BAB III

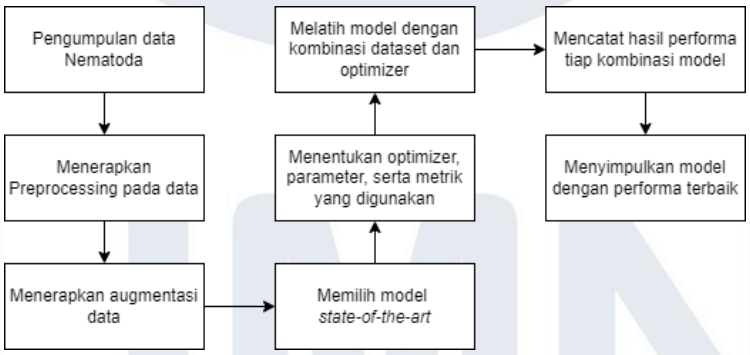
ANALISIS DAN PERANCANGAN SISTEM

3.1 Metode Penelitian

3.1.1. Studi Literatur

Pada tahap ini, dilakukan studi literatur mengenai hasil penelitian yang berkaitan dengan topik yang diteliti. Studi literatur dilakukan dengan membaca karya ilmiah terkait yang ada di internet, serta melakukan diskusi dengan salah satu dosen Program Studi Teknik Komputer. Selain itu panduan teknis seperti artikel pada website "*Medium"* juga berguna untuk mempelajari cara implementasi sistem yang akan digunakan. Referensi lainnya yang digunakan dalam penelitian ini adalah paper yang berkaitan dengan topik hama di bidang agrikultur, karya ilmiah mengenai model yang akan digunakan, serta dokumentasi teknis terhadap model yang akan digunakan.

3.1.2. **Pengumpulan, Pemrosesan, dan Augmentasi Data**

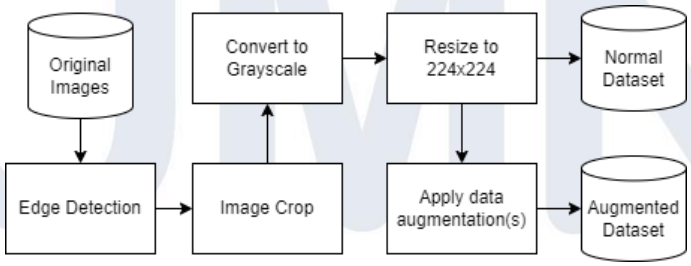


Gambar 3.1 Diagram alur pelaksanaan penelitian

Berikut merupakan gambaran umum penelitian yang akan dilakukan. Pertama, proses pengumpulan data dilakukan dengan mencari penelitian serupa yang menyediakan dataset secara publik. Salah satu penelitian yang menyediakan dataset Nematoda adalah penelitian oleh Xue Qing [9]. Dataset I-Nema ini menyediakan taksa Nematoda dengan spesies *Acrobeles sp., Acrobeloides sp., Amplimerlinius sp., Aphelenchoides sp., Aporcelaimus sp., Axonchium sp., Discolimus sp., Ditylenchus sp., Dorylaimus sp., Eudorylaimus sp., Helicotylenchus sp.,*

Mesodorylaimus sp., Miconchus sp., Mylonchulus sp., Panagrolaimus sp., Pratylenchus sp., Pristionchus sp., Rhbiditis sp., dan *Xenocriconema sp.*

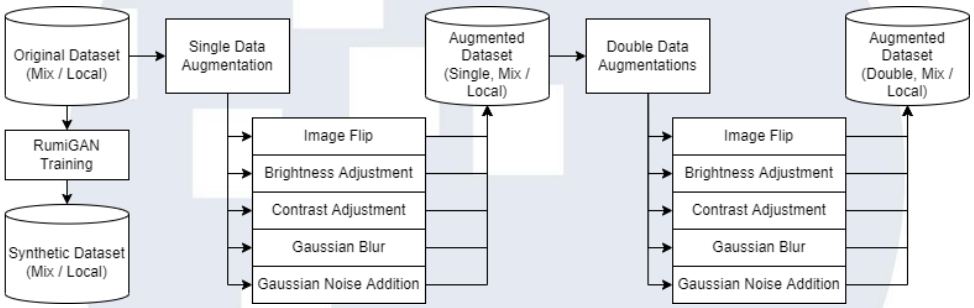
Untuk mendapatkan dataset Nematoda dengan jenis yang sering ditemukan di Indonesia, diperlukan untuk mengumpulkan data dari tanah langsung. Dataset didapatkan dari kolaborasi dengan Bagian Nematologi, Lab. Ilmu Hama Tumbuhan, Departemen Proteksi Pertanian, Fakultas Pertanian Universitas Gajah Mada (UGM). Metode pemrosesan data yang dilakukan akan serupa dengan penelitian yang telah dilakukan sebelumnya [9]. Sampel Nematoda dipisahkan dan dilakukan klasifikaksi secara manual terlebih dahulu. Klasifikasi dilakukan berdasarkan karakteristik morfologi dari genus tersebut. Kemudian diambil gambar dan diproses lebih lanjut agar dapat digunakan dalam pelatihan model. Letak dari specimen dan tingkat kecerahan tiap gambar berbeda-beda. Tiap gambar dilakukan segmentasi menggunakan *edge detection,* kemudian hasil deteksi digunakan untuk menentukan daerah kosong pada gambar dan gambar dipotong. Hal ini dilakukan untuk mengurangi bagian gambar yang tidak memiliki informasi penting. Gambar lalu diubah warnanya menjadi *grayscale,* karena identifikasi dilakukan hanya berdasarkan morfologi dari tiap specimen. Genus dari familia yang didapatkan adalah *Criconema, Criconemoides, Helicotylenchus, Hemicycliophora, Hirschmanniella, Hoplolaimus, Meloidogyne, Pratylenchus, Radopholus, Trichodorus,* dan *Xiphinema.*



Gambar 3.2 Diagram alur pemrosesan data

Data yang didapat kemudian dibentuk menjadi dua dataset, yaitu dataset lokal yang berasal dari Bagian Nematologi UGM, dan dataset campuran yang berupa kombinasi dataset lokal dan dataset I-Nema. Dataset lokal digunakan untuk melatih model pada jenis Nematoda yang sering ditemukan di Indonesia saja, sedangkan dataset campuran digunakan untuk menguji performa model secara umum terhadap

semua jenis Nematoda. Kemudian resolusi semua gambar diubah menjadi 224 x 224 agar dapat digunakan oleh semua model. Lalu dilakukan proses augmentasi data secara *on the fly* (*online augmentation*) pada dataset untuk meningkatkan variasi pada dataset dan mencegah *overfitting* ketika melakukan proses pelatihan dengan melakukan berbagai operasi transformasi gambar.



Gambar 3.3 Diagram augmentasi data pada tiap dataset

Transformasi gambar yang dilakukan adalah proses membalik gambar (secara *vertical* dan *horizontal*), penambahan *noise*, pengaburan gambar, pengubahan kecerahan, dan pengubahan kontras. Jenis transformasi gambar ini umum digunakan dalam meningkatkan variasi dataset pada permasalahan *deep learning*. Augmentasi lain seperti translasi atau transformasi afin tidak dilakukan karena beberapa hasil augmentasi akan menutup fitur penting (misal, kepala atau ekor dari spesimen) yang dapat menurunkan akurasi model, serta transformasi rotasi akan menghasilkan hal serupa dengan proses membalik gambar. Berikut implementasi yang digunakan untuk augmentasi dataset.

- Pembalikan gambar dilakukan secara acak pada tiap data dalam dataset, dan tipe pembalikan juga dipilih secara acak, dengan kemungkinan tiap operasi sebesar 50%. Rata-rata distribusi data sekitar 25% gambar tidak ada pembalikan, 25% *horizontal flip*, 25% *vertical flip*, dan 25% gambar dengan operasi keduanya.
- Penambahan *noise* pada gambar berupa penambahan *white gaussian noise* pada gambar dengan nilai *mean* sebesar 0 dan standar deviasi sebesar 0.15. Dilakukan secara acak dengan kemungkinan sebesar 50% pada tiap data.

- Proses pengaburan gambar dilakukan dengan menggunakan *gaussian filter* dengan ukuran kernel sebesar 3x3 dengan standar deviasi sebesar 1. Dilakukan secara acak dengan kemungkinan sebesar 50% pada tiap data.
- Pengubahan kecerahan berupa peningkatan kecerahan gambar secara acak, peningkatan pencerahan minimum sebesar 0 dan maksimum sebesar 0.3. Operasi dilakukan pada semua data pada dataset.
- Pengubahan kontras berupa peningkatan kontras secara acak, dengan faktor minimum sebesar 0.3 dan maksimum sebesar 3. Operasi dilakukan pada semua data.

Proses augmentasi dilanjutkan dengan kombinasi dua (2) metode augmentasi, seperti peningkatan kecerahan dan kontras. Pilihan kombinasi augmentasi ditentukan dari lima (5) metode augmentasi sebelumnya. Bergantung pada performa model terhadap suatu metode augmentasi, metode tersebut dapat dikeluarkan dari pilihan dari kombinasi.

Proses augmentasi lain yang diuji adalah penambahan gambar hasil sintesis. Sintesis gambar akan dilakukan menggunakan model GAN dengan *RumiGAN framework*. Model GAN yang akan digunakan adalah variasi LSGAN dengan pertimbangan kualitas gambar yang dihasilkan dan proses pembelajaran model yang lebih stabil dibanding variasi lainnya [24]. Model ini akan dilatih terhadap satu per satu genus / familia, tidak semua jenis secara langsung agar tidak menghasilkan gambar Nematoda yang tidak dapat di identifikasi. Model GAN akan dilatih dengan memberikan satu kelas genus Nematoda sebagai dataset positif dan genus lainnya sebagai dataset negatif. Pelatihan model akan dilakukan sampai hasil Generator menghasilkan gambar yang dapat diidentifikasi sebagai Nematoda atau Generator gagal konvergen. Proses augmentasi data dengan sintesis diharapkan memberikan variasi gambar yang akan mencegah *overfitting* dan dapat memperluas data Nematoda yang tersedia.

Dalam melatih model, dataset yang akan digunakan dibagi menjadi tiga bagian, yaitu dataset pelatihan, dataset validasi, dan dataset tes, dengan rasio 80:10:10.

Dataset campuran memiliki 26 kelas klasifikasi dengan total 3595 gambar Nematoda, dengan pembagian menjadi 2876 data untuk pelatihan, 360 data untuk validasi, dan 359 data untuk pengujian dan perbandingan performa antar model. Dataset lokal memiliki 11 kelas klasifikasi dengan total 957 data, yang dibagi menjadi 766, 96, dan 95 data untuk pelatihan, validasi, dan pengujian. Untuk augmentasi data, augmentasi data hanya diterapkan pada dataset pelatihan.

Tabel 3.1 Distribusi Genus Nematoda pada dataset campuran

Mix Dataset Genus	**# Of Samples**
Genus Acrobeles	71
Genus Acrobeloides	184
Genus Amplimerlinius	24
Genus Aphelenchoides	347
Genus Aporcelaimus	128
Genus Axonchium	170
Genus Criconema	3
Genus Criconemoides	103
Genus Ditylenchus	330
Genus Dorylaimus	38
Genus Eudorylaimus	86
Genus Helicotylenchus	213
Genus Hemicycliophora	6
Genus Hirschmanniella	130
Genus Hoplolaimus	151
Genus Meloidogyne	211
Genus Mesodorylaimus	96
Genus Miconchus	57
Genus Mylonchulus	139
Genus Panagrolaimus	327
Genus Pratylenchus	402
Genus Pristionchus	196
Genus Radopholus	12
Genus Rhabditis	81
Genus Trichodorus	30
Genus Xiphinema	60
Total	**3595**

Tabel 3.2 Distribusi Genus Nematoda pada dataset lokal

Local Dataset Genus	# Of Samples
Genus Criconema	3
Genus Criconemoides	103
Genus Helicotylenchus	135
Genus Hemicycliophora	6
Genus Hirschmanniella	130
Genus Hoplolaimus	151
Genus Meloidogyne	211
Genus Pratylenchus	116
Genus Radopholus	12
Genus Trichodorus	30
Genus Xiphinema	60
Total	**957**

3.1.3. Perancangan Sistem Klasifikasi

Proses klasifikasi taksa Nematoda akan dilakukan dengan beberapa model *state-of-the-art* yang memiliki performa tinggi untuk masalah klasifikasi gambar. Salah satu tolok ukur performa klasifikasi gambar adalah akurasi model pada dataset ImageNet [28]. Pemilihan model yang akan diuji dipilih berdasarkan tolok ukur ini. Penerapan model akan menggunakan *Keras* [29] dan *Tensorflow Library* yang sudah menyediakan implementasi dari berbagai macam model *machine learning*. Model CNN yang akan digunakan untuk klasifikasi adalah ResNet [12] dan EfficientNet [14]. Selain itu, model lain yang akan digunakan adalah CoAtNet [19].

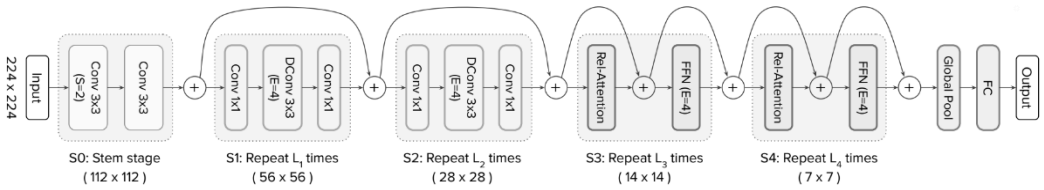
Model ResNet dipilih berdasarkan performa dalam klasifikasi Nematoda, yang diteliti oleh Xuequan Lu pada tahun 2021 [9]. Model ResNet yang akan digunakan adalah model ResNet101V2. Kemudian, model CoAtNet dipilih berdasarkan performanya dalam klasifikasi dataset ImageNet dan berupa model *hybrid*, gabungan dari arsitektur CNN dan *Transformer* [19]. Variasi CoAtNet yang akan digunakan adalah CoAtNet-0. Model EfficientNet dipilih berdasarkan performa yang tinggi dengan jumlah parameter yang kecil dibanding model-model lainnya [14][28]. Variasi EfficientNet yang akan digunakan adalah EfficientNetV2M, varian yang ukurannya relatif besar dan memiliki performa tinggi. Variasi dari tiap keluarga model dipilih dengan pertimbangan spesifikasi *hardware* yang digunakan dan ukuran dari dataset, karena variasi model dengan parameter yang lebih banyak

akan cenderung lebih mudah *overfitting* ketika dilatih pada dataset dengan ukuran yang relatif kecil.

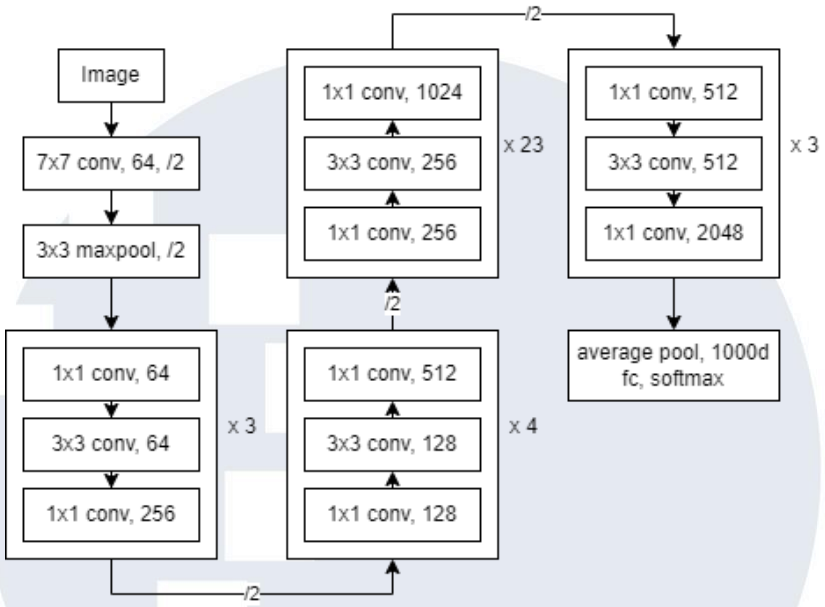
Tabel 3.3 Perbandingan jumlah parameter pada model yang digunakan dengan layer klasifikasi. Sumber: Keras Library

Model Architecture	**Trainable Parameters (approx.)**
ResNet101v2	42,576,023
CoAtNet-0	22,528,257
EfficientNetV2M	52,887,819

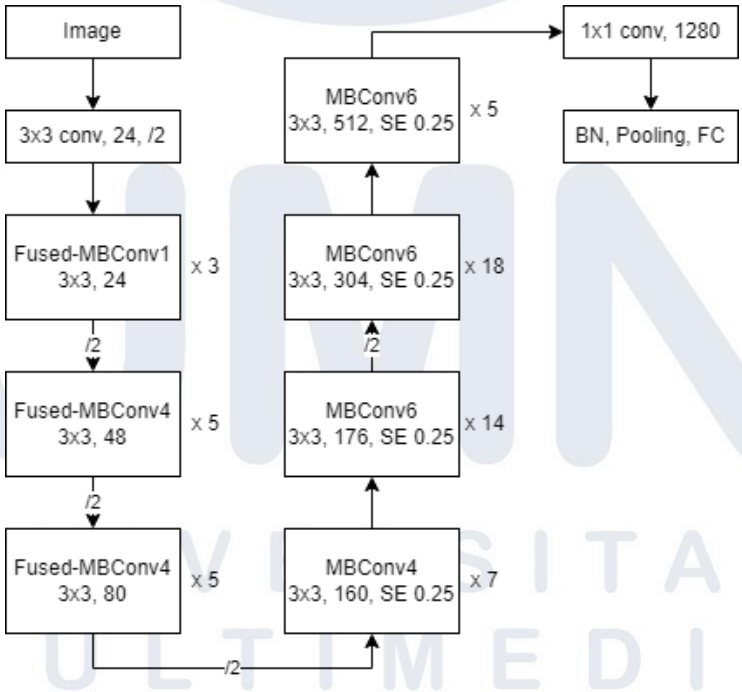
Struktur tiap model yang dipilih dapat dilihat pada Gambar 3.4, Gambar 3.5, dan Gambar 3.6. Dalam melakukan klasifikasi, tiap model ditambahkan *dense* layer dengan aktivasi *softmax*, dengan jumlah kelas menyesuaikan dengan dataset yang digunakan (26 untuk dataset campuran dan 11 untuk dataset lokal). Struktur usulan klasifikasi model dengan *fully connected layer* digambarkan pada Gambar 3.7.



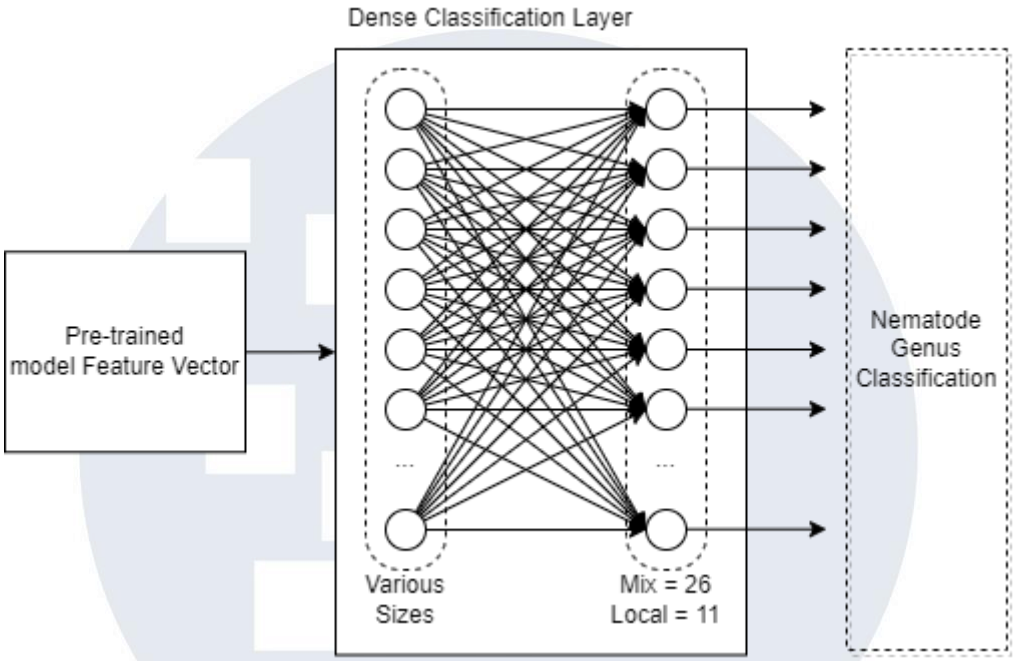
Gambar 3.4 Diagram arsitektur model CoAtNet



Gambar 3.5 Diagram arsitektur model ResNet101



Gambar 3.6 Diagram arsitektur model EfficientNetV2M



Gambar 3.7 Diagram layer klasifikasi untuk tiap model yang diuji

Dalam melatih model, digunakan tiga *optimizer* untuk mencari tahu *optimizer* terbaik untuk tiap model dan apakah arsitektur atau variasi dataset mempengaruhi penggunaan *optimizer* yang ada. Hal ini juga mencegah ketidakadilan ketika kombinasi model dan dataset tepat dengan *hyperparameter* yang digunakan, akan menghasilkan performa yang tinggi. *Optimizer* yang akan digunakan adalah *Adam*, *SGD*, dan *RMSprop*. *Adam* dipilih karena performanya yang terbaik dibandingkan dengan *optimizer* lainnya dan berupa *adaptive optimizer* yang memiliki momentum, kemudian *SGD* digunakan untuk merepresentasikan *gradient descent optimizer* dan SGD sering digunakan untuk mendapatkan performa *state-of-the-art* pada penelitian mengenai model *deep learning*. *RMSprop* digunakan untuk perbandingan ketika menggunakan *adaptive optimizer* tanpa momentum.

Implementasi yang tersedia untuk tiap model adalah model yang sudah dilatih (*pretrained*) dengan menggunakan dataset ImageNet. Model kemudian diberikan layer klasifikasi dengan jumlah kelas sesuai dengan dataset yang dilatih (dataset campuran atau dataset lokal). Untuk menjaga konsistensi, tiap model akan dilatih dengan *hyperparameter* yang sama, yaitu layer klasifikasi dengan aktivasi *softmax*, ukuran tiap *batch* sebesar 32, parameter *optimizer* yang sama (*learning rate* 0.001

untuk Adam dan RMSprop, serta 0.01 untuk SGD; *momentum* SGD dan RMSprop 0; nilai beta 0.9 – 0.999 untuk Adam), serta *loss function* berupa *sparse cross-entropy* untuk klasifikasi *multi-label*. *Epoch* pelatihan bergantung pada dataset yang digunakan, yaitu 60 *epoch* untuk pelatihan pada dataset campuran dan 100 *epoch* untuk dataset lokal. Ukuran *input* dari tiap model disamakan ke ukuran 224x224x3.

3.1.4. Metrik Penilaian

Terdapat beberapa metrik penilaian yang akan digunakan untuk menilai performa dari tiap model. Dua metrik utama yang akan digunakan adalah metrik yang digunakan oleh penelitian terdahulu [9], yaitu metrik *Overall Accuracy* dan *Mean Class Accuracy*. Metrik *Overall Accuracy* digunakan untuk mengevaluasi akurasi terhadap seluruh gambar pada dataset test. Metrik *Mean Class Accuracy* digunakan untuk mengevaluasi rata-rata dari akurasi klasifikasi tiap genus pada dataset, menjadi indikasi untuk menilai apakah model dapat mempelajari karakteristik morfologi dari tiap kelas lebih baik atau tidak dibanding model lainnya. Metrik lain yang akan digunakan adalah *F1-Score*, serta rata-rata dari *Precision* dan *Recall*. Metrik *F1-Score* adalah metrik penilaian klasifikasi berdasarkan rata-rata harmonik dari *precision* dan *recall*, cocok digunakan ketika ukuran distribusi data dalam kelas dataset tidak seimbang. Metode rata-rata F1-Score yang digunakan adalah *weighted average* untuk mendapatkan akurasi pada dataset tidak seimbang. Rata-rata dari *precision* dan *recall*, serta nilai F1-Score dicatat untuk menjaga kelengkapan data.

3.2 Implementasi Sistem

Implementasi sistem pelatihan dan inferensi model menggunakan bahasa pemrograman Python dengan menggunakan library *Keras* [29] dan *TensorFlow* dalam menerapkan model *deep learning*. Implementasi dari tiap model akan mengikuti dokumentasi yang sudah disediakan, kemudian dilakukan *fine-tuning* model menggunakan *training parameter* yang sudah dijabarkan di bab sebelumnya. *Platform* yang akan digunakan adalah *Google Colab Pro*, baik dalam proses sintesis gambar maupun pelatihan model dan inferensi klasifikasi taksa Nematoda.

Hardware yang akan digunakan adalah spesifikasi hardware yang dapat disediakan pada *platform Google Colab Pro* dengan spesifikasi minimum GPU NVIDIA P100 atau NVIDIA T4, RAM ~25GB, dan CPU Xeon Processor@2.3GHz. Detil spesifikasi lebih lanjut tidak diberikan oleh pihak Google Colab. Hardware tersebut dinilai cukup untuk mengeksekusi pelatihan model dan inferensi sintesis motif dengan waktu yang wajar. Biaya langganan Google Colab Pro per bulan adalah $9.99.

Implementasi sistem pertama yang dilakukan adalah melakukan instalasi komponen dan *library* yang akan digunakan pada *notebook*. Lalu, melakukan *pre-processing* data sesuai dengan proses yang sudah dijelaskan di bab sebelumnya. Kemudian menyiapkan dataset sesuai dengan dokumentasi pada tiap model agar dapat dibaca oleh model tersebut, serta menerapkan augmentasi data pada dataset secara acak. Kemudian model dilatih dengan *hyperparameter* tertentu dan hasil proses pelatihan (akurasi pelatihan dan validasi, serta *training loss* dan *validation loss*) di plot, serta berdasarkan dataset test dihitung metrik *overall accuracy, mean class accuracy, f1-score, precision,* dan *recall*. Hasil kemudian dicatat dalam suatu lembar kerja dan dibandingkan performanya untuk tiap model dengan kombinasi *optimizer* dan dataset yang digunakan.