

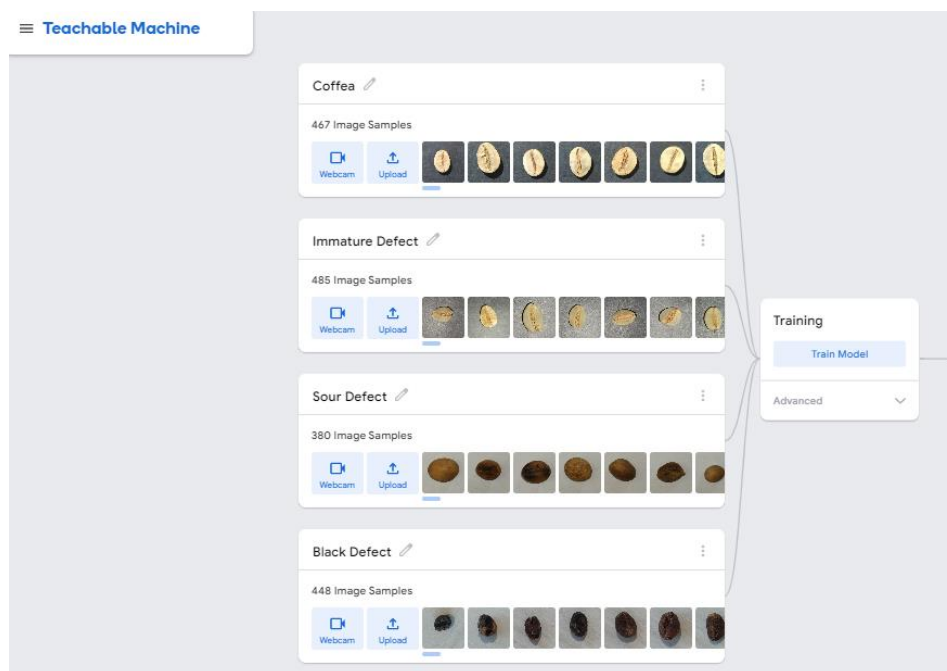
BAB IV

HASIL DAN PEMBAHASAN

4.1. Tahap Persiapan *Dataset*

Pada tahap ini data yang berupa gambar akan dilakukan proses *training* agar dapat menghasilkan model yang sesuai. Proses ini menggunakan sebuah *web-tools teachable machine* yakni sebuah aplikasi berbasis web yang digunakan untuk membuat model klasifikasi.

Gambar yang akan diklasifikasi berupa gambar biji kopi yang terbagi menjadi empat *class* yakni *coffea*, *Immature Defect*, *Sour Defect*, dan *Black Defect*. Pengelompokkan biji kopi ini menggunakan jenis kopi robusta yang dikategorikan pada bentuk *defect* atau cacat. Sehingga dapat membedakan objek gambar yang akan dihasilkan. Berikut ini merupakan tampilan *web tools teachable machine* pada Gambar 4.1.



Gambar 4.1 Tampilan *Web Teachable machine*

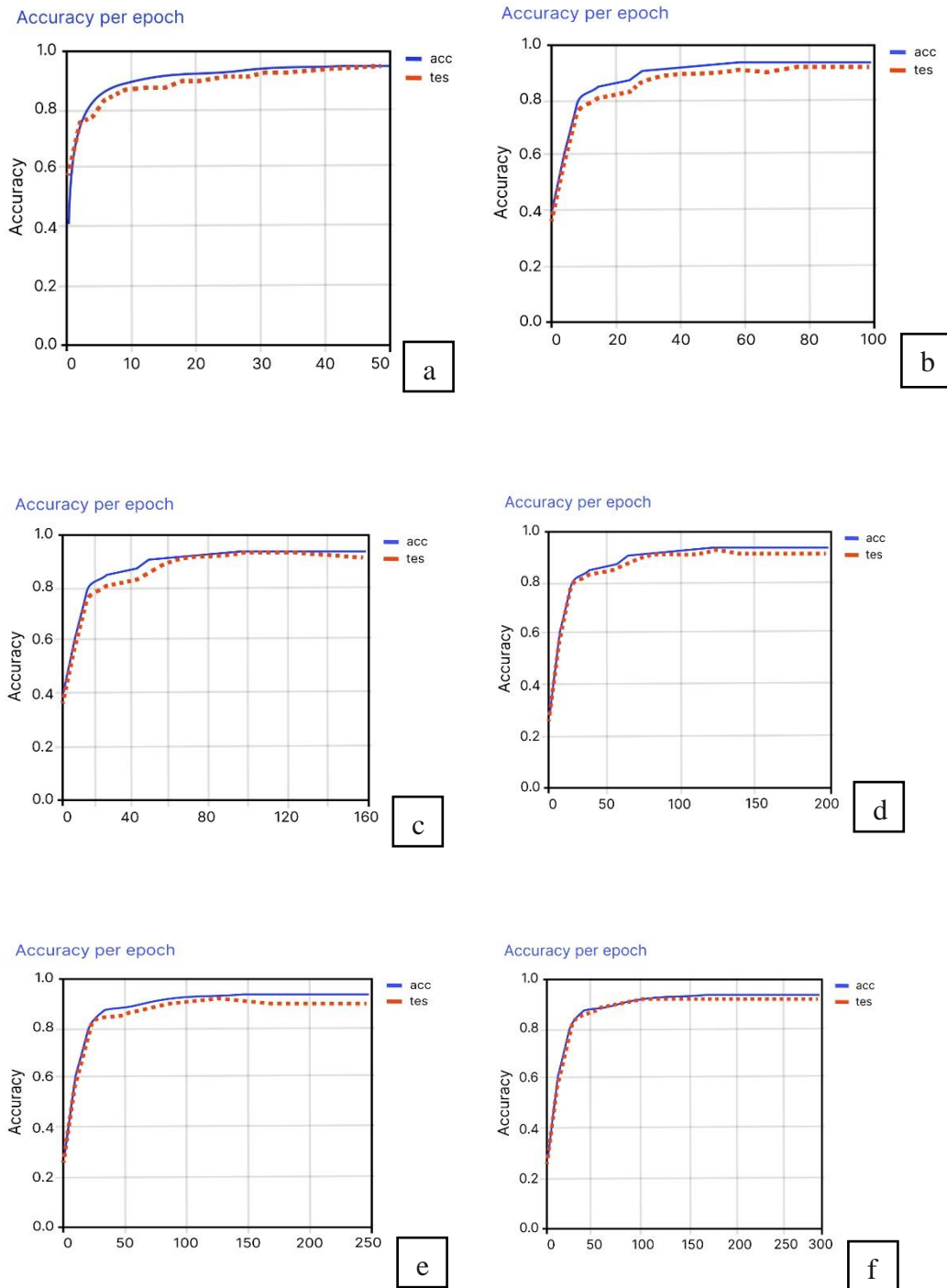
Berdasarkan Gambar 4.1 merupakan tampilan dari hasil *input dataset* biji kopi yang akan di-*training* menggunakan *web tools teachable machine* pada proses ini *dataset di-training* dengan mempertimbangkan jenis *dataset*, *batch size* dan juga *learning rate*.

Dataset merupakan *hyperparameter* yang menentukan berapa kali algoritma *deep learning* bekerja melewati seluruh *dataset*, yang artinya satu *dataset* tercapai ketika semua batch telah dilewatkan melalui *neural network* satu kali. *Batch size* merupakan jumlah sampel *dataset* yang akan dilalui dalam satu waktu.

Pada *batch size* ini dapat menentukan jumlah sampel yang akan dikerjakan. *Learnig rate* merupakan salah satu parameter *training* untuk menghitung nilai koreksi bobot pada waktu proses *training*. Pada hal ini *learning rate* menjadi parameter seberapa cepat proses *training* yang akan dilakukan.

4.2. Pengaruh Dataset Terhadap Model

Pada tahap pengujian yang dilakukan dengan mengetahui baiknya sebuah model *neural network* yang dibuat dengan membandingkan *dataset* terhadap model sehingga dapat menjadi bahan acuan dalam menentukan tingkat akurasi model yang diinginkan. Pada penelitian ini digunakan enam kategori *epoch* yang berbeda yaitu 50, 100, 150, 200, 250, 300 dengan batch size 64 dengan *dataset* yang sama. Variasi *epoch* dilakukan untuk menemukan model *learning* terbaik dan ideal pada kasus penelitian kualitas biji kopi ini. *Epoch* yang rendah berpotensi untuk mengalami kegagalan dalam pembelajaran model, sementara pada kasus *epoch* yang lebih tinggi berpotensi terjadinya *overfitting*. *Epoch* standar dalam *machine learning* adalah 100 iterasi, tetapi pada pembelajaran yang lebih berat penambahan hingga 500 *epoch* dapat dipertimbangkan. Hasil dari variasi *epoch* akan ditunjukkan dalam bentuk hasil *training* berupa grafik *train accuracy* dan *train testing* yang biasa disebut dengan *validation*. Model yang baik merupakan model yang mencapai tingkat konvergensi yang ditandai dengan tidak adanya perubahan nilai pada parameter akurasi mendekati 1. Hasil dari visualisasi proses *training* dengan variasi *epoch* ditunjukkan pada Gambar 4.2.



Gambar 4.2 Hasil Training *Dataset* per *Epoch*, (a) epoch 50, (b) epoch 100, (c) epoch 150, (d) epoch 200, (e) epoch 250, (f) epoch 300.

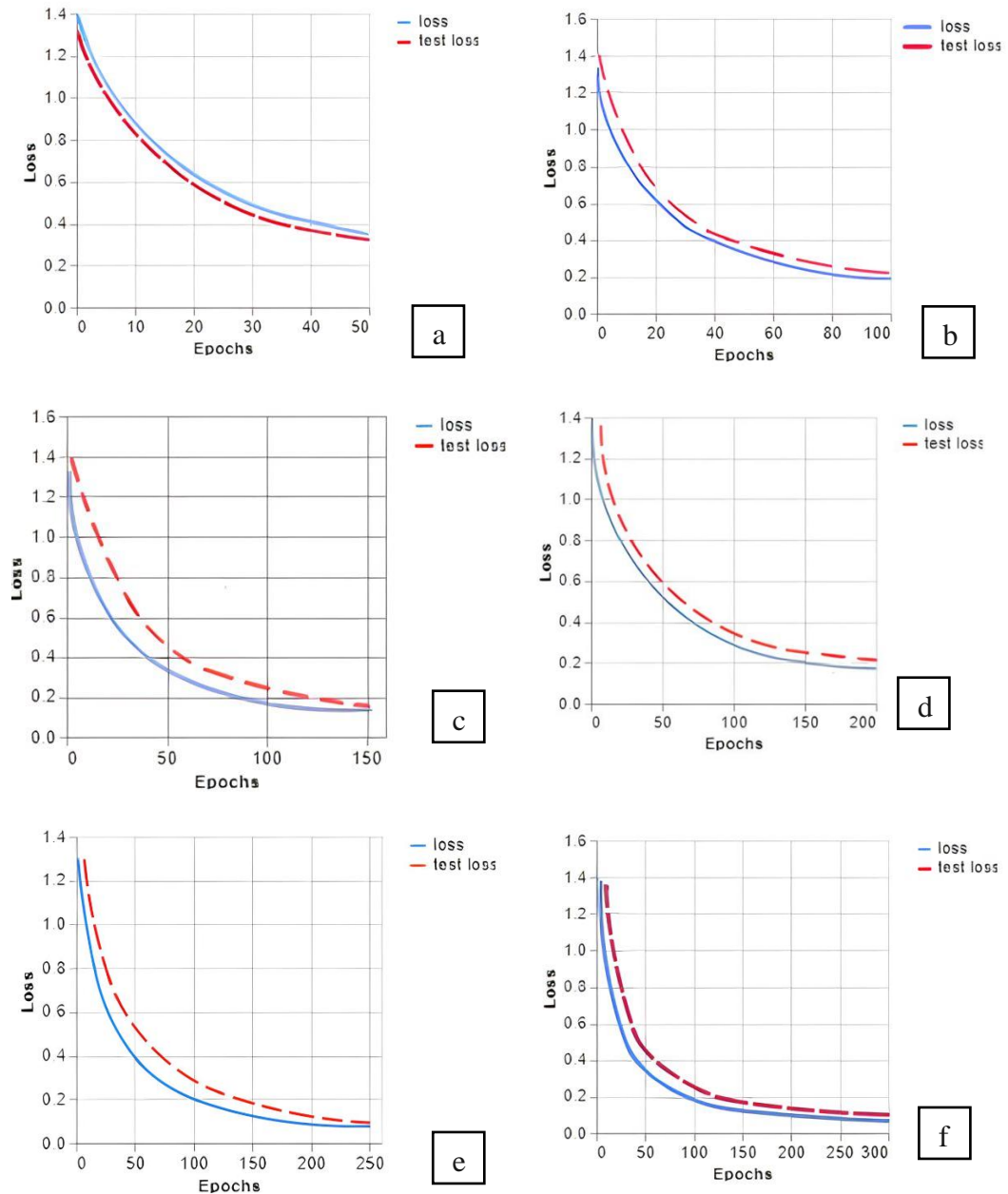
Gambar 4.2 merupakan hasil dari *training dataset* yang dilakukan. Dapat dilihat peningkatan akurasi biji kopi per-*dataset*, pada penelitian ini terdapat 6 parameter *dataset* untuk menentukan jenis model apa yang akan digunakan pada pengujian aplikasi. Gambar 4.2 (a) *Epoch* 50 memiliki akurasi sebesar 93% dengan test akurasi sebesar 89% dan *test sample* akurasi sebesar 89%, dalam hal ini *test* akurasi bertujuan untuk mengidentifikasi model yang dilatih terhadap gambar *independent* yang tidak digunakan dalam *training*. Akurasi bertujuan untuk mengidentifikasi gambar yang digunakan baik untuk *training* dan juga testing. Sehingga selisih dari akurasi dan test akurasi sebesar 4%. Gambar 4.2 (b) *Epoch* 100 memiliki akurasi sebesar 95% dengan test akurasi sebesar 91%. Sehingga selisih dari akurasi dan test akurasi Berdasarkan Tabel *dataset* 100 sebesar 4%. Gambar 4.2 (c) *Epoch* 150 memiliki akurasi sebesar 95% dengan *test* akurasi sebesar 95%. Sehingga selisih dari akurasi dan test akurasi pada Gambar *Epoch* 150 sebesar 0%. Gambar 4.2 (d) *Epoch* 200 memiliki akurasi sebesar 96% dengan test akurasi sebesar 93%. Selisih dari akurasi dan test akurasi pada Gambar *Epoch* 200 sebesar 3%. Gambar 4.2 (e) *Epoch* 250 memiliki akurasi 96% dengan test akurasi sebesar 95%. Sehingga selisih dari akurasi dan test akurasi pada Gambar *Epoch* 250 sebesar 1%. Gambar 4.2 (f) *Epoch* 300 memiliki akurasi 97% dengan test akurasi sebesar 95%. Sehingga selisih dari akurasi dan test akurasi Berdasarkan Tabel 300 sebesar 2%.

4.3. Pengaruh *Loss* Terhadap Model

Pada pengujian yang dilakukan dengan mengetahui nilai *loss*, yakni nilai dari perhitungan *loss function* dari *training dataset* dan prediksi dari model. Pengujian ini akan terfokus pada konvergensi nilai *loss* model yang telah dilakukan proses *training* dengan *setting epoch* sebesar 50, 100, 150, 200, 250, 300. Konvergensi *loss* dapat diindikasikan dengan tidak adanya perubahan nilai *loss* secara fluktuatif dan nilai mendekati nilai 0.

Variasi rendah pada nilai *epoch* akan memberikan waktu *training* yang lebih cepat, tetapi kemungkinan gagal *learning* yang cukup tinggi. Pada kasus variasi yang lebih tinggi dapat memberikan waktu untuk melakukan proses *training*, tetapi

dengan membutuhkan waktu yang lebih lama. Pada Gambar 4.3 merupakan visualisasi hasil *training* dengan hasil *loss training*.



Gambar 4.3 Hasil *Loss Training*, (a) *Epoch 50*, (b) *Epoch 100*, (c) *Epoch 150*, (d) *Epoch 200*, (e) *Epoch 250*, (f) *Epoch 300*

Gambar 4.3 di atas merupakan hasil *loss training* pada tiap *dataset* yang dilakukan. Pada penelitian ini mengetahui nilai *loss* dan *test loss* dengan tujuan mengetahui antara keluaran yang dibuat oleh algoritma yang sedang di training

dengan keluaran yang diharapkan. Pada *function loss* ini akan mengevaluasi seberapa baik algoritma dalam memodelkan data dan menghasilkan *output* yang tepat, jika prediksi model mengandung banyak *error*/kesalahan maka *loss function* akan menghasilkan angka yang lebih tinggi. Begitu sebaliknya jika model sudah cukup bagus, maka *loss function* akan memberikan nilai yang lebih rendah. Gambar 4.3 (a) *dataset 50 loss function* mencapai 35% dengan *test loss* mencapai 33%. Selisih antara *loss function* dengan *test loss* pada *dataset 50* yakni 2%. Gambar 4.3 (b) *dataset 100 loss function* mencapai 18% dengan *test loss* sebesar 21%. Sedangkan selisih antara *loss function* dengan *test loss* pada *dataset 100* yakni 3%. Pada Gambar 4.3 (c) *dataset 150 loss function* mencapai 13% dengan *test loss* mencapai 17%. Selisih antara *loss function* dengan *test loss* pada *dataset 150* yakni 4%. Gambar 4.3 (d) *dataset 200 loss function* mencapai 9% dengan *test loss* sebesar 15%. Sedangkan selisih antara *loss function* dengan *test loss* pada *dataset 200* yakni 6%. Gambar 4.3 (e) *dataset 250 loss function* mencapai 7% dengan *test loss* sebesar 9%. Selisih antara *loss function* dengan *test loss* pada *dataset 250* sebesar 2%. Gambar 4.3 (f) *dataset 300 loss function* mencapai 6% dengan *test loss* mencapai 9%. Sedangkan selisih antara *loss function* dengan *test loss* pada *dataset 300* yakni 3%.

4.4. Analisis Model CNN Learning

Pada pembahasan ini analisis dari masing-masing penelitian ini dilakukan menggunakan *teachable machine*. Parameter yang digunakan sebagai analisis model ini ialah *F1-Score*, *accuracy*, *precision*, dan *recall* dari tiap-tiap *dataset*, selain itu terdapat juga parameter lainnya seperti *class* yang dibagi menjadi empat yakni *coffea*, *Immature Defect*, *Sour Defect*, dan *Black Defect*.

4.4.1. Analisis Object Classification Class 1 Coffea

Pada penelitian ini menggunakan metode CNN dengan klasifikasi gambar biji kopi. Hasil penelitian akan ditunjukkan menggunakan *confussion matrix* per objek dari klasifikasi gambarnya. Penelitian ini mengacu pada parameter dari *object classification class 1 (Coffea)*, oleh sebab itu parameter hasil dari *confussion matrix*

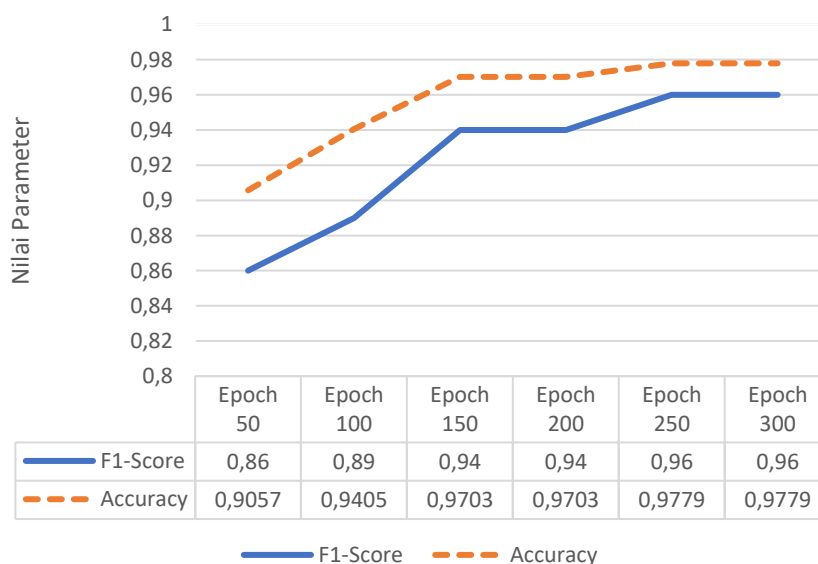
didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.1 di bawah ini.

Tabel 4.1 *Object Classification Class 1 Coffea*

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	92,57	0,85	0,87	0,86
100	94,05	0,92	0,87	0,89
150	97,03	0,92	0,97	0,94
200	97,03	0,92	0,97	0,94
250	97,79	0,94	0,97	0,96
300	97,79	0,94	0,97	0,96
\bar{X}	96,04	0,915	0,936	0,952

Berdasarkan Tabel 4.1 merupakan hasil pengukuran dari parameter *object classification coffea* dengan menggunakan metode CNN. Berdasarkan tabel 4.1 tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian *sample coffea* sebesar 96,04%

Evaluasi *F1-score* nilai rata-rata nya sebesar 0,952 menunjukkan standar performa yang cukup tinggi. Nilai rata-rata *precision* dan *recall* yang merupakan variabel yang berpengaruh terhadap *F1-score* itu sendiri sebesar 0,915 dan 0,936. Berdasarkan data Berdasarkan Tabel 4.1 dapat ditentukan grafik dari perbandingan sebagai.



Gambar 4.4 Performa Model pada *Object Classification Class 1 Coffea*

Gambar 4.4 merupakan grafik performa dengan diperlihatkan *f1-score* dan juga *accuracy*. Berdasarkan Gambar 4.4 ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*. Pada *dataset* ke 250 dan ke 300 memiliki *accuracy* dan *f1-score* tertinggi.

Berdasarkan keseluruhan model yang didapatkan memiliki rata-rata *accuracy* sebesar 0,9404, nilai rata-rata *precision* sebesar 0,915, nilai rata-rata *recall* sebesar 0,936, dan nilai rata-rata *f1-score* sebesar 0,952, dari nilai rata-rata yang didapatkan pada penelitian ini, disimpulkan bahwa model yang diperoleh menggunakan *web-tools teachable machine* memiliki hasil *learning* dan *prediction* yang cukup baik, serta memiliki kehandalan dalam mendeteksi objek menggunakan kamera android, namun dalam proses pengambilan *dataset*-nya terdapat *noise* cahaya yang masuk tetapi bisa digunakan dalam mengenali objek.

4.4.2. Analisis Object Classification Class 2 Immature Defect

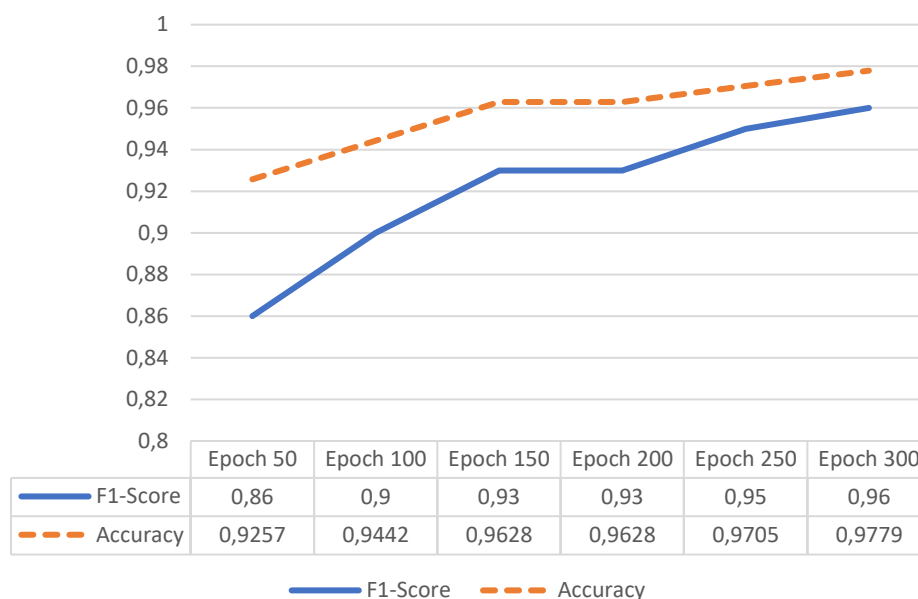
Pada analisis selanjutnya yang mengacu pada parameter dari *object classification class 2 immature defect*. Pada pembahasan ini menggunakan parameter hasil dari *confussion matrix* yang didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.2 di bawah ini.

Tabel 4.2 *Object Classification Class 2 Immature Defect*

<i>Epoch</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
50	92,57%	0,88	0,85	0,86
100	94,42%	0,88	0,91	0,90
150	96,28%	0,97	0,90	0,93
200	96,28%	0,97	0,90	0,93
250	97,05%	0,97	0,92	0,95
300	97,79%	0,97	0,97	0,96
\bar{X}	95,73%	0,94	0,91	0,92

Berdasarkan Tabel 4.2 merupakan hasil pengukuran dari parameter *object classification Immature Defect* dengan menggunakan metode CNN. Berdasarkan Tabel 4.2 tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada sampel *Immature Defect* sebesar 95,73%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,921, untuk nilai rata-rata *precision* dan *recall* yang merupakan nilai harmonik dari *f1-score* itu sendiri sebesar 0,94 dan

0,908. Berdasarkan data Berdasarkan Tabel 4.2 dapat ditentukan grafik dari perbandingan yang ditunjukkan pada Gambar 4.5 berikut.



Gambar 4.5 Performa Model pada *Object Classification Class 2*
Immature Defect

Pada Gambar 4.5 merupakan grafik performa dengan perbandingan *f1-score* dan juga *accuracy*. Berdasarkan Gambar 4.5 hasil dari performa model dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*. Titik tertinggi pada nilai *f1-score* dan *accuracy* terdapat pada *dataset* 250 dan 300. Dari keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,9573, nilai rata-rata *precision* sebesar 0,94, nilai rata-rata *recall* sebesar 0,908, dan nilai rata-rata *f1-score* sebesar 0,921.

Berdasarkan nilai rata-rata yang didapatkan pada percobaan ini dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik. Serta memiliki keandalan dalam mendeteksi objek menggunakan kamera android, namun dalam proses pengambilan *dataset* nya terdapat noise cahaya yang masuk tetapi bisa digunakan dalam mengenali objek.

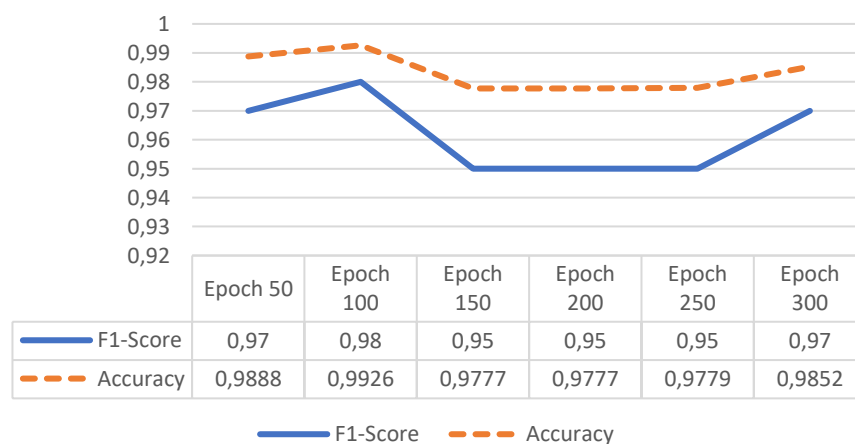
4.4.3. Analisis Object Classification Class 3 Sour Defect

Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 3 (Sour Defect)*, pada parameter ini hampir sama dengan analisis sebelumnya yakni membahas parameter hasil dari *confussion matrix* didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.3 dibawah ini.

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
50	98,88	0,98	0,97	0,97
100	99,26	0,98	0,98	0,98
150	97,77	0,93	0,96	0,95
200	97,77	0,93	0,96	0,95
250	97,79	0,93	0,96	0,95
300	98,52	0,97	0,97	0,97
\bar{X}	98,33	0,95	0,96	0,96

Tabel 4.3 *Object Classification Class 3 Sour Defect*

Berdasarkan Tabel 4.3 merupakan hasil dari pengukuran dari parameter *object classification* pada class 3 dengan metode CNN. Berdasarkan tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian sample *Sour Defect* sebesar 98,33%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,961, untuk nilai rata-rata *precision* dan recall ialah sebesar 0,953 dan 0,966. Berdasarkan data Berdasarkan Tabel 4.3 dapat ditentukan grafik dari perbandingan pada Gambar 4.6.



Gambar 4.6 Performa Model pada *Object Classification Class 3 Sour Defect*

Gambar 4.6 merupakan grafik performa dengan perbandingan *f1-score* dan juga *accuracy* pada *object classification class 3*. Berdasarkan gambar di atas ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan nilai *accuracy*. Titik tertinggi pada nilai *f1-score* dan *accuracy* terdapat pada *dataset 250 dan 300*.

Berdasarkan keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,9833, dan nilai rata-rata *f1-score* sebesar 0,961. Nilai yang didapatkan pada penelitian ini dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik. Keandalan dalam mendeteksi objek menggunakan kamera Android memiliki reliabilitas yang cukup, tetapi faktor pencahayaan dapat menjadi penurunan keandalan algoritma.

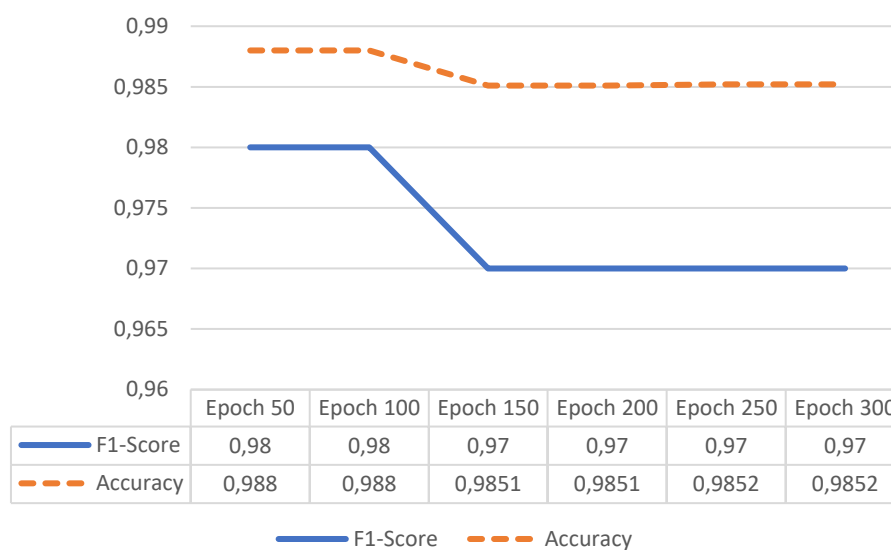
4.4.4. Analisis Object Classification Class 4 Black Defect

Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 4 (Black Defect)*, pada parameter ini hampir sama dengan analisa sebelumnya yakni membahas parameter hasil dari confusion matrix didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.4 dibawah ini.

Tabel 4.4 *Object Classification Class 4 (Black Defect)*

<i>Epoch</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
50	98,88%	0,97	0,99	0,98
100	98,88%	0,99	0,99	0,98
150	98,51%	0,97	0,97	0,97
200	98,51%	0,97	0,97	0,97
250	98,52%	0,97	0,97	0,97
300	98,52%	0,97	0,97	0,97
\bar{X}	98,63%	0,973	0,976	0,973

Berdasarkan Tabel 4.4 merupakan hasil dari pengukuran dari parameter object classification pada class 4 dengan metode CNN. Berdasarkan Tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian sample black coffea sebesar 98,63%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,973. untuk nilai rata-rata *precision* dan recall ialah sebesar 0,973 dan 0,976. Berdasarkan data Berdasarkan Tabel 4.4 dapat ditentukan grafik dari perbandingan pada Gambar 4.7.



Gambar 4.7 Performa Model pada *Object Classification Class 4 Black Defect*

Gambar 4.7 merupakan grafik performa dengan membandingkan *f1-score* dan juga *accuracy*. Berdasarkan gambar di atas ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*. Titik tertinggi pada nilai *f1-score* dan *accuracy* terdapat pada *dataset 205* dan *300*.

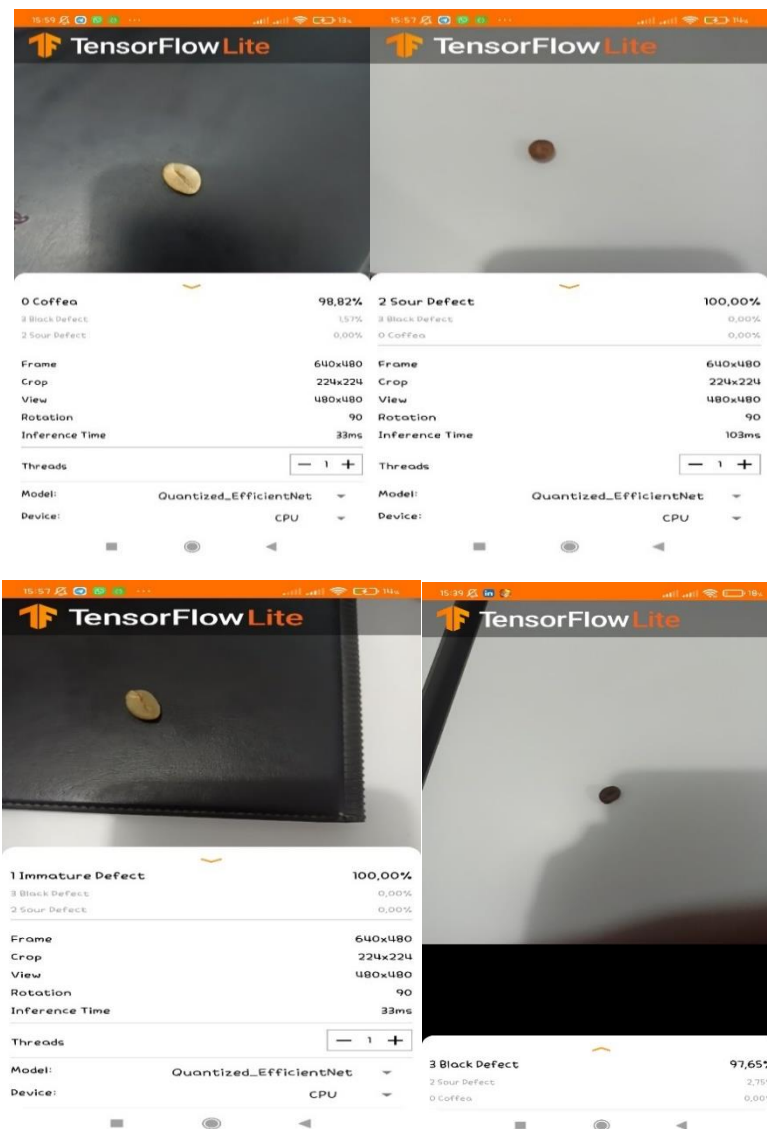
Berdasarkan keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,9863, nilai rata-rata *precision* sebesar 0,973, nilai rata-rata *recall* sebesar 0,976, dan nilai rata-rata *f1-score* sebesar 0,921. Berdasarkan nilai rata-rata yang didapatkan pada penelitian ini ialah dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik.

4.5. Hasil dan Pengujian Implementasi Pada Android

Pengujian ini bertujuan untuk melihat performa dan akurasi yang diperoleh setelah model diimplementasikan pada perangkat Android. Dalam hal ini terdapat empat parameter yang akan dibahas antara lain uji akurasi aplikasi terhadap model dan yang kedua uji coba performa aplikasi.

4.5.1. Uji Coba Akurasi Terhadap Model

Uji coba ini dilakukan untuk mengetahui tingkat akurasi dari aplikasi pengklasifikasian pada biji kopi. Pengujian ini dilakukan secara *real-time* dengan menggunakan citra pada biji kopi yang sebelumnya sudah *di-training*. *Dataset* model yang digunakan pada uji coba ini menggunakan model dengan *dataset* 300. Pada gambar di bawah ini adalah hasil dari uji coba akurasi model menggunakan aplikasi *Tensorflow Lite Image Classification Example* dengan *extension file* nya model.TFlite yang sebelumnya sudah dibuat.



Gambar 4.8 Hasil Tingkat Akurasi Pada Aplikasi

Gambar 4.8 di atas merupakan hasil uji coba aplikasi yang dibuat, pada aplikasi ini merupakan aplikasi yang digunakan untuk mengetahui nilai cacat pada biji kopi. Penelitian ini terdapat 4 *Class* yakni *coffea* untuk kualitas biji yang baik, *immature defect* untuk cacat tingkat satu, *sour defect* untuk cacat tingkat dua, dan yang terakhir *black defect* untuk cacat tingkat tiga.

Berdasarkan keempat *class* ini dapat ditentukan dari nilai presentase nya seperti gambar di atas, yang merupakan *confidence score* atau nilai keyakinan pada aplikasi dalam mengkategorikan gambar. Pada aplikasi ini *confidence score* total mencapai 100%. Di bawah ini merupakan Tabel 4.5 hasil dari uji coba terhadap dua puluh citra biji kopi dengan masing-masing kelas sebanyak lima biji *sample*.

Tabel 4.5 Tingkat Akurasi Pada Aplikasi

Jenis Nilai Cacat	Jumlah Sample	Tingkat Akurasi (%)	Rata-Rata Nilai Keyakinan (%)
Coffea	5	100	96
Immature Defect	5	100	97
Sour Defect	5	100	95
Black Defect	5	100	97

Berdasarkan Tabel 4.5 di atas dapat diketahui bahwa aplikasi dapat mengklasifikasi pada biji kopi dengan sangat baik dengan tingkat akurasi yang didapat mencapai sempurna dan dengan tingkat keyakinan di atas 90%, namun hal ini dipengaruhi oleh pencahayaan yang sesuai dan juga *background* yang sesuai dengan sample sebelumnya. Jika hal tersebut tidak sesuai, maka hasil dari pengambilan nilai keyakinannya tidak akan mencapai 90%, karena dipengaruhi faktor pencahayaan dan juga background pengambilan gambarnya.

4.5.2. Pengaruh Cahaya Terhadap Pengambilan Objek

Pada pengujian ini pengaturan cahaya berpengaruh terhadap hasil dari pengambilan objek gambar, sehingga dalam hal ini penting sekali untuk mengatur intensitas pencahayaan agar kemampuan identifikasi objek, semakin rendah tingkat pencahayaan maka semakin sulit bagi sistem untuk mengidentifikasi objek gambar pada biji kopi dan begitu juga sebaliknya. Berikut ini Tabel 4.6 yang merupakan hasil dari pengambilan objek gambar berdasarkan instensitas pencahayaan.

Tabel 4.6 Pengambilan Objek Berdasarkan Intensitas Cahaya dengan Jarak 20 cm

Percobaan	Durasi (detik)	Jarak (cm)	Intensitas (Flux)	Kemampuan Identifikasi (%)
1	15	20	200	99
2	20	20	200	98
3	25	20	200	99
4	30	20	200	98
5	35	20	200	99

Berdasarkan Tabel 4.6 di atas merupakan hasil pengukuran akurasi berdasarkan intensitas cahaya dengan jarak 20 cm. Berdasarkan data tersebut dapat disimpulkan semakin besar intensitas cahaya yang diperoleh semakin besar performa mendeteksi objek. Parameter lain yang dapat mempengaruhi aplikasi dalam mendeteksi objek yaitu *noise* pada objek yang disebabkan oleh besarnya intensitas cahaya, namun itu bisa diatasi dengan mengatur jarak antara pencahayaan dengan objek yang akan di ambil. Berdasarkan Tabel 4.7 merupakan hasil pengukuran dengan variasi jarak sejauh 30 cm.

Tabel 4.7 Pengambilan Objek Berdasarkan Intensitas Cahaya dengan Jarak 30 cm

Percobaan	Durasi (detik)	Jarak (cm)	Intensitas (Flux)	Kemampuan Identifikasi (%)
1	15	30	197	96
2	20	30	197	96
3	25	30	197	97
4	30	30	197	97
5	35	30	197	97

Berdasarkan Tabel 4.7 di atas merupakan hasil pengukuran akurasi berdasarkan intensitas cahaya dengan jarak 30 cm. Berdasarkan Tabel tersebut terdapat lima kali percobaan yang dilakukan. Berdasarkan data tersebut dapat disimpulkan semakin besar intensitas cahaya yang diperoleh semakin besar aplikasi mendeteksi objek. Parameter lain yang dapat mempengaruhi aplikasi dalam mendeteksi objek yaitu *noise* pada objek yang disebabkan oleh besarnya intensitas cahaya, namun hal tersebut dapat diatasi dengan mengatur jarak antara pencahayaan dengan objek yang akan diambil. Berdasarkan Tabel 4.8 merupakan hasil dari pengambilan data pada jarak 40 cm.

Tabel 4.8 Pengambilan Objek Berdasarkan Intensitas Cahaya dengan Jarak 40 cm

Percobaan	Durasi (detik)	Jarak (cm)	Intensitas (Flux)	Kemampuan Identifikasi (%)
1	15	40	134	97
2	20	40	133	97
3	25	40	134	97
4	30	40	134	97
5	35	40	134	96

Berdasarkan Tabel 4.8 di atas merupakan hasil pengukuran akurasi berdasarkan intensitas cahaya dengan jarak 40 cm. Berdasarkan Tabel tersebut terdapat lima kali percobaan yang dilakukan. Berdasarkan data ini dapat disimpulkan semakin besar intensitas cahaya yang diperoleh semakin besar aplikasi mendeteksi objek.

Parameter yang dapat mempengaruhi aplikasi dalam mendeteksi objek yaitu *noise* pada objek yang disebabkan oleh besarnya intensitas cahaya, namun itu bisa diatasi dengan mengatur jarak antara pencahayaan dengan objek yang akan diambil. Berdasarkan Tabel 4.9 merupakan hasil pengambilan data pada jarak 60 cm.

Tabel 4.9 Pengambilan Objek Berdasarkan Intensitas Cahaya dengan Jarak 60 cm

Percobaan	Durasi (detik)	Jarak (cm)	Intensitas (Flux)	Kemampuan Identifikasi (%)
1	15	60	112	99
2	20	60	113	98
3	25	60	112	99
4	30	60	112	98
5	35	60	112	99

Berdasarkan Tabel 4.9 di atas merupakan hasil pengukuran akurasi berdasarkan intensitas cahaya dengan jarak 60 cm. Berdasarkan Tabel tersebut terdapat lima kali percobaan yang dilakukan, berdasarkan data ini dapat disimpulkan semakin besar intensitas cahaya yang diperoleh semakin besar aplikasi mendeteksi objek.

Parameter lain yang dapat mempengaruhi aplikasi dalam mendeteksi objek yaitu *noise* pada objek yang disebabkan oleh besarnya intensitas cahaya, namun itu bisa diatasi dengan mengatur jarak antara pencahayaan dengan objek yang akan diambil.

4.5.3. Uji Coba Performa Aplikasi

Pada pengujian performa, aplikasi Tensorflow *lite*, memiliki fitur-fitur yang dapat digunakan untuk mengubah jenis model TFLite beserta optimisasi yang ada pada Tensorflow *lite*. Disini terdapat dua jenis model TFLite yaitu *Quantized* dan *Unquant*.

Model *quantized* merupakan model TFLite yang sudah melakukan proses *quantization*. *Quantization* merupakan proses optimisasi yang dimiliki oleh TFLite yang memiliki fungsi mengurangi ukuran dan mempermudah proses komputasi dan mempertahankan akurasi.

Perbedaannya dengan model *Unquant* yakni dari segi kinerja jenis datanya untuk *quantization* memiliki jenis data berjenis *Integer* 8 bit, sedangkan untuk *Unquant* berjenis data *float* 32, untuk penelitian ini digunakan jenis data *quantized*, hal ini dikarenakan kecepatan pemrosesan datanya lebih cepat. Namun memiliki kekurangan ketelitian pada model sehingga akurasi dari *quantized* lebih kecil dibandingkan dengan *unquant*. Perbandingan ukuran *file* pada Gambar 4.4 di bawah.

 model_quant.tflite	28/08/2020 05.15	TFLITE File	723 KB
 model_unquant.tflite	28/08/2020 05.17	TFLITE File	2.047 KB

Gambar 4.9 Ukuran TFLite *file* Model

Pada Gambar 4.9 merupakan perbandingan ukuran dari model yang telah di *generate* dari Teachable Machine. Ukuran dari model *unquant* memiliki ukuran yang lebih besar dengan tipe data *float* 32.

4.1.1. Uji Coba Performa Aplikasi

Pada pengujian ini dilakukan untuk membandingkan model TFLite *Floating Point* dan juga dengan objek gambar yang sama yang diperoleh dari beberapa situs secara *real time*, performa pada aplikasi dapat dilihat melalui fitur *Interference Time* atau kecepatan aplikasi dalam mengklasifikasikan gambar. Berdasarkan Tabel 4.10 merupakan hasil uji coba performa aplikasi dengan menggunakan NNAPI, GPU dan CPU.

Tabel 4.10 Pengaruh CPU Terhadap Performa Aplikasi

Percobaan	<i>Interference Time</i> (ms)	<i>Confidence Score</i> (%)
1	35	98,36
2	32	97,10
3	30	90,98
4	30	90,40
5	30	90,50

Berdasarkan Tabel 4.10 di atas merupakan hasil dari percobaan untuk mengetahui performa aplikasi pada CPU, dimana percobaan dilakukan sebanyak tiga kali untuk mendapatkan perbandingan yang sesuai. Titik terbesar pada *interference time* mencapai 35 ms dengan *score* 98,36%, untuk titik terendah sebesar 30 ms dengan *score* 90,40%.

Pada percobaan ini CPU berperan penting dalam proses pengklasifikasian, selain itu cpu merupakan *hardware* dasar dalam mengoptimalkan aplikasi. Sehingga proses pengklasifikasian dapat berjalan sempurna. Berdasarkan Tabel 4.11 merupakan hasil pengaruh NNAPI terhadap performa aplikasi.

Tabel 4.11 Pengaruh NNAPI Terhadap Performa Aplikasi

Percobaan	<i>Interference Time</i> (ms)	<i>Confidence Score</i>
1	25	85,36%
2	26	86,10%
3	25	87,98%
4	25	87,40%
5	25	87,50%

Berdasarkan Tabel 4.11 di atas merupakan hasil dari percobaan untuk mengetahui performa aplikasi pada NNAP. Nnnpi merupakan *software* yang dirancang untuk menjalankan operasi komputasi intensif bagi *machine learning* pada perangkat Android. Pada percobaan ini terdapat lima kali percobaan agar dapat mengetahui perbandingan yang terjadi.

Dapat disimpulkan bahwa *interference time* pada NNAPI menghasilkan kecepatan yang paling lambat dibandingkan dengan CPU, hal ini disebabkan CPU merupakan *hardware* dasar pada perangkat aplikasi pengklasifikasian. Selain itu ada beberapa faktor yang dapat membuat performa NNAPI lebih lambat dibandingkan dengan CPU salah satu nya ialah karena *driver* NNAPI pada

perangkat aplikasi tidak tersedia, sehingga salah satu nya menjadi penyebab proses pengklasifikasian pada aplikasi berjalan lambat.

Tabel 4.12 pengaruh GPU Terhadap Performa Aplikasi

Percobaan	Interference Time (ms)	Confidence Score
1	45	98,36%
2	45	97,30%
3	47	92,88%
4	47	92,30%
5	45	92,20%

Berdasarkan Tabel 4.12 di atas merupakan hasil dari percobaan untuk mengetahui performa aplikasi pada GPU. Pada percobaan terdapat lima kali percobaan agar dapat mengetahui perbandingan yang terjadi. Dapat disimpulkan bahwa performa GPU menghasilkan akselerasi yang hampir sama dengan CPU. Perbedaan nya dalam hal ini dikarenakan GPU sendiri dikhususkan untuk memproses grafis sedangkan CPU memproses berbagai intruksi umum sementara.

Pada percobaan ini juga terdapat beberapa kendala dalam melakukan percobaan performa aplikasi menggunakan *device* GPU. Hal tersebut disebabkan dalam pemrosesannya membutuhkan spesifikasi perangkat yang sangat *powerfull*, karena pada *device* GPU mengandalkan performa grafis dalam proses pengklasifikasiannya.