average Pooling 2D dilakukan dengan mengambil nilai rata-rata dari nilai matriks yang dilewati oleh filter. Dalam penelitian ini, filter yang digunakan memiliki ukuran 2x2 dengan *stride* sebesar 1.



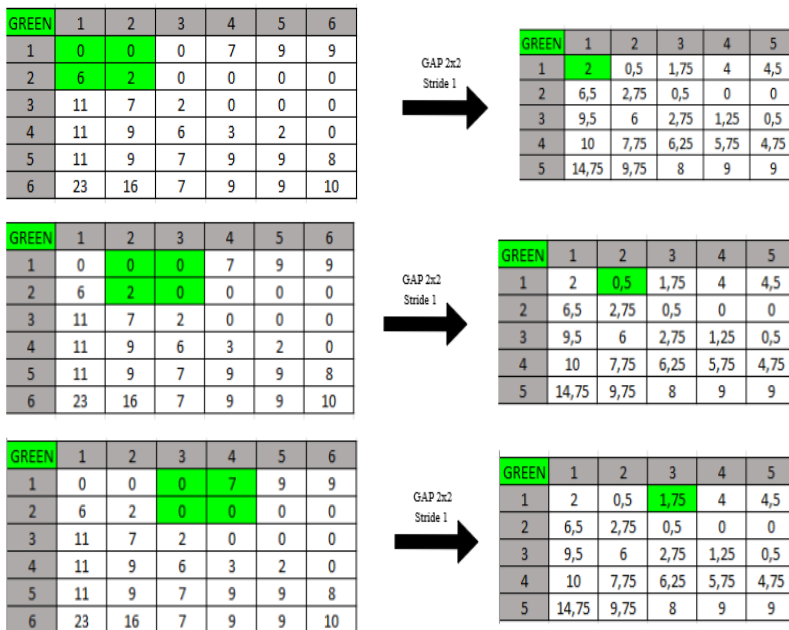
Gambar 41. Proses *global average pooling* pada *channel red*

Adapun proses matematisnya sebagai berikut.

$$\begin{aligned}
 - [1,1] &= \frac{(19+18+0+0)}{4} = 9,25 \\
 - [1,2] &= \frac{(18+18+0+0)}{4} = 9 \\
 - [1,3] &= \frac{(18+18+0+5)}{4} = 10,25
 \end{aligned}$$

Proses ini akan berlangsung hingga hasil *Global Average Pooling* membentuk matriks 5x5.

2) *Global Average Pooling Channel Green*

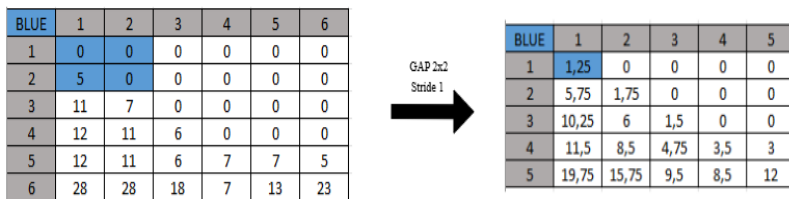


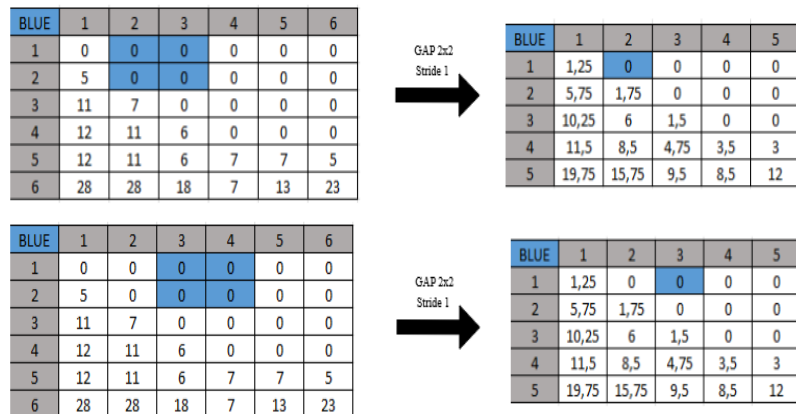
Gambar 42. Proses *global average pooling* pada *channel green*

- $[1,1] = \frac{(0+0+6+2)}{4} = 2$
- $[1,2] = \frac{(0+0+2+0)}{4} = 0,5$
- $[1,3] = \frac{(0+7+0+0)}{4} = 1,75$

Proses ini akan berlangsung hingga hasil *Global Average Pooling* membentuk matriks 5x5.

3) *Global Average Pooling Channel Blue*





Gambar 43. Proses *global average pooling* pada *channel blue*

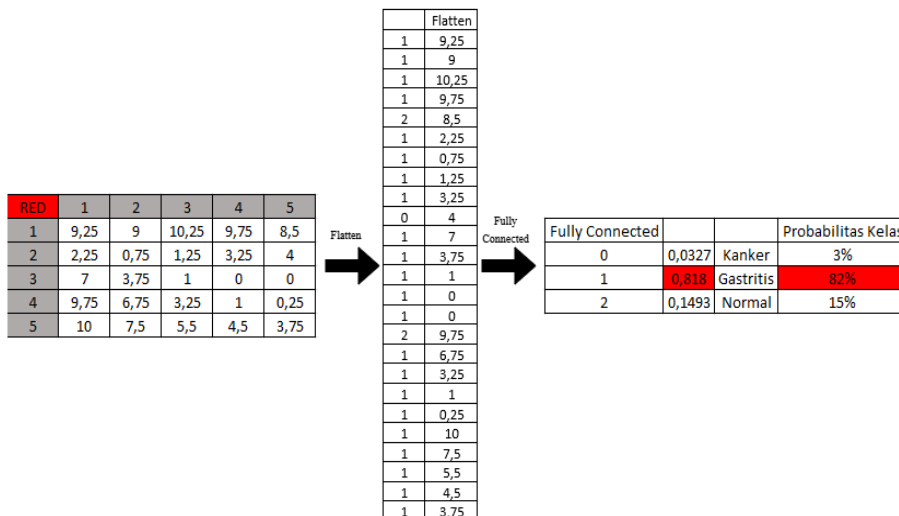
- $[1,1] = \frac{(0+0+5+0)}{4} = 1,25$
- $[1,2] = \frac{(0+0+0+0)}{4} = 0$
- $[1,3] = \frac{(0+0+0+0)}{4} = 0$

Proses ini akan berlangsung hingga hasil *Global Average Pooling* membentuk matriks 5x5.

2.5.11 Flatten & fully connected layer

Setelah beberapa convolutional dan pooling layers, fitur map yang dihasilkan kemudian diratakan menjadi vector 1 dimensi (flattening) agar dapat dihubungkan dengan lapisan fully connected. Proses flattening ditujukan agar jaringan dapat mengintegrasikan informasi spasial yang telah dipelajari ke dalam proses klasifikasi atau prediksi akhir. Kemudian selanjutnya masuk ke fully connected layer yang akan menghubungkan setiap neuron yang ada kemudian pada bagian lapisan output yang terdapat fungsi aktivasi softmax akan menghasilkan probabilitas masing-masing label yaitu kanker, gastritis dan normal. Adapun proses matematisnya yaitu dengan menghitung total nilai dari masing-masing kelas kemudian di bagi dengan total nilai dari seluruh kelas untuk menghitung persentase probabilitas tiap label. Berikut ilustrasi proses nya.

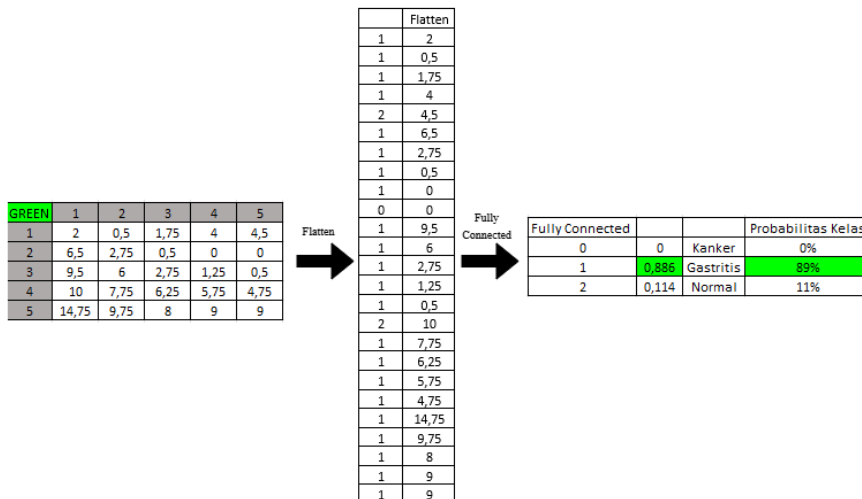
a. Channel Red



Gambar 44. Proses flatten dan fully connected layer channel red

- **Kelas 0** = $4 / 122,25 = 0,0327 = 3\%$
- **Kelas 1** = $(9,25 + 9 + 10,25 + 9,75 + 2,25 + 0,75 + 1,25 + 3,25 + 7 + 3,75 + 1 + 0 + 0 + 6,75 + 3,25 + 1 + 0,25 + 10 + 7,5 + 5,5 + 4,5 + 3,75) / 122,25 = 0,818 = 82\%$
- **Kelas 2** = $(8,5 + 9,75) / 122,25 = 0,1493 = 15\%$

b. Channel Green

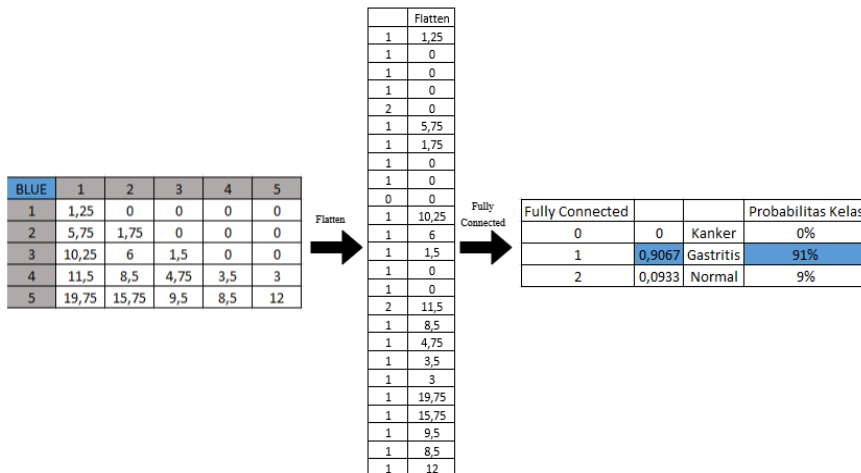


Gambar 45. Proses flatten dan fully connected layer channel green

- **Kelas 0** = $0 / 127,5 = 0 = 0\%$

- **Kelas 1** = $(2 + 0,5 + 1,75 + 4 + 6,5 + 2,75 + 0,5 + 0 + 9,5 + 6 + 2,75 + 1,25 + 0,5 + 7,75 + 6,25 + 5,75 + 4,75 + 14,75 + 9,75 + 8 + 9 + 9) / 127,5 = 0,886 = 89\%$
- **Kelas 2** = $(4,5 + 10) / 127,5 = 0,114 = 11\%$

c. *Channel Blue*



Gambar 46. Proses *flatten* dan *fully connected layer channel blue*

- **Kelas 0** = $0 / 123,25 = 0 = 0\%$
- **Kelas 1** = $(1,25 + 0 + 0 + 0 + 5,75 + 1,75 + 0 + 0 + 10,25 + 6 + 1,5 + 0 + 0 + 8,5 + 4,75 + 3,5 + 3 + 19,75 + 15,75 + 9,5 + 8,5 + 12) / 123,25 = 0,9067 = 91\%$
- **Kelas 2** = $(0 + 11,5) / 123,25 = 0,0933 = 9\%$

2.5.12 Hasil klasifikasi

Pada tahap klasifikasi ini, *fully connected layer* digunakan untuk merepresentasikan seluruh output dari proses *pooling layer*. Output ini berupa probabilitas dari setiap kelas yang mungkin ada pada dataset. Kelas dengan probabilitas tertinggi kemudian dipilih sebagai prediksi akhir dari model. Nilai probabilitas dari setiap label penyakit lambung pada masing-masing channel *RGB* akan dihitung rata-ratanya untuk menentukan kelas hasil klasifikasi. Sebagai contoh, pada citra input, hasil klasifikasi menunjukkan bahwa label atau kelas gastritis memiliki persentase sebesar 87%.

Fully Connected			Probabilitas Kelas
0	0,0327	Kanker	3%
1	0,818	Gastritis	82%
2	0,1493	Normal	15%

Fully Connected			Probabilitas Kelas
0	0	Kanker	0%
1	0,8863	Gastritis	89%
2	0,1137	Normal	11%

0		
1	87%	Gastritis
2		

Fully Connected			Probabilitas Kelas
0	0	Kanker	0%
1	0,9067	Gastritis	91%
2	0,0933	Normal	9%

Gambar 47. Hasil klasifikasi citra lambung

2.6 Export Model CNN

Pada tahapan ini, model *CNN* yang telah dikembangkan akan disimpan dan di export menggunakan library TensorFlow yang akan mengkonversi model *CNN* tersebut ke dalam format ekstensi *h5*.

2.7 Skenario Percobaan

Untuk mendapatkan model dengan nilai akurasi terbaik, akan dilakukan beberapa percobaan dengan beberapa skenario yang melibatkan 2 transfer learning yang berbeda serta menghapus salah satu kelas dari label dataset yaitu polip. Adapun skenario yang akan dilakukan yaitu sebagai berikut:

- Skenario pertama: menggunakan *CNN*, 4 dense layer (8,16,32,4) dengan 10 epoch pada 4 kelas label klasifikasi dataset.
- Skenario kedua: menggunakan *CNN*, 4 dense layer (8,16,32,4) dengan 10 epoch pada 3 kelas label klasifikasi dataset.
- Skenario ketiga: menggunakan *GAP 2D*, 4 dense layer (8,16,32,4) dengan VGG16 dan 10 epoch pada 4 kelas label klasifikasi dataset.
- Skenario keempat: menggunakan *GAP 2D*, 4 dense layer (8,16,32,4) dengan VGG16 dan 10 epoch pada 3 kelas label klasifikasi dataset.
- Skenario kelima: menggunakan *GAP 2D*, 4 dense layer (8,16,32,4) dengan ResNet50V2 dan 10 epoch pada 4 kelas label klasifikasi dataset.
- Skenario keenam: menggunakan *GAP 2D*, 4 dense layer (8,16,32,4) dengan ResNet50V2 dan 10 epoch pada 3 kelas label klasifikasi dataset.

Berdasarkan skenario yang telah dicantumkan diatas, perbandingan akan dilakukan sesuai dengan jumlah tiap kelas, dan model Transfer Learning yang digunakan.