

**PENERAPAN *MOBILENET ARCHITECTURE* PADA IDENTIFIKASI
FOTO CITRA MAKANAN INDONESIA**

SKRIPSI

Diajukan Untuk Memenuhi Salah Satu Syarat Guna Memperoleh
Gelar Sarjana Komputer (S.Kom) Pada Program Studi Teknik Informatika
Fakultas Teknik dan Ilmu Komputer Universitas Nusantara PGRI Kediri



Disusun Oleh :

MUHAMMAD FARID WIJAYANTO

NPM : 2013020107

**FAKULTAS TEKNIK DAN ILMU KOMPUTER (FTIK)
UNIVERSITAS NUSANTARA PERSATUAN GURU
REPUBLIK INDONESIA
UN PGRI KEDIRI
TAHUN 2024**

BAB I

PENDAHULUAN

A. Latar Belakang

Mulai tahun 2020, memotret makanan sebelum makan sudah menjadi kebiasaan yang membawa kegembiraan bagi pecinta kuliner. Itu sebabnya situs media sosial seperti Facebook dan Instagram dipenuhi dengan foto masakan yang pernah Anda coba, rekomendasi masakan, bahkan masakan yang Anda coba buat sendiri. Namun, orang yang melihat foto tersebut seringkali lupa atau tidak mengetahui nama makanan tersebut (Annisa, 2023).

Foto makanan di media sosial seringkali tidak mencantumkan nama makanan tersebut. Untuk itu, beberapa peneliti telah melakukan penelitian untuk mengidentifikasi nama-nama makanan dalam gambar. Sarah Yuli Evangelista Simarmata melakukan penelitian klasifikasi citra makanan pada tahun 2019 menggunakan *learning vector quantization* (LVQ) berbasis histogram warna dan ekstraksi fitur GLCM (*gray level co-occurrence matriks*). Gambar makanan yang digunakan dalam penelitian ini merupakan data primer yang diperoleh dari foto gambar makanan yang diambil oleh peneliti sendiri dengan menggunakan kamera *smartphone* "Huawei Nova 3i." Data yang digunakan terdiri dari 15 kelas makanan dan 17 data untuk masing-masing kelas, sehingga total keseluruhannya adalah 255 data. Gambar makanan yang diambil terdiri dari makanan utuh, setengah bagian, dan gambar diambil dari sudut vertikal dan miring. Hasil pengujian menunjukkan nilai akurasi terbaik sebesar 53.33 dengan nilai parameter terbaik *learning rate* 0.1, nilai *epoch* maksimum 2, nilai pengurangan *learning rate* 0.01, dan nilai *learning rate* minimum 0.01 untuk data latih 240 dan data uji. 15. Namun akurasi yang diperoleh masih 53,33%. (Simarmata, 2019).

Salah satu metode yang sering digunakan peneliti untuk klasifikasi gambar adalah *deep learning*. *Deep Learning* merupakan kecerdasan buatan dengan representasi sederhana dengan lapisan tersembunyi untuk mengolah data yang dapat menciptakan konsep yang kompleks. *Deep Learning* dapat dilatih

untuk mengenali dan mengklasifikasikan objek (Hartono dkk, 2022). *Convolutional Neural Network* (CNN/ConvNet) adalah bagian dari *deep neural network*, sejenis jaringan saraf tiruan yang biasa digunakan dalam pengenalan dan pemrosesan gambar. Algoritma ini dirancang khusus untuk memproses data piksel dan gambar visual (Trivusi, 2022).

Citra Mahaputri pada tahun 2022, membuat pengenalan citra atau gambar untuk mendeteksi makanan dengan menggunakan model CNN. Data yang digunakan berupa citra gambar makanan tradisional nusantara sebanyak 1.202 gambar yang terdiri dari 20 kelas makanan dengan ukuran 224 x 224 piksel. Berdasarkan hasil penelitiannya didapatkan bahwa model EfficientNetV2 menghasilkan performa yang baik. Dimana akurasi yang didapat sebesar 83,82%, yang lebih tinggi dibandingkan dengan hasil model Resnet50 dan EfficientNetB6 (80,08%) (Mahaputri & Hari Wisana, 2022).

Berdasarkan uraian masalah dan penelitian sebelumnya, maka penulis mengajukan penelitian guna mengidentifikasi gambar makanan menggunakan metode CNN dengan *MobileNet Architecture* untuk mendapatkan performa terbaik dari mengidentifikasi jenis makanan. Penelitian ini dilakukan simulasi menggunakan *EfficientNetV2 Architecture* dan *MobileNet Architecture*. *MobileNet Architecture* di dapatkan performa yang lebih baik dari komputasi dan hasil akurasi daripada *EfficientNetV2*. Pada penelitian ini data diperoleh dari *roboflow*, kemudian data tersebut dilakukan pemrosesan data dan dilakukan pemodelan serta mencari performa terbaik menggunakan metode CNN dengan *MobileNet Architecture* sehingga dikenali nama makanan gambar tersebut.

B. Identifikasi Masalah

Banyaknya foto makanan Indonesia yang diposting di media sosial tanpa disertai informasi nama makanan mengakibatkan kesulitan bagi orang untuk mengidentifikasi jenis makanan yang ditampilkan.

C. Rumusan Masalah

Berdasarkan latar belakang dan identifikasi masalah, rumusan dari penelitian ini adalah :

1. Bagaimana implementasi model *Convolutional Neural Networks* dengan *MobileNet Architecture* ?
2. Bagaimana mendapatkan performa terbaik dari model *Convolutional Neural Networks* dengan menggunakan *MobileNet Architecture* ?

D. Batasan Masalah

Untuk mencapai hasil yang maksimal dalam penelitian dan fokus pada suatu masalah tertentu, maka batasan masalah yang akan diteliti adalah berikut :

1. Peneliti hanya akan menggunakan bahasa pemrograman *Python* untuk mengimplementasikan *MobileNet Architecture*.
2. Peneliti menggunakan *Tensorflow* sebagai *framework* implementasi *MobileNet Architecture*.
3. Peneliti hanya menggunakan data uji 20% dari total data gambar atau dataset yang ada.
4. Peneliti menggunakan 20 foto citra makanan indonesia.
5. Label nama makanan pada dataset diberikan sesuai asal dari makanan tersebut.
6. Data gambar yang akan digunakan adalah gambar atau foto makanan yang spesifik dan jelas. Gambar atau foto haruslah minimal 35 untuk tiap jenis makanan.
7. Hasil akhir penelitian ini adalah sistem yang dapat melakukan klasifikasi foto citra makanan indonesia dengan output akhir berupa identifikasi nama dari foto citra makanan indonesia.

E. Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah pada penelitian, tujuan penelitian ini adalah :

1. Untuk mengetahui hasil dari identifikasi foto citra makanan indonesia menggunakan Arsitektur dari *MobileNet*.
2. Untuk mencapai performa terbaik dari model *Convolutional Neural Network* menggunakan arsitektur *MobileNet*

F. Manfaat dan Kegunaan Penelitian

Sistem dapat memberikan kemudahan bagi kebanyakan masyarakat terutama pecinta kuliner untuk mengidentifikasi jenis makanan indonesia.

G. Metode Penelitian

1. Studi Literatur

Mengumpulkan bahan-bahan dari berbagai sumber, antara lain buku, jurnal, artikel, dokumen penelitian terdahulu, dan sumber lain yang berkaitan dengan penelitian ini. Selanjutnya dikembangkan ulasan dari sumber-sumber tersebut untuk memperoleh materi terperinci, mengidentifikasi masalah, dan mengembangkan teori tentang masalah penelitian.

2. Pengumpulan Data

Pengumpulan data dilakukan dan diperoleh dari laman berikut universe.roboflow.com/bangkit/indonesian-food-pedsx/dataset dengan total 903 citra gambar.

3. Analisa dan Desain Sistem

Peneliti melakukan analisis sistem berdasarkan tinjauan literatur untuk mengidentifikasi metode klasifikasi yang dapat digunakan untuk mengidentifikasi jenis makanan. Hal ini memungkinkan untuk membuat desain sistem yang sesuai, yang kemudian dapat digunakan untuk implementasi sistem sesuai dengan desain sistem yang dibuat.

4. Implementasi

Peneliti melakukan tahap implementasi algoritma dan model dengan menggunakan bahasa pemrograman *python*. Hasil dari fase ini adalah program yang sesuai dengan desain yang dibuat selama tahap analisis dan desain sistem.

5. Uji Coba

Pengujian yang dilakukan fokus pada logika sistem dan memastikan bahwa semua bagian diuji. Hal ini dilakukan untuk

menguji sistem agar meminimalisir kesalahan dan memastikan keluaran yang dihasilkan sudah sesuai dengan yang dituju.

6. Maintenance

Tahap ini tidak mengecualikan bahwa sistem yang berjalan dapat berubah karena kesalahan yang tidak terdeteksi yang terjadi selama pengujian, atau sistem perlu beradaptasi dengan lingkungan baru, sehingga melakukan pemeliharaan pada sistem.

7. Laporan

Tahapan penulisan laporan didapat dari studi literatur, pengumpulan data, melakukan analisis dan perancangan sistem, implementasi, pengujian dan pemeliharaan, dengan hasil pembahasan dan kesimpulan.

H. Jadwal Kegiatan

Adapun jadwal dilaksanakan penelitian seperti pada tabel (1.1) :

Tabel 1.1 Jadwal Penelitian

No.	Kegiatan	Bulan					
		Juli	Agt	Sep	Okt	Nev	Des
1	Studi Literatur						
2	Pengumpulan Data						
3	Analisa dan Desain Sistem						
4	Implementasi						
5	Uji Coba						
6	Maintenance						
7	Laporan						

I. Sistematika Penulisan

Sistematika dalam penulisan laporan penelitian ini sebagai berikut :

BAB I : PENDAHULUAN

Bab ini berisi latar belakang, identifikasi masalah, rumusan masalah, tujuan penelitian, batasan masalah, manfaat dan kegunaan penelitian, metode penelitian, jadwal penelitian serta sistematika penulisan laporan. Pada dasarnya, ini memberikan gambaran umum tentang konteks dan maksud penelitian.

BAB II : TINJAUAN PUSTAKA

Bab ini membahas kajian literatur atau referensi terkait yang menjadi dasar teoritis untuk penelitian. Ini mencakup penelitian terdahulu, konsep-konsep teoritis, atau kerangka pemikiran yang mendukung penelitian yang sedang dilakukan.

BAB III : ANALISA DAN PEMODELAN SISTEM

Bab ini menjelaskan analisa sistem dan hasil desain sistem serta kebutuhan – kebutuhan yang didapatkan dari analisa dan desain sistem yang sudah dibuat.

BAB IV : HASIL DAN EVAKUASI

Bagian ini memaparkan hasil dari implementasi sistem atau eksperimen yang telah dilakukan. Selain itu, mencakup evaluasi terhadap hasil tersebut, termasuk interpretasi data, analisis kesalahan, dan penilaian kinerja sistem yang dikembangkan.

BAB V : PENUTUP

Bab ini terdapat simpulan dan harapan - harapan yang dituliskan berdasarkan rancangan tinjauan pustaka.

BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. Klasifikasi

Klasifikasi adalah suatu metode pengelompokan objek berdasarkan ciri-ciri yang dimiliki objek yang diklasifikasikan tersebut. Klasifikasi dapat dilakukan dengan berbagai cara, secara manual atau dengan bantuan teknologi. Klasifikasi manual adalah klasifikasi yang dilakukan oleh manusia tanpa bantuan algoritma komputer cerdas. Sedangkan Pengklasifikasian menggunakan berbagai algoritma teknologi termasuk *Naive Bayes*, *Support Vector Machines*, *Decision Trees*, *Fuzzy*, dan *Artificial Neural Networks*, meningkatkan efektivitas dalam analisis data (Wibawa dkk, 2018).

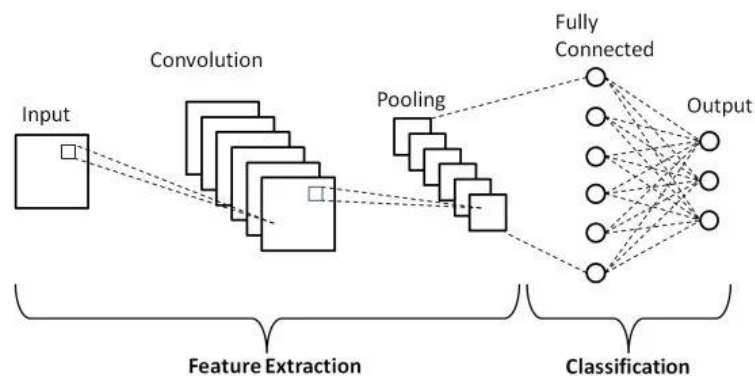
2. Deep Learning

Proses pembelajaran AI disebut juga pembelajaran. Dalam bidang AI, terdapat proses pembelajaran yang spesifik atau mendetail yang disebut *deep learning*. *Deep learning* merupakan proses pembelajaran yang menggunakan algoritma yang berkaitan dengan hukum matematika yang berfungsi seperti otak manusia. Pembelajaran mendalam digunakan untuk berbagai tugas, termasuk memprediksi peluang dan peristiwa, mengenali objek, dan mendiagnosis penyakit. Salah satu kemungkinan penerapan deep learning adalah pada bidang pengolahan citra atau pengolahan citra digital. Sistem pengolahan citra bertujuan untuk membantu manusia mengenali atau mengklasifikasikan objek secara efisien, cepat, dan akurat, serta dapat memproses data dalam jumlah besar sekaligus (Maulana & Rochmawati, 2019).

3. Convolutional Neural Networks (CNN)

Convolutional Neural Networks merupakan salah satu jenis algoritma *deep learning* yang dapat menerima masukan gambar dan menentukan objek pada gambar, serta dapat digunakan sebagai sumber pengenalan gambar untuk

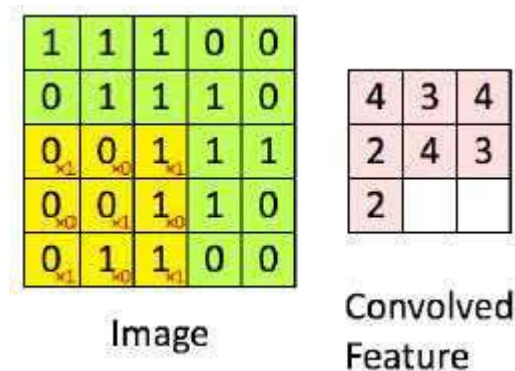
membedakan setiap gambar yang ada (Iswantoro & UN, 2022). CNN menggunakan serangkaian *convolutional layer* untuk melatih dan menguji setiap gambar masukan, kemudian dilakukan proses *pooling* untuk mengekstrak fitur dari gambar masukan yang berurutan. Setelah proses *pooling* selesai, gambar kemudian diratakan dan dimasukkan ke dalam proses *fully connected layer* untuk menyelesaikan klasifikasi. Arsitektur yang menggambarkan proses dalam metode CNN (*Convolutional Neural Network*) diilustrasikan dalam gambar (2.1) (Felix dkk, 2019).



Gambar 2.1 *Convolutional Neural Networks*

4. Convolutional Layers

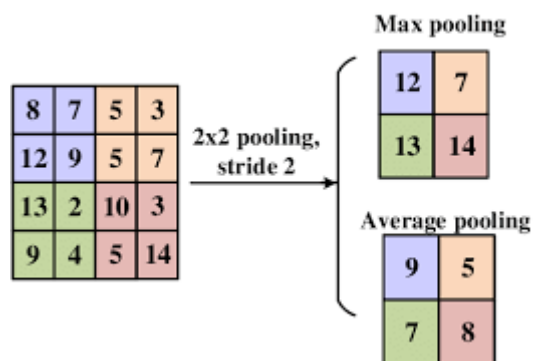
Convolutional layer ini merupakan operasi konvolusi pada keluaran lapisan sebelumnya. Lapisan ini merupakan proses utama yang mendasari CNN dan merupakan lapisan utama terpenting dari sistem yang digunakan dalam CNN. Konvolusi dalam pengolahan citra berarti menerapkan kernel (kotak kuning) ke citra pada semua kemungkinan *offset*, seperti yang ditunjukkan pada Gambar (2.2). Citra keseluruhannya adalah kotak lipat berwarna hijau. Kernel berpindah dari pojok kiri atas ke pojok kanan bawah. Hasil konvolusi citra ini ditunjukkan pada gambar sebelah kanan (Sandi dkk, 2022).



Gambar 2.2 Convolutional Layer

5. Pooling Layers

Lapisan penyatuan adalah lapisan yang digunakan untuk mengurangi jumlah fitur yang diambil dari lapisan sebelumnya. Keuntungannya adalah efektivitas model meningkat seiring Anda terus melatihnya. Metode pooling yang umum digunakan adalah max pooling, dan nilai tersebut diperoleh dari hasil convolutional layer. Jenis pengumpulan lainnya adalah pengumpulan rata-rata atau pengumpulan norma L2 (Putra dkk, 2023). Ilustrasi pooling layer bisa dilihat oleh gambar (2.3).



Gambar 2.3 Pooling Layers

6. Fully-Connected Layers

Fully connected layer adalah susunan dimana semua neuron yang diaktifkan di lapisan sebelumnya terhubung sepenuhnya ke neuron di lapisan berikutnya, mirip dengan *neural network* biasa. Berbeda dengan susunan *fully connected* yang memiliki neuron yang terhubung sepenuhnya, susunan dalam

konvolusi biasa ini tidak selalu terhubung ke wilayah masukan. Namun, karena kedua susunan beroperasi pada dot produk, penggunaannya tidak jauh berbeda (Iswantoro & UN, 2022).

7. Parameter CNN yang Signifikan

a. *Kernel-Size*

Filter atau kernel ini digunakan untuk mendapatkan fragmen (*strides*) dari gambar. (Hendriyana & Maulana, 2020). *Kernel size* ini menentukan ukuran jendela geser. Umumnya disarankan untuk menggunakan ukuran jendela yang lebih kecil, sebaiknya nilai ganjil seperti 1, 3, 5, dan 7 (Umer, 2023).

b. Fungsi Aktivasi

Fungsi aktivasi adalah fungsi matematika lapisan tersembunyi yang digunakan untuk memodifikasi *Neural Networks* secara nonlinier. Fungsi aktivasi yang tersedia dalam framework TensorFlow meliputi *Tanh*, *Softsign*, *Sigmoid*, *Softmax*, *Softplus*, *Linear*, *Relu*, *Elu*, dan *Exponential* (Gunawan, 2020).

c. Pengeluaran (*Dropout*)

Dropout merupakan fungsi matematika yang digunakan untuk menentukan persentase *dropout* nilai yang dianggap terburuk dari sekumpulan nilai yang dicapai selama suatu masa studi. Tujuan dari penghapusan ini adalah untuk mengurangi *overfitting* pada model, khususnya kondisi dimana nilai akurasi yang dihasilkan terlalu sempurna, misal kondisi bernilai 1 (Gunawan, 2020).

d. Optimasi

Metode optimasi merupakan metode pencarian yang memperbarui nilai bobot melalui pengurangan atau penambahan pada model pengembangan ANN sehingga dapat dihasilkan bobot yang optimal (Suradiradja, 2021). Metode optimasi antara lain adalah *Adamax*, *RMShop*, *Adam*, *SGD*, *Adagrad*, *Adadelata* (Suradiradja, 2021).

8. Mobile Net

MobileNet adalah arsitektur *convolutional neural networks* (CNN) yang dapat mengatasi kebutuhan untuk menghitung data dalam jumlah besar. Secara umum, perbedaan mendasar antara arsitektur MobileNet dan CNN adalah keduanya menggunakan lapisan konvolusional dengan ketebalan filter yang sama dengan ketebalan gambar masukan. MobileNet membagi konvolusi menjadi *depthwise convolution* dan *pointwise convolution* (Hendriyana & Maulana, 2020).

B. Kajian Pustaka

Penelitian yang membahas tentang algoritma CNN salah satunya dilakukan oleh Isna Wulandari dkk yang berjudul “Klasifikasi Citra Digital Bumbu dan Rempah dengan Algoritma Convolutional Neural Network (CNN)”. Pada penelitian ini mempunyai 3 kelas yaitu ginseng, jahe, dan lengkuas. Hasil pengujian penelitian ini didapatkan nilai akurasi data training sebesar 0,9875 dan nilai loss 0,0769. Nilai akurasi data testing sebesar 0,85 dan nilai loss 0,4773. Sementara itu, pengujian dengan data baru yaitu masing-masing 3 citra untuk setiap kategori menghasilkan akurasi sebesar 88,89% (Wulandari dkk, 2020).

Pada penelitian lain juga membahas tentang algoritma CNN dengan judul “Purwarupa Sistem Deteksi COVID-19 Berbasis Website Menggunakan Algoritma Convolutional Neural Network” yang diteliti oleh Ari Fadli dkk. Dalam penelitian ini, sistem berbasis situs web untuk mendeteksi virus COVID-19 dari gambar x-ray toraks dikembangkan menggunakan algoritma convolutional neural network. Hasil penelitian yang diperoleh menunjukkan bahwa sistem dapat mengklasifikasikan x-ray toraks menjadi tiga kelas yaitu (COVID-19, virus pneumonia, dan normal). Nilai akurasi yang diperoleh mencapai 89,6% dan nilai F1 mencapai 87,9% (Fadli dkk, 2021).

Penelitian yang sama menggunakan algoritma CNN dilakukan oleh Nur Fadlia dan Rifki Kosasih dengan judul “Klasifikasi Jenis Kendaraan Menggunakan Metode Convolutional Neural Network (CNN)”. Pada penelitian ini data yang digunakan adalah 120 gambar yang terdiri dari gambar mobil,

sepeda motor, dan sepeda. Hasil pengujian dan evaluasi model pada tiga jenis kendaraan dengan menggunakan paket Keras menunjukkan akurasi sebesar 94,4% pada tahap pelatihan dan 73,3% pada tahap pengujian (Fadlia & Kosasih, 2019).

Penelitian serupa juga membahas tentang “Convolution Neural Network (CNN) Untuk Pengklasifikasian Citra Makanan Tradisional” yang ditulis oleh Akhmad Rohim dkk. Hasil penelitian ini menunjukkan bahwa membangun arsitektur model jaringan saraf konvolusional untuk mengklasifikasikan citra makanan tradisional memerlukan empat lapisan konvolusional, empat lapisan max pooling, dan dua lapisan yang terhubung penuh. Arsitektur ini diperoleh karena mencapai presisi sebesar 73%, recall sebesar 69%, dan Fscore sebesar 69% dengan nilai loss minimal sebesar 0.000044 dalam 15 epoch selama proses pembelajaran (Rohim dkk, 2019).

Pada penelitian lain yang membahas tentang algoritma CNN dilakukan oleh Abdul Kholik dengan judul “Klasifikasi menggunakan Convolutional Neural Network (CNN) pada tangkapan layar halaman instagram”. Dalam penelitian ini, peneliti mengklasifikasikan tangkapan layar halaman Instagram ke dalam lima kelas yaitu (kecantikan, keluarga, kebugaran, mode, dan makanan) serta menargetkan total 3.770 data gambar tangkapan layar halaman Instagram. Hasil uji Confusion Matrix pada penelitian ini menunjukkan skor presisi sebesar 91%, skor presisi sebesar 93%, skor recall sebesar 90%, dan skor F1 sebesar 91%. Model yang dihasilkan juga diuji pada data yang berbeda dari dataset, total 25 data gambar screenshot diuji pada model, dan model mampu memprediksi 20 gambar screenshot halaman Instagram (Kholik, 2021).

BAB III

ANALISA DAN KEBUTUHAN SISTEM



A. Desain Sistem




1. Kebutuhan Data

a. Data Input

Data yang digunakan dalam penelitian ini berupa gambar citra yang berjumlah 18 jenis makanan yang diperoleh melalui pada laman berikut universe.roboflow.com/bangkit/indonesian-food-pedsx/dataset, serta 2 jenis makanan yang diperoleh dengan melakukan *scraping* pada website.

Tabel 3.1 Contoh Dataset

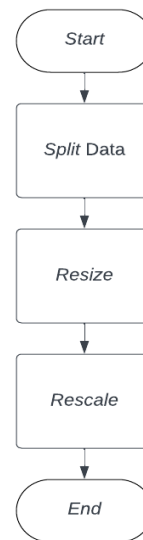
No	Gambar	Kategori Makanan
1		Ayam Betutu
2		Beberuk Terong

No	Gambar	Kategori Makanan
3		Coto Makassar
4		Gudeg
5		Kerak Telor

b. Gambaran Proses

a). *Preprocessing Data*

Hal pertama yang perlu disiapkan adalah dataset agar dapat melanjutkan ke tahap selanjutnya, lakukan pembagian atau *split* untuk step awal berdasarkan label yang telah ditentukan. Kemudian dilakukan proses *resize* yang digunakan untuk menyesuaikan semua ukuran (lebar dan tinggi) gambar agar serupa. Lalu dilakukan proses *rescale* untuk merubah rentang nilai piksel dalam gambar atau citra.

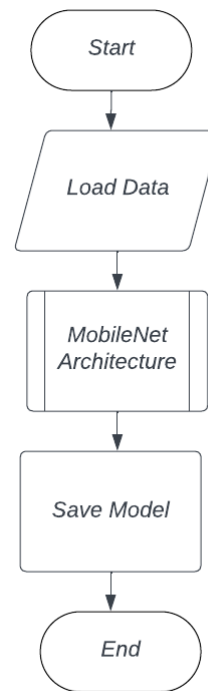


Gambar 3.1 *Preprocessing Data*

b). *Training*

Proses *training data* ini dimulai dengan memuat data yang telah diproses sebelumnya. Data tersebut kemudian diproses menggunakan metode *Convolutional Neural Networks* (CNN). Untuk proses ini dilakukan dengan mengganti beberapa *parameter* yang berada dalam CNN, yaitu Fungsi Aktivasi, Dropout dan Optimasi. Untuk fungsi aktivasi ini akan dilakukan percobaan meliputi *Tanh*, *Softsign*, *Sigmoid*, *Softmax*, *Softplus*, *Linear*, *Relu*, *Elu*, dan *Exponential*. Pada percobaan selanjutnya akan dilakukan perubahan nilai *dropout* dengan nilai 0.1, 0.2, dan 0.3. Percobaan

terakhir akan dilakukan dengan mengganti metode optimasi antara lain adalah *Adamax*, *RMSprop*, *Adam*, *SGD*, *Adagrad*, *Adadelta*.



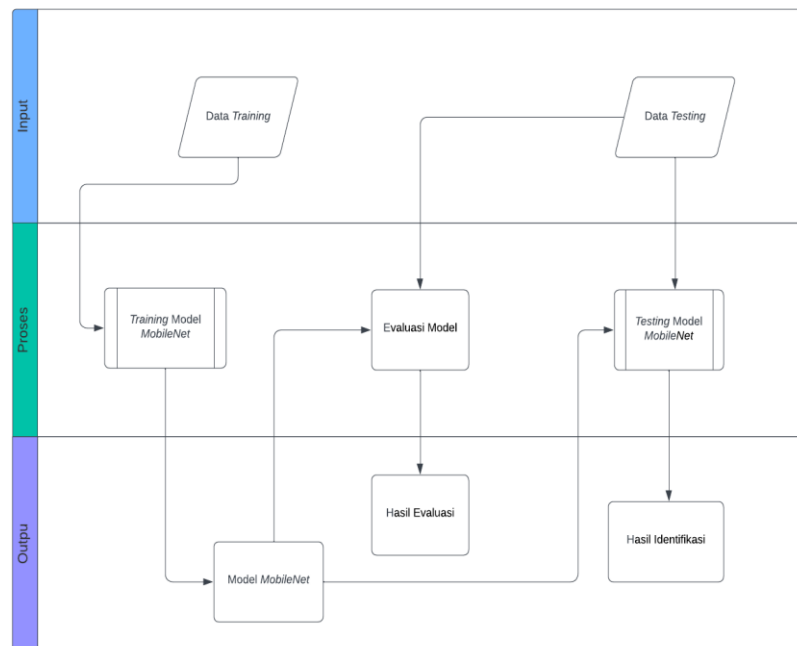
Gambar 3.2 Training

c. Output

Keluaran yang dihasilkan dari proses analisa metode *Convolutional Neural Networks* (CNN) dengan arsitektur MobileNet berupa perbandingan dari tahap *training* yang dilakukan serta hasil dari identifikasi dari foto citra makanan indonesia.

2. Desain Sistem

Sebelum membangun sistem, kita perlu membuat desain yang menunjukkan sistem seperti apa yang akan dibangun. Berikut disajikan desain sistem dari penelitian ini pada gambar (3.3).



Gambar 3.3 Desain Sistem

3. Simulasi Algoritma

Arsitektur *MobileNet* adalah jaringan efisien yang membagi konvolusi menjadi *depthwise convolution* 3×3 dan *convolution* 1×1 *pointwise*, yang secara efektif mengurangi biaya komputasi dan jumlah parameter.

Tabel 3.2 Architecture MobileNet

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32 \text{ dw}$	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64 \text{ dw}$	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128 \text{ dw}$	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128 \text{ dw}$	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256 \text{ dw}$	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256 \text{ dw}$	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5 x Conv dw / s1 Conv / s1	$3 \times 3 \times 512 \text{ dw}$ $1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$ $14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512 \text{ dw}$	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024 \text{ dw}$	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Pada Tabel (3.2) “conv” diartikan standar dari *convolution*, sedangkan “conv dw” merupakan *depthwise separable convolution*. *Stride* disimbolkan dengan “s”, sedangkan “s1” dapat diartikan *stride* dengan ukuran 1×1 , dan “s2” diartikan *stride* dengan ukuran 2×2 .

Pada tabel *filter Shape* yang mempunyai nilai “ $3 \times 3 \times 3 \times 32$ ” terdiri dari *filter* 3×3 (lebar dan tinggi) yang mempunyai 3 saluran (*channel*) yang diterapkan pada *filter* yang berjumlah 32, sedangkan nilai *filter shape* yang mempunyai akhir simbol “dw” dengan contoh nilai “ $3 \times 3 \times 32 \text{ dw}$ ” dapat dinyatakan dalam bentuk *filter depthwise separable*. Lapisan ini mempunyai

32 *filter* yang diterapkan pada setiap saluran input secara terpisah, dan setelah itu dilakukan 1×1 (*pointwise convolutional*) untuk menggabungkan informasi dari semua saluran.

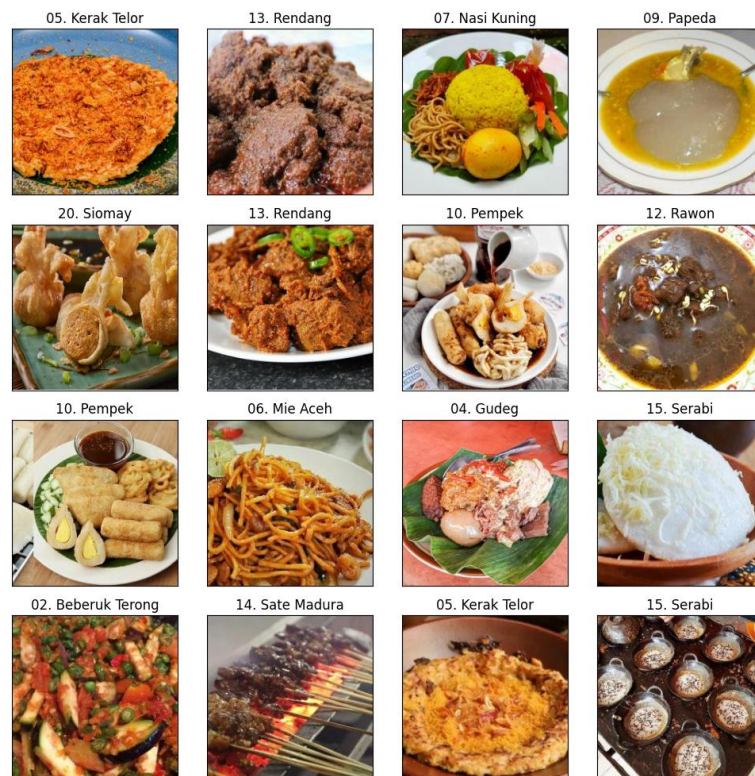
Pada tabel *Input Size* ini merupakan sebuah ukuran citra *input* yang melalui lapisan dari konvolusi ini. Pada simulasi ini ada beberapa *filter* yang diterapkan yaitu *filter* dengan nilai 32, 64, 128, 256, 514, 1024. Setelah dilakukan proses *depthwise separable convolutional*, tahap selanjutnya dilakukan *average pooling* dengan *stride* ukuran 1×1 pada *filter* 7×7 , lalu dilakukan proses *fully connected* dengan *stride* 1×1 , dan diakhiri dengan aktivasi *softmax* untuk hasil klasifikasi dengan lapisan akhir 1000.

BAB IV

IMPLEMENTASI DAN HASIL

A. Preprocessing Data

Setelah dilakukannya *Split* data atau pembagian data untuk data *training*, dan data *testing* maka *preprocessing* data dilakukan. *Preprocessing* ini mencakup *resize* dan *rescale*. Tujuan dari *resize* ini mengubah ukuran gambar menjadi 224x224 piksel, yang bertujuan juga untuk memastikan semua gambar nantinya yang masuk kedalam model mempunyai dimensi yang sama. Sedangkan, pada proses *rescale* ini mengubah nilai piksel gambar dan akan dinormalisasikan dari rentang [0, 255] menjadi rentang [0, 1]. Proses *rescale* ini membantu dalam mempercepat dan membuat stabil pelatihan model dengan memastikan nilai input lebih kecil dan lebih konsisten.



Gambar 4. 1 Data Gambar *resize* dan *rescale*

Tahap berikutnya ada pada data struktur dan pada penelitian ini terdiri dari 20 kelas dengan rentang jumlah gambar antara 30 - 55. Dataset yang sudah melewati tahap *preprocessing* diatas kemudian dibagi menjadi beberapa data yaitu, 80% data *training* dan 20% data *testing*. Berikut pada Tabel 4.1 disajikan dataset dari penelitian ini.

Tabel 4. 1 Dataset Jumlah Gambar

No	Label	Total
1	Ayam Betutu	55
2	Beberuk Terong	40
3	Coto Makassar	49
4	Gudeg	54
5	Kerak Telor	45
6	Mie Aceh	40
7	Nasi Kuning	38
8	Nasi Pecel	50
9	Papeda	55
10	Pempek	55
11	Pueyeum	55
12	Rawon	55
13	Rendang	55
14	Sate Madura	55
15	Serabi	55
16	Soto Banjar	44
17	Soto Lamongan	55
18	Tahu Sumedang	48
19	Nasi Goreng	48
20	Siomay	44

B. Training Model

Setelah tahap sebelumnya diselesaikan yaitu, *preprocessing* data serta pembagian data. Pada bagian ini dilakukan eksperimen untuk menciptakan model yang mempunyai performa paling bagus. Eksperimen akan dilakukan dengan berbagai permainan dalam parameter di dalam *Convolutional Neural Network* yaitu, Fungsi aktivasi dan Pengeluaran (*Dropout*) serta hanya menggunakan Optimasi satu jenis bernama ADAM. Kemudian, pengujian juga dilakukan dengan 2 cara. Cara yang pertama dilakukan untuk melihat hasil sebelum disimpan menjadi model, sedangkan cara yang kedua dilakukan untuk melihat hasil setelah disimpan menjadi model untuk dilihat perbedaannya. Berikut hasil dari pengujian dapat dilihat pada Tabel 4.2 dan Tabel 4.3 dibawah.

Tabel 4. 2 Hasil Pengujian Sebelum Save Model

No	Epoch	Activation	Dropout	Optimation	Accuracy	loss
1	15	Tanh	0.2	adam	88.44%	0.33096
2	30	Tanh	0.2	adam	88.94%	0.3242
3	50	Tanh	0.2	adam	91.96%	0.2691
4	15	Tanh	0.4	adam	88.94%	0.30856
5	30	Tanh	0.4	adam	92.46%	0.26053
6	50	Tanh	0.4	adam	92.46%	0.27405
7	15	Tanh	0.8	adam	75.88%	0.71519
8	30	Tanh	0.8	adam	87.44%	0.39149
9	50	Tanh	0.8	adam	89.45%	0.31793
10	15	Softsign	0.2	adam	90.95%	0.328
11	30	Softsign	0.2	adam	91.96%	0.24757
12	50	Softsign	0.2	adam	89.95%	0.32313
13	15	Softsign	0.4	adam	88.94%	0.32475
14	30	Softsign	0.4	adam	91.96%	0.27879
15	50	Softsign	0.4	adam	91.96%	0.29301

No	Epoch	Activation	Dropout	Optimation	Accuracy	loss
16	15	Softsign	0.8	adam	82.41%	0.64563
17	30	Softsign	0.8	adam	88.44%	0.35535
18	50	Softsign	0.8	adam	92.46%	0.24975
19	15	Softplus	0.2	adam	88.44%	0.39565
20	30	Softplus	0.2	adam	90.95%	0.42715
21	50	Softplus	0.2	adam	90.95%	0.41006
22	15	Softplus	0.4	adam	89.95%	0.32861
23	30	Softplus	0.4	adam	90.95%	0.30562
24	50	Softplus	0.4	adam	90.95%	0.39287
25	15	Softplus	0.8	adam	70.35%	1.73296
26	30	Softplus	0.8	adam	83.42%	0.82116
27	50	Softplus	0.8	adam	89.95%	0.4534
28	15	Relu	0.2	adam	90.95%	0.28003
29	30	Relu	0.2	adam	88.94%	0.35704
30	50	Relu	0.2	adam	89.95%	0.37999
31	15	Relu	0.4	adam	89.45%	0.31737
32	30	Relu	0.4	adam	87.94%	0.36452
33	50	Relu	0.4	adam	90.95%	0.32636
34	15	Relu	0.8	adam	40.70%	2.47293
35	30	Relu	0.8	adam	49.25%	2.05481
36	50	Relu	0.8	adam	66.33%	1.5435
37	15	Sigmoid	0.2	adam	91.96%	0.39328
38	30	Sigmoid	0.2	adam	89.45%	0.3306
39	50	Sigmoid	0.2	adam	90.45%	0.28862
40	15	Sigmoid	0.4	adam	91.96%	0.45046
41	30	Sigmoid	0.4	adam	89.95%	0.32218
42	50	Sigmoid	0.4	adam	91.46%	0.27264
43	15	Sigmoid	0.8	adam	60.30%	1.92

No	Epoch	Activation	Dropout	Optimation	Accuracy	loss
44	30	Sigmoid	0.8	adam	80.90%	1.03097
45	50	Sigmoid	0.8	adam	87.94%	0.54084
46	15	Linear	0.2	adam	87.94%	0.39992
47	30	Linear	0.2	adam	88.94%	0.36718
48	50	Linear	0.2	adam	88.44%	0.43021
49	15	Linear	0.4	adam	85.93%	0.59026
50	30	Linear	0.4	adam	83.92%	0.59862
51	50	Linear	0.4	adam	84.42%	0.95569
52	15	Linear	0.8	adam	85.93%	0.46975
53	30	Linear	0.8	adam	88.94%	0.38278
54	50	Linear	0.8	adam	86.93%	0.53093
55	15	Elu	0.2	adam	88.44%	0.35412
56	30	Elu	0.2	adam	89.95%	0.33281
57	50	Elu	0.2	adam	90.45%	0.33586
58	15	Elu	0.4	adam	85.93%	0.40247
59	30	Elu	0.4	adam	88.94%	0.35035
60	50	Elu	0.4	adam	89.95%	0.41438
61	15	Elu	0.8	adam	79.90%	0.63777
62	30	Elu	0.8	adam	85.93%	0.42224
63	50	Elu	0.8	adam	88.94%	0.29944

Tabel 4. 3 Hasil Pengujian Setelah Save Model

No	Epoch	Activation	Dropout	Optimation	Accuracy	loss
1	30	Tanh	0.4	adam	98.99%	0.05286
2	50	Tanh	0.4	adam	98.99%	0.06318
3	50	Softsign	0.2	adam	98.49%	0.07163
4	50	Softsign	0.8	adam	98.49%	0.08556

No	Epoch	Activation	Dropout	Optimation	Accuracy	loss
5	15	Tanh	0.2	adam	98.49%	0.09048
6	30	Softplus	0.4	adam	97.99%	0.09411
7	15	Softsign	0.2	adam	97.99%	0.09731
8	30	Softsign	0.2	adam	97.99%	0.07405
9	50	Softsign	0.4	adam	97.99%	0.07595
10	30	Tanh	0.2	adam	97.99%	0.07207
11	15	Relu	0.2	adam	97.49%	0.11585
12	30	Relu	0.2	adam	97.49%	0.08073
13	50	Relu	0.4	adam	97.49%	0.0876
14	30	Softplus	0.2	adam	97.49%	0.12186
15	50	Softplus	0.2	adam	97.49%	0.11078
16	15	Softplus	0.4	adam	97.49%	0.0919
17	15	Softsign	0.4	adam	97.49%	0.08593
18	30	Softsign	0.4	adam	97.49%	0.07933
19	50	Tanh	0.2	adam	97.49%	0.08274
20	15	Tanh	0.4	adam	97.49%	0.09041
21	50	Relu	0.2	adam	96.98%	0.14748
22	30	Relu	0.4	adam	96.98%	0.11088
23	50	Softplus	0.4	adam	96.98%	0.18799
24	30	Elu	0.4	adam	96.48%	0.15818
25	50	Elu	0.4	adam	96.48%	0.18445
26	15	Relu	0.4	adam	96.48%	0.14757
27	15	Sigmoid	0.4	adam	96.48%	0.27035
28	15	Elu	0.2	adam	95.98%	0.13143
29	30	Elu	0.2	adam	95.98%	0.15853
30	50	Elu	0.2	adam	95.98%	0.14973
31	15	Elu	0.4	adam	95.98%	0.14101
32	30	Elu	0.8	adam	95.98%	0.16421

No	Epoch	Activation	Dropout	Optimation	Accuracy	loss
33	50	Elu	0.8	adam	95.98%	0.13884
34	15	Linear	0.2	adam	95.98%	0.17226
35	30	Linear	0.2	adam	95.98%	0.1362
36	50	Linear	0.2	adam	95.98%	0.1919
37	30	Linear	0.8	adam	95.98%	0.15827
38	15	Sigmoid	0.2	adam	95.98%	0.20238
39	50	Sigmoid	0.2	adam	95.98%	0.12124
40	50	Sigmoid	0.4	adam	95.98%	0.11189
41	15	Softplus	0.2	adam	95.98%	0.12281
42	30	Linear	0.4	adam	95.48%	0.15482
43	50	Linear	0.4	adam	95.48%	0.31908
44	30	Softsign	0.8	adam	95.48%	0.18172
45	50	Tanh	0.8	adam	95.48%	0.14965
46	15	Linear	0.8	adam	94.97%	0.15669
47	50	Softplus	0.8	adam	94.97%	0.25234
48	50	Linear	0.8	adam	94.47%	0.21624
49	30	Sigmoid	0.2	adam	94.47%	0.14908
50	30	Sigmoid	0.8	adam	94.47%	0.317
51	30	Sigmoid	0.4	adam	93.97%	0.14979
52	15	Linear	0.4	adam	93.47%	0.25679
53	30	Softplus	0.8	adam	93.47%	0.61344
54	50	Sigmoid	0.8	adam	92.96%	0.37982
55	30	Tanh	0.8	adam	92.96%	0.28353
56	15	Elu	0.8	adam	90.45%	0.38964
57	15	Softsign	0.8	adam	90.45%	0.49533
58	15	Tanh	0.8	adam	81.91%	0.62679
59	50	Relu	0.8	adam	75.88%	1.19304
60	15	Softplus	0.8	adam	75.88%	1.6082

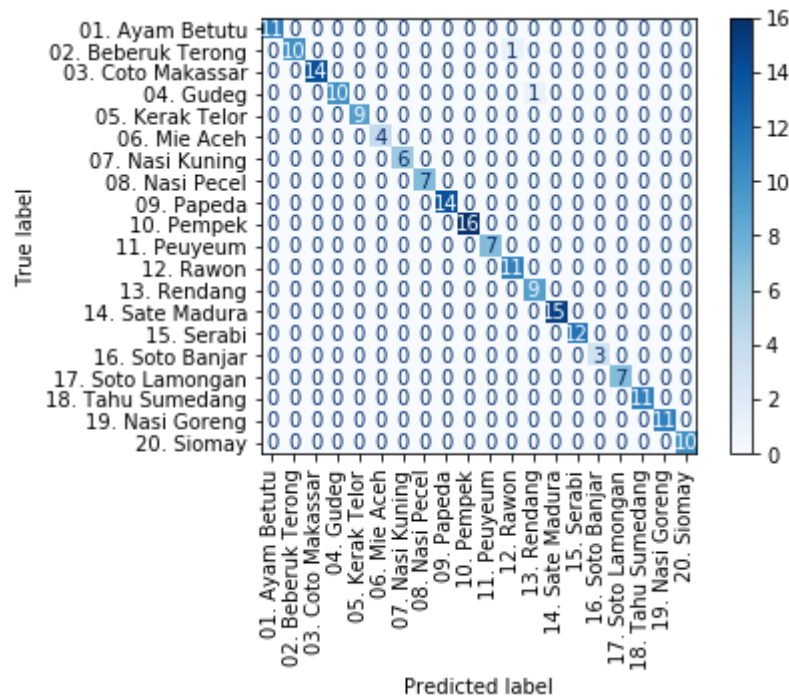
No	Epoch	Activation	Dropout	Optimation	Accuracy	loss
61	15	Sigmoid	0.8	adam	67.34%	1.81054
62	30	Relu	0.8	adam	63.82%	1.57432
63	15	Relu	0.8	adam	56.78%	1.8649

Setelah dilakukan beberapa pengujian dengan bermain dalam parameter serta epoch serta dalam pengujian sebelum model disimpan dan setelah model disimpan, dapat disimpulkan performa terbaik bisa dilihat dengan menggunakan *epoch* 30 dengan fungsi aktivasi *Tanh* dan *dropout* sebesar 0.4 yang diberikan optimasi Adam mendapatkan accuracy 98.99% dan loss 0.05286.

C. Testing Model

Pengujian atau *testing* dilakukan dengan menggunakan 3 macam *epoch* (15, 30, 50), fungsi aktivasi (*Tanh*, *Softsign*, *Softplus*, *Relu*, *Sigmoid*, *Linear*, *Elu*), *Dropout* sebesar (0.2, 0.4, 0.8) dan hanya menggunakan optimasi Adam. Pengujian juga dapat dilakukan setelah model disimpan karena dapat dilihat pada Tabel 4.2 dan Tabel 4.3 terdapat cukup perbedaan dalam accuracy dan loss serta pastinya model yang dieksekusi nantinya juga pasti model yang telah disimpan.

Setelah dilakukan training model dari proses diatas maka dilanjutkan untuk *testing* model atau pengujian model dan telah ditemukan untuk menggunakan model terbaik dari hasilnya. Dan dari hasil pengujian yang telah dilakukan didapatkan confusion matrix seperti pada Gambar 4.2 berikut.



Gambar 4. 2 Confussion Matrix

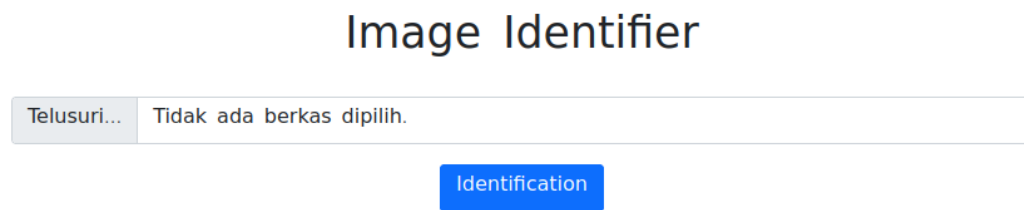
Setelah didapatkan *confussion matrix* pada Gambar 4.2 diatas, didapatkan hasil perhitungan *accuracy*, *preciission*, *recall*, dan *f1-score* pada Gambar 4.3 berikut.

	precision	recall	f1-score	support
01. Ayam Betutu	1.000000	1.000000	1.000000	11.000000
02. Beberuk Terong	1.000000	0.909091	0.952381	11.000000
03. Coto Makassar	1.000000	1.000000	1.000000	14.000000
04. Gudeg	1.000000	0.909091	0.952381	11.000000
05. Kerak Telor	1.000000	1.000000	1.000000	9.000000
06. Mie Aceh	1.000000	1.000000	1.000000	4.000000
07. Nasi Kuning	1.000000	1.000000	1.000000	6.000000
08. Nasi Pecel	1.000000	1.000000	1.000000	7.000000
09. Papeda	1.000000	1.000000	1.000000	14.000000
10. Pempek	1.000000	1.000000	1.000000	16.000000
11. Peuyeum	1.000000	1.000000	1.000000	7.000000
12. Rawon	0.916667	1.000000	0.956522	11.000000
13. Rendang	0.900000	1.000000	0.947368	9.000000
14. Sate Madura	1.000000	1.000000	1.000000	15.000000
15. Serabi	1.000000	1.000000	1.000000	12.000000
16. Soto Banjar	1.000000	1.000000	1.000000	3.000000
17. Soto Lamongan	1.000000	1.000000	1.000000	7.000000
18. Tahu Sumedang	1.000000	1.000000	1.000000	11.000000
19. Nasi Goreng	1.000000	1.000000	1.000000	11.000000
20. Siomay	1.000000	1.000000	1.000000	10.000000
accuracy	0.989950	0.989950	0.989950	0.989950
macro avg	0.990833	0.990909	0.990433	199.000000
weighted avg	0.990871	0.989950	0.989952	199.000000

Gambar 4.3 *Report Classification*

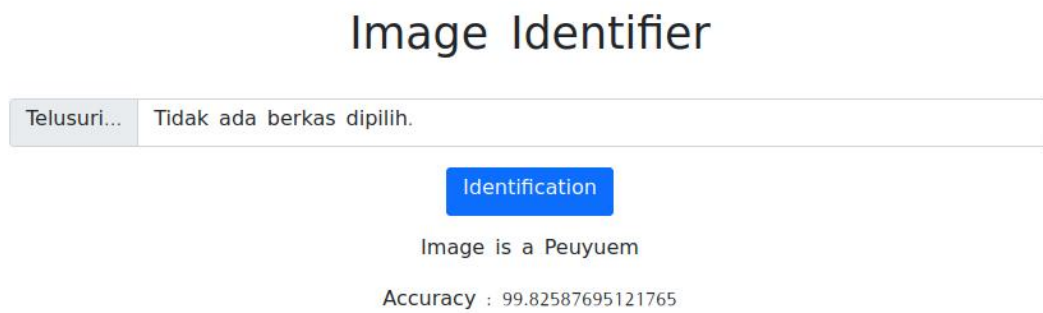
D. Identifikasi Gambar dengan Website

Pada tahap identifikasi ini, peneliti membuat sebuah aplikasi berbasis website yang dapat terintegrasi pada model yang telah dibuat pada tahap tahap sebelumnya. Peneliti menggunakan kerangka kerja dari python yaitu, Flask untuk membangun website ini. Website ini hanya mempunyai satu halaman yaitu, *main page* (halaman utama). Berikut peneliti sajikan Gambar 4.4 yang merupakan tampilan dari *main page* website ini.



Gambar 4. 4 Main page

Pada halaman utama tersebut user dapat langsung bisa menekan sebuah teks “Telusuri” atau bisa pada *textbox* “Tidak ada berkas dipilih”. Setelah itu user akan langsung diarahkan pada penyimpanan lokal, dan user diberikan untuk memilih gambar yang ada dipenyimpanan lokal untuk dilakukan identifikasi. Setelah gambar telah dipilih user bisa langsung saja menekan tombol “*Predict Image*”. Tahap berikutnya sistem akan menerima gambar dari user dan akan memprosesnya menggunakan *Flask* dan model yang telah dilatih sebelumnya. Ketika proses identifikasi selesai user akan diberikan hasil dari prediksi serta berapa akurasi yang didapatkan dari identifikasi tersebut seperti pada Gambar 4.5 dibawah.



Gambar 4. 5 Hasil dari Identifikasi

BAB V

PENUTUP

A. Kesimpulan

Dari hasil penelitian dan implementasi yang telah dilakukan, terdapat beberapa kesimpulan yang dapat diambil:

1. Arsitektur *MobileNet* dari *Convolutional Neural Network* sangat bagus dalam mengenali foto makanan yang mempunyai object sangat banyak. Hal ini dapat dilihat dari beberapa pengujian dan didalam pengujian melibatkan banyak parameter yang telah disediakan oleh *Convolutional Neural Network*.
2. Pengujian dalam klasifikasi ini sangat penting dilakukan untuk menciptakan sebuah model yang bagus, maka dari itu penelitian ini menekankan pada uji daya parameter yang telah ada.

B. Saran

Berdasarkan hasil evaluasi dan kesimpulan yang telah dilakukan oleh peneliti, ada beberapa saran yang dapat diberikan oleh peneliti untuk pengembangan selanjutnya:

1. Peningkatan volume data: dikarenakan banyak sekali macam makanan di negara kita yaitu Indonesia, akan lebih baik lagi apabila terdapat lebih banyak jenis makanan yang bisa dimasukkan kedalam penelitian selanjutnya.
2. Penggunaan parameter lain: Dikarenakan peneliti saat ini menekankan menggunakan parameter dropout, activation maka diharapkan untuk penelitian selanjutnya bisa untuk bisa membawa parameter parameter lain yang berada dalam *Convolutional Neural Network*.
3. Pemanfaatan model: Peneliti mengharapkan penelitian ini bisa dikembangkan dengan menggunakan metode lain, dengan memanfaatkan hasil dari penelitian ini.

DAFTAR PUSTAKA

- Asaf M.M, D. A. S. (2020, 01). UPAYA PEMENUHAN KEBUTUHAN DASAR MANUSIA, (Jurnal Cakrawarti, Vol. 02. No. 02).
- Dwi R, A. A. (2023, September 27). Kenapa Orang-orang Suka Mengunggah Foto Makanan di Media Sosial? detikFood. Retrieved January 14, 2024, from <https://food.detik.com/info-kuliner/d-6951794/kenapa-orang-orang-suka-mengunggah-foto-makanan-di-media-sosial>
- Fadli, A., Ramadhani, Y., & Aliim, M. S. (2021). Purwarupa Sistem Deteksi COVID-19 Berbasis Website Menggunakan Algoritma Convolutional Neural Network, (Vol. 5 No. 5), 876 – 883.
- Fadlia, N., & Kosasih, R. (2019). KLASIFIKASI JENIS KENDARAAN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN), Vol 24, No 3. <http://dx.doi.org/10.35760/tr.2019.v24i3.2397>
- Felix, Faisal, S., Butarbutar, T. F. M., & Sirait, P. (2019, Oktober 25). Implementasi CNN dan SVM untuk Identifikasi Penyakit Tomat via Daun. (VOL 20, NO 2). <https://doi.org/10.55601/jsm.v20i2.670>
- Gunawan, I. (2020, 12 31). Optimasi Model Artificial Neural Network Untuk Klasifikasi Paket Jaringan, 14. <https://doi.org/10.51901/simetris.v14i2.135>
- Hartono, I., Noertjahyana, A., & Santoso, L. W. (2022). Deteksi Masker Wajah dengan Metode Convolutional Neural Network, (Vol 10, No 1).
- Hendriyana, & Maulana, a. H. (2020, Februari 02). Identifikasi Jenis Kayu menggunakan Convolutional Neural Network dengan Arsitektur Mobilenet. (Vol 4 No 1 (2020): Februari 2020), 70 - 76. <https://doi.org/10.29207/resti.v4i1.1445>
- Hendriyana, & Maulana, Y. H. (2020). Identifikasi Jenis Kayu menggunakan Convolutional Neural Network, 4(vol. 4, no. 1, pp. 70–76).
- Iswantoro, D., & UN, D. H. (2022, Juli). Klasifikasi Penyakit Tanaman Jagung Menggunakan Metode Convolutional Neural Network (CNN). urnal Ilmiah Universitas Batanghari Jambi,, (Vol 22, No 2 (2022)), 900-905. <http://dx.doi.org/10.33087/jiubj.v22i2.2065>

- Kholik, A. (2021). KLASIFIKASI MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) PADA TANGKAPAN LAYAR HALAMAN INSTAGRAM, (JDMSI , Vol.2 , No.2), 10 - 20.
- Kurnia, D. A., Setiawan, A., Amalia, D. R., Arifin, R. W., & Setiyadi, D. (2020, 06 21). Image Processing Identifacation for Indonesian Cake Cuisine using CNN Classification Technique, 1783. 10.1088/1742-6596/1783/1/012047
- Mahaputri, C., & Hari Wisana, I. D. G. (2022, 12 21). Introductional to Traditional Archipelago Foods using The Cpnvolutional Neural Network (CNN), (Vol 1 No 1 (2022): Desember).
- Maulana, F. F., & Rochmawati, N. (2019, 01 01). Klasifikasi Citra Buah Menggunakan Convolutional Neural Network, 01(Vol 1 No 02 (2019)), 02. <https://doi.org/10.26740/jinacs.v1n02.p104-108>
- Putra, C. M., Triayudi, A., & Ningsih, S. (2023, 05 30). Face Mask Recognition Menggunakan Model CNN (Convolutional Neural Network) Berbasis Python dan OpenCV, 4(Vol 4 No 3 (2023): May 2023), 722-730. <https://doi.org/10.47065/josyc.v4i3.3532>
- Rohim, A., Sari, Y. A., & Tibyani. (2019, July). Convolution Neural Network(CNN) Untuk Pengklasifikasian Citra Makanan Tradisional, (Vol. 3, No. 7), 7037-7042.
- Sandi, K. M., Yudha, A. P., Aryanto, N. D., & Farabi, M. A. (2022, Juli 31). Klasifikasi sampah menggunakan Convolutional Neural Network. Indonesian Journal of Data and Science (IJODAS), (Vol 3, No 2), 72 - 81. <https://doi.org/10.56705/ijodas.v3i2.33>
- Simarmata, S. Y. E., Sari, Y. A., & Adinugroho, S. (2019). Klasifikasi Citra Makanan Menggunakan Algoritme Learning Vector Quantization Berdasarkan Ekstraksi Fitur Color Histogram dan Gray Level Co-occurrence Matrix. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 3(3), 2369-2378.
- Suradiradja, K. H. (2021, 12). Algoritme Machine Learning Multi-Layer Perceptron dan Recurrent Neural Network untuk Prediksi Harga Cabai Merah Besar di Kota Tangerang, 14, 4. <https://doi.org/10.30998/faktorexacta.v14i4.10376>

- Trivusi. (2022, July 28). Pengertian dan Cara Kerja Algoritma Convolutional Neural Network (CNN). Trivusi. Retrieved December 27, 2023, from <https://www.trivusi.web.id/2022/04/algoritma-cnn.html>
- Umer, A. (2023, May 28). Understanding Convolutional Neural Networks : A Beginner's Journey into the Architecture. Medium. Retrieved January 14, 2024, from <https://medium.com/codex/understanding-convolutional-neural-networks-a-beginners-journey-into-the-architecture-aab30dface10>
- Wibawa, A. P., Aji Purnama, M. G., Akbar, M. F., & Dwiyanto, F. A. (2018, 3). Metode-metode Klasifikasi. Metode-metode Klasifikasi, 3(Vol. 3, No. 1, Maret 2018), 1.
- Wulandari, I., Yasin, H., & Widiharih, T. (2020). KLASIFIKASI CITRA DIGITAL BUMBU DAN REMPAH DENGAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN), (JURNAL GAUSSIAN, Volume 9, Nomor 3), 273 - 282.