

# Peningkatan Performa MobilenetV3 dengan *Squeeze-and-Excitation* (Studi Kasus Klasifikasi Kesegaran Ikan Berdasarkan Mata Ikan)

**GALIH ASHARI RAKHMAT, MUHAMMAD FIKRI HAEKAL**

Program Studi Informatika, Institut Teknologi Nasional Bandung  
Email: [galihas@itenas.ac.id](mailto:galihas@itenas.ac.id)

*Received* 23 Desember 2022 | *Revised* 13 Februari 2023 | *Accepted* 13 Maret 2023

## **ABSTRAK**

*Ikan mengandung banyak nutrisi, protein tinggi yang memiliki banyak manfaat untuk tubuh manusia. Hal tersebut, membuat banyak masyarakat mengkonsumsi ikan sebagai makanan sehari – hari. Sehingga diperlukan ketelitian masyarakat ketika membeli ikan menjadi perhatian serius dalam memilih ikan terkait kesegarannya. Penelitian dilakukan dengan mengklasifikasikan 24 kelas kesegaran ikan berdasarkan mata ikan dengan menguji performa arsitektur MobileNetV3 dengan Squeeze-and-excitation. Hasil yang didapatkan model dengan kinerja terbaik diperoleh pada hyperparameter learning rate 0.00001, batch size train 10 val 10, optimizer ADAM, epochs 100 dengan model arsitektur MobileNetV3-Small. berdasarkan hasil evaluasi performa model didapatkan 68% accuracy, 69% precision, 67% recall dan 68% f1-score pada pengujian 876 data uji dengan 24 kelas yaitu terdiri dari tingkat kesegaran dan jenis ikan yang berbeda.*

**Kata kunci:** *MobileNetV3, Hyperparameter, Mata Ikan, Klasifikasi*

## **ABSTRACT**

*Fish contains many nutrients, high protein which has many benefits for the human body. This, makes many people consume fish as daily food. So that people need to be careful when buying fish to be a serious concern in choosing fish related to its freshness. The research was conducted by classifying 24 classes of fish freshness based on fish eye by testing the performance of the MobileNetV3 architecture with Squeeze-and-excitation. The results obtained for the model with the best performance were obtained at hyperparameter learning rate 0.00001, batch size train 10 val 10, ADAM optimizer, epochs 100 with the MobileNetV3-Small architectural model. Based on the results of the model performance evaluation, it obtained 68% accuracy, 69% precision, 67% recall and 68% f1-score on 876 test data with 24 classes consisting of different levels of freshness and types of fish.*

**Keywords:** *MobileNetV3, Hyperparameter, Fish Eye, Classification*

## 1. PENDAHULUAN

Ikan mengandung banyak nutrisi, protein tinggi yang memiliki manfaat bagi tubuh manusia, kemudian, ikan juga memiliki kandungan air yang tinggi. Ikan yang baik adalah ikan yang masih segar, sehingga banyak disukai oleh konsumen. Penanganan dan sanitasi yang baik sangat diperlukan untuk tetap menjaga kesegaran ikan, makin lama berada di udara terbuka maka makin menurun kesegarannya. Kualitas ikan yang menurun bisa dilihat dari perubahan pada mata ikan. Kesegaran ikan merupakan tolak ukur ikan itu baik atau buruk (**Diana, 2013**).

Tingginya protein dan nutrisi yang terdapat di ikan membuat banyak orang mengonsumsi ikan sebagai makanan sehari-hari. Hal ini menyebabkan peningkatan konsumsi ikan dengan kebutuhan tinggi bagi masyarakat maka ketelitian masyarakat ketika membeli ikan menjadi perhatian serius dalam memilih ikan. Pada umumnya, ikan yang dijual di pasar adalah ikan segar, tetapi tidak jarang ditemukan ikan tidak segar yang sudah mengalami pengawetan dengan es selama beberapa hari (**Prasetyo et al., 2022**).

Deep learning adalah metode *machine learning*, yang terdiri dari *input layer*, *hidden layer* dan *output layer*. Teknik pengenalan sebelum *deep learning* mengklasifikasikan tugas melalui informasi fitur semantik, beberapa contoh fitur semantik adalah sudut, tepi, dan bentuk. *Deep learning* telah banyak digunakan untuk kasus klasifikasi citra, dimana *model* akan mengambil *input* dalam bentuk citra dan *output* yang dihasilkan berupa hasil prediksi atau klasifikasi dari citra input. Pada dasarnya algoritma *deep learning* menggunakan *neural network* untuk menentukan koneksi antara *input* dan *output* (**Zaki et al., 2020**).

Arsitektur yang akan digunakan adalah *MobileNet* yang memiliki tujuan untuk menciptakan *model* arsitektur yang ringan. sehingga, dapat diterapkan pada sistem *mobile*. Peneliti akan melakukan Implementasi 2 *model* dari arsitektur *MobileNet* yaitu *model MobileNetV3-Large* dan *MobileNetV3-Small*. *Model* tersebut akan disesuaikan dengan kebutuhan sistem. Perbedaan *model* tersebut terletak pada jumlah *layer* yang digunakan, pada *MobileNetV3-Large* lapisan kovolusi adalah sebanyak 15 *layer*, sedangkan *MobileNetV3-Small* sebanyak 11 *layer* (**Qian et al., 2021**).

Berdasarkan penelitian sebelum yang dilakukan oleh (**Prasetyo et al., 2022**), didapatkan hasil jumlah parameter pada *model*/MB-BE sebanyak 3.16 juta. Sehingga, peneliti berkeinginan untuk melakukan implementasi arsitektur *MobileNetV3-Large* dan *MobileNetV3-Small* pada Arsitektur *MobileNetV3* yang diharapkan dapat menurunkan jumlah parameter pada *model*. Pada studi kasus klasifikasi kesegaran ikan. *Future work* dari penelitian sebelum yaitu hasil yang diperoleh dari kombinasi metode *MobileNet* dengan *Bottleneck Expansion (MB-BE)* memiliki akurasi di bawah 70% yaitu 63,21%, oleh karena itu, dibutuhkan metode *MobileNet* untuk meningkatkan performa dari sisi akurasi dan penurunan jumlah parameter *model*.

Tujuan penelitian ini untuk melanjutkan penelitian, melakukan kontribusi penelitian pada penelitian sebelum yang dilakukan oleh (**Prasetyo et al., 2022**) dengan menerapkan arsitektur *MobileNetV3* sebagai *improvement* dari metode *MobileNet* dengan melakukan implementasi dari 2 *model* dari arsitektur *MobileNetV3* yaitu *MobileNetV3-Large* dan *MobileNetV3-Small*. Klasifikasi yang dilakukan pada penelitian sebanyak 3 kelas tingkat kesegaran ikan yaitu ikan sangat segar, ikan segar, dan ikan tidak segar dengan 8 jenis ikan.

## 2. METODE PENELITIAN

### 2.1. MobileNetV3

*MobileNetV3* merupakan arsitektur CNN yang termasuk efisien dan hasil peningkatan dari versi sebelum yakni *MobileNetV1* dan *MobileNetV2*. Pada *MobileNetV1*, fitur pertama yang ditambahkan merupakan *Depthwise Separable Convolution* serta penggunaan *Batch Normalization* (A. G. Howard et al., 2017), yang dimana kedua metode tersebut diajukan dikarenakan dapat memangkas komputasi dan mempercepat proses *training* sebuah *model*, lalu pada *MobileNetV2* pengembangan performa dilakukan dengan penambahan *Linear Bottlenecks* dan *Inverted Residual* (Sandler et al., 2018), peningkatan performa dari kedua metode tersebut ada pada pemangkasan *information loss* dari sebuah aktivasi *linear* dan efisiensi pemakaian memori, Keunggulan yang ditawarkan oleh arsitektur ini merupakan akurasi yang tinggi dengan latensi yang rendah. Pada intinya *MobileNetV3* merupakan arsitektur yang efisien dan ringan yang dapat digunakan pada perangkat *mobile* (Chu et al., 2020). *MobileNetV3* terbagi menjadi 2 jenis yaitu *MobileNetV3-Large* dan *MobileNetV3-Small* yang dimana letak perbedaannya yaitu pada jumlah dan jenis *layer* yang dipakai (A. Howard et al., 2019). Gambar 1 menjelaskan arsitektur *MobilenetV3-Large* dan Gambar 2 menjelaskan arsitektur *MobilenetV3-Small*.

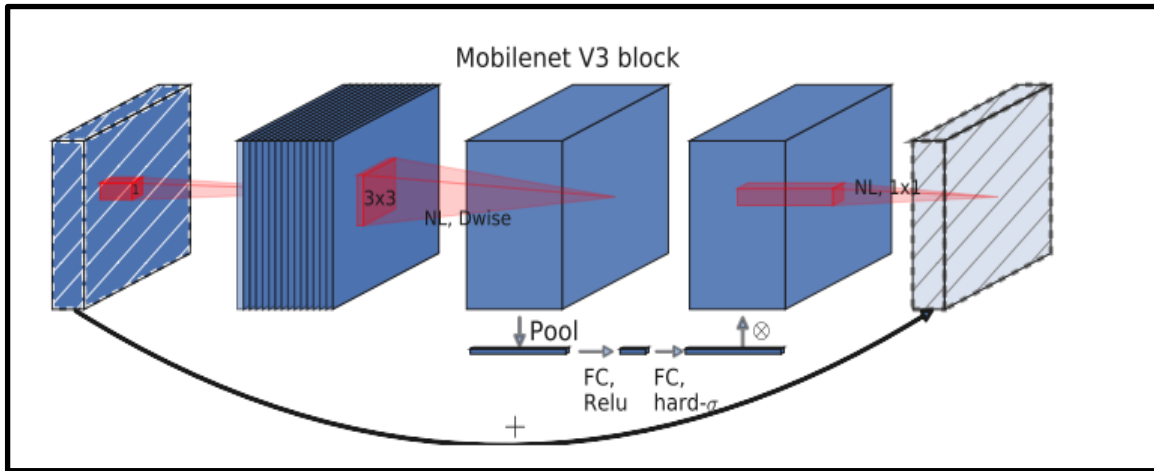
Input	Operator	exp size	#out	SE	NL	s
224 <sup>2</sup> × 3	conv2d	-	16	-	HS	2
112 <sup>2</sup> × 16	bneck, 3x3	16	16	-	RE	1
112 <sup>2</sup> × 16	bneck, 3x3	64	24	-	RE	2
56 <sup>2</sup> × 24	bneck, 3x3	72	24	-	RE	1
56 <sup>2</sup> × 24	bneck, 5x5	72	40	✓	RE	2
28 <sup>2</sup> × 40	bneck, 5x5	120	40	✓	RE	1
28 <sup>2</sup> × 40	bneck, 5x5	120	40	✓	RE	1
28 <sup>2</sup> × 40	bneck, 3x3	240	80	-	HS	2
14 <sup>2</sup> × 80	bneck, 3x3	200	80	-	HS	1
14 <sup>2</sup> × 80	bneck, 3x3	184	80	-	HS	1
14 <sup>2</sup> × 80	bneck, 3x3	184	80	-	HS	1
14 <sup>2</sup> × 80	bneck, 3x3	480	112	✓	HS	1
14 <sup>2</sup> × 112	bneck, 3x3	672	112	✓	HS	1
14 <sup>2</sup> × 112	bneck, 5x5	672	160	✓	HS	2
7 <sup>2</sup> × 160	bneck, 5x5	960	160	✓	HS	1
7 <sup>2</sup> × 160	bneck, 5x5	960	160	✓	HS	1
7 <sup>2</sup> × 160	conv2d, 1x1	-	960	-	HS	1
7 <sup>2</sup> × 960	pool, 7x7	-	-	-	-	1
1 <sup>2</sup> × 960	conv2d 1x1, NBN	-	1280	-	HS	1
1 <sup>2</sup> × 1280	conv2d 1x1, NBN	-	k	-	-	1

Gambar 1. Arsitektur *MobileNetV3-Large* (Howard A. S., 2019)

Input	Operator	exp size	#out	SE	NL	s
224 <sup>2</sup> × 3	conv2d, 3x3	-	16	-	HS	2
112 <sup>2</sup> × 16	bneck, 3x3	16	16	✓	RE	2
56 <sup>2</sup> × 16	bneck, 3x3	72	24	-	RE	2
28 <sup>2</sup> × 24	bneck, 3x3	88	24	-	RE	1
28 <sup>2</sup> × 24	bneck, 5x5	96	40	✓	HS	2
14 <sup>2</sup> × 40	bneck, 5x5	240	40	✓	HS	1
14 <sup>2</sup> × 40	bneck, 5x5	240	40	✓	HS	1
14 <sup>2</sup> × 40	bneck, 5x5	120	48	✓	HS	1
14 <sup>2</sup> × 48	bneck, 5x5	144	48	✓	HS	1
14 <sup>2</sup> × 48	bneck, 5x5	288	96	✓	HS	2
7 <sup>2</sup> × 96	bneck, 5x5	576	96	✓	HS	1
7 <sup>2</sup> × 96	bneck, 5x5	576	96	✓	HS	1
7 <sup>2</sup> × 96	conv2d, 1x1	-	576	✓	HS	1
7 <sup>2</sup> × 576	pool, 7x7	-	-	-	-	1
1 <sup>2</sup> × 576	conv2d 1x1, NBN	-	1280	-	HS	1
1 <sup>2</sup> × 1280	conv2d 1x1, NBN	-	k	-	-	1

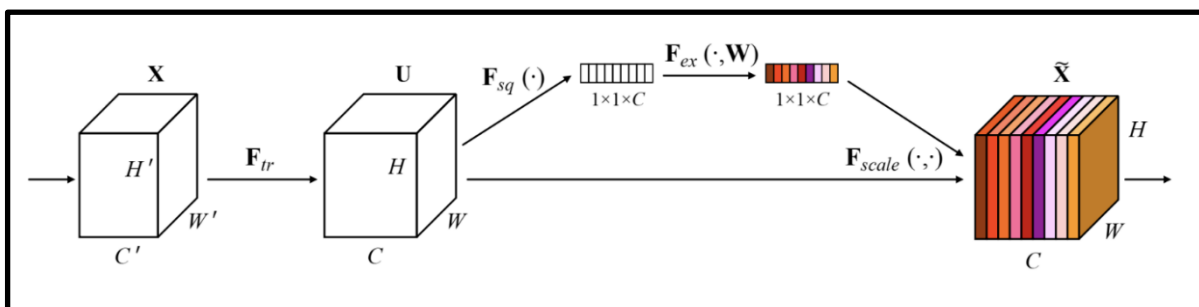
Gambar 2. Arsitektur *MobileNetV3-Small* (Howard A. S., 2019)

SE merupakan singkatan dari *Squeeze-And-Excite*. *Exp size* merupakan singkatan dari *expansion size*, *#out* merupakan jumlah *filter* yang dipakai. NL merupakan singkatan dari *nonlinearity*. Lalu, HS merupakan singkatan dari aktivasi *hard-swish* dan RE merupakan singkatan aktivasi *Rectified Linear Unit (ReLU)*. NBN merupakan singkatan dari *no batch normalization*, dan *s* merupakan singkatan dari *stride*.



Gambar 3. *MobileNetV3 Block* (Howard A. S., 2019)

Pada Gambar 3 yaitu penjelasan mengenai gambaran arsitektur *MobileNetV3*. Tahapan pertama merupakan proses input citra dengan konvolusi2d, *kernel* 3x3, *stride* 2, dengan fungsi aktivasi *hard-swish* pada citra ukuran 224x224 dengan 3 *channel* warna yaitu *red, green, and blue*. Lalu selanjutnya dilakukan proses konvolusi dengan *bottleneck* 3x3 yang dimana proses konvolusi diawali dengan konvolusi 1x1 pada input untuk mengurangi *depth* lalu konvolusi 3x3 dapat lebih ringan memproses fitur konvolusi karena *depth* lebih rendah dan konvolusi 1x1 untuk *output* agar *depth* dapat kembali seperti *input*, lalu menggunakan *ReLU* sebagai fungsi aktivasi. Tujuan utama dari proses *bottleneck* tersebut adalah peningkatan efisiensi *model*. Lalu *layer* berikutnya merupakan fungsi *bottleneck* yang sama sebanyak 11 kali untuk *MobileNetV3-Small* dan 15 kali untuk *MobileNetV3-Large*, pada setiap *layer* tersebut hanya terdapat perbedaan pada parameter *stride*, fungsi aktivasi dan ukuran *kernel*. Kemudian terdapat proses *pooling* 7x7 yang dimana berfungsi untuk mengurangi dimensi dari *feature maps*, yang dimana mengurangi jumlah parameter yang dapat dipelajari dan mengurangi jumlah komputasi yang dilakukan pada *mobile*. Setelah itu terdapat konvolusi2d *kernel* 1x1 dan menggunakan *no batch normalization* dan aktivasi menggunakan *hard-swish*. Pada *layer* akhir, terdapat *layer* yang sama akan tetapi penggunaan berbeda karena *layer* ini digunakan untuk *output* citra.



Gambar 4. *Squeeze-And-Excitation Block* (Hu et al., 2018)

Gambar 4 menjelaskan terkait *SE block* yang merupakan fitur optimisasi arsitektur baru yang diperkenalkan pada *MobileNetV3* yang bekerja sebagai representasi sebuah nilai keseluruhan pada sebuah *channel* dengan proses sebagai berikut:

- *Input* berupa *convolution block*.
- Setiap *channel* melalui proses “*squeeze*” dan menjadi nilai numerik tunggal melalui *average pooling*.
- Sebuah *dense layer* dengan *ReLU* menambahkan sifat *non-linear* dan kompleksitas *channel output* dikurangi oleh rasio.
- Penambahan *dense layer* dengan fungsi aktivasi *sigmoid* ditambahkan kepada setiap *channel*.
- Terakhir, *weight* dari setiap fitur konvolusi ditambahkan berdasarkan jaringan yang sudah dibuat pada *squeeze*.

## 2.2. Data Batch Size

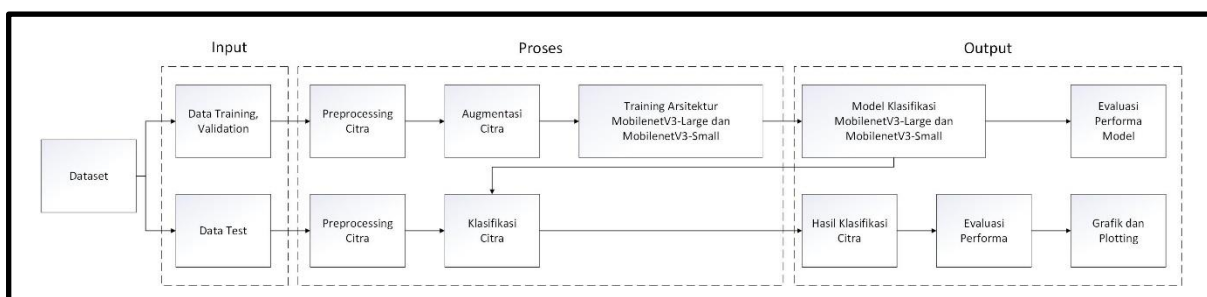
*Batch Size* adalah *hyperparameter* yang mewakili angka dari sampel data latih yang akan digunakan selama pelatihan model. *Batch Size* merupakan salah satu *parameter* yang dapat mempengaruhi *overfitting* dan waktu komputasi saat pembuatan *model*. Permasalahan *overfitting* seringkali terjadi saat proses evaluasi *model* pada *CNN*, agar tidak terjadi hal tersebut, perlu diperhatikan pada beberapa faktor berikut, seperti *dataset* latih yang banyak dari jumlah data dan variasi data, penambahan *regularizer* seperti *dropout rate* (Ying, 2019). dan *hyper-parameter* seperti *learning rate* yang optimal (Radiuk, 2017). Adapun solusi yang didapat, menurut penelitian (Kandel & Castelli, 2020) yang membahas perihal pengaruh *batch size* pada *dataset*, menjelaskan bahwa *batch size* dapat mengurangi *overfitting* pada hasil evaluasi *model*.

## 2.3. Learning Rate

*Learning rate* merupakan salah satu *hyperparameter* pelatihan yang digunakan untuk melakukan *training* pada *dataset*. *Learning rate* ditetapkan untuk menghitung nilai dari koreksi bobot saat melakukan proses *training*. Pada saat melakukan pemilihan nilai *learning rate*, apabila terlalu kecil dapat membutuhkan waktu yang lebih lama dan sebaliknya apabila terlalu besar maka pembelajaran akan menjadi kurang optimal karena proses pelatihan yang terlalu cepat dan proses pelatihan menjadi tidak stabil.

## 2.4. Block Diagram Sistem

Pada proses perancangan sistem klasifikasi kesegaran ikan menggunakan *MobileNetV3-Large* dan *MobileNetV3-Small* terdapat beberapa tahapan yang dilakukan untuk membuat sebuah *model* pada penelitian ini pada Gambar 5 merupakan *block diagram* sistem pada penelitian.



Gambar 5. Block Diagram Sistem

*Input* berupa citra yang berasal dari *FFE Dataset* (Prasetyo et al., 2022). Pada data *training* dan data validasi, dilakukan *preprocessing* pada citra masukan dengan dilakukannya *resize* menjadi 224 x 224 dengan 3 buah *channel*/warna *RGB* (*Red*, *Green*, dan *Blue*) dan melakukan

proses augmentasi yaitu dengan *horizontal flip* dan *vertical flip*. Pada proses *pre-processing* ini menyesuaikan dengan format citra *model MobileNet*. Kemudian dilakukan augmentasi citra untuk menambahkan variasi pada *dataset*. Proses selanjutnya menggunakan *Convolutional Neural Network* arsitektur *MobileNetV3-Large* dan *MobileNetV3-Small* dengan operasi umum seperti *depthwise separable convolution*, *bottleneck block*, dan *squeeze exitation*. *Output* yang dihasilkan berupa hasil pengklasifikasian kesegaran ikan. Setelah pengujian dilakukan selanjutnya evaluasi model menggunakan *confusion matrix* lalu dilakukan *plotting* pada hasil yang didapatkan untuk mengetahui performa *model* yang dihasilkan.

## 2.4. Metrik Evaluasi

Evaluasi kinerja *model* klasifikasi yang telah dirancang dilakukan dengan menghitung tingkat akurasi, *sensitifity*, *specificity* dan *F-Measure* dari evaluasi *confusion matrix* sebagai pengujian performa model dari proses pengujian *dataset* (Agarwal et al., 2020). Pada *confussion matrix* terdapat 4 istilah sebagai berikut :

- *TP (True Positive)* : perkiraan banyaknya data positif dan terbukti kebenarannya.
- *FP (False Positive)* : perkiraan banyaknya data negatif dan terbukti kebenarannya.
- *TN (True Negative)* : perkiraan banyaknya data positif dan tidak terbukti kebenarannya.
- *FN (False Negative)* : perkiraan banyaknya data negatif dan tidak terbukti kebenarannya.

*Accuracy* : Rasio prediksi benar dengan total keseluruhan data, akurasi merupakan kriteria utama dalam evaluasi *model*. Perhitungan akurasi menggunakan Persamaan (1).

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \quad (1)$$

*Precision* : Bagian dari suatu objek diprediksi benar. Perhitungan *precision* menggunakan Persamaan (2).

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

*Recall* : Mengetahui seberapa akurat kinerja suatu model untuk mengklasifikasi benar atau menghitung banyaknya kekeliruan klasifikasi *false negative* pada *model*. Perhitungan *recall* menggunakan Persamaan (3).

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

*F-Score* : Perbandingan rata – rata harmonik antara nilai *precision* dan nilai *recall*. Perhitungan *F1Score* menggunakan Persamaan (4).

$$F1-Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

## 2.5. Dataset

*Dataset* citra mata ikan yang digunakan *Freshness of Fish Eyes Dataset (FFE)* (Prasetyo et al., 2022) yang dibuat oleh Eko Prasetyo, dan Rani Purbaningtyas pada tahun 2021. *Dataset* diambil dari *website* <https://data.mendeley.com/datasets/xzyx7pbr3w/1> Kumpulan data ini terdiri dari 4392 citra mata ikan, dengan 8 jenis ikan, dan masing-masing terdiri dari ikan yang sangat segar (hari 1 dan 2), ikan segar (hari ke 3 dan 4), dan ikan tidak segar (hari ke 5 dan 6). Setiap spesies dibagi menjadi tiga tingkat kesegaran sehingga jumlah kelas adalah 24 kelas. Akuisisi citra dilakukan dengan cara merawat ikan dengan penyimpanan dalam *styrofoam box*

selama enam hari. Ponsel digunakan untuk memotret ikan dengan berbagai latar belakang dan pencahayaan, setiap gambar berisi jenis ikan yang sama dengan jumlah ikan yang bervariasi.

*Dataset* citra ini dibagi menjadi 3 yaitu data pelatihan, data validasi dan data pengujian dengan persentase pembagian untuk data latih 60%, data validasi 20% dan data *test* 20% (**Prasetyo et al., 2022**) dari total keseluruhan *dataset* sebanyak 4392 data. Data pelatihan digunakan pada proses pelatihan. Proses berikutnya dilakukan validasi dengan penggunaan data validasi. Dan proses terakhir merupakan evaluasi data pengujian untuk menentukan tingkat prediksi *model*. Detil pembagian *dataset* ditunjukkan pada Tabel 1.

**Tabel 1. Pembagian *Dataset***

Kelas	Proses		
	Train	Validasi	Test
Ikan mujair sangat segar	173	58	58
Ikan mujair segar	104	35	35
Ikan mujair tidak segar	98	32	32
Ikan nila sangat segar	196	66	66
Ikan nila segar	139	46	46
Ikan nila tidak segar	148	49	49
Ikan bandeng sangat segar	100	34	34
Ikan bandeng segar	98	32	32
Ikan bandeng tidak segar	102	34	34
Ikan samge sangat segar	103	35	35
Ikan samge segar	75	25	25
Ikan samge tidak segar	75	24	24
Ikan kuniran sangat segar	186	62	62
Ikan kuniran segar	152	50	50
Ikan kuniran tidak segar	138	46	46
Ikan senangin sangat segar	48	16	16
Ikan senangin segar	48	16	16
Ikan senangin tidak segar	48	16	16
Ikan kembung sangat segar	202	67	67
Ikan kembung segar	130	43	43
Ikan kembung tidak segar	133	42	42
Ikan gulamah sangat segar	48	16	16
Ikan gulamah segar	48	16	16
Ikan gulamah tidak segar	48	16	16
Total	2640	876	876

Setelah pembagian *dataset* pada Tabel 1 akan dilakukan proses pembuatan *model* dalam klasifikasi kesegaran ikan.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. *Hyperparameter*

Nilai *hyperparameter* ditentukan pada *model MobileNetV3-Large, MobilenetV3-Small, MobilenetV2* dan *MobilenetV1* yang dilakukan pada percobaan adalah *optimizer, batch size* dan *learning rate*. *Hyperparameter* yang digunakan berdasarkan eksperimen peneliti dan berdasarkan *hyperparameter* penelitian sebelum (**Prasetyo et al., 2022**). Pada Tabel 2 menjelaskan *hyperparameter* yang digunakan pada penelitian.

**Tabel 2. *Hyperparameter* yang digunakan**

<i>Hyperparameter</i>		
Jenis <i>Hyperparameter</i>	Hasil Eksperimen Peneliti	Hasil Peneliti Sebelum
Optimizer	ADAM	RMSprop
Batch size	Training 10, Val 10	Training 24, Val 22
Learning rate	0.00001	1e-5
Loss function	Categorical cross-entropy	Categorical cross-entropy
Weight	Imagenet	Imagenet
Input image size	224 x 224	224 x 224
Fully Connected layer	512 neuron	1024 neuron

### 3.2. Pemrosesan Awal *Dataset*

Citra pada *dataset* yang digunakan dilakukan pemrosesan awal dengan melakukan *resize* menjadi 224 x 224 piksel dengan 3 buah *channel*/warna *RGB* (*Red*, *Green*, dan *Blue*). Kemudian untuk menambahkan variasi pada *dataset* dilakukan augmentasi citra dengan *horizontal flip* dan *vertical flip*. Gambar 7 menunjukkan *sample* citra yang diambil dari *dataset* yang digunakan.

**Gambar 7. *Sample* Citra *Dataset***

### 3.3. Pelatihan *Model*

Pada penelitian dilakukan pelatihan pada 18 variasi *model*. Berdasarkan *hyperparameter* eksperimen peneliti pada Tabel 3, dan *hyperparameter* peneliti sebelum (**Prasetyo et al., 2022**) pada Tabel 4. Pada proses pelatihan dihasilkan *model* terbaik pada *learning rate* 0.0001, *epoch* 100, dengan ukuran *batch size train* 10 val 10, dan *optimizer ADAM* dengan *model* Arsitektur *MobileNetV3-Small* mendapatkan hasil *accuracy* sebesar 69%. Pada Gambar 8 (a) menunjukkan hasil *loss* dan (b) hasil *accuracy*.

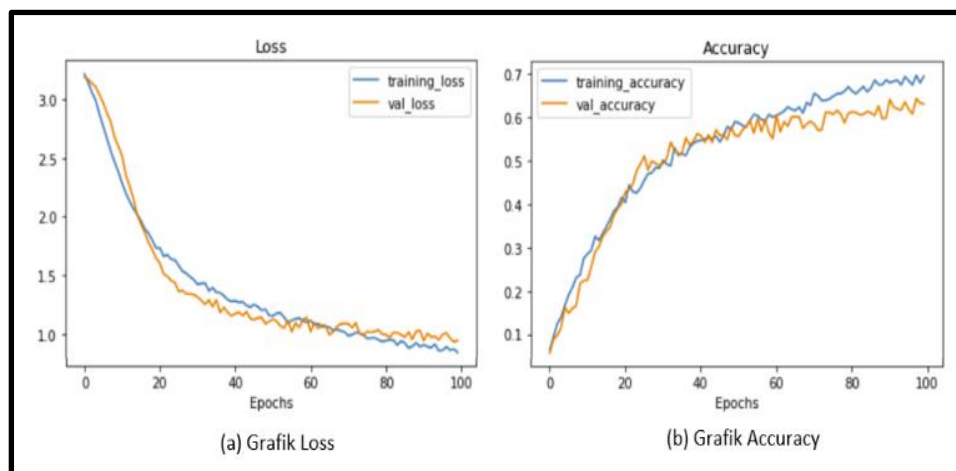
**Tabel 3. Detil Nilai *Hyperparameter* Eksperimen Peneliti**

No	Model	Hyperparameter eksperimen oleh peneliti			
		Optimizer	Lr	Batch size	Epoch
1	MobileNetV3-Large	ADAM	0.00001	Train 10, val 10	50, 100 dan 200
2	MobileNetV3-Small	ADAM	0.00001	Train 10, Val 10	50, 100 dan 200



**Tabel 4. Detil Nilai *Hyperparameter* Pada Peneliti Sebelum**

No	Model	Hyperparameter pada peneliti sebelum			
		Optimizer	Lr	Batch size	Epoch
1	MobileNetV3-Large	RMSprop	1e-5	Train 24, val 22	50, 100 dan 200
2	MobileNetV3-Small	RMSprop	1e-5	Train 24, Val 22	50, 100 dan 200
3	MobileNetV2	RMSprop	1e-5	Train 24, val 22	50, 100 dan 200
4	MobileNetV1	RMSprop	1e-5	Train 24, Val 22	50, 100 dan 200



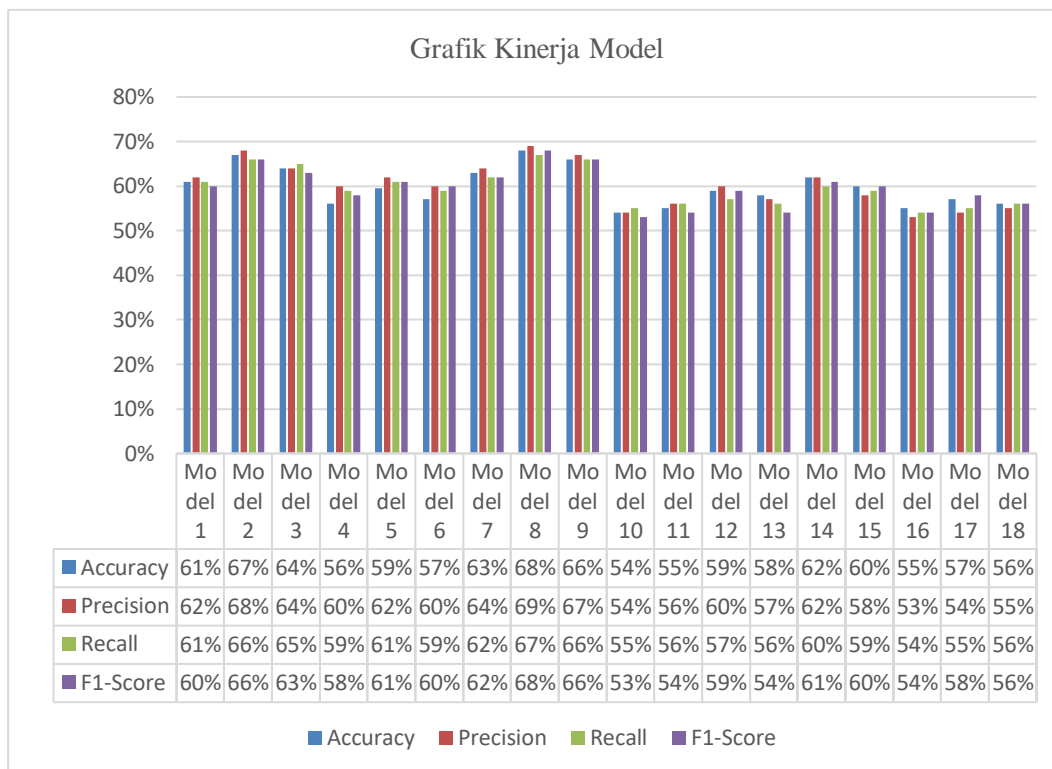
**Gambar 8 (a) Grafik *Loss* (b) Grafik *Accuracy***

### 3.4. Skenario Pengujian

Pengujian sistem dalam melakukan klasifikasi kesegaran ikan yang terdapat pada citra dengan menggunakan metode *convolutional neural network* arsitektur *MobileNetV3-Large* dan *MobileNetV3-Small* serta beberapa metode sebagai *reverse engineering* dari penelitian sebelum (**Prasetyo et al., 2022**) yaitu *MobileNetV1* dan *MobileNetV2*. Pengujian dilakukan dengan membandingkan beberapa *model* pelatihan, yang selanjutnya dilakukan pengujian akurasi menggunakan data uji. Pengujian model yang didapat tersebut memiliki *hyperparameter* yang bervariasi digunakan pada pelatihan *model* terhadap 876 citra uji. Dengan konfigurasi *model* pelatihan awal ditunjukkan pada bagian 4.3 pada Tabel 3 untuk *hyperparameter* eksperimen peneliti dan pada Tabel 4 untuk *hyperparameter* pada penelitian sebelum (**Prasetyo et al., 2022**).

### 3.5. Hasil Pengujian Sistem

Skema pelatihan dengan mengukur kinerja dari sistem yang dibuat menggunakan *model evaluation* untuk menghitung tingkat *accuracy*, *precision* dan *recall*. Pada Gambar 9 menunjukkan grafik perbandingan kinerja antara 18 variasi *model* yang telah dilatih dan dilakukan pengujian dengan data uji sebanyak 876 citra kesegaran ikan. Didapatkan akurasi tertinggi pada model 8 yaitu *model* arsitektur *MobileNetV3-Small* (*epoch* = 100, dan *learning rate* = 0.0001, *optimizer* = *ADAM*, *batch size* = *train* 10, *val* 10) dengan 68% *accuracy*, 69% *precision*, 67% *recall* dan 68% *F1-score*.



**Gambar 9. Grafik Perbandingan Kinerja Model**

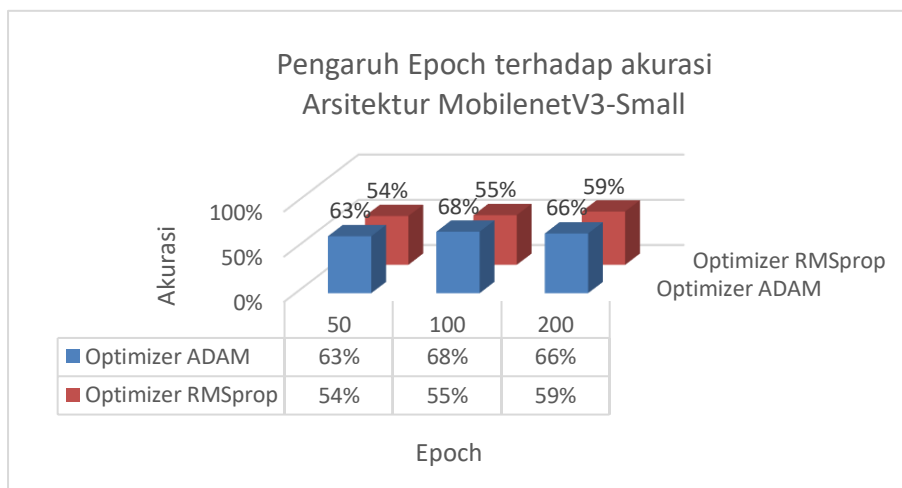
Pada Tabel 5, menjelaskan keterangan dari grafik yang terdapat pada Gambar 9.

**Tabel 5. Keterangan Grafik**

Model	Model	Variasi	
		Epoch	Optimizer
1	MobileNetV3-Large	50	ADAM
2		100	
3		200	
4		50	RMSProp
5		100	
6		200	
7	MobileNetV3-Small	50	ADAM
8		100	
9		200	
10		50	RMSProp
11		100	
12		200	
13	MobileNetV2	50	RMSProp
14		100	
15		200	
16	MobileNetV1	50	RMSProp
17		100	
18		200	

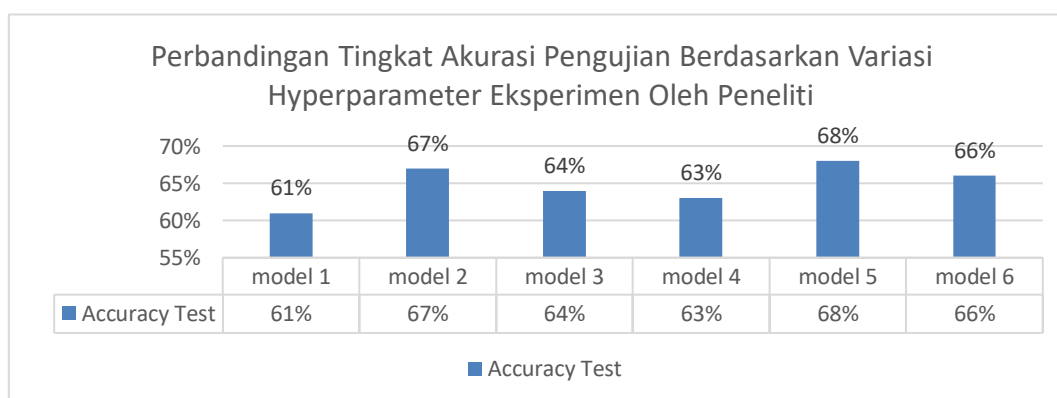
Peningkatan Performa MobilenetV3 dengan *Squeeze-and-Excitation*  
(Studi Kasus Klasifikasi Kesegaran Ikan Berdasarkan Mata Ikan)

*Epoch* yang digunakan pada saat melakukan pelatihan *model* akan berpengaruh pada akurasi yang dihasilkan, Gambar 10 menunjukkan pengaruh jumlah *epoch* yang digunakan terhadap akurasi yang dihasilkan. Hasil tersebut menunjukkan bahwa hasil tertinggi dari evaluasi *model* menunjukkan *learning curve* yang baik pada saat *training model*. Sehingga pada *epoch* 200 berdasarkan hasil yang didapatkan terhadap eksperimen yang dilakukan bahwa hasil evaluasi lebih rendah dikarenakan pada hasil *training model* menunjukkan *learning curve overfitting* sehingga berdampak pada akurasi pengujian yang lebih rendah.

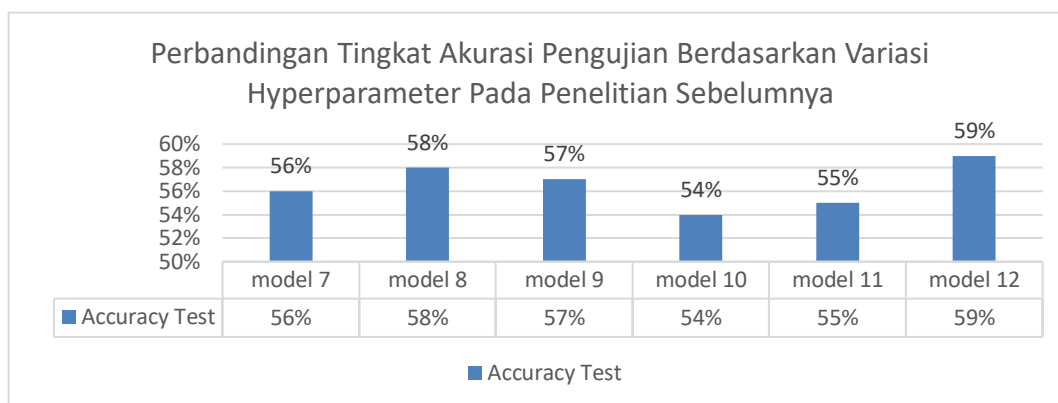


**Gambar 10. Grafik Pengaruh Nilai *Epoch* dan *Learning Rate* terhadap Akurasi**

Dilanjutkan dengan skema pengujian model berdasarkan *hyperparameter* eksperimen peneliti dan *hyperparameter* peneliti sebelum yang akan dilakukan perbandingan pada model arsitektur *MobileNetV3-Large* dan *MobileNetV3-Small*. kemudian akan diuji pada 876 citra uji kesegaran ikan. Pada hasil pengujian didapatkan akurasi pengujian tertinggi pada *model 5* dengan arsitektur *MobileNetV3-Small* dengan *epoch* 100 *lr* 0.00001 dengan nilai akurasi 68%. Perbandingan tingkat akurasi berdasarkan variasi *hyperparameter* eksperimen peneliti dapat dilihat pada Gambar 11 dan pada Gambar 12 pada *hyperparameter* pada peneliti sebelum (Prasetyo et al., 2022). Tabel 6 menjelaskan detail hasil pengujian pada model arsitektur *MobileNetV3-Large* dan *MobileNetV3-Small* dengan variasi *hyperparameter*.



**Gambar 11. Perbandingan Hasil Akurasi Berdasarkan Variasi Nilai *Hyperparameter* Eksperimen Peneliti**



**Gambar 12.** Perbandingan Hasil Akurasi Pengujian Berdasarkan Variasi Nilai *Hyperparameter* Penelitian Sebelum

**Tabel 6.** Detail Hasil pengujian

Model	Nama Model	Acc	Prs	Rcl	F1 Scr	Params (million)
1	MbV3-Large (50 epoch, optimizer ADAM)	61%	62%	61%	60%	3.65
2	MbV3-Large (100 epoch, optimizer ADAM)	67%	68%	66%	66%	3.65
3	MbV3-Large (200 epoch, optimizer ADAM)	64%	64%	65%	63%	3.65
4	MbV3-Small (50 epoch, optimizer ADAM)	63%	64%	62%	62%	1.37
5	MbV3-Small (100 epoch, optimizer ADAM)	68%	69%	67%	68%	1.37
6	MbV3-Small (200 epoch, optimizer ADAM)	66%	67%	66%	66%	1.37
7	MbV3-Large (50 epoch, optimizer RMSprop)	56%	60%	59%	58%	3.65
8	MbV3-Large (100 epoch, optimizer RMSprop)	59%	62%	61%	61%	3.65
9	MbV3-Large (200 epoch, optimizer RMSprop)	57%	60%	59%	60%	3.65
10	MbV3-Small (50 epoch, optimizer RMSprop)	54%	54%	55%	53%	1.37
11	MbV3-Small (100 epoch, optimizer RMSprop)	55%	56%	56%	54%	1.37
12	MbV3-Small (200 epoch, optimizer RMSprop)	59%	60%	57%	59%	1.37

Pada Tabel 6 menjelaskan dari hasil akhir pengujian performa *model* dengan variasi *model MobileNetV2* dan *MobileNetV1* terkait *reverse engineering* dari penelitian sebelum

**Tabel 6. Hasil Akhir Pengujian Performa Model**

Model	Params (million)	Train data accuracy / %	Val. data accuracy / %	Test			
				Accuracy / %	Precision / %	Recall / %	F1-Score / %
MobileNetV3-Large	3.65	76	67	67	68	66	66
MobileNetV3-Small	1.37	69	65	68	69	67	68
MBBE	3.16	57	63	60	58	58	58
MobileNetV2	2.25	65	45	62	62	60	61
MobileNetV1	3.22	60	40	57	54	55	58

Berdasarkan Pemaparan dari Tabel 4.36 bahwa hasil performa *model* yang dimana hasil dari penelitian sebelum yaitu (Prasetyo et al., 2022) di tuliskan kembali hasil dari performa *model* yaitu *MBBE (MobileNet Bottleneck with Expansion)* untuk dapat melihat hasil performa *model* secara keseluruhan dikarenakan penelitian ini melanjutkan penelitian dari (Prasetyo et al., 2022). Oleh sebab itu, arsitektur *MobileNetV3* yaitu *MobileNetV3-Small* menjadi lebih unggul dari model dengan menggunakan *MobileNetV3-Large*, *MobileNetV2*, *MBBE* dan *MobileNetV1* dapat dikatakan bahwa metode *MobileNetV3* merupakan *state-of-the-art* untuk studi kasus klasifikasi kesegaran ikan berdasarkan mata ikan. Peneliti melakukan lanjutan dari penelitian yang dilakukan oleh (Prasetyo et al., 2022) dengan mengimplementasi arsitektur *MobileNetV3* untuk dapat menghasilkan akurasi dan jumlah parameter *model* yang rendah, kemudian melakukan *set* nilai *hyperparameter* baru. Untuk jumlah parameter yang dihasilkan yaitu 1,378,604 dari *model MobileNetV3*, menghasilkan akurasi *training* dan akurasi *validation* yaitu 69% dan 65% serta untuk metrik pengujian dihasilkan nilai *f1-score* 68%, *presicion* 69%, *recall* 67% serta akurasi pengujian sebesar 68%. dan nilai *loss* pada pengujian 876 data uji sebesar 0,91.

#### 4. KESIMPULAN

Berdasarkan hasil yang didapatkan bahwa *model* yang memiliki hasil yang terbaik merupakan *model* dengan arsitektur *MobilenetV3-Small* dengan *hyperparameter* *epoch* 100, *learning rate* 0.00001, *batch size train* 10 dan *validation* 10, *optimizer* *ADAM*, dengan jumlah parameter 1,378,604, menghasilkan akurasi dalam pengujian sebesar 68%, *precision* 69%, *recall* 67%, dan *f1-score* 68%, dan nilai *loss* pada pengujian 876 data uji sebesar 0,91. Berdasarkan hasil yang didapatkan bahwa *MobileNetV3-Small* lebih unggul dari *MobileNetV3-Large* dibuktikan dengan hasil akurasi pengujian sebesar 68%. Hal tersebut dapat dikatakan belum berhasil dikarenakan sering terjadi kesalahan dalam melakukan klasifikasi kesegaran ikan berdasarkan mata ikan. Hal tersebut disebabkan karena *dataset* pada setiap kelas tidak sama jumlahnya. Kemudian *splitting* data yang digunakan dengan persentase pembagian data 60% data *training* 20% data validasi dan 20% data *testing*. Dapat dikatakan jumlah data *training* sangat kurang sehingga akan mempengaruhi kinerja *model* dalam melakukan tugas klasifikasi kesegaran ikan berdasarkan mata ikan. *Dataset* yang digunakan pada penelitian ini menggunakan *Freshness of Fish Eyes Dataset (FFE) dataset*. Namun, jumlah data pada setiap kelas dapat dikatakan kurang sehingga penggunaan *dataset* pelatihan *model* kurang maksimal.

Maka dari itu penulis menyarankan untuk menggunakan *dataset* lain dengan jumlah citra yang lebih banyak di setiap kelasnya dan dapat dilakukan augmentasi data seperti *flip* dan *zoom* saat membuat *model* pelatihan untuk menambah variasi *dataset*. *Splitting* data disarankan dengan persentase pembagian data pada data training, validasi dan *testing* yaitu 80%, 10%, 10%.

## DAFTAR PUSTAKA

- Agarwal, M., Gupta, S. K., & Biswas, K. K. (2020). Development of Efficient CNN model for Tomato crop disease identification. *Sustainable Computing: Informatics and Systems*, 28, 100407.
- Chu, X., Zhang, B., & Xu, R. (2020). Moga: Searching beyond mobilenetv3. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4042–4046.
- Diana, F. M. (2013). Omega 3 dan kecerdasan anak. *Jurnal Kesehatan Masyarakat Andalas*, 7(2), 82–88.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *ArXiv Preprint ArXiv:1704.04861*.
- Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., & Vasudevan, V. (2019). Searching for mobilenetv3. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1314–1324.
- Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7132–7141.
- Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*, 6(4), 312–315.
- Prasetyo, E., Purbaningtyas, R., Adityo, R. D., Suciati, N., & Fatichah, C. (2022). *Combining MobileNetV1 and Depthwise Separable convolution bottleneck with Expansion for classifying the freshness of fish eyes*. Information Processing in Agriculture. <https://doi.org/10.1016/j.inpa.2022.01.002>
- Qian, S., Ning, C., & Hu, Y. (2021). MobileNetV3 for Image Classification. *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering, ICBAIE 2021, Icbaie*, 490–497. <https://doi.org/10.1109/ICBAIE52039.2021.9389905>
- Radiuk, P. M. (2017). Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. *Information Technology and Management Science*, 20(1), 20–24.

Peningkatan Performa MobilenetV3 dengan *Squeeze-and-Excitation*  
(Studi Kasus Klasifikasi Kesegaran Ikan Berdasarkan Mata Ikan)

- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. (2018). *Mobilenetv2: Inverted residuals and linear bottlenecks. in proc. ieee conference on computer vision and pattern recognition.*
- Ying, X. (2019). An overview of overfitting and its solutions. *Journal of Physics: Conference Series, 1168*, 022022.
- Zaki, S. Z. M., Zulkifley, M. A., Stofa, M. M., Kamari, N. A. M., & Mohamed, N. A. (2020). Classification of tomato leaf diseases using MobileNet v2. *IAES International Journal of Artificial Intelligence, 9(2)*, 290.