

# Klasifikasi Tulisan Tangan Pada Resep Obat Menggunakan Convolutional Neural Network

Mohammad Farid Naufal<sup>1</sup>, Joko Siswanto<sup>2\*</sup>, Muhammad Ghifari Kusuma Wicaksono<sup>3</sup>

<sup>1,2,3</sup>Program Studi Teknik Informatika, Universitas Surabaya

E-mail: <sup>1</sup>faridnaufal@staff.ubaya.ac.id, <sup>2</sup>joko\_siswanto@staff.ubaya.ac.id,

<sup>3</sup>masghiff@gmail.com

\*penulis korespondensi

## Abstrak

Obat merupakan bahan kimia yang dapat merepresentasikan tubuh secara fisiologi dan psikologi ketika dikonsumsi. Obat sebagai alat bantu untuk menyembuhkan dari berbagai macam penyakit. Dengan berkembangnya zaman dan bertambahnya wawasan, menyebabkan bertambah juga jenis obat-obatan yang memiliki banyak manfaat dan kegunaannya. Penelitian ini bertujuan untuk mendeteksi nama obat dalam resep dokter menggunakan *Convolutional Neural Network* (CNN) dengan transfer learning. Metode transfer learning merupakan metode yang populer dalam mengklasifikasi gambar digital yang berguna untuk mempercepat proses klasifikasi. Penelitian ini membandingkan lima arsitektur transfer learning yaitu VGG16, Resnet, Xception, LeNet, dan GoogleNet. Penelitian ini juga menggunakan grayscale, resizing, dan median filter pada tahap *preprocessing*. *Preprocessing* digunakan untuk meningkatkan kualitas citra pada citra resep obat dan menghilangkan *noise* pada citra. ResNet-50 merupakan arsitektur terbaik untuk mengklasifikasi nama obat. Pada percobaan menggunakan ResNet-50, mendapatkan *F1 score* tertinggi yaitu sebesar 97,56% dan waktu training rata-rata 0,25 detik setiap epoch. Dapat disimpulkan Resnet merupakan arsitektur terbaik untuk mengklasifikasikan nama obat dalam citra resep dokter serta dapat mendeteksi nama obat secara akurat.

Kata kunci: Resep, Obat, CNN, Preprocessing, Transfer Learning

## Abstract

*Medicines are chemical substances that can represent the body both physiologically and psychologically when consumed. Medicines are used as tools to cure various diseases. With the development of time and increasing knowledge, many types of drugs have been discovered that have many benefits and uses. This study aims to detect the names of drugs in doctor's prescriptions using Convolutional Neural Network (CNN) with transfer learning. Transfer learning is a popular method in classifying digital images that is useful for accelerating the classification process. This study compares five transfer learning architectures: VGG16, ResNet, Xception, LeNet, and GoogleNet. This study also uses grayscale, resizing, and median filtering in the preprocessing stage. Preprocessing is used to improve the quality of drug prescription images and eliminate noise in the images. ResNet is the best architecture for classifying drug names. In the experiment using ResNet, the highest F1 score was obtained at 97.56% and the average training time was 0.25 seconds per epoch. It can be concluded that ResNet is the best architecture for classifying drug names in doctor's prescription images and can accurately detect drug names.*

*Keywords: Prescription, Medicine, CNN, Preprocessing, Transfer Learning.*

## 1. PENDAHULUAN

Kemajuan teknologi saat ini dan masuknya era revolusi industri 4.0 membuat banyak perubahan yang sangat signifikan dalam hal teknologi, termasuk di bidang kesehatan. Dengan penggunaan aplikasi pada bidang kesehatan juga dapat mengurangi *medical error* yang terjadi. *Medical error* adalah kejadian yang merugikan pasien akibat pemakaian obat selama dalam

penanganan tenaga kesehatan yang sebetulnya dapat dicegah. Kesalahan dalam peresepan (*prescribing*) dan penggunaan obat (*administering*) merupakan dua hal penyebab terbanyak dalam *medical error*. Namun penulisan yang sulit dibaca merupakan faktor yang bisa meningkatkan kesalahan terapi.

Ackhar et al [1] melakukan pengenalan tulisan tangan dokter dengan melakukan kombinasi antara tulisan tangan normal dengan tulisan tangan dokter pada 66 jenis obat yang berbeda namun tidak disebutkan jenis obatnya. Dengan menggunakan arsitektur VGG16 dan mempertimbangkan waktu pelatihan dan jumlah input, dapat menghasilkan data akurasi sebesar 95%. Batasan penelitian ini adalah masih menggunakan bantuan tulisan tangan biasa untuk melatih model klasifikasi. Selain itu penelitian ini juga belum diterapkan pada sebuah aplikasi secara real-time untuk mengklasifikasikan jenis obat pada resep tulisan tangan dokter.

Arrahman et al [2] melakukan klasifikasi nama obat dari tulisan tangan dokter. Metode yang digunakan pada penelitian sebelumnya yaitu menggunakan metode GLCM dan Backpropagation Neural Network. Dataset yang digunakan pada penelitian ini didapatkan dari proses pengambilan nama obat pada gerakan pena di layar sentuh dan pengambilan gambar nama obat dari scanner atau kamera digital. Selanjutnya dataset yang didapatkan dilanjutkan dengan tahap preprocessing, metode yang digunakan pada tahap preprocessing yaitu, *cropping image*, *resize*, *grayscale*, dan *median filter*. Setelah dilakukan proses preprocessing dilanjutkan dengan proses ekstraksi fitur GLCM dan Backpropagation Neural Network. Hasil akurasi yang didapatkan pada penelitian dengan menggunakan metode Backpropagation Neural Network mendapatkan akurasi sebesar 70% dalam melakukan klasifikasi nama obat tulisan tangan dokter.

Jain et al [3] menggunakan model CNN-Bi-LSTM dengan CTC untuk klasifikasi karakter obat dari resep dokter. Model terdiri dari tiga komponen, yaitu lapisan konvolusi untuk ekstraksi fitur, jaringan Bi-LSTM untuk membuat prediksi untuk setiap frame dan decoding akhir untuk mengartikan setiap karakter dalam urutan yang diakui. Linear layer ditambahkan setelah biLSTM layer untuk menghitung probabilitas akhir. Ditambahkan juga korpus manual yang berisi istilah medis yang umum digunakan dalam resep untuk meningkatkan akurasi output prediksi. Salah satu kelemahan dari penelitian ini adalah kemungkinan akurasi model dapat terpengaruh oleh variasi gaya tulisan tangan, yang dapat menyebabkan prediksi yang tidak akurat. Selain itu, penggunaan singkatan Latin dan istilah medis yang tidak umum diketahui dapat mengakibatkan kesalahan dalam keluaran prediksi, meskipun ada bias yang diberikan pada istilah medis. Efektivitas model juga dapat terbatas oleh ukuran dan kelengkapan korpus yang dibangun secara manual yang digunakan dalam studi. Selain itu, penelitian ini tidak menyebutkan pengujian atau evaluasi kinerja model pada dataset yang lebih besar atau di lingkungan dunia nyata, yang dapat membatasi generalisasi perhitungan performa.

Hassan et al [4] memperkenalkan sebuah metode untuk mengenali tulisan tangan nama obat dan dosisnya. Sistem ini menggunakan teknik pra-pemrosesan, Convolutional Neural Network, dan Optical Character Recognition untuk mengenali nama obat dan dosisnya dengan akurasi mencapai 70%. Sistem ini membantu meminimalkan kesalahan penulisan dan meningkatkan kepercayaan diri farmasis dalam memberikan obat kepada pasien. Batasan dari penelitian ini adalah hanya memiliki akurasi pelatihan sebesar 73% dan akurasi pengujian sebesar 50%. Untuk meningkatkan akurasi, akan digunakan dataset yang lebih besar dengan lebih banyak nama obat yang ditulis tangan dan akan diterapkan teknik klasifikasi yang lebih banyak. Selain itu, perbandingan hasil OCR dengan dataset yang berisi semua nama obat akan dilakukan untuk mengidentifikasi obat yang diresepkan.

Palani et al [5] Paper ini membahas penggunaan Long Short-Term Memory (LSTM) berbasis Convolutional Neural Network (CNN) untuk mengembangkan sebuah model yang dapat membedakan tulisan dokter dalam resep medis. Dengan menggunakan Deep Convolution Recurrent Neural Network (RNN) untuk melatih model pengawasan ini, gambar input di-segmentasi menggunakan segmentasi Otsu dan diolah untuk mengidentifikasi huruf dan kata, serta mengkategorikan mereka menjadi 56 karakter yang berbeda yang telah ditentukan. Meskipun model yang dibuat menggunakan LSTM based Convolutional Neural Network (CNN)

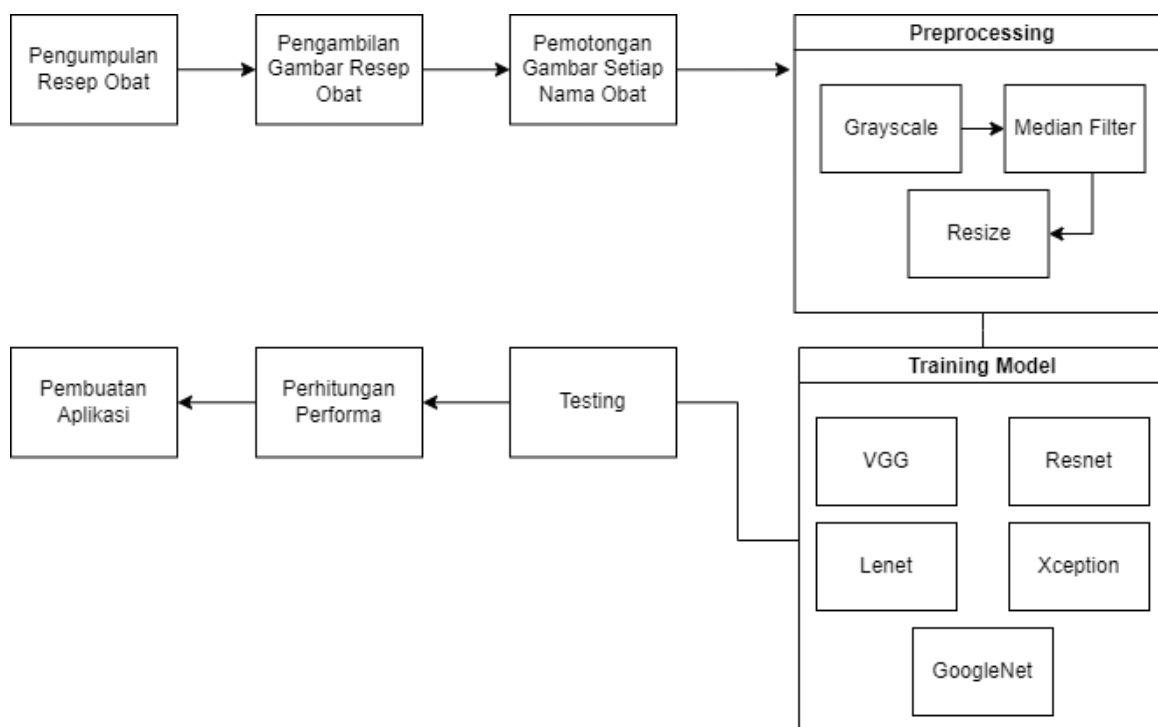
dapat membedakan tulisan tangan dokter pada resep medis, akurasi pengujian hanya mencapai 50%. Selain itu, untuk meningkatkan akurasi, dataset yang lebih besar dapat digunakan.

Kelima penelitian yang disebutkan memfokuskan pada pengenalan tulisan tangan dokter dalam resep medis, dengan tujuan untuk melakukan klasifikasi nama obat dan dosisnya. Beberapa penelitian menggunakan teknik klasifikasi seperti GLCM dan Backpropagation Neural Network, sedangkan yang lain menggunakan model Convolutional Neural Network (CNN) yang dikombinasikan dengan metode seperti Bi-LSTM dan Optical Character Recognition (OCR). Beberapa penelitian tersebut juga memperhitungkan faktor seperti waktu pelatihan dan jumlah input untuk meningkatkan akurasi model. Namun, batasan dari penelitian tersebut adalah masih menggunakan bantuan tulisan tangan biasa untuk melatih model klasifikasi, tidak diterapkannya model secara real-time pada sebuah aplikasi untuk mengklasifikasikan jenis obat pada resep tulisan tangan dokter, serta masih terbatasnya dataset yang digunakan dan ukuran korpus yang dibangun secara manual yang dapat membatasi generalisasi performa model. Meskipun CNN secara luas digunakan dalam pengenalan citra, penggunaannya dalam klasifikasi citra resep obat mungkin memiliki beberapa keterbatasan. CNN dirancang khusus untuk mengekstraksi fitur-fitur spasial dari citra, seperti pola visual, tekstur, dan bentuk. Namun, dalam citra resep obat, fitur-fitur ini mungkin tidak selalu menjadi fokus utama untuk klasifikasi. Selain itu, jika dataset citra resep obat terbatas, CNN yang dilatih dari awal mengalami masalah overfitting dan tidak mampu menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya. LSTM dan Bi-LSTM: LSTM dan Bi-LSTM dirancang untuk memodelkan urutan data dengan mempertahankan informasi konteks yang relevan. Namun, dalam kasus klasifikasi citra resep obat, tidak ada urutan yang jelas dalam citra tersebut. Informasi yang relevan untuk klasifikasi terkandung dalam elemen visual dan struktural di dalam citra, bukan dalam urutan yang membutuhkan pemrosesan temporal seperti yang dilakukan oleh LSTM dan Bi-LSTM. Oleh karena itu, menggunakan model berbasis urutan seperti LSTM dan Bi-LSTM tidak memberikan keuntungan yang signifikan dalam klasifikasi citra resep obat.

Oleh karena itu, dibutuhkan model CNN yang dikombinasikan dengan Transfer Learning untuk dapat mengatasi limitasi tersebut dan meningkatkan akurasi pengenalan tulisan tangan dokter pada resep medis. Resep obat yang diklasifikasikan dalam penelitian ini diambil melalui input gambar yang telah di-crop pada bagian nama obatnya. Pemilihan area khusus pada gambar ini dilakukan karena nama obat seringkali menjadi informasi yang paling relevan dan penting dalam konteks klasifikasi resep obat. Dengan melakukan crop pada bagian ini, perhatian kita difokuskan pada karakteristik dan fitur penting yang berkaitan dengan identifikasi dan klasifikasi obat. Penelitian ini mengkombinasikan CNN dengan Transfer learning untuk mengatasi masalah variasi gaya tulisan tangan, penggunaan istilah medis yang tidak umum, dan batasan pada jumlah data yang digunakan. Penggunaan Transfer learning dapat membantu meningkatkan akurasi model dengan memanfaatkan model yang sudah terlatih. Aplikasi secara real-time pada sistem klasifikasi obat pada resep tulisan tangan dokter juga perlu dikembangkan untuk memperluas penggunaan teknologi ini di lingkungan klinis. Sebelum dilakukan proses klasifikasi nama obat menggunakan metode CNN dan transfer learning dilakukan proses *preprocessing*. Proses *preprocessing* yang akan dilakukan menggunakan metode *Grayscale*, *Median Filter*, dan *Resize* gambar. Proses *preprocessing* dibutuhkan untuk mempersiapkan gambar sebelum dilakukan proses *training* dan *testing* dan dapat meningkatkan hasil akurasi dari klasifikasi gambar.

## 2. METODE PENELITIAN

Metodologi penelitian yang digunakan pada penelitian ini meliputi pengumpulan resep obat, pengambilan gambar resep obat, pemotongan gambar resep obat menjadi sebuah hanya nama obat, *preprocessing*, *training* model, *testing*, perhitungan performa, dan pembuatan aplikasi pembacaan resep obat. Pada gambar 1 menunjukkan tahap metodologi penelitian yang dilakukan dalam penelitian ini:



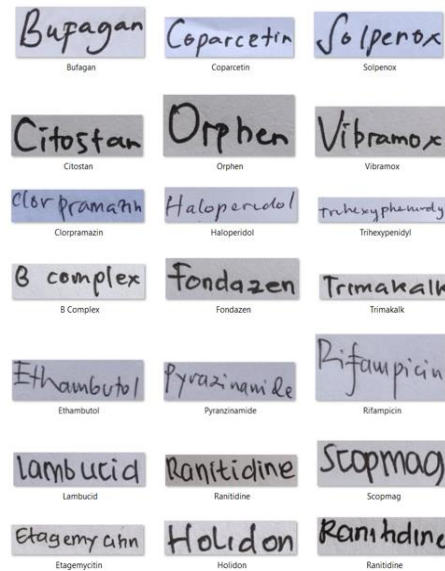
Gambar 1. Tahap Proses Metodologi Penelitian

## 2.1 Pengumpulan Dataset

Jumlah total dataset digunakan sebanyak 2100 gambar nama obat, yang terbagi jadi 20 jenis macam nama obat. Setiap gambar yang digunakan merupakan gambar original yang diambil pada resep obat. Proses pertama yaitu melakukan pengumpulan resep obat, kemudian dilakukan pengambilan gambar resep obat. Proses pengambilan gambar resep obat dilakukan dengan menggunakan lensa kamera pada *handphone* Iphone 8 Plus. Setelah dilakukan pengambilan gambar resep obat, dilanjutkan dengan proses pemotongan gambar resep obat. Proses pemotongan gambar resep obat atau *cropping* dilakukan secara manual untuk hanya mendapatkan nama obat. Penelitian ini menggunakan dataset citra resep obat yang diambil secara mandiri dan bersifat privat. Pengambilan resep obat ini didapatkan dari resep obat yang ditulis tangan oleh dokter umum dan dokter spesialis. Tidak ada spesifikasi jarak pengambilan gambar yang spesifik. Namun, dataset ini belum pernah digunakan dalam penelitian lain sebelumnya, sehingga penelitian ini memiliki keunikan dalam menggunakan dataset yang belum dieksplorasi sebelumnya. Penggunaan dataset private ini memberikan kesempatan untuk mengevaluasi performa model CNN dalam konteks khusus citra resep obat yang belum terdokumentasikan sebelumnya. Tabel 1 menunjukkan rincian dataset terdiri jumlah gambar pada setiap nama obat, dan resep obat berdasarkan dari sub keahlian dokter. Dari jumlah keseluruhan dataset atau gambar nama obat yang telah dikumpulkan, dataset dibagi menjadi 85% untuk *training model* dan 15 % untuk *testing*. Pada Gambar 2 merupakan contoh dataset nama obat yang digunakan pada sistem ini.

**Tabel 1.** Jumlah Gambar Setiap Nama Obat

Nama Obat	Jumlah Gambar	Sub Keahlian Dokter
B Complex	100	Dokter Kandungan
Fondazen	100	
Trimakalk	100	
Clorpramazin	100	Dokter Jiwa
Haloperidol	100	
Thyhexyphenidyl	100	
Bufagan	100	Dokter Anak
Coparctin	100	
Solpenox	100	
Lambucid	100	Dokter Penyakit Dalam KGEH
Scopmag	100	
Ranitidine	100	
Rifampicin	100	Dokter Paru-Paru
Ethambutol	100	
Pyranzinamid	100	
Citostan	100	Dokter Bedah
Orphen	100	
Vibramox	100	
Etagemyctin	100	Dokter Umum
Holidon	100	
Ranitidine	100	

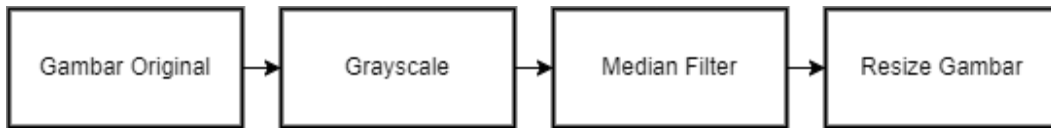


Gambar 2. Contoh gambar nama obat pada resep dokter

## 2.2 Preprocessing

Tahapan *preprocessing* dilakukan pada saat sebelum melakukan *training* dan *testing*. *Preprocessing* ini bertujuan untuk membantu sistem training dan testing, agar dapat memproses klasifikasi lebih mudah dan mendapatkan nilai akurasi yang tinggi. Beberapa fungsi yang digunakan pada tahap *preprocessing* yaitu dapat menghilangkan *noise* pada gambar sehingga gambar yang dihasilkan menjadi lebih *smooth*. Metode yang digunakan pada tahap *preprocessing* yaitu: Cropping, Grayscaleing [6], Median Filtering [7], dan Resizing [8]. Pada gambar 3 merupakan tahapan metode yang digunakan pada *preprocessing*. Dalam konteks penggunaan aplikasi, proses cropping dilakukan oleh pengguna saat menggunakan aplikasi tersebut. Setelah pengguna mengambil gambar resep obat menggunakan kamera perangkat mereka, aplikasi memberikan kemampuan untuk memotong (crop) gambar dengan cara yang mudah dan intuitif. Median filtering cenderung mempertahankan detail dalam citra dengan lebih baik daripada mean filtering. Mean filtering cenderung menyebabkan blurring atau kehilangan detail, terutama pada citra dengan detail kecil atau struktur yang penting seperti karakter dalam resep obat. Dalam

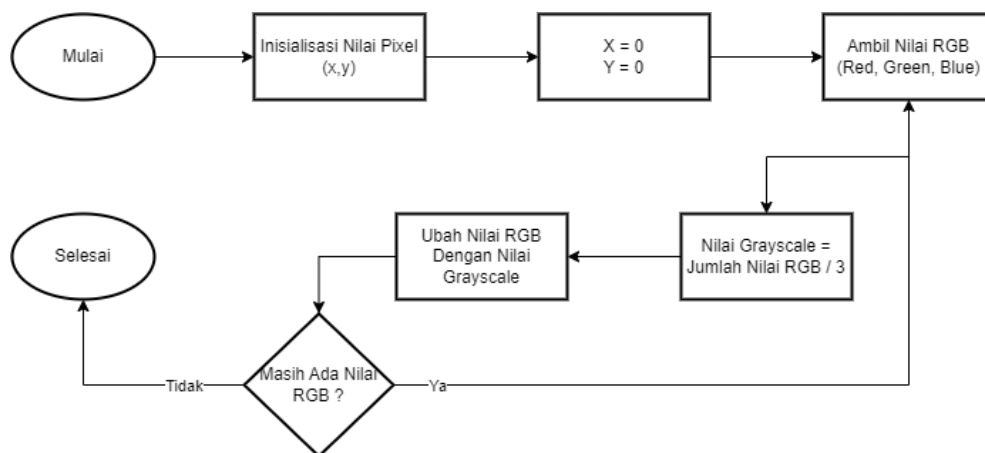
klasifikasi citra resep obat, menjaga kejelasan dan integritas karakter sangat penting untuk memastikan akurasi dalam proses klasifikasi.



Gambar 3. Tahap Preprocessing

1) *Grayscale*

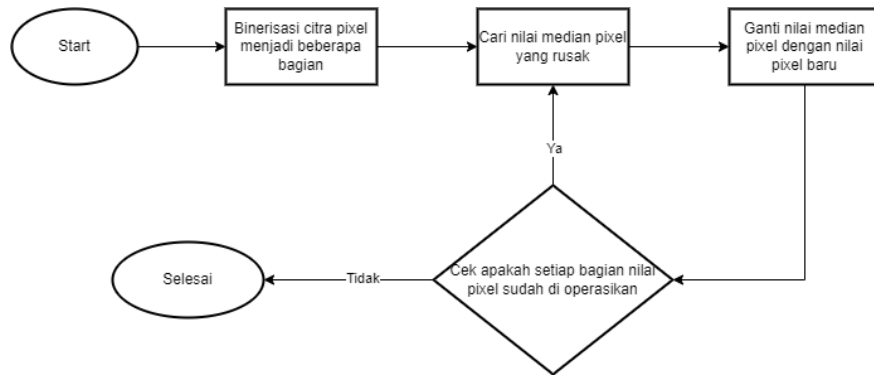
Proses *grayscale* digunakan untuk perubahan warna pada gambar yang pada saat gambar masih original atau gambar awal setelah dilakukan pemotongan secara manual. Pada proses *grayscale* ini dilakukan perubahan warna menjadi ke abu-abuan. Pada gambar 4 flowchart proses perubahan gambar dari original menjadi *grayscale*.



Gambar 4. Flowchart Proses Grayscale

2) *Median Filter*

Proses Median Filter ini digunakan untuk melakukan penghilangan *noise* atau bercak hitam putih yang terdapat pada gambar setelah dilakukan proses *grayscale*. Gambar yang dihasilkan dari proses median filter ini gambar menjadi lebih smooth atau halus. Fungsi dari median filter ini juga dapat meningkatkan performa yang dihasilkan dari proses training dan testing. Erkan et al [9] menggunakan metode median filter traditional dan frekuensi. Pengujian dilakukan dengan menggunakan. Metode median filter frekuensi dapat mengecualikan nilai pixel yang memiliki *noise* dari nilai warna abu-abuan baru, dan hanya fokus pada keunikan nilai abu-abu. Pada gambar 5 merupakan flowchart dari proses median filter. Gambar 6 menunjukkan proses sebelum dan sesudah proses median filtering.



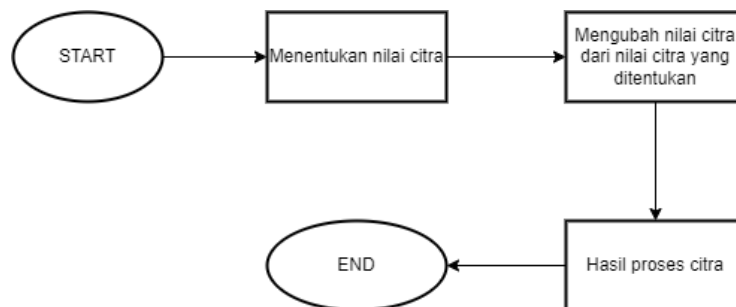
Gambar 5. Flowchart Proses Median Filter



Gambar 6. Hasil proses sebelum (a) dan sesudah Median filter (b)

### 3) *Resize*

Resize gambar bertujuan untuk melakukan perubahan ukuran pixel pada gambar yang semula memiliki ukuran pixel yang berbeda-beda pada setiap gambar. Proses resize gambar ini akan dilakukan untuk menyamakan ukuran pixel di setiap gambar yaitu, 250x150. Agar dapat memudahkan dalam proses training dan testing. Input gambar dalam ukuran seragam 250x150 digunakan untuk memastikan konsistensi dan keseragaman data yang masuk ke dalam model klasifikasi. Pada gambar 7 merupakan flowchart dari proses resize gambar. Gambar 8 menunjukkan hasil sebelum dan sesudah resizing. Proses menentukan nilai citra yang dimaksud adalah proses penginputan nilai hasil resize gambar.



Gambar 7. Flowchart Proses Resize

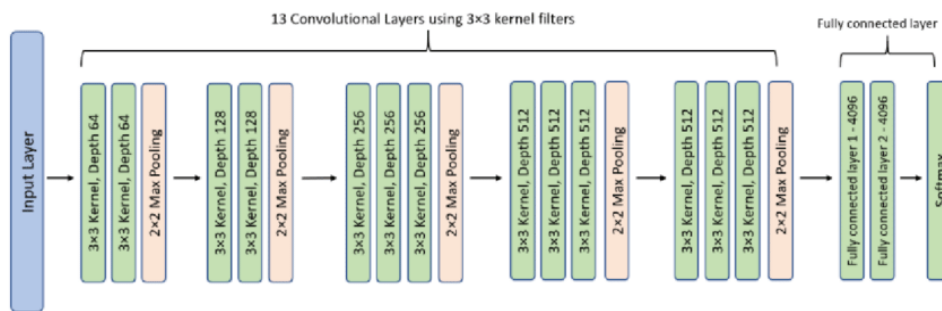


Gambar 8. Hasil proses sebelum (a) dan sesudah *resizing* (b)

### 2.3 Training Model

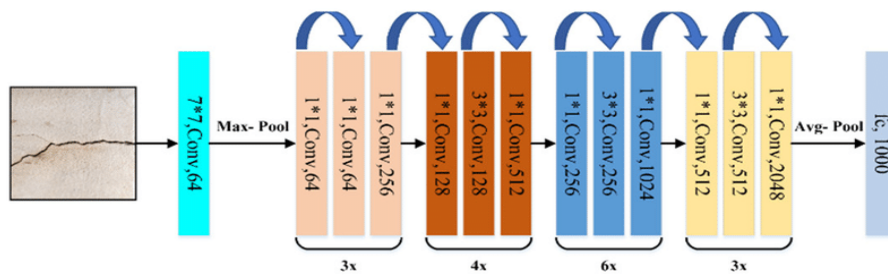
Pada proses training digunakan untuk melatih model arsitektur CNN yang digunakan, agar dapat mengenali nama obat. Proses ini menggunakan 85% dari dataset yang dimiliki, serta untuk nilai batch size yang digunakan sebesar 32. Hasil dari proses training yang dilakukan pada setiap model arsitektur disimpan dengan format ekstensi HDF5. Hasil penyimpanan dari proses training berisi data berupa model, bobot training, dan informasi yang digunakan untuk proses testing. Model CNN yang dibandingkan dalam klasifikasi citra resep obat, seperti VGG16, ResNet, Xception, LeNet, dan GoogleNet, dipilih karena mereka memiliki arsitektur yang telah efektif dalam pengenalan citra pada berbagai tugas dan dataset.

Model VGG-16 terdiri dari 16 layer (13 layer convolutional, 2 fully connected layer, dan 1 layer classifier) yang memiliki ukuran filter sebesar 3x3. Model VGG juga mendapatkan penghargaan pada ImageNet Large Scale Visual Recognition Challenge (ILSVRC) pada tahun 2014. Perbedaan utama pada tiap lapisan konvolusi yaitu terletak di jumlah filter di masing-masing lapisan [10]. Gambar 9 menunjukkan arsitektur dari VGG-16



Gambar 9. Arsitektur VGG-16 [10]

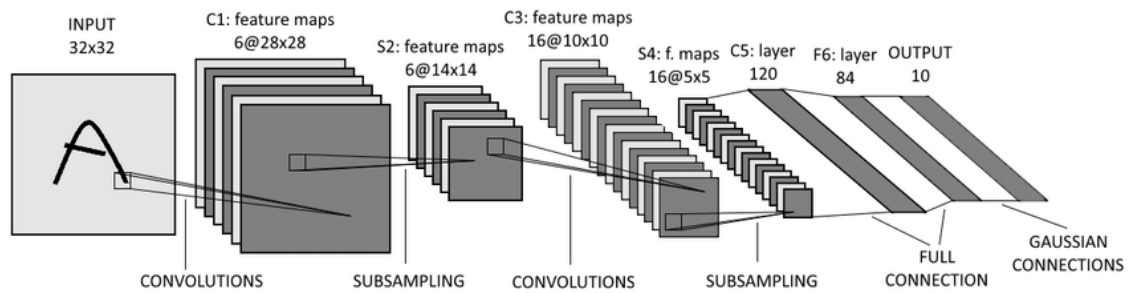
Model arsitektur Resnet dapat melakukan koneksi antar residu layer, hal ini dapat mengurangi nilai loss yang didapatkan. Koneksi antar residu layer juga dapat meningkatkan kinerja selama fase training. Koneksi antar residu pada sebuah lapisan mengartikan sebuah output pada suatu layer konvolusi [11]. Gambar 10 menunjukkan arsitektur dari ResNet-50



Gambar 10. Arsitektur ResNet-50 [11]

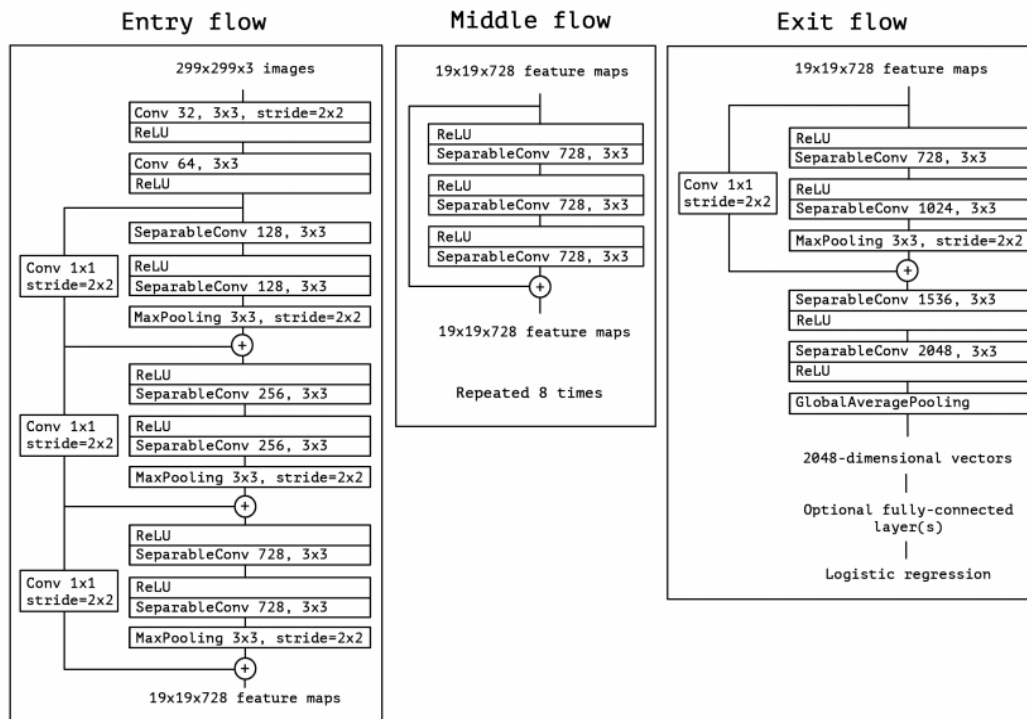
Lenet adalah sebuah arsitektur yang sangat efektif dari semua jenis arsitektur CNN [12]. Namun tingkat kemudahan dari arsitektur Lenet dapat mengabaikan proses konvolusi antar layer yang dapat menyebabkan nilai arsitektur yang menjadi rendah. Gambar 11 menunjukkan arsitektur dari LeNet.





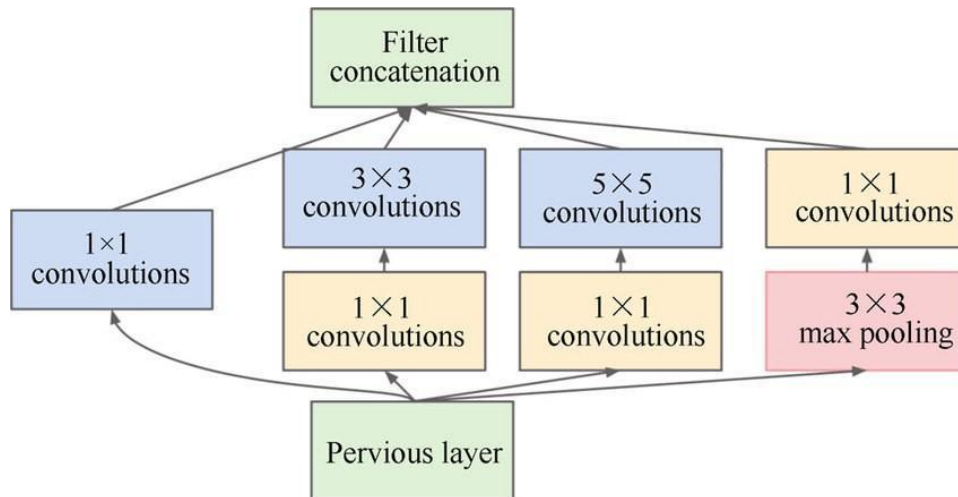
Gambar 11. Arsitektur LeNet [12]

Arsitektur Xception memodifikasi struktur dari model Inception dengan mengganti lapisan konvolusi dengan operasi depthwise separable convolution. Pada operasi depthwise separable convolution, konvolusi diterapkan pada masing-masing channel pada setiap titik piksel gambar (depthwise convolution), kemudian diikuti dengan konvolusi pointwise untuk menggabungkan informasi dari seluruh channel (pointwise convolution). Hal ini mengurangi jumlah parameter dan waktu komputasi yang dibutuhkan, namun tetap mempertahankan performa yang baik [13]. Gambar 12 menunjukkan arsitektur Xception.



Gambar 12. Arsitektur Xception [13]

GoogleNet menggunakan konsep Inception Module, di mana lapisan konvolusi yang berbeda dengan ukuran filter yang berbeda digabungkan untuk meningkatkan performa jaringan. GoogleNet memiliki banyak layer yang dalam bentuk blok-blok, di mana setiap blok terdiri dari beberapa Inception Module yang berbeda, dilanjutkan dengan layer pooling dan normalisasi [14], [15]. Arsitektur ini juga menggunakan teknik dropout untuk mengurangi overfitting. Gambar 13 menunjukkan arsitektur dari GoogleNet.



Gambar 13. Arsitektur GoogleNet [15]

Selanjutnya pada Tabel 2 merupakan spesifikasi komputer yang digunakan untuk melakukan training dataset.

Tabel 2. Spesifikasi komputer yang digunakan

Parameter	Specification
CPU	Intel(R) Core(TM) i7-10750H CPU @2.60GHz
RAM	16.0 GB
Space of Disk	350 GB
GPU Model Name	Nvidia Geforce GTX 1650 Ti

Tabel 3. Model Training Parameter

Transfer Learning Parameter	Optimizer	Learning Rate	Fully Connected Layer Parameter
VGG-16	Adam	0,0007	4096
ResNet-50			
Xception			
Lenet			
GoogleNet			

Table 3 menunjukkan parameter training model yang digunakan. Penggunaan nilai learning rate yang sama saat melakukan pelatihan model CNN memiliki beberapa alasan. Salah satu alasan utamanya adalah untuk menghindari kompleksitas dalam pengaturan dan tuning parameter. Dengan menggunakan nilai learning rate yang sama, proses pelatihan menjadi lebih sederhana dan mudah diimplementasikan. Selain itu, penggunaan nilai learning rate yang sama juga dapat mengurangi waktu komputasi yang dibutuhkan untuk pelatihan model. Jika kita menggunakan learning rate yang berbeda untuk setiap iterasi, hal ini akan memperpanjang waktu pelatihan karena perlu melakukan penyesuaian nilai learning rate pada setiap langkah. Selain alasan waktu komputasi dan kesederhanaan, penggunaan nilai learning rate yang sama juga dapat menghindari masalah seperti ketidakstabilan dan divergensi dalam pelatihan model. Jika nilai learning rate terlalu besar, model dapat mengalami loncatan yang signifikan dalam ruang

parameter dan tidak dapat mencapai konvergensi yang baik. Di sisi lain, jika nilai learning rate terlalu kecil, proses pelatihan dapat menjadi sangat lambat atau bahkan model mungkin sulit untuk belajar pola yang kompleks. Dengan menggunakan nilai learning rate yang sama, kita dapat mempertahankan keseimbangan antara stabilitas pelatihan dan kecepatan konvergensi.

#### 2.4 Testing

Pada proses validation atau testing ini dilakukan test pada semua model yang telah dilakukan proses training sebelumnya. Hal ini dilakukan agar dapat mengetahui nilai akurasi dari setiap model masing-masing. Proses validation ini dilakukan dengan menggunakan 15% dari total dataset yang digunakan pada setiap class atau nama obat. Namun penggunaan jumlah epoch yang dilakukan pada setiap model memiliki jumlah yang berbeda-beda. Pada proses testing ini juga menampilkan hasil dari perhitungan performa dalam melakukan klasifikasi atau mendeteksi nama obat. Detail gambar Confussion Matrix dan F1 Score hasil dari proses testing pada setiap model dapat dilihat pada gambar 3 - 7.

Pada rumus atau persamaan 1 merupakan perhitungan *accuracy*, yaitu dengan membagi penjumlahan dari *true positif* (TP) dan *true negatif* (TN). Kemudian dibagi dengan nilai penjumlahan *true positif* dan *true negatif*, dan *false positif* (FP), dan *false negatif* (FN).

$$\begin{aligned} \text{Accuracy} \\ &= \frac{TP + TN}{TP + TN + FP + FN} \end{aligned} \quad (1)$$

Pada rumus persamaan 2 menampilkan perhitungan mendapatkan nilai *Precision* yaitu membagi nilai *true positif*, dengan penjumlahan *true positif* (TP) dan *false positif* (FP)

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

Pada rumus persamaan 3 menampilkan perhitungan mendapatkan nilai *Recall* yaitu membagi nilai TP dengan hasil penjumlahan TP dan FN.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

Pada rumus persamaan 4 menampilkan perhitungan untuk mendapatkan nilai *F1 Score* yaitu membagi hasil perkalian *Precision* dan *Recall*, dengan hasil penjumlahan *Precision* dan *Recall*.

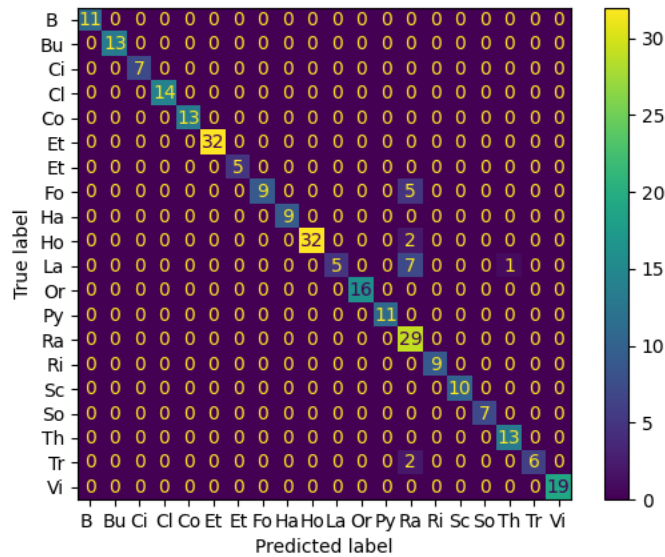
$$\begin{aligned} \text{F1 Score} \\ &= \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned} \quad (4)$$

### 3. HASIL DAN PEMBAHASAN

Bagian ini menjelaskan hasil penelitian yang telah dijelaskan pada bagian sebelumnya. Proses training yang dilakukan merupakan proses untuk melatih dataset pada saat pengujian pada proses testing. Kemudian pada proses testing dilakukan percobaan yang berguna untuk melakukan pembuktian bahwa sistem dapat melakukan klasifikasi atau menerjemahkan nama obat. Pembuktian hasil menggunakan Confussion Matrix dan F1 Score hasil dari proses testing untuk setiap arsitektur. Dari hasil proses testing yang telah dilakukan dapat dilihat pada Tabel 4 yang merupakan hasil akurasi pada setiap model arsitektur. Arsitektur ResNet-50 memiliki nilai tertinggi dari semua model arsitektur yang digunakan, hasil akurasi tersebut sebesar 97.56%.

Tabel 4. Hasil Akurasi Setiap Model

Metrics	VGG	ResNet-50	Xception	Lenet	GoogleNet
Training Time/Epoch (s)	2.5	3	2.5	1.5	3
Epoch	150	20	300	1000	1000
Testing Accuracy	94.07%	97.56%	77.00%	83.27%	71.77%



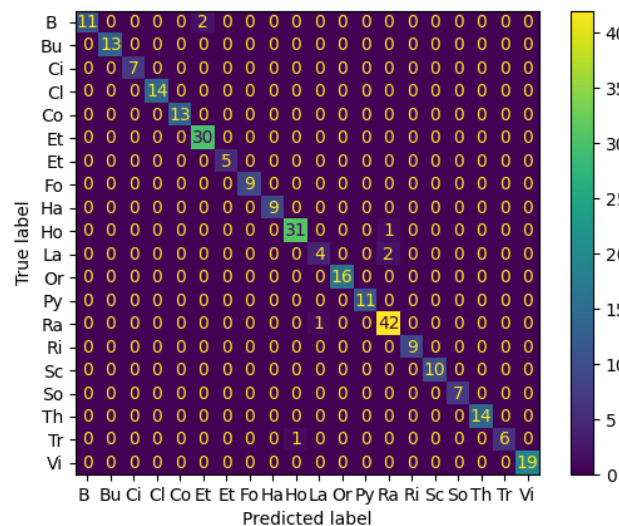
Gambar 14. Confussion Matrix VGG-16

Tabel 5. F1 Score VGG-16

Inisial Obat	Nama Obat	Precision	Recall	F1 Score
B	B Complex	1.00	1.00	1.00
Bu	Bufagan	1.00	1.00	1.00
Ci	Citostan	1.00	1.00	1.00
Cl	Clorpramazin	1.00	1.00	1.00
Co	Coparcetin	1.00	1.00	1.00
Et	Etagemycetin	1.00	1.00	1.00
Et	Etambutol	1.00	1.00	1.00
Fo	Fondazen	1.00	0.64	0.78
Ha	Haloperidol	1.00	1.00	1.00
Ho	Holidon	1.00	0.94	0.97
La	Lambucid	1.00	0.38	0.56
Or	Orphen	1.00	1.00	1.00
Py	Pyranzinamid	1.00	1.00	1.00

Ra	Ranitidine	0.64	1.00	0.78
Ri	Rifampicin	1.00	1.00	1.00
Sc	Scopmag	1.00	1.00	1.00
So	Solpenox	1.00	1.00	1.00
Th	Thryhexypenidyl	0.93	1.00	0.96
Tr	Trimakalk	1.00	0.75	0.86
Vi	Vibramox	1.00	1.00	1.00

Tabel 5 menunjukkan F1-Score untuk model VGG-16 dan Gambar 14 menunjukkan confusion matrix dari VGG-16. Pada hasil testing dengan menggunakan arsitektur transfer learning VGG-16 dan menggunakan epoch sebanyak 150 mendapatkan akurasi sebesar 94.07% dan menampilkan output yaitu confusion matrix serta perhitungan F1 Score. Dilihat dari hasil F1 Score, bahwa pada kelas nama obat Lambucid mendapatkan nilai F1 Score terkecil yaitu 0.56. Dari gambar kelas Lambucid sehingga mendapatkan nilai F1 score yang kecil mungkin disebabkan oleh pada huruf yang kurang jelas sehingga model kesulitan dalam menerjemahkannya. Namun hal ini juga dikonfirmasi pada nilai di confusion matrix kelas Lambucid mendapatkan nilai 5. Dari hasil confusion matrix juga terdapat 3 nilai tertinggi pada kelas Etagemycetin, Holidon, dan Ranitidine yang mendapatkan nilai rata - rata 30.



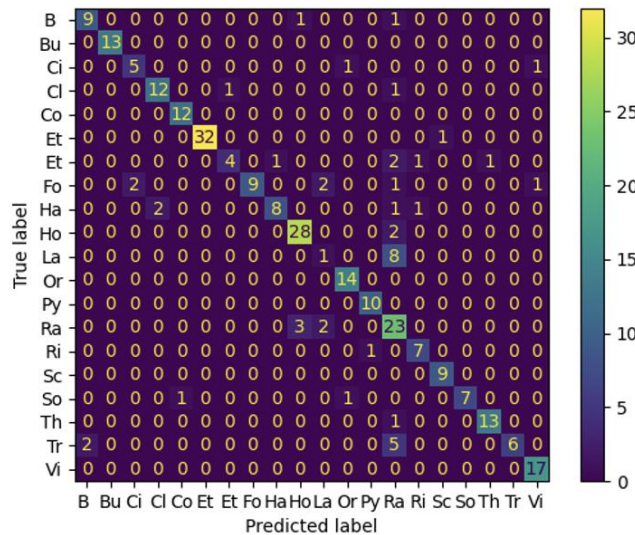
Gambar 15. Confussion Matrix ResNet-50

Tabel 6. F1 Score ResNet

Inisial Obat	Nama Obat	Precision	Recall	F1 Score
B	B Complex	1.00	0.85	0.92
Bu	Bufagan	1.00	1.00	1.00
Ci	Citostan	1.00	1.00	1.00
Cl	Clorpramazin	1.00	1.00	1.00
Co	Coparcetin	1.00	1.00	1.00
Et	Etagemycetin	0.94	1.00	0.97
Et	Etambutol	1.00	1.00	1.00
Fo	Fondazen	1.00	0.64	0.78
Ha	Haloperidol	1.00	1.00	1.00
Ho	Holidon	0.97	0.97	0.97
La	Lambucid	0.80	0.67	0.73
Or	Orphen	1.00	1.00	1.00
Py	Pyranzinamid	1.00	1.00	1.00
Ra	Ranitidine	0.93	0.98	0.95

Ri	Rifampicin	1.00	1.00	1.00
Sc	Scopmag	1.00	1.00	1.00
So	Solpenox	1.00	1.00	1.00
Th	Thryhexypenidyl	1.00	1.00	0.96
Tr	Trimakalk	1.00	0.86	0.92
Vi	Vibramox	1.00	1.00	1.00

Tabel 6 menunjukkan F1-Score untuk model ResNet-50 dan Gambar 15 menunjukkan confusion matrix dari ResNet-50. Hasil dari testing dengan menggunakan arsitektur transfer learning Resnet mendapatkan hasil akurasi sebesar 97.06%. Hal ini juga di konfirmasi pada nilai confusion matrix dan perhitungan penilaian F1 Score. Pada hasil perhitungan F1 score yang didapatkan, hasil penilaian F1 score pada arsitektur Resnet dan VGG memiliki kesamaan. Namun dengan menggunakan arsitektur Resnet pada kelas Lambucid mendapatkan peningkatan dengan nilai 0.73. Tidak hanya itu nilai tersebut juga hampir sama pada kelas nama obat Fondazen yang mendapatkan nilai 0.78. Perbedaan yang sangat signifikan antar hasil testing dengan menggunakan arsitektur Resnet dan VGG yaitu, pada penilaian confusion matrix Resnet pada kelas Ranitidine mendapatkan nilai tertinggi sebesar 42. Hal ini mungkin dikarenakan jumlah dataset yang digunakan pada kelas Ranitidine memiliki jumlah terbanyak dibandingkan jumlah dataset nama obat lainnya.



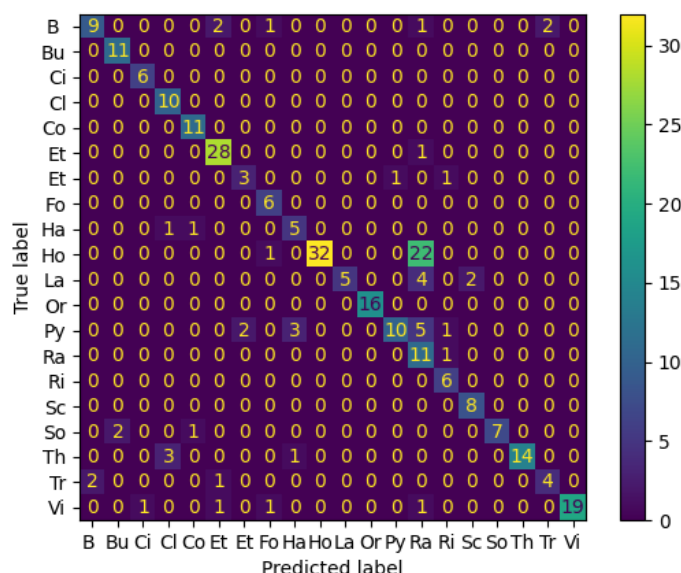
Gambar 16. Confussion Matrix Xception

Tabel 7. F1 Score Xception

Inisial Obat	Nama Obat	Precision	Recall	F1 Score
B	B Complex	0.82	0.82	0.82
Bu	Bufagan	1.00	1.00	1.00
Ci	Citostan	0.71	0.71	0.71
Cl	Clorpramazin	0.86	0.86	0.86
Co	Coparcetin	0.92	1.00	0.96
Et	Etagemycetin	1.00	0.97	0.98
Et	Etambutol	0.80	0.44	0.57
Fo	Fondazen	1.00	0.60	0.75
Ha	Haloperidol	0.89	0.67	0.76
Ho	Holidon	0.88	0.93	0.90
La	Lambucid	0.20	0.11	0.14
Or	Orphen	0.88	1.00	0.93

Py	Pyranzinamid	0.91	1.00	0.95
Ra	Ranitidine	0.51	0.82	0.63
Ri	Rifampicin	0.78	0.88	0.82
Sc	Scopmag	0.90	1.00	0.95
So	Solpenox	1.00	0.78	0.88
Th	Thryhexypenidyl	0.93	0.93	0.93
Tr	Trimakalk	1.00	0.46	0.63
Vi	Vibramox	0.89	1.00	0.94

Tabel 7 menunjukkan F1-Score untuk model Xception dan Gambar 16 menunjukkan confusion matrix dari Xception. Pada hasil testing menggunakan arsitektur transfer learning Xception dengan percobaan sebanyak 300 epoch mendapatkan hasil akurasi sebesar 77.0%. Hal ini juga dikonfirmasi dari hasil perhitungan F1 Score. Pada kelas nama obat Lambucid mendapatkan nilai terkecil dari kelima uji coba testing yang telah dilakukan yaitu sebesar 0.14. Selanjutnya pada kelas nama obat Bufagan mendapatkan penilaian F1 score tertinggi yaitu 1.00. Pada hasil confusion matrix dapat dilihat pada kelas Lambucid mendapatkan nilai 1, dan pada kelas Etagemycetin mendapatkan nilai tertinggi yaitu sebesar 32.



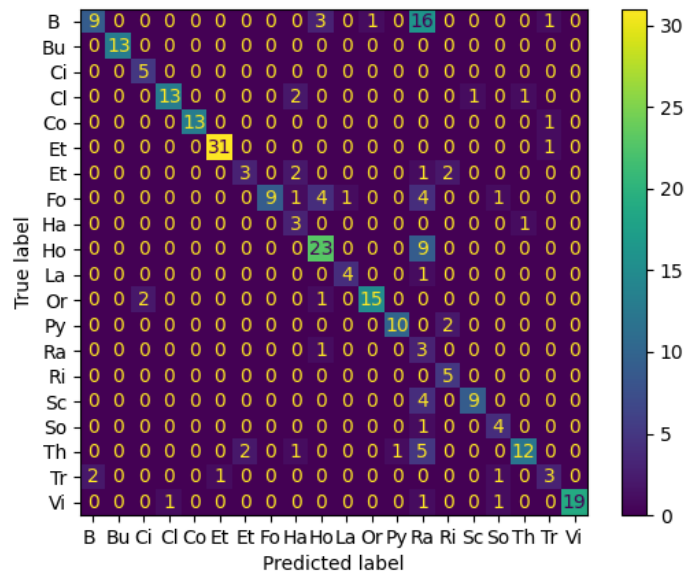
Gambar 17. Confussion Matrix Lenet

Tabel 8. F1 Score LeNet

Inisial Obat	Nama Obat	Precision	Recall	F1 Score
B	B Complex	0.82	0.60	0.69
Bu	Bufagan	0.85	1.00	0.92
Ci	Citostan	0.86	1.00	0.92
Cl	Clorpramazin	0.71	1.00	0.83
Co	Coparcetin	0.85	1.00	0.92
Et	Etagemycetin	0.88	0.97	0.92
Et	Etambutol	0.60	0.60	0.60
Fo	Fondazen	0.67	1.00	0.80
Ha	Haloperidol	0.56	0.71	0.63
Ho	Holidon	1.00	0.58	0.74
La	Lambucid	1.00	0.45	0.62
Or	Orphen	1.00	1.00	1.00

Py	Pyranzinamid	0.91	0.48	0.62
Ra	Ranitidine	0.24	0.92	0.39
Ri	Rifampicin	0.67	1.00	0.80
Sc	Scopmag	0.80	1.00	0.89
So	Solpenox	1.00	0.70	0.82
Th	Thryhexypenidyl	1.00	0.78	0.88
Tr	Trimakalk	0.67	0.57	0.62
Vi	Vibramox	1.00	0.83	0.90

Tabel 8 menunjukkan F1-Score untuk model LeNet dan Gambar 15 menunjukkan confusion matrix dari LeNet-50. Dari hasil testing dengan menggunakan arsitektur transfer learning Lenet dengan percobaan sebanyak 1000 epoch yang memakan waktu 1.5 detik per epoch, mendapatkan hasil akurasi sebesar 83.27%. Hal ini juga di konfirmasi dari hasil perhitungan F1 Score dan confusion Matrix. Dapat dilihat pada hasil perhitungan F1 Score kelas nama obat Ranitidine mendapatkan hasil F1 Score terkecil yaitu sebesar 0.39. Pada hasil perhitungan confusion matrix, kelas nama obat Ethambutol 3. Namun kelas nama obat Ethagemycetin dan Holidon mendapatkan nilai confusion matrix tertinggi dengan rata-rata nilai sebesar 30.



Gambar 18. Confussion Matrix GoogleNet

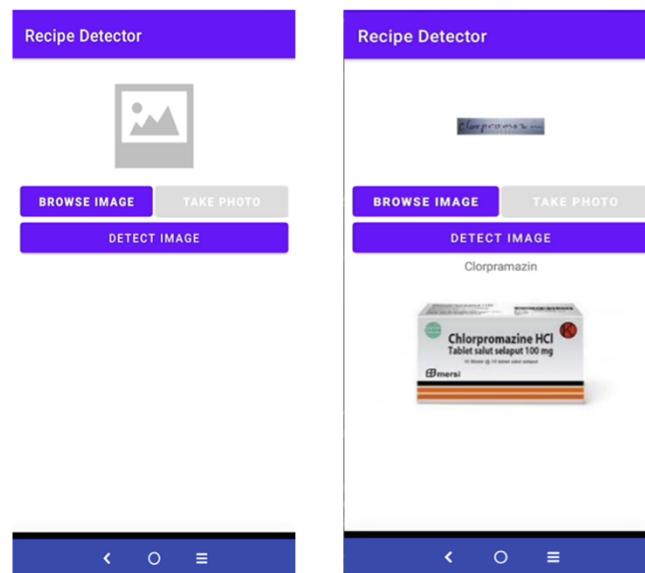
Tabel 9. F1 Score GoogleNet

Inisial Obat	Nama Obat	Precision	Recall	F1 Score
B	B Complex	0.82	0.30	0.44
Bu	Bufagan	1.00	1.00	1.00
Ci	Citostan	0.71	1.00	0.83
Cl	Clorpramazin	0.93	0.76	0.84
Co	Coparcetin	1.00	0.93	0.96
Et	Etagemycetin	0.97	0.97	0.97
Et	Etambutol	0.60	0.38	0.46
Fo	Fondazen	1.00	0.45	0.62
Ha	Haloperidol	0.33	0.75	0.46
Ho	Holidon	0.72	0.72	0.72
La	Lambucid	0.80	0.80	0.80
Or	Orphen	0.94	0.83	0.88



Py	Pyranzinamid	0.91	0.83	0.87
Ra	Ranitidine	0.07	0.75	0.12
Ri	Rifampicin	0.56	1.00	0.71
Sc	Scopmag	0.90	0.69	0.78
So	Solpenox	0.57	0.80	0.67
Th	Thryhexypenidyl	0.86	0.57	0.69
Tr	Trimakalk	0.50	0.43	0.46
Vi	Vibramox	1.00	0.86	0.93

Tabel 9 menunjukkan F1-Score untuk model GoogleNet dan Gambar 16 menunjukkan confusion matrix dari GoogleNet. Dari hasil proses training dengan menggunakan arsitektur transfer learning GoogleNet sebanyak 1000 epoch mendapatkan akurasi terkecil dari kelima percobaan arsitektur yang digunakan yaitu, sebesar 71.77%. Hal ini juga dikonfirmasi dari hasil F1 score, pada kelas nama obat Ranitidine mendapatkan nilai terkecil dari perhitungan F1 Score mendapatkan nilai sebesar 0.12. Pada hasil confusion matrix kelas Ethambutol, Ranitidine, Haloperidol, dan Threhexypenidyl mendapatkan nilai terkecil yaitu sebesar 3. Namun pada kelas Etagemycetin mendapatkan nilai tertinggi yaitu sebesar 31.



Gambar 19. Aplikasi Klasifikasi Nama Obat

Setelah dilakukan proses klasifikasi dan perbandingan dengan menggunakan kelima arsitektur transfer learning, dilakukan pembuatan aplikasi berbasis android yang dapat menerjemahkan atau mengklasifikasi nama obat. Pembuatan aplikasi ini menggunakan model ResNet-50, dikarenakan model ResNet-50 mendapatkan nilai akurasi tertinggi dari kelima arsitektur transfer learning lainnya. Aplikasi ini menggunakan tiga fitur pendukung untuk memudahkan pengguna aplikasi, yaitu unggah gambar, ambil gambar, dan bentuk visual kotak obat. Gambar 19 menunjukkan user interface aplikasi klasifikasi resep obat. Fitur unggah gambar dan ambil gambar agar pengguna dapat melakukan unggah gambar pada penyimpanan di *smartphone*, kemudian fitur ambil gambar agar pengguna dapat melakukan pengambilan gambar secara langsung dengan menggunakan kamera di *smartphone*. Selanjutnya fitur pemberian visual kotak obat agar pengguna dapat mengetahui bentuk atau kotak obat. Pembuatan aplikasi ini diharapkan agar pengguna aplikasi tidak mengalami kesulitan atau kesalahan dalam melakukan pembacaan nama obat pada resep obat.

#### 4. KESIMPULAN DAN SARAN

Dari penelitian yang telah dilakukan uji coba mendapatkan hasil arsitektur terbaik dalam melakukan klasifikasi nama obat dengan menggunakan transfer learning model arsitektur ResNet. Dari penelitian ini juga dapat disimpulkan sistem berjalan dengan baik, serta sangat membantu para pasien dan apoteker dalam membaca nama obat atau resep obat yang diberikan dokter umum maupun dokter spesialis. Harapan kepada peneliti selanjutnya untuk jumlah dataset dan penulisan resep obat dapat ditambahkan lebih banyak lagi, dikarenakan tulisan tangan nama obat pada resep obat memiliki berbagai macam tulisan yang berbeda-beda setiap dokter umum maupun dokter spesialis. Selain itu penggunaan LSTM dalam kombinasi dengan model CNN dapat menjadi pendekatan yang menarik untuk lebih mengenali resep obat karena mengenali Urutan dan Konteks: LSTM (Long Short-Term Memory) adalah jenis jaringan saraf rekuren (RNN) yang memiliki kemampuan untuk mengenali urutan data. Dalam konteks resep obat, urutan kata dalam nama obat atau instruksi penggunaan dapat memiliki arti penting. Dengan menggunakan LSTM, model dapat mengambil informasi kontekstual dan mempelajari hubungan antara kata-kata dalam urutan tertentu, sehingga dapat meningkatkan pemahaman tentang resep obat secara keseluruhan.

#### DAFTAR PUSTAKA

- [1] R. Achkar, K. Ghayad, R. Haidar, S. Saleh, and R. Al Hajj, "Medical handwritten prescription recognition using CRNN," *CITS 2019 - Proceeding 2019 Int. Conf. Comput. Inf. Telecommun. Syst.*, Aug. 2019, doi: 10.1109/CITS.2019.8862004.
- [2] Arrahman, Purwanto, and P. Nurtantio, "Klasifikasi Nama Obat Tulisan Tangan Dokter dengan Metode GLCM dan Backpropagation Neural Network," *J. Teknol. Inf.*, vol. 12, pp. 123–134, 2016.
- [3] T. Jain, R. Sharma, and R. Malhotra, "Handwriting Recognition for Medical Prescriptions using a CNN-Bi-LSTM Model," *2021 6th Int. Conf. Converg. Technol. I2CT 2021*, Apr. 2021, doi: 10.1109/I2CT51068.2021.9418153.
- [4] E. Hassan, H. Tarek, M. Hazem, S. Bahnacy, L. Shaheen, and W. H. Elashmwai, "Medical Prescription Recognition using Machine Learning," *2021 IEEE 11th Annu. Comput. Commun. Work. Conf. CCWC 2021*, pp. 973–979, Jan. 2021, doi: 10.1109/CCWC51732.2021.9376141.
- [5] N. Palani and N. Sampath, "Detecting and Extracting Information of Medicines from a Medical Prescription Using Deep Learning and Computer Vision," *2022 Int. Conf. Knowl. Eng. Commun. Syst.*, pp. 1–6, Dec. 2022, doi: 10.1109/ICKECS56523.2022.10060502.
- [6] C. Kanan and G. W. Cottrell, "Color-to-Grayscale: Does the Method Matter in Image Recognition?," *PLoS One*, vol. 7, no. 1, p. e29740, Jan. 2012, doi: 10.1371/JOURNAL.PONE.0029740.
- [7] G. R. Arce and M. P. McIoughlin, "Theoretical analysis of the max/median filter," *IEEE Trans. Acoust.*, vol. 35, no. 1, pp. 60–69, 1987, doi: 10.1109/TASSP.1987.1165036.
- [8] Y. S. Wang, C. L. Tai, O. Sorkine, and T. Y. Lee, "Optimized scale-and-stretch for image resizing," *ACM SIGGRAPH Asia 2008 Pap. SIGGRAPH Asia '08*, 2008, doi: 10.1145/1457515.1409071.
- [9] U. Erkan, S. Enginoğlu, D. N. H. Thanh, and L. M. Hieu, "Adaptive frequency median filter for the salt and pepper denoising problem," *IET Image Process.*, vol. 14, no. 7, pp. 1240–1247, 2020, doi: 10.1049/iet-ipr.2019.0398.

- [10] R. Saini, P. Semwal, and T. H. Jaware, "Brain Tumor Classification Using VGG-16 and MobileNetV2 Deep Learning Techniques on Magnetic Resonance Images (MRI)," pp. 300–313, 2023, doi: 10.1007/978-3-031-27609-5\_24.
- [11] A. Victor Ikechukwu, S. Murali, R. Deepu, and R. C. Shivamurthy, "ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images," *Glob. Transitions Proc.*, vol. 2, no. 2, pp. 375–381, Nov. 2021, doi: 10.1016/J.GLTP.2021.08.027.
- [12] Y. Lecun, L. Bottou, Y. Bengio, and P. Ha, "LeNet," *Proc. IEEE*, no. November, pp. 1–46, 1998.
- [13] F. Chollet, "XCeption: Deep Learning with Depthwise Separable Convolutions," *Comput. Vis. Found.*, 2016, doi: 10.4271/2014-01-0975.
- [14] C. Szegedy *et al.*, "Going deeper with convolutions," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.
- [15] R. Anand, T. Shanthi, M. S. Nithish, and S. Lakshman, "Face Recognition and Classification Using GoogleNET Architecture," *Adv. Intell. Syst. Comput.*, vol. 1048, pp. 261–269, 2020, doi: 10.1007/978-981-15-0035-0\_20/COVER.