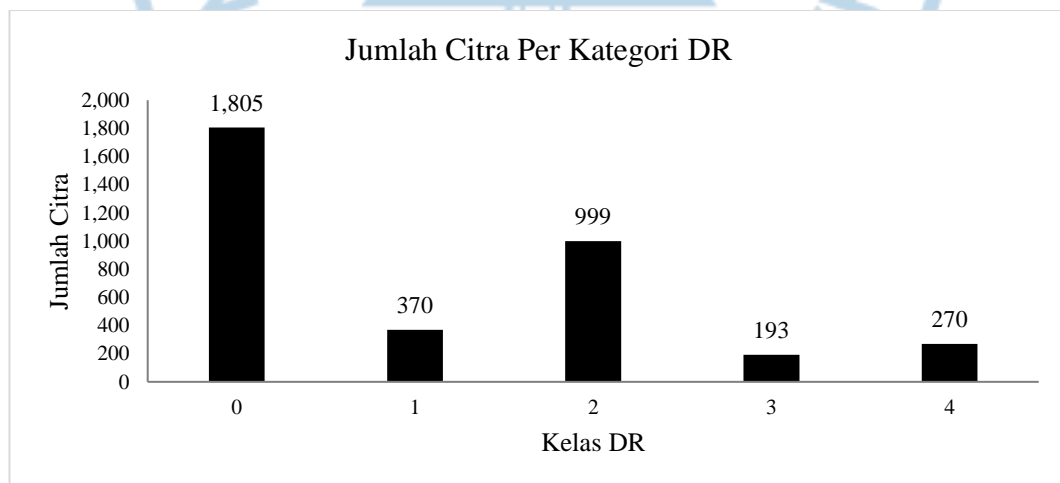


BAB 5 HASIL DAN DISKUSI

Hasil dari setiap metodologi penelitian yang diajukan oleh BAB 4 akan dijabarkan. Mulai dari kondisi dataset APTOS 2019; efek augmentasi dan visualisasi augmentasi yang diberikan selama *training*; visualisasi setiap tahap dari masing-masing *preprocessing*; dan hasil *training* model yang menitikberatkan pada performa model khususnya akurasi.

5.1 *Exploratory Data Analysis*

Dataset APTOS 2019 yang digunakan untuk *exploratory data analysis* diambil dari set *training* dengan jumlah 3.662 citra. Jumlah citra tersebut terbilang kecil untuk *training* model *deep learning* sehingga sangat mungkin model akan mengalami *overfitting*. Salah satu cara untuk mengatasi hal ini adalah menggunakan teknik augmentasi. Augmentasi yang dimaksud dibahas lebih detail pada sub-bab 5.3. Jumlah citra dari masing-masing kelas bisa dilihat pada Gambar 5.1. Tampak bahwa kelas 0-normal memiliki jumlah citra paling banyak. Kelas yang tidak seimbang ini diatasi dengan menggunakan bobot kelas seperti yang dilakukan oleh Harry Pratt [31] dan Gabriel [34].



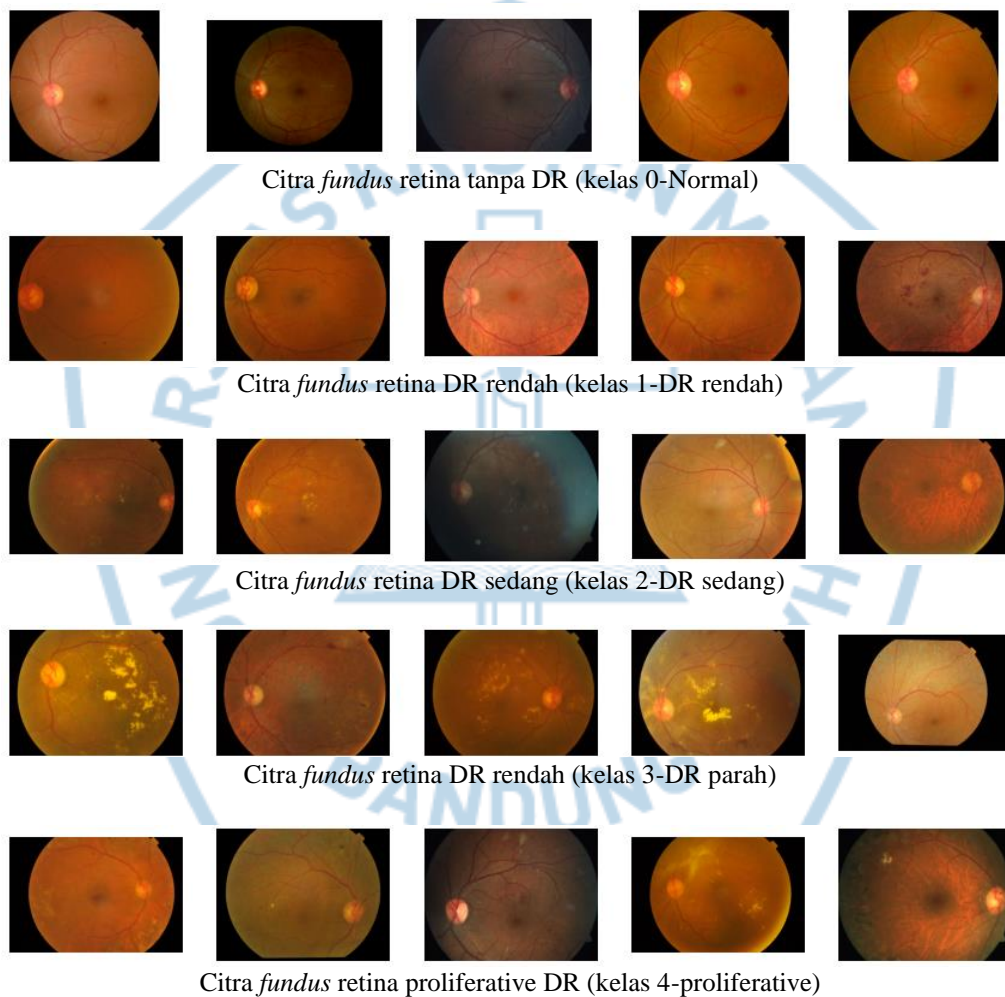
Gambar 5.1 Jumlah citra dari setiap kelas DR

Citra *fundus* retina pada dataset APTOS 2019 memiliki kondisi yang bervariasi, seperti tingkat iluminasi, bentuk retina, dan ukuran citra. Pada Gambar

5.2 dapat dilihat sampel dari citra *fundus* retina untuk setiap kelas DR. Dari gambar tersebut dapat dilihat bahwa:

- Iluminasi citra beragam, ada yang lebih gelap dan ada yang lebih terang
- Bentuk citra retina berbeda-beda, ada yang melingkar, ada yang melingkar namun terpotong bagian atas atau bawahnya.

Kondisi dataset yang bervariasi tersebut akan diatasi dengan *preprocessing* seperti yang dibahas pada sub-bab 4.4.3 sampai 4.4.6.



Gambar 5.2 Citra *fundus* retina dari setiap kelas DR

5.2 Hasil *Preprocessing*

Hasil dari *preprocessing* akan membentuk dataset baru, yaitu dataset dengan citra yang sudah diubah ukurannya, dataset Ben Graham, dataset Nakhon, dataset Ramasubramanian, dan dataset *enhanced green*. Sehingga terdapat lima

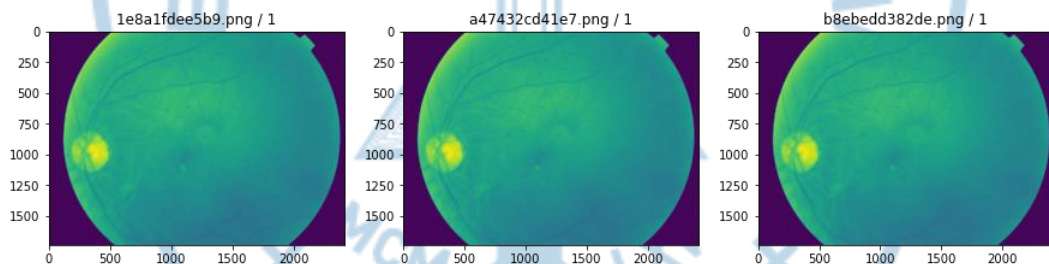
dataset yang digunakan untuk *training* model sesuai dengan *preprocessing* yang dilalui.

5.2.1 Menghapus Duplikasi Citra

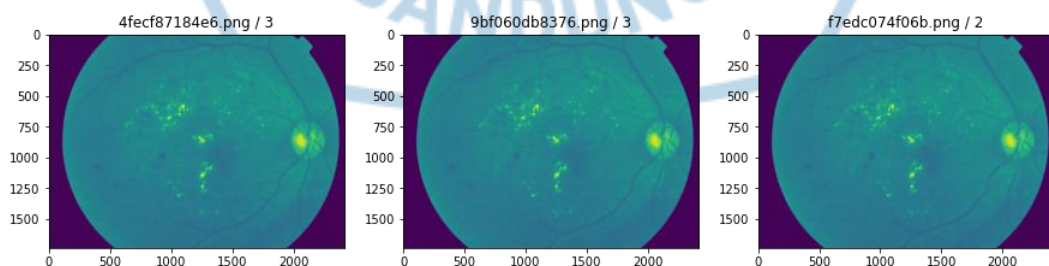
Algoritma *difference hashing* digunakan untuk memeriksa apakah dalam dataset APTOS 2019 terdapat duplikasi citra. Algoritma *difference hashing* akan membentuk *signature* yang dibentuk dari pixel citra itu sendiri. Dari hasil pemeriksaan ditemukan citra dengan duplikasi sebagai berikut:

- Satu buah citra dengan empat duplikasi
- Empat buah citra dengan tiga duplikasi
- 121 citra dengan dua duplikasi

Pada Gambar 5.3 dapat dilihat duplikasi citra sebanyak tiga buah dengan label yang sama. Label terletak pada bagian atas citra retina. Citra duplikasi seperti ini diambil satu citra saja. Namun jika citra duplikasi memiliki label yang berbeda seperti pada Gambar 5.4 maka citra dieliminasi semuanya dari dataset.



Gambar 5.3 Contoh gambar duplikasi dengan label yang sama



Gambar 5.4 Contoh gambar duplikasi dengan label yang berbeda

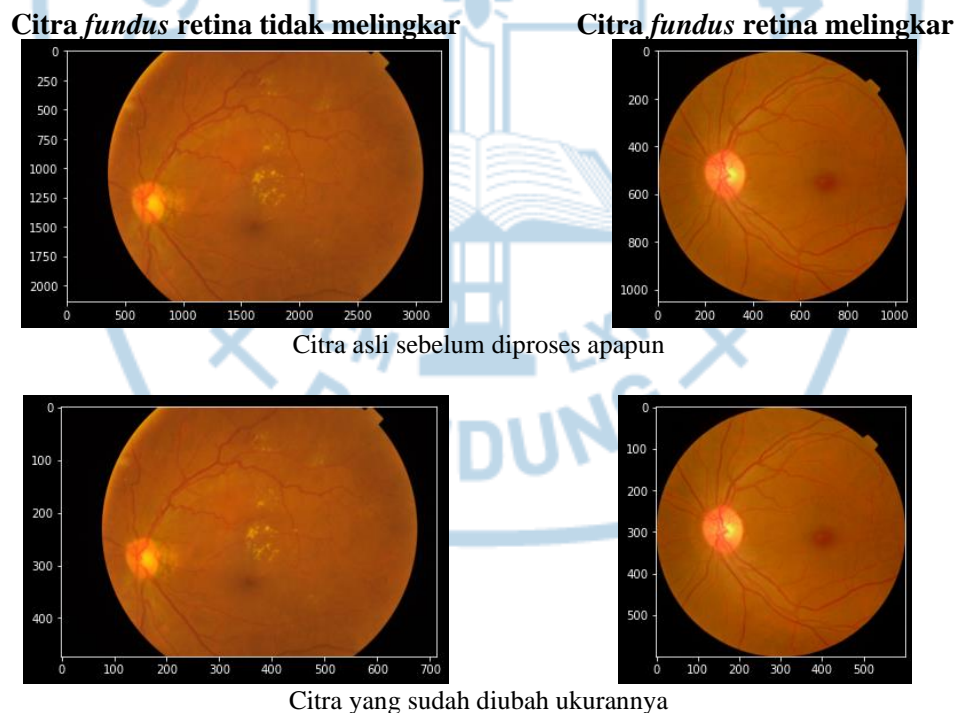
Setelah citra dibersihkan dari duplikasi data, maka diperoleh total dataset yang digunakan adalah 3.498 seperti yang dapat dilihat pada Tabel 5.1.

Tabel 5.1 Jumlah Citra Sebelum dan Setelah Menghapus Duplikasi

| Tingkat DR | Sebelum | Setelah |
|------------|---------|---------|
| 0 | 1.805 | 1.796 |
| 1 | 370 | 338 |
| 2 | 999 | 921 |
| 3 | 193 | 173 |
| 4 | 270 | 270 |
| Total | 3.662 | 3.498 |

5.2.2 Mengubah Ukuran Citra

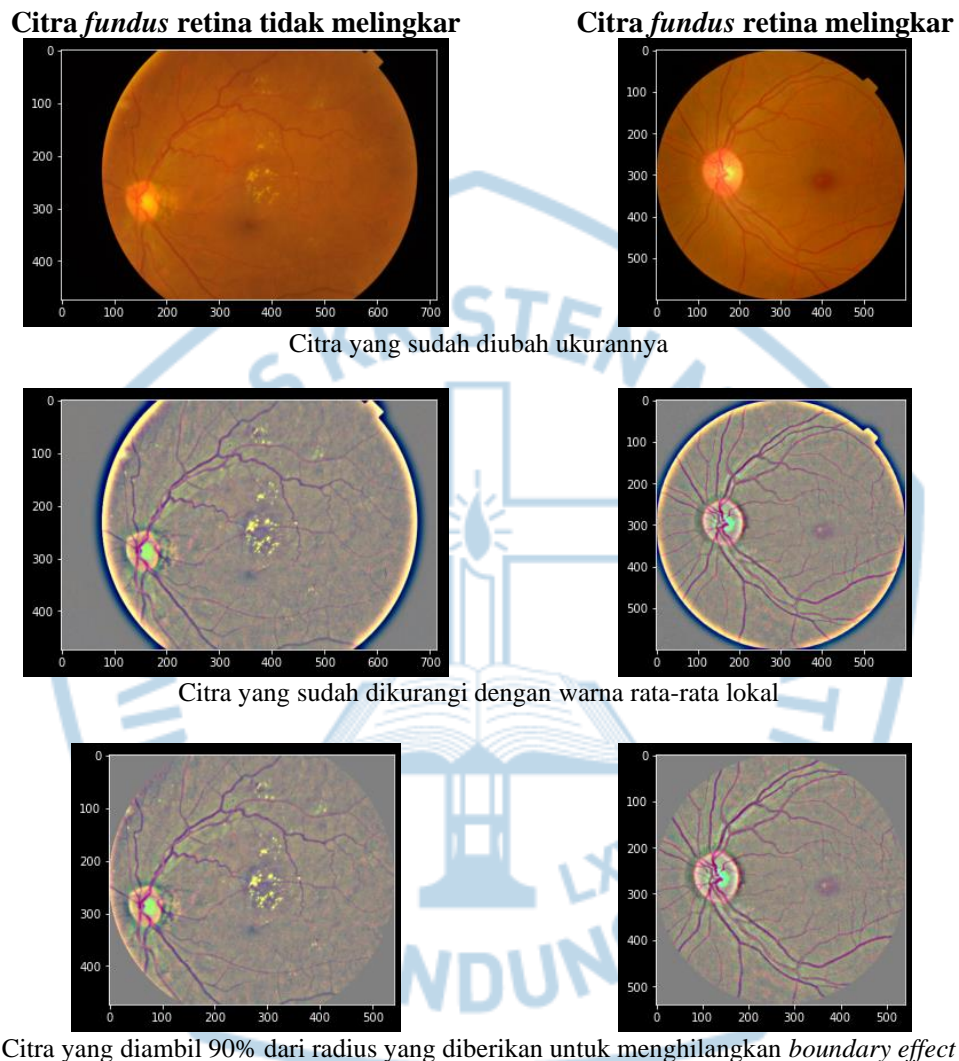
Citra pada dataset diubah ukurannya dengan teknik yang digunakan oleh Ben Graham. Pengubahan ukuran citra ini bertujuan untuk mengurangi beban komputasi selama proses *training* ataupun *preprocessing* lainnya. Citra diubah ukurannya dengan radius *pixel* 300 pada area retina. Radius pixel 300 adalah angka yang cukup mengingat citra *training* akan berukuran $299 \times 299 \times 3$. Pada Gambar 5.5 dapat dilihat hasil citra yang sudah diubah ukurannya dengan radius 300 *pixel* untuk area retina.

Gambar 5.5 Hasil *preprocessing* mengubah ukuran citra

5.2.3 *Preprocessing* Ben Graham

Hasil *preprocessing* Ben Graham tampak seperti pada Gambar 5.6. Tujuan *preprocessing* ini adalah membuang variasi antara citra seperti kondisi iluminasi,

resolusi kamera, dan lain-lain [13]. Pada gambar dapat dilihat bahwa kontras dari citra lebih homogen karena *preprocessing* ini mengurangi warna rata-rata lokal dan dipetakan ke *grayscale*. Efek pemotongan melingkar sebanyak 10% dari radius yang diberikan menghilangkan “*boundary effect*”.



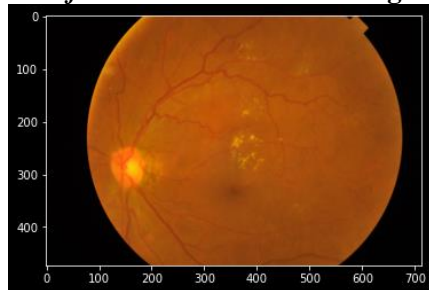
Gambar 5.6 *Preprocessing* Ben Graham

5.2.4 *Preprocessing* Nakhon

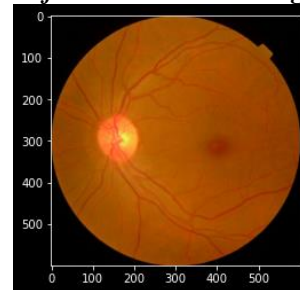
Preprocessin Nakhon mengadopsi *preprocessing* yang dilakukan Ben Graham untuk mengatasi kondisi variasi citra. Namun Nakhon tidak membuang “*boundary effect*” sehingga dapat dilihat terdapat garis melingkar pada citra *fundus* retina. Selain itu *preprocessing* ini memotong setiap citra dengan melingkar sempurna, sehingga bagian retina akan terpotong jika citra *fundus* dari awalnya

tidak melingkar. Jika dilihat pada Gambar 5.7, citra yang dari semula tidak melingkar akan kehilangan sebagian area retina ketika dipotong melingkar.

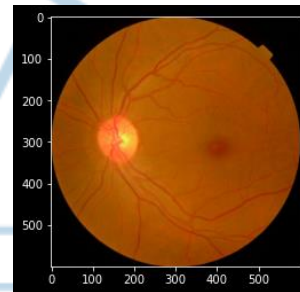
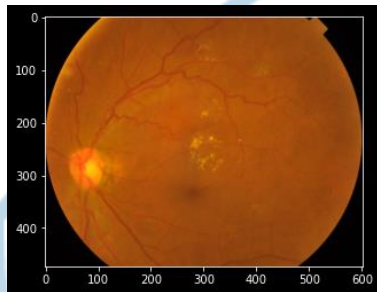
Citra *fundus* retina tidak melingkar



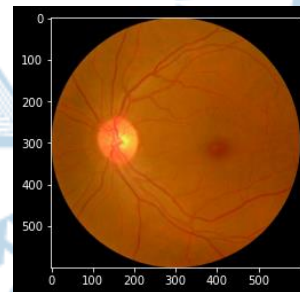
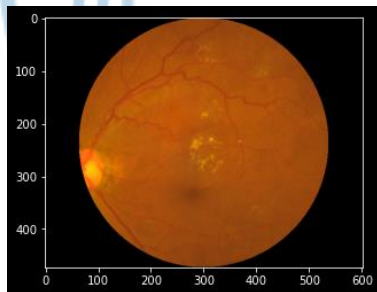
Citra *fundus* retina melingkar



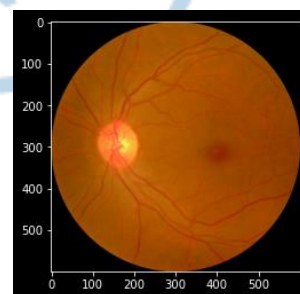
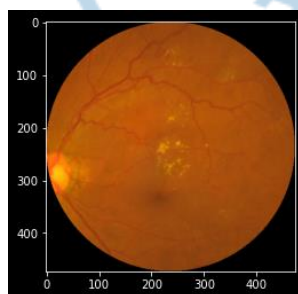
Citra yang sudah diubah ukurannya



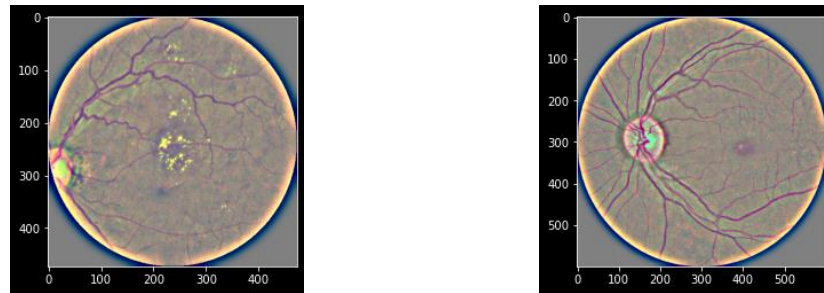
Citra yang dibuang margin warna latar hitam



Citra yang dipotong melingkar



Citra dibuang bagian margin warna hitam lagi



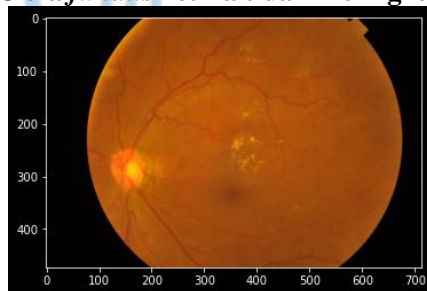
Citra yang sudah dikurangi dengan warna rata-rata lokal

Gambar 5.7 Hasil *preprocessing* Nakhon

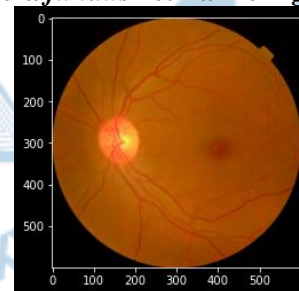
5.2.5 *Preprocessing* Ramasubramanian

Preprocessing Ramasubramanian membuang *salt and pepper noise* dengan menggunakan *median filter* dan dilanjutkan dengan CLAHE agar fitur tersembunyi pada citra *fundus* retina lebih jelas seperti *exudates*, *microaneurysms*, *fovea*, dan pembuluh darah. *Preprocessing* ini hanya menggunakan *channel* hijau saja yang dijadikan tiga *channel*. Visualisasi tahap *preprocessing* Ramasubramanian dapat dilihat pada Gambar 5.8.

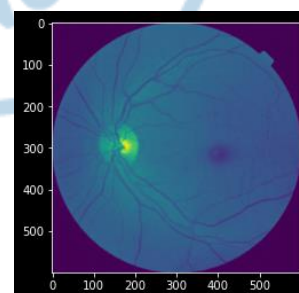
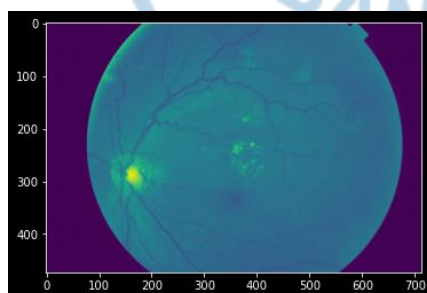
Citra *fundus* retina tidak melingkar



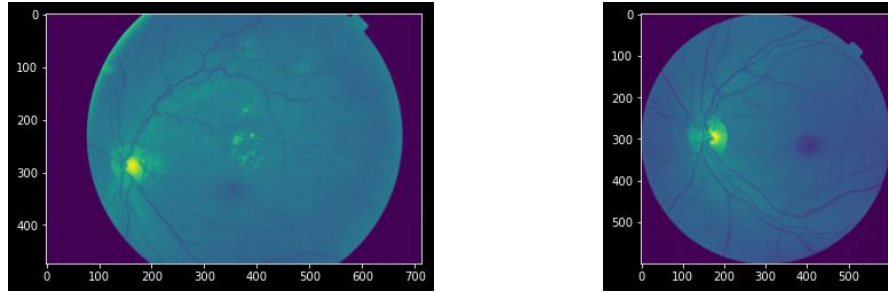
Citra *fundus* retina melingkar



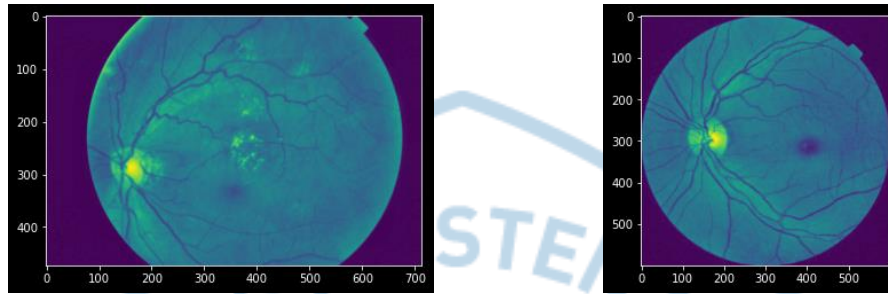
Citra yang sudah diubah ukurannya



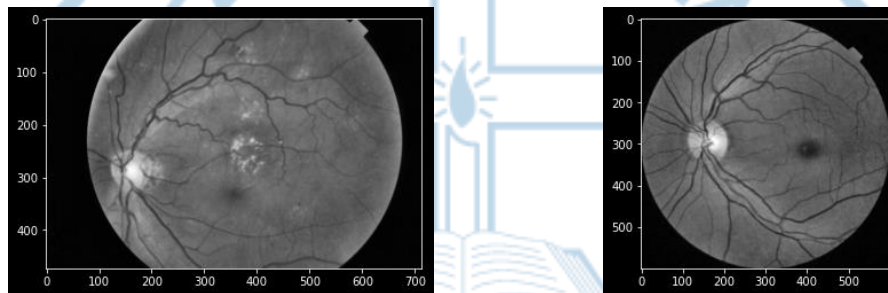
Citra diambil *channel* hijau saja



Citra yang sudah dibuang *salt and pepper noise* dengan *median filter*



Citra yang diperbaiki kontrasnya dengan CLAHE

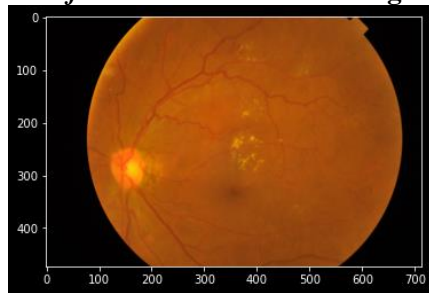
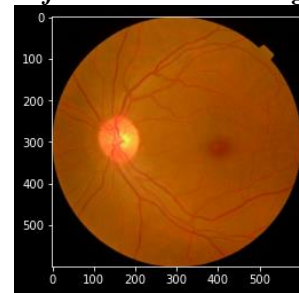


Citra tiga *channel* yang merupakan *channel* hijau semua

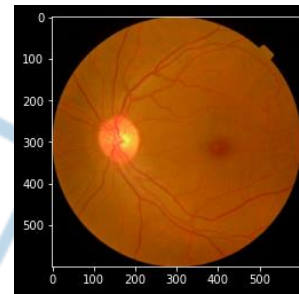
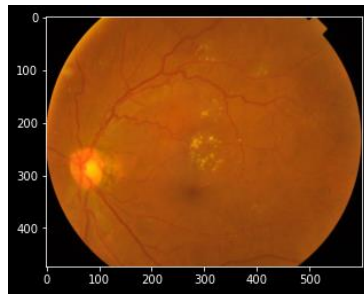
Gambar 5.8 Hasil *preprocessing* Ramasubramanian

5.2.6 *Preprocessing Enhanced Green*

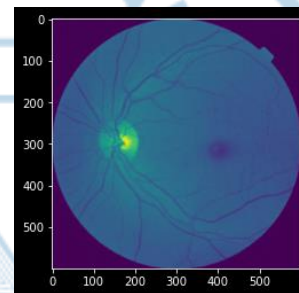
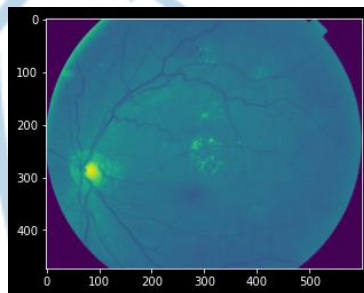
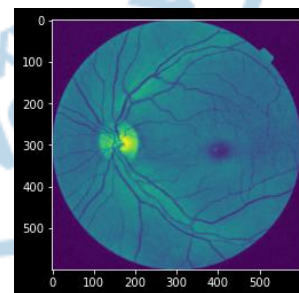
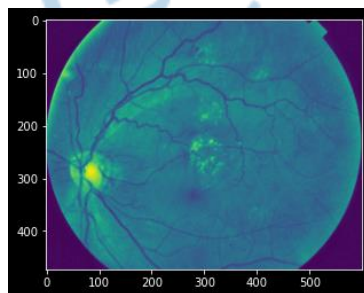
Preprocessing enhanced green mengatasi kondisi dataset yang beragam dari dua aspek. Pertama memperbaiki kontras pada citra dengan CLAHE. Kedua menajamkan citra dengan *unsharp masking* sehingga garis-garis batas antara objek seperti titik-titik hitam ataupun garis-garis pembuluh darah pada citra *fundus* retina tampak lebih jelas. *Preprocessing enhanced green*, seperti namanya, hanya menggunakan *channel* hijau saja yang dibuat menjadi tiga *channel*. Adapun tahapan dari *preprocessing enhanced green* dapat dilihat pada Gambar 5.9.

Citra *fundus* retina tidak melingkarCitra *fundus* retina melingkar

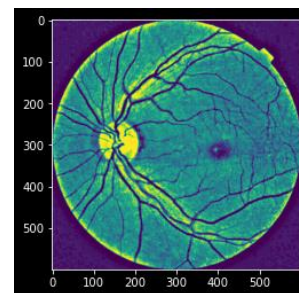
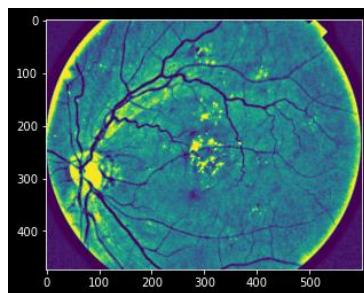
Citra yang sudah diubah ukurannya

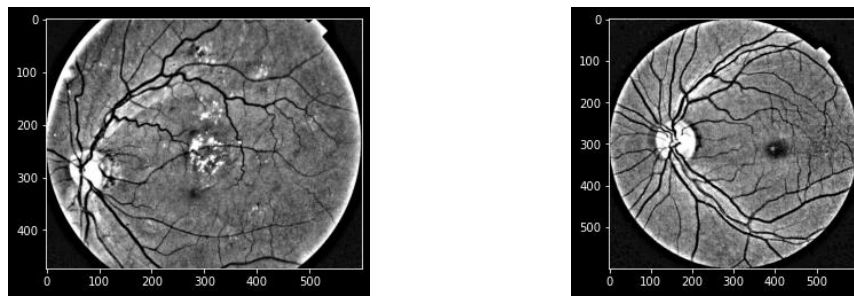


Citra yang dibuang margin warna latar hitam

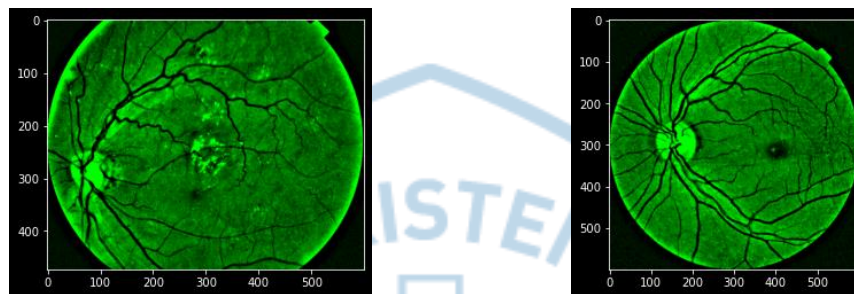
Citra diambil *channel* hijau saja

Citra yang diperbaiki kontrasnya dengan CLAHE

Citra yang ditingkatkan dengan *unsharp masking*



Citra tiga *channel* yang merupakan *channel* hijau semua (G, G, G)



Citra tiga *channel* (R, G, B) di mana nilai *channel* R dan B adalah 0

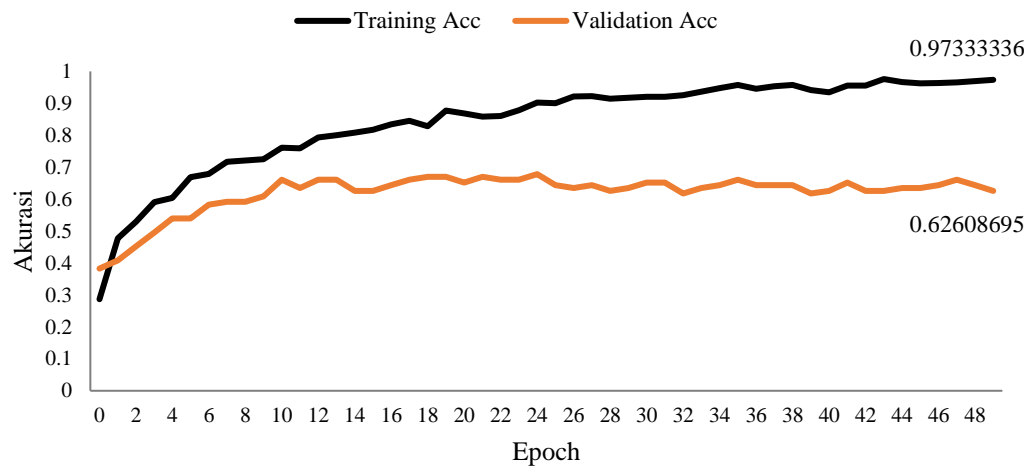
Gambar 5.9 Hasil *preprocessing enhanced green*

5.3 Augmentasi

Seperti yang sudah dijelaskan pada sub-bab 4.5 bahwa tujuan augmentasi adalah memberikan data dengan berbagai transformasi agar model memberikan hasil yang lebih generalisasi. Hal ini dibuktikan dengan melakukan percobaan kecil. Model tanpa augmentasi dan dengan augmentasi di-*training* dengan dataset yang terdiri dari 750 citra *training* dan 115 citra validasi. Citra ini dipilih secara acak dan dibuat seimbang untuk setiap kelasnya. Arsitektur yang digunakan adalah arsitektur *fastai modified* (Gambar 4.15) dengan *hyperparameter* seperti pada Tabel 4.3 dan Tabel 4.5. Visualisasi augmentasi yang diterapkan pada citra *training* dapat dilihat pada Gambar 5.14 sampai dengan Gambar 5.20.

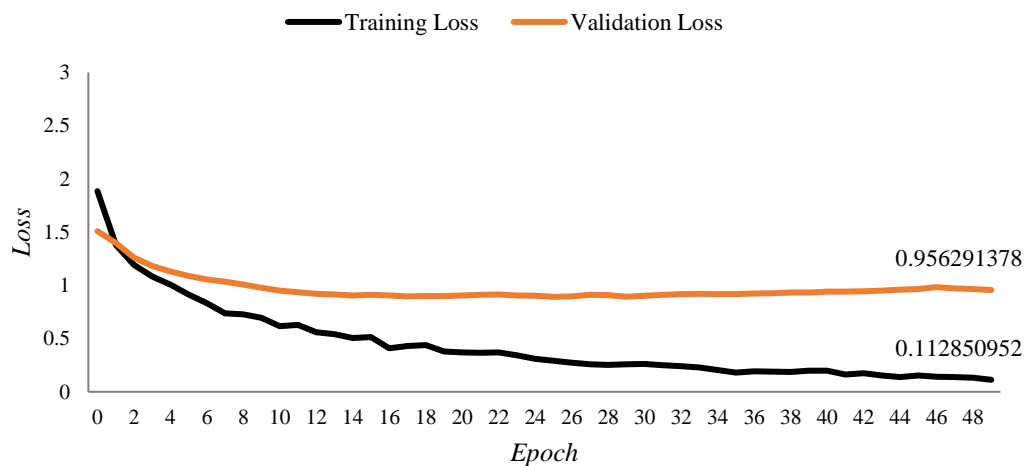
Performa model tanpa augmentasi dapat dilihat pada Gambar 5.10 dan Gambar 5.11. Pada kedua gambar tersebut tampak bahwa model *overfit* terhadap dataset *training*. Hal ini ditandai dengan akurasi saat *training* hampir 100%, sedangkan akurasi dari dataset validasi sekitar 60%. Demikian juga dengan nilai *loss* yang sangat kecil pada dataset *training* jika dibandingkan dengan nilai *loss* dari dataset validasi.

Grafik Nilai Akurasi Model Fastai Modified Tanpa Augmentasi



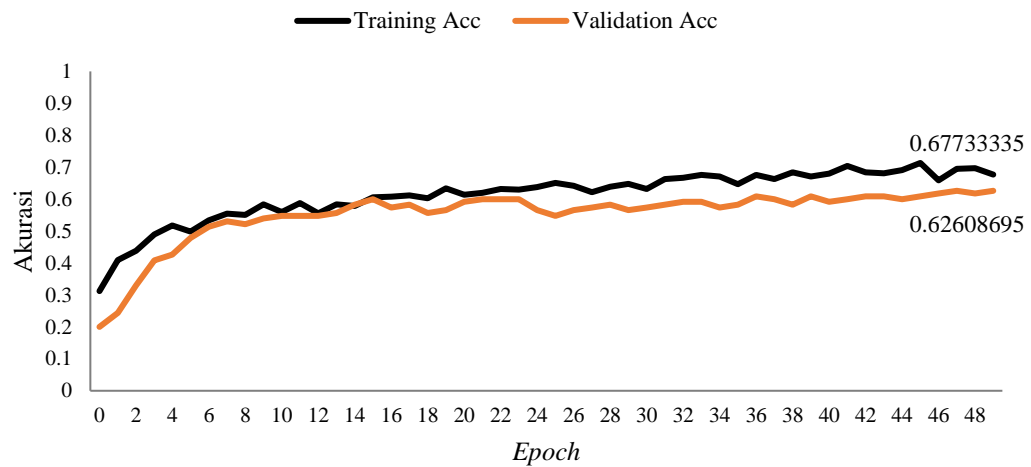
Gambar 5.10 Grafik nilai akurasi tanpa augmentasi

Grafik Nilai Loss Model Fastai Modified Tanpa Augmentasi

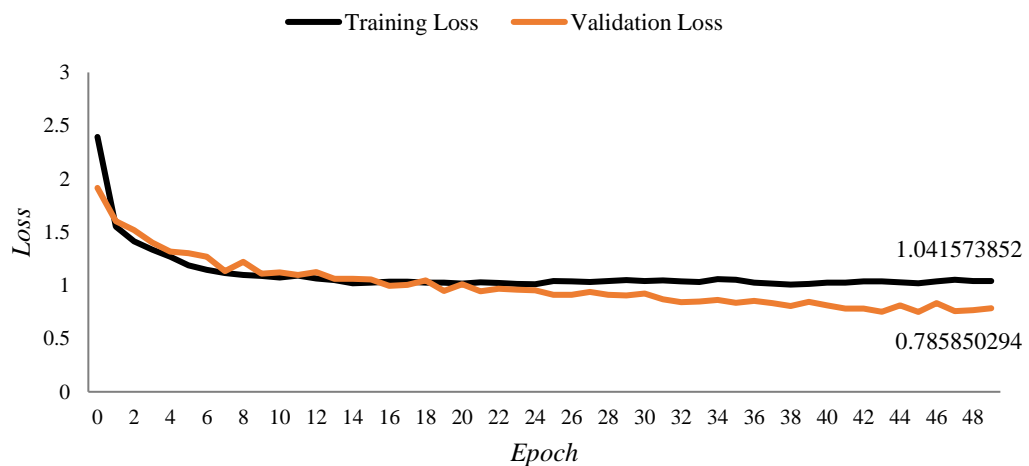


Gambar 5.11 Grafik nilai loss tanpa augmentasi

Performa model yang di-*training* dengan augmentasi ditunjukkan oleh Gambar 5.12 dan Gambar 5.13. Pada dua gambar tersebut tampak bahwa nilai akurasi dan nilai *loss training* hampir beriringan dengan akurasi dan *loss* validasi. Hal ini memberikan gambaran bahwa model dengan augmentasi memberikan performa yang lebih general dibandingkan dengan model tanpa augmentasi.

Grafik Akurasi Model *Fastai Modified* Dengan Augmentasi

Gambar 5.12 Grafik nilai akurasi dengan augmentasi

Grafik Nilai *Loss* Model *Fastai Modified* Dengan Augmentasi

Gambar 5.13 Grafik nilai loss dengan augmentasi

Hasil evaluasi model diukur dari dataset validasi setelah training sebanyak 50 *epoch* selesai. Hasil dari model yang di-*training* tanpa augmentasi dan dengan augmentasi dapat dilihat pada Tabel 5.2. Meskipun kedua model ini memiliki nilai akurasi yang sama, namun jika dilihat dilihat selama training tampak bahwa model tanpa augmentasi *overfitting* terhadap dataset *training*.

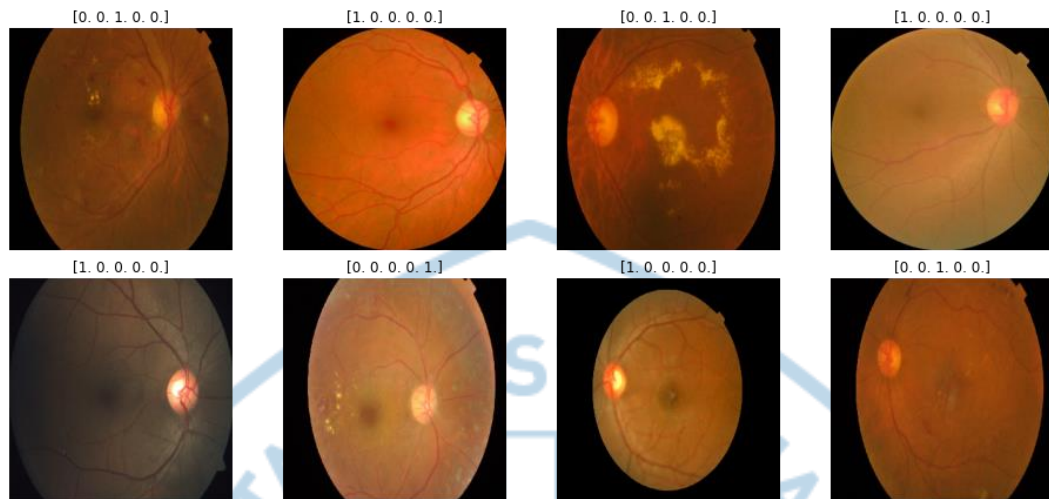
Tabel 5.2 Performa model tanpa dan dengan augmentasi

| Model | Loss | Accuracy |
|-------------------|--------|----------|
| Tanpa Augmentasi | 0.9563 | 62.61% |
| Dengan Augmentasi | 1.0416 | 62.61% |

5.3.1 Visualisasi Augmentasi

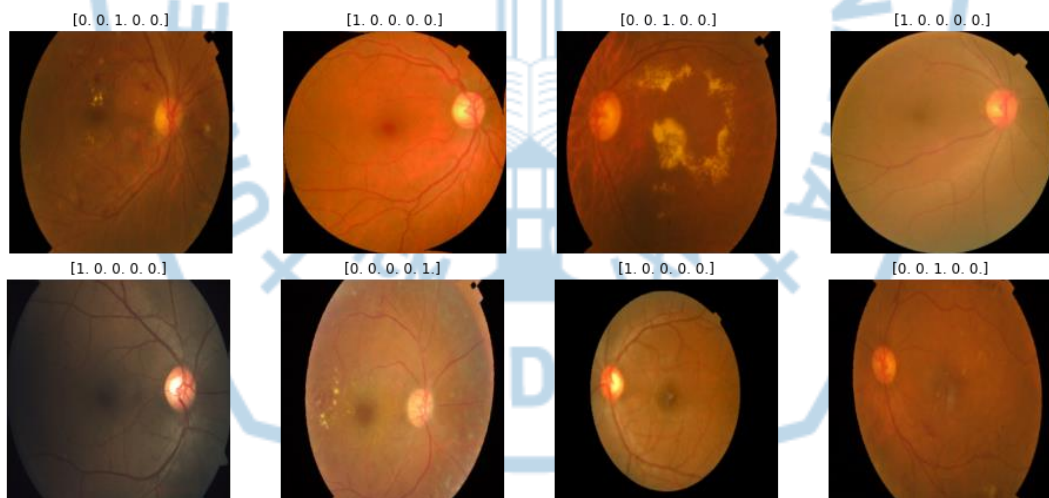
Visualisasi dari setiap augmentasi yang dilakukan dapat dilihat pada Gambar 5.15 sampai Gambar 5.20.

- Tanpa augmentasi



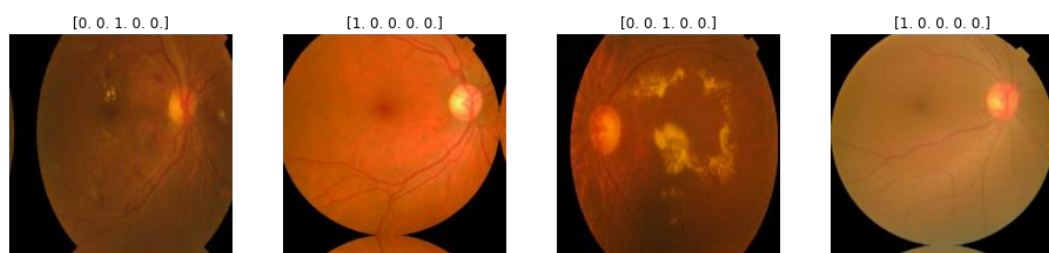
Gambar 5.14 Citra tanpa augmentasi

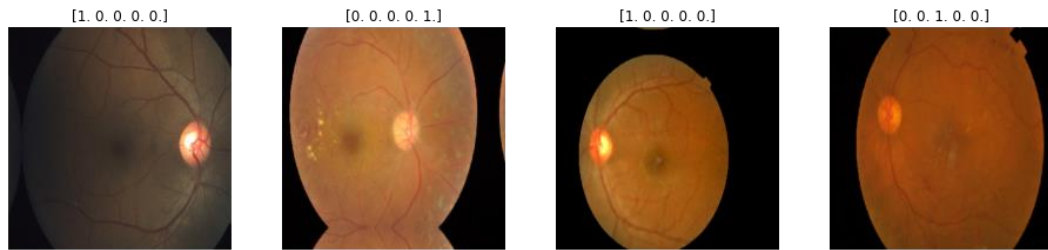
- Augmentasi rotasi citra 10°



Gambar 5.15 Citra dengan augmentasi rotasi 10°

- Pergeseran citra pada sumbu x dan sumbu y

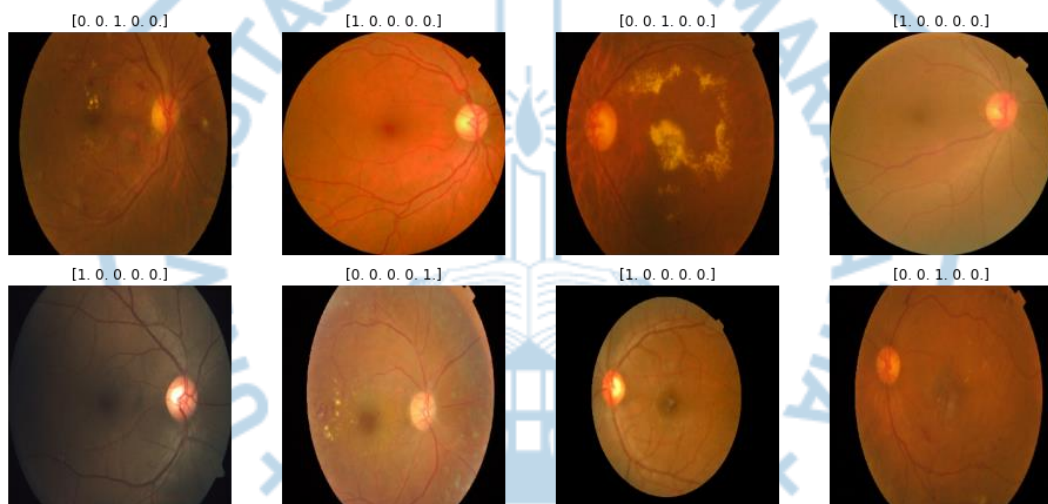




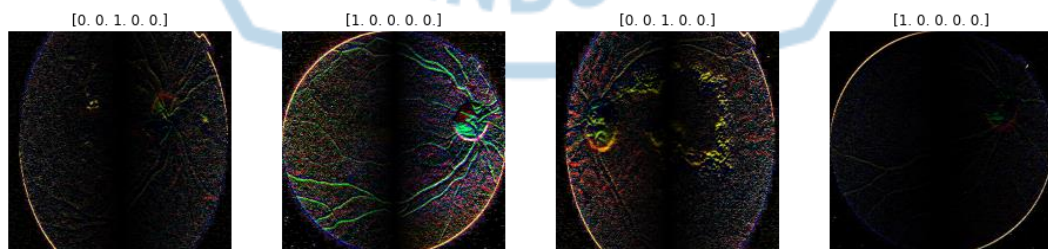
Gambar 5.16 Citra dengan augmentasi pergeseran di sumbu x dan y

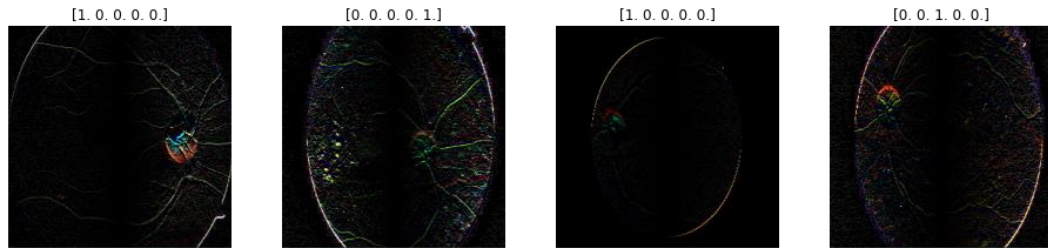
- *Shear* secara acak.

Jika dilihat dengan kasat mata, augmentasi ini tampak tidak memberikan perubahan terhadap citra. Namun jika dilihat selisih citra sebelum dan setelah augmentasi *shear* tampak perubahan yang dilakukan dengan augmentasi ini seperti pada Gambar 5.18.



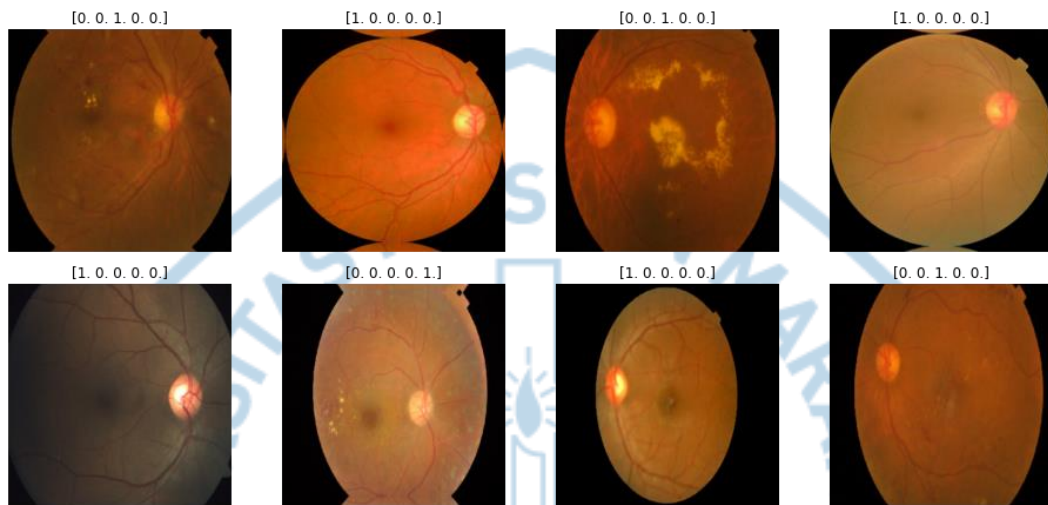
Gambar 5.17 Citra dengan augmentasi *shear*





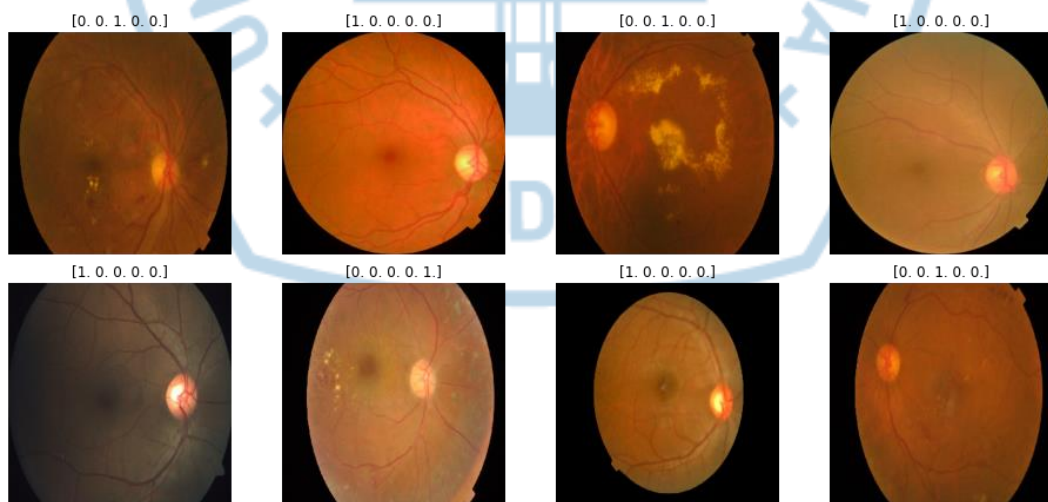
Gambar 5.18 Selisih citra asli dengan citra augmentasi shear

- Augmentasi citra dengan *zoom* 90% - 110%



Gambar 5.19 Citra dengan augmentasi zoom 90% - 110%

- Augmentasi citra dengan *flipping* horizontal dan vertikal



Gambar 5.20 Citra dengan augmentasi flipping horizontal dan vertikal

5.4 Hasil *training* model

Sub-bab ini membahas performa dari setiap hasil *training* model yang diusulkan pada sub-bab 4.6. Titik berat performa model adalah nilai akurasi yang

diukur dari dataset validasi. Training model yang dilakukan meliputi: pemilihan arsitektur *classifier* (lihat sub-bab 4.6.1), *training* model dengan pendekatan *end to end learning* (lihat sub-bab 4.6.2), *training* model dengan pendekatan *transfer learning* yang datasetnya dilalui beberapa tahap *preprocessing* (lihat sub-bab 4.6.3), *fine tuning* *n*-blok Inception v3 (lihat sub-bab 4.6.4), dan *training* model *transfer learning* untuk membandingkan citra dengan *file format* PNG dengan JPEG (lihat sub-bab 4.6.3).

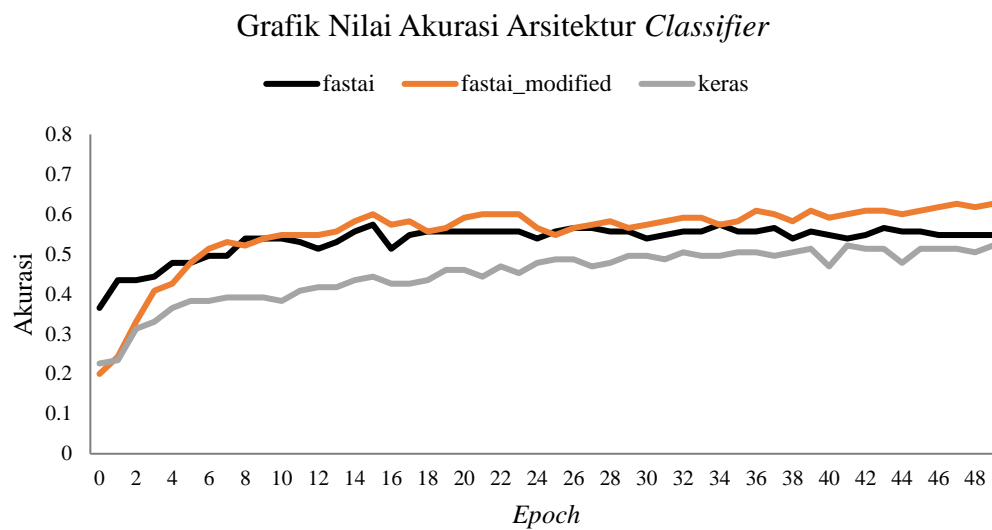
5.4.1 Pemilihan Arsitektur Classifier

Hasil *training* tiga arsitektur *classifier* yang diuji pada penelitian ini (arsitektur yang dimaksud bisa dilihat pada sub-bab 4.6.1) bisa dilihat pada Tabel 5.3. Hasil tersebut diukur dari dataset validasi setelah *training* sebanyak 50 *epoch* selesai. Model *fastai_modified* memberikan performa paling baik dengan nilai loss 1.0416 dan akurasi 62.61%.

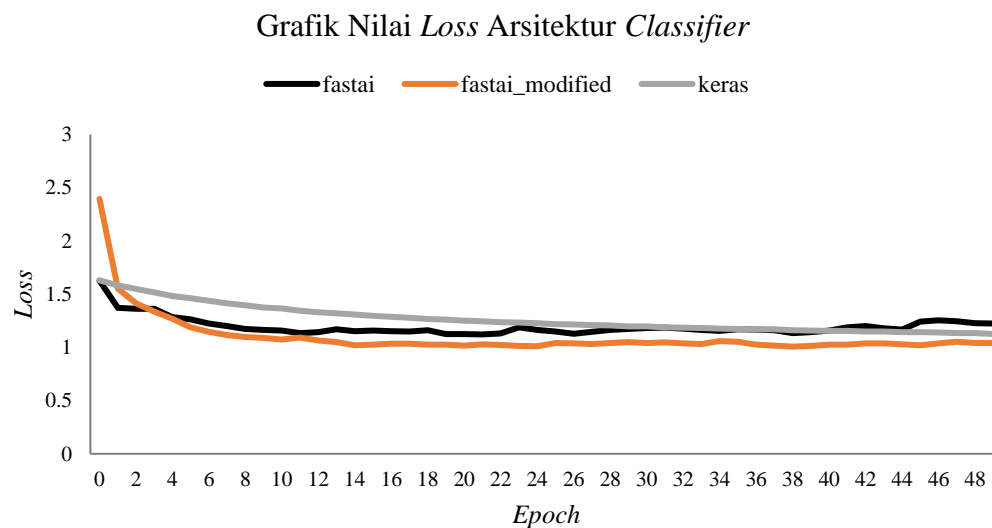
Tabel 5.3 Performa akurasi *classifier* yang ditambahkan pada bagian akhir Inception v3

| Model | Loss | Accuracy |
|------------------------|--------|----------|
| <i>fastai</i> | 1.2238 | 54.78% |
| <i>fastai_modified</i> | 1.0416 | 62.61% |
| <i>keras</i> | 1.1260 | 52.17% |

Jika dibandingkan dengan model *keras*, baik *fastai* maupun *fastai_modified* memberikan hasil lebih baik dari sisi *loss* dan akurasi. Hal ini dikarenakan *fastai* dan *fastai_modified* menggunakan fitur dua kali lebih banyak jika dibandingkan dengan *keras* dan memiliki *hidden layer* dan *hidden unit* yang lebih banyak juga. Semakin dalam sebuah model, maka kemampuan ataupun kapasitas model untuk memetakan data *input* ke *output* akan semakin baik. Arsitektur *fastai_modified* ini yang digunakan untuk *training* model selanjutnya. Performa model selama *training* bisa dilihat pada Gambar 5.21 untuk nilai akurasi dan Gambar 5.22 untuk nilai *loss*. Grafik akurasi menunjukkan bahwa *fastai_modified* memberikan margin akurasi lebih tinggi jika dibandingkan dengan dua model lainnya. Grafik *loss* menunjukkan bahwa model *fastai_modified* lebih cepat konvergen jika dibandingkan dengan dua model lainnya.



Gambar 5.21 Grafik nilai akurasi dari tiga arsitektur classifier



Gambar 5.22 Grafik nilai loss dari tiga arsitektur classifier

5.4.2 End to end learning

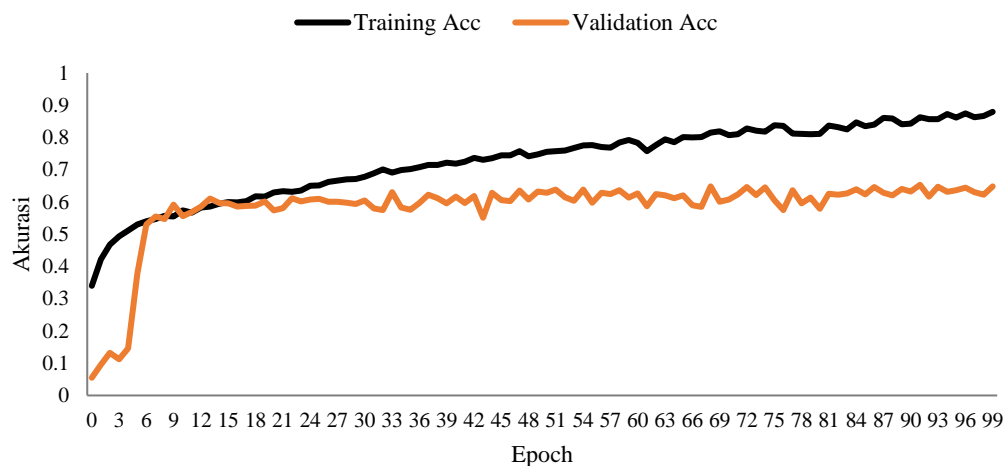
Salah satu tantangan *training* model dengan pendekatan *end to end learning* adalah ketersediaan dataset. Menurut Ian Goodfellow dan kawan-kawan [17], *rule of thumb training* model *deep supervised learning* setidaknya membutuhkan 5.000 data per kategori. *Overfit* menjadi tantangan ketika *training* model dengan dataset dalam ukuran kecil. *Overfit* ditandai dengan nilai akurasi yang diukur dari dataset *training* jauh lebih baik dibandingkan dengan nilai akurasi yang diukur pada dataset validasi. Demikian juga untuk nilai *loss*, nilai *loss* yang diukur pada dataset *training*

jauh lebih kecil jika dibandingkan dengan nilai *loss* yang diukur pada dataset validasi. Jika dilihat pada Gambar 5.23 dan Gambar 5.24 tampak bahwa model yang di-*training* dengan pendekatan *end to end learning* memberikan hasil yang *overfit*. Performa setelah *training* sebanyak 100 *epoch* selesai dapat dilihat pada Tabel 5.4, sedangkan *hyperparameter* untuk *training* model dapat dilihat pada Tabel 4.3 dan Tabel 4.6. Solusi atau alternatif untuk *training* model dengan kondisi keterbatasan dataset adalah dengan pendekatan *transfer learning*.

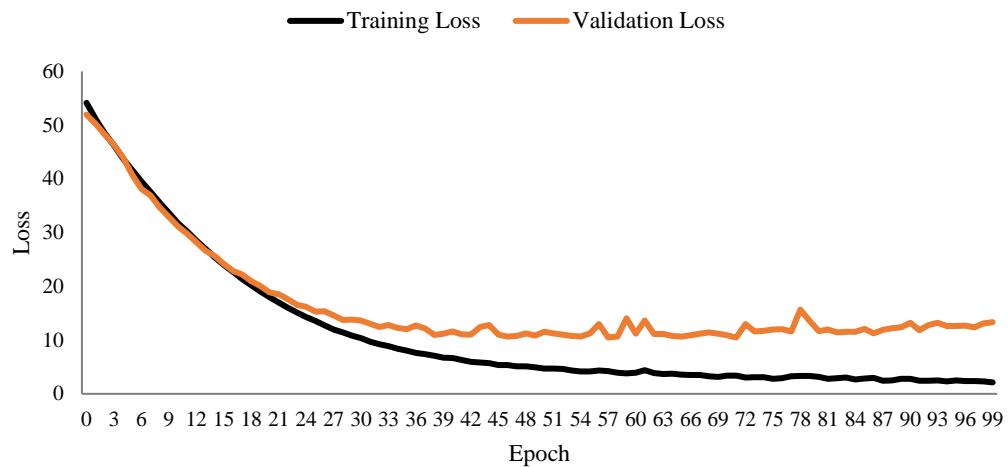
Tabel 5.4 Performa model *end to end learning*

| Model | Loss | Acc | Precision | Recall | AUC |
|---------|------|--------|-----------|--------|--------|
| Scratch | 1.58 | 64.84% | 66.74% | 63.21% | 88.04% |

Grafik Nilai Akurasi *Training* dan *Validation*



Gambar 5.23 Grafik nilai akurasi model *end to end learning*

Grafik Nilai *Loss Training* dan *Validation*Gambar 5.24 Grafik nilai loss model *end to end learning*

5.4.3 Pengaruh *Preprocessing* Terhadap Performa Model

Training model *transfer learning* diawali dengan *training* bagian *classifier* yang diikuti dengan *fine tuning* dua blok Inception v3 seperti yang sudah dijelaskan pada sub-bab 4.6.3. Hyperparameter *training* bisa dilihat pada Tabel 4.3 dan Tabel 4.7. *Training* dilakukan terhadap lima kondisi dataset sehingga menghasilkan lima model, yaitu:

- Model *rescale_png* adalah model yang di-*training* dengan dataset asli yang sudah diubah ukurannya.
- Model *ben_graham* adalah model yang di-*training* dengan dataset yang sudah melalui *preprocessing* Ben Graham.
- Model *nakhon* adalah model yang di-*training* dengan dataset yang sudah melalui *preprocessing* Nakhon.
- Model *ramasubramanian* adalah model yang di-*training* dengan dataset yang sudah melalui *preprocessing* Ramasubramanian.
- Model *enhanced_green* adalah model yang di-*training* dengan dataset yang sudah melalui *preprocessing enhanced green* di mana ketiga *channel* dari citra ini adalah *channel* hijau (G, G, G).
- Model *green_0g0* adalah model yang di-*training* dengan dataset yang sudah melalui *preprocessing enhanced green* di mana nilai *channel* R dan B adalah 0.

5.4.3.1 *Training Classifier*

Hasil *training* bagian *classifier* dari lima kondisi dataset dapat dilihat pada Tabel 5.5. Nilai ini diukur dari dataset validasi setelah *training* sebanyak 50 *epoch* selesai dan merupakan hasil rata-rata dari 5-fold *cross validation*. Pada tabel tersebut dapat dilihat bahwa *preprocessing* dengan metode *enhanced green* memberikan hasil yang paling baik dengan nilai: akurasi 76.10%, *precision* 81.92%, *recall* 69.90% dan AUC 95.12%. Model kedua terbaik diberikan oleh model *green_0g0* dengan nilai akurasi 75.59%. Model ketiga terbaik diberikan oleh *ramasubramanian* dengan nilai akurasi 74.10%. Sedangkan model *rescale_png*, *ben_graham*, dan *nakhon* menempati posisi keempat, kelima dan keenam. Dari keenam model ini dapat dilihat bahwa *preprocessing enhanced green* memberikan *benefit* terhadap performa model seperti yang ditunjukkan oleh model *enhanced green* dan model *green_0g0*.

Tabel 5.5 Metrik hasil *training classifier*

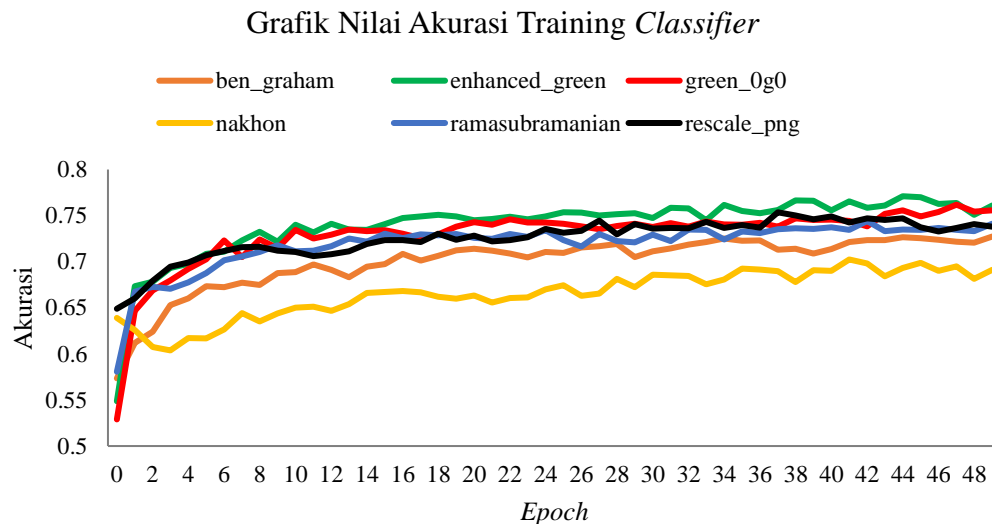
| Model | Loss | Acc | Precision | Recall | AUC |
|------------------------|--------|---------------|-----------|--------|--------|
| <i>ben_graham</i> | 2.0868 | 72.73% | 79.52% | 65.09% | 94.16% |
| <i>enhanced_green</i> | 1.9345 | 76.10% | 81.92% | 69.90% | 95.12% |
| <i>green_0g0</i> | 1.9047 | 75.59% | 81.17% | 69.21% | 94.99% |
| <i>nakhon</i> | 2.0161 | 69.10% | 78.70% | 60.09% | 93.05% |
| <i>ramasubramanian</i> | 1.9119 | 74.10% | 81.26% | 67.41% | 94.92% |
| <i>rescale_png</i> | 1.9378 | 73.78% | 80.70% | 66.81% | 94.53% |

F1 score untuk setiap kelas DR dari lima model ini dapat dilihat pada Tabel 5.6. Pada tabel tersebut kelas 0 memberikan nilai paling tinggi. Hal ini dikarenakan kelas 0 memiliki jumlah citra paling banyak (kolom *support*). Sedangkan kelas 3 memberikan nilai paling kecil karena jumlah citra kelas 3 paling sedikit meskipun pembobotan kelas selama *training* sudah diberikan. Dari tabel ini juga dapat dilihat bahwa F1 score tertinggi dari setiap kelas DR diberikan oleh model dengan *preprocessing* yang berbeda. DR kelas 0 diberikan oleh model *ramasubramanian*, DR kelas 1 diberikan oleh model *rescale_png*, DR kelas 2 dan 4 diberikan oleh model *enhanced_green*. DR kelas 3 diberikan oleh model *green_0g0*.

Tabel 5.6 F1 Score training classifier

| DR Level | rescale_png | graham | nakhon | ramasubramanian | enhanced_green | green_0g0 | Support |
|----------|--------------|--------|--------|-----------------|----------------|--------------|---------|
| 0 | 95.6% | 95.2% | 95.0% | 96.4% | 95.8% | 95.6% | 359 |
| 1 | 52.60 | 46.2% | 46.2% | 49.6% | 46.2% | 49.2% | 68 |
| 2 | 59.6% | 56.2% | 48.6% | 57.6% | 64.6% | 62.8% | 184 |
| 3 | 36.6% | 31.4% | 34.4% | 37.6% | 37.6% | 40.8% | 35 |
| 4 | 38.2% | 42.2% | 30.4% | 41.4% | 47.4% | 44.0% | 54 |

Grafik akurasi selama *training* dapat dilihat pada Gambar 5.25. Grafik ini diukur dari dataset validasi selama *training* berlangsung. Mulai dari *epoch* kedelapan model *enhanced_green* secara konsisten memberikan margin akurasi lebih tinggi jika dibandingkan dengan model lainnya.



Gambar 5.25 Grafik nilai akurasi training classifier

5.4.3.2 Fine Tuning

Setelah *training* bagian *classifier* selesai, model dilanjutkan dengan *fine tuning* sebanyak dua blok Inception. Hasil *fine tuning* dari setiap model dapat dilihat pada Tabel 5.7. Nilai ini diukur dari dataset validasi setelah *training* sebanyak 50 *epoch* selesai dan merupakan hasil rata-rata dari 5-fold cross validation. Pada tabel tersebut tampak bahwa model dengan *preprocessing enhanced green* memberikan hasil yang paling baik dengan nilai: akurasi 78.79%, *precision* 82.97%, *recall* 74.64%, dan AUC 95.81%. Model *green_0g0* menempati posisi kedua dengan nilai akurasi 77.82% yang diikuti model *rescale_png* di posisi ketiga dengan nilai akurasi

77.27%. Sedangkan model *ramasubramanian*, *ben_graham* dan *nakhon* menempati posisi keempat, kelima, dan keenam. Dari keenam model ini dapat dilihat bahwa *preprocessing enhanced green* memberikan *benefit* terhadap performa model seperti yang ditunjukkan oleh model *enhanced green* dan model *green_0g0*.

Tabel 5.7 Metrik hasil *fine tuning*

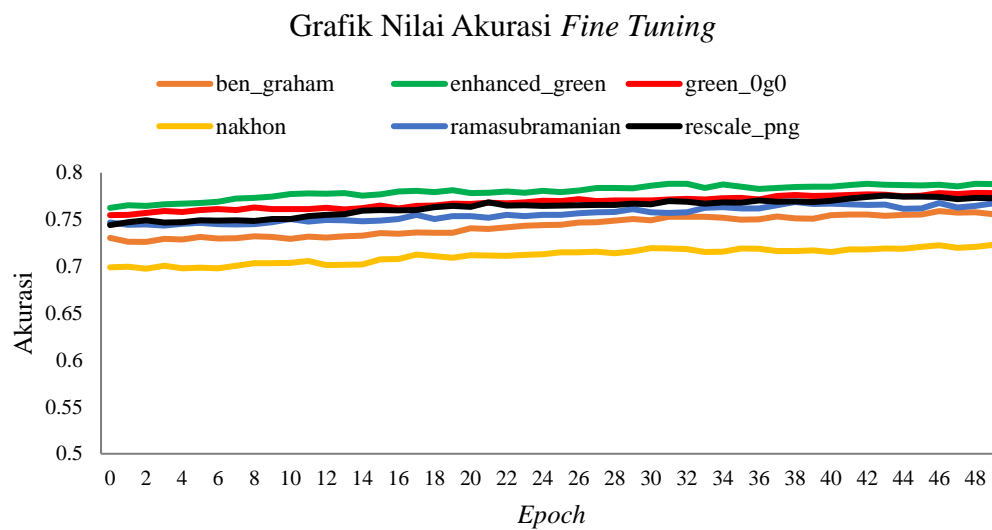
| Model | Loss | Acc | Precision | Recall | AUC |
|------------------------|--------|---------------|-----------|--------|--------|
| <i>ben_graham</i> | 1.9764 | 75.59% | 80.35% | 70.04% | 94.97% |
| <i>enhanced_green</i> | 1.8280 | 78.79% | 82.97% | 74.64% | 95.81% |
| <i>green_0g0</i> | 1.8074 | 77.82% | 81.71% | 73.41% | 95.59% |
| <i>nakhon</i> | 1.8945 | 72.30% | 79.56% | 65.72% | 94.25% |
| <i>ramasubramanian</i> | 1.8076 | 76.73% | 81.17% | 72.44% | 95.62% |
| <i>rescale_png</i> | 1.8294 | 77.27% | 80.79% | 72.41% | 95.32% |

F1 score untuk setiap kelas DR dari lima model ini dapat dilihat pada Tabel 5.8. Pada tabel tersebut kelas 0 memberikan nilai paling tinggi. Hal ini dikarenakan kelas 0 memiliki jumlah citra paling banyak (kolom *support*). Sedangkan kelas 3 memberikan nilai paling kecil karena jumlah citra kelas 3 paling sedikit meskipun pembobotan kelas selama *training* sudah diberikan. Dari tabel ini juga dapat dilihat bahwa F1 score tertinggi dari setiap kelas DR diberikan oleh model dengan *preprocessing* yang berbeda. DR kelas 0, 1, dan 3 diberikan oleh model *ramasubramanian*. DR kelas 2 dan 4 diberikan oleh model *enhanced_green*.

Tabel 5.8 F1 Score *fine tuning*

| DR Level | <i>rescale_png</i> | <i>grah am</i> | <i>nakh on</i> | <i>ramasubram anian</i> | <i>enhanced_g reen</i> | <i>green_ 0g0</i> | <i>Supp ort</i> |
|----------|--------------------|----------------|----------------|-------------------------|------------------------|-------------------|-----------------|
| 0 | 96.2% | 96.6% | 96.2% | 97.4% | 96.4% | 96.2% | 359 |
| 1 | 55.2% | 53.0% | 51.0% | 55.4% | 52.2% | 53.2% | 68 |
| 2 | 67.2% | 60.6% | 55.0% | 62.4% | 68.8% | 66.2% | 184 |
| 3 | 41.4% | 36.6% | 35.6% | 42.0% | 41.4% | 41.4% | 35 |
| 4 | 43.4% | 44.8% | 33.4% | 44.8% | 52.0% | 50.4% | 54 |

Grafik akurasi selama *training* dapat dilihat pada Gambar 5.26. Grafik ini diukur dari dataset validasi setelah *fine tuning* sebanyak 50 *epoch* selesai. Model *enhanced_green* secara konsisten mulai dari epoch pertama sampai ke 50 memberikan margin akurasi lebih tinggi dibandingkan dengan model lainnya yang diikuti model *rescale_png*, *ramasubramanian*, *ben_graham*, dan *nakhon*.



Gambar 5.26 Grafik nilai akurasi fine tuning

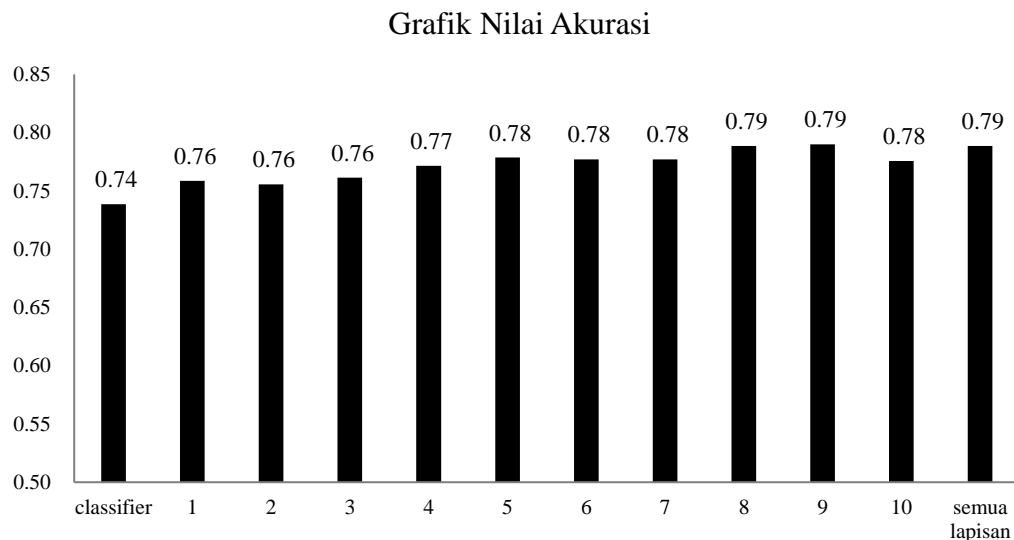
5.4.4 Fine Tuning n block inception

Training pada bagian ini menggunakan dataset asli yang diubah ukurannya saja seperti yang dijelaskan pada sub-bab 4.4.2 dan *hyperparameter* ataupun tahapan *training* bisa dilihat pada sub-bab 4.6.4. Percobaan ini memiliki tujuan untuk membuktikan secara empiris banyaknya blok Inception v3 yang perlu dilakukan *fine tuning* pada model yang ditraining dengan pendekatan *transfer learning*, khususnya untuk deteksi DR pada dataset APTOS 2019. Hasil *training* n -blok Inception v3 bisa dilihat pada Tabel 5.9. Nilai pada tabel tersebut diukur pada dataset validasi setelah *training* sebanyak 50 *epoch* selesai.

Tabel 5.9 Performa Model yang di-training sebanyak n -blok Inception v3

| <i>Block</i> | <i>Loss</i> | <i>Acc</i> | <i>Precision</i> | <i>Recall</i> | <i>AUC</i> |
|-------------------|-------------|---------------|------------------|---------------|------------|
| <i>classifier</i> | 1.9498 | 73.86% | 80.78% | 67.86% | 94.21% |
| 1 | 1.8637 | 75.86% | 79.84% | 69.57% | 94.73% |
| 2 | 1.8552 | 75.57% | 79.62% | 72.00% | 94.97% |
| 3 | 1.8412 | 76.14% | 79.68% | 71.71% | 95.16% |
| 4 | 1.8365 | 77.14% | 81.79% | 73.14% | 95.24% |
| 5 | 1.8227 | 77.86% | 81.79% | 74.43% | 95.41% |
| 6 | 1.8174 | 77.71% | 81.36% | 73.57% | 95.53% |
| 7 | 1.8051 | 77.71% | 81.49% | 74.86% | 95.64% |
| 8 | 1.8095 | 78.86% | 81.64% | 75.57% | 95.54% |
| 9 | 1.8149 | 79.00% | 81.41% | 74.43% | 95.59% |
| 10 | 1.8239 | 77.57% | 81.40% | 75.00% | 95.49% |
| Semua lapisan | 1.8081 | 78.86% | 81.85% | 76.00% | 95.74% |

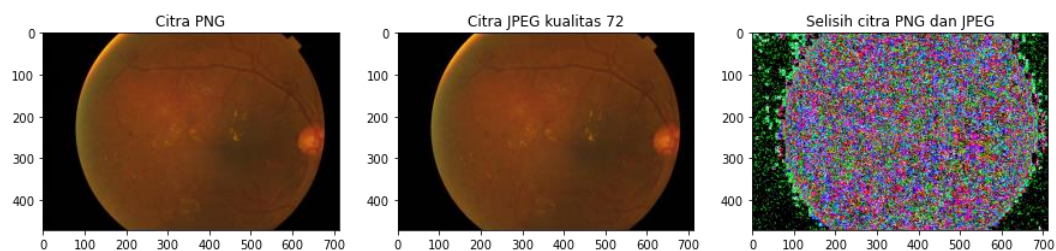
Gambar 5.27 dan adalah visualisasi Tabel 5.9 dalam bentuk grafik batang untuk nilai akurasi model. Jika mengacu pada grafik ini, *fine tuning* terbaik dilakukan sebanyak 9 blok Inception. Tampak pada grafik juga *fine tune* dari lapisan manapun memberikan *benefit* terhadap performa model dengan rentang kenaikan akurasi sebesar 1.71% - 5.14%.



Gambar 5.27 Akurasi model dengan *fine tuning* n-blok Inception v3

5.4.5 Komparasi PNG dan JPEG

Pada bagian ini membandingkan model yang di-*training* dengan file format PNG dan JPEG kualitas 72 untuk melihat pengaruhnya terhadap performa model yang diberikan. *Training* model ini merupakan tahap awal ketika mau menggabungkan dataset yang berbeda file formatnya. Misalkan jika dataset APTOS 2019 mau ditambahkan dengan dataset EYEPACS 2015 agar dataset berukuran lebih besar. Dataset APTOS 2019 memiliki file format PNG, sedangkan dataset EYEPACS 2015 memiliki file format JPEG.



Gambar 5.28 Artifak pada citra JPEG akibat kompresi

Training model PNG dan JPEG dilakukan dengan pendekatan *transfer learning*. Mula-mula model di-*training* bagian *classifier* yang dilanjutkan dengan *fine tuning* dua blok Inception v3. Hyperparameter mengikuti nilai pada Tabel 4.3 dan Tabel 4.7. Model di-*training* dengan *5-fold cross validation* sebanyak 50 *epoch*.

Berdasarkan Tabel 5.10 dan Tabel 5.11 dapat dilihat bahwa baik ketika *training* bagian *classifier* maupun *fine tuning*, model dengan dataset file format JPEG memberikan hasil yang lebih baik. Hal ini disebabkan karena model Inception v3 sebelumnya di-*training* menggunakan dataset dari ImageNet yang file formatnya adalah JPEG [49]. Model *pre-trained* sudah di-*training* untuk memetakan citra yang memiliki efek kompresi JPEG seperti pada Gambar 5.28 kanan.

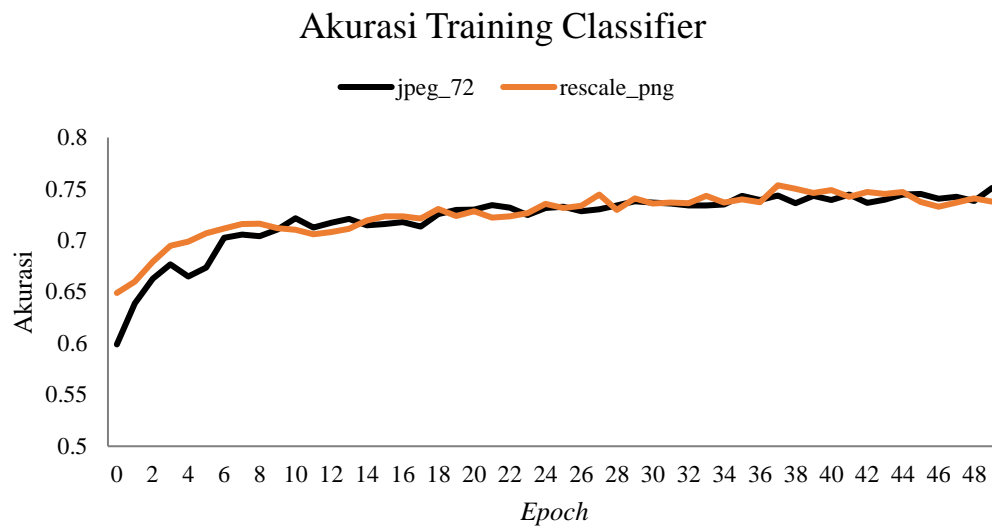
Tabel 5.10 Hasil evaluasi model PNG & JPEG setelah *training classifier*

| Model | Loss | Acc | Precision | Recall | AUC |
|--------------------|--------|---------------|-----------|--------|--------|
| <i>rescale_png</i> | 1.9378 | 73.78% | 80.70% | 66.81% | 94.53% |
| <i>jpeg_72</i> | 1.9232 | 75.10% | 81.08% | 68.58% | 94.92% |

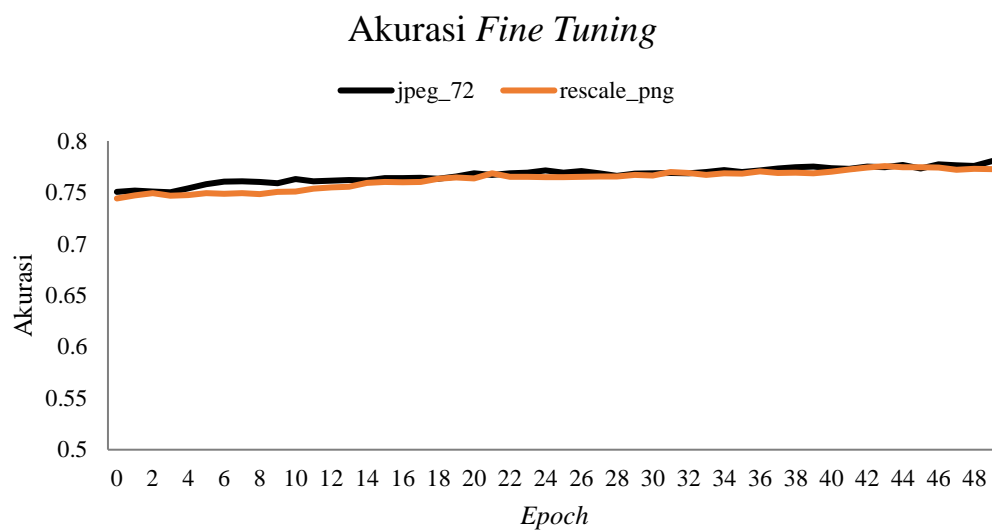
Tabel 5.11 Hasil evaluasi model PNG & JPEG setelah *fine tuning* dua block Inception

| Model | Loss | Acc | Precision | Recall | AUC |
|--------------------|--------|---------------|-----------|--------|--------|
| <i>rescale_png</i> | 1.8294 | 77.27% | 80.79% | 72.41% | 95.32% |
| <i>jpeg_72</i> | 1.8113 | 78.05% | 81.91% | 73.27% | 95.70% |

Grafik akurasi selama *training classifier* dapat dilihat pada Gambar 5.29 dan grafik akurasi selama *fine tuning* dapat dilihat pada Gambar 5.30. Tampak bahwa model *jpeg_72* memberikan hasil akurasi yang lebih baik jika dibandingkan dengan model *rescale_png*



Gambar 5.29 Grafik akurasi *training classifier* untuk file format yang berbeda



Gambar 5.30 Grafik akurasi fine tuning untuk file format yang berbeda