

Klasifikasi Jajanan Khas Bali Untuk Preservasi Pengetahuan Kuliner Lokal Menggunakan Arsitektur VGG-16

Ni Luh Widi Rahayu¹, Nyoman Gunantara², Made Sudarma³

^{1,2,3}Program Studi Magister Teknik Elektro Fakultas Teknik Universitas Udayana,
Jalan P.B. Sudirman Denpasar-Bali, Indonesia

email : luhwidirahayu@gmail.com¹, gunantara@unud.ac.id², msudarma@unud.ac.id³

Received : November, 2023

Accepted : December, 2023

Published : April, 2024

Abstract

The use of Deep Learning technology, especially image recognition, is an appropriate for classifying digital images. This research aims to classify images of typical Balinese snacks by identifying the optimal model within the VGG-16 architecture. The evaluation involves comparing accuracy, recall, precision, and f1-score across different test scenarios, including dropout weight, batch sizes, and epochs. The training dataset for this research comprises 2,445 image data points of typical Balinese snacks, distributed among 10 classes. These classes include Klepon (320 images), aLaklak (207 images), Kaliadrem (222 images), Jaje Lukis (327 images), Jaje Batun Bedil (189 images), Pisang Rai (200 images), Jaje Piling-piling (234 images), Jaje Wajik (241 images), Ongol-ongol (308 images), and Bubur Injin (197 images). Each class is represented by 50 image datas, resulting in a total of 500 image datas per class. The best model derived from the VGG-16 architecture for classifying typical Balinese snacks achieved an accuracy level of 97.5%, precision of 87.9%, recall of 87%, and an f1-score of 87.4%. This performance was achieved with a dropout test parameter of 20%, a batch size of 64, and an epoch setting of 1000. The evaluation was conducted on test image data separate from the training and validation datasets.

Keywords: Clasifications; Traditional Balinese Snacks; Deep Learning; CNN; VGG-16

Abstrak

Pemanfaatan teknologi Deep Learning khususnya teknologi pengenalan gambar merupakan suatu media yang tepat digunakan untuk melakukan klasifikasi citra digital. Pada penelitian ini bertujuan untuk melakukan klasifikasi citra jajanan khas Bali dengan mencari model terbaik dari arsitektur VGG-16 dengan melakukan perbandingan tingkat akurasi, recall, precision dan f1-score dari skenario pengujian yaitu bobot dropout, jumlah batch size dan jumlah epoch yang digunakan. Dataset training yang digunakan dalam penelitian ini sebanyak 2.445 data citra jajanan khas Bali dengan total kelas sebanyak 10 kelas yaitu Klepon 320 citra, Laklak 207 citra, Kaliadrem 222 citra, Jaje Lukis 327 citra, Jaje Batun Bedil 189 citra, Pisang Rai 200 citra, Jaje Piling-piling 234 citra, Jaje Wajik 241 citra, Ongol-ongol 308 citra dan Bubur Injin 197 citra. Dataset sebanyak 50 data citra jajanan khas Bali pada setiap kelasnya dengan jumlah total 500 data citra. Model terbaik yang didapatkan dari arsitektur VGG-16 dalam melakukan klasifikasi jajanan khas Bali yaitu dengan tingkat akurasi sebesar 97,5%, presisi 87,9%, recall 87% serta f1-score sebesar 87,4% dengan parameter pengujian dropout 20%, batch size 64 serta epoch 1000 pada data citra uji diluar data pelatihan dan validasi.

Kata Kunci: Klasifikasi; Jajanan Khas Bali; Deep Learning; CNN; VGG-16

1. PENDAHULUAN

Pangan merupakan sebuah kebutuhan pokok yang harus selalu ada dalam setiap aspek kehidupan manusia. Pangan bukan hanya sebagai objek simbolik yang dipahami untuk menopang kehidupan atau sekadar untuk menghilangkan rasa lapar, tetapi pangan juga merupakan identitas dan daya tarik wisata di suatu wilayah. Di Indonesia hampir setiap daerah mempunyai makanan dan jajanan tradisional, dimana makanan dan jajanan tradisional adalah makanan yang diproduksi atau dikonsumsi secara turun temurun dengan menggunakan bahan-bahan lokal yang disiapkan secara khusus di suatu daerah. Setiap daerah pasti mempunyai makanan khas daerah[1] yang diwariskan secara turun temurun, diantaranya makanan dan jajanan tradisional Bali. Tak hanya sekadar santapan harian, makanan dan jajanan tradisional Bali juga mencakup warisan budaya, unsur penyembuhan, dan sarana ekspresi adat dan tradisi. Saat ini, terlihat tren di mana penduduk lokal maupun wisatawan lebih memilih makanan alami, termasuk makanan tradisional, karena dianggap lebih sehat dan memiliki resiko gangguan kesehatan yang lebih rendah jika dibandingkan dengan makanan modern yang tinggi lemak dan gula[2][3].

Makanan dan jajanan tradisional Bali penting untuk memperkuat jati diri bangsa. Namun, dalam penelitian sebelumnya menyatakan bahwa makanan dan jajanan tradisional Bali sebagai warisan budaya lambat laun akan memudar dan tersingkir[4]. Di tengah arus globalisasi, makanan tradisional Bali bersaing dengan hidangan modern dari berbagai negara. Terkadang, generasi muda lebih tertarik pada makanan asing daripada makanan tradisional Bali. Selain itu, pengetahuan masyarakat Bali tentang makanan tradisional mereka sendiri relatif rendah, yang dibuktikan dengan hasil penelitian yang dilakukan di salah satu kecamatan di Kabupaten Bangli, yaitu sebesar 56,7% masyarakat tidak mengetahui keberagaman makanan tradisional Bali[4]. Selain itu, berdasarkan hasil survey yang telah dilakukan pada penelitian sebelumnya dengan menyebarkan kuesioner kepada 20 responden berusia 17-25 tahun didapatkan hasil bahwa kalangan muda belum mengetahui informasi tentang jajanan tradisional khas Bali[3]. Saat ini, data mengenai jumlah dan jenis makanan

tradisional Bali belum dapat dipastikan. Berdasarkan studi tahun 1999 oleh Pusat Penelitian Makanan Tradisional (PKMT) Universitas Udayana ada sebanyak 281 jenis makanan utama, 174 jajanan dan 73 minuman tradisional di Bali[4]. Pencarian *dataset* tentang jajanan tradisional Bali juga masih sedikit dan tidak mudah didapatkan di media sosial sehingga harus menciptakan *dataset* itu sendiri. Hal ini menunjukkan bahwa perlu adanya upaya yang lebih besar untuk mengenalkan dan melestarikan kekayaan kuliner tradisional Bali. Oleh karena itu, dilakukannya penelitian ini bertujuan untuk mendapatkan model terbaik yang mampu mengenali dan mengklasifikasi citra jajanan khas Bali dengan memanfaatkan teknologi *Artificial Intelligence*, khususnya *deep learning* yang nantinya dapat diimplementasikan pada sebuah sistem informasi.

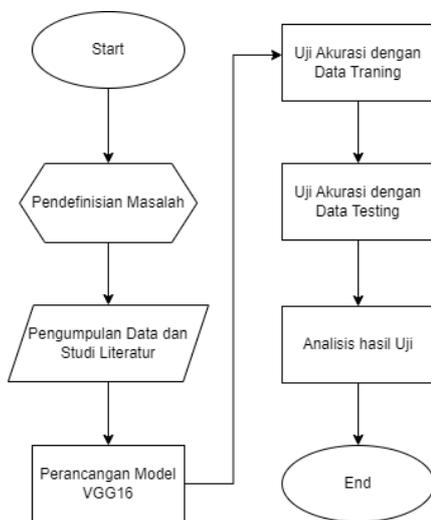
Berdasarkan penelitian sebelumnya yang telah dilakukan didapatkan hasil akurasi yang tinggi dalam pengenalan citra digital menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur VGG-16 dimana sebanyak 10.000 data citra makanan nasional digunakan sebagai data *training* dan untuk menguji akurasi dengan memfokuskan pengujian pada 20 kelas citra makanan dengan hasil akurasi terendah sebesar 66% dan akurasi tertinggi mencapai 98%[10]. Selain itu, penelitian serupa yang membahas mengenai klasifikasi telah dilakukan dengan melakukan klasifikasi makanan khas Thailand menggunakan VGG-16 menghasilkan tingkat akurasi rata-rata klasifikasi sebesar 88,33% pada dataset TFF yang berisi 3.960 gambar[11]. Oleh karena itu, dalam penelitian ini akan menggunakan metode *Convolutional Neural Network* (CNN) yang merupakan salah satu algoritma *deep learning* yang sangat efektif dalam melakukan klasifikasi gambar dari jumlah data yang banyak dan dapat memperoleh nilai akurasi yang tinggi[5][6][7][8][9].

Dalam melakukan pengenalan dan klasifikasi jajanan tradisional pada penelitian ini akan menggunakan model dari arsitektur VGG-16, dimana dalam penelitian ini akan menggunakan *dropout* pada layer arsitektur VGG-16 untuk mencegah terjadinya *overfitting* pada proses *training* maupun *testing*[12][13][14][15], selain itu *preprocessing* yang meliputi *resize* dan *zoom image* dan *flip image* juga akan dilakukan. Jumlah *dataset* yang akan digunakan pada

penelitian ini sebanyak 2.945 data citra dengan 10 kelas jajanan khas Bali. Skenario pengujian dengan melakukan perbandingan tingkat akurasi, *recall*, *precision*, dan *f1-score* yang didapat antara jumlah *batch size*, jumlah *dropout* dan jumlah *epoch* tiap model. Hasil evaluasi dari skenario pengujian tersebut akan mendapatkan perbandingan hasil yang berbeda pada tiap modelnya dalam melakukan pengenalan dan klasifikasi jajanan tradisional, sehingga dapat diketahui kinerja arsitektur VGG-16 dalam melakukan klasifikasi data citra jajanan khas Bali.

2. METODE PENELITIAN

Dalam tahap ini akan dipaparkan alur dari penelitian yang dirancang oleh peneliti untuk mendapatkan model terbaik dari arsitektur VGG-16 yang mampu mengklasifikasi jajanan khas Bali, yang meliputi pendefinisian masalah, pengumpulan data dan studi literatur, perancangan model VGG-16, uji akurasi dengan data *training*, uji akurasi dengan data *testing* dan analisis hasil uji. Adapun *flowchart* dari alur penelitian ini dapat dilihat pada gambar 1 berikut.



Gambar 1. *Flowchart* Alur Penelitian Klasifikasi Jajanan khas Bali

Sesuai dengan gambar 1 secara umum alur penelitian dimulai dengan pendefinisian masalah, kemudian dilanjutkan dengan melakukan pengumpulan data citra yang akan dijadikan dataset *training* dan *testing*, studi literatur dilakukan dari berbagai sumber pustaka untuk mendapatkan informasi tentang jajanan khas Bali, metode Konvolusi, *Deep*

Learning, *Convolutional Neural Network* (CNN), arsitektur VGG-16 serta literatur yang berhubungan dengan klasifikasi jajanan khas Bali sebagai dasar pemecahan masalah yang telah teridentifikasi pada tahap sebelumnya, kemudian akan dilakukan perancangan model VGG-16, uji akurasi dengan data *training* dilakukan untuk melihat model yang dirancang dapat mengenali dan melakukan klasifikasi pada data latih, uji akurasi dengan data *testing* dilakukan untuk melihat model yang dirancang dapat mengenali dan melakukan klasifikasi pada data uji yaitu data baru diluar data pelatihan, tahap terakhir akan dilakukan analisis hasil uji dan mencari model terbaik berdasarkan perbandingan dari parameter uji yang sudah ditetapkan.

2.1 Pengumpulan Dataset

Pada penelitian ini, pengembangan model CNN dengan arsitektur VGG-16 menggunakan dataset berupa data citra Jajanan khas Bali. Dataset yang digunakan merupakan citra digital yang diambil melalui *google image* dan data citra jajanan khas Bali yang diambil secara langsung menggunakan kamera handphone Iphone XR 128GB dengan spesifikasi kamera 12 MP Wide yang digunakan sebagai dataset *training* dan *testing*, dimana jumlah dataset *training* yang akan digunakan pada penelitian ini akan dibagi menjadi 2 yaitu data *train* dan data *validation* dengan perbandingan 80 : 20 yaitu sebanyak 2.945 data citra dengan 10 kelas jajanan khas Bali yaitu Klepon 370 citra, Lalkak 257 citra, Kaliadrem 272 citra, Jaje Lukis 377 citra, Jaje Batun Bedil 239 citra, Pisang Rai 250 citra, Jaje Piling-piling 284 citra, Jaje Wajik 291 citra, Ongol-ongol 358 citra dan Bubur Injin 247 citra.

Selain itu, data uji yaitu data citra baru diluar data pelatihan juga digunakan pada penelitian ini yang bertujuan untuk menguji performa tiap model yang didapat sebanyak 10 citra pada tiap kelasnya dengan jumlah total keseluruhan sebanyak 100 data citra jajanan khas Bali. Ukuran data citra yang akan digunakan sebagai dataset *training* masih berukuran sembarang yang nantinya akan dilakukan *resize* sesuai dengan ukuran data input dari model VGG-16 pada proses *preprocessing*. Adapun contoh data citra jajanan khas Bali yang diambil melalui *google image* dan citra yang diambil secara langsung menggunakan kamera handphone dapat dilihat pada gambar 2 berikut.

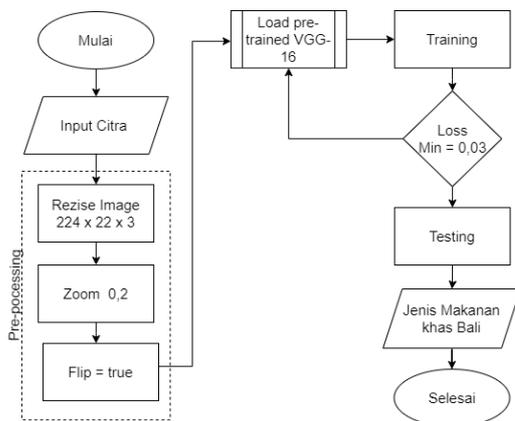


Gambar 2. Contoh Citra Dataset

Gambar 2 merupakan beberapa contoh citra jajanan khas Bali yang akan digunakan sebagai dataset *training* dan dataset *testing*, dengan urutan batun bedil, bubur injin, jaje lukis, piling-piing, jaje wajik, kaliadrem, klepon, lakkak dan ongol-ongol.

2.2 Alur Pemodelan Klasifikasi

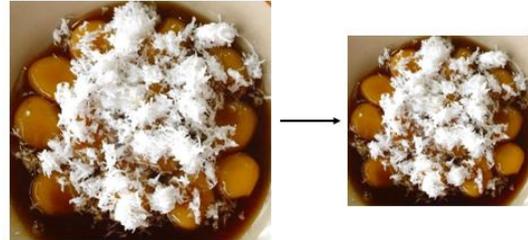
Perancangan model klasifikasi jajanan khas Bali ini disusun berdasarkan beberapa tahapan yang dapat dilihat pada gambar 3 berikut.



Gambar 3. Alur Pemodelan Klasifikasi

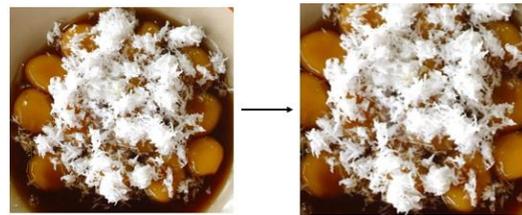
Gambar 3 merupakan alur pemodelan klasifikasi yang akan dimulai dengan proses penginputan citra digital dengan ukuran sembarang atau

ukuran asli. Selanjutnya akan masuk pada tahap *preprocessing* yaitu *resize* dimana proses ini dilakukan untuk mengubah ukuran citra inputan menjadi 224x224 sesuai dengan ukuran inputan dari VGG-16 yang dapat dilihat pada gambar 4 berikut.



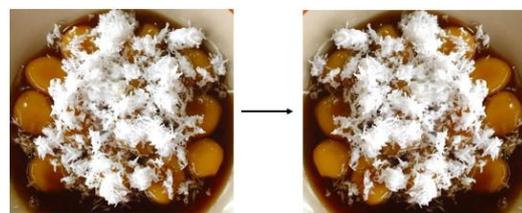
Gambar 4. Contoh Resize Image

Selain itu zoom 0,2 atau 20% juga dilakukan pada tahap *preprocessing*, dimana zoom adalah teknik augmentasi dengan melakukan perbesaran gambar sesuai dengan nilai yang digunakan. Hasil zoom dapat dilihat pada gambar 5 berikut.



Gambar 5. Contoh Zoom Image

Selanjutnya adalah tahap *flip image* dimana citra inputan akan dimiringkan akan di flip secara *horizontal* dan dapat dilihat pada gambar 6 berikut.



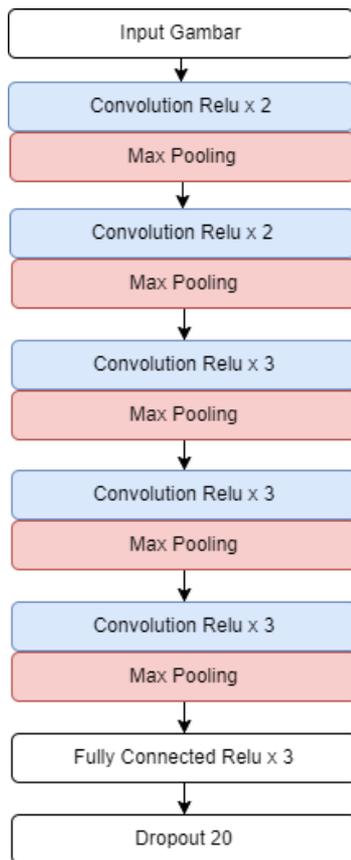
Gambar 6. Contoh Flip Horizontal

Setelah tahap *preprocessing* maka akan dilanjutkan dengan perancangan model *Convolutional Neural Network* (CNN) dengan arsitektur VGG-16 yang akan digunakan pada proses *training* dengan batas kesalahan minimum 0,03 atau 3%. Jika model yang dibangun sudah dapat mencapai batas

minimum kesalahan tersebut maka akan dilanjutkan ke proses *testing* dengan data uji baru diluar data latih.

2.2 Arsitektur VGG-16

VGG16 adalah sebuah arsitektur deep learning yang memiliki 16 lapisan yaitu 13 lapisan konvolusional dan 3 lapisan terhubung penuh (fully connected layer)[15][16]. Adapun pemodelan VGG16 dapat dilihat seperti pada gambar 7 berikut.



Gambar 7. Model VGG-16

VGG-16 menggunakan konsep blok untuk membentuk lapisan konvolusional, setiap lapisan konvolusionalnya memiliki ukuran 3 x 3 dan stride 1 dan aktivasi relu yang berfungsi untuk mencegah nilai negatif tidak diteruskan ke lapisan berikutnya[17]. Pada akhir blok, digunakan lapisan penggabungan maksimal (max pooling layer) dengan ukuran 2 x 2 dan stride 2. Selain itu layer fully connected dapat berisikan dropout yang berfungsi untuk mengurangi overfitting dan mempercepat proses learning pada model yang dibangun[18][19].

Model CNN yang dirancang dalam penelitian ini menggunakan arsitektur VGG-16 yang telah dilatih sebelumnya (pre-trained) pada dataset ImageNet[20], selain itu arsitektur VGG-16 yang dikembangkan pada penelitian ini, menggunakan semua layer *convolution* yang ada serta menggunakan *dropout* sebesar 20%. Adapun hasil load model dari arsitektur VGG-16 pada dataset jajanan khas Bali dapat dilihat pada gambar 8 berikut.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool1 (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool1 (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590880
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590880
block3_pool1 (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool1 (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool1 (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 512)	12845568
dense_1 (Dense)	(None, 256)	131328
dense_2 (Dense)	(None, 64)	16448
dropout (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 10)	650

Total params: 27708682 (105.70 MB)
 Trainable params: 12993994 (49.57 MB)
 Non-trainable params: 14714688 (56.13 MB)

Gambar 8. Hasil Load Model VGG-16

2.3 Training

Proses *training* yang dilakukan pada model arsitektur VGG-16 yang telah dilatih sebelumnya (*pre-trained*) dengan dataset besar yaitu *ImageNet*. Penggunaan model VGG-16 yang sudah pernah dilatih sebelumnya (*pre-trained*) pada dataset besar yaitu *ImageNet* bertujuan agar model yang dilatih untuk melakukan

klasifikasi jajan khas Bali tidak melakukan pelatihan dari awal, sehingga akan menghasilkan akurasi yang baik serta menghemat waktu pelatihan. Pelatihan ulang dilakukan pada dataset jajan khas Bali untuk menyesuaikan hasil prediksi menggunakan parameter *batch size* sebesar 32 dan 64, bobot dropout 20% dan jumlah *epoch* sebanyak 500 dan 1000. Selain itu, parameter *Optimizer* yaitu "adam" dengan *learning rate* 0,001 juga digunakan dalam penelitian ini. Adapun skenario proses *training* yang akan dilakukan dalam penelitian ini untuk mendapatkan model terbaik dapat dilihat pada tabel 1 berikut.

Tabel 1. Skenario Proses Training

Parameter Pengujian	
Input Citra	224x224
Jumlah Epoch	500 dan 1000
Batch Size	32 dan 64
Bobot Dropout	0 dan 0,2
Learning Rate	0,001

Pada tahap training akan melakukan pelatihan terhadap model dengan parameter yang berbeda untuk setiap modelnya. Hasil dari tiap model yang sudah dilatih pada tahap ini nantinya akan digunakan sebagai model uji pada data baru diluar data pelatihan. Proses pelatihan menggunakan *IDE Jupyter Notebook* pada *Google Collaboratory* dengan *runtime type* A100 GPU *High-RAM*. Adapun hasil dari proses pelatihan model yang dilakukan pada penelitian ini dapat dilihat pada gambar 9 dan 10 berikut.

```
Epoch 491/500
74/74 [=====] - 602s 8s/step - loss: 0.0317 - accuracy: 0.9906 - val_loss: 0.0357 - val_accuracy: 0.9900
Epoch 492/500
74/74 [=====] - 601s 8s/step - loss: 0.1196 - accuracy: 0.9646 - val_loss: 0.1326 - val_accuracy: 0.9520
Epoch 493/500
74/74 [=====] - 607s 8s/step - loss: 0.1120 - accuracy: 0.9697 - val_loss: 0.1227 - val_accuracy: 0.9580
Epoch 494/500
74/74 [=====] - 592s 8s/step - loss: 0.0903 - accuracy: 0.9642 - val_loss: 0.0276 - val_accuracy: 0.9920
Epoch 495/500
74/74 [=====] - 589s 8s/step - loss: 0.0676 - accuracy: 0.9602 - val_loss: 0.0039 - val_accuracy: 0.9980
Epoch 496/500
74/74 [=====] - 593s 8s/step - loss: 0.0359 - accuracy: 0.9572 - val_loss: 0.0085 - val_accuracy: 1.0000
Epoch 497/500
74/74 [=====] - 594s 8s/step - loss: 0.0594 - accuracy: 0.9766 - val_loss: 0.0091 - val_accuracy: 0.9980
Epoch 498/500
74/74 [=====] - 593s 8s/step - loss: 0.0594 - accuracy: 0.9642 - val_loss: 2.8998e-04 - val_accuracy: 1.0000
Epoch 499/500
74/74 [=====] - 595s 8s/step - loss: 0.0359 - accuracy: 0.9642 - val_loss: 2.9972e-04 - val_accuracy: 1.0000
Epoch 500/500
74/74 [=====] - 595s 8s/step - loss: 0.0359 - accuracy: 0.9642 - val_loss: 2.9972e-04 - val_accuracy: 1.0000
```

Gambar 9 Hasil Proses Training Epoch 500

```
38/38 [=====] - 55s 1s/step - loss: 9.9830e-04 - accuracy: 0.9996 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 991/1000
38/38 [=====] - 54s 1s/step - loss: 0.0011 - accuracy: 0.9996 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 992/1000
38/38 [=====] - 54s 1s/step - loss: 0.0012 - accuracy: 0.9996 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 993/1000
38/38 [=====] - 55s 1s/step - loss: 8.2930e-04 - accuracy: 0.9996 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 994/1000
38/38 [=====] - 54s 1s/step - loss: 0.0018 - accuracy: 0.9987 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 995/1000
38/38 [=====] - 54s 1s/step - loss: 7.5381e-04 - accuracy: 0.9996 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 996/1000
38/38 [=====] - 53s 1s/step - loss: 8.3985e-04 - accuracy: 0.9996 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 997/1000
38/38 [=====] - 54s 1s/step - loss: 0.0225 - accuracy: 0.9979 - val_loss: 1.3194e-06 - val_accuracy: 1.0000
Epoch 998/1000
38/38 [=====] - 54s 1s/step - loss: 0.0068 - accuracy: 0.9975 - val_loss: 1.8626e-09 - val_accuracy: 1.0000
Epoch 999/1000
38/38 [=====] - 54s 1s/step - loss: 0.0046 - accuracy: 0.9983 - val_loss: 1.0104e-04 - val_accuracy: 1.0000
Epoch 1000/1000
38/38 [=====] - 54s 1s/step - loss: 0.0167 - accuracy: 0.9954 - val_loss: 2.9270e-09 - val_accuracy: 1.0000
```

Gambar 10 Hasil Proses Training Epoch 1000

Gambar 9 dan 10 merupakan hasil dari proses *training* dengan *epoch* 500 dan 1000, dimana hasil *training* pada *epoch* 500 dan 1000 menunjukkan accuracy sebesar 0.9642 dan 0.9954 dengan *loss* masing-masing sebesar 0.0359 dan 0.0167 yang artinya model telah berhasil menyesuaikan diri dengan data pelatihan serta *runtime* yang dibutuhkan untuk *training* per *epoch*-nya adalah masing-masing adalah 59s 8s/step dan 54s 1s/step dalam 74 dan 38 step (*batch*) yang dijalani dalam satu *epoch*. Sedangkan untuk nilai *val_loss* masing-masing sebesar 2.9972e-04 dan 2.9270e-09 serta *val_accuracy* 1.0000 adalah metrik hasil pelatihan pada data validasi atau data yang tidak digunakan selama pelatihan, tetapi digunakan untuk menguji kinerja model. Nilai "*val_loss*" yang sangat rendah dan "*val_accuracy*" yang mendekati 1.0 menunjukkan bahwa model sangat baik dalam menggeneralisasi dari data pelatihan ke data validasi. Secara keseluruhan, hasil ini menunjukkan bahwa model CNN yang dirancang dengan arsitektur VGG-16 telah berhasil mengalami pelatihan selama 500 dan 1000 *epoch* dengan parameter yang berbeda, hasilnya sangat baik pada data latih maupun data validasi. Model yang dikembangkan ini mempunyai tingkat akurasi yang tinggi dan nilai *loss* yang sangat rendah, yang membuktikan bahwa model tersebut telah belajar dengan baik.

2.4 Testing

Proses *testing* atau pengujian merupakan proses untuk menguji model yang diperoleh dari proses pelatihan sebelumnya. Pengujian dilakukan dengan menguji tiap model menggunakan data citra baru diuar data pelatihan. Pengujian model yang dirancang

dengan menggunakan data citra yang masuk ke dalam kelompok data *testing* dengan teknik pengujian menggunakan metode *confusion matrix*. Pada proses ini performa model yang dirancang dapat dianalisa dari sisi *accuracy*, *precision*, *recall* dan *f-measure*. *Accuracy* adalah rasio prediksi benar (*positive* dan *negative*) dengan keseluruhan data, sementara *precision* adalah rasio pediksi benar positif dibandingkan dengan keseluruhan data yang benar positif[21]. Pada pengujian akan dilihat akurasi antara *accuracy*, *precision*, *recall* dan *f-measure*. Pengujian dinyatakan baik apabila keempat parameter pengujian tersebut memiliki *range value* yang berdekatan dan tidak ada penurunan di bawah 10%. Jika terjadi hal seperti ini bisa dipastikan terjadi *overfitting* pada proses *training* dan *testing* data, maka dari itu diterapkanlah *dropout* untuk mengurangi atau mencegah terjadinya hal tersebut. Sesuai dengan proses *training* yang telah dilakukan sebelumnya, dimana pada penelitian ini prosedur pelatihan diatur sebanyak 500 dan 1000 *epoch* dengan *loss* 0.0359 dan 0.0167 dan *accuracy* 0.9642 dan 0.9954 dengan menggunakan perbandingan dataset 80 : 20 yaitu 80% data citra train dan 20% data citra validasi untuk setiap kelasnya. Keakuratan 0.9642 dan 0.9954 membuktikan bahwa model telah dapat melakukan klasifikasi jenis jajanan khas Bali dengan tingkat akurasi sebesar 96,42% dan 99,54% dengan kesalahan klasifikasi sebesar 3,59% dan 1,67% pada dataset training. Untuk menguji kemampuan model dalam melakukan klasifikasi gambar jajanan khas Bali, maka perlu dilakukan pengujian model dengan menggunakan gambar jajanan khas Bali di luar data pelatihan.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengujian

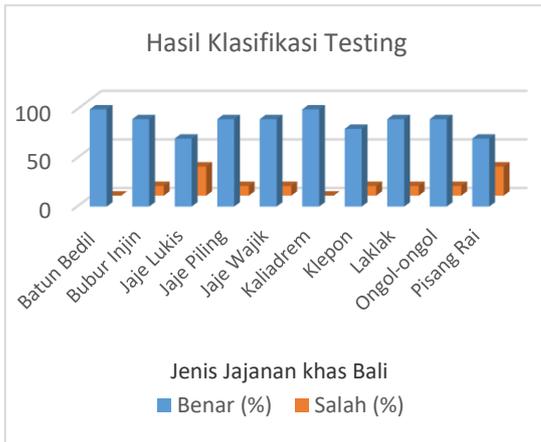
Pada tahapan ini akan dibahas tentang hasil dari pengujian yang telah dilakukan pada model yang telah dikembangkan sebelumnya menggunakan data baru diluar pelatihan yang bertujuan untuk mengetahui apakah model yang dikembangkan dan dilatih sebelumnya mampu menggeneralisasikan data citra baru serta akan dilakukan analisis hasil uji jika model yang dibangun tidak dapat menggeneralisasikan data baru secara baik atau menderita *overfitting*. Data uji yang digunakan sebanyak 100 citra, dimana terdiri dari 10 citra dari setiap kelasnya yaitu Klepon 10 citra, Laklak 10 citra,

Kaliadrem 10 citra, Jaje Lukis 10 citra, Jaje Batun Bedil 10 citra, Pisang Rai 10 citra, Jaje Piling-piling 10 citra, Jaje Wajik 10 citra, Ongol-ongol 10 citra dan Bubur Injin 10 citra. Pengujian model dilakukan dengan menginputkan citra satu per satu ke dalam model yang telah dilatih dan memastikan model dapat melakukan klasifikasi dengan benar. Selain itu, hasil prediksi yang dihasilkan dari uji model ini akan dicatat dan dituangkan dalam bentuk matriks konfusi yang bertujuan untuk menghitung *accuracy*, *precision*, *recall* dan *F1-score* dari hasil uji tersebut. Setelah melakukan uji model yang telah di *training* dengan data baru diluar data latih didapatkan hasil yang sesuai dengan tabel 2 berikut.

Tabel 2. Hasil Pengujian Akurasi Klasifikasi Jajanan Khas Bali

Jenis	Benar		Salah	
	Jumlah Data	Persentase (%)	Jumlah Data	Persentase (%)
Batun Bedil	10	100	0	0
Bubur Injin	9	90	1	10
Jaje Lukis	7	70	3	30
Jaje Piling	9	90	1	10
Jaje Wajik	9	90	1	10
Kaliadrem	10	100	0	0
Klepon	8	80	2	20
Laklak	9	90	1	10
Ongol-ongol	9	90	1	10
Pisang Rai	7	70	3	30
Terendah	7	70	0	0
Tertinggi	10	100	3	30

Berdasarkan hasil pengujian akurasi pada tabel 2 diatas, model yang dikembangkan memperoleh tingkat akurasi tertinggi sebesar 100% dan tingkat akurasi terendah sebesar 70%. Hasil yang telah diperoleh tersebut dapat diartikan bahwa model VGG-16 yang dikembangkan telah berhasil mengenali citra jajanan khas Bali yang diinputkan oleh user. Adapun visualisasi dari hasil pengujian akurasi klasifikasi jajanan khas Bali pada model VGG-16 yang dirancang dapat dilihat pada gambar 11 sebagai berikut.



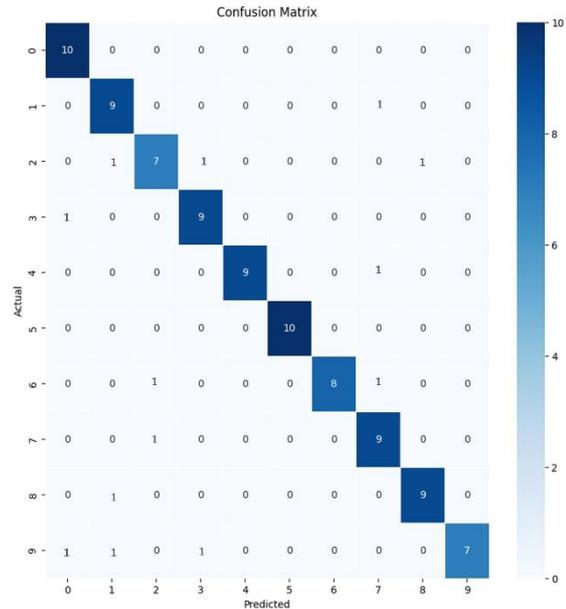
Gambar 11. Grafik Pengujian Akurasi Klasifikasi Jajanan Khas Bali

Berdasarkan gambar 11 yang merupakan grafik dari hasil pengujian tingkat akurasi model dalam mengenali jenis jajanan khas Bali yang diinputkan. Berdasarkan hasil pengujian yang dilakukan hasil pengujian terendah diperoleh pada inputan Jaje Lukis dan Pisang Rai, rendahnya tingkat akurasi tersebut disebabkan karena ciri visual Jaje Lukis dan Pisang Rai memiliki kemiripan dengan ciri visual citra jajanan lain. Sedangkan hasil akurasi tertinggi diperoleh pada saat menginputkan citra Batun Bedil dan Kaliadrem, dimana hal ini terjadi karena bentuk dan fitur pada citra Batun Bedil dan Kaliadrem tidak banyak dimiliki oleh data citra jajanan lain serta pola objek citra yang sangat mudah dikenali.

3.2 Analisis Hasil Uji

Analisis hasil uji dilakukan untuk mengevaluasi hasil yang telah didapatkan pada pengujian model serta melakukan evaluasi pada kinerja model VGG-16 yang telah dikembangkan selain itu analisis hasil uji juga dilakukan untuk melihat seberapa banyak persentase kesalahan klasifikasi yang diperoleh dari model yang telah dilatih sebelumnya. Kemampuan model VGG-16 yang telah dikembangkan dalam menggeneralisasi data citra baru perlu dianalisis, apakah model VGG-16 yang dikembangkan menderita *overfitting* atau tidak. Hal ini akan sangat bermanfaat untuk evaluasi dan perbaikan model VGG-16 yang dikembangkan. Dalam hal ini hasil yang telah didapatkan dari hasil uji model VGG-16 yang dikembangkan akan direpresentasikan dalam bentuk *confusion matrix* dengan 10 kelas yang berbeda sesuai dengan dataset yang digunakan.

Adapun *confusion matrix* dari hasil pengujian model dapat dilihat pada gambar 12 berikut.



Gambar 12. Confusion matrix dengan 10 Kelas

Berdasarkan hasil *Confusion matrix* diatas didapatkan hasil klasifikasi dari masing-masing kelas yang berbeda. Masing-masing kelas yang diuji menggunakan data citra sebanyak 10 citra. Perhitungan *Confusion matrix* bertujuan untuk melihat akurasi, presisi, *recall* dan *F1-score* dari masing-masing kelas setelah itu akan dilakukan perhitungan nilai rata-rata dari matriks-matriks yang didapat. Perhitungan *accuracy* dilakukan dengan menentukan nilai benar positif, benar negatif, salah positif, dan salah negatif antara kelas dan hasil prediksi dari model yang telah diuji. Penentuan benar positif, benar negatif, salah positif, dan salah negatif dilakukan pada masing-masing kelasnya dengan melihat hasil prediksi pada tabel *Confusion matrix* diatas. Adapun perhitungan nilai benar positif, benar negatif, salah positif, dan salah negatif yang didapat dari hasil pengujian dapat dilihat pada tabel gambar 13 berikut.

```
'Batun Bedil': {'TP': 10, 'TN': 88, 'FP': 2, 'FN': 0},
'Bubur Injin': {'TP': 9, 'TN': 87, 'FP': 3, 'FN': 1},
'Jaje Lukis': {'TP': 7, 'TN': 88, 'FP': 2, 'FN': 3},
'Jaje Piling': {'TP': 9, 'TN': 89, 'FP': 1, 'FN': 1},
'Jaje Wajik': {'TP': 9, 'TN': 90, 'FP': 0, 'FN': 1},
'Kaliadrem': {'TP': 10, 'TN': 90, 'FP': 0, 'FN': 0},
'Klepon': {'TP': 8, 'TN': 90, 'FP': 0, 'FN': 2},
'Laklak': {'TP': 9, 'TN': 87, 'FP': 3, 'FN': 1},
'Ongol-ongol': {'TP': 9, 'TN': 89, 'FP': 1, 'FN': 1},
'Pisang rai': {'TP': 7, 'TN': 90, 'FP': 0, 'FN': 3},
```

Gambar 13. Perhitungan prediksi Confusion matrix

Berdasarkan gambar 13 diatas didapat nilai dari benar positif, benar negatif, salah positif, dan salah negatif pada setiap kelasnya. Selanjutnya, perhitungan akurasi akan dihitung menggunakan rumus 1 berikut.

$$Akurasi = \frac{(tp+tn)}{(tp+tn+fp+fn)} \quad (1)$$

Akurasi Total = 975/1000 = 0,975

Akurasi total yang didapat dari 10 kelas adalah sebesar 0.975 yang dapat diartikan akurasi total dari hasil prediksi model yang telah diuji pada 10 kelas di luar data uji adalah sebesar 97,5%. Selanjutnya, perhitungan presisi akan dihitung menggunakan rumus 2 berikut.

$$Precision = \frac{(tp)}{(tp+fp)} \quad (2)$$

Presisi total = 87/99 = 0,879

Presisi total yang didapatkan pada hasil prediksi 10 kelas yang telah dilakukan adalah sebesar 0.87 yang dapat diartikan bahwa prediksi positif model benar yang dihasilkan adalah sebesar 87,9%. Selanjutnya adalah menghitung *recall* (sensitivitas) model terhadap kebenaran positif yang seharusnya dengan menggunakan rumus 3 berikut.

$$f1 - measure = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \quad (3)$$

$F1-score = 2 \times (0,878 \times 0,87) / (0,878 + 0,87) = 0,874$

F1 Score adalah rata-rata harmonik dari presisi dan *recall*, dimana pada hasil uji yang telah dilakukan pada 10 kelas data uji diperoleh hasil dari *F1-score* sebesar 0,874 yang dapat diartikan bahwa rata-rata keseimbangan antara presisi dan *recall* sebesar 87,4%.

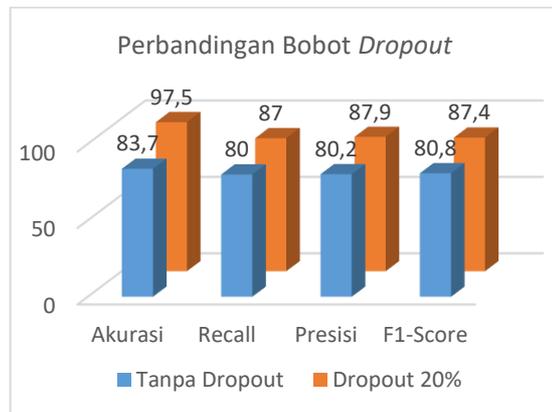
Evaluasi model akan dilakukan dengan perbandingan antara hasil *accuracy*, *precision*, *recall* dan *f1-score* dari hasil pengujian model dengan data baru dan hasil *training* yang sudah dilakukan untuk mencari model terbaik dari VGG-16 yang sudah dikembangkan. Adapun beberapa parameter perbandingan yang digunakan adalah bobot *dropout*, banyaknya *epoch* dan banyaknya *batch size* yang digunakan pada model VGG-16. Perbandingan ini bertujuan untuk melakukan evaluasi terhadap model yang dikembangkan, serta dapat melihat parameter apa saja yang berpengaruh terhadap keakuratan hasil klasifikasi dari model yang telah dikembangkan. Adapun perbandingan

bobot *dropout* yang digunakan pada model yang dibangun dapat dilihat pada gambar 7 berikut.

Tabel 3. Perbandingan Hasil Pengujian Terhadap Bobot *Dropout*

Ket	Akurasi (%)		Recall (%)		Presisi (%)		F1-Score (%)	
	Train	Test	Train	Test	Train	Test	Train	Test
0	95,52	83,7	93,8	80	92	80,2	92,34	80,8
20	99	97,5	94	87	95,91	87,9	94,94	87,4

Pada tabel 3 berisikan hasil klasifikasi pengujian terhadap bobot *dropout* yang telah digunakan sebagai perbandingan data uji. Dalam penelitian ini model diuji dengan dan tanpa *dropout*, dimana model yang diuji tanpa menggunakan *dropout* menghasilkan jumlah akurasi sebesar 83,7% sedangkan model yang menggunakan *dropout* sebesar 20% menghasilkan akurasi yang lebih tinggi yaitu sebesar 97,5%. Penggunaan *dropout* mengurangi model mengalami *overfitting*, dimana model dapat mengenali citra *training* dengan akurasi yang tinggi namun tidak mampu menggeneralisasi kemampuan yang sama pada data uji baru yang belum pernah dilihat sebelumnya. Selanjutnya bobot *dropout* 20% akan digunakan dalam model untuk menguji citra masukan pada model yang memiliki performa terbaik dapat dilihat pada gambar 14 berikut.



Gambar 14. Grafik Performa Pengaruh Bobot *Dropout*

Gambar 14 merupakan visualisasi dari pengaruh bobot *dropout* yang ditetapkan pada penelitian ini. Bobot *dropout* yang diujikan adalah tanpa *dropout* dan dengan *dropout* sebesar 20%. Berdasarkan grafik visualisasi di atas, performa model dengan bobot *dropout* sebesar 20% merupakan model terbaik dan menghasilkan *accuracy*, *precision*, *recall* dan *f1-score* yang lebih tinggi daripada model yang tidak menggunakan *dropout*. Selisih antara *accuracy*,

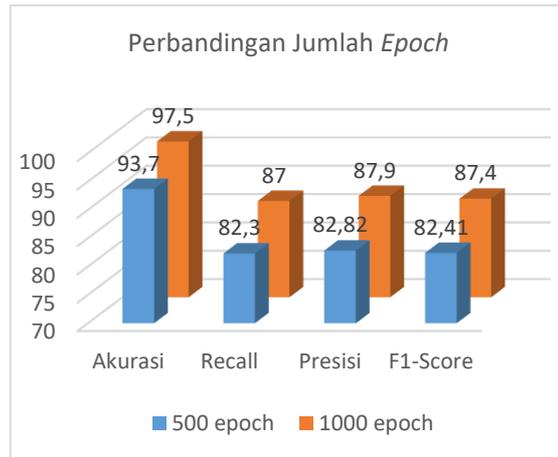
precision, *recall* dan *f1-score* yang dihasilkan dari pengujian masing-masing sebesar 3,8%, 7%, 7,67% dan 6,63%. Hal ini menunjukkan bahwa model yang dikembangkan dengan *dropout* 20% dapat menghasilkan pengujian yang lebih baik, dimana penggunaan *dropout* sendiri bertujuan untuk mengurangi beberapa persen neuron pada proses klasifikasi pada model yang dikembangkan sehingga dapat mencegah terjadinya *overfitting*.

Evaluasi model selanjutnya adalah dengan perbandingan hasil *accuracy*, *precision*, *recall* dan *f1-score* dari hasil pengujian model dengan data baru dan hasil *training* yang sudah dilakukan untuk mencari model terbaik dari VGG-16 yang sudah dikembangkan dengan jumlah *epoch* yang digunakan. Dimana jumlah *epoch* yang diperlukan untuk melatih model dengan baik dapat bervariasi tergantung pada tugas, kompleksitas model, dan dataset yang digunakan. Terlalu sedikit *epoch* mungkin tidak akan mencapai performa yang baik, sementara terlalu banyak *epoch* bisa menyebabkan *overfitting*, yaitu model dapat memahami pelatihan dengan sangat baik tetapi tidak dapat menggeneralisasi data baru dengan baik. Adapun perbandingan jumlah *epoch* yang digunakan pada model yang dibangun dapat dilihat pada tabel 4 berikut.

Tabel 4. Perbandingan Hasil Pengujian Terhadap Jumlah *Epoch*

Ket	Akurasi (%)		Recall (%)		Presisi (%)		F1-Score (%)	
	Train	Test	Train	Test	Train	Test	Train	Test
500	96,42	93,7	92,2	82,3	92	82,82	92,34	82,41
1000	99	97,5	94	87	95,91	87,9	94,94	87,4

Pada tabel 4 berisikan hasil klasifikasi pengujian terhadap jumlah *epoch* yang digunakan. Dalam penelitian ini model diuji dengan jumlah *epoch* 500 dan 1000, dimana model yang diuji menggunakan jumlah *epoch* sebanyak 500 menghasilkan jumlah akurasi sebesar 93,7% sedangkan model yang menggunakan jumlah *epoch* sebanyak 1000 menghasilkan akurasi yang lebih tinggi yaitu sebesar 97,5%. Selanjutnya jumlah *epoch* 1000 akan digunakan dalam model untuk menguji citra masukan pada model yang memiliki performa terbaik dapat dilihat pada gambar 15 berikut.



Gambar 15. Grafik Performa Pengaruh Jumlah *Epoch*

Gambar 15 merupakan visualisasi dari pengaruh jumlah *epoch* yang ditetapkan pada penelitian ini. Jumlah *epoch* yang diujikan adalah 500 dan 1000. Berdasarkan grafik visualisasi di atas, performa model dengan jumlah *epoch* 1000 merupakan model terbaik dan menghasilkan *accuracy*, *recall*, *precision* dan *f1-score* yang lebih tinggi daripada model yang menggunakan jumlah *epoch* 500. Selisih antara *accuracy*, *recall*, *precision* dan *f1-score* yang dihasilkan dari pengujian masing-masing sebesar 3,8%, 4,7%, 5,05% dan 5,02%. Hal ini membuktikan bahwa model yang dikembangkan dengan menggunakan jumlah *epoch* 1000 terbukti menghasilkan pengujian yang lebih baik, dimana penggunaan *epoch* sendiri bertujuan untuk meningkatkan kemampuan klasifikasi pada sebuah data citra, selain untuk menghafal pola *epoch* juga membantu memastikan bahwa model dapat melakukan generalisasi dengan baik pada data yang belum pernah dilihat. Penggunaan jumlah *epoch* yang tepat dalam melatih model sangat penting dalam menghasilkan model yang baik.

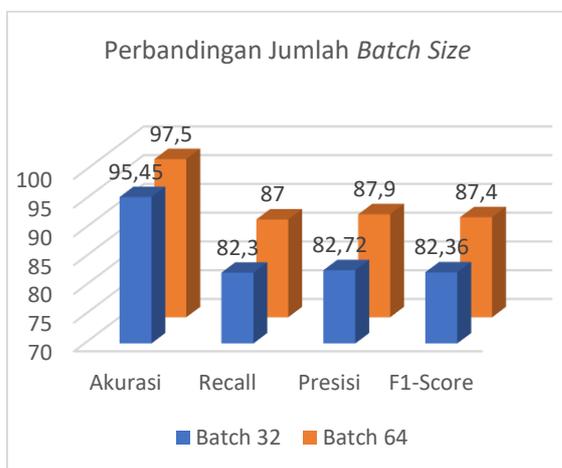
Evaluasi model selanjutnya adalah dengan perbandingan hasil *accuracy*, *recall*, *precision* dan *f1-score* dari hasil pengujian model dengan data baru dan hasil *training* yang sudah dilakukan untuk mencari model terbaik dari VGG-16 yang sudah dikembangkan dengan jumlah *batch size* yang digunakan. Dimana besarnya *batch size* akan dipengaruhi oleh sejumlah faktor, termasuk kompleksitas model, ukuran dataset, dan sumber daya komputasi yang tersedia. Penggunaan *batch size* memungkinkan penggunaan sumber daya komputasi yang lebih efisien. Memproses seluruh dataset pelatihan dalam satu iterasi

akan membutuhkan waktu dan sumber daya yang sangat tinggi, terutama jika dataset besar. Dengan menggunakan *batch size*, komputasi dapat dilakukan dalam *batch* kecil, yang mempercepat pelatihan. Adapun perbandingan jumlah *batch size* yang digunakan pada model yang dibangun dapat dilihat pada tabel 5 berikut.

Tabel 5. Perbandingan Hasil Pengujian Terhadap Jumlah *Batch Size*

Ket	Akurasi (%)		Recall (%)		Presisi (%)		F1-Score (%)	
	Train	Test	Train	Test	Train	Test	Train	Test
32	96,23	95,45	91,35	82,3	92,7	82,72	92,34	82,36
64	99	97,5	94	87	95,91	87,9	94,94	87,4

Pada tabel 5 berisikan hasil klasifikasi pengujian terhadap banyaknya *batch size* yang digunakan. Dalam penelitian ini model diuji dengan jumlah *batch size* 32 dan 64, dimana model yang diuji menggunakan jumlah *batch size* sebanyak 32 menghasilkan jumlah akurasi sebesar 95,45% sedangkan model yang menggunakan jumlah *batch size* sebanyak 64 menghasilkan akurasi yang lebih tinggi yaitu sebesar 97,5%. Selanjutnya jumlah *batch size* 64 akan digunakan dalam model untuk menguji citra masukan pada model yang memiliki performa terbaik dapat dilihat pada gambar 16 berikut.



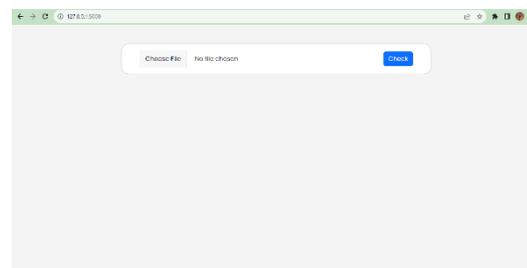
Gambar 16. Grafik Performa Pengaruh Jumlah *Batch Size*

Gambar 16 merupakan visualisasi dari pengaruh jumlah *batch size* yang ditetapkan pada penelitian ini. Jumlah *batch size* yang diujikan adalah 32 dan 64. Berdasarkan grafik visualisasi di atas, performa model dengan jumlah *batch size* 64 merupakan model terbaik dan menghasilkan *accuracy*, *recall*, *precision* dan *f1-*

score yang lebih tinggi daripada model yang menggunakan jumlah *batch size* 32. Selisih antara *accuracy*, *recall*, *precision* dan *f1-score* yang dihasilkan dari pengujian masing-masing sebesar 2,05%, 4,7%, 5,15% dan 5,07%. Hal ini membuktikan bahwa model yang dikembangkan dengan menggunakan jumlah *batch size* 64 terbukti mendapatkan hasil pengujian yang lebih baik.

3.3 Implementasi Model pada Sistem

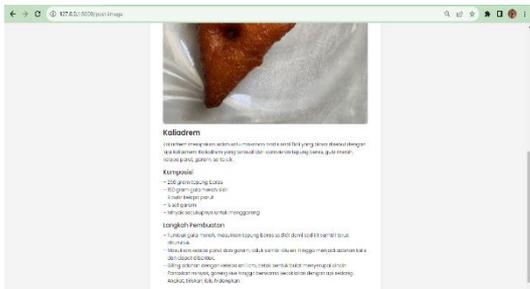
Pada tahap ini akan dipaparkan mengenai tampilan user interface sistem yang dibangun. Setelah melakukan pengujian dan evaluasi pada model VGG-16 yang dirancang maka didapatkan sebuah model terbaik untuk digunakan pada penelitian ini. Dalam penelitian ini penggunaan model dengan menerapkan bobot *dropout* sebesar 20%, jumlah *epoch* 1000 serta ukuran *batch size* 64. Dalam hal ini user interface yang dibangun berbasis web, dimana user akan menginputkan sebuah citra uji jajanan khas Bali sesuai dengan kelas yang sudah dilatih pada model yang digunakan. Adapun tampilan user interface sistem yang dibangun dapat dilihat pada gambar 17 berikut.



Gambar 17. Tampilan UI Awal Sistem

Pada gambar 17 diatas merupakan tampilan user interface dari sistem yang dibangun. Pada tampilan tersebut terdapat fitur untuk mengupload gambar dan button untuk cek data citra yang diinput oleh user. Setelah user mengupload data citra, maka akan muncul tampilan gambar yang diinputkan yang berisikan deskripsi, komposisi beserta langkah-langkah pembuatan dari jajanan khas Bali yang diinputkan. Data citra yang diinputkan adalah data citra uji diluar pelatihan sesuai dengan data uji yang dijadikan pengujian dalam model yang dibangun. Tujuan dikembangkannya user interface ini adalah mengacu pada pengenalan jajanan khas Bali dengan memanfaatkan *Artificial Intelligence* khususnya *Deep Learning*

yang sedang banyak dikembangkan oleh para peneliti, dimana pengembangan user interface ini juga bertujuan untuk dapat membantu melestarikan budaya Bali khususnya pada bidang kuliner. Adapun tampilan user interface system yang sudah diinputkan citra dapat dilihat pada gambar 18 berikut.



Gambar 18. Tampilan UI Akhir Sistem

Pada gambar 18 merupakan tampilan user interface dari sistem yang sudah diinputkan gambar. Adapun yang ditampilkan pada user interface tersebut terdapat jenis, komposisi serta langkah-langkah pembuatan dari jajanan khas Bali sesuai dengan gambar citra yang diinputkan. Beberapa jajanan khas Bali yang diinputkan masih mengalami kesalahan klasifikasi, dimana hal tersebut dikarenakan fitur yang terdapat pada citra yang diinputkan memiliki kemiripan dengan data citra jajanan khas Bali tertentu. Namun ada juga citra jajanan khas Bali yang diuji sebanyak 10 citra dan berhasil diklasifikasikan dengan benar semua sesuai dengan kelasnya, yaitu data citra Kaliadrem dan Batun Bedil. Kesalahan klasifikasi yang terjadi sudah dilakukan analisis hasil uji pada sub bab sebelumnya, dimana hasil uji dari model terbaik dengan tingkat akurasi 97,5% yang digunakan pada penelitian ini.

4. KESIMPULAN

Adapun kesimpulan yang dapat diambil dari penelitian yang berjudul Klasifikasi Jajanan Khas Bali Untuk Preservasi Pengetahuan Kuliner Lokal Menggunakan Arsitektur VGG-16 adalah

DAFTAR PUSTAKA

- [1] I. M. Purna and K. Dwikayana, "Betutu Bali : Menuju Kuliner Diplomasi Budaya Indonesia," *Patanjala J. Penelit. Sej. dan Budaya*, vol. 11, no. 2, p. 265, 2019, doi: 10.30959/patanjala.v11i2.478.
- [2] Lumanwauw, "PENGEMBANGAN

arsitektur VGG-16 yang dikembangkan mampu menghasilkan akurasi sebesar 99%, presisi 95,91% *recall* 94% dan *f1-score* 94% pada data *training* dan validasi. Arsitektur VGG-16 yang dikembangkan mampu melakukan klasifikasi pada data citra jajanan Khas Bali dengan menghasilkan tingkat akurasi sebesar 97,5%, presisi 87,9%, *recall* 87% serta *f1-score* sebesar 87,4% dengan parameter pengujian *dropout* 20%, *batch size* 64 serta *epoch* 1000 pada data citra uji diluar data pelatihan dan validasi. Tingkat akurasi yang dihasilkan pada data uji tergolong baik, dimana hasil uji menunjukkan tingkat akurasi yang tinggi walaupun tidak lebih tinggi dari tingkat akurasi dari data *training* namun model yang dibangun tidak mengalami *overfitting* dan mampu melakukan generalisasi pada data citra baru. Model yang dibangun dianggap *overfitting* jika memiliki selisih yang signifikan, semakin besar selisihnya, semakin mungkin model mengalami *overfitting*. Sebuah selisih yang signifikan, seperti selisih lebih dari 10% atau 20%, dapat menjadi petunjuk kuat. Namun dalam penelitian ini selisih akurasi yang dihasilkan dari data *training* dan data uji sebesar 1,5%. Tingkat akurasi yang dihasilkan dari model VGG-16 yang dibangun pada penelitian ini sangat dipengaruhi oleh beberapa parameter, diantaranya jumlah dataset dan kompleksitas suatu dataset, persentase penggunaan *dropout* yang tepat sangat diperlukan sesuai dengan kebutuhan model yang dibangun, ukuran *batch size* juga akan mempengaruhi hasil akurasi model serta banyaknya *epoch* yang digunakan. Percobaan harus dilakukan untuk menentukan jumlah yang optimal untuk tugas dan dataset tertentu, hal ini berguna untuk menemukan titik optimal agar model yang dibangun tidak mengalami *overfitting*. Hasil dari model VGG-16 dengan performa terbiik akan diimplementasikan pada sbuh sstm informasi dengan tujuan untuk membantu pengenalan jajanan khas Bali berdasarkan dataset yang sudah dilatih sebelumnya.

MAKANAN TRADISIONAL BALI PADA PASAR MALAM DI PASAR SINDU , SANUR , BALI Nelsye Lumanauw Pasar malam Sindu menjadi tempat favorit bagi wisatawan yang tinggal di sekitar wilayah Sanur untuk menikmati makanan tradisional Indonesia . Banyak makan," *J. Ilm. Hosp. Manag.*, vol. 9, no. 1, pp. 83–92, 2018.

- [3] C. Purnama Yanti and S. Ika Murpratiwi, "Pengembangan Augmented Reality Interaktif untuk Pengenalan Jajanan Tradisional Bali dengan Marker Based Tracking," vol. 6, no. 2, pp. 2622–4615, 2021, doi: 10.32493/informatika.v6i2.12084.
- [4] C. I. R. Marsiti, N. M. Suriani, and N. W. Sukerti, "Strategi Pengembangan Makanan Tradisional Berbasis Teknologi Informasi Sebagai Upaya Pelestarian Seni Kuliner Bali," *J. IKA*, vol. 17, no. 2, p. 128, 2019, doi: 10.23887/ika.v17i2.19844.
- [5] R. J. Gunawan, B. Irawan, and C. Setianingsih, "Pengenalan Ekspresi Wajah Berbasis Convolutional Neural Network Dengan Model Arsitektur VGG16 Facial Expression Recognition Based On Convolutional Neural Network with VGG16 Architecture Model," *e-Proceeding Eng.*, vol. 8, no. 5, p. 6442, 2021.
- [6] E. Tanuwijaya and A. Roseanne, "Modifikasi Arsitektur VGG16 untuk Klasifikasi Citra Digital Rempah-Rempah Indonesia," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 1, pp. 189–196, 2021, doi: 10.30812/matrik.v21i1.1492.
- [7] A. M. Rizki and N. Marina, "Klasifikasi Kerusakan Bangunan Sekolah Menggunakan Metode Convolutional Neural Network Dengan Pre-Trained Model Vgg-16," *J. Ilm. Teknol. dan Rekayasa*, vol. 24, no. 3, pp. 197–206, 2019, doi: 10.35760/tr.2019.v24i3.2396.
- [8] I. B. K. Sudiatmika and I. G. A. A. S. Dewi, "Pengenalan Karakter Wayang Bali Dengan Arsitektur VGG-16," *J. Teknol. Inf. dan Komput.*, vol. 7, no. 1, pp. 47–53, 2021.
- [9] M. A. Subhi and S. M. Ali, "A deep convolutional neural network for food detection and recognition," *2018 IEEE EMBS Conf. Biomed. Eng. Sci. IECBES 2018 - Proc.*, pp. 284–287, 2019, doi: 10.1109/IECBES.2018.08626720.
- [10] I. P. A. E. Darma Udayana, M. Sudarma, and P. G. Surya Cipta Nugraha, "Implementation of Convolutional Neural Networks to Recognize Images of Common Indonesian Food," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 846, no. 1, 2020, doi: 10.1088/1757-899X/846/1/012023.
- [11] N. Hnoohom and S. Yuenyong, "Thai Fast Food Image Classification Using Deep Learning," *1st Int. ECTI North. Sect. Conf. Electr. Electron. Comput. Telecommun. Eng. ECTI-NCON 2018*, pp. 116–119, 2018.
- [12] R. Windiawan, A. Suharso, and S. Artikel, "Identifikasi Penyakit pada Daun Kopi Menggunakan Metode Deep Learning VGG16 INFO ARTIKEL ABSTRAK," *Exploreit*, vol. 13, no. 2, pp. 9–16, 2019.
- [13] M. T. Islam, B. M. N. Karim Siddique, S. Rahman, and T. Jabid, "Image Recognition with Deep Learning," *2018 Int. Conf. Intell. Informatics Biomed. Sci. ICIBMS 2018*, vol. 3, pp. 106–110, 2018, doi: 10.1109/ICIBMS.2018.8550021.
- [14] P. A. Nugroho, I. Fenriana, and R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (Cnn) Pada Ekspresi Manusia," *Algor*, vol. 2, no. 1, pp. 12–21, 2020.
- [15] I. Wulandari, H. Yasin, and T. Widiari, "Klasifikasi Citra Digital Bumbu Dan Rempah Dengan Algoritma Convolutional Neural Network (Cnn)," *J. Gaussian*, vol. 9, no. 3, pp. 273–282, 2020, doi: 10.14710/j.gauss.v9i3.27416.
- [16] S. Yadav, Alpana, and S. Chand, "Automated Food image Classification using Deep Learning approach," *2021 7th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2021*, pp. 542–545, 2021, doi: 10.1109/ICACCS51430.2021.9441889.
- [17] G. Ciocca, P. Napoletano, and R. Schettini, "Food Recognition: A New Dataset, Experiments, and Results," *IEEE J. Biomed. Heal. Informatics*, vol. 21, no. 3, pp. 588–598, 2017, doi: 10.1109/JBHI.2016.2636441.
- [18] A. Kurniadi and M. Fal Sadikin, "Implementasi Convolutional Neural Network Untuk Klasifikasi Varietas Pada Citra Daun Sawi Menggunakan Keras Implementation of Neural Network Convolutionals For Classification of Variety on Image of Collards Meat Leaves Using The Keras," *J. Comput. Inf. Technol.*, vol. 4, no. 1, pp. 25–33, 2020, [Online]. Available: <http://e-journal.unipma.ac.id/index.php/doubleclick>

- [19] A. Ramos-Michel, M. Pérez-Cisneros, E. Cuevas, and D. Zaldivar, "Image Classification with Convolutional Neural Networks," *Stud. Comput. Intell.*, vol. 967, pp. 445–473, 2021, doi: 10.1007/978-3-030-70542-8_18.
- [20] A. Ramanath, S. Muthusrinivasan, Y. Xie, S. Shekhar, and B. Ramachandra, "Ndpi versus cnn features in deep learning for land cover classification of aerial images," *Int. Geosci. Remote Sens. Symp.*, vol. 2019-July, pp. 6483–6486, 2019, doi: 10.1109/IGARSS.2019.8900165.
- [21] A. Kholik, "Klasifikasi Menggunakan Convolutional Neural Network (Cnn) Pada Tangkapan Layar Halaman Instagram," *J. Data Min. dan Sist. Inf.*, vol. 2, no. 2, pp. 10–20, 2021, [Online]. Available: <https://ejurnal.teknokrat.ac.id/index.php/JDMSI/article/view/1345/673%0Ahttps://ejurnal.teknokrat.ac.id/index.php/JDMSI/article/view/1345>