

AN206 – Pan Tilt Color Tracking menggunakan HaViMo 2.0 dan DT-AVR Low Cost Micro System

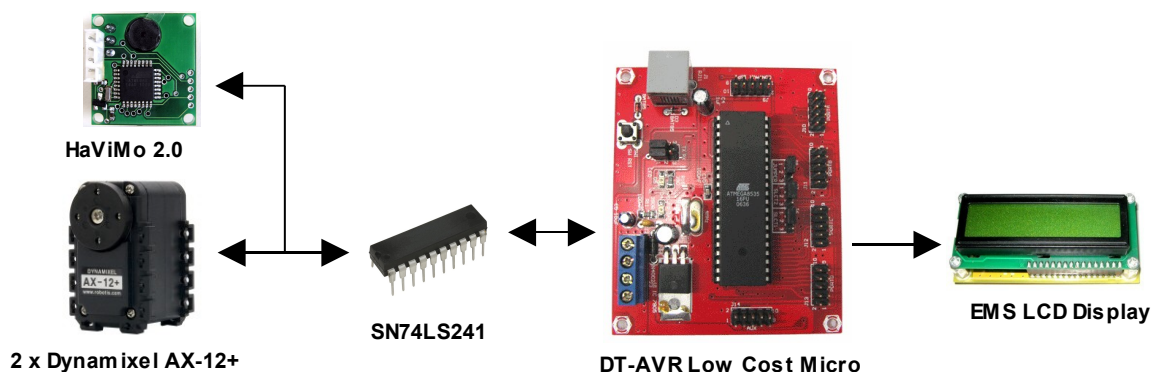
Oleh: Tim IE

Modul kamera pengolah citra belakangan ini semakin populer digunakan pada aplikasi robotika. Modul kamera tersebut biasa digunakan pada robot sepakbola *humanoid*, robot penjejak warna, kontes robot cerdas, dan aplikasi lain yang memerlukan pengolahan citra warna. Salah satu modul kamera yang cukup terkenal yaitu **HaViMo 2.0** (Hamid Vision Module versi 2.0) dari HaViSys. HaViMo 2.0 merupakan modul kamera CMOS yang terintegrasi dengan *chip processor* untuk melakukan *image processing* berbasis warna seperti *Region Growing* dan *Gridding* secara langsung sehingga pengguna bisa mendapatkan hasil pengolahan citra yang siap digunakan. HaViMo 2.0 memiliki format protokol yang sama dengan protokol dynamixel. Pada aplikasi ini telah dibuat suatu perangkat *pan tilt* kamera yang mampu mendeteksi dan mengikuti pergerakan bola dengan warna merah. Proses kalibrasi dan pemilihan warna yang akan dideteksi dilakukan terlebih dahulu melalui komputer dengan software HaViMoGUI. Sebagai aktuator *pan tilt* digunakan dua servo **Dynamixel AX12+** dan sebagai kontroler utama digunakan **DT-AVR Low Cost Micro System**. Sama dengan AN203, pada aplikasi ini digunakan protokol komunikasi standard dynamixel dan IC **74LS241** sebagai jembatan antarmuka *half duplex TTL* dengan *UART* mikrokontroler.

Berikut adalah perlengkapan yang diperlukan dalam aplikasi ini :

- 1x DT-AVR Low Cost Micro System (dengan Crystal 8MHz)
- 1x EMS LCD Display
- 2x AX-12+ servo
- 1x IC SN74LS241 dan 1x Resistor 10K Ohm.
- Bioloid Frame :
 - 1x FP04-F2
 - 3x FP04-F3
 - 1x FP04-F9
- Power Supply +12VDC dengan kemampuan arus hingga 3A.
- USB2Dynamixel (untuk kalibrasi HaViMo)
- Bola plastik atau target lainnya yang cukup besar dan berwarna merah.
- Beberapa kabel jumper dan 2x kabel servo konektor 3 pin Dynamixel.
- Mur dan baut secukupnya.

Adapun blok diagram dari AN206 adalah sebagai berikut.



Gambar 1
Blok Diagram AN206

Hubungan antar modul adalah sebagai berikut :

DT-AVR Low Cost Micro System	SN74LS241
GND (J13 pin 1)	GND (Pin 10)
VCC (J13 pin 2)	VCC (Pin 20)
PORTD.0/RXD (J13 Pin 3) **	1Y4 (Pin 12)***
PORTD.1/TXD (J13 Pin 4) **	2A1 (Pin 11)***
PORTD.7/Direction Control (J13 Pin 10)*	1G (Pin 1)**
	2G (Pin 19)**

*) Pin tidak mutlak, dapat diganti dengan pin lain dengan penyesuaian program

**) Pin ini mutlak, tidak dapat diganti dengan pin yang lain.

***) Pin tidak mutlak, dapat diganti dengan pin lain yang sesuai fungsinya.

Tabel 1
Hubungan DT-AVR LCMS dengan SN74LS241

SN74LS241	AX-12+	HaViMo 2.0
GND (Pin 10)	GND	GND
1A4 (Pin 8)***	DATA (dengan <i>pull up</i> eksternal Resistor 10K Ohm)	DATA (menggunakan jalur data yang sama dengan AX-12+)
2Y1 (Pin 9)***		

***) Pin tidak mutlak, dapat diganti dengan pin lain yang sesuai fungsinya.

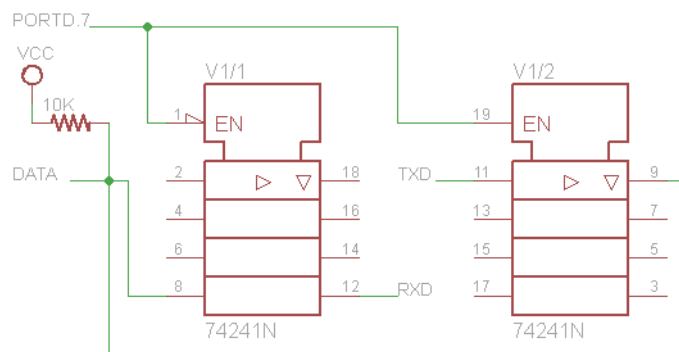
Tabel 2
Hubungan SN74LS241 dengan AX-12+

DT-AVR Low Cost Micro System	EMS LCD Display
GND (J10 pin 1)	GND (J3 pin 1)
VCC (J10 pin 2)	+5V (J3 pin 2)
PORTA.0 (J10 Pin 3)*	RS (J3 pin 3)
PORTA.1 (J10 Pin 4)*	R/W (J3 pin 4)
PORTA.2 (J10 Pin 5)*	E (J3 pin 5)
PORTA.3 (J10 Pin 6)*	BL (J3 pin 6)
PORTA.4 (J10 Pin 7)*	DB4 (J3 pin 7)
PORTA.5 (J10 Pin 8)*	DB5 (J3 pin 8)
PORTA.6 (J10 Pin 9)*	DB6 (J3 pin 9)
PORTA.7 (J10 Pin 10)*	DB7 (J3 pin 10)

*) Pin tidak mutlak, dapat diganti dengan pin lain dengan penyesuaian program

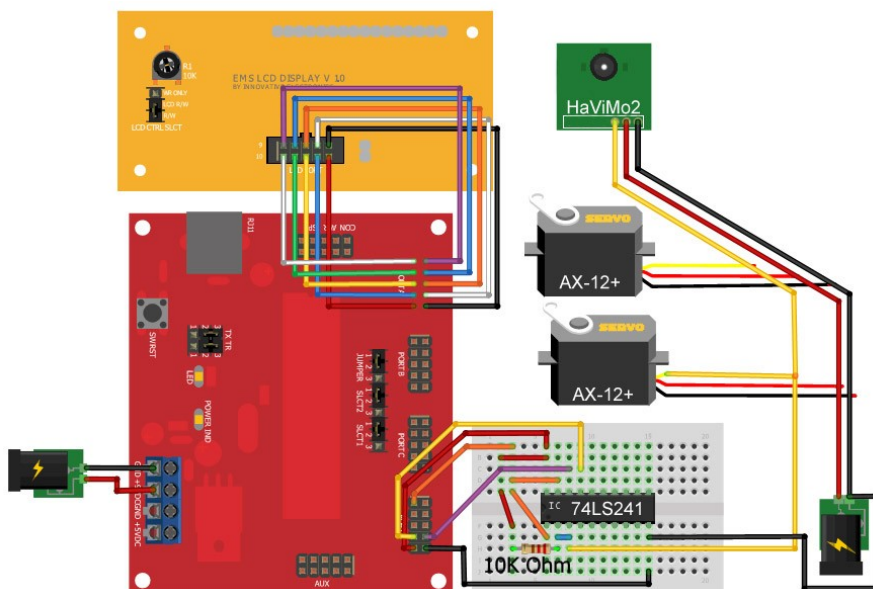
Tabel 3
Hubungan DT-AVR LCMS dengan EMS LCD Display

Skema koneksi SN74LS241 dengan modul-modul yang lain adalah sebagai berikut :



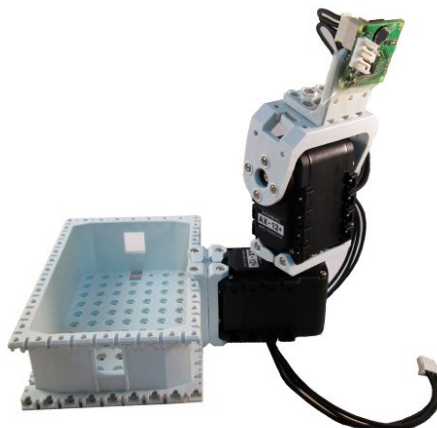
Gambar 2
Skema koneksi SN74LS241 dengan AX-12+ dan DT-AVR LCMS

Ilustrasi koneksi antar modul pada AN206 ini adalah sebagai berikut :



Gambar 3
Ilustrasi AN203

Bentuk konfigurasi *pan tilt* menggunakan AX-12+ dan Bioloid frame adalah sebagai berikut :



Gambar 4
Konfigurasi *pan tilt* servo

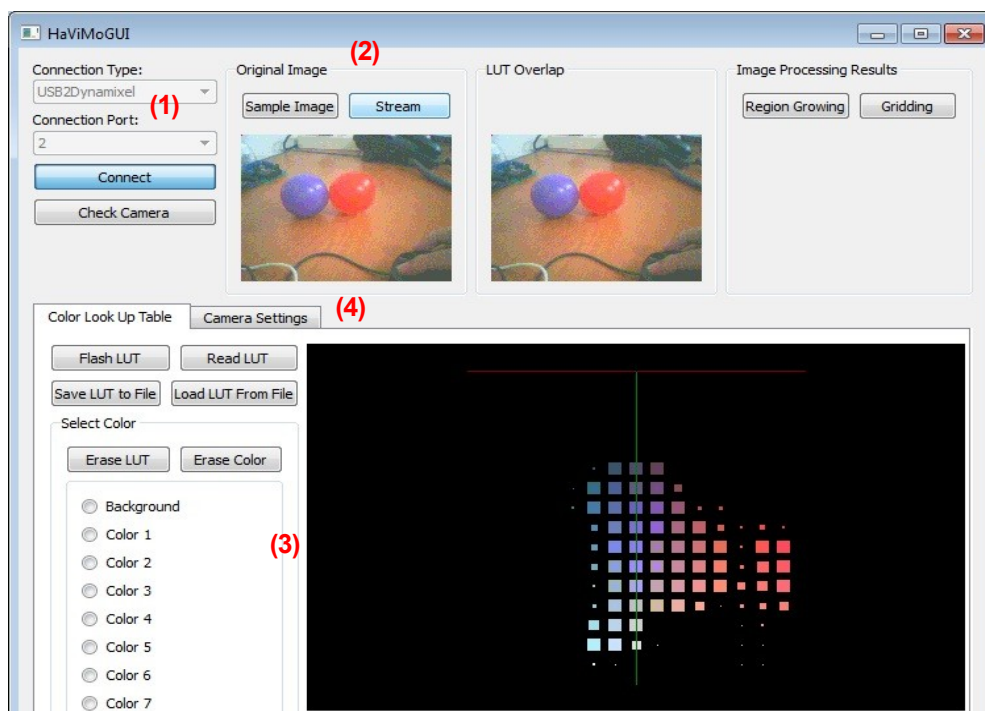
Modul – modul diatas perlu dikonfigurasi terlebih dahulu agar dapat bekerja sesuai dengan yang diharapkan. Berikut ini adalah langkah – langkah konfigurasi yang perlu dilakukan :

- DT-AVR Low Cost Micro System
 - Mikrokontroler ATmega8535 pada DT-AVR Low Cost Micro System menggunakan osilator eksternal berupa crystal dengan nilai frekuensi 4 Mhz. Untuk mendukung baudrate komunikas 1Mbps maka diperlukan penggantian *Crystal Oscillator* dengan frekuensi 8MHz atau 16MHz, pada AN ini digunakan XTAL 8MHz. Silahkan melakukan pengaturan *fusebit* pada ATmega8535 dahulu agar dapat bekerja dengan kristal eksternal 8MHz. Informasi lebih detil mengenai pengaturan fusebit terdapat pada AN177.
 - Komunikasi antara ATmega8535 dengan AX-12+ dan HaViMo 2.0 dilakukan menggunakan komunikasi serial UART TTL. Maka dari itu perlu dilakukan pengaturan jumper J4 dan J5 pada posisi 2-3.

1	2	3	J4
1	2	3	J5

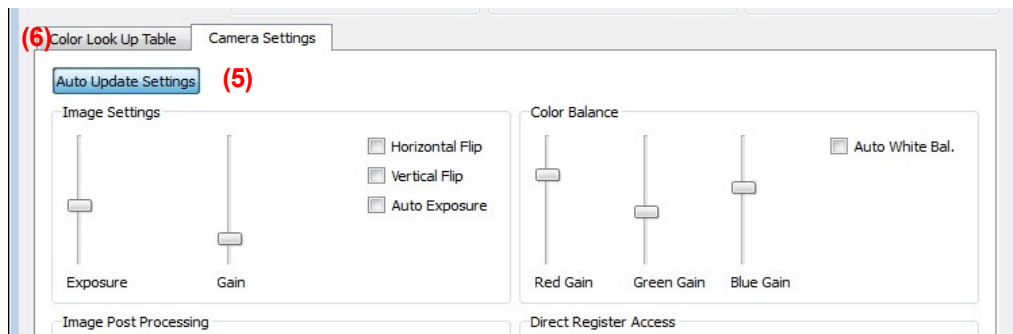
Gambar 5
Pengaturan Jumper J4 dan J5 pada DT-AVR Low Cost Micro System

- HaViMo 2.0
 - Untuk bisa mengenali warna-warna tertentu, maka modul HaViMo 2.0 perlu terlebih dahulu dikalibrasi dan diprogram melalui software komputer HaViMoGUI.
 - Langkah-langkah pemilihan warna yang akan dideteksi adalah sebagai berikut :
 1. Hubungkan Modul HaViMo dengan USB2Dynamixel, pastikan *switch* selektor pada USB2Dynamixel telah berada pada posisi *TTL*.
 2. Berikan catu daya pada Modul HaViMo 2.0.
 3. Tancapkan USB2Dynamixel pada port USB komputer (diasumsikan driver telah terinstal).
 4. Buka program HaViMoGUI.
 5. Pada *Connection Type*, pilih USB2Dynamixel (1).
 6. Pada *Connection Port*, pilih port COM serial dari USB2Dynamixel.
 7. Klik *Connect*, apabila berhasil maka pada sudut kiri bawah dari HaViMoGUI akan ada keterangan status “Port Open”
 8. Kemudian klik *Check Camera*, apabila berhasil maka akan ada keterangan status “HaViMo Found”



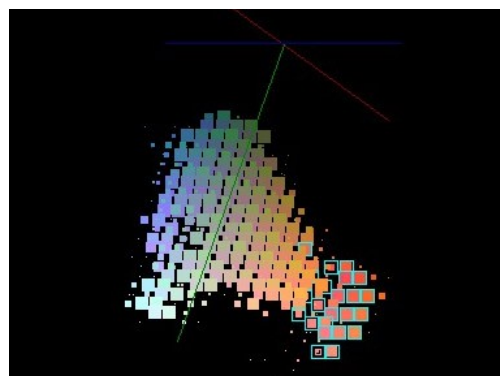
Gambar 6
Tampilan HaViMoGUI

9. Persiapkan kondisi target/bola warna merah, kondisi pencahayaan lingkungan, serta latar belakang target, hal ini penting karena proses kalibrasi warna untuk *tracking* akan dilakukan.
10. Arahkan modul kamera HaViMo kepada target pada jarak yang diinginkan.
11. Kik *Sample Image*, maka akan didapatkan gambar *Original Image (2)* serta Grafik Warna 3D pada halaman *Color Look Up Table (3)*.
12. Langkah selanjutnya yang perlu dilakukan adalah menyesuaikan parameter pada tab *Camera Settings (4)*. Pada Tab *Camera Settings* dapat dilakukan pengaturan standard hasil capture dari HaViMo 2.0 sebelum dilakukan *Image Processing*. Hal-hal yang perlu diperhatikan pada tahap ini antara lain, kerataan pencahayaan, keseimbangan warna, dan tingkat kontras. Tahapan ini termasuk bagian sangat penting, karena dengan kualitas gambar yang baik dan merata akan mempermudah proses selanjutnya yaitu pemilihan warna yang akan dideteksi.



Gambar 7
Tampilan *Camera Settings* HaViMoGUI

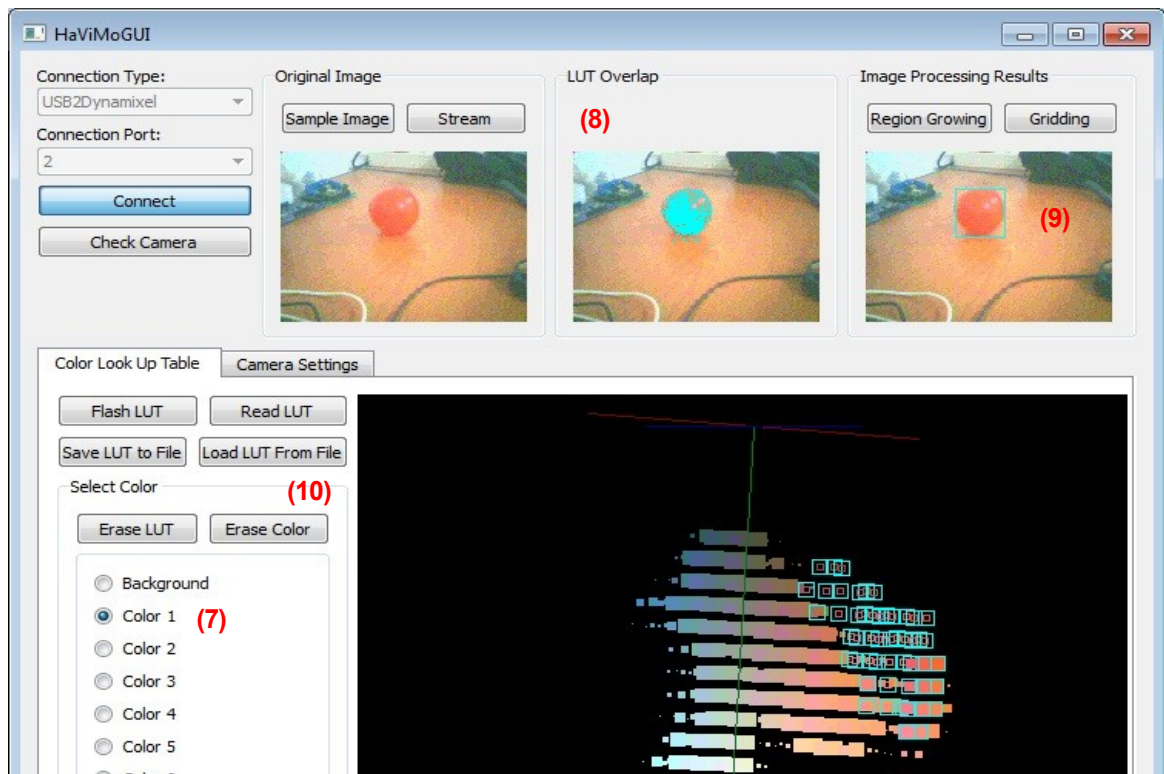
13. Pada *box Original Image (2)* Klik *Stream* agar gambar ditampilkan secara update terus menerus.
14. Klik *Auto Update Settings* untuk mulai mengatur berbagai macam *setting*-nya (5). Setelah selesai Klik kembali *Auto Update Settings* untuk menghentikan fungsi pengaturan secara langsung tersebut dan Klik *Stream (2)* sekali lagi untuk menghentikan proses *streaming image*.
15. Langkah selanjutnya yaitu pemilihan warna yang akan dideteksi. Kembalikan tampilan pada Tab *Color Look Up Table (6)*. Klik *Sample Image (2)* untuk mendapatkan hasil terbaru dari gambar setelah proses *Camera Settings*.
16. Klik *Erase LUT* dan *Erase Color* untuk menghapus setting sebelumnya (jika ada) pada HaViMo 2.0.
17. Klik pada *radio button Color 1 (7)*, maka saat ini kita telah siap memilih warna mana yang akan dideteksi oleh HaViMo.



Gambar 8
Tampilan Grafik Warna 3D HaViMoGUI

18. Klik pada warna-warna pada Grafik Warna 3D yang mirip dengan warna bola pada tampilan *Original Image*. Secara otomatis tampilan Citra pada *LUT Overlap (8)* akan menunjukkan warna mana yang kita pilih, dengan tanda warna hijau cerah. Semakin banyak warna hijau cerah yang terlihat pada badan bola, maka itu berarti semakin banyak warna merah bola telah masuk dalam *Look Up Table*.
19. Grafik Warna 3D (3) dapat diputar-putar untuk melihat dan memilih warna-warna yang sesuai agar nilai *LUT* semakin baik.

20. Jika dirasa sudah cukup baik, maka LUT yang sudah tersusun dapat disimpan dengan cara Klik *Save LUT to file* (10).
21. Langkah selanjutnya adalah menuliskan *LUT* yang telah disusun ke HaViMo 2.0, hal ini dapat dilakukan dengan cara langsung dengan Klik *Flash LUT* (10) apabila *LUT* yang dimaksud telah/sedang dibuka. Apabila *LUT* yang diinginkan masih tersimpan, maka dapat dibuka terlebih dahulu dengan *Load LUT From File* (10).
22. Setelah proses *Flash LUT* selesai, klik *Sample Image* sekali lagi, Klik *Read LUT*, kemudian klik *Region Growing*. Apabila nilai *LUT* untuk deteksi bola merah sudah baik, maka posisi bola akan bisa dideteksi dengan tanda bingkai persegi empat berwarna hijau muda (9).



Gambar 8
Tampilan HaViMoGUI sukses mendeteksi *object*.

Setelah menghubungkan modul-modul tersebut menggunakan kabel jumper, lakukan pengecekan kembali menggunakan *multimeter* untuk memastikan koneksi antar modul telah terhubung dengan baik. Pastikan juga bahwa tidak terjadi hubungan singkat antara VCC dan GND sebelum memberikan catu daya.

Apabila konfigurasi di atas telah selesai, lakukan langkah – langkah berikut ini :

1. Berikan catu daya +9V DC s.d. +12V DC pada terminal biru J2 DT-AVR LCMS (**perhatikan polaritas catu daya, Jika polaritas terbalik dapat menyebabkan kerusakan modul**).
2. Hubungkan *programmer* mikrokontroler AVR yang mendukung fitur ISP dengan DT-AVR Low Cost Micro System, seperti DT-HiQ AVR In System Programmer, DT-HiQ AVR USB ISP, atau *programmer* lainnya.
3. *Download* file dengan ekstensi *.hex* (TrackingColor.hex) yang berada di dalam folder (AN206\TrackingColor\Release) ke DT-AVR Low Cost Micro System.
4. Dikarenakan 2 buah Dynamixel AX-12+ dan HaViMo 2.0 telah terhubung dalam satu set mekanik *pan tilt* maka catu daya cukup diberikan pada salah satu konektor 3 pin dari ketiga modul tersebut.
5. Berikan input catu daya untuk servo AX-12+ melalui konektor 3 pin yang ada pada *body* servo. Catu daya disarankan memiliki spesifikasi tegangan yang stabil dan kemampuan *supply* arus yang cukup. Tegangan Operasional AX-12+ adalah rentang 9V s.d. 12V.
6. Pada aplikasi ini digunakan Switching Power Supply DVE dengan output 12V dan kemampuan arus hingga 5A, konektor Jack DC terhubung dengan SMPS2Dynamixel.
7. Konsumsi Arus 1 buah AX-12+ saat standby sekitar 50mA dan saat beroperasi mengkonsumsi arus maksimal hingga 900mA.

Protokol komunikasi Dynamixel

Protokol komunikasi pada Dynamixel Ax-12+ terbagi menjadi dua jenis yaitu data yang dikirimkan dari kontroler ke servo (*Instruction Packet*) dan data respon yang dikirim balik oleh servo kepada kontroler (*Status packet*). Keterangan lengkap mengenai paket *Instruction*, *Status*, dan *Parameter* dapat dilihat pada manual dari produk AX-12+. Kedua paket data tersebut memiliki format sebagai berikut.

● Instruction Packet

Merupakan paket yang dikirimkan oleh kontroler utama kepada servo untuk mengirimkan perintah-perintah. Struktur dari *Instruction packet* adalah sebagai berikut:



Keterangan :

- 0xFF 0xFF** : Merupakan indikator awal dari sebuah paket.
ID : Merupakan nomor ID/Identitas unique dari servo dynamixel.
Terdapat 254 nilai ID yang tersedia mulai dari 0x00 sampai dengan 0xFD. Ada nomor ID khusus yang disebut broadcasting ID yaitu nomor 0xFE. Paket yang dikirimkan dengan ID Broadcasting akan diterima dan dilaksanakan oleh semua servo yang terhubung. Paket yang terkirim dengan Broadcasting ID tidak akan menghasilkan Respon Status Packet.
- LENGTH** : Merupakan panjang dari paket dimana nilainya adalah "Jumlah parameter(N) + 2".
INSTRUCTION : Instruksi untuk aktuator Dynamixel yang akan dilakukan.
PARAMETER0...N : Digunakan jika ada informasi tambahan yang butuh untuk dikirimkan selain dari Instruksi itu sendiri.
CHECK SUM : Merupakan data yang dihitung sebagai error checking. Metode penghitungan check sum adalah sebagai berikut :

$$\text{Check Sum} = \sim(\text{ID} + \text{Length} + \text{Instruction} + \text{Parameter1} + \dots + \text{Parameter N})$$

Jika nilai yang terhitung lebih besar dari 255, maka byte terbawahlah yang ditentukan sebagai nilai check sum.

~ mewakili operasi logic NOT.

● Status Packet/Return Packet

Merupakan paket respon dari servo kepada kontroler utama setelah menerima sebuah *Instruction Packet*. Struktur dari *Status Packet* adalah sebagai berikut.



Keterangan :

- 0xFF 0xFF** : Merupakan indikator awal dari sebuah paket.
ID : Merupakan nomor ID/Identitas unique dari servo dynamixel.
Terdapat 254 nilai ID yang tersedia mulai dari 0x00 sampai dengan 0xFD. Ada nomor ID khusus yang disebut broadcasting ID yaitu nomor 0xFE. Paket yang dikirimkan dengan ID Broadcasting akan diterima dan dilaksanakan oleh semua servo yang terhubung. Paket yang terkirim dengan Broadcasting ID tidak akan menghasilkan Respon Status Packet.
- LENGTH** : Merupakan panjang dari paket dimana nilainya adalah "Jumlah parameter(N) + 2".
ERROR : Merupakan *Byte* yang merepresentasikan error yang dikirimkan oleh Servo.
INSTRUCTION : Instruksi untuk aktuator Dynamixel yang akan dilakukan.
PARAMETER0...N : Digunakan jika ada informasi tambahan yang butuh untuk dikirimkan selain dari Instruksi itu sendiri.
CHECK SUM : Merupakan yang dihitung sebagai error checking. Metode penghitungan 'Check Sum' adalah sebagai berikut :

$$\text{Check Sum} = \sim(\text{ID} + \text{Length} + \text{Instruction} + \text{Parameter1} + \dots + \text{Parameter N})$$

Jika nilai yang terhitung lebih besar dari 255, maka byte terbawahlah yang ditentukan sebagai nilai checksum.

~ mewakili operasi logic NOT.

Instruksi HaViMo 2.0

Struktur protokol dan hardware komunikasi HaViMo 2.0 dengan protokol Dynamixel adalah sama, namun ada beberapa hal yang berbeda pada alamat ID dan alamat *register*. Alamat ID HaViMo 2.0 telah diatur secara permanen pada ID 100 (desimal) atau 0x64 (heksadesimal). HaViMo 2.0 juga memiliki daftar set instruksi-instruksi khusus berkaitan dengan pengaturan pengolahan citra yang hanya dapat digunakan dengan HaViMo 2.0. Instruksi-instruksi tersebut antara lain :

Instruction	Value	No. of Parameter	Function
PING	0x01	0	No action. Used for obtaining a Status Packet
READ_REGION	0x02	2	Read Results of Region Detection
WRITE	0x03	2	Equivalent to CAP_REGION for Compatibility
READ_REG	0x0C	2	Read Camera Chip Registers
WRITE_REG	0x0D	2	Write Camera Chip Registers (1)
CAP_REGION	0x0E	0	Capture and Find Color Regions (1)
RAW_SAMPLE	0x0F	0	Sample the Raw Image (used by GUI) (2)
LUT_MANAGE	0x10	0	Enter LUT Manage Mode (used by GUI) (2)
RD_FILTHR	0x11	2	Read Noise Filter Thresholds
WR_FILTHR	0x12	2	Write Noise Filter Thresholds (1)
RD_REGTHR	0x13	2	Read Region Filter Thresholds
WR_REGTHR	0x14	2	Write Region Filter Thresholds (1)
CAP_GRID	0x15	0	Capture and Compress using Gridding algorithm (1)
READ_GRID	0x16	2	Read Results of the Gridding Algorithm (3)
SAMPLE_FAST	0x17	0	Fast Sample (used by GUI) (2) (4)

- 1) Tidak ada return paket yang dihasilkan dari Instruksi ini.
- 2) Respon berbeda dengan standar paket *ROBOTIS* maupun *RoboBuilder*.
- 3) Address yang diberikan dikali dengan 16 secara internal.
- 4) Tidak didukung pada mode *RoboBuilder*.

Tabel 4
Daftar Instruksi HaViMo 2.0

Pengolahan citra dengan *Region Growing*

Instruksi yang digunakan untuk melacak posisi warna pada Aplikasi ini adalah CAP_REGION dan READ_REGION. CAP REGION berfungsi untuk menangkap suatu citra untuk diproses menggunakan teknik *Region Growing*. Hasil dari proses *Region Growing* ini dibaca dengan menggunakan instruksi READ_REGION. Pada AN206 ini proses pembacaan register ini telah dimasukkan dalam *library* Dynamixel_Havimo.

Hasil pengolahan Region Growing memiliki format sebagai berikut :

Index	Color	Pixels	SumX				SumY				MaxX	MinX	MaxY	MinY
XX*	XX	XX XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX

*) seluruh tipe data dalam heksadesimal

Tabel 5
Format Hasil Region Growing

Hasil pembagian antara 'jumlah piksel-piksel yang terdeteksi dalam *region*' (**Pixels**) dengan 'jumlah dari koordinat X pada piksel-piksel yang terdeteksi' (**SumX**) akan menghasilkan nilai 'titik tengah' (*center point*) dari koordinat sumbu X terhadap kotak *region*, begitu pula untuk pembagian **SumY** dengan **Pixels** akan menghasilkan koordinat titik tengah sumbu Y terhadap kotak *region*.

Posisi obyek (x,y) terhadap kamera HaViMo 2.0, diperoleh melalui perhitungan sebagai berikut :

Posisi obyek pada sumbu X = **MinX + (SumX / Pixels)**

Posisi obyek pada sumbu Y = **MinY + (SumY / Pixels)**

Nilai batas sisi kiri kotak *region* (**MinX**) dan nilai batas sisi atas kotak *region* (**MinY**) perlu ditambahkan untuk memperoleh posisi sebenarnya dari obyek terhadap keseluruhan citra.

Library Dynamixel_Havimo

Kumpulan dari *instruction-instruction* Dynamixel servo dan kamera HaViMo yang telah dirangkum kedalam library program dengan nama *Dynamixel_Havimo.c* dan file header *Dynamixel_Havimo.h*. Library ini disusun berdasarkan *library* Arduino *Dynamixel.cpp* dan *Dynamixel.h* yang dibuat oleh Josué Alejandro Savage.

Beberapa fungsi-fungsi yang ada didalamnya sebagai contoh :

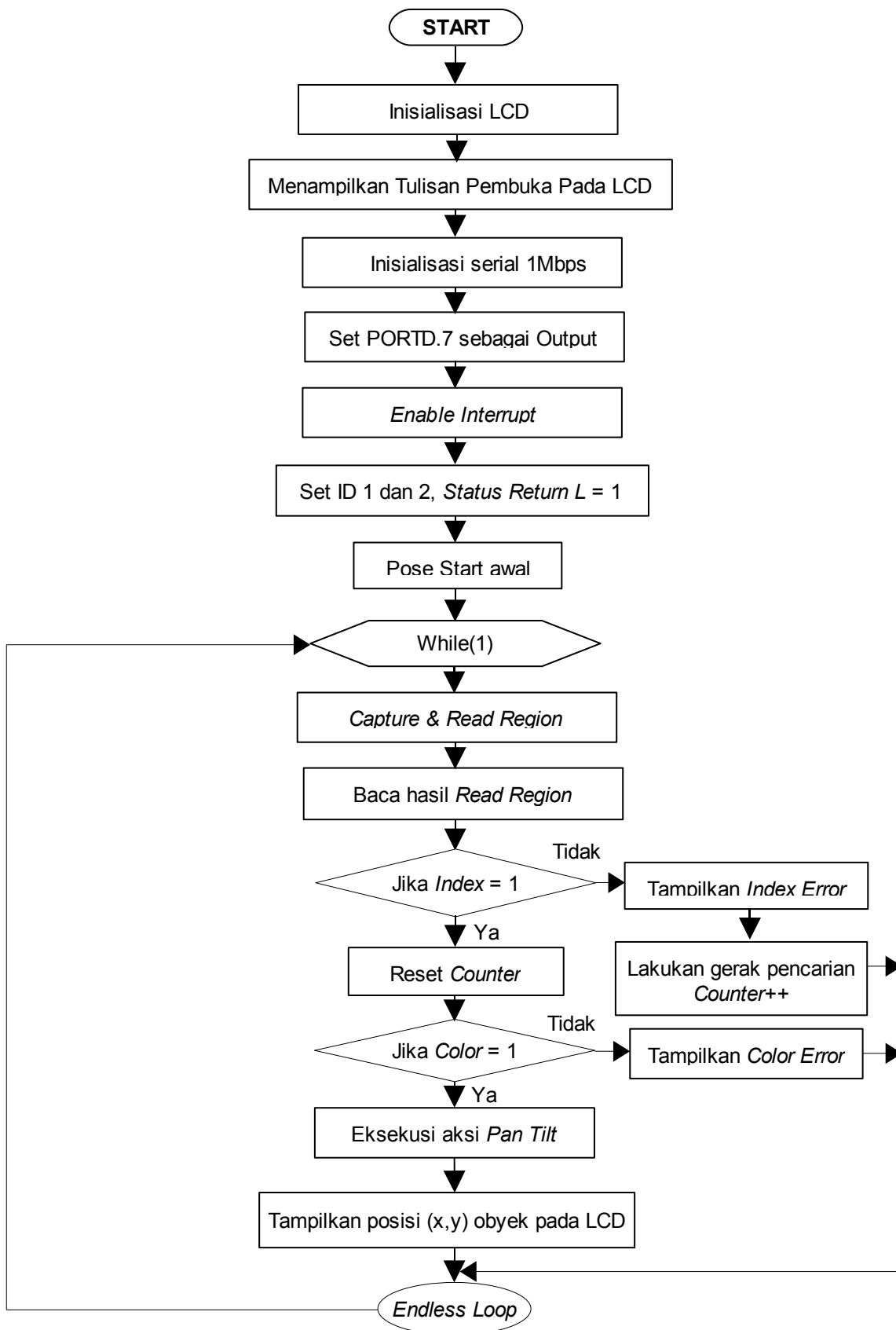
- **Dxl_readPosition()**
Deskripsi : Membaca posisi servo pada saat itu.
Sintaks : `Dxl_readPosition(ID);`
Parameter: ID – Nomor identitas unik yang dimiliki oleh servo AX-12+ Dynamixel.
Contoh :
`Dxl_readPosition(1);`
Akan mengirimkan perintah untuk membaca posisi servo dengan ID = 1.
Respon Status Packet :
Berisi nilai posisi servo dalam format integer.
- **Hvm_CapRegion()**
Deskripsi : Menangkap gambar dan kemudian dilakukan *Image Processing Region Growing*.
Sintaks : `Hvm_CapRegion();`
Parameter: -
Contoh :
`Hvm_CapRegion();`
Akan mengirimkan perintah untuk menangkap gambar dan melakukan *Region Growing*.
Respon Status Packet :
Command ini tidak memberikan respon status apapun.
- **Hvm_ReadRegion()**
Deskripsi : Membaca hasil dari *Region Growing*.
Sintaks : `Hvm_ReadRegion(unsigned char reg_address);`
Parameter: Register address hasil pembacaan pada alamat 0x10 s.d. 0xFFHeksa atau 16 s.d. 31desimal.
Contoh :
`Hvm_ReadRegion(16);`
Akan mengirimkan perintah untuk membaca nilai register 0x10 hasil dari *Region Growing*.
Respon Status Packet :
Nilai register yang ingin diketahui nilainya.

Tambahan pada *library* UART

Untuk mendukung *library* *Dynamixel_Havimo* yang sumber aslinya berjalan sebagai *library* Arduino diperlukan fungsi khusus pada program *UART*, beberapa fungsi tersebut antara lain:

- **uart_available()** = berfungsi untuk melihat jumlah byte yang diterima oleh receiver buffer
- **uart_peek()** = berfungsi untuk mengintip (*peek*) byte buffer teratas dari data diterima yang belum dibaca.
- **uart_flush()** = berfungsi membersihkan *Rx buffer* dari data diterima yang belum dibaca.

Adapun alur program dari DynamixelControl.exe adalah sebagai berikut:

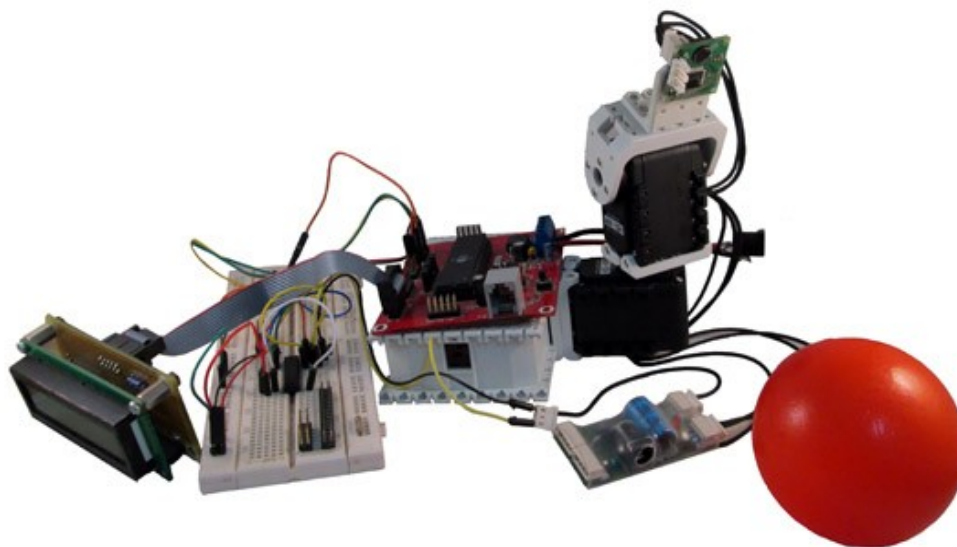


Gambar 9
Flowchart Program TrackingColor

Penjelasan urutan kerja dari program diatas adalah sebagai berikut :

1. Program melakukan inialisasi LCD 16x2 pada PORTA.
2. Menampilkan Tampilan pembuka pada LCD.
3. Inialisasi komunikasi Serial UART dengan baudrate 1Mbps.
4. Set PORTD.7 sebagai output untuk pengontrol pin *data direction*.
5. Meng-*enable* kan *Global Interrupt*.
6. Mengirimkan *Instruction Packet* untuk servo ID-1 dan ID-2 melakukan set *Status Return Level = 1*, hal ini bertujuan agar Dynamixel AX-12+ hanya memberikan balasan respon untuk *Instruction Read Data*.
7. Mengirimkan perintah agar Servo ID-1 dan ID-2 bergerak pada posisi awal (menghadap lurus kearah depan).
8. Masuk kedalam *Superloop*.
9. Mengirimkan Instruction Packet pada ID-100 (HaViMo2) agar melakukan *Capture Region* dan *Read Region*.
10. Membaca dan menyimpan hasil *Read Region* ke dalam variabel-variabel.
11. Jika data variabel Index bernilai '0', maka :
 - Menampilkan *Error* pada LCD
 - Melakukan gerak pencarian (sesuai nilai *counter*)
 - Menambah nilai *Counter*.
12. Jika data variabel Index bernilai '1' , maka :
 - *Counter* direset ke nol
 - Mengeksekusi program selanjutnya.
13. Jika data variabel *Color* bernilai '0', maka :
 - Menampilkan *Error* pada LCD.
14. Jika data variabel *Color* bernilai '1' , maka :
 - *Counter* direset ke nol
 - mengeksekusi progra selanjutnya.
15. Menggerakkan Servo ID-1 (sumbu X) dan Servo ID-2 (sumbu Y) sesuai nilai posisi yang didapatkan.
16. Menampilkan nilai posisi X dan Y pada LCD.
17. Program berjalan terus dalam *Superloop*.

Gambar koneksi keseluruhan modul dapat dilihat pada **Gambar 10**.



Gambar 10
Rangkaian antar modul pada AN206

Listing program aplikasi ini terdapat pada **AN206.ZIP**

Selamat berinovasi!

*All trademarks, company names, product names and trade names are the property of their respective owners.
All softwares are copyright by their respective creators and/or software publishers.*