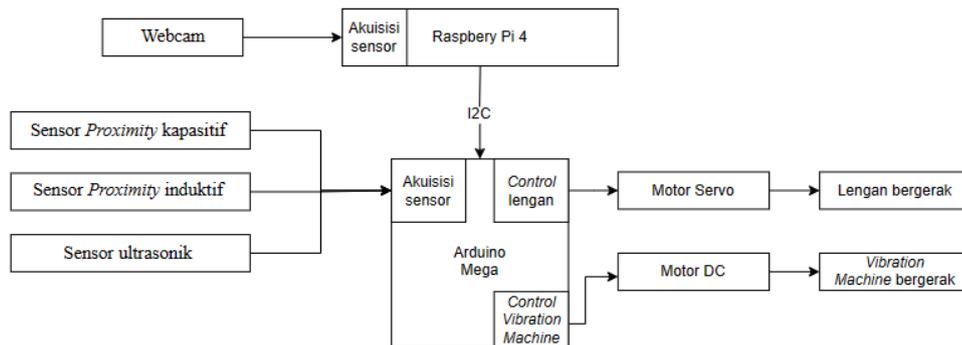


BAB III

PERANCANGAN DAN IMPLEMENTASI SISTEM

3.1 Tinjauan Desain Sistem

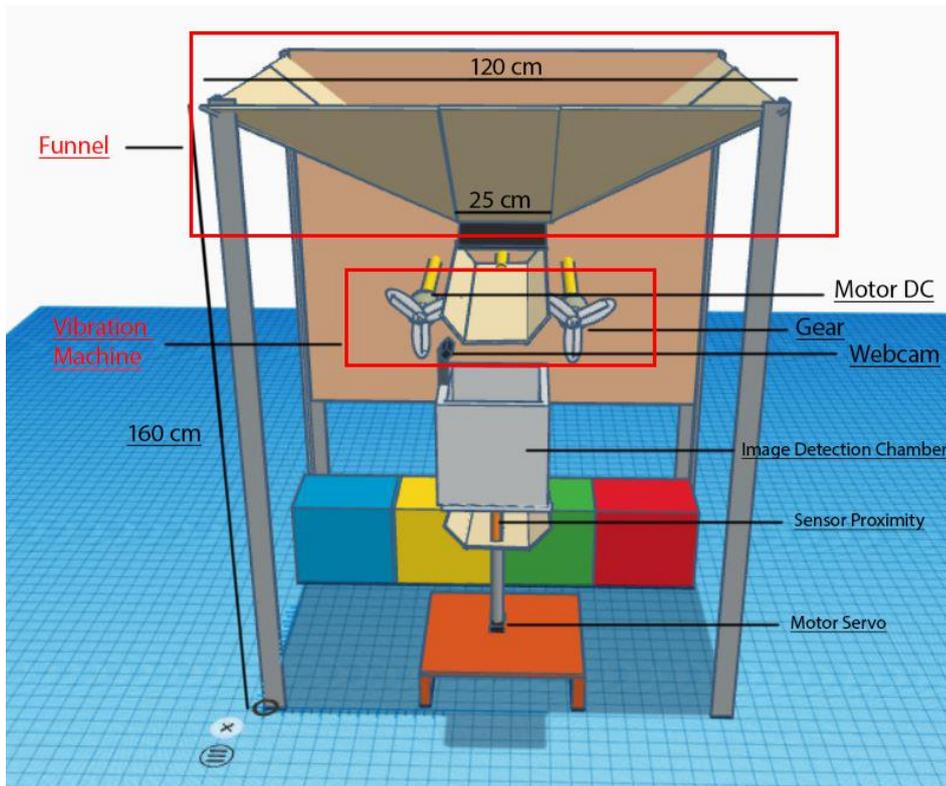
3.1.1 Desain Sistem Keseluruhan



Gambar 3.1 *Block Diagram* Pembuatan Subsystem

Gambar diatas merupakan blok diagram sistem secara keseluruhan. Dimana sistem akan dikendalikan oleh 2 kontroler, yaitu Raspberry Pi 4 dan Arduino Mega 2560. Fokus dari tulisan ini adalah pada bagian Raspberry Pi 4 dan komunikasinya ke Arduino Mega 2560.

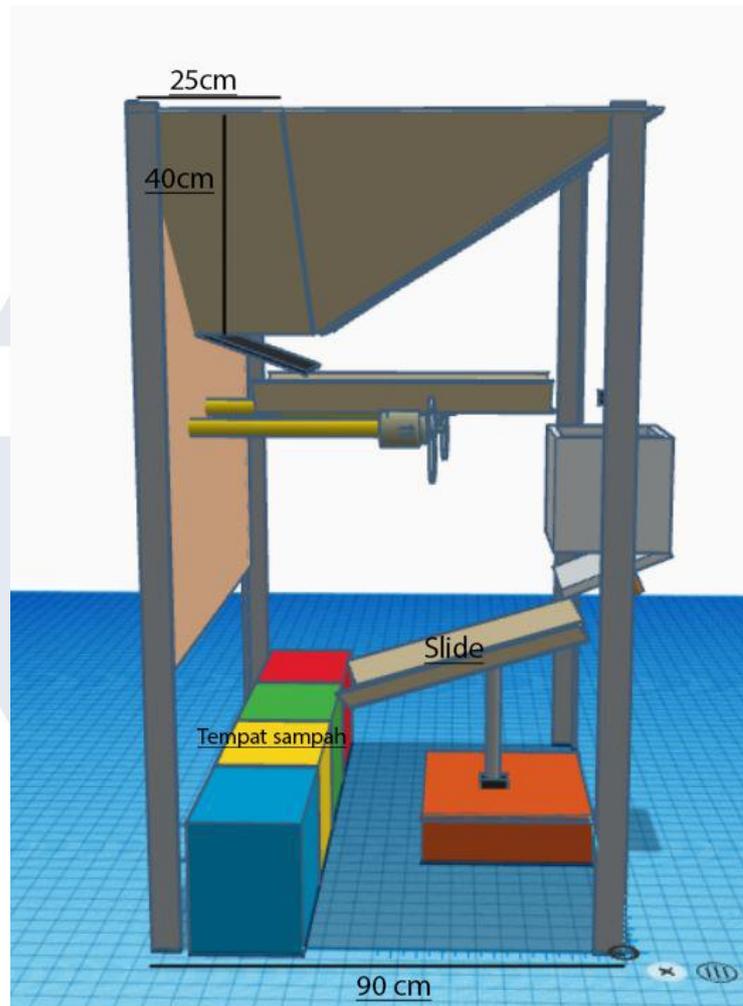
UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.2 Contoh Tampak Depan

UMMN

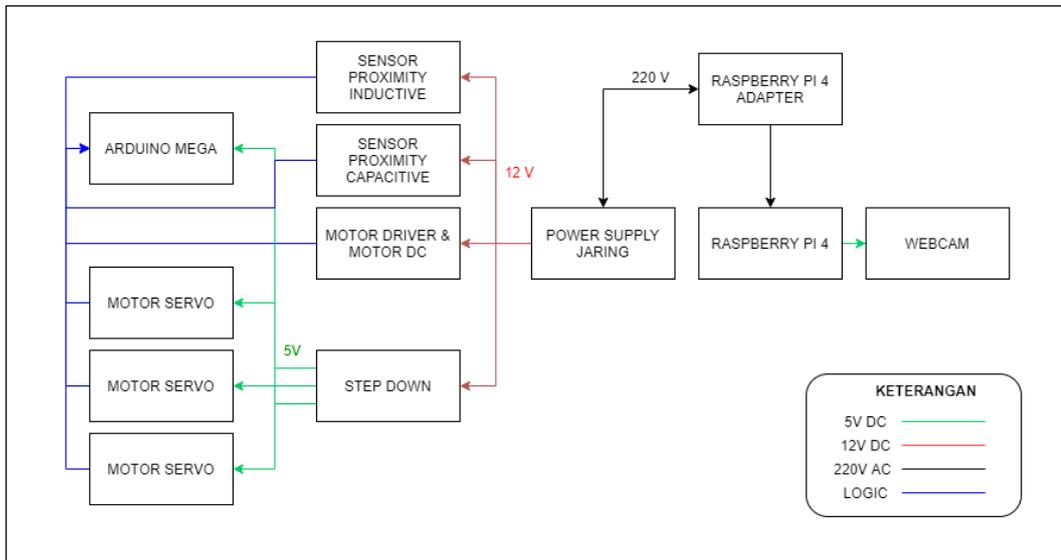
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.3 Contoh Tampak Samping

Gambar diatas adalah rancangan bentuk fisik sistem secara keseluruhan. Fokus dari tulisan ini ada pada *image detection chamber*, dimana ini adalah lokasi kerja dari Raspberry Pi 4.

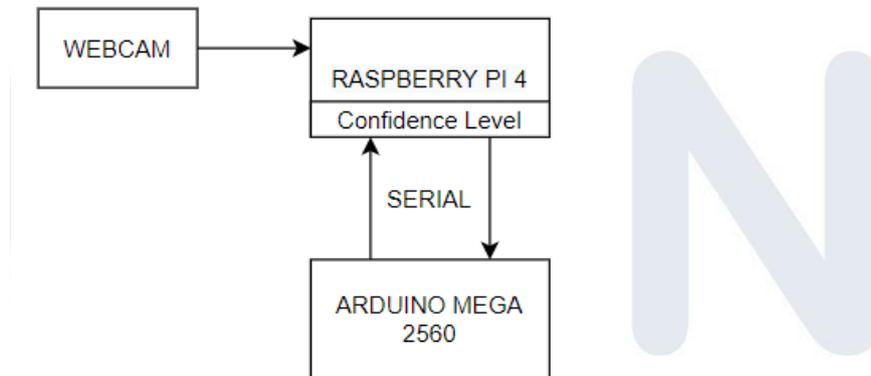
U M M N
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.4 Wiring Diagram

Gambar diatas adalah rancangan *wiring diagram* secara sederhana dari keseluruhan sistem. Subsistem pengolahan citra akan mengambil sumber tenaga listrik secara langsung dari tegangan AC 220V dengan menggunakan adaptor dari Raspberry Pi 4. Raspberry Pi 4 akan mengkonsumsi daya sebesar 5V 2A.

3.1.2 Desain Subsistem Pengolahan Citra



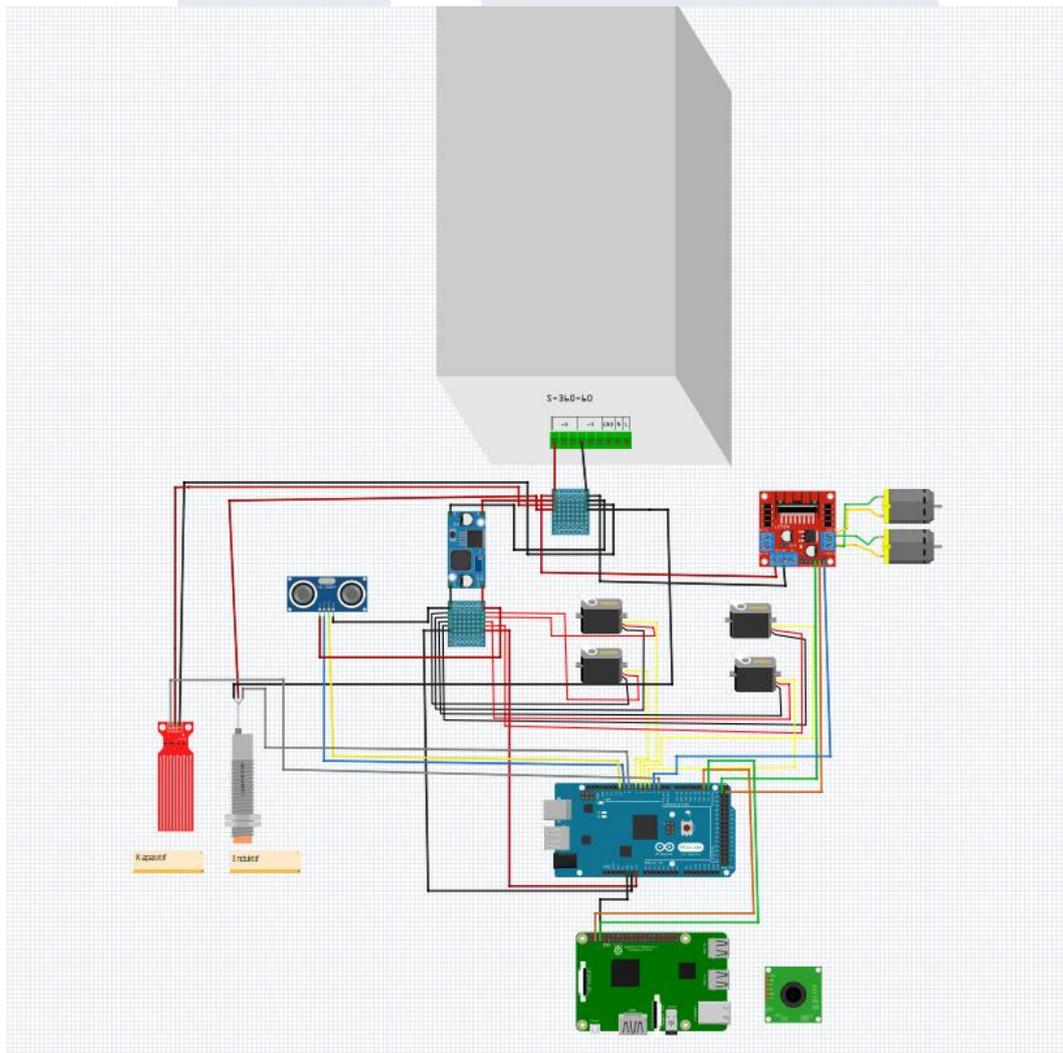
Gambar 3.5 Block Diagram Pembuatan Subsistem

Tabel 3.1 Penjelasan Subsistem Pengolahan Citra

Parameter	Keterangan
Input	<ul style="list-style-type: none"> Gambar
Output	<ul style="list-style-type: none"> <i>Confidence Level</i>
Fungsi	<ul style="list-style-type: none"> Sebagai pendeteksi sampah untuk menentukan klasifikasi sampah melalui klasifikasi gambar.

Perancangan subsistem pengolahan citra mendapatkan masukan berupa gambar yang diambil dengan menggunakan *webcam*. Hasil ini akan diproses oleh model dan diprediksi, masuk ke kategori mana sampah yang dideteksi. Hasilnya berupa *confidence level* yang akan dikirimkan ke Arduino Mega 2560 dengan menggunakan komunikasi serial.

3.1.3 Diagram Sistem



Gambar 3.6 Wiring Pembuatan Subsistem

Dalam diagram subsistem diatas, subsistem pengolahan citra hanya menggunakan beberapa kabel saja yang terhubung ke Arduino Mega 2560. Kabel

tersebut terhubung dengan Tx Rx Arduino Mega 2560 untuk berkomunikasi dengan komunikasi Serial.

3.2 Implementasi Sistem

3.2.1 Hasil Implementasi

3.2.1.1 Hasil Implementasi Kerangka

Produk ini dibangun dengan kerangka yang terbuat dari besi siku lubang, yang kemudian dilapisi dengan triplek. Bagian *funnel* untuk tempat memasukan sampah dilapisi dengan seng dan pada *vibration machine* terbuat dari *Medium Density Fiberboard* (MDF). Pintu pada *inlet* dan dinding-dinding *detection chamber* terbuat dari akrilik. Tampak keseluruhan produk dan masing-masing subsistem dapat dilihat pada Gambar 3.7.



Gambar 3.7 Tampak Depan Produk

3.2.1.2 Hasil Implementasi Subsistem Pengolahan Citra

Subsistem pengolahan citra adalah subsistem yang berfungsi untuk mengklasifikasikan sampah berdasarkan gambar yang diperoleh. Gambar yang diperoleh berasal dari sampah yang masuk ke dalam kotak pemilah, dimana diatas kotak pemilah sudah terpasang sebuah *webcam*. Subsistem

ini akan menjadi sensor utama yang digunakan untuk mendeteksi jenis sampah. Setelah mendeteksi jenis sampah tersebut, maka hasil pendeteksian tersebut akan diberikan kepada Arduino dalam bentuk nilai dengan rentang 0 - 1, dan nilai ini disebut dengan *confidence level*. Jika nilainya dibawah 0,8, maka subsistem sensor pengenalan jenis sampah akan melakukan validasi terhadap jenis sampah agar akurasi sistem meningkat.

Subsistem pengolahan citra ini menggunakan 2 komponen utama, yaitu Raspberry Pi 4 sebagai pengendali utama, dan *webcam* SPC-02 sensor yang akan mengambil gambar. Pengolahan citra yang dilakukan oleh *webcam* akan diproses oleh Raspberry Pi 4, lalu akan digunakan komunikasi serial untuk mengirimkan *confidence level* ke Arduino Mega 2560.

Lingkungan pengembangan subsistem kamera menggunakan Visual Studio (VS) Code pada Windows 10. VS code adalah sebuah *code editor* yang bisa dijalankan di OS berbasis Windows, Linux, MacOS. VS code mendukung penggunaan banyak bahasa pemrograman, dan memiliki *interface* yang bagus sehingga memudahkan dalam penggunaannya. Alasan pemilihan penggunaan *code editor* ini karena mendukung bahasa pemrograman python dan juga dilengkapi dengan berbagai *extension* yang mempermudah dalam proses pengembangan program.

Lingkungan pengembangan lain yang digunakan juga adalah Arduino IDE, dimana digunakan untuk membuat program penerima data serial dari Raspberry Pi 4. Pemilihan Arduino IDE disini karena dapat dengan mudah mengembangkan program untuk Arduino Mega 2560.

Lalu, lingkungan untuk mengaplikasikan model yang sudah dibuat adalah Raspberry Pi 4. *Operating system (OS)* yang digunakan disini adalah Raspbian GNU/ Linux 11 (bullseye). Raspbian sendiri adalah OS yang didasarkan pada sistem operasi Debian.

Implementasi penggunaan model yang dibuat juga memanfaatkan beberapa *package* yang mendukung dalam pembuatan program. Beberapa *package* tersebut antara lain *opencv-python* versi 4.5.3.56 yang digunakan

untuk memanipulasi gambar dan pengambilan data. *Package* selanjutnya juga yang digunakan adalah *tflite-runtime* versi 2.5.0 untuk mengelola model yang sudah dibuat serta melakukan pengklasifikasian berdasarkan model.

Proses implementasi subsistem pengolahan citra dimulai dari pengumpulan *datasets*. Beberapa jenis sampah dikumpulkan terlebih dahulu, seperti sampah minuman kaleng, sampah botol plastik, dan juga beberapa sampah basah berupa buah. Lalu sampah tersebut diletakkan di dalam kotak pemilah dan diambil gambar sampah tersebut. Gambar dari kotak pemilah tanpa sampah pun diambil gambarnya. Setelah proses pengambilan gambar selesai, gambar tersebut lalu di edit terlebih dahulu, seperti melakukan *crop*, lalu ukuran gambar juga diubah agar proses komputasi lebih ringan. Disaat yang bersamaan juga, gambar-gambar tersebut dikelompokkan ke dalam *folder train* yang didalamnya juga berisi *folder* jenis sampah tersebut (logam, plastik, organik, dan *nothing*).

Setelah itu gambar di *upload* ke Teachable Machine. Pada Teachable Machine terlebih dahulu dibuat kelas untuk masing-masing kelas gambar, dimana kelas yang dibuat adalah “Nothing”, “Logam”, “Plastik”, “Organik”. Lalu gambar di-*upload* sesuai dengan kelas yang sudah dibuat. Selanjutnya adalah melakukan perubahan pada beberapa parameter, seperti *epoch*, *batch size*, dan juga *learning rate* untuk mendapatkan model yang sesuai.

Setelah proses pembuatan model selesai, dibuat program *openCV* untuk mengakses *webcam*. *Webcam* tersebut akan mengambil gambar secara terus menerus dari kotak pemisah. Gambar tersebut lalu akan di *resize* dengan ukuran yang sama dengan gambar pelatihan, Setelah itu gambar akan dinormalisasi dan diubah dalam bentuk *array* untuk dianalisa oleh model. Hasil analisis prediksi oleh model akan dinyatakan dalam bentuk nilai 0 untuk “Nothing”, 1 untuk “Logam”, 2 untuk sampah “Plastik”, dan 3 untuk sampah “Organik”.

3.2.2 Hambatan Dan Solusi Implementasi

3.2.1.1 Hambatan Dan Solusi Implementasi Kerangka

Pada awal implementasi, terdapat beberapa bagian yang diubah juga. Permasalahan utama yang dihadapi adalah saat menghubungkan kedua motor DC 555 dengan salah satu *out* L298N saja, yaitu *out 3* dan *out 4*. Ketika menggunakan kedua *out* itu saja hanya untuk kedua motor DC 555, motor tidak bisa berputar, karena beban motor. Oleh karena itu, rangkaian diubah dengan menggunakan *out 1, 2, 3, dan 4*. Dimana *out 1 dan 2* untuk satu motor, sedangkan *out 3 dan 4* untuk motor lain. Dari penerapan rangkaian tersebut, kedua motor berhasil berputar dengan baik dan dapat juga dikendalikan arahnya hanya dengan memanfaatkan program yang dibuat dengan Arduino.

Permasalahan lain yang dihadapi adalah terkait dengan penggunaan roda gigi untuk menggetarkan papan *vibration machine*. Awalnya digunakan dua buah *gear* dengan 32 gigi, namun jumlah gigi yang banyak ini menyebabkan getaran yang terjadi pada papan menjadi kecil. Oleh karena itu, sistem penggerak roda gigi ini diganti dengan penggunaan *cam*. *Cam* yang dibuat memiliki ukuran yang sama dengan roda gigi sebelumnya, namun getaran yang dibuat oleh *cam* lebih kuat dan hasilnya, *vibration machine* bekerja dengan lebih baik.



Gambar 3.8 Gear 32 Gigi



Gambar 3.9 *Cam*

Permasalahan terakhir yang dihadapi adalah terkait motor servo. Kedua servo awalnya dirangkai secara paralel, dimana ketiga kabel servo dihubungkan menjadi satu. Lalu kabel tersebut dihubungkan dengan sumber tegangan 5V dan mendapatkan satu sinyal masukan PWM yang sama dari satu *pin* Arduino. Namun hanya salah satu servo saja yang bekerja, yaitu servo yang terpasang di sisi kiri. Oleh karena itu, rangkaian diubah menjadi paralel hanya untuk sumber tegangan, sedangkan *pin* PWM kedua servo dipisah sehingga kedua servo dapat berputar secara bersamaan.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.2.1.2 Hambatan Dan Solusi Implementasi Subsistem Pengolahan Citra

Permasalahan yang dihadapi adalah proses instalasi *library* pemrosesan gambar pada Raspberry Pi 4. Terdapat beberapa kesulitan yang terjadi saat proses instalasi, karena beberapa *library* yang tidak didukung oleh Raspberry Pi 4. *Library* tersebut antara lain adalah Keras dan juga Tensorflow. Oleh karena itu, solusi yang diterapkan untuk menyelesaikan masalah ini adalah dengan menggunakan Tensorflow-Lite. Tensorflow-lite adalah versi tensorflow yang sudah disederhanakan, dengan tujuan agar dapat diaplikasikan pada mesin dengan kemampuan komputasi terbatas. Tensorflow lite yang di *install* disini adalah tensorflow lite versi 2.5.0.

Permasalahan lainnya adalah penerapan pada Raspberry Pi 4, dimana Raspberry Pi 4 tidak mendukung Tensorflow dan Keras. Oleh karena itu, solusinya adalah menerapkan menggunakan Tensorflow Lite. Namun, karena Keras menggunakan format h5, sedangkan Tensorflow Lite menggunakan tflite yang memiliki ukuran *file* yang lebih kecil. Ukuran *file* yang kecil ini mengurangi kompleksitas model, sehingga akurasi dan kepresisian model berkurang. Sehingga solusi untuk mengatasi masalah tersebut adalah dengan mengubah jumlah data pelatihan, sehingga meningkatkan akurasi dan kepresisian subsistem.

