

BAB IV. ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem

Analisis sistem merujuk pada proses untuk menentukan kebutuhan sistem dan perangkat yang dibutuhkan dalam pengembangan sebuah sistem. Hal ini melibatkan analisis terhadap komponen sistem, interaksi antara komponen-komponen tersebut, serta dampak sistem terhadap pengguna atau lingkungan sekitarnya. Analisis sistem berfungsi untuk memahami secara menyeluruh sistem yang sedang diteliti, mengevaluasi kinerja dan keefektifan sistem, serta mengidentifikasi kelemahan dan potensi perbaikan yang dapat dilakukan.

4.1.1 Perangkat Keras

Perancangan sistem perangkat keras untuk mendukung penelitian ini dapat dilihat pada tabel 4.1 di bawah ini.

Tabel 4.1 Spesifikasi Kebutuhan Perangkat Keras

No.	Nama Perangkat Keras	Spesifikasi
1.	Laptop	<ul style="list-style-type: none"> • ASUS A407UF • Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz • 8,00 GB (7,88 GB usable) RAM
2.	OpenMV Cam H7+	<ul style="list-style-type: none"> • ARM® 32-bit Cortex®-M7 CPU w/ Double Precision FPU 480 MHz (1027 DMIPS) Core Mark Score: 2400, • Memori 64KB Stack • 256KB .DATA/.BSS/Heap • 32MB Frame Buffer/Stack • 512KB SDRAM Cache • 256KB DMA Buffers

4.1.2 Perangkat Lunak

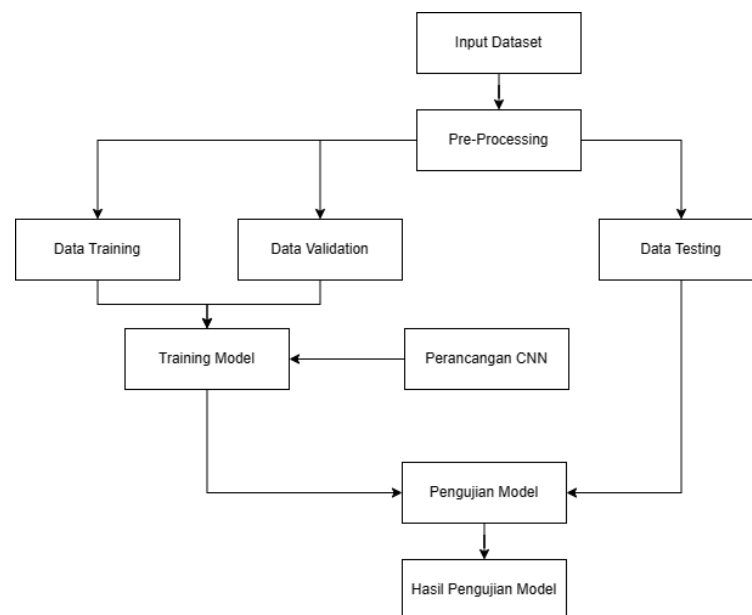
Perancangan sistem perangkat lunak untuk mendukung penelitian ini dapat dilihat pada tabel 4.2 di bawah ini.

Tabel 4.2 Spesifikasi Kebutuhan Perangkat Keras

No.	Nama Perangkat Lunak	Deskripsi
1.	OpenMV IDE	Lingkungan pengembangan terintegrasi utama untuk digunakan dengan OpenMV Cam. OpenMV IDE berfungsi sebagai <i>source code editor</i> , <i>debugger</i> , dan integrasi kamera serta sensor dengan OpenMV Cam.
2.	Python (3.8 Version)	Bahasa pemrograman yang digunakan untuk mendukung penelitian sistem klasifikasi bibit ikan lele ini.

4.1.3 Gambaran Umum Sistem

Alur sistem adalah sebuah diagram yang digunakan untuk menggambarkan bagaimana sebuah sistem atau proses beroperasi. Alur sistem menunjukkan bagaimana input diterima, bagaimana input diolah, dan bagaimana output dihasilkan. Alur sistem dapat dilihat pada gambar 4.1 di bawah ini dari fase *inputing* sampai fase *testing* untuk model *Convolutional Neural Network (CNN)*.



Gambar 4.1 Alur Sistem

Input data pada klasifikasi ukuran bibit lele menggunakan metode CNN dengan OpenMV Cam adalah *frames* bibit ikan lele yang diambil dengan kamera OpenMV Cam. *Frames* tersebut akan diterima oleh OpenMV Cam dan diteruskan pada tahap *pre-processing*. *Pre-processing* diawali dengan melakukan pelabelan

objek bibit ikan lele pada tiap *frames* dari video yang diperoleh. Pelabelan pada tiap *frames* untuk pengenalan ukuran bibit ikan lele berupa panjang bibit ikan lele dari *Grade A* (1-3 cm), *Grade B* (4-5 cm), dan *Grade C* (>6 cm). Selanjutnya kumpulan *frames* tersebut diproses dan dimanipulasi data gambarnya. Hal tersebut dilakukan agar dapat melakukan augmentasi data gambar pada saat proses *training* nantinya. Augmentasi data gambar dapat mencakup teknik seperti *rescaling*, *pergeseran*, rotasi, *zoom*, dan lain-lain.

Langkah selanjutnya yaitu dilakukan pembagian data yang telah dikumpulkan (*dataset*) menjadi tiga (3) bagian, yaitu data latih (*data training*), data validasi (*data validation*) dan data pengujian (*data testing*). Ketiga bagian inilah yang digunakan untuk membangun model CNN untuk sistem klasifikasi bibit ikan lele.

Perancangan CNN merupakan tahapan dalam menyusun sebuah model yang digunakan untuk melatih data dalam mengenali objek yang diinginkan. Model yang disusun terdiri dari jumlah *layers* yang digunakan, penentuan *filter*, penentuan ukuran *kernel*, dan penentuan fungsi aktivasi dan ukuran *pooling*. Setelah dilakukan perancangan model CNN, tahapan selanjutnya adalah melakukan pelatihan (*training*) model dengan menggunakan *data training* dan *data validation* yang telah dibuat sebelumnya. Selanjutnya, pada proses pengujian model digunakan jumlah *epoch* (iterasi) untuk menentukan berapa kali jaringan (*neuron*) akan melakukan pelatihan. Pada tahap ini terdapat fungsi *loss* yang digunakan untuk melihat performa dari model CNN.

Tahapan selanjutnya adalah melakukan pengujian model dengan data *data testing*. Tahapan ini dilakukan untuk menguji tingkat akurasi dari model *Convolutional Neural Network* (CNN) yang telah dibuat. Tingkat akurasi ini menunjukkan tingkat kebenaran dari pengklasifikasian jenis ukuran *grade* bibit ikan lele. Semakin tinggi tingkat akurasi berarti menunjukkan bahwa model dapat melakukan pengklasifikasian ketiga jenis *grade* dengan baik. Sedangkan nilai *loss* menunjukkan tingkat kesalahan prediksi model selama pelatihan. Grafik *loss* yang ideal akan menunjukkan penurunan yang stabil dan berkelanjutan seiring dengan waktu.

4.2 Perancangan Sistem

Perancangan sistem merupakan sebuah proses merencanakan dan merancang struktur, komponen, dan fungsi dari suatu sistem yang akan dibangun. Perancangan sistem dilakukan untuk memastikan bahwa sistem yang akan dibangun sesuai dengan kebutuhan dan tujuan yang diinginkan. Pada penelitian ini, melibatkan desain sistem, arsitektur sistem, proses pelatihan (*training*) model dan proses pengujian (*testing*) model.

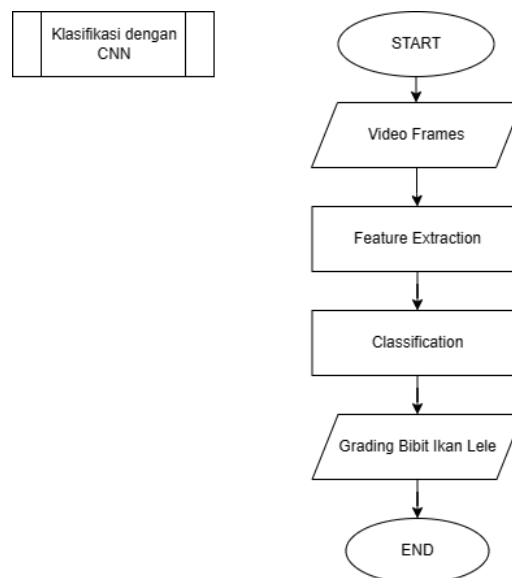
4.2.1 Desain Sistem

Pada penelitian ini, alur dari proses untuk melakukan *grading* bibit ikan lele berdasarkan ukuran dengan metode *Convolutional Neural Network* (CNN) dapat dilihat pada gambar 4.2 dan 4.3 di bawah ini.



Gambar 4.2 Alur Proses *Grading*

Proses awal dengan memasukkan citra video rekaman dari bibit ikan lele. Kemudian dari citra video tersebut, dilakukan pendeteksian dan penangkapan *frame* yang terdapat bibit ikan lele dalam waktu yang ditentukan. Setelah terdeteksi, maka dilakukan klasifikasi dengan CNN. Dengan klasifikasi tersebut, diharapkan dapat mengetahui hasil klasifikasi *grading* bibit ikan lele. Berikut gambar yang menampilkan proses klasifikasi dengan metode CNN.



Gambar 4.3 Submenu Alur Proses *Grading*

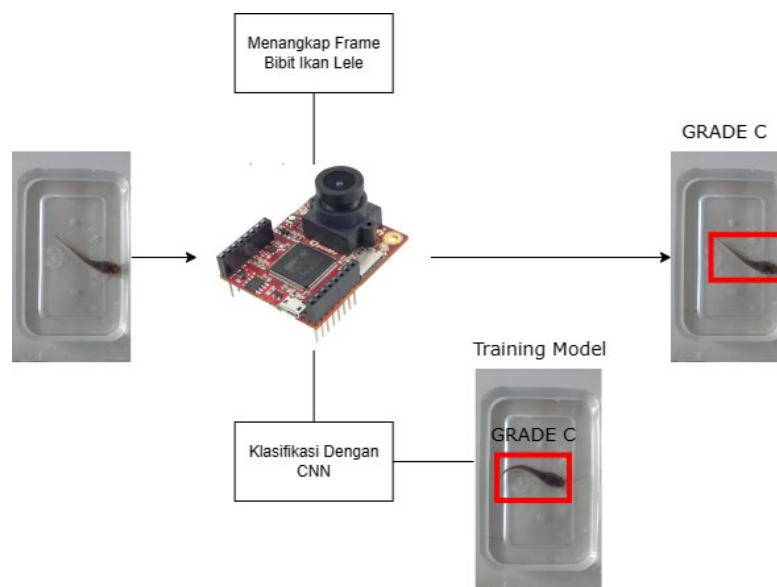
Dalam *sub-menu* Klasifikasi Dengan CNN, tahap awal yaitu mengambil *frame video* yang mendeteksi objek bibit ikan lele. Kemudian pada bagian *feature extraction*, dengan CNN yang terdiri dari beberapa *layers* termasuk *convolutional layers* bertanggung jawab untuk mengekstraksi fitur-fitur visual dari *frame* bibit ikan lele. *Convolution layers* menggunakan *filters* atau kernel-kernel kecil yang bergerak secara *convolution* pada input. Proses *convolution* ini membantu mengenali pola atau fitur khusus dalam *frame*, seperti berupa garis, sudut, panjang, lebar dan lengkungan bibit ikan lele yang ada di dalam *frame*. Selanjutnya, *layer pooling* (seperti MaxPooling) dapat digunakan untuk mereduksi dimensi fitur yang dihasilkan oleh *convolutional layers*, sehingga mengurangi jumlah parameter yang harus dipelajari oleh model.

Setelah melewati *convolutional layers* dan *pooling*, *output* yang dihasilkan akan menjadi representasi fitur yang lebih abstrak dan kaya akan informasi mengenai kriteria *grade* bibit ikan lele tersebut. Representasi ini kemudian digunakan sebagai *input* untuk *layer* berikutnya dalam CNN, seperti *fully connected layer* atau *classifier*, untuk melakukan tugas klasifikasi *grade* bibit ikan lele.

Setelah melalui berbagai proses tersebut, bisa didapatkan hasil prediksi dari model tersebut untuk klasifikasi bibit ikan lele termasuk *grade A*, *grade B*, atau *grade C*.

4.2.2 Arsitektur Sistem

Arsitektur sistem merepresentasi dari struktur komponen yang saling terkait dari sebuah sistem, yang menggambarkan bagaimana komponen-komponen tersebut bekerja bersama untuk mencapai tujuan. Arsitektur sistem juga menunjukkan bagaimana komponen-komponen tersebut terkait dengan lingkungan eksternal sistem.



Gambar 4.4 Arsitektur Sistem

Pada gambar 4.4 di atas menjelaskan rancangan dari sistem yang akan dibangun. OpenMV Cam H7 Plus cam mengambil video sebagai input data. Selanjutnya video dimasukkan akan dilakukan proses pra-pemrosesan yaitu pendeteksian bibit lele. Kemudian frame video bibit lele yang telah didapatkan dilakukan klasifikasi menggunakan model CNN, sehingga menghasilkan prediksi ukuran bibit lele sesuai dengan ukurannya.

4.2.3 Training Model

Dalam penelitian ini, proses pelatihan menggunakan arsitektur MobilenetV2 sebagai model *pre-trained*. Proses pelatihan model CNN menggunakan MobileNetV2 melibatkan beberapa langkah utama, antara lain:

- *Dataset*

Untuk penelitian ini, digunakan 3 bagian yang didapat dari *dataset*, yaitu *data training*, *data validation*, dan *data testing*. Setiap data gambar memiliki warna RGB berformat .bmp. Masing-masing *dataset* blobs.fit() dan

blobs.rect dilakukan *training model* dengan 2 ukuran gambar berbeda, yaitu 96 x 96 pixel dan 128 x 128 pixel. Sehingga, terdapat 2 kali percobaan untuk setiap *dataset* dengan ukuran gambar yang berbeda.

- *Pre-Processing*

Dataset pada penelitian ini diproses sebelum digunakan untuk pelatihan. Proses ini mencakup *resizing* gambar ke ukuran yang disesuaikan oleh model MobileNetV2 (96 x 96 pixel dan 128 x 128 pixel).

- *MobileNetV2 Model Building*

Salah satu fitur utama MobileNetV2 adalah efisiensi komputasinya. Arsitektur ini dirancang untuk mengoptimalkan jumlah parameter dan operasi komputasi yang diperlukan dalam jaringan. Dengan demikian, model MobileNetV2 dapat mendukung operasional OpenMV Cam dengan daya komputasinya yang terbatas. Arsitektur ini memungkinkan model untuk menangkap fitur-fitur penting dari gambar secara bertahap dan mempercepat proses konvolusi.

MobileNetV2 menggunakan konvolusi terbagi (*depthwise separable convolution*) untuk mengurangi beban komputasi. Konvolusi terbagi terdiri dari dua tahap: konvolusi terbagi (*depthwise convolution*) dan konvolusi saluran (*pointwise convolution*). Konvolusi terbagi mengurangi jumlah parameter dengan memisahkan operasi konvolusi spasial dan konvolusi saluran. MobileNetV2 menggunakan *bottleneck layer* yang terdiri dari konvolusi 1x1 untuk mengurangi dimensi ruang dan ruang parameter sebelum operasi konvolusi terbagi dilakukan. Hal ini membantu mengurangi beban komputasi dan kompleksitas model.

MobileNetV2 menggunakan fungsi aktivasi ReLU6, yang memiliki rentang nilai keluaran antara 0 dan 6. Fungsi ini membantu membatasi rentang nilai dan mempercepat konvergensi model.

Dengan menggunakan arsitektur MobileNetV2, juga memungkinkan model untuk menangkap fitur-fitur penting dari gambar secara bertahap dan mempercepat proses konvolusi. MobileNetV2 menggunakan skala resolusi ganda (*multi-resolution scale*) untuk mengoptimalkan deteksi bibit ikan lele pada berbagai ukuran. Dengan menggunakan lapisan konvolusi dengan

resolusi yang berbeda, model dapat mengenali bibit ikan lele dengan baik pada resolusi yang beragam.

4.2.4 Testing Model

Setelah model dilatih, dilakukan pengujian untuk mengetahui kinerjanya menggunakan *data testing* yang telah dipersiapkan sebelumnya. Model akan membuat prediksi untuk setiap data dalam *data testing* dan hasilnya akan dibandingkan dengan label yang sebenarnya. Metrik evaluasi seperti nilai akurasi, presisi, recall, atau F1-score dapat digunakan untuk mengukur kinerja model.

Nilai presisi (*precision*) menggambarkan sejauh mana model dapat mengidentifikasi dengan benar contoh positif dari kelas yang diinginkan. Dalam CNN, ini berarti berapa persen dari objek yang diklasifikasikan sebagai positif oleh model benar-benar merupakan objek yang diinginkan. Precision dihitung dengan membagi jumlah contoh positif yang diklasifikasikan dengan benar oleh model dengan jumlah total contoh yang diklasifikasikan sebagai positif.

Nilai *recall* mengukur sejauh mana model dapat menemukan semua contoh positif dari kelas yang diinginkan. Dalam CNN, *recall* menggambarkan berapa persen objek positif yang diidentifikasi oleh model dibandingkan dengan jumlah objek positif sebenarnya. *Recall* dihitung dengan membagi jumlah contoh positif yang diklasifikasikan dengan benar oleh model dengan jumlah total contoh positif dalam *dataset*.

F1-score adalah ukuran komposit yang menggabungkan nilai *precision* dan *recall* menjadi satu angka. Ini berguna untuk mengevaluasi model secara keseluruhan, menggabungkan informasi tentang seberapa baik model mengklasifikasikan contoh positif dan seberapa baik model dapat menemukan semua contoh positif yang ada.