p-ISSN : 2443-2210
e-ISSN : 2443-2229

Implementasi *DenseNet* Untuk Mengidentifikasi Kanker Kulit Melanoma

http://dx.doi.org/10.28932/jutisi.v6i3.2814

Jasman Pardede ✉ #1, Dwi Adi Lenggana Putra #2

#Jurusan Teknik Informatika, Institut Teknologi Nasional
Jl. PH. H. Mustofa No.23, Kota Bandung
[1]jasman@itenas.ac.id
[2]dwiputra47@gmail.com

Abstract — **Skin is a part of a human body that covers the entire body and protect the lower layer from direct sunlight and another microorganism. Because of that, skin cells are always changing and could be changed because of genetic mutation that causes skin cancer. In general, skin cancer is divided into three groups, namely : skin cancer Basal cell carcinoma, skin cancer Squamous cell carcinoma, and skin cancer Melanoma. Melanoma skin cancer is caused by abnormal growth in melanocyte cells. Several methods are proposed to predict Melanoma skin cancer using ResNet, LeNet, and Support Vector Machine. System performance is measured based on the value of accuracy, precision, recall, and f-measure. This experiment is conducted using a Melanoma skin cancer dataset that obtained the average value in terms of accuracy, precision, recall, and f-measure are 0.94, 0.95, 0.92, and 0.94 respectively. Based on that result, the proposed DenseNet121 performs better with 0.94 accuracy, compared with ResNet, LeNet, and Support Vector Machine method.**

Keywords— **Convolutional Neural Network; Image Classification; Melanoma Classification; DenseNet121.**

I. PENDAHULUAN

Kulit merupakan bagian tubuh manusia yang sangat luas untuk penyusun tubuh dan untuk menutupi seluruh permukaan pada tubuh manusia [1]. Kanker kulit adalah salah satu kanker yang paling umum didiagnosis di seluruh dunia, terutama pada populasi berkulit putih, insiden dan kematian terus meningkat selama dekade terakhir. Insiden kanker kulit dijumpai 5,9 – 7,8 % dari semua jenis kanker per tahun. Kanker kulit yang paling banyak di Indonesia adalah karsinoma sel basal (65,5%), diikuti karsinoma sel skuamosa (23%), melanoma maligna (7,9%) dan kanker kulit lainnya [2].

Kanker kulit disebabkan oleh pertumbuhan sel-sel kulit yang tidak terkontrol. Sel-sel yang ada pada penyusun kulit adalah sel skuamosa, sel basal, dan sel melanosit. Pertumbuhan abnormal pada sel melanosit menyebabkan terjadinya kanker kulit melanoma. Penyebab utama yang sering diderita pada kanker kulit adalah terkena paparan sinar ultraviolet (UV). Sinar UV berasal dari matahari, *tanning bad*, atau *sun lamp*. Sinar UV-lah yang merusak DNA sel penyusun kulit [3].

Image Classification merupakan proses yang dapat mengklasifikasikan gambar sesuai dengan kategori tertentu [4]. *Convolutional Neural Network* (CNN) dirancang untuk memproses suatu data yang ada dalam bentuk banyak *array*, contohnya gambar warna yang terdiri dari 2D array yang mengandung piksel dalam tiga macam warna yaitu *Red*, *Green*, dan *Blue*. Ada berbagai macan bentuk CNN adalah 1D untuk sinyal dan urutan biasa nya digunakan untuk bahasa, 2D untuk gambar atau suara; dan 3D untuk video atau gambar volumetrik [5].

Dense Convolutional Network (DenseNet), yang menghubungkan setiap lapisan/blok ke setiap lapisan/blok lainnya dengan cara umpan maju. Sedangkan jaringan *convolution* tradisional dengan L lapisan memiliki koneksi L - satu antara setiap lapisan dan lapisan berikutnya jaringan memiliki koneksi langsung L (L + 1) / 2. DenseNet memiliki beberapa kelebihan diantaranya masalah gradien yang dapat teratasi, memperkuat lapisan, lapisan yang berulang-ulang, dan mengurangi jumlah parameter [6].

Beberapa penelitian klasifikasi citra melanoma dan bukan melanoma yang menggunakan metode *Support Vector Machine*[7] dan CNN diantaranya ResNet50[8] dan LeNet5[9]. Pada penelitian ini mengimplementasikan arsitektur DenseNet-121 untuk membangun model klasifikasi kanker kulit melanoma dan bukan melanoma.

II. LANDASAN TEORI

A. *Deep Learning*

Deep Learning (pembelajaran mendalam) merupakan suatu bidang yang ada pada *machine learning* yang diperkenalkan oleh Dechter pada tahun 1986. *Deep Learning* menggunakan algoritma *backpropagation* menunjukkan suatu mesin harus mengubah parameter yang digunakan untuk menghitung nilai di setiap lapisan dari nilai di lapisan sebelumnya atau proses nya berjalan mundur pada setiap lapisannya [5]. *Deep Learning* adalah penggunaan

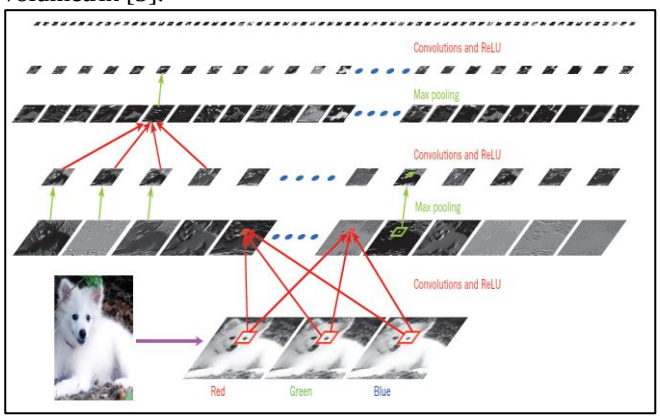
p-ISSN : 2443-2210
e-ISSN : 2443-2229

layer yang jumlahnya dapat mencapai ratusan sehingga dapat disebut dengan istilah "*deep*" [10].

Deep learning terbagi menjadi tiga kategori pendekatan yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. Salah satunya potensi dari *deep learning* adalah mengganti fitur buatan tangan dengan algoritma yang efisien untuk pembelajaran *hierarkis unsupervised* (fitur tanpa pengawasan) atau *semi-supervised feature learning* (semi-diawasi) dan *hierarchical feature extraction* (ekstraksi fitur). Penerapan deep learning penerapan deep learning telah digunakan dalam beberapa bidang seperti klasifikasi gambar, klasifikasi video, *object detection*, *object recognition*, *text-to-speech*, *natural language processing*, *robotic*, *text classification*, dan *singing synthesis* [11].

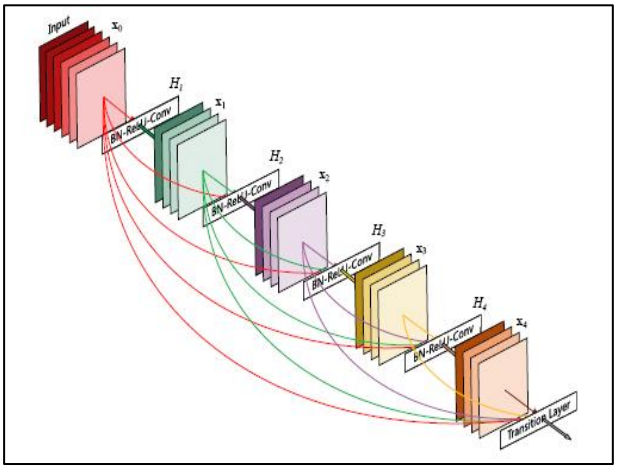
B. *DenseNet*

Convolutional Neural Network (CNN) dirancang untuk memproses suatu data yang ada dalam bentuk banyak array, contohnya gambar warna yang terdiri dari 2D array yang mengandung piksel dalam tiga macam warna yaitu *Red*, *Green*, dan *Blue* seperti yang diilustrasikan pada gambar 1. Ada berbagai macam bentuk CNN adalah 1D untuk sinyal dan urutan biasanya digunakan untuk bahasa, 2D untuk gambar atau suara; dan 3D untuk video atau gambar volumetrik [5].



Gambar 1. Arsitektur CNN Secara Umum [5]

Dense Convolutional Network (DenseNet), yang menghubungkan setiap lapisan/blok ke setiap lapisan/blok lainnya dengan cara umpan maju. Sedangkan jaringan konvolusional tradisional dengan L lapisan memiliki koneksi L - satu antara setiap lapisan dan lapisan berikutnya jaringan memiliki koneksi langsung L (L + 1) / 2. DenseNet memiliki beberapa keunggulan menarik: meringankan masalah gradien-gradien, memperkuat penyebaran fitur, mendorong penggunaan kembali fitur, dan secara substansial mengurangi jumlah parameter [6].



Gambar 2. Arsitektur DenseNet [6]

Pada gambar 2 untuk setiap komposisi lapisan menggunakan *batch normalization*, ReLU *activation* dan *convolution* dengan filter 3x3. Pada setiap blok ada masukan berupa matriks sesuai dengan pixel citra kemudian masuk ke proses *batch normalization* untuk mengurangi adanya *overfitting* pada saat proses *training*, ReLU *activation* digunakan untuk mengubah nilai x menjadi 0 jika nilai x tersebut bernilai negatif, sedangkan sebaliknya untuk nilai x tetap dipertahankan apabila nilai tidak kurang dari 0, *convolution* dengan filter 3x3 proses citra matriks yang sudah dilakukan operasi ReLU *activation* akan dikalikan dengan matriks *convolution* dengan filter 3x3 dan keluaran yang dihasilkan berupa nilai matriks yang sudah di proses sebelumnya.

Bottleneck : Bahwa *convolution* dengan filter 1×1 dapat disebut sebagai lapisan *bottleneck* sebelum setiap *convolution* dengan filter 3×3 untuk mengurangi jumlah peta fitur masukan, dengan meningkatkan efisiensi komputasi. Desain ini sangat efektif untuk DenseNet dan merujuk ke jaringan dengan lapisan *bottleneck*, yaitu, ke BN-ReLU-Conv (1 × 1) -BN-ReLU-Conv (3 × 3), sebagai DenseNet-B. Kecuali ditentukan, setiap konvolusi 1 × 1 mengurangi masukan ke peta fitur.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv; 3 × 3 conv] × 24	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48	[1 × 1 conv; 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv; 3 × 3 conv] × 16	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Gambar 3. Arsitektur DenseNet Dengan *Layer* [6]

Pada gambar 3 merupakan arsitektur DenseNet yang terdiri dari beberapa lapisan terdapat DenseNet-121, DenseNet-169, DenseNet-201, dan DenseNet-264. Di dalamnya terdapat 4 *dense block* dan 3 *transition layer* yang masing-masing memiliki proses *dense block* berisikan *batch normalization*, ReLU *activation* dan *convolution* dengan filter 3x3. Lapisan antara 2 blok yang berdekatan disebut sebagai *transition layer* dan perubahan ukuran fitur melalui *convolution* dan *pooling (average pooling)*.Tahap klasifikasi menggunakan *global average pooling* kemudian tahap selanjutnya softmax *activation.*

1) *Preprocessing*

Pada tahap awal proses preprocessing yaitu dilakukan *resize* untuk mengubah ukuran citra dengan memperkecil ukuran citra pada arah horizontal dan/atau vertikal menjadi ukuran 224x224 piksel. Hal ini bertujuan untuk menyeragamkan ukuran dari masing-masing citra yang digunakan selama proses pelatihan dan pengujian dikarenakan terjadinya perbedaan ukuran akan mengakibatkan proses pelatihan yang lama dan tingkat akurasi akan mempengaruhi.

Normalized data yaitu normalisasi piksel dangan antara dari 0 sampai 1. *Preprocessing* ini digunakan pada arsitektur LeNet [12]. Digunakan seperti pada persamaan (1).

rescale = img/255 ……...(1)

Dengan :

rescale= hasil dari preprocessing dengan rescale

img= nilai matriks dari citra

2) *Convolution*

Convolution layer adalah blok utama di dalam CNN yang terdiri dari beragam filter yang di inisial secara acak untuk melakukan operasi konvolusi yang berfungsi sebagai ekstraksi fitur untuk mempelajari representasi fitur dari suatu input gambar. Pada *convolution layer*, *neuron* tersusun menjadi *feature maps*. Setiap neuron pada feature map sebagai *receptive field*, terhubung pada neuron-neuron dari *convolution layer* sebelumnya melalui serangkaian bobot yang dilatih , biasa juga disebut dengan filter bank [5] yang diilustrasikan pada persamaan (2).

$h(x,y)=f(x,y)*g(x,y)$ ……...(2)

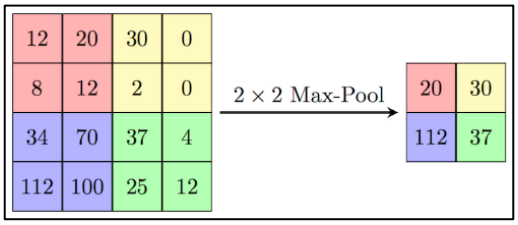
$h(x,y)$ = hasil dari proses *convolution*

$f(x,y)$= nilai matrik citra

$g(x,y)$= kernel *convolution*

3) *Pooling*

Pooling adalah untuk mencapai invarian spasial dengan mengurangi resolusi peta fitur. Setiap peta fitur yang dikumpulkan sesuai dengan satu peta fitur dari lapisan sebelumnya. Dalam sebagian besar CNN, metode subsampling yang digunakan adalah *max pooling*. *Max pooling* membagi output dari *convolution layer* menjadi beberapa grid kecil lalu mengambil nilai maksimal dari setiap grid untuk menyusun matriks citra yang telah direduksi. Sedangkan *average pooling* membagi output menjadi beberapa grid kecil untuk mengambil nilai rata-rata dari setiap grid yang ada pada citra untuk menyusun matriks yang sudah direduksi [13]. Pada gambar 4 diilustrasikan proses dari *max pooling.*



Gambar 4. Ilustrasi Max Pooling

4) *Batch Normalization*

Operasi *batch normalization* digunakan untuk mempercepat proses training dan meningkatkan *learning rates* pada saat pembuatan model. *Batch normalization* bekerja dengan menyamakan distribusi pada setiap nilai input yang selalu berubah dikarenakan perubahan parameter pada *layer* sebelumnya selama proses *training* [14].

Persamaan *Mini Batch Mean* :

$$\mu B = \frac{1}{m}\sum_{i=1}^{m} x_i \quad \text{.....(3)}$$

Persamaan *Mini Batch Variance* :

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu B)^2 \quad \text{.....(4)}$$

Persamaan normalisasi:

$$\hat{x}_i = \frac{x_i - \mu B}{\sqrt{\sigma_B^2 + \epsilon}} \quad \text{.....(5)}$$

Persamaan *scale and shift*:

$y_i = \gamma \hat{x}_i + \beta$ .....(6)

Dengan :

μB = merupakan nilai rata-rata dari *batch*

σB = standar deviasi dari *mini batch*

m = jumlah dataset pelatihan

$\hat{x}$ = *zero-centered* dan normalisasi input

ϵ = varible tambahan, biasanya bernilai 10−8 untuk menghindari pembagian dengan 0 jika σ = 0 dalam beberapa estiamsi

y_i = *output scaled* dan *shifted* dari operasi *batch normalization*

γ = *scale* dengan nilai = 1

β = *shift* dengn nilai = 0

5) *ReLU Activation*

Rectified Linear Units (ReLU) untuk fungsi aktivasi yang diperkenalkan oleh Geoffrey Hinton dan Vinod Nair dan digunakan pada *neural network*, digunakan untuk mengubah nilai x menjadi 0 jika nilai x tersebut bernilai negatif, sedangkan sebaliknya untuk nilai x tetap dipertahankan apabila nilai tidak kurang dari 0 [15]. Digunakan seperti pada persamaan (7).

$f(xi) = \max(0, xi)$ .....(7)

Dengan :

$f(xi)$ = nilai dari ReLU activation

xi = nilai matriks dari citra

6) *Softmax Activation*

Softmax diterapkan pada lapisan terakhir pada jaringan saraf. Softmax lebih dari itu umum digunakan daripada ReLU, sigmoid atau tanh. Ini digunakan untuk menghitung distribusi probabilitas dari vektor bilangan real. Fungsi Softmax menghasilkan output yang merupakan kisaran nilai antara 0 dan 1, dengan jumlah probabilitas sama dengan 1 [16]. Digunakan seperti pada persamaan (8).

$$f(xi) = \frac{\exp(xi)}{\sum_j exp^{(xi)}} \quad \text{.....(8)}$$

Dengan :

$f(xi)$ = hasil dari fungsi softmax *activation*

(xi) = kelas ke i (i=1,2). pada penelitian ini memakai 2 kelas

j = nilai dari vektor

7) *Cross Entropy*

Cross entropy merupakan fungsi untuk kerugian dan gradien digunakan dalam *multi-classification.* Untuk mengukur entropi relatif antara dua distribusi probabilitas pada data yang sama. *Cross entropy* fungsi kerugian yang digunakan ketika melatih model *neural network.* Cara kerjanya mengurangi log negatif dari dataset [17]. Digunakan seperti pada persamaan (9).

$$L_{cross-entropy} = -\sum_j y_i \log(\hat{y}) \quad \text{.....(9)}$$

Dengan :

$L_{cross-entropy}$=hasil dari *cross entropy*

$\hat{y}$=hasil dari softmax *activation*

y_i = nilai dari kelas, bernilai 1 jika kelas yang benar dan bernilai 0 jika kelas yang salah

j= kelas ke-j(1,2)

C. *Kanker Kulit Melanoma*

Kanker kulit merupakan pertumbuhan abnormal pada sel melanosit menyebabkan terjadinya kanker kulit melanoma. Penyebab utama yang sering diderita pada kanker kulit adalah terkena paparan sinar ultraviolet (UV) [3].

Gambar 5. Kanker Kulit Melanoma

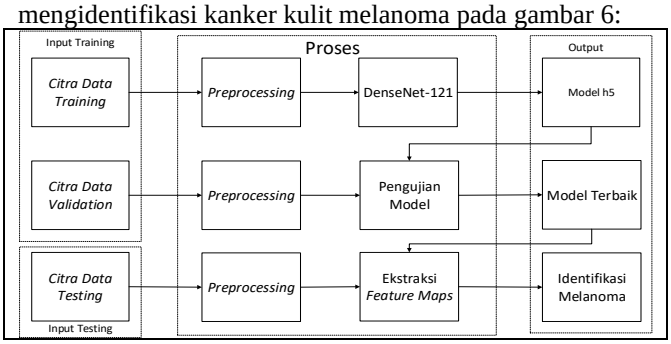
Melanoma adalah jenis kanker kulit yang jarang dan sangat berbahaya. Kondisi ini dimulai dari kulit manusia dan bisa menyebar ke organ lain dalam tubuh. Kemunculan tahi lalat baru atau perubahan pada tahi lalat yang sudah ada biasanya menjadi pertanda umum atau gejala melanoma. Melanoma memiliki bentuk yang tidak beraturan dan lebih dari satu warna. Tahi lalat yang terserang melanoma bisa terasa gatal dan bisa mengalami pendarahan, selain itu,

ukurannya juga bisa melebihi tahi lalat normal dengan ilustrasi pada gambar 5.

III. PERANCANGAN

A. *Block Diagram*

Dalam penelitian ini ada beberapa tahapan untuk mengidentifikasi kanker kulit melanoma penelitian ini merupakan penelitian yang di dasarkan berdasarkan beberapa penelitian yang sudah ada dan berikut ini pada gambar merupakan tahapan secara umum untuk mengidentifikasi kanker kulit melanoma pada gambar 6:



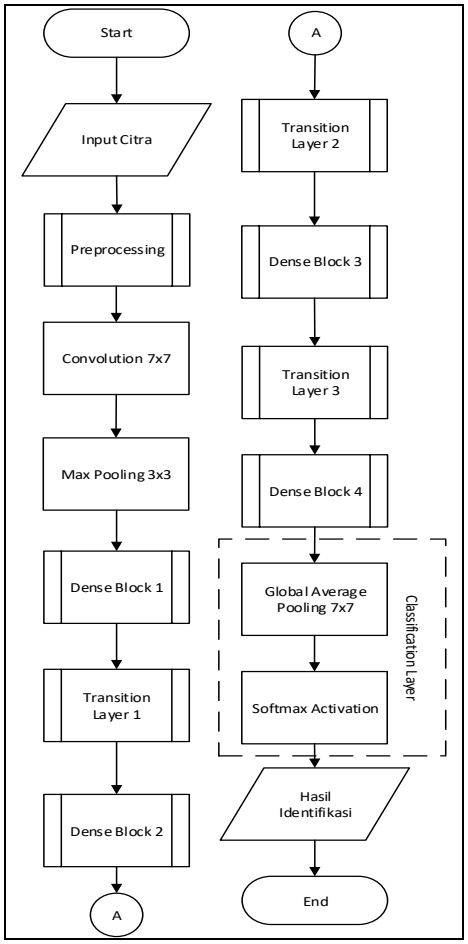
Gambar 6. Block Diagram

Pada persamaan sub-bab 2 dengan nomor A, B, dan C, data kanker kulit melanoma untuk melakukan *training* seperti data *training*, data *validation*, dan data *testing*. Pada *input* training dilakukan tahapan *training* yang ada pada data *training* dengan data *validation*. Pada tahap proses nya melakukan *preprocessing* seperti *resize* menjadi 224x224 piksel dan *normalized* data sesudah di *resize* data tersebut. Pada citra data *training* menggunakan proses dengan model DenseNet yang terdapat proses *feature extraction* dan *classification.* Pada model DenseNet terdapat operasi umum *batch normalization*, ReLU *activation*, dan *convolution*. Model DenseNet dengan 121*layer* memiliki proses *dense block* 1, *transition layer* 1, *dense block* 2, *transition layer* 2, *dense block* 3, *transition layer* 3, *dense block* 4 dan *classification layer* yang menghasilkan *output* model dengan format h5. Pada data *validation* proses dilakukan pengujian model dengan data *training* yang akan menghasilkan *output* dengan model yang terbaik dalam sisi bobot.

Pada citra data *testing* masuk ke dalam proses *preprocessing* dengan *resize* dengan menjadi 224x224 piksel dan *normalized* data kemudian melakukan proses ekstraksi *feature maps* dengan mengambil nilai bobot terbaik dari data *training* sehingga akan menghasilkan *output* identifikasi dari kanker kulit melanoma yang memiliki label 0 untuk melanoma dan 1 untuk *not* melanoma.

B. *Diagram Alir Sistem*

Pada gambar 7 pada persamaan sub-bab 2 dengan nomor A dan B yang diilustrasikan *flowchart* keseluruhan sistem dalam menentukan identifikasi kanker kulit melanoma. Sistem dimulai dengan input citra atau gambar kemudian melakukan *preprocessing* kemudian sesudah di *preprocessing* dilakukan nya operasi *convolution* 7x7 dengan *strides* 2 , operasi dilanjutkan dengan *max pooling* 3x3 dengan *strides* 2 dan akan mendapatkan suatu nilai matriks yang diproses ke operasi *dense block* 1, *transition layer* 1, *dense block* 2, *transition layer* 2, *dense block* 3, *transition layer* 3, *dense block* 4 dan kemudian nilai matriks yang sudah diproses oleh operasi sebelum nya di proses kembali pada tahapan *classification layer* dengan operasi *global average pooling* 7x7 dan kemudian softmax *activation* adalah operasi terakhir dari model DenseNet121 ini untuk menentukan nilai mendekati antara 0 dan 1 untuk 0 identifikasi "Melanoma" dan untuk nilai 1 identifikasi "NotMelanoma".

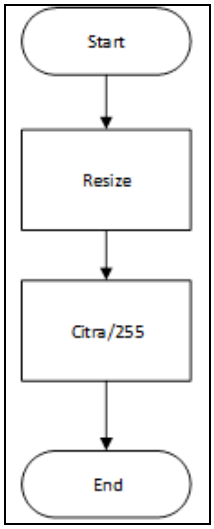


Gambar 7. Diagram Alir Keseluruhan Sistem

1) Preprocessing

Untuk tahapan sub proses preprocessing yang ada pada persamaan sub-bab 2 dengan nomor 1 seperti gambar 8 yang diilustrasikan sesudah citra dimasukan ke tahap *preprocessing* yang berupa ada tahap *resize* 224x224 piksel sesudah di *resize* menjadi 224x224 piksel citra tersebut di *normalized* data citra/255 sehingga nilai matriks pada citra tersebut menjadi 0 sampai 1.

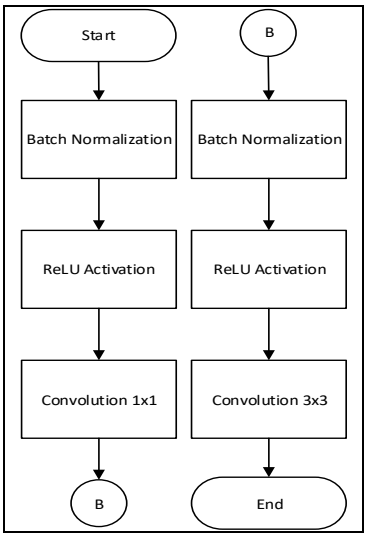
p-ISSN : 2443-2210
e-ISSN : 2443-2229



Gambar 8. Diagram Alir Preprocessing

2) DenseBlock

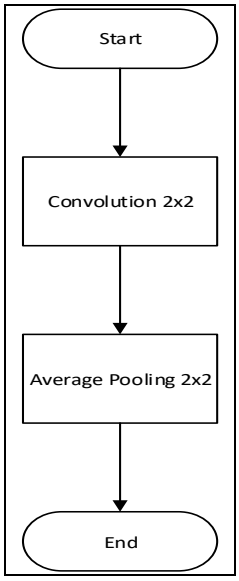
Pada gambar 9 untuk tahap sub proses *dense block* yang ada pada persamaan sub-bab 2 dengan nomor 2, 4, dan 5 terdapat operasi *batch normalization*, ReLU *activation*, *convolution* 1x1 operasi ini dinamakan *bottleneck* pada didalam proses *dense block* tersebut kemudian operasi selanjutnya *batch normalization*, ReLU *activation*, *convolution* 3x3 dimana nilai matriks tersebut saling keterhubungan satu sama lain sehingga nilai matriks tersebut di digabungkan. Pada proses *dense block* terdapat operasi yang dikalikan sehingga mencapai *layer* 121, *dense block* 1 terdapat operasi *convolution* yang dikalikan 6, *dense block* 2 terdapat operasi *convolution* yang dikalikan 12, *dense block* 3 terdapat operasi *convolution* yang dikalikan 24, dan *dense block* 4 terdapat operasi *convolution* yang dikalikan 16.



Gambar 9. Diagram Alir Dense Block

3) Transition Layer

Pada gambar 10 yang diilustrasikan sub proses *transition layer* yang ada pada persamaan sub-bab 2 dengan nomor 2 dan 3 terdapat operasi *convolution* 2x2 dan *average pooling* 1x1 dengan *strides* 2, di proses ini *transition layer* terdapat 3 proses yaitu *transition layer* 1, *transition layer* 2, dan *transition layer* 3. Proses ini dilakukan sesudah proses *dense block* agar proses transisi antara *dense block* dengan *transition layer*.



Gambar 10. Diagram Alir Transition Layer

C. *Penggunaan Dataset*

Pada *dataset* citra-citra kanker kulit melanoma yang didapat dari Alexander Scarlat[18]. Data citra yang digunakan pada proses *training*, *validation*, dan *testing* merupakan citra yang berbeda yang ada pada persamaan sub-bab 2 dengan nomor C. *Dataset* yang digunakan bisa dilihat pada Tabel I.

TABEL I
PENGGUNAAN DATASET

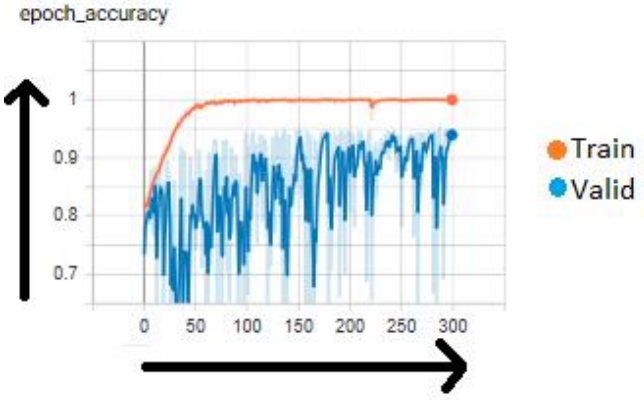
No	Kelas	Training	Validation	Testing
1	Melanoma	5341	1781	1781
2	Not Melanoma	5341	1781	1780

IV. IMPLEMENTASI DAN PENGUJIAN

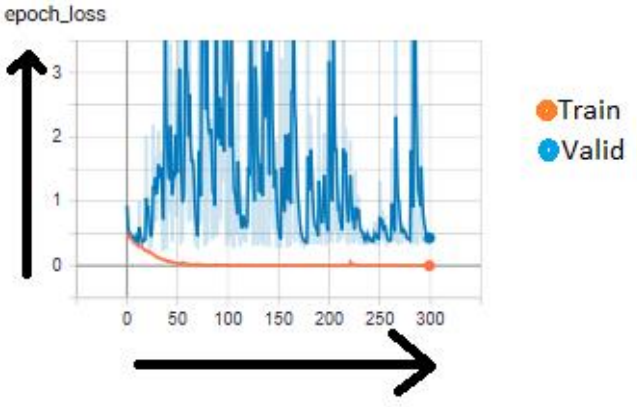
Pengujian sistem dalam mengidentifikasi kanker kulit melanoma yang terdapat pada citra dengan menggunakan metode *convolution neural network* dengan arsitektur DenseNet121. Implementasi ini dilakukan dengan dua mesin. Penelitian ini melakukan proses *training* menggunakan google *collaboratory* dengan spesifikasi GPU Tesla P100 16GB, RAM 27GB, dan *storage* 68GB. Proses *testing* menggunakan processor Intel Core i5-7200U @2.5GHz, RAM 8GB, storage 1TB. Tensorflow dan keras sebagai penunjang dari penelitian ini.

1) Training

Penelitian ini untuk mengklasifikasi gambar, menggunakan citra data *training* dan data *validation*. Penelitian ini pada proses *training* dibangun menggunakan google *collaboratory* dengan 300 *epoch*, *optimizer stochastic gradient descent*, *learning rates* 0,1 dan 32 *batch size* dalam *training* dengan menggunakan proses arsitektur DenseNet121 yang mendapatkan model *.h5. Memanggil citra *training* dan *validation* kemudian citra tersebut dibagi dengan 32 dari *batch size* sehingga 1 *epoch* melewati proses 334 citra kemudian diselesaikan sampai 300 *epoch* dimana dengan arsitektur DenseNet121 mendapatkan nilai terendah *loss* pada *training* 0.0001 dan untuk nilai terendah *validation* 0.2491. Untuk nilai tertinggi dari *training* accuracy 1 dan untuk nilai tertinggi dari *validation* 0.9455. *Training* dilakukan selama 12 jam. Model yang sudah didapatkan diintegrasikan dengan *website* menggunakan *framework* flaks untuk dilakukan pengujian terhadap citra kanker kulit melanoma untuk mengetahui tingkat akurasi yang didapatkan dari model tersebut. Pada gambar 11 untuk hasil *loss* dan gambar 12 hasil *accuracy* seperti hasil dari *training* dari arsitektur DenseNet121.



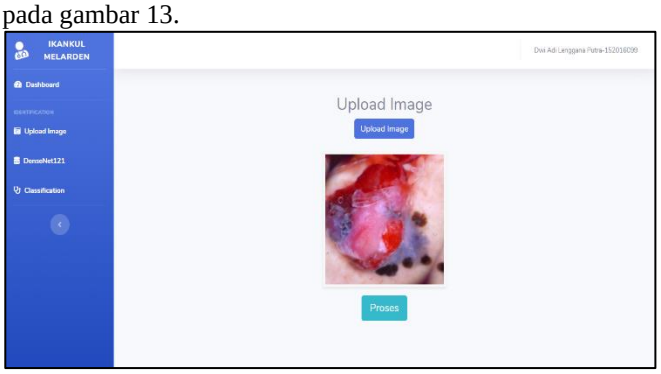
Gambar 11. Hasil Nilai Loss



Gambar 12. Hasil Nilai Accuracy

2) Proses Upload Image

Pada tahapan ini pengguna dapat memasukan dataset sebagai citra data uji. Tahapan tersebut dapat dilakukan dengan melakukan menekan tombol "upload image", maka sistem akan menampilkan folder dan pengguna dapat memilih data uji yang tersedia. Selanjutnya menekan tombol "proses" melakukan proses dari arsitektur DenseNet121. Berikut adalah tampilan dari proses masukan dataset seperti pada gambar 13.



Gambar 13. Implementasi Upload Image

3) Proses Prediksi

Pada tahapan ini pengguna dapat melihat hasil identifkasi kanker kulit melanoma dari "upload image" dan proses arsitektur DenseNet121. Berikut adalah tampilan dari hasil identifikasi kanker kulit melanoma seperti pada gambar 14.

Gambar 14. Implementasi Hasil Identifikasi Kanker Kulit Melanoma

4) Pengujian Sistem

Pengujian pencarian citra dilakukan dengan menggunakan citra kanker kulit melanoma, pengujian dilakukan dengan menguji 3561 citra. Dengan mengukur kinerja arsitektur DenseNet-121 menggunakan *confusion matrix* dimana akan mengukur kinerja sistem secara keseluruhan seperti *accuracy*, *precision*, *recall*, dan *f measure* dapat dihitung dengan persamaan berikut[19]:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad \text{.....(9)}$$

$$Precision = \frac{TP}{TP+FP} \quad \text{.....(10)}$$

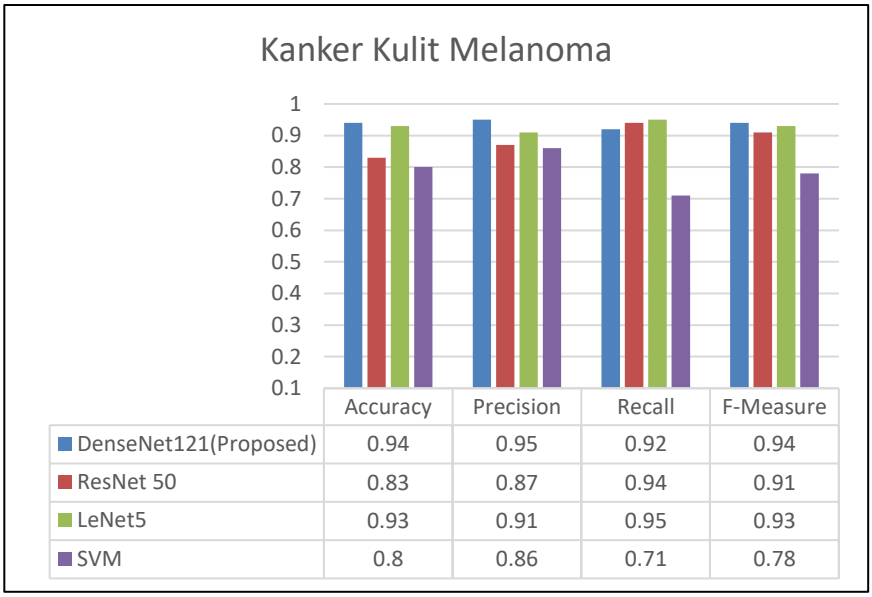
p-ISSN : 2443-2210
e-ISSN : 2443-2229

$$Recall = \frac{TP}{TP+FN} \quad \text{.....(11)}$$

$$F\ Measure = 2 * \frac{(Precision*Recall)}{(Precision+Recall)} \quad \text{.....(12)}$$

Dengan :
TP = *True Positive*
FP = *False Postive*
FN = *False Negative*
TN = *True Negative*

Pada gambar 15 perbandingan kinerja antara metode lain dengan hasil pengujian dari penelitian pada dataset kanker kulit melanoma. Dengan hasil dari kinerja sistem dengan hasil dari data yang sudah diuji dengan 3561 citra diantaranya 1781 citra label melanoma dan 1780 citra label bukan melanoma. Pada hasil pengujian dalam mengidentifikasi label melanoma dengan arsitektur DenseNet121 mendapatkan nilai *accuracy* 0,94, *precision* 0,95, *recall* 0,92, dan *f-measure* 0,94. Untuk arsitektur ResNet50 dari perbedaan jumlah dataset, *preprocessing*, *optimizers* dan *learning rate* mendapatkan nilai *accuracy* 0,83, *precision* 0,87, *recall* 0,94 dan *f-measure* 0,91. Kemudian untuk arsitektur LeNet5 dari perbedaan jumlah dataset, *preprocessing*, *optimizers* dan *learning rate* mendapatkan nilai *accuracy* 0,93, *precision* 0,91, *recall* 0,95 dan *f-measure* 0,93. Dan yang terakhir untuk metode *Support Vector Machine* dari perbedaan jumlah dataset dan *preprocessing* mendapatkan nilai *accuracy* 0,8, *precision* 0,86, *recall* 0,71 dan *f-measure* 0,78.



	Accuracy	Precision	Recall	F-Measure
DenseNet121(Proposed)	0.94	0.95	0.92	0.94
ResNet 50	0.83	0.87	0.94	0.91
LeNet5	0.93	0.91	0.95	0.93
SVM	0.8	0.86	0.71	0.78

Gambar 15. Perbandingan kinerja accuracy, precision, recall, dan F-Measure antara DenseNet121 dengan metode lain

V. KESIMPULAN

Pada penelitian ini telah mengimplementasikan CNN dengan arsitektur DenseNet121 untuk mengidentifikasi kanker kulit melanoma. Metode yang diusulkan mampu melakukan klasifikasi kanker kulit melanoma dengan nilai rata-rata *accuracy*, *precision*, *recall*, dan *F-Measure* masing-masing adalah 0,94, 0,95, 0,92 dan 0,94.Dengan *preprocessing* yang digunakan pada aristektur DenseNet121 mempengaruhi dari tingkat akurasi. Berdasarkan hasil eksperimen diperoleh bahwa sistem yang diusulkan memiliki kinerja terbaik jika dibandingkan dengan metode lain yang pernah diusulkan.

DAFTAR PUSTAKA

[1] F. Nuraeni, Y. H. Agustin, dan E. N. Yusup, "Aplikasi Pakar Untuk Diagnosa Penyakit Kulit Menggunakan Metode Forward Chaining Di Al Arif Skin Care Kabupaten Ciamis," *Semin. Nas. Teknol. Inf. Dam Multimed.*, hal. 6–7, 2016.

[2] S. Wilvestra, S. Lestari, dan E. Asri, "Studi Retrospektif Kanker Kulit di Poliklinik Ilmu Kesehatan Kulit dan Kelamin RS Dr. M Djamil Padang Periode Tahun 2015-2017," *J. Kesehat. Andalas*, vol. 7, no. Supplement 3, hal. 47–49, 2018.

[3] A. C. Society, "Melanoma Skin Cancer," *Ski. Cancer*, 2016.

[4] T. Bow dan T. Bow, "Case study: Image clustering," *Heterog. Comput. with OpenCL 2.0 Third Ed.*, hal. 213–228, 2015.

[5] Y. Lecun, Y. Bengio, dan G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, hal. 436–444, 2015.

[6] G. Huang, Z. Liu, L. Van Der Maaten, dan K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, hal. 2261–2269, 2017.

[7] S. Mustafa, A. B. Dauda, dan M. Dauda, "Image processing and SVM classification for melanoma detection," *Proc. IEEE Int. Conf. Comput. Netw. Informatics, ICCNI 2017*, vol. 2017-Janua, no. February 2018, hal. 1–5, 2017.

[8] A. Budhiman, S. Suyanto, dan A. Arifianto, "Melanoma Cancer Classification Using ResNet with Data Augmentation," in *2019 2nd International Seminar on Research of Information*

Technology and Intelligent Systems, ISRITI 2019, 2019, hal. 17–20.

[9] R. Refianti, A. B. Mutiara, dan R. P. Priyandini, "Classification of melanoma skin cancer using convolutional neural network," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 3, hal. 409–417, 2019.

[10] Mathworks, "What Is Deep Learning?," 2018. [Daring]. Tersedia pada: https://www.mathworks.com/discovery/deep-learning.html.

[11] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, hal. 85–117, 2015.

[12] K. S. Choi, J. S. Shin, J. J. Lee, Y. S. Kim, S. B. Kim, dan C. W. Kim, "In vitro trans-differentiation of rat mesenchymal cells into insulin-producing cells by rat pancreatic extract," *Biochem. Biophys. Res. Commun.*, vol. 330, no. 4, hal. 1299–1305, 2005.

[13] D. Scherer, A. Müller, dan S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6354 LNCS, no. PART 3, hal. 92–101, 2010.

[14] S. Ioffe dan C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *32nd International Conference on Machine Learning, ICML 2015*, 2015, vol. 1, hal. 448–456.

[15] A. F. Agarap.(2018) Deep Learning using Rectified Linear Units (ReLU). [Online] Tersedia : https://arxiv.org/abs/1803.08375

[16] C. Nwankpa, W. Ijomah, A. Gachagan, dan S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," *Mach. Learn. (cs.LG); Comput. Vis. Pattern Recognit.*, vol. 1, no. 1, hal. 1–20, 2018.

[17] K. P. Murphy, "Machine learning : a probabilistic perspective," in *Adaptive computation and machine learning series*, London: The MIT Press, 2012, hal. 1–1104.

[18] A. Scarlat, "Melanoma Dataset," 2018. [Daring]. Tersedia pada: https://www.kaggle.com/drscarlat/melanoma.

[19] J. Pardede dan M. G. Husada, "Comparison Of VSM, GVSM, And LSI In Information Retrieval For Indonesian Text," *J. Teknol.*, vol. 78, no. 5–6, hal. 51–56, Mei 2016.