ISSN: 1829-667X / E-ISSN: 2460-9021
DOI: 10.31515/telematika.v18i1.4025

VGG16 Transfer Learning Architecture for Salak Fruit Quality Classification

Klasifikasi Kualitas Buah Salak dengan Transfer Learning Arsitektur VGG16

Rismiyati[1], Ardytha Luthfiarta[2]

[1] Departemen Ilmu Komputer/Informatika, Universitas Diponegoro, Indonesia
[2] Prodi Teknik Informatika, Universitas Dian Nuswantoro, Indonesia

[1*] rismiyati@live.undip.ac.id, [2] ardytha.luthfiarta@dsn.dinus.ac.id

**: Penulis korenspondensi* (*corresponding author*)

Informasi Artikel
Received: 4 December 2020
Revised: 18 January 2021
Accepted: 8 February 2021
Published: 28 February 2021

Abstract

Purpose*: This study aims to differentiate the quality of salak fruit with machine learning. Salak is classified into two classes, good and bad class.*
Design/methodology/approach*: The algorithm used in this research is transfer learning with the VGG16 architecture. Data set used in this research consist of 370 images of salak, 190 from good class and 180 from bad class. The image is preprocessed by resizing and normalizing pixel value in the image. Preprocessed images is split into 80% training data and 20% testing data. Training data is trained by using pretrained VGG16 model. The parameters that are changed during the training are epoch, momentum, and learning rate. The resulting model is then used for testing. The accuracy, precision and recall is monitored to determine the best model to classify the images.*
Findings/result*: The highest accuracy obtained from this study is 95.83%. This accuracy is obtained by using a learning rate = 0.0001 and momentum 0.9. The precision and recall for this model is 97.2 and 94.6.*
Originality/value/state of the art*: The use of transfer learning to classify salak which never been used before.*

Keywords:salak; transfer learning; VGG16; deep learning
Kata kunci: salak; transfer learning; VGG16; deep learning

Abstrak

Tujuan: Penelitian ini bertujuan untuk membedakan kualitas buah salak dengan *machine learning*. Salak diklasifikasikan menjadi dua kelas, kelas bagus dan jelek.
Perancangan/metode/pendekatan: Algoritma yang digunakan dalam penelitian ini adalah transfer learning dengan arsitektur VGG16. Data set yang digunakan dalam penelitian ini terdiri dari 370 citra salak, 190 dari kelas baik

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI: 10.31515/telematika.v18i1.4025

dan 180 dari kelas buruk. Citra masukan akan mengalami *preprocessing* dengan mengubah ukuran dan menormalkan nilai piksel pada gambar. Citra hasil *preprocessing* dibagi menjadi 80% data latih dan 20% data uji. Data pelatihan dilatih dengan menggunakan model VGG16 yang telah dilatih sebelumnya. Parameter yang diubah selama pelatihan adalah *epoch, momentum, dan learning rate*. Model yang dihasilkan kemudian digunakan untuk pengujian. Akurasi, *precision*, dan *recall* dipantau untuk menentukan model terbaik untuk melakukan klasifikasi salak,

Hasil: Akurasi tertinggi yang diperoleh dari penelitian ini adalah 95,83%. Akurasi ini diperoleh dengan menggunakan *learning rate* = 0,0001 dan momentum 0,9. Presisi dan perolehan model ini adalah 97,2 dan 94,6.

Keaslian/ *state of the art*: Penggunaan *transfer learning* untuk mengklasifikasikan salak belum pernah digunakan sebelumnya.

1. Pendahuluan

Buah salak adalah buah lokal dari Indonesia. Produsen utama buah salak di Indonesia adalah Provinsi Jawa Tengah, Provinsi Sumatra Utara, Provinsi Jawa Timur dan Provinsi DIY. Berdasar data dari Badan Pusat Statistik, produksi salak pada tahun 2019 mencapai 965.205 Ton. Sejumlah 842.350 ton, atau 87% dari total produksi tahun tersebut, dihasilkan di empat provinsi tersebut, sebagaimana ditunjukkan pada data pada **Tabel 1** [1]. Distribusi untuk buah salak dilakukan di pasar lokal dalam negeri dan pasar ekspor. Jumlah ekspor buah salak tahun 2019 mencapai 1.651 ton, meningkat 33% dari ekspor pada tahun sebelumnya sejumlah 1.233 ton.

Distribusi buah salak memerlukan proses pemilihan buah yang teliti, agar salak yang terkirim tidak menjadi mudah busuk. Hal ini terutama berlaku untuk pengiriman jarak jauh dan tujuan ekspor. Buah salak yang masuk dalam kategori bagus dan layak dikirim harus merupakan salak yang tidak terlalu matang, tidak mengandung cacat sobek maupun cacat busuk. Buah salak sendiri memiliki warna kecoklatan serta mempunyai tekstur yang bersisik. Salak dengan kematangan yang sedang mempunyai ukuran sisik sedang serta distribusi warna gelap dan terang yang merata pada kulitnya. Salak yang sudah terlalu matang mempunyai ukuran sisik lebih besar dengan warna didominasi warna cerah. Perbedaan antara salak yang mempunyai usia sedang untuk panen dan terlalu matang dapat dilihat pada **Gambar 1**.

Pemilihan buah juga dilakukan untuk memastikan salak yang mempunyai cacat busuk dan cacat sobek tidak dimasukkan. Pemilihan buah ini selama ini masih dilakukan secara manual. Pemilihan buah secara otomatis dapat dilakukan dengan menggunakan perangkat komputer. Sebuah algoritma yang mampu membedakan antara salak yang bagus dan tidak bagus dapat digunakan dalam sistem pemilihan otomatis ini

Tabel 1. Data Produksi Salak tiap Provinsi di Indonesia

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI: 10.31515/telematika.v18i1.4025

Data Produksi Salak	**Tahun (Dalam Ton)**				
Provinsi	2015	2016	2017	2018	2019
Jawa Tengah	471463	354770	576361	416860	482949
Sumatera Utara	192585	118619	162622	194455	235506
Jawa Timur	105019	77805	97164	101943	102283
Di Yogyakarta	73283	73741	37913	90296	41200
Jawa Barat	56981	22220	22602	23602	25640
Bali	27204	17007	13960	18621	15285
Provinsi Lain	38670	38188	43223	50727	52905

(a) Salak cukup umur panen, dengan ukuran sisik sedang, dan distribusi warna gelap dan terang yang seimbang

(b) Salak terlalu matang, ukuran sisik lebih besar, dan warna lebih cerah

Gambar 1. Contoh Gambar Salak dengan umur yang berbeda.

Penelitian untuk membedakan kualitas buah pernah dilakukan sebelumnya. Penelitian untuk melakukan deteksi kecacatan buah manggis dengan menggunakan *convolutional neural network* (CNN) menghasilkan akurasi 98%[2]. Penelitian lain menggunakan *classical machine learning* dilakukan untuk membedakan kualitas beberapa buah dengan beberapa algoritma. Buah yang menjadi objek penelitian adalah buah alpukat, apel, pisang dan jeruk. Algoritma yang digunakan adalah *k Nearest Neighbor* (kNN), *Support Vector Machine* (SVM), *sparse representative classifier* (SRC), dan *artificial neural network* (ANN). Hasil penelitian menunjukkan SVM mencapai akurasi tertinggi dengan akurasi 98,48% [3]. Penelitian serupa untuk mengklasifikasikan buah tomat dilakukan dengan menggunakan algoritma KNN, *multi layer perceptron* (MLP), dan *k-Means clustering.* Hasil penelitian menunjukkan algoritma kNN dan MLP mampu mendapatkan akurasi di atas 90% [4]. Penelitian sejenis dilakukan untuk membedakan kualitas buah apel, pear, jeruk, dan lemon dengan menggunakan *InceptionV3*. Masing-masing buah dibedakan menjadi empat kualitas kelas. Hasil penelitian ini menunjukkan akurasi sebesar 85% [5].

Sementara itu, penelitian untuk membedakan kualitas buah salak pernah dilakukan dengan menggunakan *classical machine learning* dan *deep learning*. Penelitian dengan menggunakan *classical machine* learning dilakukan dengan menggunakan fitur berupa *Histogram of Gradient*.

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI: 10.31515/telematika.v18i1.4025

Classifier yang digunakan adalah ELM dan SVM. Akurasi yang didapatkan adalah 95% dan 97,3% untuk ELM dan SVM [6]. Sementara itu, metode deep learning untuk membedakan kualitas salak dilakukan dengan menggunakan CNN. Akurasi yang didapatkan adalah 81,45% [7].

Dari penelitian terdahulu tersebut dapat dilihat bahwa baik deep learning maupun classical machine learning mampu memberi hasil yang baik. Dalam kasus klasifikasi buah salak, CNN masih mendapat akurasi yang kurang dibandingkan metode classical machine learning. Hal ini dapat dikarenakan jumlah data yang digunakan yang masih sedikit, dimana dalam penelitian tersebut hanya menggunakan 370 data.

Perkembangan dalam metode deep learning yang sekarang banyak digunakan dalam training, terutama dalam dunia citra adalah dengan digunakannya *transfer learning*. *Transfer learning* adalah metode *deep learning* dimana model yang telah dilatih pada suatu permasalahan dipakai untuk permasalahan lain. *Transfer learning* memungkinkan pelatihan *deep learning* mendapatkan akurasi yang tinggi bahkan dengan menggunakan jumlah sample yang sedikit [8]. Salah satu arsitektur yang paling awal digunakan dalam *transfer learning* adalah VGG-16. VGG16 merupakan arsitekture *deep convolutional neural network* dengan total jumlah layer sejumlah 16 [9].Dalam kasus citra, data set yang digunakan untuk melatih arsitektur deep learning biasanya adalah ImageNet ImageNet adalah dataset yang terdiri dari jutaan gambar dari 1000 kelas. ImageNet banyak digunakan untuk kompetisi *machine learning*. [10]

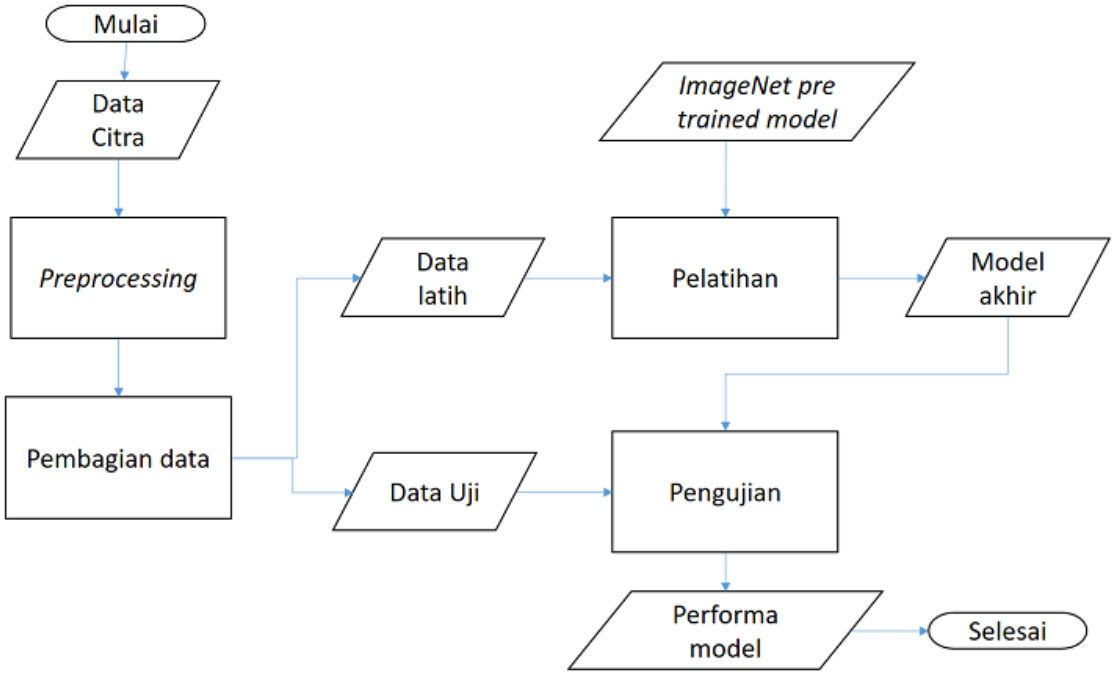
Beberapa penelitian sebelumnya yang menggunakan VGG16 untuk klasifikasi citra digunakan antara lain untuk klasifikasi penyakit tanaman terong, klasifikasi batik, dan klasifikasi citra otak manusia. Penggunaan VGG16 dalam mendeteksi tanaman buah terong mampu mengkategorikan empat penyakit tanaman tersebut dengan akurasi 99, 4% [11]. Penelitian lain untuk melakukan klasifikasi batik dengan VGG16 bertujuan membedakan batik menjadi 5 kelas jenis dasar batik. Hasil penelitian menunjukkan akurasi sebesar 89±7%[12]. Sementara itu penelitan dengan VGG16 juga dilakukan untuk melakukan klasifikasi pada citra *CT scan* otak manusia ke kelas normal dan abnormal. Hasil dari penelitian tersebut menunjukkan akurasi sebesar 100% [13]. VGG16 juga digunakan untuk mendeteksi cacat pada hasil pengelasan. Penelitian tersebut dilakukan dengan menggunakan 3000 citra latih. Akurasi yang didapat untuk mendeteksi cacat pengelasan tersebut adalah 97.2% [14]. Dari beberapa penelitian tersebut, dapat disimpulkan bahwa *transfer learning* dengan arsitektur VGG16 mampu memberikan akurasi pada pengenalan citra secara umum.

Dalam penelitian ini, *transfer learning* untuk membedakan kualitas buah salak dilakukan. Kualitas yang dimaksud adalah kualitas bagus dan tidak bagus, sehingga kelas yang dipakai dalam penelitian ini adalah kelas bagus dan tidak bagus. Buah salak dengan kualitas bagus mempunyai warna yang tidak terlalu gelap dan tidak terlalu terang. Selain itu salak juga tidak boleh memiliki cacat, baik cacat sobek maupun cacat busuk. Dalam penelitian ini, arsitektur yang digunakan adalah arsitektur VGG16. Arsitektur ini dipilih karena merupakan arsitektur *transfer learning* paling awal yang berhasil melakukan klasifikasi citra imageNet dengan akurasi yang tinggi. Model awal yang digunakan dalam pelatihan model VGG16 didapatkan dari model yang telah dilatih dengan menggunakan ImageNet data set. Penelitian ini ditujukan untuk mengetahui kemampuan *transfer learning* untuk melakukan klasifikasi pada jumlah data set yang terbatas. Selain itu, penelitian ini juga ditujukan untuk melihat pengaruh *learning rate* dan momentum yang digunakan pada optimasi *Stochastic Gradient Descent* dalam pelatihan.

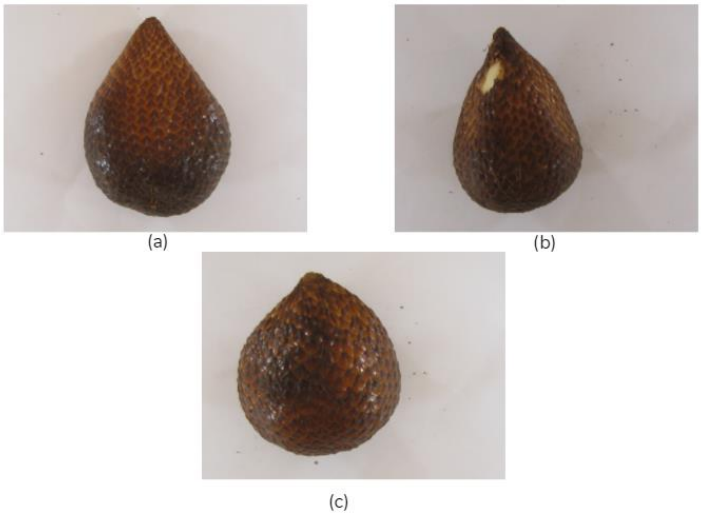
ISSN: 1829-667X / E-ISSN: 2460-9021
DOI: 10.31515/telematika.v18i1.4025

2. Metode/Perancangan

Penelitian ini dilakukan sebagaimana ditunjukkan pada **Gambar 2**. Data citra salak digunakan sebagai citra masukan. Data tersebut kemudian dikenakan proses *preprocessing*. Setelah itu, data dibagi menjadi data training (80%) dan data testing (20%). Data training kemudian dilatih dengan menggunakan VGG 16, dimana bobot awal diambil dari model yang telah dilatih pada ImageNet. Hasil akhir model kemudian dievaluasi dengan menggunakan data testing. Hasil evaluasi berupa akurasi, *precision* dan *recall* diamati untuk masing-masing skenario penelitian.



Gambar 2. Alur penelitian



Gambar 3. Contoh citra dari kelas (a) bagus, (b) cacat sobek, dan (c) cacat busuk

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI: 10.31515/telematika.v18i1.4025

2.1. Data Set

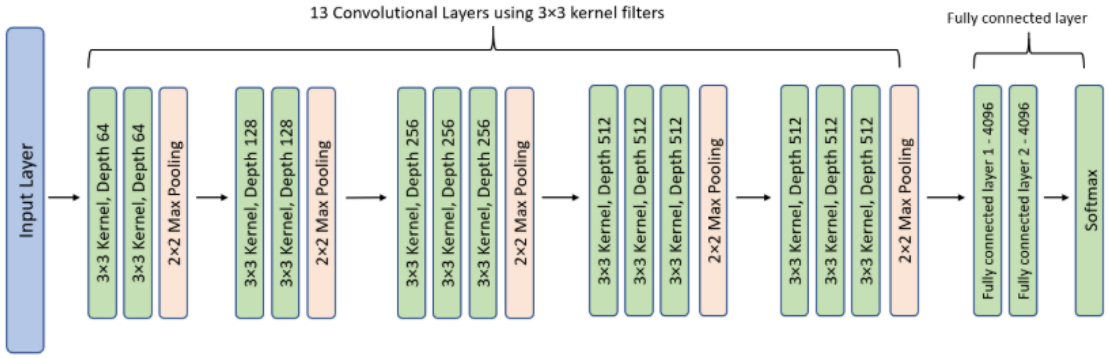
Data set yang digunakan pada penelitian diambil dari data set penelitian sebelumnya tentang klasifikasi salak [6], [7]. Data set ini sejumlah 370 citra yang terdiri dari 190 citra dari kelas bagus, dan 180 citra dari kelas jelek. Citra dari kelas jelek diambil dari salak yang terlalu matang, mengandung cacat busuk, dan mengandung cacat sobek. Pemilihan citra ini dilakukan oleh ahli, dimana dalam hal ini ahli merupakan eksportir buah salak. Contoh citra dari kelas bagus, jelek karena cacat sobek dan cacat busuk ditunjukkan pada **Gambar 3**. Contoh citra dari kelas jelek karena terlalu matang ditunjukkan pada **Gambar 1b**.

2.2. Preprocessing

Preprocessing dilakukan dengan melakukan pengubahan ukuran citra. Ukuran citra dirubah menjadi 224x224x3 pixel. Karena fitur warna merupakan fitur penting dalam pengenalan ini, maka citra tidak dikonversi ke citra greyscale. Selain pengubahan ukuran citra, nilai pixel dari citra juga dinormalisasi menjadi nilai antara 0 dan 1. Setelah dilakukan normalisasi, citra dibagi menjadi 80% data training dan 20% data testing.

2.3. VGG16

Setelah dilakukan pembagian data set, citra dilatih dengan menggunakan arsitektur VGG16. VGG16 adalah arsitektur deep neural network yang terdiri dari 16 layer. Arsitektur VGG16 ditunjukkan pada **Gambar 4**. VGG16 mempunyai 13 layer lapisan konvolusi, 2 lapisan yang fully connected, dan 1 lapisan classifier.



Gambar 4. Arsitektur VGG16 [9]

Pada Gambar 4, semua lapisan konvolusi mempunyai ukuran kernel 3x3. Perbedaan utama tiap lapisan konvolusi terletak pada jumlah filter di masing-masing lapipsan. 2 lapisan konvolusi pertama mempunyai jumlah filter 64. Sementara itu lapisan 3 dan 4 mempunyai jumlah kernel 128. Begitu juga lapisan konvolusi lain mempunyai jumlah filter berbeda-beda, yaitu 256 (lapisan 4,5,6) dan 512 (lapisan 7,8,9,10,11,12). 2x2 max pooling dilakukan setelah lapisan konvolusi 2, 4, 7, 10 dan 13. Keluaran dari pooling terakhir dihubungkan ke fully connected layer, dan pada akhirnya akan terhubung ke classifier untuk menentukan kelas dari citra.

Dalam penelitian ini, classifier di lapisan terakhir menggunakan *sigmoid classifier*. Optimasi dilakukan dengan menggunakan *Stochastic Gradient Descent*(SGD). SGD dipilih karena, walaupun sederhana, SGD mampu memberikan performa lebih baik dibandingkan *optimizer* lain seperti Adam pada pelatihan *deep learning* [15]. SGD sendiri adalah metode optimasi

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI: 10.31515/telematika.v18i1.4025

berulang yang secara acak mengambil satu sampel selama iterasi k, dan menggunakan sample tersebut untuk menghitung gradien. *Stochastic Gradient* yang dihasilkan kemudian digunakan untuk memperbarui bobot jaringan sesuai dengan laju pembelajaran atau *learning rate* yang digunakan. [16]

Dengan menggunakan SGD, perbaruan bobot pada lapisan VGG16 dilakukan sebagaimana ditunjukkan pada (1), dimana w adalah bobot awal, dan g adalah gradien. *Learning rate* adalah seberapa cepat lompatan yang ingin dilakukan dalam perbaruan bobot. *Learning rate* yang terlalu kecil dapat mengakibatkan pelatihan menjadi lama. Akan tetapi jika *learning rate* yang digunakan terlalu besar, dapat mengakibatkan pelatihan tidak dapat mencapai nilai optimal.

$$w = w - learning\_rate * g \quad (1)$$

Sementara itu, dalam metode SGD juga dapat digunakan momentum. Penggunaan momentum membantu mempercepat vektor gradien ke arah yang benar, sehingga pelatihan menjadi lebih cepat konvergen. Jika pelatihan penggunakan momentum, maka perbaruan bobot dilakukan dengan menggunakan (2). Nilai momentum yang banyak digunakan adalah 0.9. [17]

$$velocity = momentum * velocity - learning\_rate * g \quad (2)$$
$$w = w + velocity$$

Dalam penelitian ini, penelitian akan dilakukan dengan menggunakan momentum dan tanpa momentum. Untuk masing-masing skenario, *learning rate* akan dirubah antara 0,00001 sampai 0,001. Sementara itu jumlah *epoch* akan divariasikan antara 100 dan 200 *epoch*.

2.4. Evaluasi model

Evaluasi model dilakukan dengan memonitor jumlah *true positif, true negatif, false positif*, dan *false negatif*. Dari data-data tersebut, dapat dihitung akurasi, precision, dan recall dari model yang dihasilkan. True positif adalah data salak kelas bagus yang diprediksi ke kelas bagus. True negatif adalah data kelas jelek yang diprediksi ke kelas jelek. *False positif* adalah data kelas jelek yang diprediksi ke kelas bagus. Sementara itu false negatif adalah data kelas bagus yang diprediksi ke kelas jelek.

Dari nilai tersebut, akurasi, precision dan recall bisa dihitung sebagai mana persamaan (3)(4) dan (5).

$$Akurasi = (tp + tn)/(tp + tn + fp + fn) \quad (3)$$

$$precision = (tp)/(tp + fp \quad (4)$$

$$Recall = tp/(tp + fn) \quad (5)$$

2.5. Skenario penelitian

Skenario yang dilakukan pada penelitian ini adalah sebagai berikut:

2.5.1. Skenario 1. Pelatihan dengan momentum = 0

Dalam skenario ini, pelatihan dilakukan dengan menggunakan *learning rate* dari 0.00001 sampai 0.001. untuk masing-masing *learning rate*, *epoch* yang digunakan adalah 100 dan 200

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI: 10.31515/telematika.v18i1.4025

2.5.2. Skenario 2. Pelatihan dengan momentum = 0.9

Dalam skenario ini, pelatihan dilakukan dengan menggunakan *learning rate* dari 0.00001 sampai 0.001. untuk masing-masing *learning rate*, *epoch* yang digunakan adalah 100 dan 200

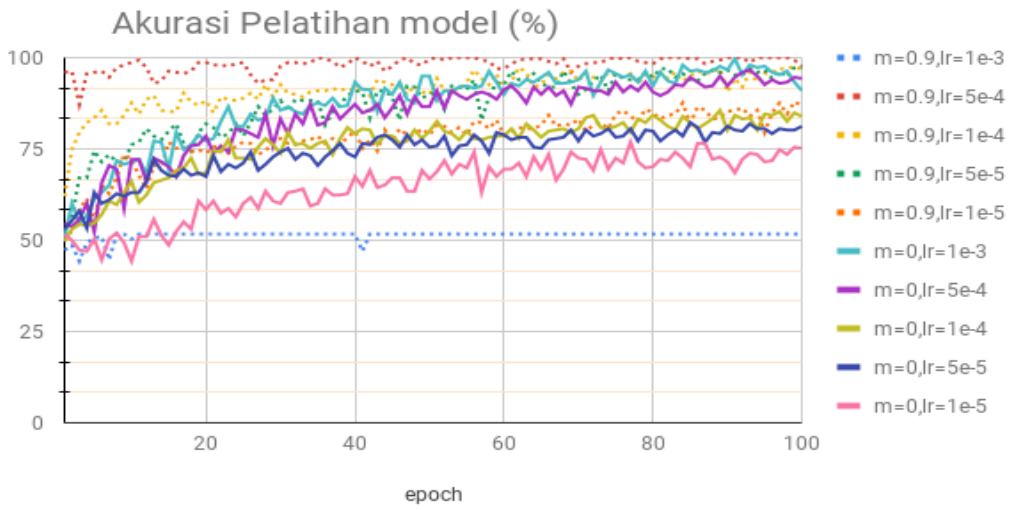
3. HASIL DAN PEMBAHASAN

Proses pelatihan untuk kedua skenario dengan menggunakan *epoch* 100 ditunjukkan pada **Gambar 5**. Pelatihan tanpa menggunakan momentum dengan nilai *learning rate* 0,00001 ditunjukkan dengan garis berkelanjutan dengan warna merah muda. Pelatihan dengan variabel tersebut mencapai akurasi pelatihan 75% pada *epoch* 100. Pelatihan dengan menggunakan *learning rate* lebih tinggi ditunjukkan dengan garis berkelanjutan dengan warna hijau tua (0,0005), hijau (0,0001), ungu (0,0005) dan biru muda(0,001). Pelatihan dengan *learning rate* paling besar yaitu 0,001 ditunjukkan dengan warna biru muda dan garis berkelanjutan. Pelatihan ini mencapai akurasi lebih dari 98% dengan jumlah *epoch* yang sama. Dari hasil tersebut, dapat dilihat bahwa nilai *learning rate* yang digunakan sangat mempengaruhi kecepatan pelatihan. Hal ini dapat dijelaskan karena perubahan bobot yang terjadi menjadi lambat *ketika learning rate* yang digunakan kecil. Meningkatkan nilai *learning rate* membuat perubahan bobot menjadi semakin cepat. Akan tetapi jika nilai *learning rate* terlalu besar, maka perubahan bobot yang terjadi juga semakin besar. Hal ini mengakibatkan model tidak mampu mencapai konvergen.

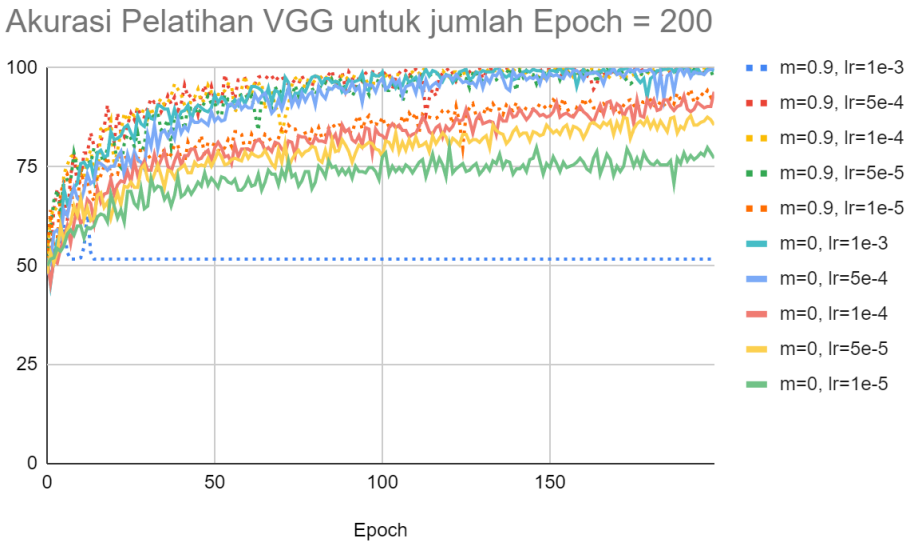
Pada **Gambar 5** dapat juga dilihat bahwa untuk learning rate yang kecil, penggunaan momentum mampu mempercepat proses pelatihan. Pelatihan dengan menggunakan nilai momentum sebesar 0,9 untuk ditunjukkan dengan garis putus-putus pada grafik dengan warna merah muda (0,00001), biru tua (0,00005), hijau (0,0001), ungu (0,0005), dan biru muda (0,001). Dibandingkan dengan nilai *learning rate* yang sama pada garis yang berkelanjutan, garis putus-putus mencapai akurasi lebih tinggi. Sebagai contoh, untuk nilai *learning rate* 0,00001 dengan momentum 0,9 mampu mencapai akurasi pelatihan sebesar 87%. Dibandingkan dengan ketika tanpa menggunakan momentum, akurasi ini lebih tinggi. Akan tetapi untuk nilai *learning rate* yang cukup besar, dalam hal ini 0.001 atau garis putus-putus warna biru muda, penggunaan momentum justru membuat model tidak mampu melakukan pelatihan dengan optimal. Hal ini dikarenakan perubahan bobot yang terjadi setiap *epoch* akan terlalu besar, karena merupakan fungsi dari *learning rate* dan momentum sebagaimana ditunjukkan pada persamaan (2).

Hasil pelatihan untuk jumlah *epoch* 200 ditunjukkan pada **Gambar 6**. Penambahan jumlah *Epoch* menjadi 200 membuat model dengan *learning rate* yang lebih kecil menjadi semakin konvergen. Hal ini ditunjukkan pada semua garis berkelanjutan pada grafik yang merepresentasikan akurasi pelatihan tanpa momentum untuk semua *learning rate.* Sementara itu, pelatihan dengan menggunakan momentum ditunjukkan dengan garis putus-putus pada grafik. Hasil pelatihan dengan momentum tersebut menunjukkan menunjukkan hal yang sama sebagaimana pelatihan tanpa momentum, kecuali untuk nilai *learning rate* 0,001 yang ditunjukkan dengan garis putus-putus berwarna biru pada grafik. Penggunaan momentum dan *learning rate* 0,001 dalam pelatihan mengakibatkan perubahan gradient yang terlalu besar, sehingga sistem tidak mampu melakukan pembelajaran dengan efektif. Model yang lebih konvergen dalam pelatihan diharapkan akan menghasilkan akurasi yang lebih tinggi dalam evaluasi dengan data testing.

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI: 10.31515/telematika.v18i1.4025



Gambar 5. Pelatihan model VGG 16 dengan menggunakan momentum dan tanpa momentum untuk *Epoch* 100



Gambar 6. Pelatihan model VGG 16 dengan menggunakan momentum dan tanpa momentum untuk *Epoch* 200

Akurasi pengujian dengan menggunakan scenario 1 dan 2 ditunjukkan pada **Tabel 2**. Pada tabel tersebut dapat dilihat bahwa akurasi pengujian tertinggi pada skenario 1 didapatkan ketika menggunakan *learning rate* 0,001 dengan jumlah *epoch* 200. Pada skenario 2, hasil akurasi tertinggi didapatkan ketika menggunakan *learning rate* = 0,0001, dengan jumlah *epoch* 100. Hal ini sesuai juga dengan apa yang ditunjukkan pada **Gambar 5** dan **Gambar 6**, dimana untuk kedua nilai *learning rate* tersebut telah mencapai pelatihan yang konvergen.

Tabel 2. Akurasi model hasil pelatihan dan pengujian dengan VGG-16

	Momentum= 0				**Momentum=0.9**			
	***Epoch*=100**		***Epoch*=200**		***Epoch*=100**		***Epoch*=200**	
	Train accuracy (%)	**Test accuracy (%)**	**Train accuracy (%)**	**Test accuracy (%)**	**Train accuracy (%)**	**Test accuracy (%)**	**Train accuracy (%)**	**Test accuracy (%)**
Lr=0,001	90,94	90,28	99,32	**95,83**	51,67	50	51,67	50
Lr=0,0005	94,3	94,44	100	91,67	97,65	88,88	100	94,4
Lr=0,0001	83,89	80’54	93,95	87,5	98,66	**95,83**	100	93,06
Lr=0,00005	81,21	80,56	85,57	83,3	97,31	84,72	97,31	93,06
Lr=0,00001	75,17	75	77,18	75	86,91	83,33	93,96	87,5

Confusion matrix untuk akurasi tertinggi ditunjukkan pada **Tabel 3**. *True positif, true negative, false positive* , dan *false negative* dari model yang dihasilkan adalah sejumlah 35, 34, 1, dan 2. Dari nilai-nilai tersebut dapat dihitung precision dan recall dari model terbaik adalah 97,2% dan 94,6% sebagaimana ditunjukkan pada perhitungan sesuai dengan persamaan (3), (4), dan (5). Nilai *precision* 97,2% berarti dari semua data yang diprediksi masuk ke kelas bagus, 97,2% data tersebut adalah memang benar dari kelas bagus. Sementara nilai *recall* 94,6% berarti bahwa semua data pada kelas bagus, 94,6% akan diprediksi secara benar.

$$precision=(tp)/(tp+fp)=35/36=97.2\%$$

$$Recall = tp/(tp+fn)=35/37=94.6\%$$

Tabel 3. *Confusion matrix* terbaik hasil pelatihan dengan VGG-16

Kelas sebenarnya / Prediksi	**Bagus**	**Jelek**
Bagus	35	1
Jelek	2	34

Dari hasil penelitian ini, dapat dilihat bahwa pelatihan menggunakan *transfer learning* dengan arsitektur VGG16 mampu melakukan klasifikasi kualitas buah salak dengan akurasi 95,83%.

4. KESIMPULAN

Penelitian untuk membedakan kualitas buah salak telah dilakukan. Salak dibagi menjadi dua kelas, yaitu kelas bagus dan kelas jelek. Salak dengan kualitas jelek merupakan salak yang terlalu matang, mengandung cacat busuk, dan mengandung cacat robek. Jumlah data yang digunakan dalam penelitian ini adalah 370 data, yang terdiri dari 190 salak kelas bagus, dan 180 salak kelas jelek. Pelatihan dilakukan menggunakan *transfer learning* dengan arsitektur VGG16. Dalam melakukan pelatihan, parameter yang dirubah adalah *learning rate, epoch* dan momentum. Hasil penelitian menunjukkan, arsitektur ini mampu mendapatkan akurasi tertinggi sebesat 95,83% dengan menggunakan *learning rate* =0,0001 dan momentum=0,9. Jika tanpa

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI: 10.31515/telematika.v18i1.4025

menggunakan momentum, maka akurasi terbaik didapatkan dengan menggunakan *learning rate*=0,001. Penelitian selanjutnya dapat dilakukan untuk meneliti metode-metode *transfer learning* lain untuk pengenalan citra ini.

Daftar Pustaka

[1] BPS, "Data produksi buah holtikultura salak di Indonesia," 2019. [Online]. Available: bps.go.id. [Accessed: 23-Aug-2019].

[2] L. Marifatul Azizah, S. Fadillah Umayah, and F. Fajar, "Deteksi Kecacatan Permukaan Buah Manggis Menggunakan Metode Deep Learning dengan Konvolusi Multilayer," *Semesta Tek.*, vol. 21, no. 2, pp. 230–236, 2018.

[3] A. Bhargava and A. Bansal, "Automatic Detection and Grading of Multiple Fruits by Machine Learning," *Food Anal. Methods*, vol. 13, no. 3, pp. 751–761, 2020.

[4] W. D. N. Pacheco and F. R. J. López, "Tomato classification according to organoleptic maturity (coloration) using machine learning algorithms K-NN, MLP, and K-Means Clustering," *2019 22nd Symp. Image, Signal Process. Artif. Vision, STSIVA 2019 - Conf. Proc.*, 2019.

[5] A. Pande, M. Munot, R. Sreeemathy, and R. V. Bakare, "An Efficient Approach to Fruit Classification and Grading using Deep Convolutional Neural Network," *2019 IEEE 5th Int. Conf. Converg. Technol. I2CT 2019*, pp. 1–7, 2019.

[6] Rismiyati and H. A. Wibawa, "Snake Fruit Classification by Using Histogram of Oriented Gradient Feature and Extreme Learning Machine," *ICICOS 2019 - 3rd Int. Conf. Informatics Comput. Sci. Accel. Informatics Comput. Res. Smarter Soc. Era Ind. 4.0, Proc.*, pp. 2–6, 2019.

[7] Rismiyati and S. N. Azhari, "Convolutional Neural Network implementation for image-based Salak sortation," *Proc. - 2016 2nd Int. Conf. Sci. Technol. ICST 2016*, pp. 77–82, 2017.

[8] K. Wang, X. Gao, Y. Zhao, X. Li, D. Dou, and C.-Z. Xu, "Pay Attention to Features, Transfer Learn Faster CNNs," *8th Int. Conf. Learn. Represent. ICLR 2020 - Conf. Track Proc.*, pp. 1–14, 2020.

[9] S. Tammina, "Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images," *Int. J. Sci. Res. Publ.*, vol. 9, no. 10, p. p9420, 2019.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "ImageNet: A Large-Scale Hierarchical Image Database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 248–255.

[11] A. Krishnaswamy Rangarajan and R. Purushothaman, "Disease Classification in Eggplant Using Pre-trained VGG16 and MSVM," *Sci. Rep.*, vol. 10, no. 1, pp. 1–11, 2020.

[12] Y. Gultom, A. M. Arymurthy, and R. J. Masikome, "Batik Classification using Deep Convolutional Network Transfer Learning," *J. Ilmu Komput. dan Inf.*, vol. 11, no. 2, p.

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI: 10.31515/telematika.v18i1.4025

59, 2018.

[13] T. Kaur and T. K. Gandhi, "Automated brain image classification based on VGG-16 and transfer learning," *Proc. - 2019 Int. Conf. Inf. Technol. ICIT 2019*, pp. 94–98, 2019.

[14] B. Liu, X. Zhang, Z. Gao, and L. Chen, "Weld defect images classification with VGG16-based neural network," *Commun. Comput. Inf. Sci.*, vol. 815, pp. 215–223, 2018.

[15] P. Zhou, J. Feng, C. Ma, C. Xiong, H. O. I. Steven, and E. Weinan, "Towards theoretically understanding why SGD generalizes better than ADAM in deep learning," *arXiv*, no. 1, pp. 19–21, 2020.

[16] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv:1609.04747v2*, pp. 1–14.

[17] V. Bushaev, "Stochastic Gradient Descent with momentum," 2017. [Online]. Available: https://towardsdatascience.com/stochastic-gradient-descent-with-momentum-a84097641a5d. [Accessed: 30-Nov-2020].