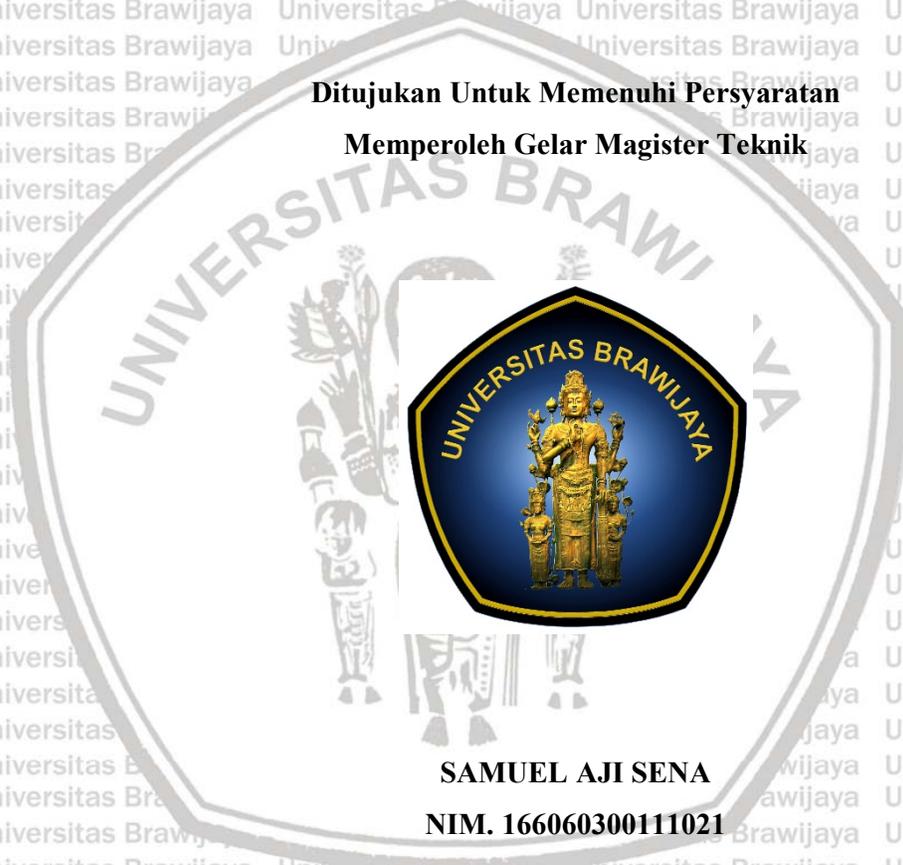


**DETEKSI KANKER PAYUDARA JENIS *INVASIVE DUCTAL*
CARCINOMA DENGAN MENGGUNAKAN *RESIDUAL*
*CONVOLUTIONAL NEURAL NETWORK***

TESIS
PROGRAM MAGISTER TEKNIK ELEKTRO
MINAT SISTEM KOMUNIKASI DAN INFORMATIKA

Ditujukan Untuk Memenuhi Persyaratan
Memperoleh Gelar Magister Teknik



SAMUEL AJI SENA
NIM. 166060300111021

UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2019



ABSTRACT

Samuel Aji Sena, Department of Electrical Engineering, Faculty of Engineering, University of Brawijaya, Juli 2019, *Deteksi Kanker Payudara Jenis Invasive Ductal Carcinoma dengan Menggunakan Residual Convolutional Neural Network*, Supervisor: Panca Mudjirahardjo and Sholeh Hadi Pramono.

This research presents a breast cancer detection system using deep learning method.

Breast cancer detection in a large slide of biopsy image are hard task because it needs manual observation by pathologist to find the malignant region.

Deep learning model used in this research are made up of multiple layer of residual convolutional neural network and instead of using other type of classifier, multilayer neural network were used as the classifier and stacked together and trained using end-to-end training approach.

The system are trained using invasive ductal carcinoma dataset from Hospital of the University of Pennsylvania and The Cancer Institute of New Jersey. From this dataset. 80% and 20% were randomly sampled and used as training and testing data respectively. Training a neural network on an imbalanced dataset are quite challenging. Weighted loss function were used as the objective function to tackle this problem. We achieve 78.26% dan 78.03% for Recall and F1-Score metric respectively which are an improvement compared to previous approach.

Keyword: IDC, breast cancer, deep learning, CNN, neural network, cross entropy

BAB I PENDAHULUAN

1.1. Latar Belakang

Kanker payudara merupakan salah satu penyakit yang paling mematikan khususnya bagi kaum wanita. Berdasarkan data Kementerian Kesehatan Republik Indonesia, pada tahun 2013, 1,4 dari 100 wanita Indonesia terdeteksi mengidap kanker payudara yang artinya terdapat sekitar 347.000 kasus dan data dari Badan Penyelenggara Jaminan Sosial (BPJS) Indonesia pada rentang waktu antara tahun 2014 hingga 2015, terdapat peningkatan jumlah penanganan dan pembiayaan pasien kanker payudara. Sedangkan prediksi dari Kementerian Kesehatan, pada tahun 2017 akan ada 9 juta orang meninggal dunia karena penyakit kanker payudara. (Kemenkes RI, 2017). Untuk menanggulangi permasalahan tersebut, deteksi dini penyakit kanker payudara sangat diperlukan. Salah satu jenis kanker payudara yang sering terjadi adalah *invasive ductal carcinoma* (IDC). Setidaknya 80% diagnosa kanker payudara adalah *invasive ductal carcinoma*. (BreastCancer.Org, 2018)

Pemeriksaan histopatologi dilakukan untuk mendeteksi penyakit kanker payudara yang biasanya dilakukan dengan cara melakukan analisis terhadap jaringan tubuh secara visual. Pemeriksaan biopsi cukup memakan waktu dan tenaga karena patologis harus dapat memisahkan area sel yang ganas pada sebuah slide yang terdiri dari mayoritas sel yang normal. Pemeriksaan ini sangat penting untuk menentukan tahapan dari kanker agar segera dilakukan penanganan yang sesuai.

Penelitian terkait tentang deteksi kanker payudara berdasarkan citra biopsi sebelumnya telah dilakukan oleh Angel Cruz-Roa, dkk pada tahun 2014. Pada penelitian tersebut, peneliti melakukan klasifikasi jenis kanker payudara berdasarkan citra biopsi dengan menggunakan *Convolutional Neural Network* dengan menggunakan dataset IDC dari Hospital of University of Pennsylvania and The Cancer Institute of New Jersey. Penelitian tersebut menggunakan hanya 2 buah convolution layer dan 2 buah pooling layer tanpa memperhatikan distribusi data yang seimbang sehingga performa dari sistem yang dihasilkan belum optimal sehingga diperlukan pengembangan dalam aspek metode penanganan distribusi data yang tidak seimbang dan arsitektur *neural network* yang lebih baik sehingga menghasilkan performa yang lebih baik.

Perkembangan teknologi *deep learning* bertumbuh sangat pesat pada zaman modern seperti saat ini. *Deep learning* dapat digunakan pada bidang *Natural Language Processing* (NLP), *Speech Recognition*, *Robotic*, *Computer Vision*, dll. Hampir semua permasalahan di segala bidang dapat diselesaikan dengan menggunakan arsitektur *neural network* yang sesuai. *Convolutional Neural Network* (CNN) contohnya. Sejak tahun 2012, CNN telah banyak digunakan untuk menyelesaikan permasalahan pada bidang *computer vision* seperti *object recognition* dan *object detection* yang dapat diterapkan pada segala bidang mulai dari bisnis, teknologi informasi hingga bidang medis. Pada bidang medis, *Convolutional Neural Network* biasanya digunakan untuk membantu diagnosa terhadap suatu penyakit yang memerlukan pemeriksaan visual seperti deteksi kanker kulit berdasarkan citra kulit maupun pemeriksaan biopsi untuk mendeteksi *invasive ductal carcinoma* yang akan dilakukan pada penelitian ini.

Namun CNN konvensional tidak dapat menghasilkan performa yang optimal jika digunakan pada dataset yang mempunyai resolusi yang kecil, karena pada setiap proses konvolusi akan terjadi pengurangan resolusi *feature map* sehingga jumlah *layer* CNN yang bisa digunakan relatif kecil. Sehingga pada penelitian ini akan digunakan jenis lain dari CNN yaitu Residual CNN yang memungkinkan penggunaan *layer* CNN yang lebih banyak sehingga performa sistem akan menjadi lebih baik.

1.2. Rumusan Masalah

Pada penelitian Deteksi Kanker Payudara Jenis *Invasive Ductal Carcinoma* dengan Menggunakan *Residual Convolutional Neural Network* ini dapat dirumuskan suatu permasalahan sebagai berikut :

1. Bagaimana penerapan *Residual Convolutional Neural Network* untuk mendeteksi kanker payudara jenis *Invasive Ductal Carcinoma* pada citra biopsi?
2. Bagaimana penerapan *weighted loss* untuk membangun sistem deteksi kanker payudara jenis *Invasive Ductal Carcinoma* dengan performa yang optimal?
3. Bagaimana analisa performa sistem sistem yang dirancang berdasarkan tingkat akurasi pada *dataset* dengan distribusi data yang tidak seimbang?

1.3. Batasan Masalah

Pada penelitian Deteksi Kanker Payudara Jenis *Invasive Ductal Carcinoma* dengan Menggunakan *Residual Convolutional Neural Network* ini akan diberikan batasan-batasan berupa :

1. *Dataset* yang digunakan adalah dataset IDC dari Hospital of University of Pennsylvania and The Cancer Institute of New Jersey.
2. Klasifikasi dilakukan terhadap setiap *patch* yang diambil dari *Whole Slide Image* (WSI).
3. *Patch* yang digunakan berukuran 50 x 50 *pixel*

1.4. Tujuan Penelitian

Penulis berharap pada proses pelatihan maupun proses pengujian pada Sistem Deteksi Kanker Payudara Jenis *Invasive Ductal Carcinoma* dengan Menggunakan *Residual Convolutional Neural Network* ini dapat dijalankan dengan baik dan model yang telah selesai dilatih dapat menghasilkan hasil yang baik dalam melakukan deteksi terhadap *invasive ductal carcinoma*.

1.5. Manfaat Penelitian

Penelitian ini bermanfaat bagi berbagai pihak yang terlibat pada pemeriksaan kanker payudara jenis *invasive ductal carcinoma*. Selain itu, penelitian ini juga bermanfaat bagi pihak lain yang ingin mengembangkan sistem deteksi kanker payudara untuk mendapatkan hasil yang lebih baik lagi.

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Relevan

Penjelasan secara lengkap mengenai beberapa penelitian yang terkait ditunjukkan sebagai berikut :

Penelitian oleh Scott Doyle, dkk pada tahun 2008, membuat sistem yang dapat mengklasifikasi citra biopsi untuk mendeteksi adanya sel kanker payudara. Citra biopsi akan diekstrak fiturnya. Fitur yang digunakan oleh peneliti terbagi menjadi dua jenis yaitu *texture feature* dan *graph feature*. *Texture feature* yang digunakan adalah *Grey Level, Haralick Feature* dan *Gabor Filter Feature* sedangkan *Graph Feature* yang digunakan adalah Voronoi Diagram, Delaunay Triangulation, Minimum Spanning Tree dan *Nuclear Feature*. Seluruh *feature* ini akan sebagai masukan dari *Support Vector Machine (SVM)* untuk menentukan adanya indikasi kanker atau tidak.

Penelitian oleh S. Issac Niwas, dkk pada tahun 2011, mencoba untuk melakukan klasifikasi terhadap jenis kanker payudara. Peneliti menggunakan algoritma k-means clustering untuk melakukan segmentasi terhadap citra biopsi tersebut, selanjutnya *color space* citra hasil segmentasi akan dirubah menjadi format HSI. Proses selanjutnya adalah melakukan transformasi gabor dan mengekstrak fiturnya yang terdiri dari *mean, standard deviation, skewness, smoothness, entropy, contract, energy*, dll. Terdapat 15 buah fitur yang selanjutnya akan digunakan pada proses klasifikasi menggunakan *Support Vector Machine (SVM)*.

Kedua penelitian diatas masih menggunakan *feature* yang membutuhkan *pre-processing* yang rumit untuk mendapatkan kualitas *feature* yang baik untuk dapat dikenali oleh *classifier* dengan baik.

Penelitian oleh Angel Cruz-Roa, dkk pada tahun 2014 sudah mulai menggunakan teknik *deep learning* untuk melakukan deteksi terhadap adanya *invasive ductal carcinoma* pada citra biopsi. *Convolutional Neural Network* digunakan sebagai *feature extractor* dan *Multilayer Perceptron* digunakan sebagai *classifier*. Dataset yang digunakan pada penelitian tersebut adalah dataset IDC dari Hospital of University of Pennsylvania and The Cancer Institute of New Jersey yang juga akan digunakan pada penelitian ini.

Berikut adalah ringkasan penelitian yang relevan terhadap penelitian deteksi kanker payudara jenis *invasive ductal carcinoma* dengan menggunakan *residual convolutional neural network*:

Tabel 2.1 Ringkasan Penelitian

No	Penulis	Judul	Metode	Hasil
1	Scott Doyle (2008)	<i>Automated grading of breast cancer histopathology using spectral clustering with textural and architectural image features</i>	<i>Texture dan Graph Feature</i> dengan SVM sebagai <i>classifier</i> .	Sistem dapat berjalan dengan baik dan dapat memberikan grade terhadap status kanker berdasarkan citra biopsy dengan menggunakan texture feature.
2	S. Issac Niwas (2011)	<i>Log-gabor wavelets based breast carcinoma classification using least square support vector machine.</i>	Log-Gabor Wavelets Feature dengan <i>Least Square SVM</i> sebagai <i>classifier</i>	Sistem dapat berjalan dengan baik dan dapat melakukan klasifikasi terhadap sel kanker berdasarkan citra biopsy dengan menggunakan <i>log-gabor feature</i> .
3	Angel Cruz-Roa (2014)	<i>Automatic detection of invasive ductal carcinoma in whole slide images with Convolutional Neural Networks.</i>	<i>Deep Learning</i> dengan menggunakan CNN dan MLP.	<i>Precision: 0.783, Sensitivity: 0.691, Specificity: 0.924, F-Measure = 0.734</i> pada dataset IDC dari Hospital of University of Pennsylvania and The Cancer Institute of New Jersey
4	Andre Esteva (2017)	<i>Dermatologist-level classification of skin cancer with deep neural networks</i>	<i>Deep Learning</i> dengan menggunakan <i>Deep Residual CNN</i> dan MLP	Sistem dapat mengklasifikasi jenis kanker kulit berdasarkan citra kulit dengan sangat baik dan dapat menyamai performa <i>Dermatologist</i> .

2.2. Kanker Payudara

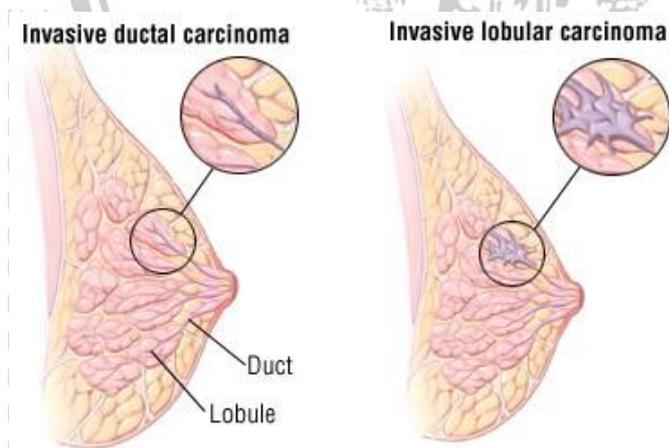
Kanker payudara adalah sel kanker yang tumbuh pada area payudara. Sel kanker ini dapat menyebar pada bagian atau organ lain disekitar payudara jika tidak segera ditangani (Edna Kapenhas, 2010). Menurut data dari Organisasi Breastcancer, kanker payudara dibagi menjadi 9 tipe, yaitu:

1. *Ductal Carcinoma In Situ (DCIS)*
2. *Invasive Ductal Carcinoma (IDC)*
3. *Invasive Lobular Carcinoma (ILC)*
4. *Inflammatory Breast Cancer*
5. *Lobular Carcinoma In Situ (LCIS)*
6. *Molecular Subtypes of Breast Cancer*
7. *Paget's Disease of the Nipple*
8. *Phyllodes Tumors of the Breast*
9. *Metastatic Breast Cancer*

Penangan kanker payudara yang biasa dilakukan adalah dengan cara operasi untuk mengambil sel kanker agar tidak menyebar, *chemotherapy* dan obat-obatan yang dirancang khusus untuk menghambat perkembangan sel kanker. (American Cancer Society, 2017)

2.2.1. *Invasive Ductal Carcinoma (IDC)*

Invasive Ductal Carcinoma atau yang biasa juga disebut *Infiltrating Ductal Carcinoma* merupakan tipe kanker payudara yang sering terjadi, dimana 80% kasus kanker payudara merupakan IDC. Kanker ini dimulai dari kulit atau jaringan disekitar payudara dan menyebar melalui kelenjar susu dan menyebar keseluruh jaringan payudara.



Gambar 2.1 *Invasive Ductal Carcinoma*

Jika tidak segera ditangani IDC dapat menyebar ke kelenjar getah bening dan mungkin menuju jaringan tubuh yang lain. IDC sering terjadi pada wanita yang berusia 55 tahun atau lebih, meskipun tidak menutup kemungkinan bisa terjadi pada wanita dibawah 55 tahun maupun pria. (BreastCancer.Org, 2018)

2.3. Citra Digital

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek.

Citra terbagi 2 yaitu citra yang bersifat analog dan ada citra yang bersifat digital. Citra analog adalah citra yang bersifat continue seperti gambar pada monitor televisi, foto sinar X, dan lain-lain. Sedangkan pada citra digital adalah citra yang dapat diolah oleh komputer (Sutoyo, 2009).

Citra dapat didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada citra tersebut (Putra, 2010:19).

2.3.1. Pengolahan Citra

Pengolahan citra adalah segala kegiatan yang dilakukan untuk melakukan perubahan pada citra tersebut seperti memanipulasi dan memodifikasi citra dengan berbagai cara (Sutoyo, 2009). Pengolahan citra berdasarkan tujuan operasi pengolahan citra dapat dikategorikan sebagai berikut:

a. Peningkatan Kualitas Citra (*Image Enhancement*)

Operasi yang bertujuan untuk melakukan peningkatan terhadap kualitas citra dengan meningkatkan fitur tertentu pada citra.

b. Pemulihan Citra (*Image Restoration*)

Operasi pemulihan citra bertujuan untuk mengembalikan kondisi citra pada kondisi yang diketahui sebelumnya, yang diakibatkan oleh faktor pengganggu yang menyebabkan penurunan kualitas citra.

2.3.2. Resolusi Citra

Resolusi citra merupakan istilah untuk menerangkan ukuran citra digital yang digambarkan dengan banyaknya piksel yang digunakan untuk memvisualisasikan citra digital. Resolusi dalam citra juga menjadi acuan untuk menentukan tingkat detailnya suatu citra. Semakin tinggi resolusinya semakin tinggi pula tingkat detail dari citra tersebut begitu juga sebaliknya. Resolusi dalam citra digital memiliki dua jenis resolusi, yaitu:

1. Resolusi Sparsial

Resolusi spasial ini merupakan ukuran halus atau kasarnya visualisasi citra dengan melakukan pembagian piksel pada baris dan kolom. Resolusi ini dipakai untuk menentukan banyaknya piksel per satuan panjang, umumnya satuan resolusi yang digunakan adalah dpi (*dot per inch*). Resolusi ini berpengaruh pada detail pada suatu citra digital.

2. Resolusi Kecemerlangan

Resolusi intensitas kecermerlangan (*brightness*) juga disebut dengan kedalaman bit warna (*Bit Depth*), kedalaman bit warna (*Bit Depth*) adalah ukuran nilai banyaknya pembagian tingkat gradasi warna yang mempengaruhi halus kasarnya tingkat gradasi saat dilakukan kuantisasi. *Bit Depth* menentukan informasi warna untuk divisualisasikan dalam setiap piksel. Semakin besar nilai dari *Bit Depth*, semakin banyak tingkat gradasi yang dapat dihasilkan dan mempengaruhi tingkat kualitas gambar yang dihasilkan (Sutoyo, 2009).

2.3.3. Color Spaces

Fungsi dari color spaces adalah memberikan standard dan dapat diterima secara umum. Color space adalah sistem koordinat ruang dimana setiap satu titik pada koordinat tersebut melambangkan satu buah warna. Sebagian besar color space yang ada sekarang ini berorientasi pada medium dimana citra tersebut akan digunakan. Pada medium layar monitor atau televisi, color space yang digunakan adalah RGB (Red Green Blue) sedangkan untuk keperluan cetak, color space yang digunakan adalah CMYK. (Gonzalez, 2008).

Selain itu masih banyak standard color space yang lain seperti HSL, HSV, YUV, CIELAB yang menyimpan nilai lightness atau citra monochrome dari sebuah citra warna.

2.2.3.1 Red Green Blue (RGB)

RGB adalah color space yang sangat banyak digunakan. Setiap warna dasar disimpan dalam memori dengan ukuran 1 byte, yang berarti dalam setiap 1 warna dasar mempunyai tingkat gradasi sebanyak 255 warna. Berbeda dengan citra grayscale, pada penyimpanan didalam memori setiap 1 pixel diwakili oleh 3 byte. Masing - masing byte merepresentasikan warna merah (Red), hijau (Green), biru (Blue).

2.3.4. Citra Grayscale

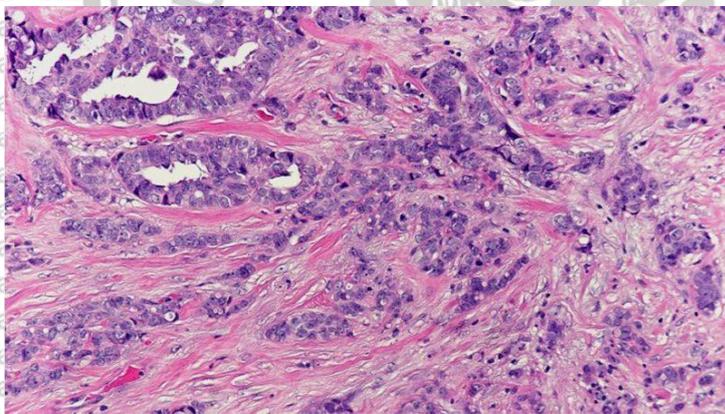
Skala keabuan adalah model penyederhanaan dari citra warna menjadi 1 bagian warna grayscale sehingga dihasilkan citra seperti pada Gambar 2.3. Untuk mengubah citra warna RGB menjadi citra grayscale, dilakukan dengan menghitung nilai rata rata dari setiap bagian warna dasar yaitu bagian warna merah, warna hijau dan warna biru. Sedangkan untuk mendapatkan nilai grayscale dari citra LAB, komponen L pada citra LAB yang memiliki nilai 0 hingga 100 akan dinormalisasi menjadi 0 hingga 255.



Gambar 2.2 GRB to Grayscale

2.3.5. Citra Histopatologi

Citra histopatologi atau biasa disebut citra biopsi didapatkan dengan cara mengambil sebagian jaringan tubuh dan diteliti dibawah mikroskop. Selanjutnya patologist akan menentukan ada atau tidaknya jaringan kanker pada citra tersebut. Analisis citra histopatologi biasanya memakan waktu yang relatif lama karena area yang diteliti sangat luas karena pembesaran pada mikroskop dan diperlukan ketelitian dari patologist untuk menemukan area yang terdapat sel kanker.



Gambar 2.3 Citra Histopathologi pada IDC

2.4. Computer Vision

Computer vision adalah sebuah disiplin ilmu yang mempelajari bagaimana merekonstruksi, menginterpretasikan, dan memahami sebuah tampilan 3 dimensi dari tampilan 2 dimensinya dalam hal sifat dari struktur tampilan tersebut. *Computer vision* berkaitan dengan pemodelan dan meniru penglihatan manusia dengan menggunakan perangkat lunak dan perangkat keras pada komputer. *Computer vision* menggabungkan ilmu pengetahuan dalam bidang ilmu komputer, teknik elektro, matematika, fisiologi, biologi, dan

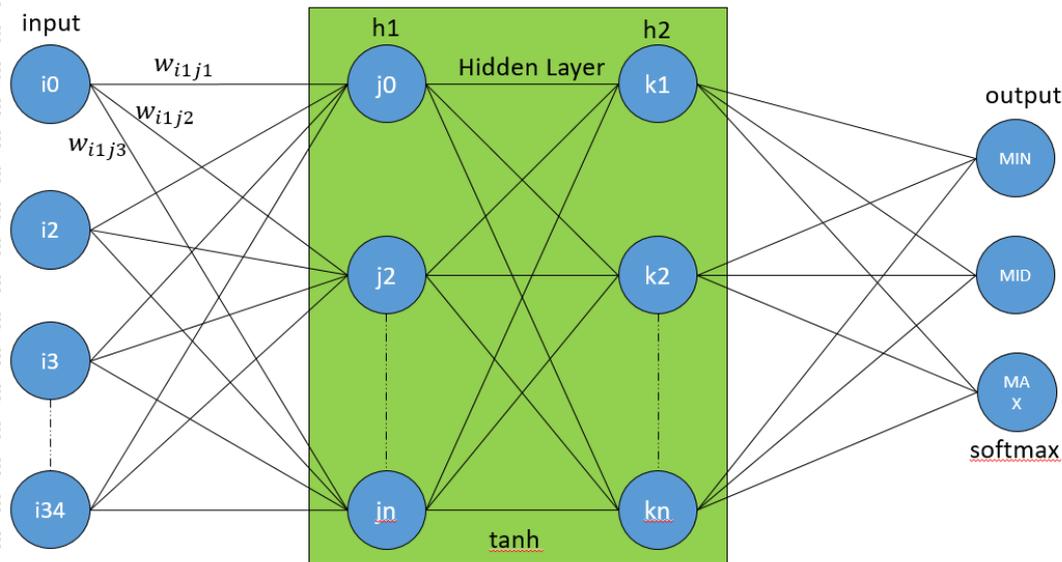
ilmu kognitif. Diperlukan ilmu dari semua bidang tersebut untuk memahami dan mensimulasikan pengoperasian sistem penglihatan manusia (Qiang Ji, 2010). Computer vision masih memiliki banyak bidang di dalamnya, seperti Image Restoration, Image Reconstruction, Image Enhancement, Object Recognition, Object Detection dan masih banyak lagi (Szeliski, 2010).

Seiring dengan perkembangan teknologi, algoritma machine learning dapat digunakan untuk memecahkan masalah pada berbagai bidang pada computer vision. Pada tahun 2012, Krizhevsky, dkk melakukan gebrakan dalam kompetisi ImageNet dimana seluruh akademika dan professional berlomba untuk membuat algoritma yang dapat mengenali obyek pada sebuah citra. Peneliti menggunakan Convolutional Neural Network (CNN) dengan hasil Top-5 error sebesar 15%, sedangkan rata-rata manusia memiliki Top-5 error sebesar 5% (Krizhevsky, 2012).

2.5. *Neural Networks*

Neural Networks (NN) adalah model yang distimulasi berdasarkan struktur saraf otak. Otak pada dasarnya belajar dari pengalaman. Bertentangan dengan model matematika tradisional, yang diprogramkan, NN mempelajari hubungan antara input dan output yang dipilih. Dengan NN, seseorang akan memiliki akses ke lingkungan pemodelan yang kuat yang memungkinkan satu tes dan mengeksplorasi model simulasi lebih cepat dan lebih mudah daripada sebelumnya. Pelatihan model didasarkan pada data digital. Input dan output data akan diperkenalkan ke jaringan dengan menggunakan program jaringan saraf. Ketika pelatihan sudah siap, model siap untuk memprediksi kinerja data. Dalam bab berikut akan dijelaskan secara singkat perbandingan antara otak dan NN untuk mendapatkan pemahaman tentang NN.

Neural network adalah elemen pemrosesan yang saling berhubungan yang berfungsi untuk memecahkan masalah tertentu. Neural network biasanya terdiri dari 3 lapisan yang berbeda. Lapisan masukan, hidden layer dan lapisan keluaran. Lapisan-lapisan ini saling berhubungan dengan sejumlah simpul yang berisi fungsi aktivasi di hidden layer. Data masukan dalam bentuk pola matriks disajikan ke dalam neural network melalui lapisan input yang kemudian dihubungkan ke hidden layer, di mana data masukan tersebut diproses untuk menentukan bobot koneksi. Demikian pula hidden layer terhubung ke lapisan output dan lagi diproses menggunakan bobot koneksi yang sama, yang merupakan output dari neural network. Gambar 2.4 merupakan model neural network menggunakan tiga lapisan,



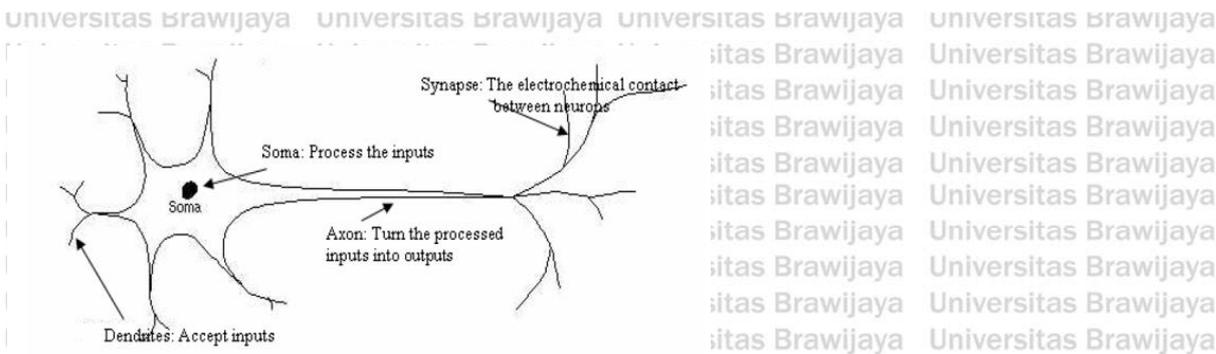
Gambar 2.4 : Neural Network

2.5.1. Analogi Otak Manusia

Komponen paling dasar dari ANN dimodelkan berdasarkan struktur otak. Beberapa bagian struktur ANN memang relatif tidak mirip dengan jaringan saraf otak dan beberapa bagian tidak mempunyai hubungan dengan jaringan saraf di otak secara biologis. Namun, ada bagian-bagian pada ANN memiliki kesamaan yang kuat dengan otak biologis dan karena itu banyak terminologi dipinjam dari neuroscience.

Otak pada dasarnya terdiri dari sejumlah besar (sekitar 10 miliar) neuron, yang saling terhubung secara besar-besaran. Neuron adalah unsur paling dasar dari otak manusia. Neuron adalah sel-sel tertentu, yang mengontrol kemampuan seperti identifikasi, berpikir, dan menerapkan pengalaman sebelumnya untuk setiap tindakan. Neuron-neuron ini saling berhubungan dengan hingga 200000 neuron lain. Kemampuan otak ini berasal dari jumlah komponen-komponen ini dan banyak koneksi di antara mereka.

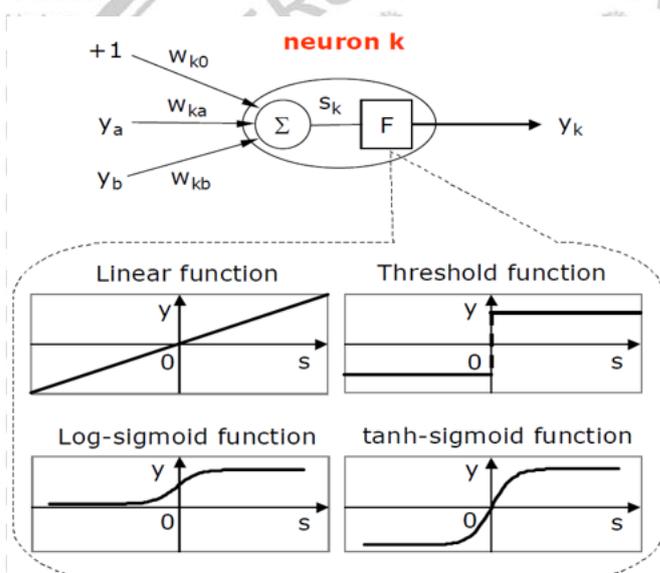
Semua neuron alami memiliki empat elemen dasar yaitu. dendrit, soma, akson, dan sinapsis. Dendrit menerima masukan dari sumber lain, yang dikombinasikan dalam beberapa cara di soma, melakukan operasi nonlinear pada hasil akson dan kemudian output akan dihasilkan pada tahap akhir yaitu sinapsis. Gambar 2.5 menunjukkan neuron biologis yang disederhanakan dan hubungan antara keempat komponen ini.



Gambar 2.5 : *Neuron* secara biologis
Sumber: (Kaiadi 1990)

2.5.2. Artificial Neuron

Unit dasar dari *neuron* buatan disimulasikan dari model biologis tetapi jauh lebih sederhana. Gambar 2.5 menunjukkan dasar-dasar dari *neuron* buatan.



Gambar 2.6 : *Neuron* ANN dan fungsi transfer
Sumber: (Kaiadi 1990)

Input diwakili oleh simbol matematika (y_a), (y_b). Setiap *input* adalah dikalikan dengan bobot sambungan yang ditentukan, yang diwakili oleh simbol matematika (w_{kj}). Selain itu ada *input* tambahan yang sama dengan +1 untuk setiap *neuron* dan bobotnya yang sesuai dengan simbol matematika (w_{k0}) disebut bias yang memperkenalkan off-set ke fungsi transfer (lihat bagian 2), yang memungkinkan *neuron* untuk memiliki *output*, bahkan jika *output* sama dengan nol. Pada langkah berikutnya beberapa operasi matematika (penjumlahan) dilakukan dan produk operasi ini diberi makan melalui fungsi transfer

nonlinier untuk menghasilkan *output*. Konstruksi semua jaringan ANN adalah sama tetapi beberapa dasar mungkin berbeda dalam yang berbeda.

2.5.3 Fungsi Aktivasi

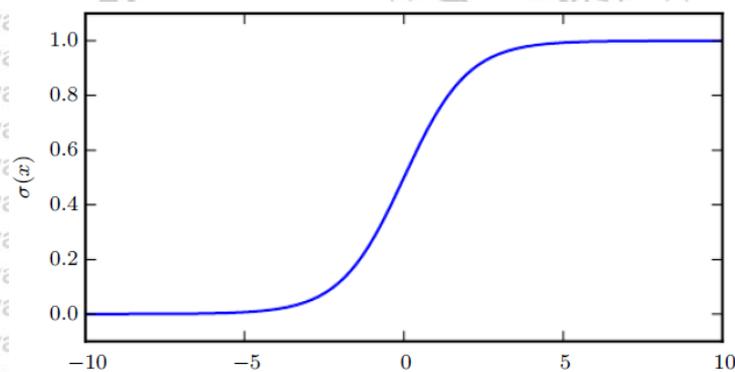
Fungsi aktivasi adalah sebuah fungsi yang menentukan apakah neuron akan aktif jika diberi masukan nilai tertentu. Secara umum terdapat dua jenis fungsi aktivasi, yaitu Fungsi Aktivasi Linear dan Non-Linear. Pemilihan fungsi aktivasi bergantung pada nilai data dan tujuan kita dalam melatih sebuah neural network. Seiring dengan pesatnya perkembangan teknologi deep learning, banyak fungsi aktivasi yang dikembangkan untuk menutupi kelemahan fungsi aktivasi yang sudah ada.

2.5.3.1 Sigmoid

Fungsi sigmoid umumnya digunakan untuk menghasilkan parameter yang berada pada rentang 0 sampai 1 saja. Fungsi sigmoid dirumuskan dengan persamaan sebagai berikut:

$$\sigma(x) = 1/(1 + e^{-x}) \quad (2.1)$$

Fungsi sigmoid akan mengalami suatu kondisi saturasi apabila memiliki masukan yang bernilai sangat positif maupun sangat negatif. Sigmoid memiliki persamaan sebagai berikut:



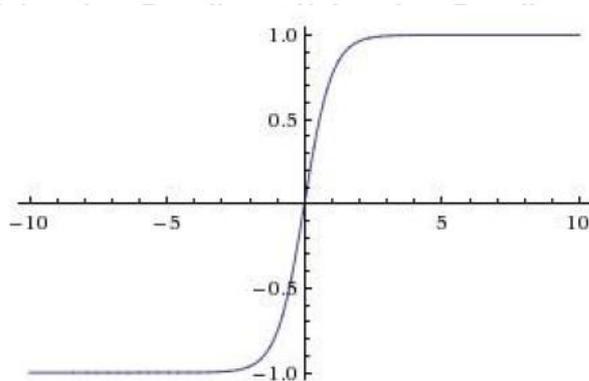
Gambar 2.7 Fungsi Sigmoid

2.5.3.2 Tanh

Fungsi Tanh memiliki sifat hampir sama seperti fungsi sigmoid hanya saja tanh memiliki rentang output antara -1 sampai dengan 1 dan zero centered. Gambar 2.7 menunjukkan grafik dari tanh. Sedangkan tanh dirumuskan dengan persamaan sebagai berikut:

berikut:

$$f(x) = \tanh(x) \quad (2.2)$$



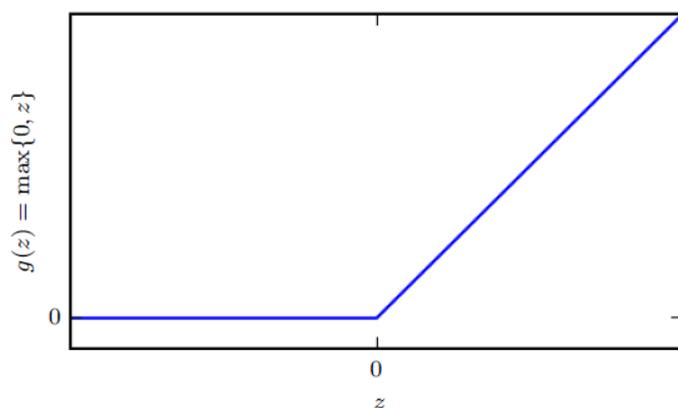
Gambar 2.8 Fungsi Tanh

2.5.3.3 Rectified Linear Unit (ReLU)

Rectified Linear Unit (ReLU) merupakan aktivasi yang paling sering digunakan dalam arsitektur neural network modern (Ian goodfellow, 2016). ReLU memiliki rumus matematika sebagai berikut:

$$f(x) = \max(0, x) \quad (2.3)$$

Fungsi ReLU memiliki beberapa karakteristik khusus yaitu tidak mengalami saturasi pada daerah dengan masukan positif, sangat efisien dalam proses komputasi dan lebih cepat mencapai titik konvergensi dibandingkan dengan sigmoid ataupun tanh. Gambar 2.8 menunjukkan grafik dari ReLU.

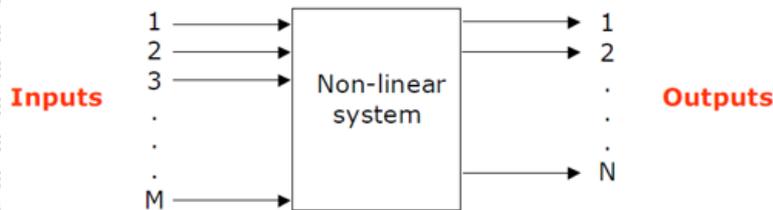


Gambar 2.9 Fungsi ReLU

2.6. Artificial Neural Network (ANN)

ANN merupakan interkoneksi *neuron* buatan. ANN mempelajari hubungan antara *input* dan *output* yang dipilih dari pengalaman sebelumnya. ANN juga melakukan tugasnya secara bersamaan (yaitu pemrosesan paralel), yang membuat ANN menjadi sangat cepat.

Sebuah ANN yang khas dapat mengidentifikasi dan mempelajari hubungan antara *input* dan *output* dari sistem multi-dimensi non-linear (Gambar 2.9).



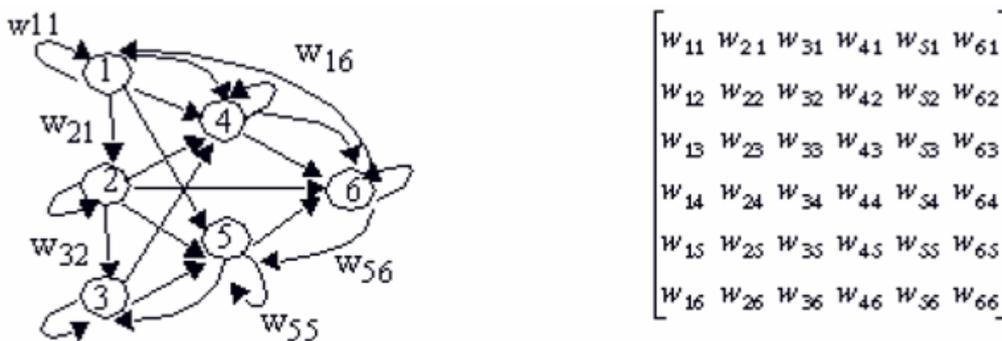
Gambar 2.10 : Sistem *non-linear* multi-dimensi

2.6.1. Arsitektur ANN

Pada umumnya arsitektur ANN diklasifikasikan menjadi dua kelompok berdasarkan pada struktur koneksi, arsitektur network umpan maju (hanya diizinkan untuk melewati satu arah, yaitu dari *input* ke *output*) dan jaringan berulang (umpan balik).

2.6.2. Jaringan Umpan Balik

Ketika semua unit di semua lapisan saling terhubung ke semua unit di semua lapisan disebut *fully-connected* yang mana dalam banyak kasus yang paling umum di gunakan pada *recurrent networks*. Gambar 2.10 menyajikan ANN *fully-connected* dengan bobot matriks.



Gambar 2.11 : ANN *fully-connected* dan bobot matrix

Sumber: (Kaiadi 1990)

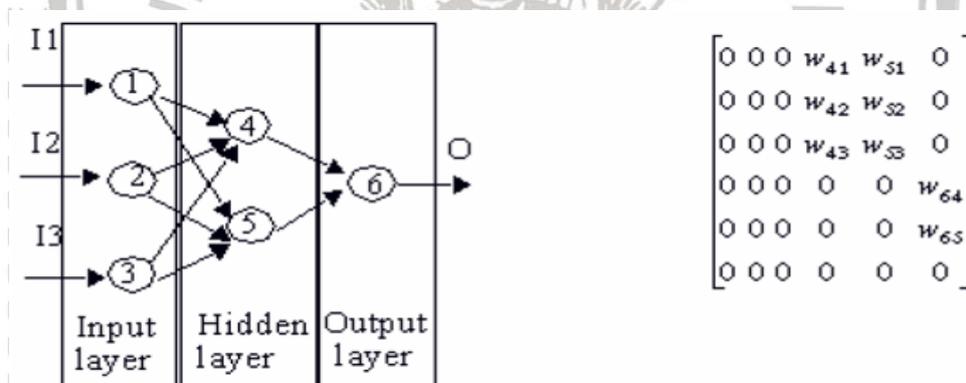
Dalam *recurrent network* beberapa koneksi mungkin tidak ada, tetapi ada koneksi yang melakukan umpan balik. “Masukan pada *recurrent network* pada waktu t , akan mempengaruhi nilai *output* jaringan untuk tahap proses di waktu berikutnya yang nilainya lebih besar daripada t ”. Oleh karena itu, *recurrent network* perlu dioperasikan dari waktu ke waktu (pengulangan) sampai didapat hasil *output* yang optimal.

2.6.3. Jaringan Umpan Maju

ANN didefinisikan sebagai umpan maju, jika matriks interkoneksi dibatasi hanya pada satu arah (tidak ada umpan balik atau koneksi sendiri). Jaringan umpan maju berlapis, yang disebut Multilayer perceptron (MLP). Perceptron dalam hal ini terdiri dari satu lapisan neuron buatan nonlinier tanpa koneksi umpan balik.

Arsitektur ANN, pada dasarnya terdiri dari satu set unit yang merupakan input layer, satu (atau lebih) hidden layer (s) dan output layer. Hidden layer tidak memiliki kontak langsung dengan lingkungan, oleh karena itu disebut hidden layer. Tidak ada proses komputasi yang dilakukan pada input layer dan karena itu komponennya disebut node input.

Gambar 2.11 menampilkan network umpan-maju dan matriks bobotnya.



Gambar 2.12 : Multilayer perceptron dan bobot matrix

Sumber: (Kaiadi 1990)

2.6.4. Pelatihan Artificial Neural Network

Proses pembelajaran disebut pelatihan dan dilakukan sesuai dengan aturan pembelajaran, yang diklasifikasikan ke dalam dua jenis berikut utama:

- *Supervised learning*: di mana kedua *input* dan *output* diketahui, ini berarti jaringan dapat menentukan prediksi dari *input* yang diberikan.
- *Unsupervised learning*: di mana *output* tidak diketahui dan *neuron* harus menemukan cara untuk mengelolanya, metode ini tidak dipertimbangkan dalam penelitian ini.

2.6.5. Learning Rate

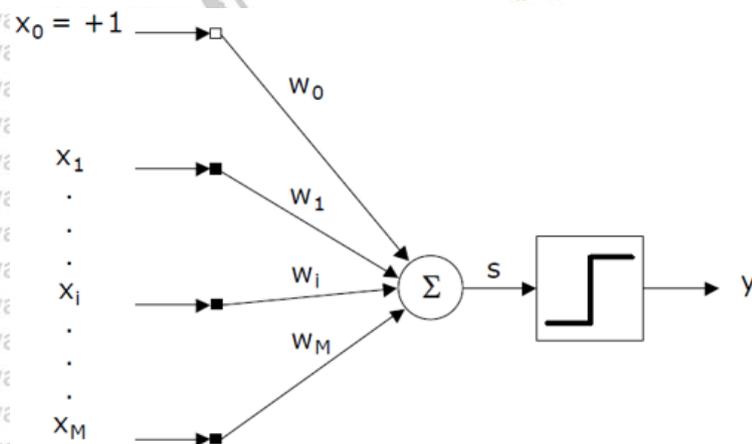
Sebagian besar fungsi pembelajaran memiliki beberapa istilah untuk *learning rate*, atau belajar konstan. Biasanya istilah ini positif dan antara nol dan satu. Dengan memilih tingkat yang lebih rendah, lebih banyak waktu dihabiskan untuk melatih ANN tetapi hasilnya akan lebih stabil. Dengan *learning rate* yang lebih cepat, pelatihan membutuhkan waktu lebih sedikit tetapi keakuratan hasilnya akan lebih buruk. Beberapa faktor lain dapat berperan dalam menentukan berapa lama waktu yang dibutuhkan untuk melatih jaringan seperti: Kompleksitas jaringan (jumlah *neuron* dan *hidden layer*), ukuran data, arsitektur dan jenis aturan pembelajaran.

2.6.6. Aturan Pembelajaran

Sebagian besar aturan pembelajaran adalah semacam variasi dari aturan pembelajaran paling terkenal dan tertua, aturan Hebb. Hebb menerbitkan bukunya *The Organization of Behaviour* pada tahun 1949, di mana ia mengusulkan bahwa pembelajaran fisiologis dilakukan melalui modifikasi sinaptik di otak, dan dengan demikian, ia memperkenalkan aturan pertama yang diketahui untuk pembelajaran self-organizing. Beberapa hukum pembelajaran utama disajikan sebagai contoh. Aturan *perceptron* dan delta atau LMS (*Least Mean Square error*) aturan, baik menggunakan pelatihan yang diawasi, dan bekerja secara iteratif untuk memperbarui bobot mereka.

2.6.7. Aturan Pembelajaran Perceptron

Perceptron terdiri dari *neuron* buatan nonlinear, yang memiliki fungsi aktivasi *threshold*. Gambar 2.12 menunjukkan *perceptron* dengan satu *neuron*.



Gambar 2.13: Perceptron

Sumber: (Kaiadi 1990)

Persamaan (2.7) menyatakan proses matematis dalam *perceptron*:

$$\begin{cases} s = \sum_{i=0}^M w_i \cdot x_i \\ y = F(s) \end{cases} \quad (2.4)$$

Error didefinisikan sebagai perbedaan antara *output* yang diinginkan (d) dan *output* aktual; persamaan (2.8) menyatakan fungsi *error*:

$$e(n) = d(n) - y(n) \quad (2.5)$$

Di mana n adalah nomor iterasi dan e adalah *error*. Pada awal pelatihan, bobot diberi nilai

acak, dan dengan memulai proses, *error* akan dihasilkan. *Perceptron* memiliki algoritme untuk menghasilkan kumpulan bobot baru dengan bantuan *error* yang dihitung. Persamaan (2.9) menyatakan fungsi bobot:

$$w(n+1) = w(n) + \eta \cdot e(n) \cdot x(n) \quad (2.6)$$

2.6.8. Aturan Delta

Aturan delta adalah salah satu yang paling umum digunakan. Aturan ini mengubah bobot dalam cara yang meminimalkan *Least Mean Square* jaringan. Aturan ini juga disebut Aturan Pembelajaran *Least Mean Square* (LMS). Persamaan (2.10) menyatakan fungsi *error*:

$$E(w) = \frac{1}{2} e^2(n) \quad (2.7)$$

Fungsi *threshold* digantikan oleh fungsi transfer linear, dan oleh karena itu *error* dapat dinyatakan sebagai

$$e(n) = d(n) - x^T(n) \cdot w(n) \quad (2.8)$$

Dimana x dan w adalah *input* dan vektor bobot.

“Asumsi utamanya adalah bahwa bobot optimal dapat ditemukan dalam arah gradien kemiringan fungsi *error* sehubungan dengan bobot (metode penurunan paling curam), yaitu [2]“

$$\Delta w(n) = -\eta \cdot \frac{\partial E(w)}{\partial w(n)} \quad (2.9)$$

Persamaan differensiasi (2.8)

$$\frac{\partial E(w)}{\partial w(n)} = e(n) \cdot \frac{\partial e(n)}{\partial w(n)} \quad (2.10)$$

Memasukkan persamaan (2.11) ke dalam persamaan (2.13)

$$\frac{\partial e(n)}{\partial w(n)} = \frac{\partial (d - x^T \cdot w)}{\partial w(n)} = -x(n) \quad (2.11)$$

Dan akhirnya, vektor bobot baru sama dengan:

$$w(n+1) = w(n) + \Delta w(n) = w(n) + \eta \cdot e(n) \cdot x(n) \quad (2.12)$$

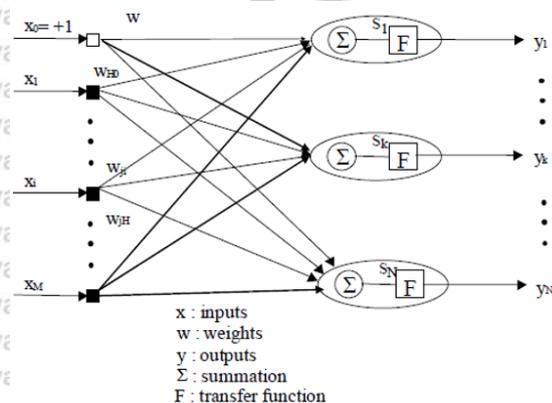
Ketika, di antara dua iterasi berulang, *error* tidak terjadi dari jumlah yang diberikan, pelatihan harus diakhiri dan bobot akan ditetapkan.

2.6.9. Backpropagation Neural Network

ANN mempunyai sangat banyak tipe network, tetapi yang difokuskan dalam penelitian ini, adalah *Multilayer perceptron* (MLP) dan *Convolutional Neural Network* (CNN) yang akan dilatih menggunakan algoritma backpropagation.

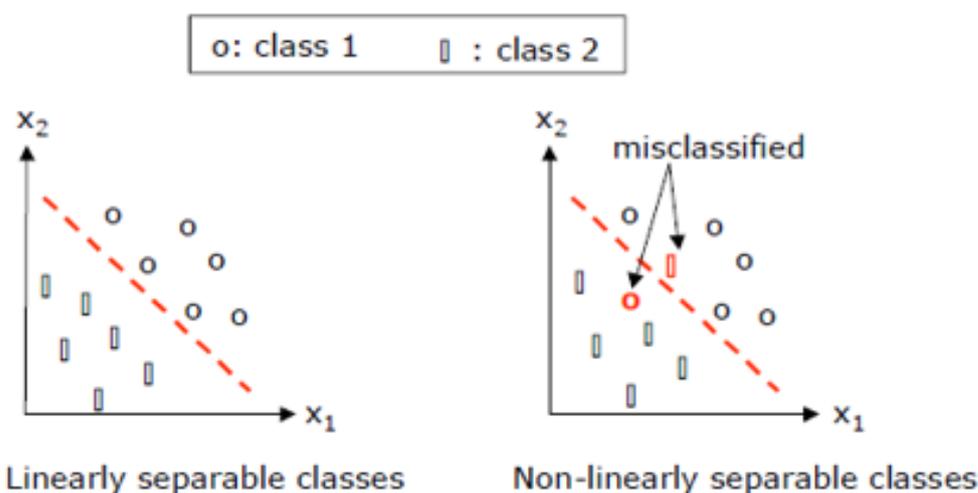
2.6.9.1. Single Layer Neural Network

Jaringan *single-layer* hanya terdiri dari satu lapisan *neuron*, sebenarnya ada dua, tetapi karena hanya yang terakhir dari lapisan yang membuat perhitungan yang sebenarnya, itu disebut jaringan *single-layer*. Alasan mengapa ini disebut sistem umpan-maju karena informasi hanya menyebar dari *input* ke *output* (lihat gambar 2.13).



Gambar 2.14 : Struktur jaringan *single-layer* umpan-maju

Masalah utama dengan jaringan single-layer adalah ketidakmampuan untuk melakukan klasifikasi fungsi non-linear (lihat gambar 2.14).

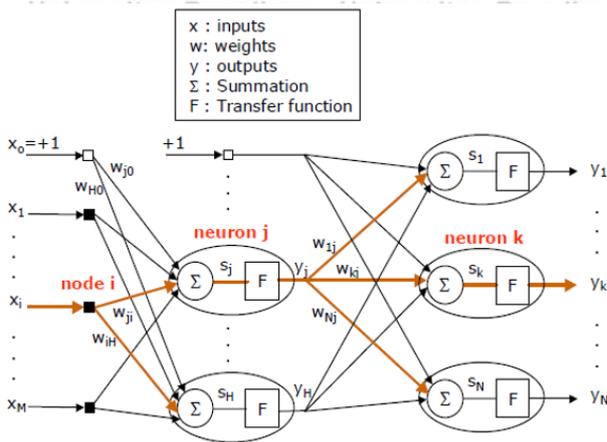


Gambar 2.15 : Linear dan Non-Linear
Sumber: (Kaiadi 1990)

Kerumitan *perceptrons single-layer* tidak cukup untuk menangani masalah kategorisasi yang lebih besar. Kedalaman logis dari masalah semacam ini terlalu besar untuk satu *perceptron* untuk ditangani. Untuk mengatasi masalah ini, dalam jaringan dibuat beberapa layer. Ini biasanya disebut sebagai MLP (*Multi Layer Perceptrons*).

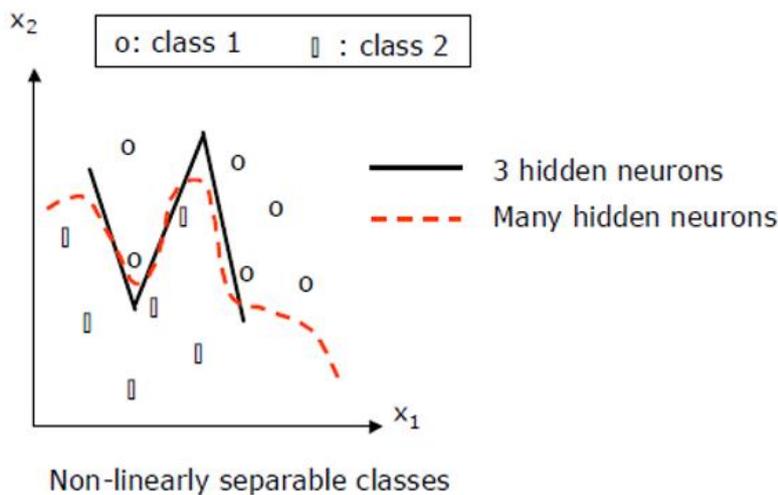
2.6.9.2. Multilayer Perceptron (MLP)

Pada jenis jaringan ini, satu atau lebih *hidden layer* berada di antara *input layer* dan *output layer* dan mereka akan membantu jaringan untuk memecahkan masalah yang lebih kompleks. Setelah jumlah *input* dan *output* telah diputuskan, jumlah *neuron* dalam *input* dan *output layer* ditetapkan. Jumlah optimal dari *neuron* tersembunyi di *hidden layer*, bagaimanapun, harus dipilih. Menemukan jumlah yang optimal dari *hidden layer* adalah dengan menggunakan *trial* dan *error*, tetapi di sini adalah aturan praktis untuk memulai dengan nilai awal yang baik dan itu dimulai dengan jumlah parameter *input* dan dengan demikian meningkatkan jumlah *hidden layer*. Multi Layer Perceptron terdiri dari beberapa *neuron* yang berada pada beberapa layer dan semua *neuron* di semua layer tersebut saling berhubungan satu dengan yang lain. Setiap hubungan tersebut memiliki nilai bobot yang akan selalu diperbaharui pada proses learning untuk mendapatkan keluaran yang optimal. Struktur Multi Layer Perceptron tampak pada gambar 2.15.



Gambar 2.16 : Struktur jaringan MLP umpan-maju satu *hidden layer*

Menggunakan jaringan umpan maju *multi-layer* akan memecahkan masalah mengklasifikasikan fungsi non-linear (Gambar 2.16).

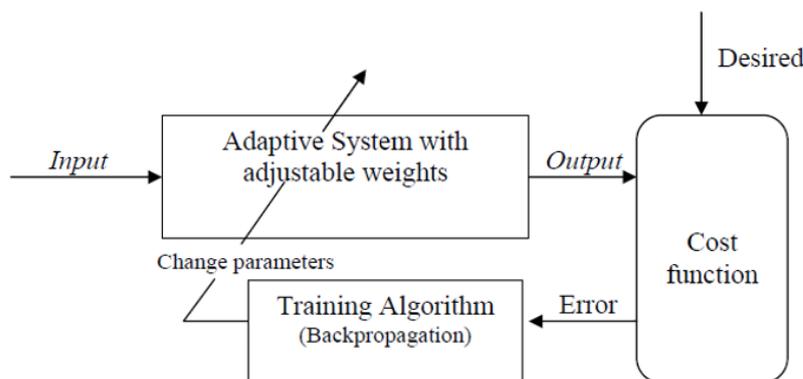


Gambar 2.17 : Kemampuan MLP melakukan klasifikasi fungsi non-linear

2.6.9.3. Algoritma *Backpropagation*

Backpropagation adalah dasar untuk melatih *supervised neural network*. Data yang digunakan sebagai *input* dikirimkan melalui jaringan, melewati layer demi layer, dan menghasilkan satu set *output* yang sudah ditentukan (*desired*). Selama data diteruskan melalui jaringan nilai bobot jaringan ditetapkan. Nilai *output* yang diperoleh dibandingkan dengan nilai *output* yang diinginkan (*desired*), disebut sebagai *backward pass*; perbedaan antara *output* yang (*desired*) dan *output* yang dihitung (*error*) digunakan untuk menyesuaikan kembali bobot pada jaringan untuk mengurangi tingkat *error* (lihat Gambar

2.17). nilai desired adalah Proses ini dilakukan berulang, yang berlanjut sampai tingkat *error* yang dapat diterima akan diperoleh. Setiap kali jaringan memproses seluruh rangkaian data (baik yang maju maupun mundur), disebut *epoch*. Jaringan dengan cara ini dilatih dan *error* berkurang disetiap *epoch* sampai didapat tingkat *error* yang dapat diterima. Metode ini disebut pelatihan *error back-propagation*.



Gambar 2.18 : Backpropagation Training
Sumber: (Kaiadi 1990)

2.6.10 Loss Function

Loss function adalah fungsi yang akan digunakan untuk menghitung selisih dari target dan keluaran sebuah neural network. Terdapat berbagai macam loss function, namun yang sering digunakan untuk regresi adalah mean squared error (MSE) dan mean average error (MAE). Sedangkan untuk klasifikasi adalah crossentropy loss. Gradient dari loss function ini akan dihitung dengan menggunakan algoritma backpropagation untuk melihat hubungan antara perubahan bobot pada setiap layer terhadap output sebuah neural network dan melakukan update terhadap bobot-bobot tersebut untuk mengurangi loss sekecil mungkin.

2.6.10.1 Weighted Cross Entropy Loss

Fungsi Cross Entropy adalah sebuah fungsi loss yang digunakan untuk tujuan klasifikasi. Cross Entropy digunakan untuk mengukur seberapa jauh hasil prediksi yang biasanya memiliki bentuk sebuah probability distribution dari target yang telah ditentukan.

Persamaan dari Cross Entropy dapat dituliskan sebagai berikut:

$$CE = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}) \quad 2.13$$

Dimana y adalah target dan \hat{y} merupakan keluaran dari sistem atau hasil prediksi. Pada penelitian ini akan digunakan bentuk lain dari Cross Entropy dengan menambahkan konstanta pada fungsi Cross Entropy. Penambahan konstanta ini diharapkan dapat mengatasi permasalahan distribusi data yang tidak seimbang. Persamaan dari Weighted Cross Entropy dapat dituliskan sebagai berikut:

$$WCE = -y \log \hat{y} \theta - (1 - y) \log(1 - \hat{y}) \theta \quad 2.14$$

Dimana y adalah target, \hat{y} merupakan keluaran dari sistem atau hasil prediksi dan θ merupakan koefisien yang merupakan variable bebas yang akan dicari nilai yang paling optimal.

2.6.11 Feature Scaling (MinMax Normalization)

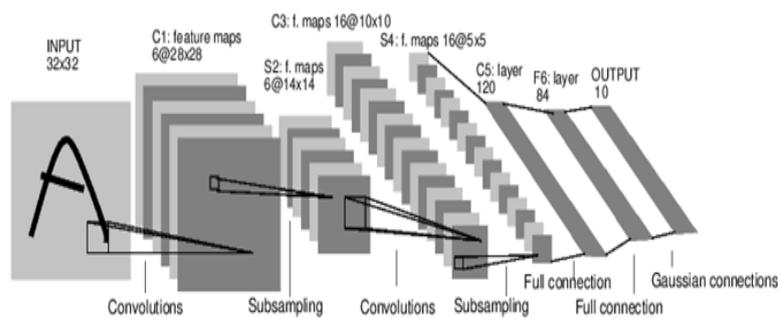
Feature scaling adalah proses yang dilakukan untuk merubah rentang data yang awalnya berada pada rentang yang besar menjadi rentang yang lebih kecil sehingga proses training menjadi lebih cepat. Pada data yang berbentuk citra RGB, seluruh nilai pada tiap *channel* berada pada rentang 0 hingga 255. Feature scaling dalam bentuk MinMax Normalization dilakukan untuk merubah feature sehingga berada pada rentang 0 hingga 1. Persamaan dari MinMax Normalization dapat dituliskan sebagai berikut:

$$\bar{x} = \frac{x - |\min(x)|}{\max(x) - \min(x)} \quad 2.15$$

2.7. Convolutional Neural Network (CNN)

Convolutional Network atau yang dikenal dengan *Convolutional Neural Network* (CNN) adalah salah satu metode dalam *neural network* yang digunakan dalam pengolahan data dengan metode topologi grid-like dengan menggunakan operasi matematika konvolusi (Ian goodfellow, 2016).

Metode ini pertama kali diperkenalkan oleh Le Cun pada tahun 1989 dan sukses diaplikasikan dalam sistem pengenalan angka dengan nilai akurasi yang tinggi. Pada umumnya CNN terdiri dari *convolutional layer*, *pooling layer* dan diakhiri dengan *fully connected layer* yang kemudian output dari *layer* tersebut diklasifikasi oleh *classifier*.



Gambar 2.19 Arsitektur CNN Untuk Pengenalan Angka

Berbeda dengan *neural network* pada umumnya, *convolutional neural network* memiliki tiga karakteristik khusus yang mampu meningkatkan performa dari suatu sistem machine learning. Dua karakteristik khusus tersebut adalah *sparse interactions* dan *parameter sharing*.

Sparse interactions memiliki arti bahwa tidak semua unit antar layer saling dihubungkan oleh suatu parameter. Sebagai contoh ketika suatu sistem memiliki m masukan dan n keluaran, maka total parameter yang dibutuhkan pada sistem yang menggunakan *neural network* adalah $m \times n$ parameter, sedangkan pada sistem yang menggunakan *convolutional neural network* total parameter yang dibutuhkan hanya $k \times m$ parameter dimana k bernilai jauh lebih kecil dari m .

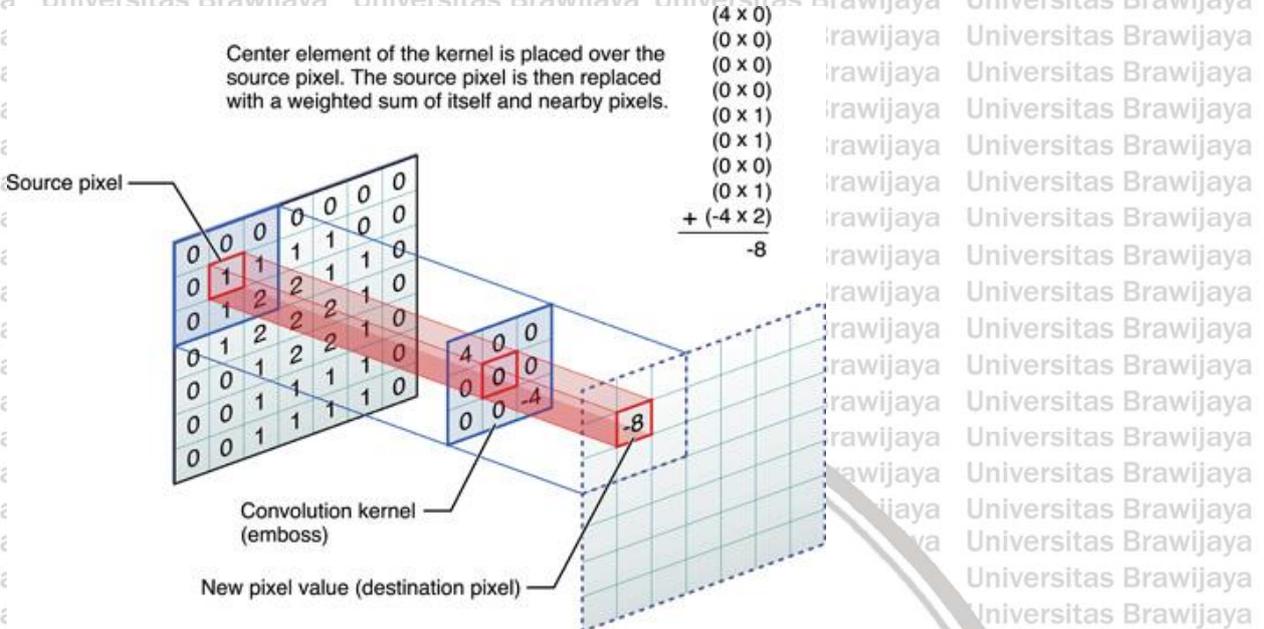
Karakteristik khusus kedua yang dimiliki CNN adalah *parameter sharing* yang memiliki arti bahwa parameter yang ditetapkan digunakan untuk lebih dari satu fungsi didalam suatu model. Pada *neural network*, setiap parameter yang menghubungkan antar unit hanya digunakan satu kali saja saat proses komputasi, sedangkan pada CNN parameter yang ditetapkan akan digunakan pada proses komputasi dari seluruh masukan dalam menghasilkan satu keluaran berupa *feature map*.

2.7.1 Convolutional Layer

Pada umumnya, konvolusi merupakan suatu operasi matematis pada dua fungsi yang memiliki nilai real. Didalam aplikasi machine learning, umumnya kita menggunakan data yang bersifat diskrit dan memiliki dimensi lebih dari 1 yang dapat dituliskan dengan rumus sebagai berikut:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i - m, j - n) K(i - m, j - n) \quad (2.16)$$

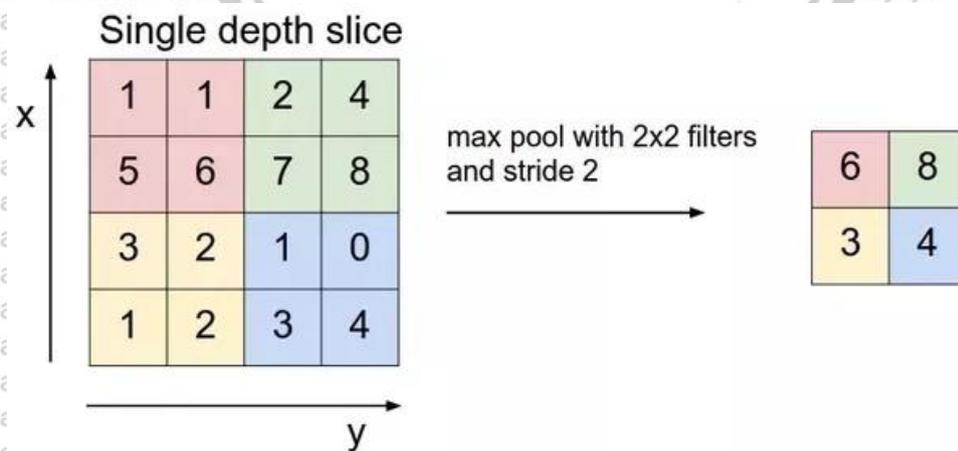
I adalah masukan sistem, K adalah kernel atau filter yang memiliki nilai parameter yang ditetapkan. Gambar 2.19 menunjukkan ilustrasi dari pengaplikasian convolutional layer pada data masukan.



Gambar 2.20 Ilustrasi dari pengaplikasian convolutional layer

2.7.2 Pooling Layer

Pooling layer memiliki fungsi untuk merubah nilai output pada lokasi tertentu sesuai dengan nilai-nilai output yang terdekat. Jenis pooling yang umum digunakan adalah max pooling. Gambar 2.20 menunjukkan cara kerja max pooling layer.



Gambar 2.21 Max Pooling Layer

Dalam gambar 2.20, dapat kita lihat sistem yang menggunakan max pooling layer dengan kernel berukuran 3×1 menghasilkan output berdasarkan nilai input terbesar dalam kernel tersebut. Akibat dari pengaplikasian layer ini, representasi dari output menjadi invariant terhadap perubahan kecil pada input. Hal ini memiliki arti bahwa apabila input mengalami sedikit perubahan, kebanyakan dari output hasil pooling layer tetap sama.

Pooling layer juga dapat digunakan dalam melakukan proses *downsampling* terhadap input sistem. Hal ini didasari karena dengan diaplikasikannya *pooling layer*, sistem telah mengetahui hubungan antar pixel yang saling berdekatan. Proses *downsampling* ini dapat diperoleh dengan mengaplikasikan *stride* atau pergeseran *kernel* sebesar lebih dari 1.

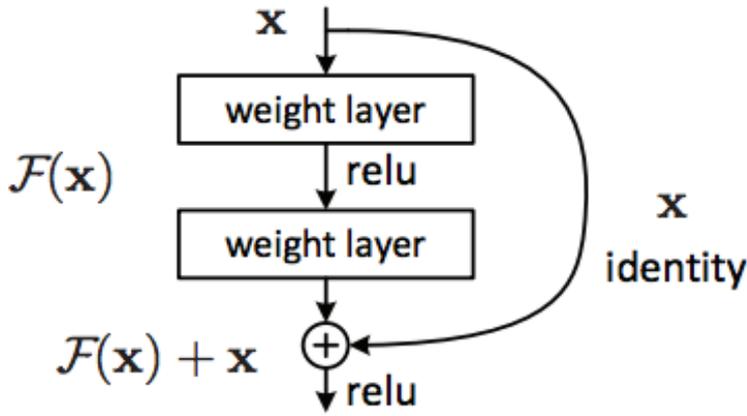
2.7.3 Residual Convolutional Neural Network

Residual Convolutional Neural Network adalah arsitektur CNN yang berbeda dari CNN pada umumnya. Jika pada standard CNN, *feature map* dari input akan dilakukan *forward propagation* menuju *output layer*. Pada saat *forward propagation*, nilai dari *feature map* tersebut akan berubah pada setiap layer dan *feature* dari layer sebelumnya tidak digunakan kembali. Hal tersebut menyebabkan semakin banyak layer yang digunakan maka *feature* yang dihasilkan kemungkinan akan sulit dikenali oleh *classifier*. Sedangkan pada Residual CNN, *feature* pada layer sebelumnya (Identity) akan digunakan kembali pada proses *forward propagation* pada layer selanjutnya, sehingga memungkinkan untuk menggunakan arsitektur dengan layer yang lebih banyak (*deeper*). Gambar 2.21 menunjukkan ilustrasi dari Residual Convolutional Neural Network.

Residual Convolutional Neural Network memiliki persamaan yang dapat dituliskan sebagai berikut:

$$\begin{aligned} A(i, j) &= (I * K)(i, j) = \sum_m \sum_n I(i - m, j - n) K(i - m, j - n) \\ B(i, j) &= (A * K)(i, j) \\ C(i, j) &= (B * K)(i, j) + A \end{aligned} \quad (2.17)$$

Dimana A merupakan hasil konvolusi pada layer pertama, B merupakan hasil konvolusi pada layer kedua dan C adalah hasil konvolusi pada layer ketiga yang melakukan penjumlahan hasil konvolusi dengan A.



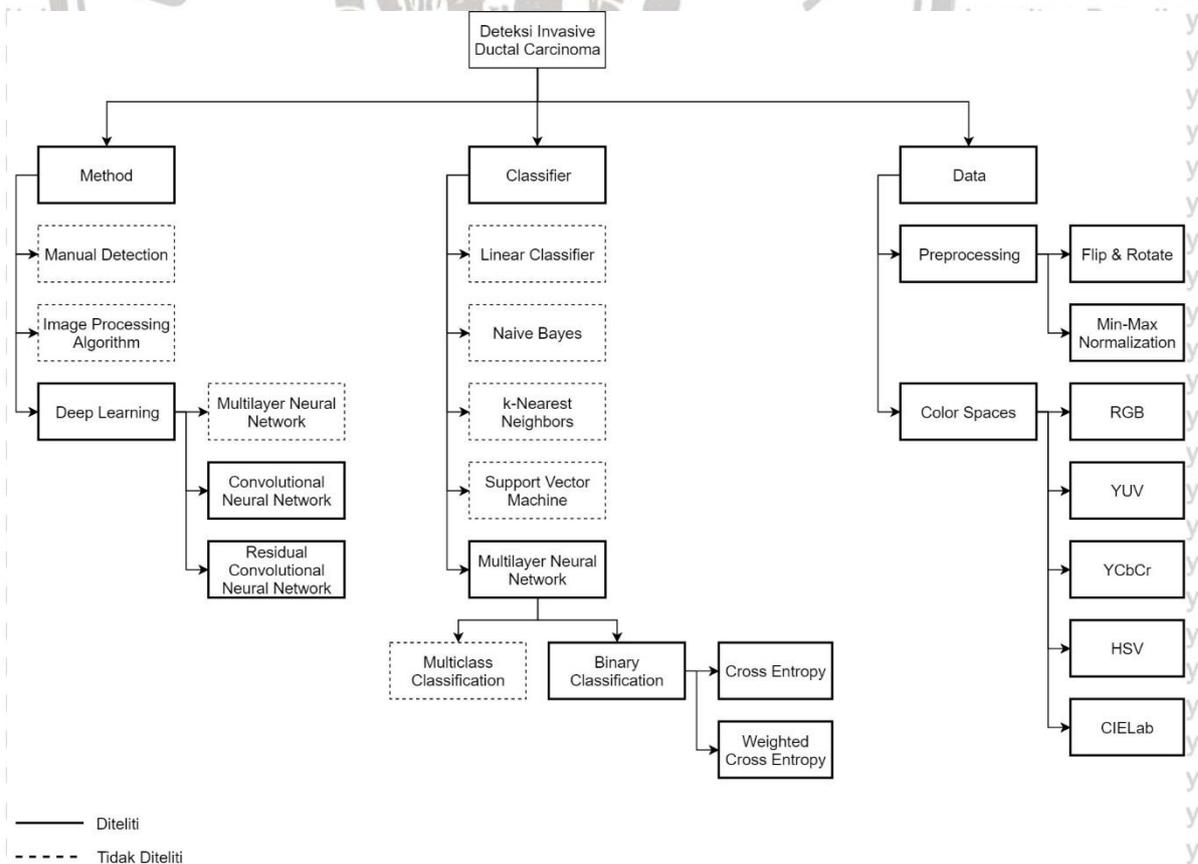
Gambar 2.22 Residual Convolutional Neural Network



BAB III KERANGKA KONSEP PENELITIAN

3.1. Kerangka Konsep Penelitian

Sistem deteksi kanker payudara jenis *Invasive Ductal Carcinoma* dengan menggunakan *Residual Convolutional Neural Network* pada penelitian ini memiliki kerangka konsep yang akan diuraikan pada bab ini. Metode umum yang digunakan adalah *Deep Learning* yang menggunakan *Residual Convolutional Neural Network*. Penelitian ini akan mereplikasi hasil dari penelitian awal yang telah dilakukan oleh Angel Cruz-Roa pada tahun 2014 yang menggunakan *convolutional neural network* konvensional untuk dijadikan sebagai pembanding. *Weighted Cross Entropy loss* digunakan sebagai *loss function* untuk menekan jumlah *false negative* pada saat pengujian sedangkan *data augmentation* digunakan untuk mengatasi distribusi data yang tidak seimbang dan beberapa *color spaces* akan digunakan sebagai input. Gambar 3.1 menunjukkan kerangka konsep dari penelitian ini.



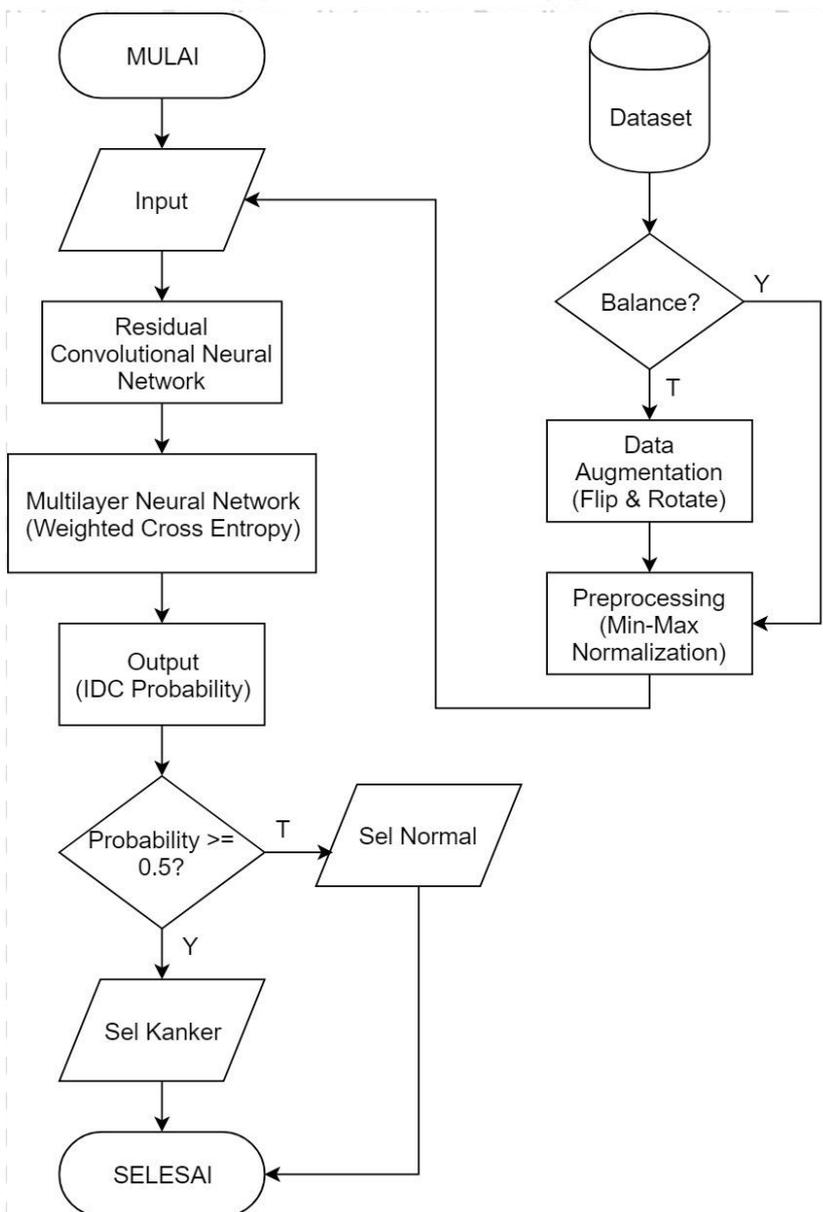
Gambar 3.1 Kerangka Konsep Penelitian

3.2 Analisis Masalah

Kanker payudara merupakan salah satu penyakit yang paling mematikan khususnya bagi kaum wanita. Salah satu jenis kanker payudara yang sering terjadi adalah *invasive ductal carcinoma* (IDC). Setidaknya 80% diagnosa kanker payudara adalah *invasive ductal carcinoma*. (BreastCancer.Org, 2018). Untuk menanggulangi permasalahan tersebut, deteksi dini penyakit kanker payudara sangat diperlukan. Saat ini metode manual digunakan oleh tenaga medis untuk mendeteksi keberadaan sel kanker pada sebuah citra biopsi, sehingga memungkinkan terjadi kesalahan. Untuk menanggulangi permasalahan tersebut perlu adanya sebuah *Computer Aided System* untuk membantu melakukan deteksi kanker pada citra biopsi. Pada penelitian yang telah dilakukan oleh Angel Cruz-Roa pada tahun 2014, telah diteliti penggunaan *convolutional neural network* konvensional untuk melakukan deteksi kanker terhadap dataset IDC dari Hospital of University of Pennsylvania and The Cancer Institute of New Jersey dengan hasil yang belum optimal yaitu nilai precision, sensitivity, specificity, F-Measure sebesar 0.654, 0.796, 0.888 dan 0.718 secara berurutan. Sehingga diperlukan metode lain untuk dapat meningkatkan performa tersebut.

3.3 Konsep Solusi

Data yang dibutuhkan dalam penelitian ini yaitu dataset IDC dari Hospital of University of Pennsylvania and The Cancer Institute of New Jersey. Pada Gambar 3.2 langkah pertama adalah membagi dataset menjadi dua bagian, satu bagian sebagai training dan satu bagian lain sebagai testing. *Preprocessing* dalam bentuk augmentation dilakukan jika distribusi data tidak seimbang. Selanjutnya *preprocessing min-max normalization* dilakukan sesuai dengan *color spaces* yang akan digunakan. *Residual Convolutional Neural Network* digunakan sebagai *feature extractor*. *Feature* ini akan diklasifikasikan dengan menggunakan *Multilayer Neural Network* dengan metode *binary classification* dengan *weighted cross entropy* sebagai *loss function* untuk menekan jumlah *false negative* pada saat pengujian. Training dilakukan secara *end-to-end* mulai dari *residual convolutional neural network* sebagai *feature extractor* hingga *multilayer neural network* sebagai *classifier* secara berurutan. Berikut Gambar 3.2 menjelaskan kerangka solusi pada penelitian ini.



Gambar 3.2 Kerangka Solusi

3.4 Hipotesis

Residual Convolutional Neural Network memiliki performa yang lebih baik jika dibandingkan dengan *convolutional neural network* konvensional untuk mendeteksi *Invasive Ductal Carcinoma* (IDC). Untuk mengatasi distribusi data yang tidak seimbang akan dilakukan teknik augmentasi data sehingga distribusi data antara citra positif dan citra negatif menjadi seimbang. Pada penelitian ini akan digunakan *multilayer neural network* sebagai *classifier* dengan menggunakan *weighted cross-entropy loss* pada saat training untuk menekan tingkat *false negative* atau meningkatkan nilai *recall* atau *sensitivity* pada saat pengujian dan F1-Score secara keseluruhan. Untuk menentukan nilai θ yang optimal pada

weighted cross entropy, akan dilakukan pelatihan dan pengujian dengan nilai θ yang naik secara bertahap. Nilai θ dengan hasil pengujian terbaik yang akan dipilih. Sehingga akan dihasilkan sebuah sistem yang dapat mendeteksi *Invasive Ductal Carcinoma* (IDC) dengan performa yang optimal.





BAB IV

METODE PENELITIAN DAN PERANCANGAN SISTEM

4.1. Tahapan Penelitian

Adapun langkah-langkah yang dilakukan dalam melakukan penelitian ini dibagi menjadi beberapa bagian yaitu :

1. Studi Literatur

Merupakan langkah yang bertujuan untuk mencari landasan teori untuk membantu dalam pembuatan sistem secara keseluruhan. Langkah ini dilakukan dengan cara membaca buku-buku literatur, *paper*, *video lecture* dan wawancara pada dosen pembimbing.

2. Perancangan Sistem

Perancangan perangkat lunak menggunakan bahasa pemrograman Python secara keseluruhan, library Numpy, Scikit-Image dan OpenCV untuk pre-processing data dan library Tensorflow untuk merancang dan melatih *Residual Convolutional Neural Network* dan *Multilayer Neural Network*.

3. Pengujian Sistem

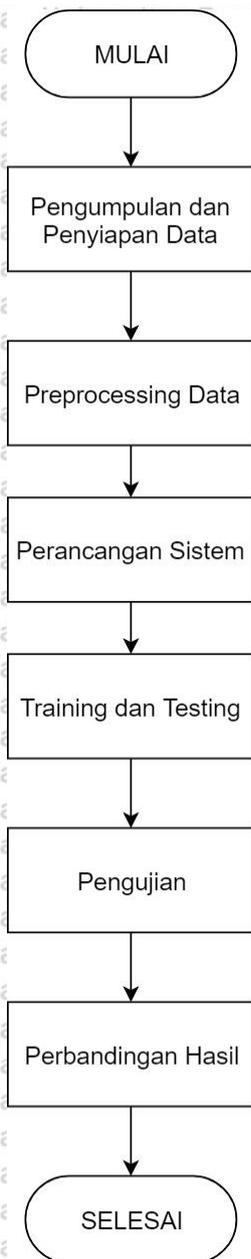
Pengujian ini dilakukan dengan tujuan sistem berjalan dengan sempurna sesuai dengan keinginan. Pengujian yang dilakukan adalah pengujian dengan menggunakan data uji yang merupakan bagian dari dataset. Ketika mengalami kesalahan pada pengujian, maka sistem akan diperbaiki sampai berjalan sesuai dengan keinginan.

4. Penyusunan Laporan

Penyusunan laporan ini dilakukan setelah semua prosedur penelitian selesai dilakukan. Pelaporan ini dilakukan secara mendetail agar dapat dijadikan literatur bagi berbagai pihak yang ingin mengembangkannya.

4.2. Metode Penelitian

Secara umum metode penelitian terdiri dari pengumpulan data, *preprocessing data*, *training*, *testing* dan perbandingan hasil. Metode penelitian yang digunakan pada penelitian ini secara rinci dapat dilihat pada gambar 4.1.

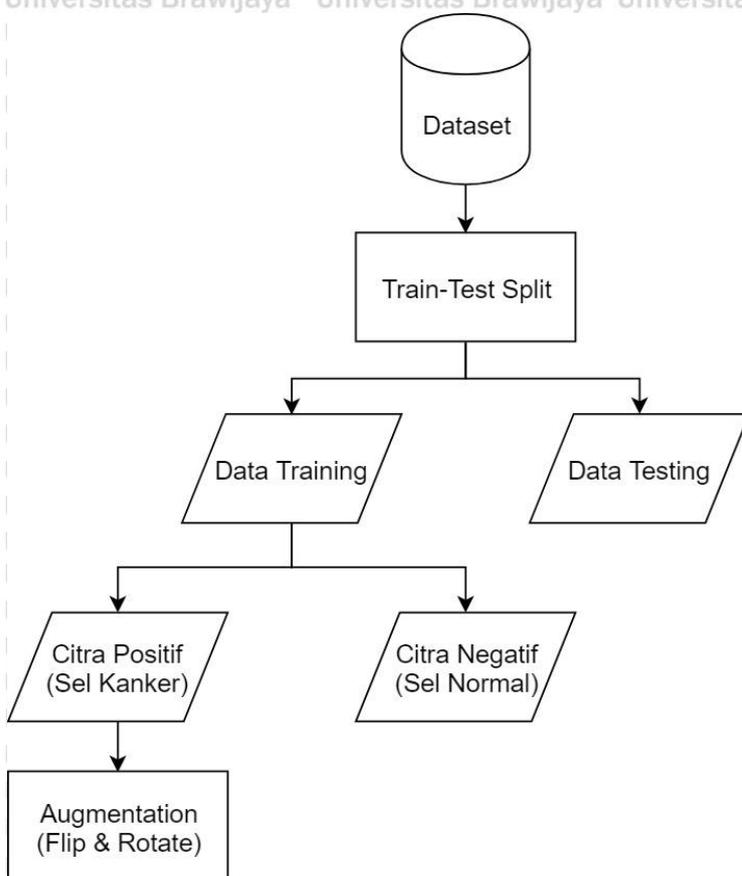


Gambar 4.1 Metode Penelitian

4.3. Pengumpulan dan Persiapan Data

Data-data yang digunakan dalam penyusunan penelitian ini adalah dataset IDC dari Hospital of University of Pennsylvania and The Cancer Institute of New Jersey. Pada *dataset* tersebut terdapat 162 slide citra biopsi hasil scan pada 40x pembesaran. Pada tiap *slide*, area yang terdapat sel kanker telah ditandai oleh *pathologist* dan selain dari area yang telah diberi label adalah sel normal. Setiap *slide* pada *dataset* tersebut akan dibagi-bagi menjadi beberapa *patch* yang lebih kecil berukuran 50 x 50 piksel sehingga dihasilkan data citra biopsi sebanyak 277.524 yang terdiri dari 198.738 citra negatif IDC dan 78.786 citra positif IDC.

Dari keseluruhan *dataset* tersebut nantinya akan diambil 80% sebagai *training* dan 20% sebagai *testing*. Pembagian data antara *training* dan *testing* dilakukan secara acak.



Gambar 4.2 Skema Pengumpulan dan Persiapan Data

Dari gambar 4.2, proses pengumpulan dan persiapan data dibagi menjadi dua tahapan, diantaranya:

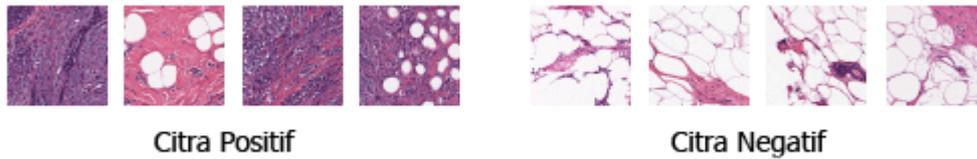
1. *Train-Test Split*

Tahap pertama adalah proses *train-test split* dengan membagi dataset menjadi 2 bagian, 80% untuk *training* dan 20% untuk *testing*. Pembagian data dilakukan secara acak.

2. *Data Augmentation*

Tahap kedua adalah *data augmentation* dengan melakukan *random flip* dan *rotate* yang dilakukan hanya pada citra positif karena jumlah data untuk citra positif lebih sedikit dari citra negatif. *Data augmentation* hanya dilakukan pada data *training* saja.

Proses ini dilakukan untuk menghindari bias pada sistem pada saat *training*.



Gambar 4.3 Citra Positif dan Citra Negatif pada Dataset

4.4. Pre-processing Data

Sebelum *dataset* siap digunakan untuk proses *training* dan *testing*, *dataset* akan melalui proses *preprocessing* terlebih dahulu. *Dataset* yang digunakan adalah citra biopsi dengan format RGB, akan diubah menjadi beberapa *color space* lain yaitu YUV, HSV, YCbCr dan CIELab. Proses konversi dilakukan dengan menggunakan *library* OpenCV.

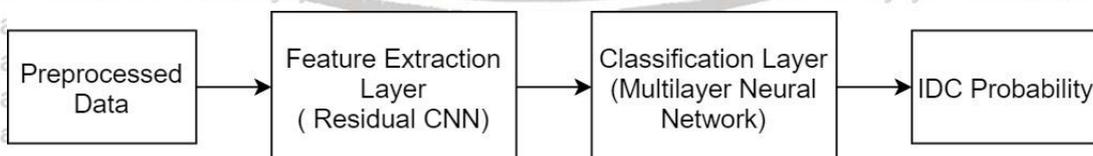
Setelah proses konversi selesai, akan dilakukan proses *feature scaling* menggunakan metode *min-max normalization* agar proses training menjadi lebih cepat dan optimal karena rentang dari *feature* berada pada nilai yang tidak terlalu jauh yaitu 0 hingga 1 atau -1 hingga 1.

Persamaan 4.1 menunjukkan rumus *min-max normalization*.

$$\bar{x} = \frac{x - |\min(x)|}{\max(x) - \min(x)} \quad 4.1$$

4.5. Perancangan Sistem

Dalam perancangan sistem secara umum, perangkat lunak yang digunakan adalah Tensorflow yang merupakan *library deep learning* yang memudahkan peneliti merancang *neural network* dan melakukan *training* hingga *testing*. Sedangkan *library* Numpy, Scikit-Image dan OpenCV akan digunakan pada saat *pre-processing*.



Gambar 4.4 Diagram Blok Sistem

Pada Gambar 4.4 dapat dijelaskan bahwa sebuah input yaitu citra biopsi yang sudah melalui proses *pre-processing* akan digunakan sebagai masukan dari *feature extraction layer* yang terdiri dari beberapa *residual convolution layer* dan *pooling layer*. Keluaran dari

feature extraction layer berupa *feature maps* yang akan dijadikan sebagai masukan dari *classification layer* yang terdiri dari *multilayer neural network* atau *multilayer perceptron*.

Keluaran dari *classification layer* adalah probability yang mengindikasikan apakah input yang digunakan merupakan sel kanker atau sel normal.

4.5.1. Perancangan *Feature Extraction Layer*

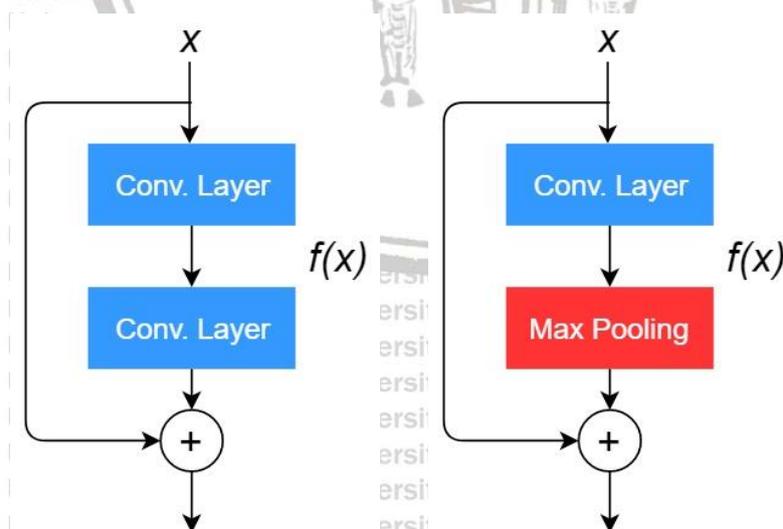
Feature extraction layer terdiri dari beberapa blok yang terdiri dari *residual convolution layer* dan *pooling layer*. *Pooling layer* yang digunakan adalah *Max Pooling Layer* yang berfungsi untuk mereduksi dimensi dari feature setelah proses konvolusi. Pada penelitian ini terdapat dua jenis residual block dengan konfigurasi sebagai berikut:

1. Residual Block Tipe I

Residual block tipe I terdiri dari dua unit convolution layer dengan filter sebesar 3x3 sebanyak 32 buah dengan *stride* 1 pada tiap layer. Keluaran dari residual block ini akan ditambahkan dengan masukannya.

2. Residual Block Tipe II

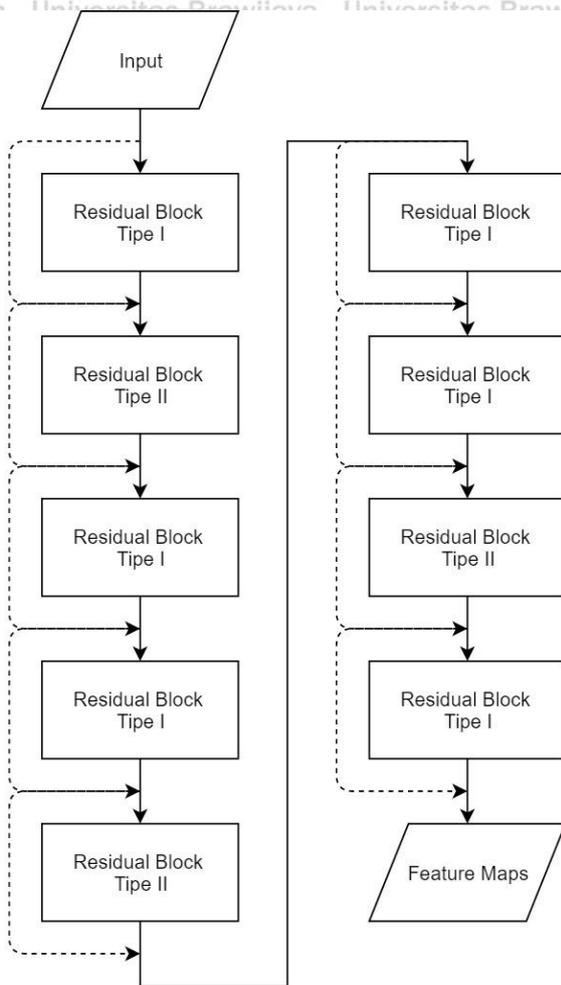
Residual block tipe II terdiri dari satu unit convolution layer dengan filter sebesar 3x3 sebanyak 32 buah dengan *stride* 1 dan diikuti dengan 1 unit max pooling layer dengan ukuran sebesar 2x2. Keluaran dari residual block ini akan ditambahkan dengan masukannya.



Gambar 4.5 Residual Block

Dari kedua tipe residual block yang telah dirancang akan dibangun sebuah feature extraction layer. Pada penelitian ini akan digunakan 6 unit residual block tipe I dan 3 unit residual block tipe II. Seluruh layer pada *feature extraction layer* menggunakan fungsi aktivasi ReLU.

Keluaran dari *feature extraction layer* ini adalah tensor 3 dimensi berukuran 5x5 sebanyak 32 buah yang biasanya disebut sebagai *feature maps*. *Feature maps* tersebut yang akan digunakan pada proses klasifikasi menggunakan *multilayer neural network*. Pada gambar 4.5 ditunjukkan arsitektur *feature extraction layer* secara lengkap.



Gambar 4.5 Arsitektur *Feature Extraction Layer*

4.5.2. Perancangan *Classification Layer*

Classification layer terdiri dari beberapa layer *neural network* dengan jumlah tertentu.

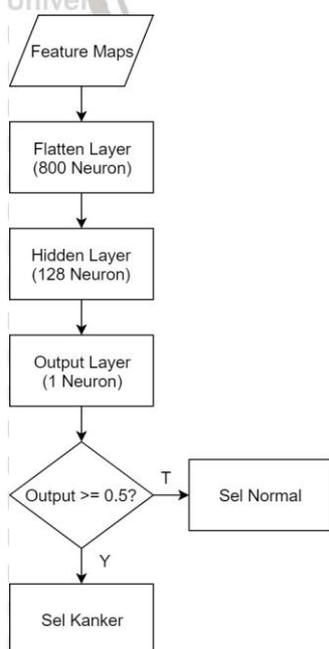
Pada penelitian ini digunakan *flatten layer* sebagai input, 1 buah *hidden layer* dan *output*

layer. *Flatten layer* digunakan untuk merubah *feature maps* keluaran dari *feature extraction layer* menjadi bentuk *vector* yang akan digunakan sebagai masukan dari *multilayer neural network*. *Feature maps* yang digunakan berukuran 6x6 sebanyak 32 buah. Setelah proses *flatten* maka dihasilkan 1152 buah *features*, sehingga jumlah neuron pada *flatten layer* adalah 1152 neuron. *Hidden layer* yang digunakan terdiri dari 256 neuron dan *output layer* terdiri dari 1 buah neuron. Fungsi aktivasi yang digunakan pada layer pertama adalah *ReLU* sedangkan fungsi aktivasi pada layer kedua atau *output layer* adalah *Sigmoid* yang memiliki rentang nilai antara 0 hingga 1 yang akan digunakan sebagai probabilitas dari hasil klasifikasi dimana nilai lebih kecil dari 0.5 merupakan negatif kanker dan nilai keluaran lebih besar dari 0.5 merupakan positif kanker.

Loss function yang digunakan pada penelitian ini adalah *Weighted Cross Entropy* yang digunakan untuk meningkatkan nilai *recall* atau *sensitivity* karena pada bidang medis jumlah *false negative* yang tinggi pada saat pengujian dianggap berbahaya karena sistem gagal mendeteksi sel kanker.

$$WCE = -y \log y \theta - (1 - y) \log(1 - y) \quad 4.2$$

Jika dilihat dari bentuk *Weighted Cross Entropy* pada persamaan 4.2. Untuk meningkatkan nilai *recall* atau *sensitivity* maka nilai θ yang seharusnya digunakan harus berada pada nilai lebih dari 1. Pada penelitian ini nilai yang digunakan akan dipilih secara bertingkat dari 1.1 hingga 1.5.

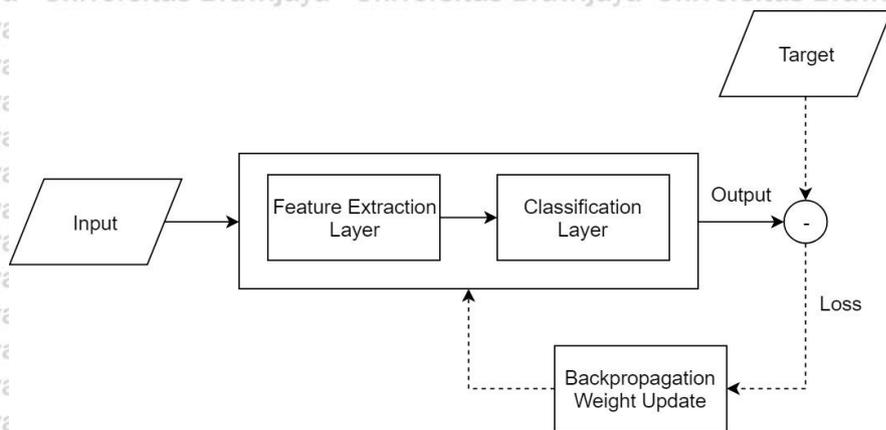


Gambar 4.6 Arsitektur *Classification Layer*

4.6. Training dan Testing

Proses training pada penelitian ini dilakukan secara *end-to-end* yang artinya input data dipropagasi maju dari *feature extraction layer* hingga *output layer* pada *classification layer*.

Hasil propagasi maju akan dibandingkan dengan target dengan menggunakan fungsi *weighted cross entropy* sehingga didapatkan nilai *loss/error* yang akan digunakan pada proses propagasi mundur atau *backpropagation* untuk melakukan *update* setiap bobot dan bias pada setiap layer pada *classification layer* dan *feature extraction layer*. Gambar 4.7 menunjukkan skema training yang digunakan pada penelitian ini.



Gambar 4.7 Training Neural Network

4.6.1. Forward Propagation

Seluruh data *training* pada proses *forward propagation* pada penelitian ini dibagi menjadi beberapa *batch*. Sehingga *loss* yang dihasilkan adalah *loss* rata-rata pada *batch* tersebut. Metode ini digunakan untuk mengatasi permasalahan nilai *loss* yang selalu berubah jika data dilakukan *forward propagation* adalah satu persatu sehingga kemungkinan untuk mencapai *loss* yang minimum menjadi sangat sulit. Berikut ini adalah proses *forward propagation* secara bertahap:

- i. Konvolusi antara input pada setiap *channel* dengan *convolution filter*.

$$A(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i - m, j - n) K(i - m, j - n) \quad (4.3)$$

- ii. *Pooling* menggunakan *Max Pooling*

$$B(i, j) = \text{MaxPool}(A(i, j)) \quad (4.4)$$

iii. Konvolusi *feature maps* pada layer selanjutnya dan tambahkan dengan keluaran dari *MaxPooling layer (Residu)*.

$$C(i, j) = (B * K)(i, j) = \sum_m \sum_n B(i - m, j - n)K(i - m, j - n) \quad (4.5)$$

$$D(i, j) = C(i, j) + B(i, j) \quad (4.6)$$

iv. Proses *flattening* dengan mengubah dari bentuk *tensor* dengan bentuk 5x5x32 menjadi *vector* secara berurutan sehingga didapatkan *features* sejumlah 800 yang akan digunakan sebagai masukan dari *classification layer (I_n)*.

v. Jumlahkan semua sinyal *input (I_n)* dan bobot (*w_{ij}*) yang masuk ke setiap *hidden unit (J_n)* pada *hidden layer (h_n)*, menggunakan Persamaan 4.7

$$h_{in_n} = b_{j_0} + \sum_{i=1}^n I_i w_{ij} \quad (4.7)$$

vi. Hitung keluaran setiap *hidden unit (J_n)* pada *hidden layer (h_n)* menggunakan fungsi aktivasi *ReLU*, menggunakan Persamaan 4.8

$$h_{out_n} = f(h_{in_n}) = \max(0, h_{in_n}) \quad (4.8)$$

vii. Jumlahkan semua sinyal yang masuk ke *output layer* pada setiap unit (*I_n*), dengan menggunakan Persamaan 4.9

$$o_{in} = b_{l_0} + \sum_{i=1}^n h_{out_n} w_{jl} \quad (4.9)$$

viii. Hitung keluaran dari setiap *output unit (I_n)* di *output layer*, menggunakan fungsi aktivasi *sigmoid*, menggunakan Persamaan 4.10

$$o_{out} = \frac{1}{1 + e^{-o_{in}}} \quad (4.10)$$

ix. Hitung *loss* atau *error* menggunakan fungsi *weighted cross entropy* menggunakan Persamaan 4.11.

$$Loss = -y \times \log(o_{out}) \times \theta - (1 - y) \times \log(1 - o_{out}) \quad (4.11)$$

4.6.2. Back Propagation

Proses selanjutnya adalah *back propagation* dimana nilai *loss* yang telah didapatkan pada *forward propagation* akan digunakan untuk melakukan *update* terhadap semua bobot dan bias pada *classification layer* dan *feature extraction layer*. *Gradient* pada masing-masing neuron, bobot dan bias akan dihitung dengan menggunakan aturan rantai atau *chain rule*. Berikut ini adalah proses *back propagation* secara bertahap:

- i. Hitung *gradient* pada *output layer* dengan melakukan penurunan partial persamaan 4.11 terhadap o_{out} , menggunakan Persamaan 4.12

$$\frac{\partial L}{\partial o_{out}} = \frac{\partial(-y \times \log(o_{out}) \times \theta - (1-y) \times \log(1-o_{out}))}{\partial o_{out}} = -\theta \left(\frac{y}{o_{out}} - \frac{1-y}{1-o_{out}} \right) \quad (4.12)$$

- ii. Hitung *gradient* pada *output layer* terhadap masukan dari *output layer* dengan melakukan penurunan partial persamaan 4.10 terhadap o_{in} menggunakan persamaan 4.13

$$\frac{\partial o_{out}}{\partial o_{in}} = \frac{\partial \left(\frac{1}{1-e^{-o_{in}}} \right)}{\partial o_{in}} = \frac{1}{1-e^{-o_{in}}} \times \left(1 - \frac{1}{1-e^{-o_{in}}} \right) \quad (4.13)$$

- iii. Hitung *gradient* pada bobot yang menuju *output layer* menggunakan *chain rule* pada persamaan 4.14 dengan melakukan substitusi dengan persamaan 4.12 dan 4.13

$$\frac{\partial L}{\partial w_{l_1}} = \frac{\partial L}{\partial o_{out}} \times \frac{\partial o_{out}}{\partial o_{in}} \times \frac{\partial o_{in}}{\partial w_{l_1}} \quad (4.14)$$

- iv. Ubah bobot yang menuju *output layer*, menggunakan persamaan 4.15 dimana parameter α adalah *learning rate*.

$$w'_{l_1} = w_{l_1} - \alpha \left(\frac{\partial L}{\partial w_{l_1}} \right) \quad (4.15)$$

Seluruh proses ini akan dilanjutkan hingga seluruh bobot dan bias pada *classification layer* dan *feature extraction layer* telah diupdate. Proses ini akan dilakukan berulang kali hingga pada saat *forward propagation* didapatkan nilai *loss* yang paling kecil yang menandakan bahwa konfigurasi bobot dan bias yang telah diupdate merupakan nilai yang optimal.

4.6.3. Mekanisme Training dan Testing

Sebelum proses *training* dilakukan, beberapa parameter yang dibutuhkan untuk proses *training* akan ditentukan terlebih dahulu. Parameter tersebut adalah sebagai berikut:

- *Batch Size*: 64
- *Learning Rate*: 0.001
- *Epoch*: 32
- *Optimizers*: *Stochastic Gradient Descent* (SGD)
- *Cross Entropy Weight*: 1, 1.1, 1.2, 1.25, 1.3, 1.4 dan 1.5
- *Color Spaces*: RGB, HSV, YUV, YCbCr dan CIELab

Training dilakukan selama 32 *epoch* dan pada setiap *epoch* setelah semua data *training* telah dipelajari akan dilakukan *testing* terhadap data *testing* yang telah disiapkan. Hasil *testing* ini akan digunakan sebagai tolak ukur performa dari model yang telah dilatih.

Konstanta *weighted cross entropy* yang diuji adalah 1, 1.1, 1.2, 1.25, 1.3, 1.4 dan 1.5. Nilai 1 mewakili nilai *loss* dari *standard cross entropy* dan akan digunakan sebagai acuan sedangkan nilai dibawah 1 tidak akan diuji karena merujuk dari persamaan dari *weighted cross entropy* yang telah dijelaskan pada bab sebelumnya, nilai konstanta dibawah 1 akan menurunkan *false positive rate* sedangkan pada penelitian ini yang perlu diturunkan adalah *false negative rate* karena jumlah citra negatif pada dataset lebih banyak.

Proses *training* dibagi menjadi dua tahap. Tahap pertama adalah dengan menguji setiap nilai *cross entropy weight* terlebih dahulu. Nilai *cross entropy weight* yang paling optimal akan digunakan pada training tahap kedua dengan mencoba seluruh *color spaces* yang akan diuji. Seluruh bobot dan bias akan disimpan kedalam sebuah file yang nantinya akan digunakan kembali pada tahap pengujian untuk mengukur performa dari model tersebut. Pengujian *color spaces* dilakukan untuk menemukan adanya perubahan performa sistem jika model dilatih dengan menggunakan beberapa *color space* yang berbeda

4.7. Pengujian

Tahapan perancangan sistem berhasil dilakukan maka tahapan selanjutnya yaitu pengujian sistem. Kemampuan sistem akan diuji dengan cara melakukan prediksi terhadap data *testing*. Ukuran yang digunakan pada proses pengujian adalah *precision*, *recall*, *specificity* dan F1-Score.

4.8. Perbandingan Hasil

Dataset yang digunakan pada penelitian ini merupakan *dataset* yang sama yang digunakan pada penelitian yang dilakukan oleh Angel Cruz-Roa pada tahun 2014 maka hasil pengujian pada penelitian ini dapat langsung dibandingkan dengan hasil penelitian terdahulu. Nilai dari *precision*, *recall*, *specificity* dan F1-Score merupakan ukuran yang digunakan untuk membandingkan hasil penelitian ini dengan hasil penelitian sebelumnya.

4.9. Analisis

Pada tahap ini dilakukan analisis terhadap sistem yang telah dibuat dengan cara membandingkan hasil perancangan dan pembuatan dengan parameter-parameter hasil

pengujian. Hasil analisis ini akan digunakan sebagai bahan dalam pengambilan kesimpulan dan saran.

4.10. Alat dan Bahan

Pada penelitian ini digunakan beberapa alat dan bahan yaitu satu unit *cloud server* dengan spesifikasi sebagai berikut :

- i. Intel Xeon 8 Cores, 8GB DDR4 RAM, Nvidia Tesla K80 64GB
- ii. Python Interpreter, library Numpy, Scikit-Image, OpenCV dan Google Tensorflow



BAB V

HASIL DAN PEMBAHASAN

5.1 Pengumpulan Data

Data yang digunakan pada penelitian ini adalah *Dataset* IDC dari Hospital of University of Pennsylvania and The Cancer Institute of New Jersey. Pada *dataset* tersebut terdapat 162 slide citra biopsi hasil scan pada 40x pembesaran. Pada tiap slide, area yang terdapat sel kanker telah ditandai oleh *pathologist* dan selain dari area yang telah diberi label adalah sel normal. Setiap slide pada *dataset* tersebut akan dibagi-bagi menjadi beberapa *patch* yang lebih kecil berukuran 50 x 50 *pixel* sehingga dihasilkan data citra biopsy sebanyak 277.524 yang terdiri dari 198.738 citra negatif IDC dan 78.786 citra positif IDC.



Gambar 5.1 Distribusi data pada *dataset*

Dari *dataset* tersebut secara acak diambil 80% untuk digunakan sebagai data *training* dan 20% digunakan sebagai data *testing*. Sehingga dihasilkan 222.019 data *training* yang terbagi menjadi 158.990 citra negatif dan 63.029 citra positif sedangkan untuk data *testing* terdapat 55.505 citra yang terdiri dari 39.748 citra negatif dan 15.757 citra positif. Seluruh data tersebut berada pada *directory* yang berbeda yang nantinya nama *directory* tersebut akan digunakan sebagai nama class pada saat *training* dan *testing*.

5.2 Pengujian

Training dilakukan di sebuah *Cloud Server* dengan menggunakan GPU. Bahasa pemrograman yang digunakan adalah Python dan *framework* yang digunakan adalah Tensorflow. Pada saat *training*, setelah setiap iterasi selesai, nilai *loss* dan *accuracy* untuk data *training* dan *testing* akan dicatat dan ditampilkan dalam bentuk grafik untuk memantau performa dari model *neural network* yang telah dirancang pada bab sebelumnya. Pengujian dilakukan untuk menemukan model terbaik dengan tahap sebagai berikut:

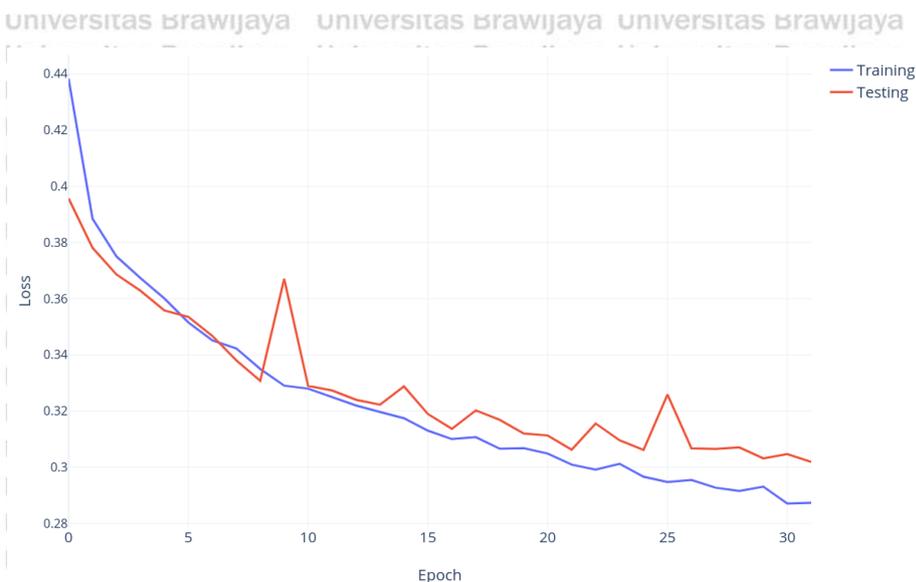
- I. *Training* dengan menggunakan nilai konstanta *weighted cross entropy* (θ) dengan nilai 1 sebagai acuan.
- II. *Training* dengan menggunakan nilai konstanta *weighted cross entropy* (θ) dengan nilai 1, 1.1, 1.2, 1.25, 1.3, 1.4 dan 1.5.
- III. Melakukan pengujian *precision*, *recall/sensitivity*, *specificity* dan F1-Score
- IV. Konstanta *weighted cross entropy* (θ) dengan nilai terbaik akan digunakan untuk melakukan *training* dengan menggunakan *color spaces* lain diantaranya adalah HSV, YUV, YCbCr, CIELab.

5.2.1 Pengujian Tahap I

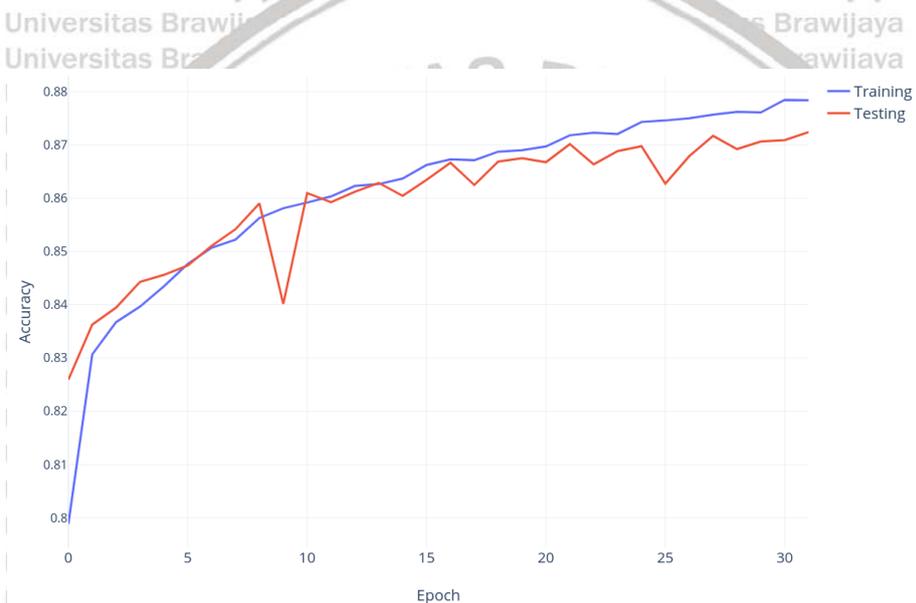
Pengujian Tahap I dilakukan untuk melatih model yang menggunakan 3 buah *residual block* seperti yang telah dibahas pada bab 4.5.1 dan mengukur performa dari sistem dengan menggunakan *standard cross entropy loss*. *Training* dilakukan dengan konfigurasi sebagai berikut:

- *Batch Size*: 64
- *Learning Rate*: 0.001
- *Epoch*: 32
- *Optimizers*: *Stochastic Gradient Descent (SGD)*
- *Cross Entropy Weight*: 1 (*Standard Cross Entropy*)
- *Color Space*: RGB

Ukuran yang dievaluasi pada saat *training* adalah *loss* dan akurasi. *Training* membutuhkan waktu sekitar 2 jam dan hasil *training* setelah 32 *epoch* adalah nilai *loss* dan akurasi pada data *training* secara berurutan adalah 0.2874 dan 87.84% sedangkan nilai *loss* dan akurasi pada data *testing* secara berurutan adalah 0.3019 dan 87.24%. Grafik *loss* dan akurasi terhadap jumlah *epoch* pada pelatihan tahap I dapat dilihat pada gambar 5.2 dan gambar 5.3.



Gambar 5.2 Grafik Loss Pelatihan pada Pengujian Tahap I



Gambar 5.3 Grafik Akurasi Pelatihan pada Pengujian Tahap I

5.2.2 Pengujian Tahap II

Pengujian Tahap II dilakukan untuk menemukan konstanta *weighted loss* yang paling optimal. Pengujian dimulai dari nilai konstanta *weighted loss* 1.1, 1.2, 1.25, 1.3, 1.4 dan 1.5 secara berurutan. *Training* dilakukan dengan konfigurasi sebagai berikut:

- *Batch Size*: 64
- *Learning Rate*: 0.001
- *Epoch*: 32
- *Optimizers*: *Stochastic Gradient Descent (SGD)*

- *Cross Entropy Weight*: 1.1, 1.2, 1.25, 1.3, 1.4 dan 1.5
- *Color Spaces*: RGB

Tabel 5.1 Hasil Pelatihan pada Pengujian Tahap II

No	Konstanta <i>Weighted Loss</i>	<i>Loss</i>		Akurasi	
		<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>
1	1	0.2874	0.3019	87.84%	87.24%
2	1.1	0.3080	0.3238	87.68%	87.10%
3	1.2	0.3389	0.3540	86.89%	86.16%
4	1.25	0.3286	0.3463	87.63%	86.92%
5	1.3	0.3333	0.3570	87.58%	86.50%
6	1.4	0.3655	0.3806	86.81%	86.40%
7	1.5	0.3639	0.3984	87.30%	86.79%
Angel Cruz, 2014		0.3363	0.3383	85.78%	85.74%

Jika merujuk dari pada persamaan *weighted cross entropy loss*, maka semakin tinggi konstanta *weighted loss*, maka nilai *loss* yang dihasilkan akan semakin tinggi. Namun hasil pada tabel 5.1 menunjukkan bahwa pada nilai konstanta antara 1.2 hingga 1.3 menghasilkan nilai *loss* cenderung turun. Hal ini disebabkan, model dapat meminimalisir *loss* dengan baik dengan menggunakan nilai konstanta tersebut tetapi nilai dari akurasi untuk semua nilai konstanta *weighted loss* relatif sama. Maka dari itu hasil pengukuran akurasi saja tidak cukup untuk mengukur performa dari sistem sehingga diperlukan pengujian yang lebih lengkap untuk menemukan konstanta *weighted loss* yang paling optimal.

5.2.3 Pengujian Tahap III

Pengujian Tahap III dilakukan untuk menemukan konstanta *weighted loss* yang paling optimal dengan cara mengukur nilai *precision*, *recall/sensitivity*, *specificity* dan F1-Score.

Seperti yang sudah dijelaskan pada bab sebelumnya bahwa untuk data medis, sistem yang optimal adalah sistem yang dapat memprediksi dataset dengan tingkat *false negative* yang paling rendah sehingga ukuran yang lebih diperhatikan adalah *recall/sensitivity*. Pada pengujian tahap III, model yang sudah dilatih akan digunakan untuk melakukan prediksi terhadap seluruh data pada *dataset* (*training* dan *testing*). Hasil klasifikasi seluruh data pada *dataset* ditunjukkan pada Tabel 5.2.

Tabel 5.2 Hasil Klasifikasi pada Pengujian Tahap III

No	Konstanta <i>Weighted Loss</i>	<i>True Positive</i> (TP)	<i>False Positive</i> (FP)	<i>True Negative</i> (TN)	<i>False Negative</i> (FN)
1	1	11245	3331	36417	4512
2	1.1	11424	2931	36187	4333
3	1.2	11842	3681	36067	3915
4	1.25	12606	4025	35723	3151
5	1.3	12373	4085	35663	3384
6	1.4	12031	3810	35938	3726
7	1.5	11667	3232	36516	4090
Angel Cruz, 2014		10824	3098	36650	4933

Berdasarkan hasil pengujian, model yang telah dilatih pada penelitian awal banyak melakukan kesalahan klasifikasi jika diberikan masukan citra positif. Model pada penelitian awal juga cenderung bias terhadap *class negative* karena tidak ada penanganan terhadap distribusi data yang tidak seimbang. Sedangkan hasil pengujian dengan menggunakan konstanta *weighted loss* sebesar 1.25 menghasilkan jumlah *false negative* paling rendah yaitu 15038.

Berdasarkan hasil pengujian tersebut akan dihitung nilai *Precision*, *Recall/Sensitivity*, *Specificity* dan F1-Score untuk menentukan konstanta *weighted loss* yang menghasilkan kombinasi nilai *recall* dan F1-Score yang paling tinggi dengan menggunakan persamaan sebagai berikut:

$$\bullet \text{ Precision} = \frac{\sum TP}{\sum TP + \sum FP}$$

$$\bullet \text{ Recall} = \frac{\sum TP}{\sum TP + \sum FN}$$

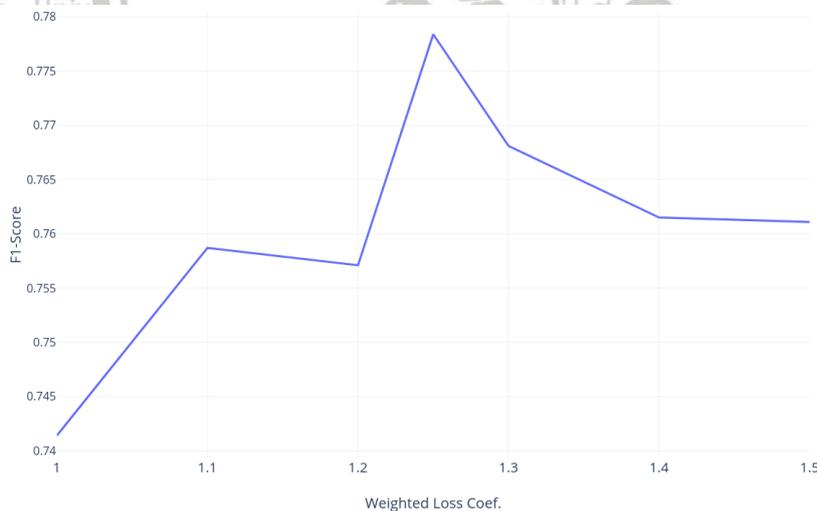
$$\bullet \text{ Specificity} = \frac{\sum TN}{\sum FP + \sum TN}$$

$$\bullet \text{ F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Tabel 5.3 Precision, Recall, Specificity dan F1-Score pada Pengujian Tahap III

No	Konstanta Weighted Loss	Precision	Recall	Specificity	F1-Score
1	1	0.7714	0.7136	0.9162	0.7414
2	1.1	0.7958	0.7250	0.9262	0.7587
3	1.2	0.7628	0.7515	0.9074	0.7571
4	1.25	0.7579	0.8000	0.8983	0.7784
5	1.3	0.7517	0.7852	0.8972	0.7681
6	1.4	0.7594	0.7635	0.9041	0.7615
7	1.5	0.7830	0.7404	0.9186	0.7611
Angel Cruz, 2014		0.7774	0.6869	0.9220	0.7294

Berdasarkan hasil yang didapat, konstanta *weighted loss* yang paling optimal adalah 1.25 yang menghasilkan nilai *Recall* dan F1-Score paling tinggi yaitu 0.8000 dan 0.7784 secara berurutan. Hasil pengukuran *Precision*, *Recall*, *Specificity* dan F1-Score secara lengkap dapat dilihat pada Tabel 5.3 sedangkan trend nilai F1-Score terhadap perubahan konstanta *weighted loss* disajikan pada gambar 5.4.

Gambar 5.4 Grafik trend perubahan nilai F1-Score terhadap θ

5.2.4 Pengujian Tahap IV

Pengujian Tahap IV dilakukan untuk menguji sistem dengan menggunakan *color spaces* lain yang sudah ditentukan sebelumnya yaitu HSV, YUV, YCbCr dan CIELab dengan menggunakan konstanta *weighted loss* sebesar 1.25. *Pre-processing* tetap dilakukan

namun dengan menggunakan *range* yang sesuai dari *color space* yang digunakan. *Training* dilakukan dengan konfigurasi sebagai berikut:

- *Batch Size*: 64
- *Learning Rate*: 0.001
- *Epoch*: 32
- *Optimizers*: *Stochastic Gradient Descent* (SGD)
- *Cross Entropy Weight*: 1.25
- *Color Space*: HSV, YUV, YCbCr dan CIELab

Hasil pelatihan menunjukkan bahwa tidak terdapat banyak perubahan jika digunakan *color spaces* yang lain kecuali jika menggunakan CIELab. Hal ini disebabkan karena *channel a* dan *b* pada CIELab berada pada rentang -128 hingga +127. Setelah normalisasi, nilai ini akan menjadi -1 hingga 1 sedangkan fungsi aktivasi yang digunakan pada seluruh *layer neural network* adalah ReLU yang memberikan keluaran 0 jika mendapat masukan dengan nilai ≤ 0 sehingga seluruh neuron tidak pernah aktif dan tidak selalu menghasilkan output 0. Hasil klasifikasi terhadap keseluruhan dataset disajikan pada Tabel 5.4 dimana jumlah *false negative* pada *color space* HSV hampir mendekati jumlah *false negative* pada *color space* RGB. Namun jumlah *false positive* pada *color space* HSV relative tinggi yaitu 25513.

Tabel 5.4 Hasil Klasifikasi pada Pengujian Tahap IV

No	Color Space	True Positive (TP)	False Positive (FP)	True Negative (TN)	False Negative (FN)
1	RGB	12606	4025	35723	3151
2	HSV	12576	5316	34432	3181
3	YUV	12194	3508	36240	3563
4	YCbCr	12332	3519	36229	3425
5	CIELab	0	0	39748	15757

Dengan menggunakan hasil pengujian tersebut akan dihitung nilai *Precision*, *Recall/Sensitivity*, *Specificity* dan *F1-Score* untuk menentukan *color space* yang paling optimal menggunakan persamaan yang sama pada pengujian tahap III.

Tabel 5.5 *Precision, Recall, Specificity dan F1-Score pada Pengujian Tahap IV*

No	Color Space	Precision	Recall	Specificity	F1-Score
1	RGB	0.7579	0.8000	0.8987	0.7784
2	HSV	0.7028	0.7981	0.8662	0.7474
3	YUV	0.7765	0.7738	0.9117	0.7752
4	YCbCr	0.7779	0.7826	0.9114	0.7803
5	CIELab	∞	0	1	∞
Angel Cruz, 2014		0.7774	0.6869	0.9220	0.7294

Berdasarkan hasil pengujian, color space yang optimal adalah YCbCr dengan mempertimbangkan nilai *recall* dan F1-Score yang paling baik. Dengan melatih sistem dengan menggunakan *color space* YCbCr didapatkan nilai *recall* sebesar 0.7826 dan F1-Score sebesar 0.7803.

Hal ini disebabkan karena pada dataset IDC tingkat brightness dari setiap citra berbeda, sehingga color space yang sesuai adalah yang memiliki komponen atau channel yang menyimpan informasi tentang brightness dari sebuah citra.

Dengan demikian dapat disimpulkan bahwa konstanta *weighted loss* yang paling optimal adalah 1.25 dengan menggunakan *color space* YCbCr dimana terjadi peningkatan sebesar 0.0957 atau 9.57% pada nilai *recall* dan 0.0509 pada nilai F1-Score

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Dari pengujian yang telah dilakukan pada penelitian ini, maka dapat disimpulkan sebagai berikut:

- a. *Residual Convolutional Neural Network* digunakan sebagai *feature extractor* dengan menggunakan 6 unit *residual block* tipe I dan 3 unit *residual block* tipe II. Pada akhir *feature extractor layer* terdapat *flatten layer* yang akan digunakan untuk mengubah *feature maps* menjadi bentuk *vector* sehingga dapat digunakan untuk melatih *classification layer* yang merupakan *multilayer neural network*.
- b. *Weighted Loss* yang digunakan pada penelitian ini adalah *Weighted Cross Entropy Loss* dengan menambahkan sebuah konstanta yang dicari dengan mencoba beberapa nilai secara bertingkat mulai dari 1, 1.1, 1.2, 1.25, 1.3, 1.4 dan 1.5. Fungsi *loss* tersebut digunakan untuk menaikkan nilai *error* pada saat *training* untuk menekan jumlah *false negative* pada saat pengujian. Konstanta yang menghasilkan performa terbaik adalah 1.25.
- c. Hasil penelitian didapatkan sistem bekerja secara optimal dengan nilai *precision*, *recall*, *specificity* dan F1-Score secara berurutan sebesar 0.7779, 0.7826, 0.9114 dan 0.7803 dengan menggunakan nilai konstanta *weighted cross entropy* sebesar 1.25 dan menggunakan *color space* YCbCr.

6.2. Saran

Penelitian yang dilakukan tentunya tidak terlepas dari kekurangan dan kelemahan. Oleh karena itu, untuk penelitian lebih lanjut, peneliti perlu memberikan saran sebagai berikut :

- a. Penggunaan fungsi aktivasi lain sehingga dapat digunakan untuk melatih sistem dengan menggunakan *color space* CIELab.
- b. Sistem dapat dilatih ulang dengan menggunakan 2 neuron pada *output layer* dan menggunakan fungsi aktivasi *softmax* dengan bentuk *weighted loss* yang sesuai.
- c. Penerapan proses Cross Validation untuk mengoptimalkan pembagian dataset antara *training* dan *testing*

- d. Sistem dapat dirubah secara total dengan menggunakan *Autoencoder* untuk melakukan *anomaly detection* mengingat jumlah data citra positif lebih sedikit dari citra negative sehingga citra positif bisa dianggap sebagai *anomaly* pada suatu sistem.



DAFTAR PUSTAKA

- American Cancer Society (2018). "About Breast Cancer." Diperoleh 17 Agustus 2018, dari <https://www.cancer.org/content/dam/CRC/PDF/Public/8577.00.pdf>
- BreastCancer.Org (2018). "Invasive Ductal Carcinoma (IDC)." Diperoleh 17 Agustus 2018, dari <https://www.breastcancer.org/symptoms/types/idc>
- Cruz-Roa A., Basavanhally A., Gonzalez F., Gilmore H., Feldman M., Ganesan S., Shih N., Tomaszewski J., Madabhushi A. (2014). "Automatic detection of invasive ductal carcinoma in whole slide images with Convolutional Neural Networks."
- Doyle, S., Agner, S., Madabhushi, A., Feldman, M., and Tomaszewski, J., (2008). "Automated grading of breast cancer histopathology using spectral clustering with textural and architectural image features,"
- Dundar, M. M., Badve, S., Bilgin, G., Raykar, V., Jain, R., Sertel, O., and Gurcan, M. N., (2011). "Computerized classification of intraductal breast lesions using histopathological images,"
- Esteva A., Kuprel B., Novoa R., Ko J., Swetter S., Blau H., Thrun S. (2017). "Dermatologist-level classification of skin cancer with deep neural network."
- Faustino-Rocha AI, Ginja M, Gama A, Ferreira R, Oliveira PA (2017). "Recent Studies & Advances in breast cancer."
- Gonzalez R., Woods R. (2008). Digital Image Processing : Third Edition
- He K., Zhang X., Ren S., Sun J. (2016). "Deep Residual Learning for Image Recognition".
- Janowczyk A., Madabhushi A. (2016). "Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases."
- Kementrian Kesehatan Republik Indonesia (2017). "Kementrian Kesehatan ajak masyarakat cegah dan kendalikan kanker." Diperoleh 17 Agustus 2018, dari <http://www.depkes.go.id/article/print/17020200002/kementrian-kesehatan-ajak-masyarakat-cegah-dan-kendalikan-kanker.html>
- Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012). "Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems."
- Naik, S., Doyle, S., Agner, S., Madabhushi, A., Feldman, M., and Tomaszewski, J., (2008). "Automated gland and nuclei segmentation for grading of prostate and breast cancer histopathology,"

Niwas, S. I., Palanisamy, P., Zhang, W., Mat Isa, N. A., and Chibbar, R., (2011). "Log-gabor wavelets based breast carcinoma classification using least square support vector machine."

Pan S., Yang Q., "A Survey on Transfer Learning". IEEE 2009

Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A.,

Khosla A., Bernstein M., Berg A., Fei-Fei L. (2015) "ImageNet Large Scale Visual Recognition Challenge"

Simonyan K., Zisserman A. (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition".

