

TUGAS AKHIR

**ALAT PENJARING ECENG GONDOK DI DANAU TONDANO
BERBASIS CITRA MENGGUNAKAN KAMERA**

*Di Ajukan Kepada Politeknik Negeri Manado Untuk Memenuhi Salah Satu Persyaratan
dalam Menyelesaikan Program Studi Diploma IV*

Jurusan Teknik Elektro

Oleh :

MARCELINO GRANTINO LALOAN

NIM. 11 024 069



Dosen Pembimbing

FANNY J DORINGIN, ST. MT
NIP. 19670430 199203 1 003

KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MANADO
JURUSAN TEKNIK ELEKTRO
2015

ABSTRAK

Laloan, Marcelino. 2015. Alat Penjaring Sampah di Air Berbasis Citra Menggunakan kamera. Tugas Akhir. Jurusan Teknik Elektro. Program Studi Teknik Informatika.

Danau Tondano adalah icon dari Kabupaten Minahasa. Dengan keindahan alam serta panoramanya menjadikan Danau Tondano sebagai tempat wisata yang ramai dikunjungi oleh wisatawan. Namun sekarang, keindahan Danau Tondano tercemari dengan banyaknya eceng gondok dan sampah plastik yang tumbuh di danau. Oleh karena itu, diperlukan alat yang bisa mengvisualisasi sampah dan eceng gondok dengan pengolahan citra sehingga mudah diangkat.

Tujuan dari penelitian ini adalah untuk membuat sebuah program yang dapat mengambil gambar / foto eceng gondok dan sampah di air serta mengolah warna gambar yang diasumsikan sebagai sampah, dan juga membuat sebuah prototype alat yang bisa menjaring eceng gondok dan sampah di air.

Pada penelitian ini, penulis menggunakan dua metode yaitu Studi Lapangan dan Studi Pustaka. Dalam hal ini, Studi Lapangan dilakukan untuk mendesain, menguji dan mengimplementasikan alat dan aplikasi yang di buat. Studi Pustaka dilakukan untuk mencari referensi yang dapat dijadikan dasar kajian dan landasan teori.

Dengan menggunakan prototype alat penjaring sampah ini, bisa memudahkan pekerjaan manusia dalam pengambilan sampah di air dan juga sebagai bentuk pengembangan terhadap teknologi *Mobile Android*.

Kata kunci: *Citra, Pengolahan Gambar, Peralatan.*

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yesus Kristus yang adalah sumber segala ilmu dan pengetahuan, yang telah memberikan hikmat, tuntunan, penyertaan serta lindungan, selama penulis menyelesaikan studi di jurusan Teknik Elektro Politeknik Negeri Manado, sehingga penulis dapat menyelesaikan pembuatan tugas akhir ini. Meskipun dalam penyusunan tugas akhir ini, banyak menghadapi berbagai macam hambatan, rintangan dan tantangan yang harus dilalui, tetapi berkat pertolongan Tuhan dan dukungan dari berbagai pihak sehingga tugas akhir ini dapat terselesaikan.

Penulis menyadari bahwa dengan keterbatasan kemampuan yang ada sehingga penulisan jauh dari kesempurnaan dan masih banyak kekurangan. Oleh karena itu dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang bersifat membangun dari semua pihak yang bertujuan kearah penyempurnaan tugas akhir ini.

Selama proses penyelesaian tugas akhir ini banyak ditunjang dengan bantuan tenaga, pemikiran baik moral maupun material dari banyak pihak. Oleh karena itu, sepantasnyalah bila pada kesempatan ini dengan segala kerendahan hati penulis menyampaikan banyak-banyak terimakasih yang setulus-tulusnya kepada:

1. Ir. Jemmy J. Rangan, MT Selaku Direktur Politeknik Negeri Manado
2. Ir. Jusuf L.Mappadang, MT selaku Ketua Jurusan Teknik Elektro
3. Ir. Nikita Sajangbati, MT Selaku Ketua Program Studi Teknik Informatika

4. Bapak Fanny J Doringin, ST. MT, selaku pembimbing yang telah memberikan banyak masukan sehingga tugas akhir ini dapat terselesaikan.
5. Dosen pengajar di Teknik Elektro terlebih khusus dosen Teknik Informatika yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis.
6. Kedua Orang tua serta adik-adik yang telah memberikan dukungan baik dalam Doa maupun materi sehingga proses penulisan tugas akhir ini dapat terselesaikan.
7. Kepada seluruh teman-teman Teknik Informatika angkatan 2011 yang telah banyak memberikan dorongan dan motivasi selama dalam proses pembelajaran dan penyusunan tugas akhir ini bisa berjalan dengan baik.
8. Kepada sahabat-sahabat terdekat penulis dari *group* TORANG-TORANG yang memberikan motivasi dalam penyusunan skripsi.
9. Kepada saudari Leydi Grace Lontaan yang senantiasa memberikan motivasi dan dukungan serta membantu dalam penyelesaian Tugas Akhir ini.

Dengan segala kerendahan hati, penulis berharap semoga apa yang tertulis dalam tugas akhir ini dapat bermanfaat bagi semua pihak.

Manado, Agustus 2015

Marcelino G. Laloan

DAFTAR ISI

LEMBAR PENGESAHAN	i
KATA PENGANTAR	ii
ABSTRAK	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	xi
DAFTAR LAMPIRAN.....	xii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian.....	2
1.4. Batasan Masalah.....	2
1.5. Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1. Pengolahan Citra Digital	4
2.1.1. Citra Kontinyu	5
2.1.2. Citra Digital	5
2.2. Piksel	6
2.3. Warna Pada Citra Digital.....	7
2.3.1. RGB	7
2.3.2. <i>Greyscale</i>	8
2.3.3. Biner (Hitam Putih)	8

2.4. <i>Thresholding</i>	9
2.5. <i>Cropping</i>	9
2.6. Histogram	9
2.7. <i>Edge Detection</i>	10
2.8. Elemen – Elemen Citra Digital.....	10
2.8.1. Kecerahan (<i>brightness</i>)	10
2.8.2. Kontras (<i>contrast</i>)	11
2.8.3. Kontur (<i>contour</i>)	11
2.8.4. Warna (<i>color</i>)	11
2.8.5. Bentuk (<i>shape</i>)	12
2.8.6. Tekstur (<i>texture</i>)	12
2.9. Format Citra Digital.....	13
2.9.1. <i>Bitmap</i> (BMP)	13
2.9.2. <i>Joint Photographic Expert Group</i> (JPEG / JPG)	13
2.10. Video Digital	13
2.10.1. Resolusi	14
2.10.2. <i>Bit Depth</i>	14
2.10.3. <i>Frame Rate</i>	15
2.11. Perangkat Lunak yang digunakan.....	15
2.11.1. Matlab (R2013a)	15
2.11.2. Arduino Software	16
2.11.3. Android	18
2.11.4. IP Camera	18

2.12. Perangkat Keras Yang Digunakan	19
2.12.1. Arduino Uno	19
2.12.2. Bluetooth HC-05	21
2.12.3. Servo	23
2.12.4. Motor DC	23
2.13. Roadmap Penelitian	24
BAB III METODOLOGI PENELITIAN	18
3.1. Waktu dan Tempat.....	25
3.2. Kerangka Konseptual	25
3.3. Diagram Blok Sistem Penjaring Sampah	26
3.4. Diagram Alur (<i>Flowchart</i>).....	28
3.5. Perancangan	33
3.5.1. Perancangan Hardware	33
3.5.1.1. Perancangan Rangkaian Arduino	34
3.5.2. Perancangan Software	38
3.5.2.1. Perancangan Dan Pembuatan Aplikasi Matlab	39
3.5.2.2. Pembuatan Program Arduino	42
BAB IV PENGUJIAN DAN ANALISA	47
4.1. Tujuan Pengujian.....	47
4.2. Pengujian Input Output Sistem	47
4.3. Pengujian Aplikasi Matlab	48
4.4. Pengujian Sistem Penjaring Sampah	52
BAB V PENUTUP	56

5.1. Kesimpulan	56
5.2. Saran	56
DAFTAR PUSTAKA	58
LAMPIRAN	

DAFTAR TABEL

Nomor		Hal.
1.	Tabel 2.1 Deskripsi Arduino Uno.....	21
2.	Tabel 2.2 Penelitian Terdahulu	22
3.	Tabel 4.1 Tabel Input	47
4.	Tabel 4.2 Tabel Output	48

DAFTAR GAMBAR

Nomor	Hal.
1. Gambar 2.1 Proses Pembentukan Citra	5
2. Gambar 2.2 Contoh Citra Digital	6
3. Gambar 2.3 Representasi Warna RGB Pada Citra Digital	7
4. Gambar 2.4 Tampilan <i>Interface</i> Matlab	16
5. Gambar 2.5 Tampilan <i>Interface</i> Arduino Software	17
6. Gambar 2.6 <i>Board</i> Arduino Uno	19
7. Gambar 2.7 Module Bluetooth HC-05	22
8. Gambar 2.8 Motor Servo	23
9. Gambar 2.9 Motor DC 3V	24
10. Gambar 3.1 Kerangka Konseptual	25
11. Gambar 3.2 Diagram Blok Sistem Penjaring Eceng Gondok dan Sampah Loop Tertutup	26
12. Gambar 3.3 Flowchart Kamera Smartphone (IP Camera)	29
13. Gambar 3.4 Flowchart Program Deteksi Warna	31
14. Gambar 3.5 Flowchart Alat	32
15. Gambar 3.6 Flowchart Keseluruhan Proses Alat Penjaring Eceng Gondok.	32
16. Gambar 3.7 Rangkaian Pengkabelan Dari Arduino Ke Servo	35
17. Gambar 3.8 Rangkaian Pengkabelan Dari Arduino Ke Bluetooth	36
18. Gambar 3.9 Rangkaian Keseluruhan Pengkabelan Arduino	37
19. Gambar 3.10 Skema Rancangan Sistem Keseluruhan	38
20. Gambar 3.11 Smartphone Samsung Ace 3	39

21. Gambar 3.12 Design Form Aplikasi	40
22. Gambar 3.13 Inisialisasi Pin Pada Program Arduino	43
23. Gambar 3.14 Perintah Untuk ke Kanan	44
24. Gambar 3.15 Perintah Untuk Kembali ke Tengah	45
25. Gambar 3.16 Perintah Untuk Menggerakkan ke Kiri	46
26. Gambar 4.1 Deteksi Warna Sampah Merah	49
27. Gambar 4.2 Deteksi Warna Sampah Merah versi 2.....	49
28. Gambar 4.3 Deteksi Warna Sampah Hijau	50
29. Gambar 4.4 Deteksi Warna Sampah Hijau versi 2	50
30. Gambar 4.5 Deteksi Warna Sampah Biru	51
31. Gambar 4.6 Deteksi Warna Sampah Biru versi 2	51
32. Gambar 4.7 Tampak Perahu Penjaring Sampah	52
33. Gambar 4.8 Koneksi Dengan Bluetooth HC-05	53
34. Gambar 4.9 Tampilan Aplikasi “Car Bluetooth RC”	53
35. Gambar 4.10 Motor Servo Bergerak ke Kiri	54
36. Gambar 4.11 Motor Servo Bergerak ke Kanan	54
37. Gambar 4.12 Submarine Motor DC	55

DAFTAR LAMPIRAN

Nomor		Hal.
1.	Kode Program Matlab	62
2.	Kode Program Arduino	70



Motto

*"It's nice to be important,
But it's more important to be nice."*

- John Cassis

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada masa sekarang, *Digital Image Processing* atau biasa disebut dengan pengolahan citra digital sudah banyak diterapkan di berbagai aspek dalam bidang ilmu komputer. Citra sendiri merupakan representasi dari suatu objek yang dapat diabadikan dengan menggunakan berbagai media, salah satunya dengan menggunakan kamera. Dalam pengolahan citra digital, ada banyak hal yang dapat dilakukan pada sebuah citra, misalnya: *brightness editing*, *contrast stretching*, *rotating*, *grayscale*, *color manipulation* dan lain sebagainya.

Pengolahan citra merupakan suatu metode atau teknik pemrograman yang digunakan untuk memproses citra atau gambar dengan jalan memanipulasinya menjadi data gambar yang diinginkan untuk mendapatkan informasi.

Danau Tondano adalah icon dari Kabupaten Minahasa. Dengan keindahan alam serta panoramanya menjadikan Danau Tondano sebagai tempat wisata yang ramai dikunjungi oleh wisatawan. Namun sekarang, keindahan Danau Tondano tercemari dengan banyaknya eceng gondok dan sampah plastic dan eceng gondok yang tumbuh di danau. Dapat diketahui eceng gondok dan sampah – eceng gondok dan sampah yang terdapat di Danau Tondano antara lain berasal dari anak – anak sungai di sekitaran Danau Tondano.

Karena penanganan eceng gondok dan sampah di Danau Tondano masih manual, oleh karena itu diperlukan alat yang bisa mengvisualisasi eceng gondok dan sampah dengan pengolahan citra sehingga eceng gondok dan sampah mudah dideteksi dan diangkat, serta tidak dipengaruhi cuaca tanpa menggunakan tenaga manusia untuk mengambil langsung eceng gondok dan sampah tersebut.

Penulis tertarik membuat perangkat lunak yang dapat digunakan untuk mengambil, mendeteksi, dan mengolah data dari foto / gambar eceng gondok dan sampah pada air, dimana citra bisa menelusuri citra perpixel sehingga bisa

diperoleh gambaran yg jelas tentang eceng gondok dan sampah tersebut, dan alat *prototype* penyaring eceng gondok dan sampah di air menggunakan arduino uno.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang teridentifikasi di atas, maka dapat dirumuskan beberapa rumusan masalah sebagai berikut:

1. Bagaimana merancang program aplikasi pengolahan citra dalam mengidentifikasi gambar / foto dan mengelolanya sehingga diperoleh data yang sesuai.
2. Bagaimana membuat *prototype* alat penjaring eceng gondok dan sampah di air menggunakan arduino uno.

1.3 Tujuan Penelitian

Adapun Tujuan dari pembuatan Tugas akhir ini adalah sebagai berikut.

1. Untuk merancang dan membuat perangkat lunak agar melakukan proses pengambilan gambar / foto eceng gondok dan sampah pada air lalu mengolah data tersebut untuk menampilkan citra yang jelas.
2. Untuk membuat sebuah *prototype* alat penjaring eceng gondok dan sampah yang bisa mengumpulkan eceng gondok dan sampah yang ada di air.

1.4 Batasan Masalah

Dalam merancang dan membuat Tugas akhir ini, penulis membatasi beberapa batasan masalah berikut.

1. Software yang digunakan menggunakan Matlab.
2. Pengujian sistem dilakukan pada loyang berisi air.
3. Pengambilan gambar menggunakan kamera *Smartphone (IP Camera)*.
4. Jarak *wireless* untuk alat berkisar ± 15 meter.
5. Diasumsikan bahwa benda berwarna Merah, Hijau, dan Biru yang terdeteksi oleh kamera adalah eceng gondok dan sampah.
6. Berat benda yang dapat dijaring tidak diperhitungkan.

1.5 Manfaat Penelitian

Manfaat dari penulisan skripsi ini adalah :

1. Dapat mempermudah mengetahui apakah citra / gambar tersebut terdapat eceng gondok dan sampah atau tidak.
2. Mendapatkan rancangan sebuah sistem otomatisasi pendeteksian eceng gondok dan sampah pada air.
3. Dapat mempermudah penjarangan eceng gondok dan sampah di air dengan segala bentuk cuaca menggunakan mikrokontroler Arduino Uno.
4. Mampu membersihkan Danau.

1.6 Metodologi Penulisan

Dalam penelitian Tugas Akhir ini, penulis menggunakan beberapa metode penelitian, sebagai berikut:

1. Studi Lapangan

Metode ini dilakukan penulis untuk mendesain, merancang, menguji, dan mengimplementasikan alat dan aplikasi yang penulis buat dengan melakukan uji coba langsung output-output yang digunakan.

2. Studi Pustaka

Studi pustaka dilakukan untuk mencari referensi yang dapat dijadikan dasar kajian dan landasan teori yang mendukung data-data informasi sebagai acuan dalam melakukan perencanaan, pembuatan, dan percobaan pada alat maupun laporan Tugas Akhir ini.

BAB II

TINJAUAN PUSTAKA

2.1 Pengolahan Citra Digital

Pengolahan citra digital (*digital image processing*) merupakan suatu proses dengan masukan berupa citra dan hasil dari proses tersebut dapat berupa informasi. Pada awalnya pengolahan citra digital ini dilakukan untuk memperbaiki kualitas citra, namun dengan semakin berkembangnya teknologi pengolahan citra digital tidak hanya memperbaiki kualitas citra, melainkan juga dapat mengekstraksi informasi-informasi yang terdapat dari citra tersebut.

Berdasarkan jenisnya suatu citra dapat diklasifikasikan menjadi 2 jenis yaitu citra kontinyu dan citra digital. Adapun penjelasan mengenai kedua jenis citra adalah sebagai berikut.

2.1.1 Citra Kontinyu

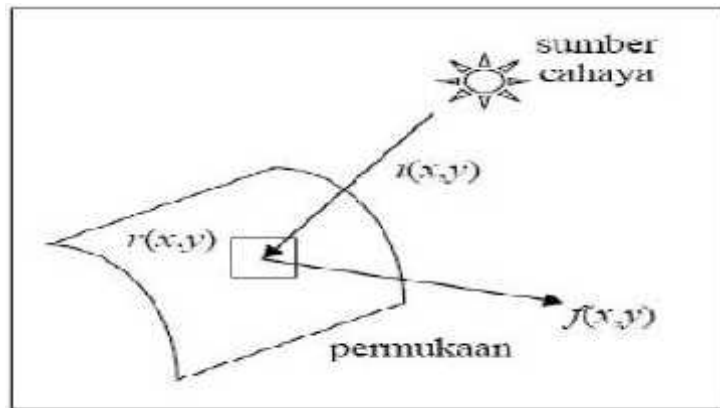
Citra kontinyu adalah fungsi intensitas 2 dimensi $f(x,y)$, adapun x dan y adalah koordinat spasial, dan f pada titik (x,y) merupakan tingkat kecerahan (*brightness*) suatu citra pada suatu titik. Suatu citra diperoleh dari penangkapan kekuatan sinar yang dipantulkan oleh objek. Suatu citra kontinyu dihasilkan dari alat-alat optik yang menerima sinyal analog dari citra tersebut, seperti mata pada manusia, kamera, pemindai (*scanner*) dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam. Secara matematis fungsi intensitas cahaya pada bidang dua dimensi disimbolkan dengan $f(x,y)$, yang dalam hal ini :

(x,y) : koordinat pada bidang dua dimensi

$f(x,y)$: intensitas cahaya (*brightness*) pada titik (x,y)

Nilai $f(x,y)$ sebenarnya adalah hasil kali dari :

- a. $i(x,y)$ = jumlah cahaya yang berasal dari sumbernya (*illumination*), nilainya antara 0 sampai tak berhingga, dan
- b. $r(x,y)$ = derajat kemampuan objek memantulkan cahaya (*reflection*), nilainya antara 0 dan 1.

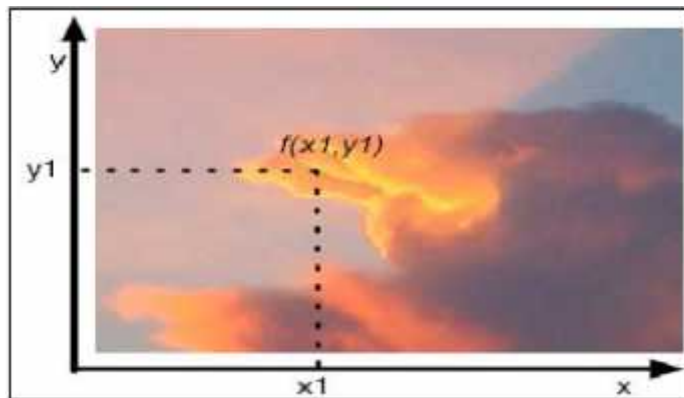


Gambar 2.1 Proses Pembentukan Citra

2.1.2 Citra Digital

Citra digital adalah kumpulan data diskrit yang berbentuk matriks dua dimensi yang angka-angkanya dari matriks tersebut menunjukkan tingkat kecerahan dari titik-titik tersebut. Titik-titik yang membangun citra digital inilah yang sering disebut dengan piksel. Untuk mendapatkan suatu citra digital ada beberapa cara untuk mengakuisisi citra tersebut, antara lain:

1. menangkap citra dengan menggunakan kamera video. Berdasarkan sensor yang digunakan, kamera video dapat dibagi menjadi 2 jenis, yaitu :
 - a. Kamera video yang menggunakan sensor CMOS (*Complementary Metal Oxide Semikonduktor*).
 - b. Melakukan konversi citra analog menjadi citra digital, pada umumnya proses konversi ini menggunakan alat-alat bantu seperti *scanner* dan lain-lain.



Gambar 2.2 Contoh Citra Digital

2.2 Piksel

Piksel adalah unsur gambar atau representasi sebuah titik terkecil dalam sebuah gambar grafis yang dihitung per inci. Seperti yang telah dijelaskan sebelumnya bahwa citra digital adalah matriks dua dimensi yang besarnya tergantung pada besar kecilnya sampling dan nilai tingkat kecerahan dari setiap titiknya dan *range* nilai dari tingkat kecerahan piksel-piksel yang ada pada cira digital tersebut bergantung pada kuantitas ketika citra tersebut diakuisisi, contohnya : pada citra digital *monochrome* dengan tingkat kuantitas 24 bit, maka citra digital tersebut memiliki *range* 224 tingkat kecerahan atau 16,777,216 tingkat kecerahan dari setiap piksel. Pada umumnya citra digital yang banyak digunakan adalah citra digital dengan tingkat kuantitas 8 bit, hal ini didasarkan atas pertimbangan besar ukuran berkas dari citra dan kemampuan mata manusia dalam membedakan warna.

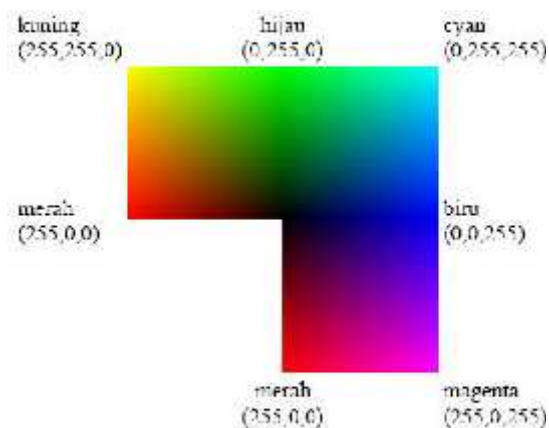
2.3 Warna pada Citra Digital

Mata sebagai indra manusia yang berfungsi untuk menerima rangsangan berupa cahaya dan selanjutnya diproses oleh otak dan diubah menjadi suatu persepsi. Sebagai alat indra yang sensitif terhadap cahaya, mata manusia juga memiliki suatu keterbatasan dalam menerima rangsangan cahaya, cahaya yang dapat diterima oleh mata manusia terbatas dari panjang gelombang 400 nm (violet) sampai 770 nm (merah).

Berdasarkan warnanya citra digital dapat di bagi menjadi beberapa jenis mode warna, antara lain sebagai berikut.

2.3.1 RGB

Jenis ini merupakan tipe mode warna yang paling umum digunakan dalam dunia citra digital, hampir semua peralatan elektronik yang dapat menampilkan citra digital menggunakan tipe jenis ini seperti monitor berwarna dan kamera. Pada umumnya citra dengan mode warna RGB memiliki masing-masing 8 bit *layer* R (*red*), 8 bit *layer* G (*green*), dan 8 bit *layer* B (*blue*). Dengan mengkomposisikan ketiga *layer* warna tersebut akan didapat 2563 atau 16,777,216 warna, yang tentunya sudah mewakili semua warna yang dapat dilihat oleh mata manusia.



Gambar 2.3 Representasi Warna RGB pada citra digital

2.3.2 *Greyscale*

Mode warna *grayscale* adalah mode warna yang dihasilkan dari perbandingan nilai-nilai *red* (R), *green* (G), dan *blue* (B) dengan beberapa konstanta tertentu sehingga menghasilkan warna abu-abu. Mode warna ini digunakan untuk mempermudah dalam melakukan proses pengolahan selanjutnya, karena dengan mengubah mode warna RGB menjadi mode warna *grayscale* maka ketiga layer matriks yang sebelumnya memiliki nilai yang berbeda-beda untuk dikonversikan menjadi satu layer matriks, dengan nilai derajat keabuan tertentu hal ini hanya dilakukan pada proses pengolahan citra digital yang tidak membutuhkan perbedaaan warna sebagai informasi yang akan diolah.

2.3.3 Biner (Hitam Putih)

Citra dengan mode warna biner merupakan citra dengan komposisi warna yang hanya terdiri dari 2 jenis warna saja, yakni hitam dan putih. Mode warna ini banyak dipakai dalam proses-proses pengenalan citra yang tidak membutuhkan informasi warna, seperti proses pengenalan morfologi dari suatu objek, penipisan objek, manipulasi bentuk objek, dan lain-lain. Ada beberapa keuntungan dari citra biner yaitu:

- Kebutuhan memori menjadi lebih kecil karena nilai derajat keabuan hanya membutuhkan representasi satu bit.
- waktu pemrosesan lebih cepat dibandingkan dengan citra *grayscale* karena banyak operasi pada citra biner yang dilakukan sebagai operasi logika.

Terdapat beberapa alasan perlunya dilakukan perubahan dari citra *grayscale* menjadi citra biner, yaitu:

- Untuk mengidentifikasi keberadaan objek, yang direpresentasikan sebagai daerah di dalam citra. Misalnya pemisahan suatu objek dari latar belakangnya.

- Untuk lebih memfokuskan pada analisis morfologi, dalam hal ini intensitas piksel tidak terlalu penting dibandingkan bentuknya.
- Untuk mengkonversikan citra yang telah ditingkatkan kualitas tepinya ke penggambaran garis-garis tepi.

2.4 Thresholding

Thresholding adalah proses untuk mengubah kuantisasi pada citra digital, dengan menggunakan teknik *thresholding* maka derajat keabuan suatu citra dapat diubah sesuai dengan keinginan, selain itu teknik *thresholding* juga sering digunakan untuk mengelompokkan beberapa warna menjadi kelas-kelas tertentu.

2.5 Cropping

Cropping adalah proses pemotongan citra pada koordinat tertentu pada area citra. Untuk memotong bagian dari citra digunakan dua koordinat, yaitu koordinat awal yang merupakan awal koordinat bagi citra hasil pemotongan dan koordinat akhir yang merupakan titik koordinat akhir dari citra hasil pemotongan. Sehingga akan membentuk bangun segi empat yang mana tiap-tiap *pixel* yang ada pada area koordinat tertentu akan disimpan dalam citra yang baru.

2.6 Histogram

Histogram adalah grafik yang menunjukkan frekuensi kemunculan setiap nilai gradasi warna. Bila digambarkan pada koordinat kartesian maka sumbu X (absis) menunjukkan tingkat warna dan sumbu Y (ordinat) menunjukkan frekuensi kemunculan. Histogram warna merupakan fitur warna yang paling banyak digunakan. Histogram warna sangat efektif mengkarakterisasikan distribusi global dari warna dalam sebuah citra digital.

Misalnya citra digital memiliki L tingkat kecerahan, yaitu dari nilai 0 sampai $L - 1$ (misalnya pada citra dengan kuantisasi derajat keabuan 8 bit, nilai derajat keabuan dari 0 – 255).

2.7 Edge Detection

Proses deteksi tepi (*edge detection*) akan melakukan konversi menjadi dua macam nilai yaitu intensitas warna rendah atau tinggi, contoh bernilai nol atau satu. Dan proses *edge detection* berguna untuk menyederhanakan analisis citra dengan mengurangi sejumlah data secara drastis untuk diproses, dengan tetap mempertahankan informasi yang penting mengenai garis-garis batas objek. Gradien adalah hasil pengukuran perubahan dalam sebuah fungsi intensitas, dan sebuah citra dapat dipandang sebagai kumpulan beberapa fungsi intensitas kontinu sebuah citra.

Perubahan mendadak pada nilai intensitas dalam suatu citra dapat dilacak menggunakan perkiraan diskrit pada gradien. Deteksi tepi akan menghasilkan nilai tinggi apabila ditemukan tepi dan nilai rendah jika sebaliknya.

2.8 Elemen-Elemen Citra Digital

Citra digital mengandung sejumlah elemen-elemen dasar, diantaranya adalah sebagai berikut.

2.8.1 Kecerahan (*brightness*)

Kecerahan adalah kata lain untuk intensitas cahaya. Kecerahan pada sebuah titik (piksel) di dalam citra bukanlah intensitas yang sebenarnya, tetapi sebenarnya adalah intensitas rata-rata dari suatu area yang melingkupinya. Sistem visual manusia mampu menyesuaikan dirinya dengan tingkat kecerahan (*brightness level*) mulai dari yang paling rendah sampai yang paling tinggi.

2.8.2 Kontras (*contrast*)

Kontras menyatakan sebaran terang (*lightness*) dan gelap (*darkness*) didalam sebuah gambar. Citra dengan kontras rendah dicirikan oleh sebagian besar komposisi citranya adalah terang atau sebagian besar gelap. Pada citra dengan kontras yang baik, komposisi gelap dan terang tersebar secara merata.

2.8.3 Kontur (*contour*)

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas padapiksel-pikselyang bertetangga. Karena adanya perubahan intensitas inilah mata kita mampu mendeteksi tepi-tepi (*edge*) objek di dalam citra.

2.8.4 Warna (*color*)

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadappanjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang () yang berbeda. Warna merah mempunyai panjang gelombang paling tinggi, sedangkan warna ungu (*violet*) mempunyai panjang gelombang paling rendah. Warna-warna yang diterima oleh mata (sistem visual manusia) merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda.

Penelitian memperlihatkan bahwa kombinasi warna yang memberikan rentang warna yang paling lebar adalah *red* (*R*), *green* (*G*), dan *blue* (*B*). Persepsi sistem visual manusia terhadap warna sangat relatif sebab dipengaruhi oleh banyak kriteria, salah satunya disebabkan oleh adaptasi yang menimbulkan distorsi. Misalnya bercak abu-abu di sekitar warna hijau akan tampak keunguan (distorsi terhadap ruang), atau jika mata melihat warna hijau lalu langsung dengan cepat melihat warna abu-abu, maka mata menangkap kesan warna abu-abu tersebut sebagai warna ungu (distorsi terhadap waktu).

2.8.5 Bentuk (*shape*)

Shape adalah properti intrinsik dari objek tiga dimensi dengan pengertian bahwa *shape* merupakan properti intrinsik utama untuk sistem visual manusia. Manusia lebih sering mengasosiasikan objek dengan bentuknya ketimbang elemen lainnya (warna misalnya). Pada umumnya, citra yang dibentuk oleh mata merupakan citra dwimatra (2 dimensi), sedangkan objek yang dilihat umumnya berbentuk trimatra (3 dimensi). Informasi bentuk objek dapat diekstraksi dari citra pada permulaan prapengolahan dan segmentasi citra. Salah satu tantangan utama pada *computer vision* adalah merepresentasikan bentuk, atau aspek-aspek penting dari bentuk.

2.8.6 Tekstur (*texture*)

Tekstur dicirikan sebagai distribusi spasial dari derajat keabuan didalam sekumpulan piksel-pikselyang bertetangga. Jadi, tekstur tidak dapat didefinisikan untuk sebuah piksel. Sistem visual manusia pada hakikatnya tidak menerima informasi citra secara independen pada setiap piksel, melainkan suatu citra dianggap sebagai suatu kesatuan. Resolusi citra yang diamati ditetapkan oleh skala pada mana tekstur tersebut dipersepsi.

Sebagai contoh, jika kita mengamati citra lantai berubin dari jarak jauh, maka kita mengamati bahwa tekstur terbentuk oleh penempatan ubin-ubin secara keseluruhan, bukan dari persepsi pola di dalam ubin itu sendiri. Tetapi, jika kita mengamati citra yang sama dari jarak yang dekat, maka hanya beberapa ubin yang tampak dalam bidang pengamatan, sehingga kita mempersepsi bahwa tekstur terbentuk oleh penempatan pola-pola rinci yang menyusun tiap ubin.

2.9 Format Citra Digital

Citra Digital memiliki beberapa format yang memiliki karakteristik tersendiri. Format pada citra digital ini umumnya berdasarkan tipe dan cara kompresi yang digunakan pada citra digital tersebut.

Ada dua format citra digital yang sering dijumpai, antara lain sebagai berikut.

2.9.1 *Bitmap* (BMP)

Merupakan format Gambar yang paling umum dan merupakan format *standard windows*. Ukuran *filenya* sangat besar karena bisa mencapai ukuran *Megabytes*. *File* ini merupakan format yang belum terkompresi dan menggunakan sistem warna RGB (*Red, Green, Blue*) di mana masing-masing warna *pixelnya* terdiri dari 3 komponen R, G, dan B yang dicampur menjadi satu. *File* BMP dapat dibuka dengan berbagai macam *software* pembuka Gambar seperti *ACDSee, Paint, Irvan View* dan lain-lain. *File* BMP tidak bisa (sangat jarang) digunakan di *web (internet)* karena ukurannya yang besar.

2.9.2 *Joint Photographic Expert Group* (JPEG/JPG)

Format JPEG merupakan format yang paling terkenal sampai sekarang ini. Hal ini karena sifatnya yang berukuran kecil (hanya puluhan/ratusan KB saja), dan bersifat portable. *File* ini sering digunakan pada bidang fotografi untuk menyimpan *file* foto. *File* ini bisa digunakan di *web (internet)*.

2.10 Video Digital

Video digital pada dasarnya tersusun atas serangkaian *frame*. Rangkaian *frame* tersebut ditampilkan pada layar dengan kecepatan tertentu, tergantung pada *frame rate* yang diberikan (dalam *frame/second*). Jika *frame rate* cukup tinggi, mata manusia tidak dapat menangkap gambar atau *frame*, melainkan menangkapnya sebagai rangkaian yang kontinyu (*video*). Masing - masing *frame*

merupakan citra digital. Suatu citra digital direpresentasikan dengan sebuah matriks yang masing - masing elemennya merepresentasikan nilai intensitas.

Jika I adalah matriks dua dimensi, $I(x,y)$ adalah nilai intensitas yang sesuai pada posisi baris x dan kolom y pada matriks tersebut. Titik-titik ditempatkan pada posisi baris x dan kolom y pada matriks tersebut. Titik-titik ditempatkan pada posisi baris x dan kolom y pada matriks tersebut. Titik-titik ditempatkan *image* di *sampling* disebut *picture elements*, atau sering dikenal sebagai piksel. Karakteristik suatu video digital *ditetapkan* oleh resolusi (*resolution*) atau *frame dimention*, piksel *depth*, dan *frame rate*.

2.10.1 Resolusi

Resolusi (*resolution*) atau *frame dimention* adalah ukuran sebuah *frame*. Resolusi dinyatakan dalam piksel x piksel. Semakin tinggi resolusi, semakin baik kualitas video yang dihasilkan, dalam artian bahwa ukuran fisiknya sama, video dengan resolusi tinggi akan lebih detail. Namun resolusi yang tinggi akan mengakibatkan jumlah *bit* yang diperlukan untuk menyimpan atau mentransmisikannya meningkat.

2.10.2 Bit Depth

Bit Depth mendeteksi jumlah *bit* yang digunakan untuk merepresentasikan tiap piksel pada sebuah *frame*. Kedalaman *bit* dinyatakan dalam *bit/piksel*. Semakin banyak jumlah *bit* yang digunakan untuk merepresentasikan sebuah piksel, yang berarti semakin tinggi kedalaman piksel-nya, maka semakin baik pula kualitasnya, dengan konsekuensi jumlah *bit* yang diperlukan menjadi lebih tinggi. Dengan 1 *byte* (*8bit*) untuk tiap piksel, diperoleh 256 *level* intensitas.

Dengan *level* intensitas sebanyak itu, umumnya mata manusia sudah dapat dipuaskan. Kedalaman piksel paling rendah terdapat pada *binary value image* yang hanya menggunakan 1 *bit/piksel*, sehingga hanya ada dua kemungkinan bagi tiap piksel, yaitu 0 untuk hitam dan 1 untuk putih. Pada kenyataannya semakin sedikit jumlah *bit* yang digunakan untuk tiap piksel, maka kualitas gambar akan semakin turun.

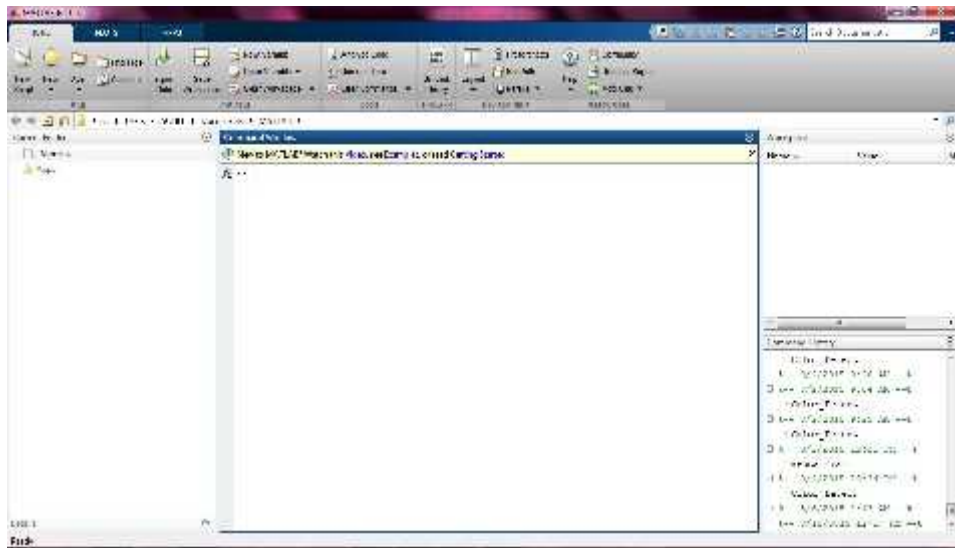
2.10.3 Frame Rate

Frame Rate menunjukkan jumlah *frame* yang digambarkan tiap detik dandinyatakan dengan *frame/second*. Sehubungan dengan *frame rate* ini, ada dua hal yang diperlukan diperhatikan yaitu kehalusan gerakan (*smooth mation*) dan kilatan (*flash*). Kehalusan gerakan *ditetapkan* oleh jumlah *frame* yang berbeda per detik. Untuk mendapatkan gerakan yang halus, video setidaknya harus menampilkan sedikitnya 25 *frame/second*. Kilatan *ditetapkan* oleh jumlah berapa kali layar digambar perdetik (*frame rate*), dengan 29 *frame/second* kilatan sudah dapat dilynapkan. Video yang berkualitas baik akan memiliki *frame rate* yang tinggi, setidaknya sesuai dengan mata manusia, yang berarti membutuhkan jumlah *bit* yang lebih tinggi.

2.11 Perangkat Lunak yang digunakan:

2.11.1 Matlab (R2013a)

MATLAB (*matrix laboratory*) adalah sebuah lingkungan komputasi numerikal dan bahasa pemrograman komputer generasi keempat. Dikembangkan oleh The MathWorks, MATLAB memungkinkan manipulasi matriks, pem-plot-an fungsi dan data, implementasi algoritma, pembuatan antarmuka pengguna, dan peng-antarmuka-an dengan program dalam bahasa lainnya. Meskipun hanya bernuansa numerik, sebuah kotak kakas (*toolbox*) yang menggunakan mesin simbolik MuPAD, memungkinkan akses terhadap kemampuan aljabar komputer. Sebuah paket tambahan, Simulink, menambahkan simulasi grafis multiranah dan Desain Berdasar-Model untuk sistem terlekat dan dinamik.



Gambar 2.4 Tampilan *Interface* Matlab

2.11.2 Arduino Software

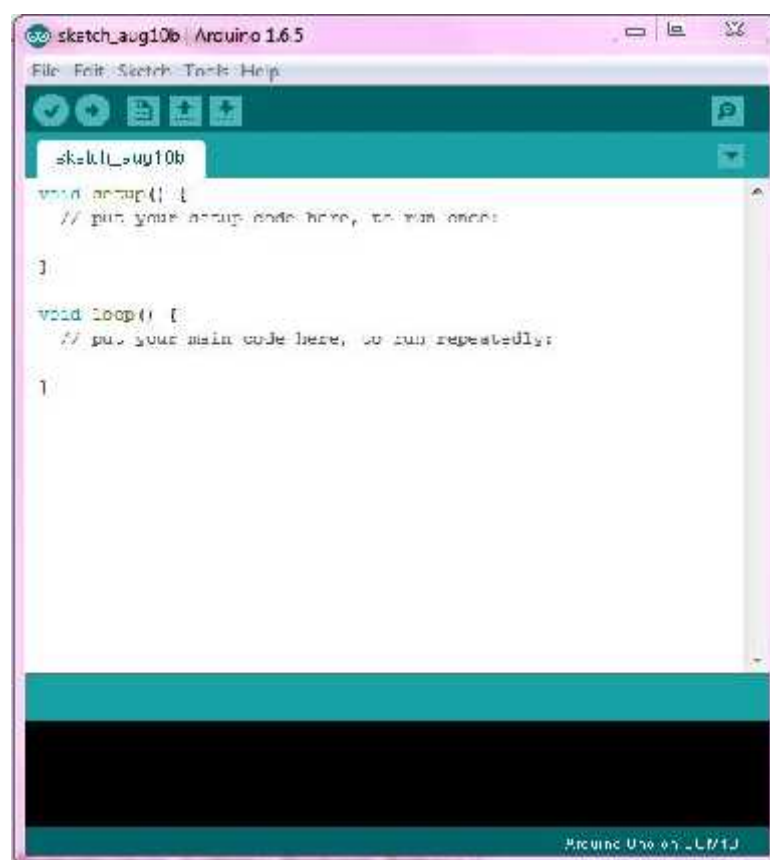
IDE Arduino adalah *software* yang sangat canggih ditulis dengan menggunakan Java. IDE Arduino terdiri dari:

1. Editor program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing*.
2. *Compiler*, sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *Processing*. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.
3. *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam memory didalam papan Arduino.

Sebuah kode program Arduino umumnya disebut dengan istilah *sketch*. Kata "*sketch*" digunakan secara bergantian dengan "kode program" dimana keduanya memiliki arti yang sama.

Bahasa pemrograman Arduino adalah bahasa C. Tetapi bahasa ini sudah dipermudah menggunakan fungsi-fungsi yang sederhana sehingga pemula pun bisa mempelajarinya dengan cukup mudah. Untuk membuat program Arduino dan mengupload ke dalam board arduino membutuhkan software Arduino IDE (Integrated Development Enviroment).

Berikut ini adalah contoh gambar software Arduino IDE (Integrated Development Enviroment) :



Gambar 2.5 Tampilan Interface Arduino Software

2.11.3 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android, Inc. didirikan di Palo Alto, California, pada bulan Oktober 2003 oleh Andy Rubin (pendiri Danger), Rich Miner (pendiri *Wildfire Communications, Inc.*), Nick Sears (mantan VP *T-Mobile*), dan Chris White (kepala desain dan pengembangan antarmuka *WebTV*) untuk mengembangkan "perangkat seluler pintar yang lebih sadar akan lokasi dan preferensi penggunanya". Tujuan awal pengembangan Android adalah untuk mengembangkan sebuah sistem operasi canggih yang diperuntukkan bagi kamera digital, namun kemudian disadari bahwa pasar untuk perangkat tersebut tidak cukup besar, dan pengembangan Android lalu dialihkan bagi pasar telepon pintar untuk menyaingi *Symbian* dan *Windows Mobile* (iPhoneApple belum dirilis pada saat itu). Meskipun para pengembang Android adalah pakar-pakar teknologi yang berpengalaman, Android Inc. dioperasikan secara diam-diam, hanya diungkapkan bahwa para pengembang sedang menciptakan sebuah perangkat lunak yang diperuntukkan bagi telepon seluler.

2.11.4 IP Camera

IP Camera adalah camera dimana dalam pengiriman control signal dan data image menggunakan IP (Internet Protocol). Melalui koneksi Ethernet maupun Wifi, data dikirim dalam bentuk format digital. Beberapa IP Camera terhubung dengan DVR (Digital Video Decoder) atau NVR (Network Video Recorder). Hal ini sering ditemui dalam penggunaan video security atau untuk surveillance dengan tujuan membantu keamanan. Apabila alat tersebut dikoneksikan ke jaringan komputer, maka video yang dihasilkan akan bisa dikendalikan dan dilihat dari tempat dimana saja kita berada.

Masing-masing IP Camera mempunyai alamatunik network, dimana fungsi dari masing-masing IP Camera adalah sebagai video server. Menurut jenis kameranya, ada berbagai macam jenis protocol, termasuk HTTP dan TCP/IP.

Nama lain dari IP Camera adalah OP Network Camera. Kualitas gambar IP Camera lebih bagus daripada camera analog, dan alat ini dapat di install secara group maupun individu.

2.12 Perangkat Keras yang digunakan:

2.12.1 Arduino Uno

Arduino adalah sebuah board mikrokontroller yang berbasis ATmega328. Arduino memiliki 14 pin input/output yang mana 6 pin dapat digunakan sebagai output PWM, 6 analog input, crystal osilator 16 MHz, koneksi USB, jack power, kepala ICSP, dan tombol reset. Arduino mampu men-*support* mikrokontroller; dapat dikoneksikan dengan komputer menggunakan kabel USB..(FeriDjuandi, 2011)



Gambar 2.6 Board Arduino Uno

Arduino adalah merupakan sebuah board minimum system mikrokontroler yang bersifat open source. Didalam rangkaian board arduino terdapat mikrokontroler AVR seri ATmega 328 yang merupakan produk dari Atmel.

Arduino memiliki kelebihan tersendiri disbanding board mikrokontroler yang lain selain bersifat open source, arduino juga mempunyai bahasa pemrogramannya sendiri yang berupa bahasa C. Selain itu dalam *board* arduino sendiri sudah terdapat *loader* yang berupa USB sehingga memudahkan kita ketika kita memprogram mikrokontroler didalam arduino. Sedangkan pada kebanyakan *board* mikrokontroler yang lain yang masih membutuhkan rangkaian *loader* terpisah untuk memasukkan program ketika kita memprogram mikrokontroler. Port USB tersebut selain untuk *loader* ketika memprogram, bisa juga difungsikan sebagai port komunikasi serial.

Arduino menyediakan 20 pin I/O, yang terdiri dari 6 pin input analog dan 14 pin digital input/output. Untuk 6 pin analog sendiri bisa juga difungsikan sebagai output digital jika diperlukan output digital tambahan selain 14 pin yang sudah tersedia. Untuk mengubah pin analog menjadi digital cukup mengubah konfigurasi pin pada program. Dalam *board* kita bisa lihat pin digital diberi keterangan 0-13, jadi untuk menggunakan pin analog menjadi output digital, pin analog yang pada keterangan board 0-5 kita ubah menjadi pin 14-19. dengan kata lain pin analog 0-5 berfungsi juga sebagai pin output digital 14-16.

Sifat *open source* arduino juga banyak memberikan keuntungan tersendiri untuk kita dalam menggunakan *board* ini, karena dengan sifat open source komponen yang kita pakai tidak hanya tergantung pada satu merek, namun memungkinkan kita bisa memakai semua komponen yang ada dipasaran.

Bahasa pemrograman arduino merupakan bahasa C yang sudah disederhanakan syntax bahasa pemrogramannya sehingga mempermudah kita dalam mempelajari dan mendalami mikrokontroller.

Deskripsi Arduino UNO:

Tabel 2.1 Deskripsi Arduino Uno

Mikrokontroler	ATmega 328
Tegangan Pengoperasian	5 V
Tegangan Input yang disarankan	7 – 12 V
Batas Tegangan Input	6 – 20 V
Jumlah pin I/O digital	14 pin digital (6 diantaranya menyediakan keluaran PWM)
Jumlah pin input Analog	6 pin
Arus DC tiap pin I/O	40mA
Arus DC untuk pin 3,3 V	50mA
Memori Flash	32 KB (ATmega 328) sekitar 0,5 KB digunakan oleh bootloader
SRAM	2 KB (ATmega 328)
EPROM	1 KB (ATmega 328)
Clock Speed	16 MHz

(Sumber: <http://www.arduino.cc>)

2.12.2 Bluetooth HC-05

Modul bluetooth seri HC memiliki banyak jenis atau varian, yang secara garis besar terbagi menjadi dua yaitu jenis 'industrial series' yaitu HC-03 dan HC-04 serta 'civil series' yaitu HC-05 dan HC-06.

Modul Bluetooth serial, yang selanjutnya disebut dengan modul BT saja digunakan untuk mengirimkan data serial TTL via bluetooth. Modul BT ini terdiri dari dua jenis yaitu Master dan Slave.

Seri modul BT HC bisa dikenali dari nomor serinya, jika nomer serinya genap maka modul BT tersebut sudah diset oleh pabrik, bekerja sebagai slave atau master dan tidak dapat diubah mode kerjanya, contoh adalah HC-06-S. Modul BT ini akan bekerja sebagai BT Slave dan tidak bisa diubah menjadi Master,

demikian juga sebaliknya misalnya HC-04M. Default mode kerja untuk modul BT HC dengan seri genap adalah sebagai Slave.

Sedangkan modul BT HC dengan nomer seri ganjil, misalkan HC-05, kondisi default biasanya diset sebagai Slave mode, tetapi pengguna bisa mengubahnya menjadi mode Master dengan AT Command tertentu.



Gambar 2.7 Module Bluetooth HC-05

2.12.3 Servo

Servo adalah sebuah motor DC yang dilengkapi rangkaian kendali dengan system *closed feedback* yang terintegrasi dalam motor tersebut. Pada motor servo, posisi putaran sumbu dari motor akan diinformasikan kembali ke rangkaian control yang ada di dalam motor servo.



Gambar 2.8 Motor Servo

Motor Servo disusun dari sebuah motor DC, gearbox, variable resistor (VR) atau potensiometer dan rangkaian control. Potensiometer berfungsi untuk menentukan batas maksimum putaran sumbu motor servo. Sedangkan sudut dari sumbu motor servo berdasarkan lebar pulsa yang ada pada pin control motor servo.

2.12.4 Motor DC

Motor Dc adalah motor listrik yang memerlukan tegangan arus searah pada kumparan medan untuk diubah menjadi energy gerak mekanik. Kumparan medan pada motor dc disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor(bagian yang berputar). Motor DC, sebagaimana namanya, menggunakan arus langsung yang tidak langsung/direct-undirectional. Motor memiliki 3 bagian komponen utama untuk dapat berputar yaitu :

1. Kutub medan

Motor DC sederhana memiliki dua kutub medan, kutub utara dan kutub selatan. Garis magnetic energi membesar melintasi ruang terbuka diantara kutub – kutub dari utara ke selatan.

2. Dinamo

Dinamo yang berbentuk silinder dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, dynamo berputar dalam medan magnet yang dibentuk oleh kutub – kutub, sampai kutub utara dan selatan berganti lokasi.

3. Komutator

Komponen ini terutama ditemukan dalam motor DC, kegunaannya adalah untuk transmisi arus antara dynamo dan sumber daya.



Gambar 2.9 Motor DC 3V

2.13 Roadmap Penelitian

Roadmap bisa diartikan dengan *a map of road* atau peta jalan untuk bisa memberi petunjuk jalan. Sedangkan penelitian dari berbagai sumber dapat diartikan sebagai proses pembentukan dari sebuah teori yang diajukan, proses pencarian dan penemuan jawaban secara ilmiah, proses mencari jawaban atau hal-hal yang ingin diketahui jawabannya, kegiatan ilmiah untuk menemukan, mengembangkan dan menguji kebenaran suatu pengetahuan, dan kegiatan ilmiah guna menemukan ilmu baru. Sehingga, roadmap penelitian bisa diartikan peta jalan atau petunjuk terhadap penelitian yang dilakukan, sebagai upaya untuk bisa mengetahui arah penelitian yang dilakukan.

Berikut table penelitian yang telah dilakukan oleh peneliti sebelumnya mengenai pengambilan sampah di air berbasis citra menggunakan kamera.

Tabel 2.2 Penelitian Terdahulu

No.	Penulis	Judul	Objek Penelitian	Hasil
1.	Nurul Azmi Ritonga	Rancang Bangun Prototype Mesin Sortir Tendan Kelapa Sawit Berbasis Pencitraan Warna Menggunakan Mikrokontroler	Proses Penyortiran kelapa sawit untuk meningkatkan harga beli TBS (Tendan Buah Segar)	Membuat mesin sortir TBS yang dapat menyeleksi buah sawit yang matang atau mentah
2.	Eka Ardhianto, Wiwien Hadikurniawati, dan Zuli Budiarso	Implementasi Metode Image Suntracting dan Metode Regionprops untuk Mendeteksi Jumlah Objek Berwarna RGB pada File Video	Obejek yang memiliki warna pokok RGB pada suatu file gambar bergerak serta dapat diketahui jumlah benda yang dimaksud	Menghasilkan aplikasi yang dapat mendeteksi warna objek pada file video yang terekam secara langsung menggunakan kamera Webcam yang

				sudah tersedia.
3.	Muhammad Andang Novianti	Alat Pendeteksi Warna Berdasarkan Warna Dasan Penyusun RGB Dengan Sensor TCS230	Pengukuran dan pendeteksian warna berdasarkan gelombang yang dihasilkan dari warna dasar penyusun RGB	System pendeteksi warna menggunakan sensor TCS230 yang dibangun dapat bekerja dengan baik dan hasil implementasi menunjukkan bahwa tingkat akurasi alat sangat tergantung beberapa hal seperti pencahayaan, jenis benda berwarna yang akan dideteksi, jarak antara sensor dengan obyek warna.

				<p>Pada pengamatan warna putih, <i>output</i> RGB tinggi karena warna putih kaya spektrum warna yang merupakan gabungan dari spektrum cahaya.</p>
--	--	--	--	---

BAB III

PERANCANGAN SISTEM PENJARING ECENG GONDOK DAN ECENG GONDOK DAN SAMPAH BERBASIS CITRA

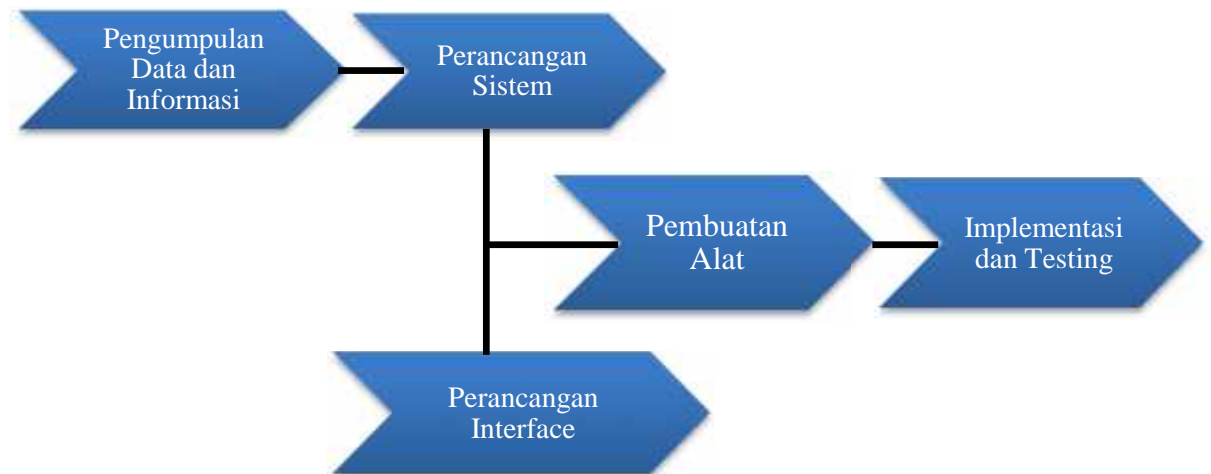
3.1 Waktu dan Tempat

- Lokasi : Kampus Politeknik Negeri Manado
- Waktu : Juni – Agustus 2015

3.2 Kerangka Konseptual

Kerangka konseptual merupakan suatu bentuk kerangka berpikir yang dapat digunakan sebagai pendekatan dalam memecahkan masalah. Biasanya kerangka penelitian ini menggunakan pendekatan ilmiah dan memperlihatkan hubungan antar variable dalam proses analisisnya.

Adapun gambar kerangka berpikir dalam penelitian ini dapat dilihat pada gambar berikut.ambar



Gambar 3.1 Kerangka Konseptual

1. Pengumpulan data & informasi

Penelitian ini diawali dengan pengumpulan informasi mengenai keadaan eceng gondok dan sampah di Danau Tondano lalu bagaimana cara penanganan eceng gondok dan sampah tersebut kepada warga sekitar, setelah mendapatkan informasi penulis merumuskan masalah penanganan eceng gondok dan sampah dan eceng gondok yang bisa dideteksi oleh kamera wireless dan sebuah *prototype* alat yang bisa digunakan untuk menjaring eceng gondok dan sampah di air.

2. Perancangan sistem.

Perancangan sistem merupakan rancang bangun sistem yang dibuat. Perancangan yang dibuat berupa sistem komunikasi kendali alat penjarangan eceng gondok dan sampah dan kamera smartphone yang untuk mengambil gambar kemudian diolah oleh aplikasi dari matlab mendeteksi warna gambar. Perancangan sistem ini dibuat berdasarkan informasi yang telah dibuat berdasarkan informasi yang telah didapat melalui research tentang teknologi komunikasi Android, WiFi, Bluetooth dan Arduino.

3. Perancangan Interface

Perancangan interface merupakan rancang dasar dari aplikasi. Pembuatan interface menggunakan Matlab R2013a.

4. Pembuatan Alat *Prototype*.

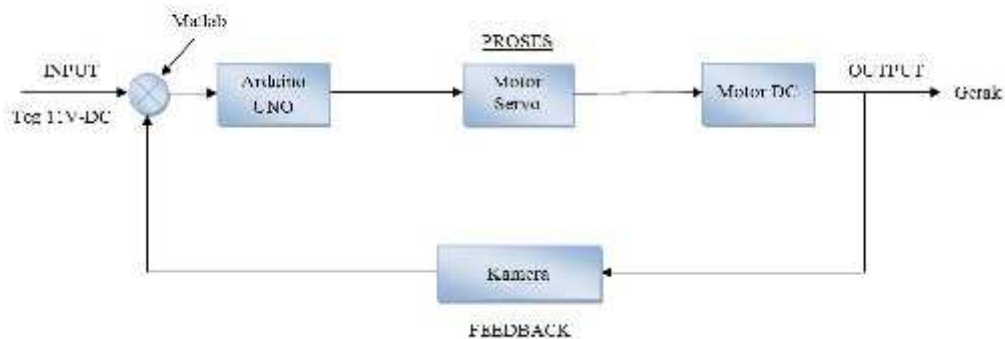
Pembuatan menggunakan komponen - komponen elektronika yang digunakan untuk pengontrolan gerak alat.

5. Implementasi & Testing

Aplikasi dan *Prototype* alat yang telah dibuat kemudian akan diuji dan diimplementasikan pada sebuah kapal mainan yang memiliki 1 Arduino, 1 Bluetooth, 1 Servo, 1 Motor DC, 1 Kamera Smartphone.

3.3 Diagram Blok

Diagram blok adalah suatu pernyataan gambar yang ringkas, dari gabungan sebabdan akibat antara masukkan dan keluaran dari suatu system.



Gambar 3.2 Diagram Blok Sistem Penjaring Eceng Gondok dan Sampah Loop Tertutup

Pada Gambar 3.2 memperlihatkan bagian-bagian dari perancangan sistem pengembangan pengolahan citra untuk penjaringan eceng gondok dan sampah di danau Tondano, yang meliputi beberapa bagian penting, yaitu:

a. Input

Pada bagian Input terdiri dari 2 yaitu tegangan 11Volt-DC sebagai sumber daya untuk menggerakkan alat penjaring dan Matlab untuk mengolah citra warna eceng gondok dan sampah di air. Kedua masukkan (*Input*) ini disebut juga masukkan awal untuk perancangan sistem pengembangan pengolahan citra alat penjaring eceng gondok dan sampah.

b. Proses

Mikrokontoller Arduino Uno menjadi otak atau salah satu inti dalam system ini, Arduino akan mengontrol Motor DC dan Motor Servo untuk dapat menggerakkan kapal penjaring. Arduino di program menggunakan IDE yang berbasis Open Source dan menggunakan bahasa pemograman C++, program perintah yang telah di-compile kemudian di upload ke dalam Arduino, yang kemudian menjadikan Arduino sebagai otak yang siap untuk menghubungkan Input dan Output dari Sistem ini sebagai penerima perintah, eksekutor, dan sebagai pengirim feedback ke android device.

c. Feedback

Pada proses perancangan sistem ini, ada 1 feedback dimana wifi akan menghubungkan kamera smartphome dengan PC, sehingga gambar yang diambil melalui kamara bisa ditampilkan di dalam Matlab.

d. Output

Bagian keluaran (*Output*) ini merupakan hasil dari beberapa masukan yang telah diproses pada bagian proses yang sudah dijelaskan sebelumnya. *Output* yang dihasilkan berupa gerak. Dalam sistem ini, citra yang diperoleh oleh Matlab berasal dari kamera smartphome (ip camera) yang diletakkan di atas kapal.

3.4 Diagram Alur (Flowchart)

Pada pembuatan aplikasi ini, dibutuhkan suatu teknik perancangan yang mempunyai struktur yang baik, biasanya diawali dengan pembuatan diagram alur (flowchart). Diagram alur digunakan untuk menggambarkan terlebih dahulu apa yang harus dikerjakan sebelum mulai merancang atau membuat suatu system seperti yang akan dijelaskan dibawah ini. Berikut adalah diagram alur (flowchart) dari aplikasi android dan program Arduino yang akan dibuat.

- Flowchart Kamera



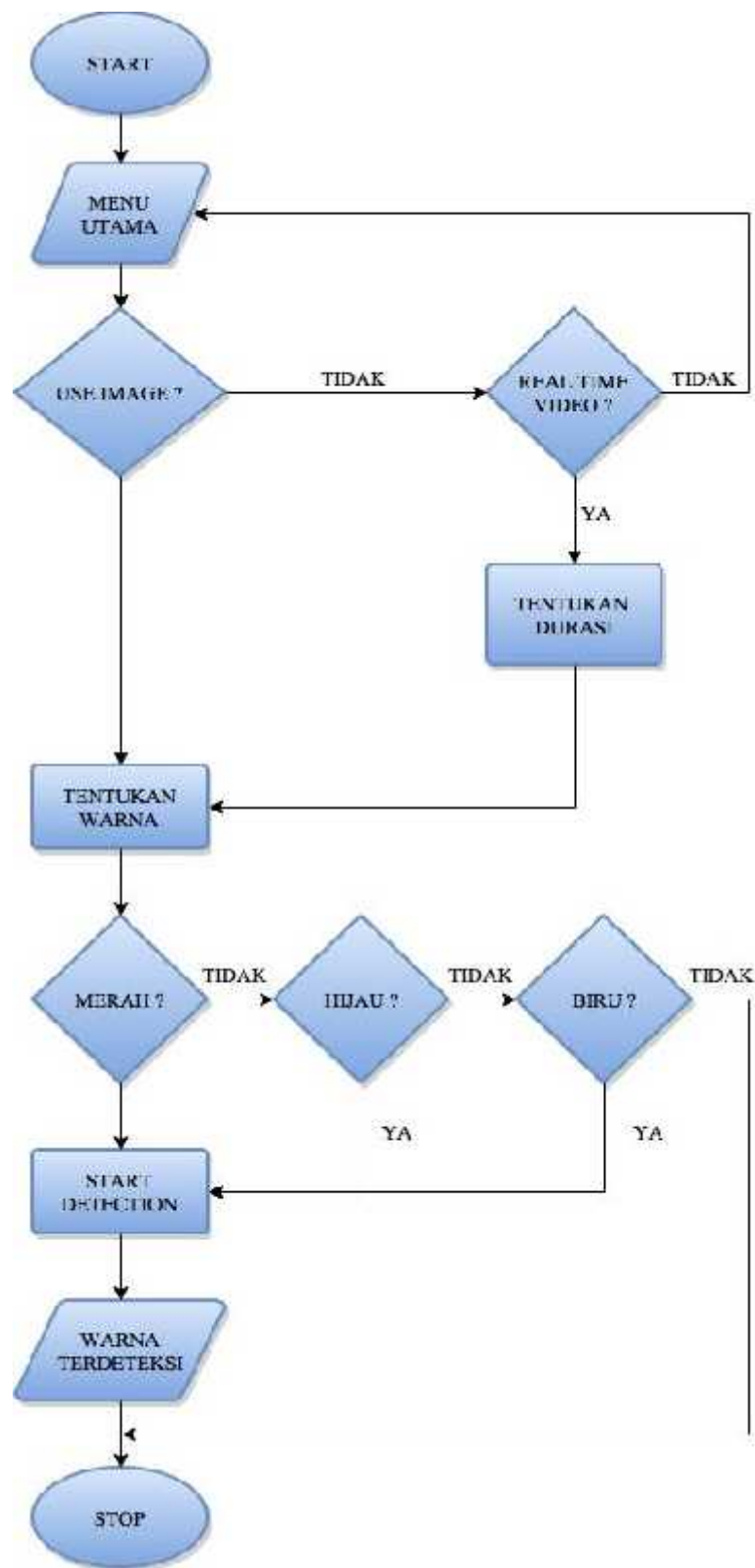
Gambar 3.3 Flowchart Kamera Smartphone (IP Camera)

Penjelasan Flowchart Kamera:

- Mulai
 - Set IP: dalam proses ini aplikasi untuk IP Camera dari android secara otomatis akan member alamat IP kemudian alamat IP tersebut diinputkan ke dalam driver IP Camera di Laptop.
 - Jika telah tersambung maka pengambilan gambar dari kamera smarphone sudah bisa dilihat dari laptop. Tetapi jika tidak, maka input kembali IP.
 - Jika kondisi diaatas telah terpenuhi, maka Live IP CAM sudah bisa digunakan.
 - Berhenti
-
- Flowchart Aplikasi

Penjelasan Flowchart Aplikasi:

- Mulai
- Ketika program pertama dijalankan, tampilan pertama yang muncul adalah form utama.
- Terdapat 2 pilihan untuk mengelola warna citra yaitu dengan Use Image dan Real Time Video.
- Jika menggunakan real time video, selanjutnya pilih durasi untuk pengelolaan citra.
- Setelah itu, pilih warna yang akan dideteksi, dengan pilihan warnanya yaitu merah, hijau, biru.
- Jika warna sudah dipilih, proses untuk mendeteksi warna sudah bisa dimulai. Deteksi akan dilakukan sesuai dengan warna yang telah dipilih sebelumnya.
- Setelah proses selesai, maka akan muncul warna yang telah terdeteksi.
- Jika semua kondisi telah terpenuhi, maka program berakhir.



Gambar 3.4 Flowchart Program Deteksi Warna

- Flowchart Alat *Prototype*

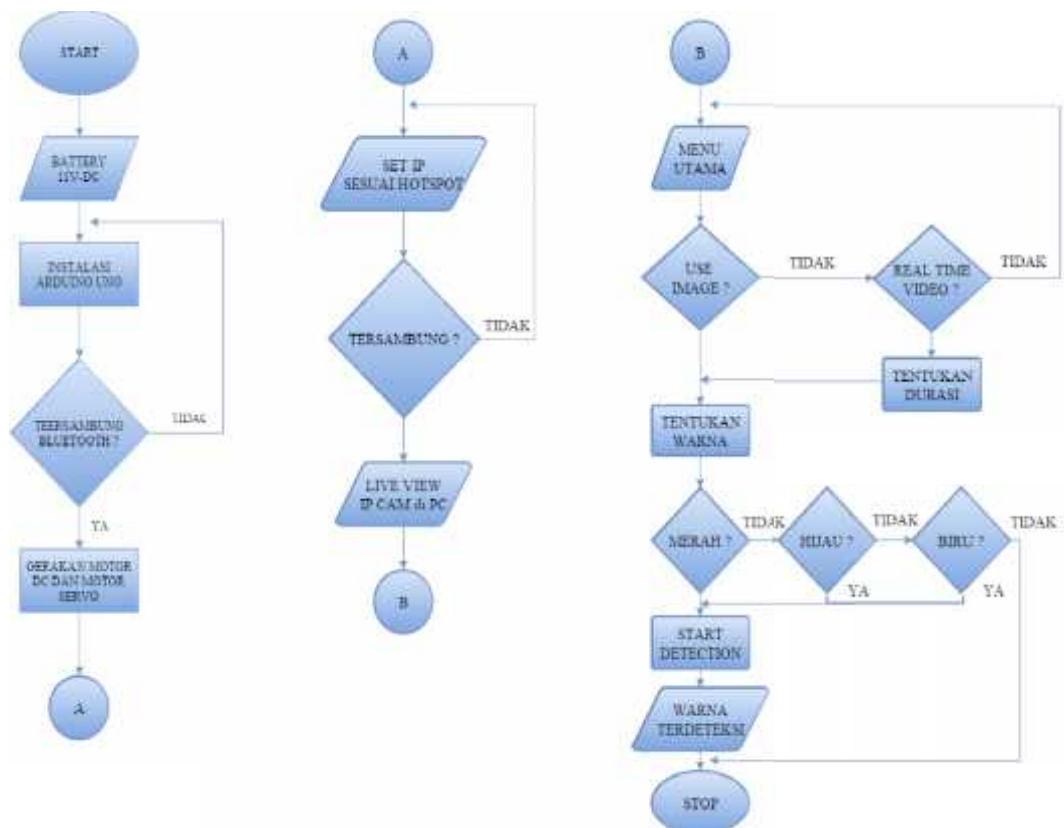


Gambar 3.5 Flowchat Alat

Penjelasan Flowchart Alat *Prototype*:

- Mulai
- Sambungkan Battery sebagai sumber arus tegangan untuk menyuplai daya ke masing-masing komponen.
- Setelah dihubungkan dengan Battery arduino secara otomatis arduino akan mengontrol komponen lain yang telah terhubung ke arduino.
- Pada bagian ini di perlukan Bluetooth sebagai sambungan komunikasi antara komponen dan aplikasi penggerak.
- Jika bluetooth telah terhubung, maka komponen-komponen tersebut bisa digerakkan melalui aplikasi dari smartphone.
- Jika semua kondisi telah terpenuhi, maka proses berhenti.

Berikut adalah gambar keseluruhan proses dari alat penjaring enceng gondok dan sampah di danau tondano berbasis citra menggunakan kamera:



Gambar 3.6 Flowchart Keseluruhan Proses Alat Penjaring Enceng Gondok

3.5 Perancangan

Pada tahap perancangan ini, penulis membagi perancangan kedalam dua bagian penting, yaitu Perancangan Hardware dan Perancangan Software. Pada perancangan hardware membahas tentang perancangan rangkaian Arduino.

Pada perancangan software sendiri terbagi menjadi dua bagian yaitu perancangan program untuk arduino menggunakan bahasa pemrograman C, dan perancangan GUI untuk mendeteksi warna menggunakan bahasa pemrograman Matlab.

3.5.1 Perancangan Hardware

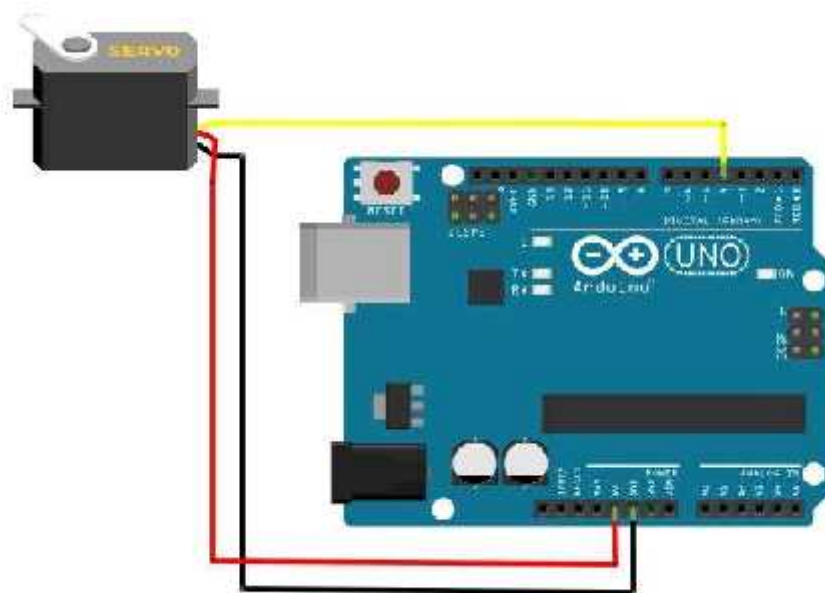
Tahap perancangan Hardware dalam system ini yaitu Perancangan Rangkaian Arduino. Adapun komponen-komponen yang digunakan pada perancangan ini yaitu:

1. Microcontroller Arduino Uno
2. Module Bluetooth HC-05
3. Battery Li-Po 1300mAh 3S 25C
4. Digital Servo Metal Gear #MG945
5. Kabel Jumper Male – Male
6. Kabel Jumper Male – Female
7. Obeng
8. Mainan Perahu
9. Styrofoam
10. Stik Kayu
11. Submarine Motor DC
12. Kamera Smartphone

3.5.1.1 Perancangan Rangkaian Arduino

Dalam sistem penjarangan eceng gondok dan sampah ini menggunakan 1 buah arduino, 1 buah motor servo, 1 buah module Bluetooth dan 1 battery 11Volt. Arduino berfungsi sebagai otak untuk menggerakkan komponen lainnya. Servo berfungsi untuk membelokkan kapal ke kiri maupun kekanan. Untuk menjalankan servo membutuhkan tegangan listrik untuk bekerja, tegangan listrik ini diambil dari arduino. Module Bluetooth digunakan untuk memenuhi kebutuhan system dalam hal komunikasi data dari Android ke Arduino. Berikut adalah rangkaian pengkabelan arduino.

1. Perancangan Servo ke Arduino



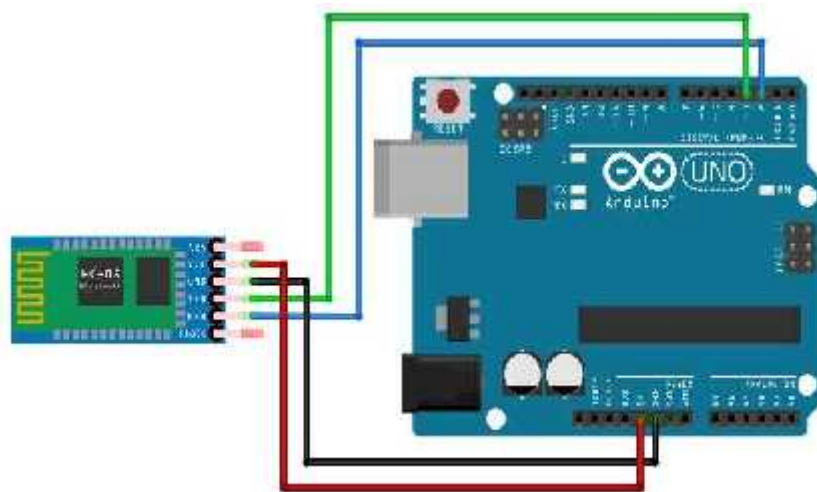
Gambar 3.7 Rangkaian Pengkabelan Dari Arduino Ke Servo

Sudut putaran dari sumbu motor servo diatur berdasarkan lebar pulsa (pulse) yang dikirim melalui kaki sinyal yang berwarna kuning, sedangkan kabel berwarna merah adalah jalur tegangan sebesar 5v dan hitam sebagai ground.

Pada Gambar 3.6 diatas, konfigurasi untuk kaki sinyal (kabel kuning) motor servo berada pada pin digital 4. Tenaga yang disuplai ke motor dalam bentuk sinyal gelombang kotak dari magnet konstan dengan merubah lebar pulsa atau Duty Cycle. Duty Cycle adalah perbandingan antara lamanya waktu on terhadap waktu off dalam suatu periode waktu.

Cara kerja dari Gambar 3.6 adalah motor servo akan bekerja ketika salah satu pin yang terhubung dengan Arduino mendapatkan perintah dari Android, dimana perintah ini akan dikonversikan oleh Arduino menjadi bilangan besaran PWM yang akan mempengaruhi besaran derajat, arah serta kecepatan putaran dari motor servo.

2. Perancangan Bluetooth ke Arduino



Gambar 3.8 Rangkaian Pengkabelan Dari Arduino Ke Bluetooth

Demi memenuhi kebutuhan system dalam hal komunikasi data dari Android ke Arduino yang menggunakan media komunikasi nirkabel, maka diperlukan sebuah device atau alat yang dapat

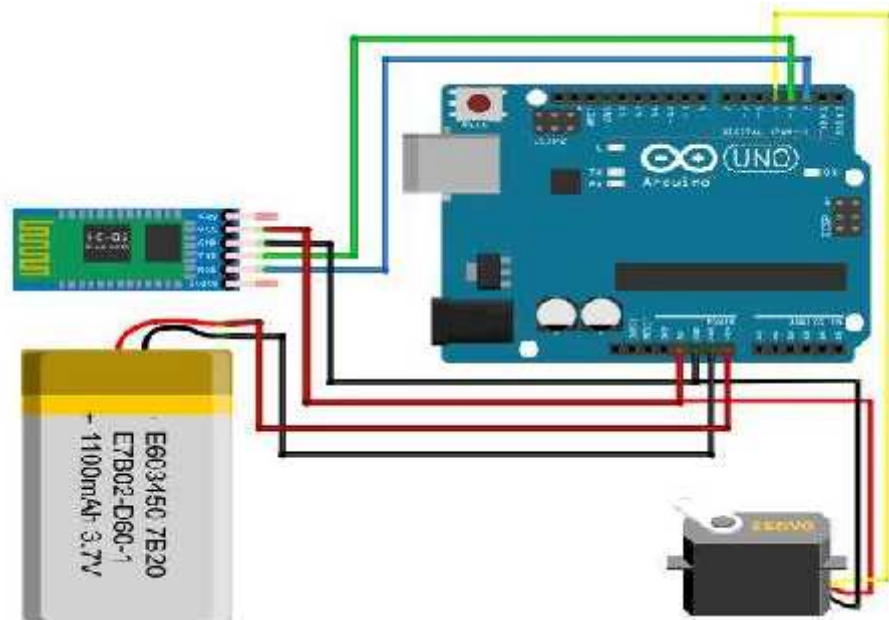
mendukung Arduino untuk dapat berkomunikasi melalui Bluetooth.

Pada system ini, penulis menggunakan module Bluetooth HC-05.

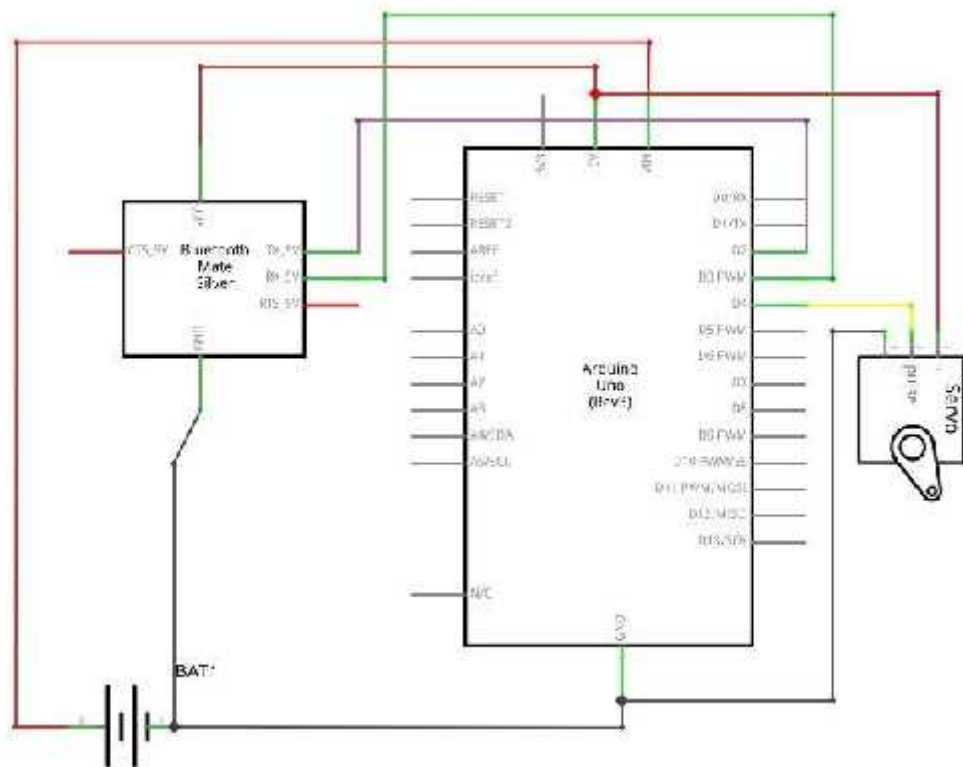
Komunikasi antara Bluetooth dan Arduino dihubungkan oleh oleh kabel jumper. Kabel jumper dihubungkan menggunakan pin digital 2 sebagai data yang akan ditransmisikan, pin digital 3 sebagai data yang telah diterima, 5V, dan GND pada arduino. Dengan menggunakan module Bluetooth HC-05 memungkinkan Arduino menerima data dari android melalui Bluetooth.

3. Rangkaian Keseluruhan Arduino

Setelah melakukan perancangan perangkat keras dari seluruh komponen utama seperti Motor Servo, Bluetooth HC-05, Battery dan Arduino serta output-output yang digunakan, maka rangkaian system keseluruhan dapat terlihat seperti pada gambar 3.8 sebagai berikut:



Gambar 3.9 Rangkaian Keseluruhan Pengkabelan Arduino



Gambar 3.10 Skema Rancangan Sistem Keseluruhan

3.5.2 Perancangan Software

Sesuai dengan system kerja alat penjaring eceng gondok dan sampah menggunakan citra ini, pendektasian dilakukan menggunakan pemrograman Matlab dan untuk instalasi board Arduino memerlukan software dengan bahasa pemrograman C. Berikut akan dijelaskan tentang pemrograman Matlab dan Arduino untuk kebutuhan system.

3.5.2.1 Perancangan Dan Pembuatan Aplikasi Matlab

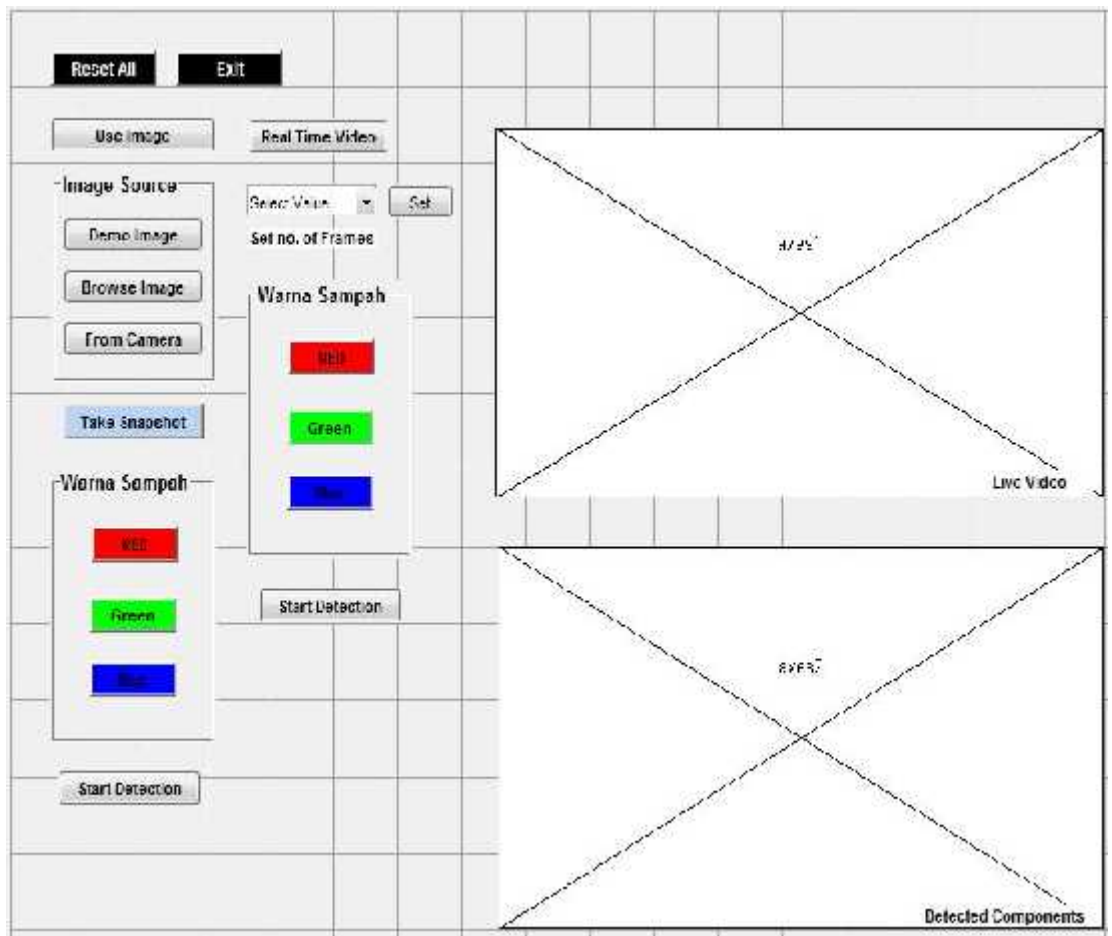
System ini menggunakan aplikasi dari pemrograman matlab untuk mendeteksi warna eceng gondok dan sampah di air, dimana aplikasi akan

membaca warna objek di air yang diasumsikan sebagai eceng gondok dan sampah. Proses pendeteksian warna bisa menggunakan dua proses yaitu dengan mengelolah gambar eceng gondok dan sampah yang telah ada kemudian dideteksi warnanya juga bisa melalui real time video. Pengambilan video sendiri menggunakan kamera Smartphone (*Ip Camera*) yang telah terhubung dengan WiFi. Untuk smartphone sendiri penulis menggunakan smartphone Samsung Ace 3.



Gambar 3.11 Smartphone Samsung Ace 3

Pembuatan aplikasi ini diawali dengan membuat interface atau tampilan GUI. Berikut adalah design dari aplikasi pendeteksi warna:



Gambar 3.12 Design Form Aplikasi

Penjelasan untuk form aplikasi adalah sebagai berikut:

1. Button Use Image
 Button ini berfungsi untuk memilih jika ingin mendeteksi warna dengan gambar yang telah ada.
2. Button Real Time Video
 Button ini berfungsi untuk memilih jika ingin mendeteksi warna menggunakan *realtime* video.
3. Button Take Snapshot
 Button ini berfungsi memotret gambar

4. Button Start Detection

Button ini berfungsi untuk memulai pendeteksian warna

5. Groupbox Image Source

Grupbox ini berfungsi untuk mengelompokkan button-button yang akan digunakan untuk memproses pendeteksian warna pada gambar yang sudah ada. Groupbox ini terdapat 3 button yaitu button *demo image* yang berfungsi untuk mendemokan semple warna, button *browse image* untuk mencari gambar dari penyimpanan, button *form camera* untuk mengambil gambar dari camera.

6. Groupbox warna eceng gondok dan sampah

Groupbox ini berfungsi untuk mengelompokkan button-button untuk memilih warna yang akan diproses. Dalam groupbox ini terdapat tiga button, yaitu button *red* dimana button ini digunakan jika ingin mendeteksi warna merah, button *green* jika ingin mendeteksi warna hijau, button *blue* jika ingin mendeteksi warna biru.

7. Combobox select Value

Combobox ini berfungsi untuk memilih durasi waktu pendeteksian warna.

8. Button Set

Button ini berfungsi untuk menetapkan durasi yang sebelumnya telah dipilih dari combobox.

9. Axes 1

Axes 1 berfungsi sebagai frame untuk menampilkan realtime video dari kamera.

10. Axes 2

Axes 2 berfungsi sebagai frame untuk menampilkan warna yang telah terdeteksi.

11. Button reset all

Untuk mengembalikan form ke tampilan awal (sebelum memulai proses)

12. Button exit

Button exit berfungsi untuk keluar dari aplikasi.

3.5.2.2 Pembuatan Program Arduino

Microcontroller Arduino Uno dapat bekerja dan memproses datagram yang dikirimkan dari aplikasi android hanya jika didalamnya sudah dimasukkan listing program, program yang dimasukkan kedalam Arduino IDE 1.6.5. fungsi program disini antara lain, menginisialisasi pin-pin mana saja yang akan menjadi output atau input, mengubah datagram yang akan dikirim dari Aplikasi Android menjadi perintah Logika “Right” atau “Left”.

Dalam pemrograman Arduino ini sendiri menggunakan bahasa pemrograman C. listing program Arduino ini dikenal dengan nama sketch. Dalam setiap sketch memiliki dua fungsi penting yaitu “void setup() {}” dan “void loop() {}”. Pembuat program Arduino ini sendiri dimulai dengan menginisialisasi pin – pin mana saja yang akan digunakan oleh system, berikut potongan codingnya :

```
RC_CanoAversion1 | Arduino 1.6.5
File Edit Sketch Tools Help
RC_CanoAversion1 $
#include <SoftwareSerial.h>
#include <Servo.h>

int bluetoothTx = 2;
int bluetoothRx = 3;

SoftwareSerial bluetooth(blueetoothTx, bluetoothRx);

Servo myservo;
int ServoPort = 4;
int ValueLeft = 60;
int ValueMid = 30;
int ValueRight = 0;
```

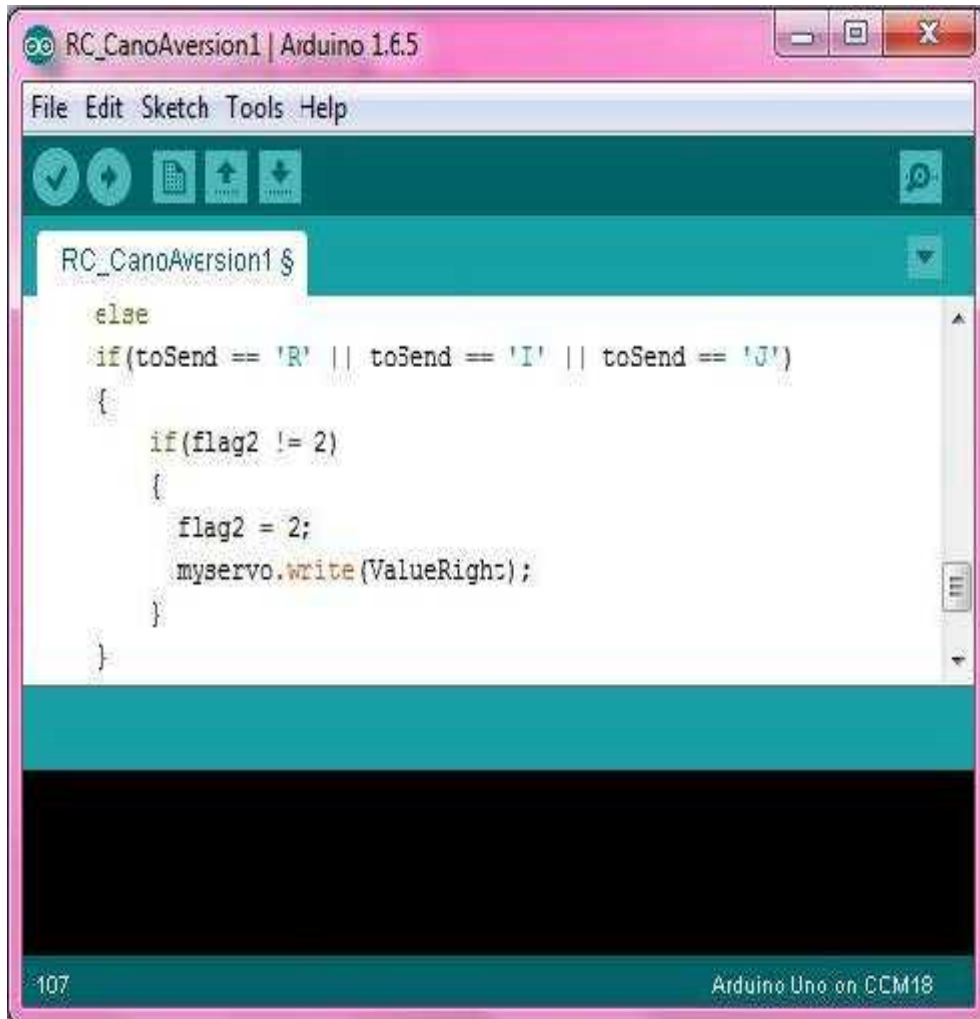
Gambar 3.13 Inisialisasi Pin pada program Arduino

Keterangan:

- a. “bluetoothTx” untuk data yang akan ditransmisikan menggunakan pin 2.
- b. “bluetoothRx” untuk data yang diterima menggunakan pin 3.
- c. “servoPort” untuk servo menggunakan pin 4.
- d. “ValueLeft” perintah untuk menggerakkan ke kanan.
- e. “ValueMid” perintah untuk menggerakkan ke tengah.
- f. “ValueRight” perintah untuk menggerakkan kekanan.

Setelah proses inisialisasi selesai, maka lanjut ke tahap selanjutnya yaitu membuat perintah untuk menggerakkan servo ke kanan, tengah, dan kiri.

- a. Perintah untuk menggerakkan Servo ke kanan

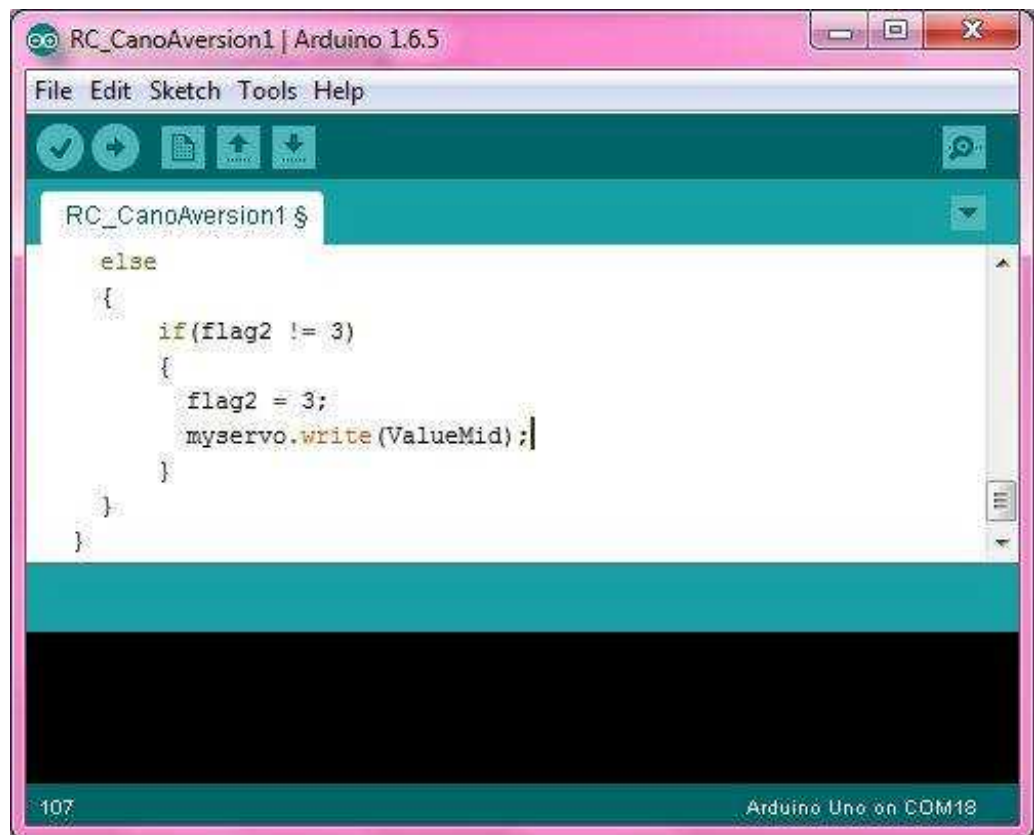
The image shows a screenshot of the Arduino IDE interface. The window title is "RC_CanoAversion1 | Arduino 1.6.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for saving, undo, redo, and uploading. The main text area contains the following C++ code snippet:

```
RC_CanoAversion1 §
else
if(toSend == 'R' || toSend == 'I' || toSend == 'J')
{
    if(flag2 != 2)
    {
        flag2 = 2;
        myservo.write(ValueRight);
    }
}
```

The status bar at the bottom left shows the line number "107" and the board name "Arduino Uno on CCM18".

Gambar 3.14 Perintah Untuk ke Kanan

b. Perintah untuk menggerakkan Servo ke tengah

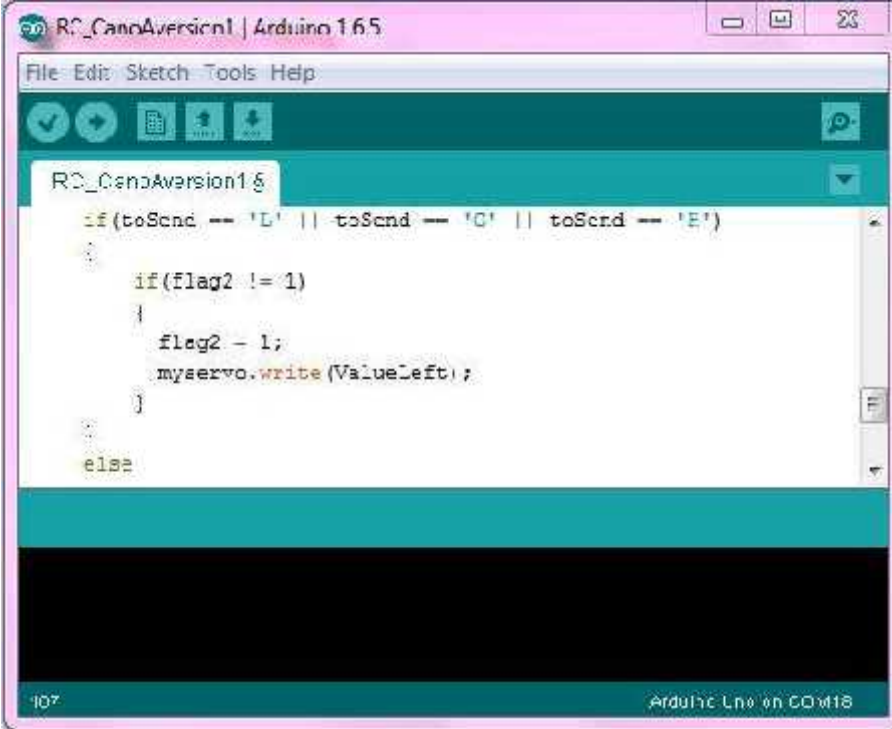
The image shows a screenshot of the Arduino IDE interface. The window title is "RC_CanoAversion1 | Arduino 1.6.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for check, run, upload, and download. The main editor area shows the following code snippet:

```
else
{
  if(flag2 != 3)
  {
    flag2 = 3;
    myservo.write(ValueMid);
  }
}
```

The status bar at the bottom left shows the line number "107" and the right side shows "Arduino Uno on COM18".

Gambar 3.15 Perintah Untuk Kembali ke Tengah

c. Perintah untuk menggerakkan Servo ke kiri



```
RC_CanoAversion1 | Arduino 1.6.5
File Edit Sketch Tools Help
RC_CanoAversion1.g
if(toSend == 'L' || toSend == '0' || toSend == 'E')
{
  if(flag2 != 1)
  {
    flag2 = 1;
    myservo.write(ValueLeft);
  }
}
else
```

Gambar 3.16 Perintah Untuk Meneggerakkan ke Kiri

BAB IV

PENGUJIAN DAN ANALISA

4.1 Tujuan Pengujian

Pengujian system ini memiliki tujuan untuk menguji kinerja serta hubungan antara perangkat keras dengan perangkat lunak sebagai program aplikasi system. Dengan pengujian ini dapat diketahui apakah alat dan aplikasi yang telah dirancang dapat bekerja sesuai dengan yang diinginkan atau tidak. Cara pengujian ini dilakukan dalam pengujian perangkat keras dan perangkat lunak.

4.2 Pengujian Input Output Sistem

Pengujian kali ini berguna untuk memetakan kembali setiap komponen dari sistem sesuai dengan fungsinya yakni sebagai input dan output. Tabel dibawah ini memuat setiap komponen utama yang digunakan berdasarkan fungsinya sebagai input atau output.

Tabel 4.1 Tabel Input

NO	INPUT	FUNGSI	KETERANGAN
1.	Arduino	Controller	Sebagai penghubung antara komponen
2.	Android Device	Remote	Pemberi perintah ke arduino
3.	Bluetooth HC-05	Master / Slave	Sebagai penerima sinyal dari smartphone
4.	Battery 1300mAh	Sumber Daya	Untuk bisa menggerakkan komponen – komponen

Tabel 4.1 merupakan tabel yang memuat komponen penting dalam sistem yang berfungsi sebagai Input atau masukan. Ketiga komponen pada tabel diatas

memiliki fungsi yang sangat penting dalam berjalannya sistem Penjaring Eceng gondok dan sampah ini.

Tabel 4.2 Tabel Output

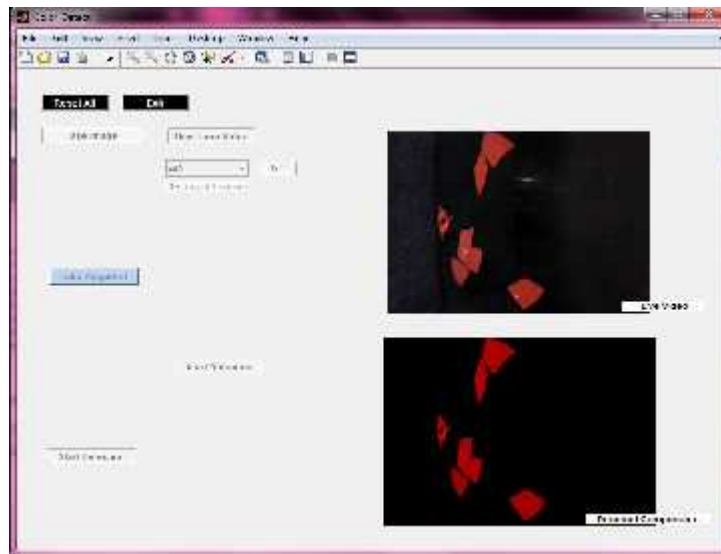
NO	OUTPUT	FUNGSI	KETERANGAN
1.	Pin 2	Bluetooth RXD	Pin Arduino yang terhubung dengan Bluetooth untuk menerima data dari smartphone.
2.	Pin 3	Bluetooth TXD	Pin Arduino yang terhubung dengan Bluetooth untuk mengirim data dari smartphone
3	Pin 4	Servo	Pin Arduino yang terhubung dengan Servo untuk bisa bergerak ke arah kiri / kanan
4.	Vin dan GRD	Battery	Pin Arduino yang terhubung dengan battery untuk Tegangan input ke board arduino ketika menggunakan tegangan dari luar dan sebagai jalur ground pada arduino

Tabel 4.2 merupakan tabel yang berisi pin – pin Arduino mana saja yang digunakan sebagai output beserta fungsinya.

4.3 Pengujian Aplikasi Matlab

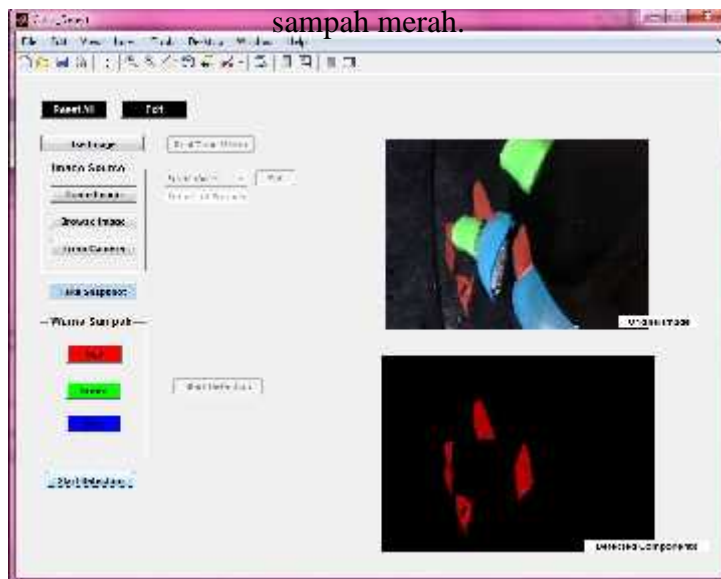
Pengujian kali ini bertujuan untuk menguji kerja dari aplikasi Matlab. Pengujian dilakukan untuk mengetahui apakah citra / gambar yang diambil sesuai dengan warna eceng gondok dan sampah yang ditentukan atau tidak.

a. Eceng gondok dan sampah Merah



Gambar 4.1 Deteksi Warna Eceng gondok dan sampah Merah

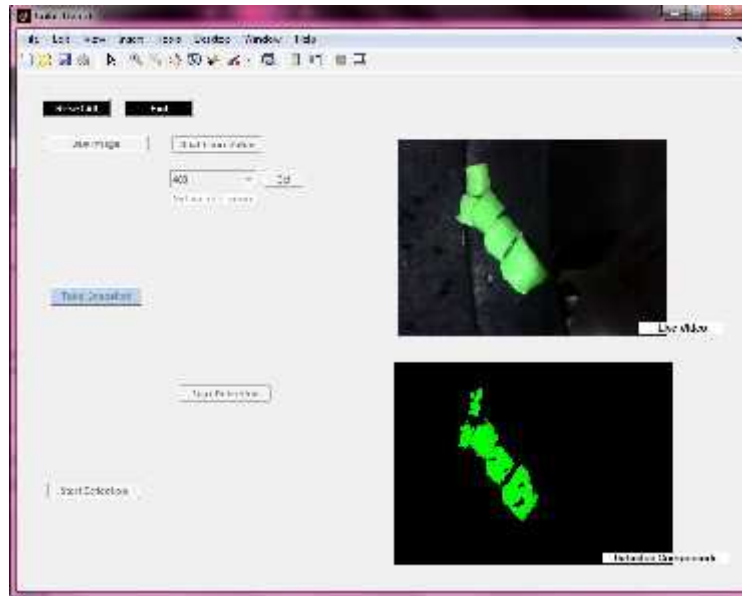
Pada gambar 4.1 terdeteksi oleh Real Time Video warna eceng gondok dan sampah merah.



Gambar 4.2 Deteksi Warna Eceng gondok dan sampah Merah versi 2

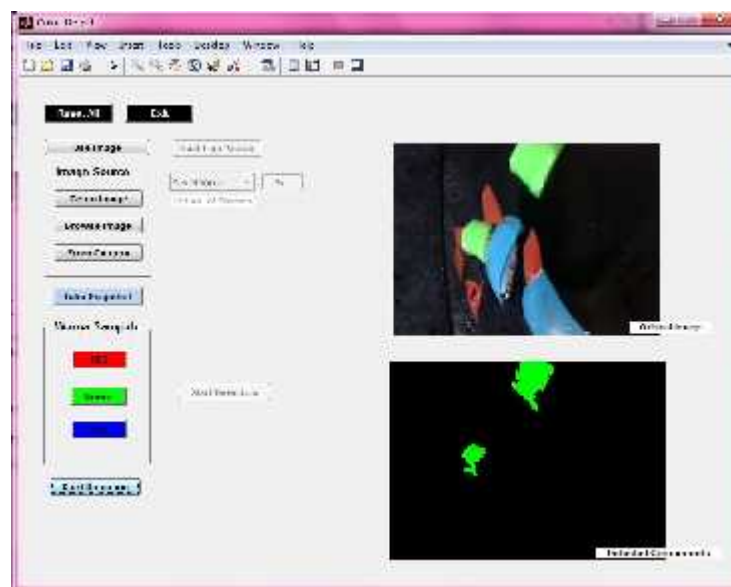
Penulis menggunakan 2 kali pengujian terhadap warna eceng gondok dan sampah merah dengan menambahkan beberapa warna lain sebagai objek dan didapatkan hasil seperti Gambar 4.2.

b. Eceng gondok dan sampah Hijau



Gambar 4.3 Deteksi Warna Eceng gondok dan sampah Hijau

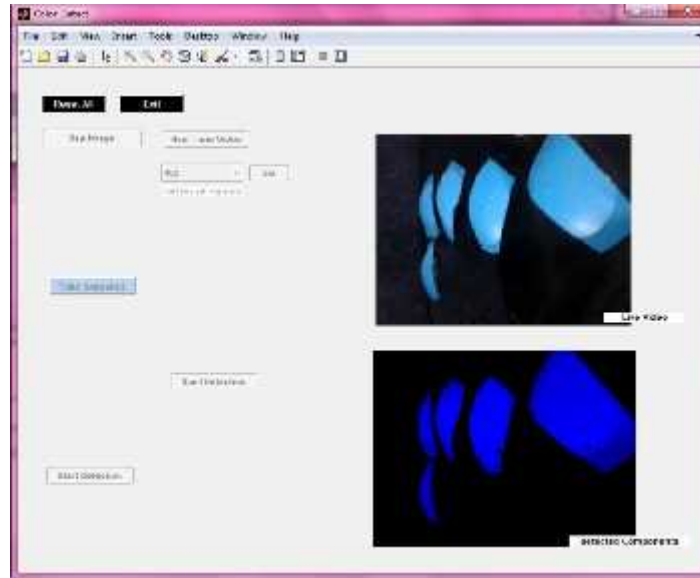
Pada gambar 4.2 terdeteksi oleh Real Time Video warna eceng gondok dan sampah hijau.



Gambar 4.4 Deteksi Warna Eceng gondok dan sampah Hijau versi 2

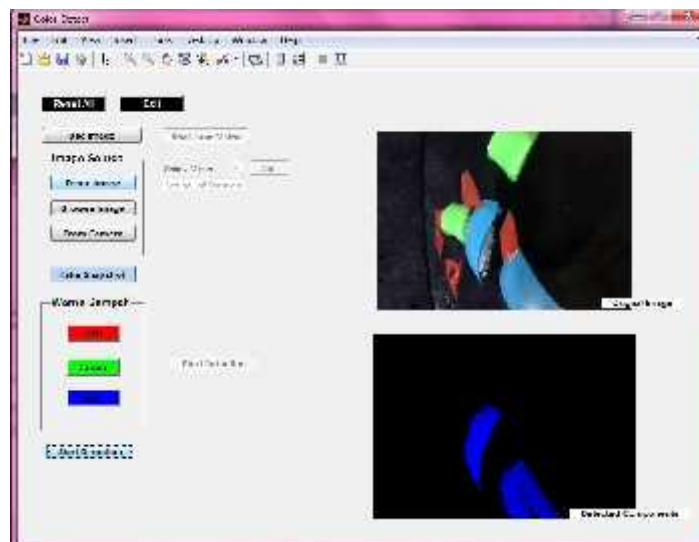
Sama halnya dengan Gambar 4.3 penulis melakukan 2 kali pengujian pada warna eceng gondok dan sampah hijau dan didapatkan hasil seperti Gambar 4.4.

c. Eceng gondok dan sampah Biru



Gambar 4.5 Deteksi Warna Eceng gondok dan sampah Biru

Pada gambar 4.5 terdeteksi oleh Real Time Video warna eceng gondok dan sampah biru.

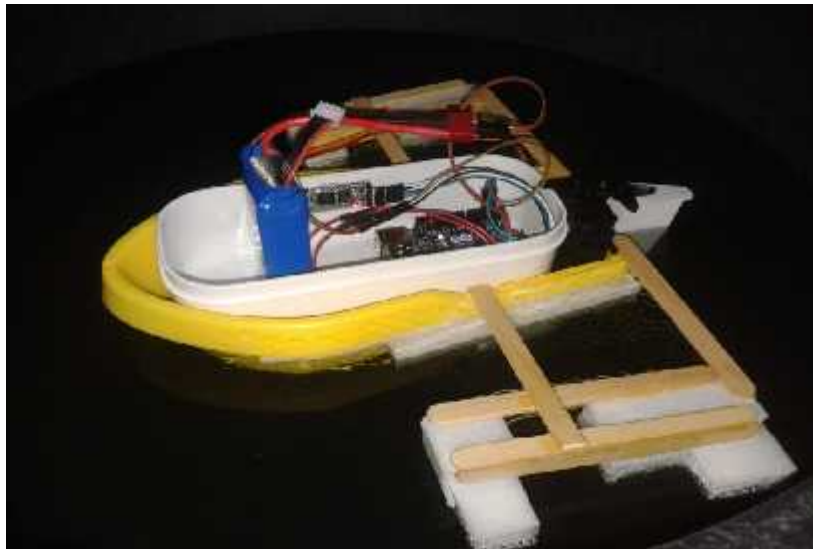


Gambar 4.6 Deteksi Warna Eceng gondok dan sampah Biru versi 2

Sama dengan warna eceng gondok dan sampah merah dan hijau, warna eceng gondok dan sampah biru juga menggunakan 2 kali pengujian dan didapatkan hasil seperti Gambar 4.6.

4.4 Pengujian Sistem Penjaring Eceng gondok dan sampah

Pada tahap ini merupakan tahap pengujian terakhir terhadap kinerja sistem penjaring sampah. Pada bagian ini akan diperlihatkan hasil uji coba Aplikasi terhadap Alat. Dimana dalam pengujian kali ini penulis menggunakan sebuah miniature perahu yang dilengkapi dengan output – output dari sistem yang telah dijelaskan tadi.



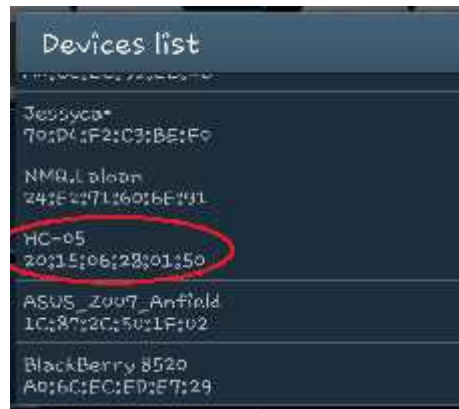
Gambar 4.7 Tampak Perahu Penjaring Eceng gondok dan sampah

Pada gambar 4.7 merupakan bentuk keseluruhan yang menjadi pusat kendali dari sistem ini, yaitu

1. Arduino Uno R3
2. Bluetooth HC-05
3. Battery Li-Po 1300mAh
4. Servo
5. Submarine DC Motor

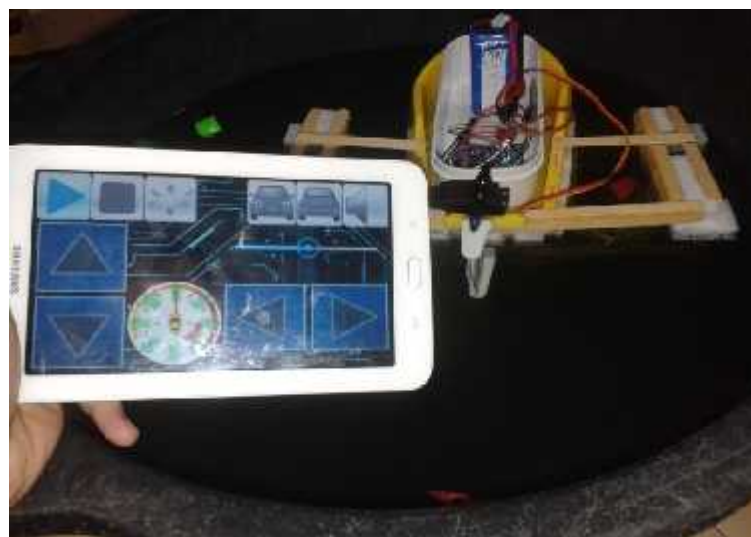
Dan pada gambar diatas terdapat seluruh komponen – komponen output yang digunakan pada sistem ini.

Sebelum memulai uji coba, penulis melakukan koneksi antara smartphone sebagai remote dengan Bluetooth HC-05 supaya smartphone dapat menggerakkan alat tersebut.



Gambar 4.8 Koneksi Dengan Bluetooth HC-05

Setelah Smartphone Android terkoneksi dengan Bluetooth HC-05 maka langkah selanjutnya tinggal mengontrol alat penjaring eceng gondok dan sampah tersebut. Disini penulis menggunakan aplikasi dari *Playstore* "Car Bluetooth RC", berikut aplikasi tersebut



Gambar 4.9 Tampilan Aplikasi "Car Bluetooth RC"

Pada aplikasi tersebut penulis dapat menggontol alat penjaring eceng gondok dan sampah tersebut ke kiri / kanan.

➤ Belok Kiri

Untuk dapat berbelok penulis menekan Button panah kiri sehingga Motor Servo bergerak ke kiri.



Gambar 4.10 Motor Servo Bergerak ke Kiri

➤ Belok Kanan

Untuk berbelok ke kanan penulis menekan Button panah kanan sehingga Motor Servo bergerak ke kanan.



Gambar 4.11 Motor Servo Bergerak ke Kanan

Setelah dapat berbelok ke kiri / kanan Servo secara otomatis akan kembali ke posisi awal (tengah), untuk koding program Motor Servo dapat dilihat pada gambar 3.13, 3.14, 3.15 dan untuk dapat bergerak maju penulis menggunakan Submarine Motor DC.



Gambar 4.12 Submarine Motor DC

BAB V

PENUTUP

5.1 Kesimpulan

Setelah melakukan perancangan dan pembuatan system pengujian beserta analisisnya, maka dapat diambil beberapa kesimpulan tentang kinerja dari keseluruhan system Penjaring eceng gondok dan sampah Menggunakan Citra adalah sebagai berikut:

1. Aplikasi Matlab dapat berkerja mendeteksi warna merah, hijau dan biru secara baik.
2. *Prototype* alat penjaring eceng gondok dan sampah di air dapat bekerja secara baik.
3. Aplikasi dapat berfungsi dengan baik untuk menampilkan serta mengontrol kamera Smartphone (IP Camera) secara live view.
4. Arduino dapat bekerja dengan baik dalam menerima, mengeksekusi dan mengirim feedback ke Android sesuai dengan perintah yang diberikan.
5. Jarak pengendalian Aplikasi terhadap Arduino kurang lebih 15 meter.
6. Kualitas IP Camara bergantung pada resolusi kamera smartphone.

5.2 Saran

Untuk pengembangan dan peningkatan lebih lanjut mengenai system Penjaring Eceng gondok dan sampah menggunakan citra, ada beberapa point yang perlu di perhatikan, antara lain:

1. Untuk optimalisasi kecepatan kapal penjaring, diharapkan menggunakan Motor DC dengan kecepatan RPM yang besar.
2. Dengan keterbatasan warna yang dapat dideteksi, penulis berharap adanya pengembangan dengan menambah lebih banyak warna yang dapat dideteksi oleh aplikasi. Dan aplikasi dapat langsung mengidentifikasi objek yang terdeteksi.

Besar harapan penulis system penjaring eceng gondok dan sampah ini dapat dikembangkan dengan baik sehingga dapat membantu banyak pihak terutama untuk kebersihan danau dan sungai.

DAFTAR PUSTAKA

- Istianto Eko Jazi**, 2014. *Pengantar Elektronika Dan Instrumen Pendekatan Project Arduino dan Android*. Penerbit CV ANDI OFFSET, Yogyakarta.
- Mikrokontrol**, 2005. *Arduino UNO*. <http://www.kelas-mikrokontrol.com/e-learning/mikrokontroler/pengantar-arduino.html>. (Diakses pada tanggal 23 Juli 2015).
- Pratt, Wiliam K**, 2007. *Digital Image Processing*, John Wiley & Sons, New Jersey.
- R. H. Sianipar**, 2014. *Pemograman Matlab Dalam Contoh Dan Peneran*. Penerbit Informarika. Bandung.
- Tuan Anh Do**, 2015. *How To Make a remote Control Boat by Arduino and Smartphone*. <https://www.youtube.com/watch?v=txXkEaWvFUc>. (Diakses pada tanggal 16 Juli 2015).
- Yuwono Marta Dinata**, 2015. *Arduino Itu Mudah*. Penerbit Alex Media Komputindo. Jakarta.

LAMPIRAN

1. Kode Program Matlab

```
function varargout = Color_Detect(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @Color_Detect_OpeningFcn, ...
                  'gui_OutputFcn', @Color_Detect_OutputFcn, ...
                  'gui_LayoutFcn', [], ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Color_Detect is made visible.
function Color_Detect_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Color_Detect (see VARARGIN)

% Choose default command line output for Color_Detect
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

set(handles.axes1,'Visible','off');
set(handles.axes2,'Visible','off');
set(handles.uipanel1,'visible','off');
set(handles.uipanel2,'visible','off');
set(handles.start,'Enable','off');
set(handles.snap,'Enable','off');
set(handles.oi,'visible','off');
set(handles.dc,'visible','off');
set(handles.uipanel3,'visible','off');
set(handles.startv,'enable','off');
set(handles.text,'enable','off');
set(handles.set,'enable','off');
set(handles.pop,'enable','off');
set(handles.text5,'visible','off');

% UIWAIT makes Color_Detect wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
```

```

function varargout = Color_Detect_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in image.
function image_Callback(hObject, eventdata, handles)
% hObject handle to image (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
set(handles.uipanel1,'visible','on');
set(handles.video,'Enable','off');

% --- Executes on button press in video.
function video_Callback(hObject, eventdata, handles)
% hObject handle to video (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
set(handles.image,'Enable','off');
set(handles.text,'enable','on');
set(handles.pop,'enable','on');

% --- Executes on button press in start.
function start_Callback(hObject, eventdata, handles)
% hObject handle to start (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
im=handles.im; x=handles.x;
switch x
case 1
axes(handles.axes2); cla;
im=handles.im;
r=im(:,1); g=im(:,2); b=im(:,3);
diff=imsubtract(r,rgb2gray(im));
bw=im2bw(diff,0.18);
area=bwareaopen(bw,300);
rm=immultiply(area,r); gm=g.*0; bm=b.*0;
image=cat(3,rm,gm,bm);
imshow(image);
% Transparcy part
[m n p]=size(im);
% mask=ones(m,n);
%i=find(image(:,:,1)==0);
% mask(i)=.9; % Change Transparacy
%imshow(im);
%hold on
%h=imshow(image);
%set(h,'AlphaData',mask);
case 2
axes(handles.axes2); cla;
im=handles.im;
r=im(:,1); g=im(:,2); b=im(:,3);
diff=imsubtract(g,rgb2gray(im));
bw=im2bw(diff,0.18);
area=bwareaopen(bw,300);

```

```

gm=immultiply(area,g); rm=r.*0; bm=b.*0;
image=cat(3,rm,gm,bm);
imshow(image);
case 3
    axes(handles.axes2); cla;
    im=handles.im;
    r=im(:,:,1); g=im(:,:,2); b=im(:,:,3);
    diff=imsubtract(b,rgb2gray(im));
    bw=im2bw(diff,0.18);
    area=bwareaopen(bw,300);
    bm=immultiply(area,b); gm=g.*0; rm=r.*0;
    image=cat(3,rm,gm,bm);
    imshow(image);
end
set(handles.dc,'visible','on');

% --- Executes on button press in red.
function red_Callback(hObject, eventdata, handles)
% hObject handle to red (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
handles.x=1;
guidata(hObject,handles);
set(handles.start,'enable','on');

% --- Executes on button press in green.
function green_Callback(hObject, eventdata, handles)
% hObject handle to green (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
handles.x=2;
guidata(hObject,handles);
set(handles.start,'enable','on');

% --- Executes on button press in blue.
function blue_Callback(hObject, eventdata, handles)
% hObject handle to blue (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
handles.x=3;
set(handles.start,'enable','on');
guidata(hObject,handles);

% --- Executes on button press in camera.
function camera_Callback(hObject, eventdata, handles)
% hObject handle to camera (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
imaqreset
info= imaqhwinfo;
name = char(info.InstalledAdaptors(end));
c_info = imaqhwinfo(name);
id = c_info.DeviceInfo.DeviceID(end);
format = char(c_info.DeviceInfo.SupportedFormats(end));
handles.vid= videoinput(name, id, format);
axes(handles.axes1);cla;
h = waitbar(0,'Please wait...');
steps = 200;
for step = 1:steps
    % computations take place here

```

```

    waitbar(step / steps)
end
close(h)

preview(handles.vid);
set(handles.snap,'Enable','on');
guidata(hObject,handles);
set(handles.uipanel2,'visible','on');
set(handles.oi,'visible','on');

% --- Executes on button press in browse.
function browse_Callback(hObject, eventdata, handles)
% hObject    handle to browse (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Browse the file from user
[file path]=uigetfile({'*.jpg';*.bmp;*.jpeg;*.png'}, 'Load Image File within Avilable Extensions');
image=[path file];
handles.file=image;
if (file==0)
    warndlg('You did not selected any file '); % fille is not selected
end
[filepath, fname, fext]=fileparts(file);
validex=({'*.bmp';*.jpg;*.jpeg;*.png'});
found=0;
for (x=1:length(validex))
    if (strcmpi(fext,validex{x}))
        found=1;
    end
end
handles.im=imread(image);
axes(handles.axes1); cla;
h = waitbar(0,'Please wait...');
steps = 100;
for step = 1:steps
    % computations take place here
    waitbar(step / steps)
end
close(h)
end
end
imshow(handles.im);
if (found==0)
    errordlg('Selected file does not match available extensions. Please select file from available extensions [ .jpg, .jpeg,
.bmp, .png] ', 'Image Format Error');
end
guidata(hObject, handles);
set(handles.uipanel2,'visible','on');
set(handles.oi,'visible','on');

% --- Executes on button press in demo.
function demo_Callback(hObject, eventdata, handles)
% hObject    handle to demo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
axes(handles.axes1); cla;
str = ('RGB Wheel','RGB Balls','RGB Cube','Color Boxes');
[Selection,ok] = listdlg('PromptString','Select an Image','SelectionMode','single','ListSize',[200 100],...
    'ListString',str);
if (ok==0)
    errordlg('You didnt select any image ','Selection Error');
end
end

```

```

switch Selection
    case 1
handles.im=imread('rgbwheel.jpg');
imshow(handles.im);
    case 2
        handles.im=imread('rgbballs.jpg');
imshow(handles.im);
    case 3
        handles.im=imread('rgbcube.jpg');
imshow(handles.im);
    case 4
        handles.im=imread('mcolor.jpg');
imshow(handles.im);
end
set(handles.uipanel2,'visible','on');
set(handles.oi,'visible','on');
guidata(hObject, handles);
%[s,v]=listdlg('PromptString','Select Image','SelectionMode','ListString',str);

```

```

% --- Executes on button press in snap.
function snap_Callback(hObject, eventdata, handles)
% hObject handle to snap (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
handles.im=getsnapshot(handles.vid);
axes(handles.axes1); cla; imshow(handles.im);
closepreview(handles.vid);
guidata(hObject,handles);

```

```

% --- Executes on button press in reset.
function reset_Callback(hObject, eventdata, handles)
% hObject handle to reset (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
axes(handles.axes1); cla;
axes(handles.axes2); cla;
set(handles.axes1,'Visible','off');
set(handles.axes2,'Visible','off');
set(handles.uipanel1,'visible','off');
set(handles.uipanel2,'visible','off');
set(handles.start,'Enable','off');
set(handles.snap,'Enable','off');
set(handles.oi,'visible','off');
set(handles.dc,'visible','off');
set(handles.video,'Enable','on');
set(handles.image,'Enable','on');
set(handles.uipanel3,'visible','off');
set(handles.startv,'enable','off');
set(handles.text,'enable','off');
set(handles.set,'enable','off');
set(handles.pop,'enable','off');
set(handles.text5,'visible','off');

```

```

% --- Executes on button press in exit.
function exit_Callback(hObject, eventdata, handles)
% hObject handle to exit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

close

% --- Executes on button press in set.
function set_Callback(hObject, eventdata, handles)
% hObject handle to set (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
handles.val=handles.frame;
guidata(hObject,handles);
set(handles.uipanel3,'visible','on');

% --- Executes on button press in startv.
function startv_Callback(hObject, eventdata, handles)
% hObject handle to startv (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
axes(handles.axes1); cla;
axes(handles.axes2); cla;
set(handles.axes1,'Visible','off');
set(handles.axes2,'Visible','off');
set(handles.uipanel1,'visible','off');
set(handles.uipanel2,'visible','off');
set(handles.start,'Enable','off');
set(handles.snap,'Enable','off');
set(handles.oi,'visible','off');
set(handles.dc,'visible','off');
set(handles.video,'Enable','off');
set(handles.image,'Enable','off');
set(handles.uipanel3,'visible','off');
set(handles.startv,'enable','off');
set(handles.text,'enable','off');
set(handles.set,'enable','off');
set(handles.pop,'enable','off');
set(handles.text5,'visible','off');
set(handles.dc,'visible','on');
set(handles.text5,'visible','on');
h = waitbar(0,'Please wait...');
steps = 200;
for step = 1:steps
    % computations take place here
    waitbar(step / steps)
end
close(h)
switch handles.cv
case 1
    imaqreset
    info= imaqhwinfo;
    name = char(info.InstalledAdaptors(end));
    c_info = imaqhwinfo(name);
    id = c_info.DeviceInfo.DeviceID(end);
    format = char(c_info.DeviceInfo.SupportedFormats(end));
    vid= videoinput(name, id, format);
    set(vid, 'FramesPerTrigger', Inf);
    set(vid, 'ReturnedColorspace', 'rgb')
    vid.FrameGrabInterval = 5;
    start(vid)
    % Set loop from the value of slider
    while(vid.FramesAcquired<=handles.val)
        % current snapshot
        im = getsnapshot(vid);

```



```

r=im(:,:,1); g=im(:,:,2); b=im(:,:,3);
diff=imsubtract(r,rgb2gray(im));
bw=im2bw(diff,0.18);
area=bwareaopen(bw,300);
rm=immultiply(area,r); gm=g.*0; bm=b.*0;
image=cat(3,rm,gm,bm);
axes(handles.axes2);
imshow(image);
axes(handles.axes1);
imshow(im);
end
stop(vid);
flushdata(vid);
clear all

```

case 2

```

imaqreset
info= imaqhwinfo;
name = char(info.InstalledAdaptors(end));
c_info = imaqhwinfo(name);
id = c_info.DeviceInfo.DeviceID(end);
format = char(c_info.DeviceInfo.SupportedFormats(end));
vid= videoinput(name, id, format);
set(vid, 'FramesPerTrigger', Inf);
set(vid, 'ReturnedColorspace', 'rgb')
vid.FrameGrabInterval = 5;
start(vid)
% Set loop from the value of slider
while(vid.FramesAcquired<=handles.val)
% current snapshot
im = getsnapshot(vid);
r=im(:,:,1); g=im(:,:,2); b=im(:,:,3);
diff=imsubtract(g,rgb2gray(im));
bw=im2bw(diff,0.18);
area=bwareaopen(bw,300);
gm=immultiply(area,g); rm=r.*0; bm=b.*0;
image=cat(3,rm,gm,bm);
axes(handles.axes2);
imshow(image);
axes(handles.axes1);
imshow(im);
end
stop(vid);
flushdata(vid);
clear all

```

case 3

```

imaqreset
info= imaqhwinfo;
name = char(info.InstalledAdaptors(end));
c_info = imaqhwinfo(name);
id = c_info.DeviceInfo.DeviceID(end);
format = char(c_info.DeviceInfo.SupportedFormats(end));
vid= videoinput(name, id, format);
set(vid, 'FramesPerTrigger', Inf);
set(vid, 'ReturnedColorspace', 'rgb')
vid.FrameGrabInterval = 5;
start(vid)
% Set loop from the value of slider
while(vid.FramesAcquired<=handles.val)
% current snapshot

```

```

        im = getsnapshot(vid);
        r=im(:,:,1); g=im(:,:,2); b=im(:,:,3);
        diff=imsubtract(b,rgb2gray(im));
        bw=im2bw(diff,0.18);
        area=bwareaopen(bw,300);
        bm=immultiply(area,b); gm=g.*0; rm=r.*0;
        image=cat(3,rm,gm,bm);
        axes(handles.axes2);
        imshow(image);
        axes(handles.axes1);
        imshow(im);
        end
        stop(vid);
        flushdata(vid);
        clear all
end

```

```

% --- Executes on button press in bluev.
function bluev_Callback(hObject, eventdata, handles)
% hObject handle to bluev (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
handles.cv=3;
guidata(hObject,handles);
set(handles.startv,'enable','on');

```

```

% --- Executes on button press in greenv.
function greenv_Callback(hObject, eventdata, handles)
% hObject handle to greenv (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
handles.cv=2;
guidata(hObject,handles);
set(handles.startv,'enable','on');

```

```

% --- Executes on button press in redv.
function redv_Callback(hObject, eventdata, handles)
% hObject handle to redv (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
handles.cv=1;
guidata(hObject,handles);
set(handles.startv,'enable','on');

```

```

% --- Executes on selection change in pop.
function pop_Callback(hObject, eventdata, handles)
% hObject handle to pop (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
val=get(hObject,'Value');
switch val
case 1
    warndlg('Please select a frame value');
    set(handles.set,'enable','off');
case 2
    handles.frame=25;
    set(handles.set,'enable','on');
case 3
    handles.frame=50;

```

```

    set(handles.set,'enable','on');
    case 4
    handles.frame=100;
    set(handles.set,'enable','on');
    case 5
    handles.frame=150;
    set(handles.set,'enable','on');
    case 6
    handles.frame=200;
    set(handles.set,'enable','on');
    case 7
    handles.frame=300;
    set(handles.set,'enable','on');
    case 8
    handles.frame=400;
    set(handles.set,'enable','on');
end
guidata(hObject,handles);
% Hints: contents = cellstr(get(hObject,'String')) returns pop contents as cell array
%     contents{get(hObject,'Value')} returns selected item from pop

% --- Executes during object creation, after setting all properties.
function pop_CreateFcn(hObject, eventdata, handles)
% hObject    handle to pop (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popmenu controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

2. Kode Program Arduino

```

#include <SoftwareSerial.h>
#include <Servo.h>

int bluetoothTx = 2;
int bluetoothRx = 3;

SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);

Servo myservo;
int ServoPort = 4;
int ValueLeft = 60;
int ValueMid = 30;
int ValueRight = 0;

void setup()
{
    //Setup Bluetooth serial connection to android

```

```

bluetooth.begin(115200);
bluetooth.print("$$$");
delay(100);
bluetooth.println("U,9600,N");
bluetooth.begin(9600);

pinMode( Motor1A, OUTPUT );
pinMode( Motor1B, OUTPUT );
digitalWrite( Motor1A, LOW );
digitalWrite( Motor1B, LOW );

pinMode( Motor2A, OUTPUT );
pinMode( Motor2B, OUTPUT );
digitalWrite( Motor2A, LOW );
digitalWrite( Motor2B, LOW );

myservo.attach(ServoPort);
myservo.write(ValueMid);
}

int flag1 = -1;
int flag2 = -1;

void loop()
{
  //Read from bluetooth and write to usb serial
  if(bluetooth.available())
  {
    char toSend = (char)bluetooth.read();
    bluetooth.write(toSend);
    if(toSend == 'S')
    {
      flag1 = 0;
      flag2 = 0;
      digitalWrite( Motor1A, LOW );
      analogWrite( Motor1B, LOW );

      digitalWrite( Motor2A, LOW );
      analogWrite( Motor2B, LOW );
    }
    if(toSend == 'F' || toSend == 'G' || toSend == 'I')
    {
      if(flag1 != 1)
      {
        flag1 = 1;
        digitalWrite( Motor1A, HIGH );
        analogWrite( Motor1B, 50 );

        //    digitalWrite( Motor2A, HIGH );
        //    analogWrite( Motor2B, 50 );
      }
    }
  }
}

```

```

if(toSend == 'B' || toSend == 'H' || toSend == 'J')
{
  if(flag1 != 2)
  {
    flag1 = 2;
    digitalWrite( Motor1B, HIGH );
    analogWrite( Motor1A, 50 );

//    digitalWrite( Motor2B, HIGH );
//    analogWrite( Motor2A, 50 );
  }
}

if(toSend == 'L' || toSend == 'G' || toSend == 'H')
{
  if(flag2 != 1)
  {
    flag2 = 1;
    myservo.write(ValueLeft);
  }
}
else
if(toSend == 'R' || toSend == 'I' || toSend == 'J')
{
  if(flag2 != 2)
  {
    flag2 = 2;
    myservo.write(ValueRight);
  }
}
else
{
  if(flag2 != 3)
  {
    flag2 = 3;
    myservo.write(ValueMid);
  }
}
}
}
}

```