

BAB III

METODE PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metode Penelitian

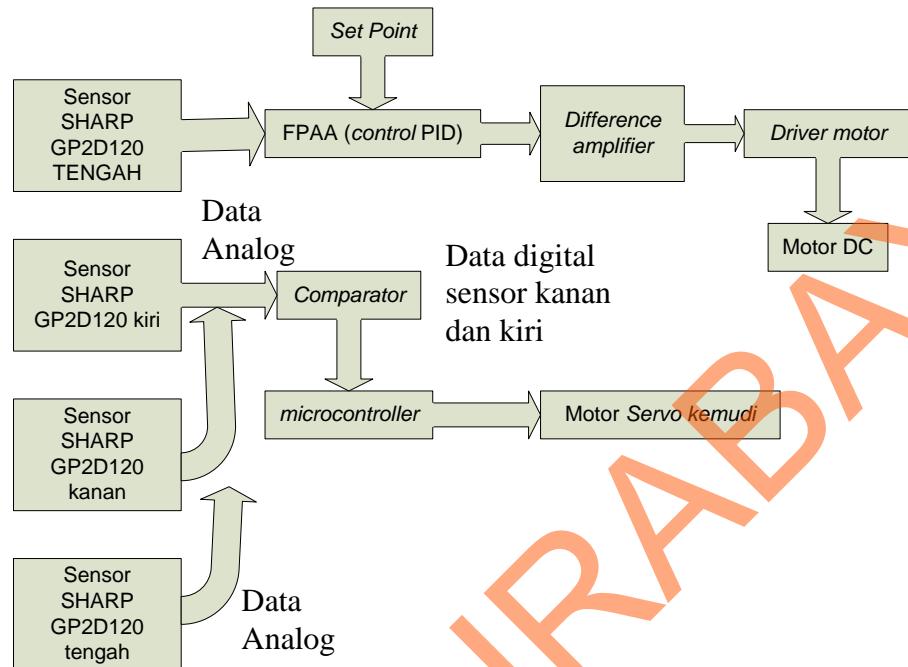
Metode penelitian yang digunakan adalah studi kepustakaan dan penelitian laboratorium. Studi kepustakaan dilakukan sebagai penunjang yang berupa data-data literatur dari masing-masing komponen, informasi dari internet dan konsep-konsep teoretis dari buku-buku penunjang.

Penelitian laboratorium berupa perancangan perangkat keras, perancangan perangkat lunak, uji coba, dan pengambilan data laboratorium. Perancangan perangkat keras dan perangkat lunak akan dibahas detil pada Sub Bab 3.3 dan 3.4. Sedangkan uji coba akan dilakukan terhadap AN221K04-V4 – *AnadigmVortex Development Board, minimum system ATTiny 2313, driver motor LM339, comparator, dan sensor sharp GP2D120*. Untuk uji coba sensor sharp GP2D120 akan diperoleh data berupa tegangan analog yang tidak linier terhadap jarak yang diukur. Data ini akan dikalibrasi yang kemudian akan digunakan dalam *PID controller*.

3.2 Rancangan Penelitian

Pada Gambar 3.1 menggambarkan blok diagram dari sistem keseluruhan. Hasil pembacaan sensor tengah akan dibaca oleh FPAA untuk kontrol PID dan *microcontroller* untuk kendali kemudi. Setelah dari FPAA akan menghasilkan sinyal PWM yang kemudian digunakan *driver motor* untuk mengatur kecepatan

motor. Ketika terdapat halangan pada jarak yang di set pada *set point* maka kecepatan laju motor akan berhenti.



Gambar 3.1 Blok Diagram Sistem keseluruhan

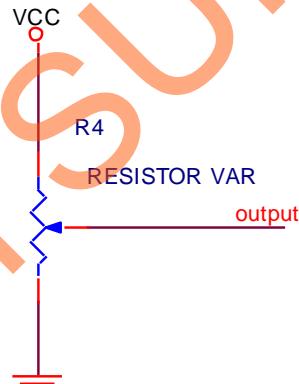
Sensor kanan, kiri, dan tengah digunakan untuk mendeteksi adanya halangan disebelah kanan, kiri, dan depan yang kemudian hasil pembacaannya digunakan oleh *microcontroller* untuk menggerakkan motor stir yang berupa motor *servo* ke kiri atau ke kanan. Sensor kanan, kiri, dan tengah merupakan sensor yang bersifat analog, artinya hasil pembacaan sensor berupa tegangan (volt) yang range nilainya 0 - 3 volt. *Microcontroller* tidak dapat menggunakan secara langsung hasil pembacaan sensor yang analog, oleh karena itu harus diubah menjadi data digital. Untuk mengubah data analog menjadi data digital, hasil pembacaan sensor harus melalui *comparator* terlebih dahulu.

3.3 Perancangan Perangkat Keras

Perancangan perangkat keras dalam penelitian ini berdasarkan diagram sistem meliputi rangkaian *set point*, modul FPAA, modul *microcontroller*, modul *comparator*, dan modul *driver motor*.

3.3.1 Rangkaian Set point

Rangkaian *set point* ini berfungsi untuk menentukan jarak yang diinginkan untuk berhenti. Gambar 3.2 adalah gambar rangkaian *set point*. Rangkaian *set point* ini menggunakan potensiometer 10 M Ω . Rangkaian *set point* merupakan rangkaian pembagi tegangan, yang *range* tegangannya dari 0-9 volt. Tegangan sumber (VCC) yang digunakan adalah 9 volt.

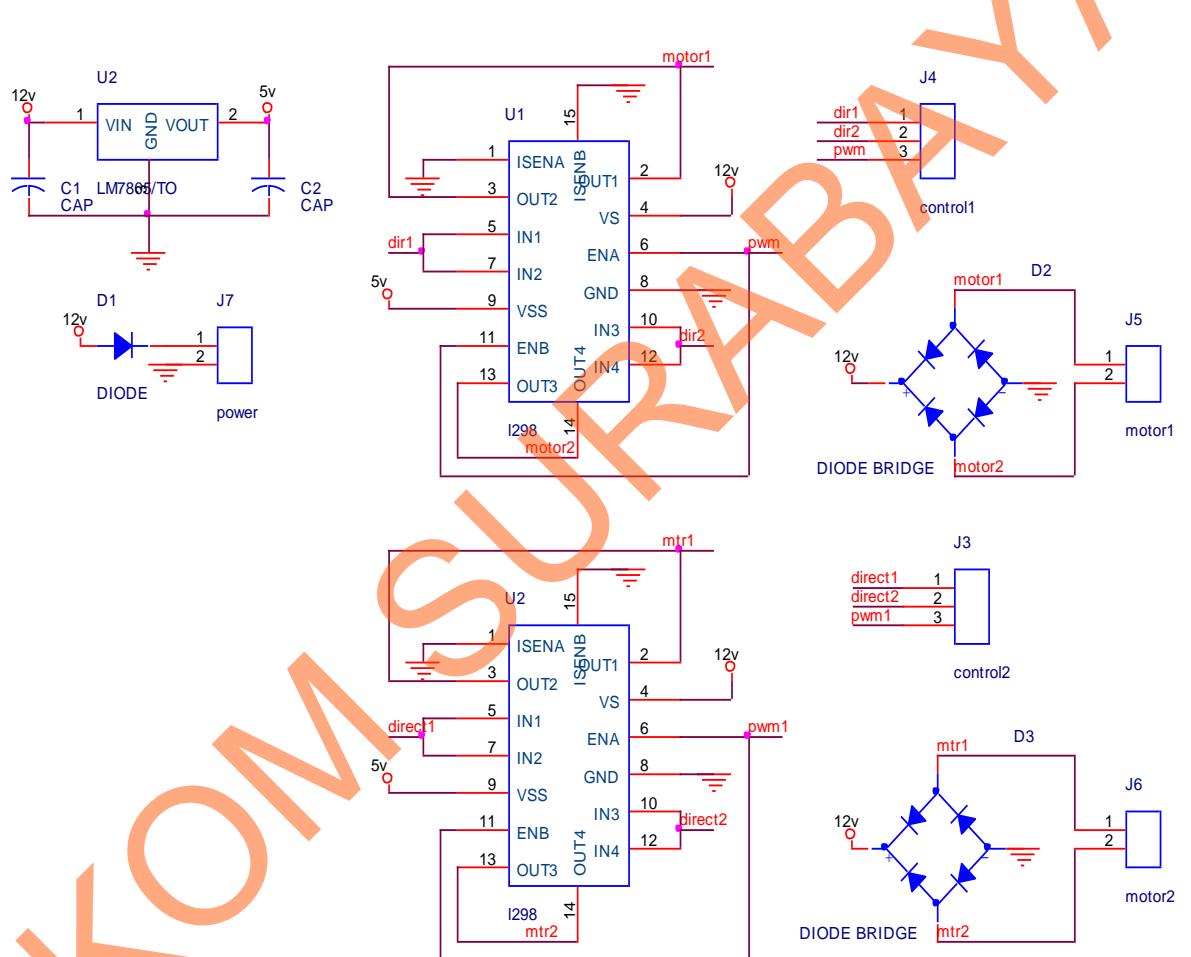


Gambar 3.2 Rangkaian *set point*

3.3.2 Rangkaian Driver Motor

Rangkaian *driver motor* merupakan komponen sangat penting untuk pengendalian motor DC. Pengendalian motor DC bisa berupa gerakan searah jarum jam atau berlawanan jarum jam. Selain itu dari *driver motor* dapat mengendalikan kecepatan motor yaitu dengan mengatur PWM (*Pulse Width Modulation*).

Driver motor yang digunakan adalah IC L298N yang mampu memberi suplai arus sampai 2 ampere pada tiap kanal. Power utama yang digunakan adalah 12 volt, sedangkan *input* L298 menerima tegangan TTL (0 sampai 5 VDC) dari *minimum system*. Rangkaian dari *driver motor* L298N ditunjukkan pada Gambar 3.3.

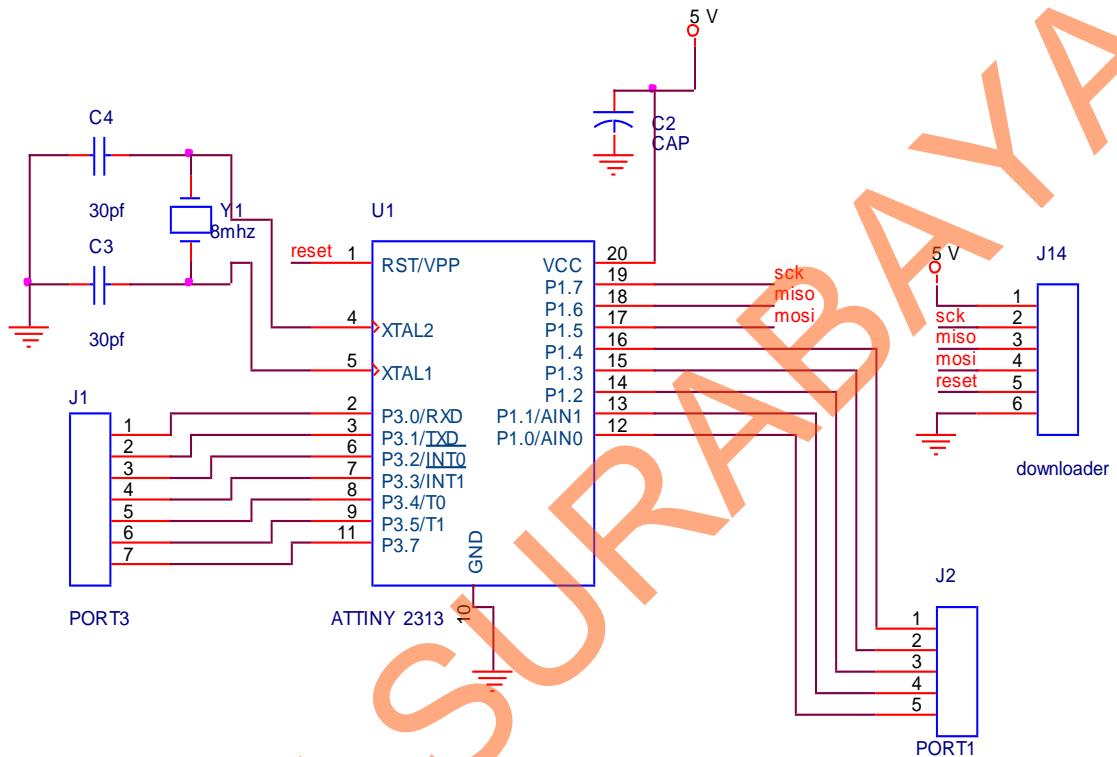


Gambar 3.3 Rangkaian dari *driver motor*

Untuk menjalankan motor DC yang penulis pakai, penulis membutuhkan arus sebesar 4 A guna menstabilkan performa robot. Maka solusi yang tepat adalah memparalel dua *channel output* dari *driver motor* tersebut menjadi satu untuk mendapatkan arus sebesar 4 A seperti yang terlihat pada Gambar 3.3.

3.3.3 Rangkaian Minimum system ATTiny 2313

Rangkaian *minimum system* ATTiny2313 digunakan sebagai pengatur belok kiri dan kanan *mobile robot* seperti pada Gambar 3.4.



Gambar 3.4 *Minimum system Microcontroller ATTiny 2313*

a. Program *Downloader*

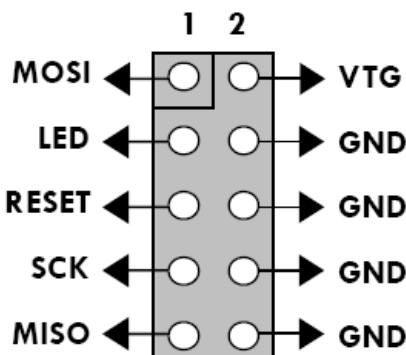
Untuk melakukan proses *download* program, yaitu file dengan ekstensi “.hex” digunakan perangkat bantu AVR USB ISP seperti pada Gambar 3.5 yang akan dihubungkan dengan *port* USB (*Universal Serial Bus*) pada komputer. Sebelum *downloader* dapat digunakan perlu dilakukan instalasi *driver* terlebih dahulu. Konfigurasi *pinout* dan keterangan dari *downloader* terdapat pada Tabel 3.1 dan Gambar 3.6.



Gambar 3.5 AVR USB ISP
(innovativeelectronics, 2010)

Tabel 3.1 Tabel Fungsi PIN

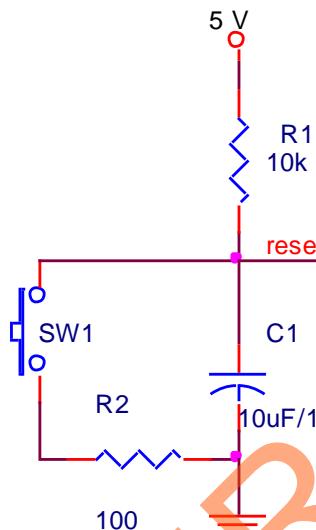
NAMA	NO.PIN	I/O	KETERANGAN
VTG	2	-	Catu daya dari project board (2.7 – 5.5 V)
GND	4, 6, 8,10	-	Titik referensi
LED	3	Output	Sinyal <i>control</i> untuk LED atau <i>multiplexer</i> (opsional)
MOSI	1	Output	Command dan data dari AVR USB ISP mkII ke target AVR
MISO	9	Input	Data dari target AVR ke AVR USB ISP mkII
SCK	7	Output	Serial <i>clock</i> , dikendalikan oleh AVR USB ISP mkII
RESET	5	Output	Reset, dikendalikan oleh AVR USB ISP mkII



Gambar 3.6 Pinout connection (innovativeelectronics, 2010)

b. Rangkaian Reset

Pin reset pada *microcontroller* adalah pin (kaki) 1. Reset dapat dilakukan secara manual atau otomatis saat power dihidupkan (*power reset ON*)



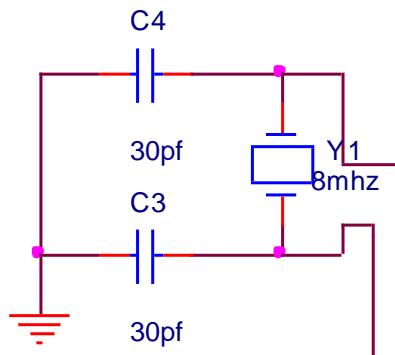
Gambar 3.7 Rangkaian Reset

Pada Gambar 3.7 merupakan gambar rangkaian reset. Reset terjadi dengan adanya logika 1 selama minimal 2 *machine cycle* yang diterima pin reset dan akan bernilai *low*. Pada saat reset bernilai *low*, *microcontroller* akan melakukan reset program yang ada di dalam *microcontroller* dan mengakhiri semua aktivitas pada *microcontroller*.

c. Rangkaian Osilator

Osilator di dalam *microcontroller* seperti pada Gambar 3.8 digunakan sebagai pembangkit pulsa clock, atau detak, karena *microcontroller* merupakan mesin sinkron, yang semua derap mesinnya dikomandani oleh pulsa *clock*. Osilator yang rangkaianya ada di dalam *microcontroller* ini memerlukan *tank-circuit* atau rangkaian resonator yang ditempatkan di luar *chip*. *Microcontroller* model lama harus ada sebuah osilator beserta *tank-circuit*-nya. Pada

microcontroller modern rangkaian osilator ada di dalam chip, resonator di luar, berupa kristal atau rangkaian LC saja.

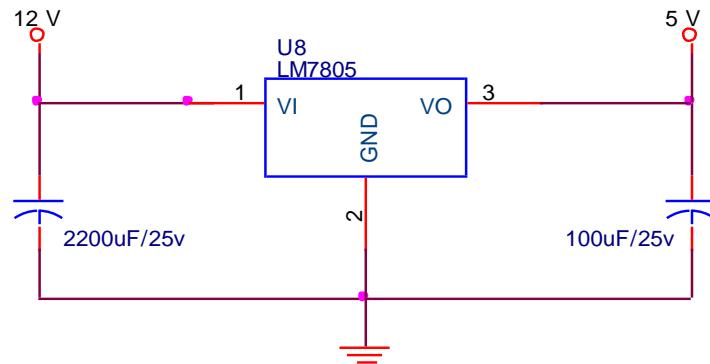


Gambar 3.8 Rangkaian osilator

d. Rangkaian Regulator

Catu daya merupakan pendukung utama bekerjanya suatu sistem. Catu daya yang biasa digunakan untuk menyuplai tegangan sebesar 5 volt adalah catu daya DC yang memiliki keluaran +5 volt. Catu daya ini digunakan untuk memberi suplai tegangan sebesar 5 volt. IC 7805 (*IC regulator*) digunakan untuk menstabilkan tegangan searah. Kapasitor digunakan untuk mengurangi tegangan kejut saat pertama kali saklar catu daya dihidupkan. Sehingga keluaran IC regulator 7805 stabil sebesar 5 volt DC. Rangkaian regulator terlihat pada Gambar

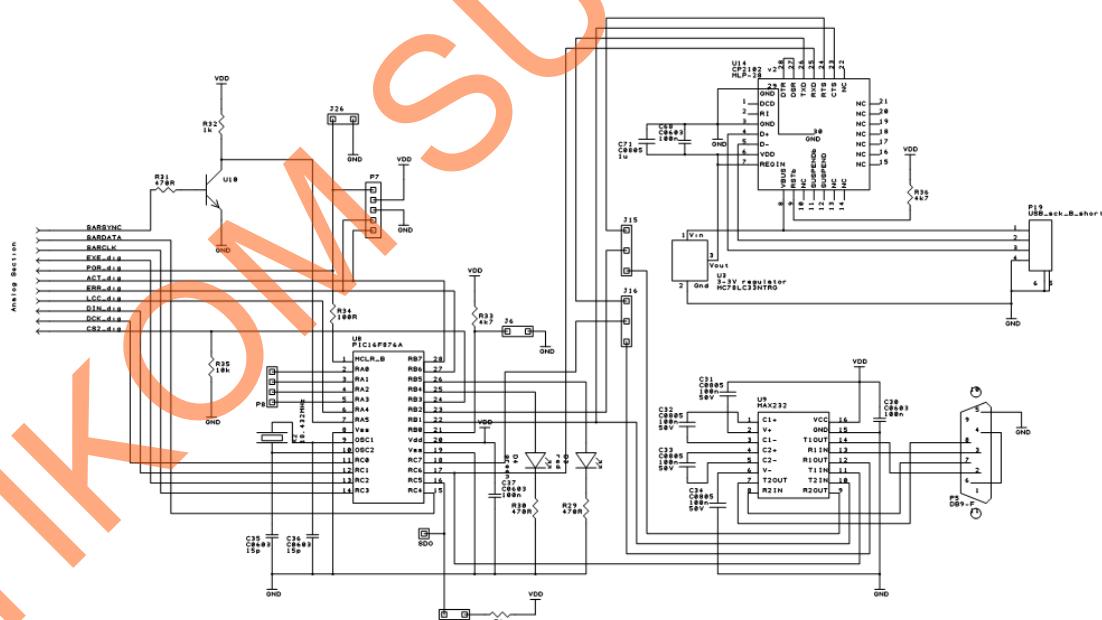
3.9.



Gambar 3.9 Rangkaian Regulator

3.3.4 Modul Anadigm AN221K04

Pada Gambar 3.10 dan Gambar 3.11 berikut ini adalah rangkaian modul FPAAs yang digunakan dalam Tugas Akhir.

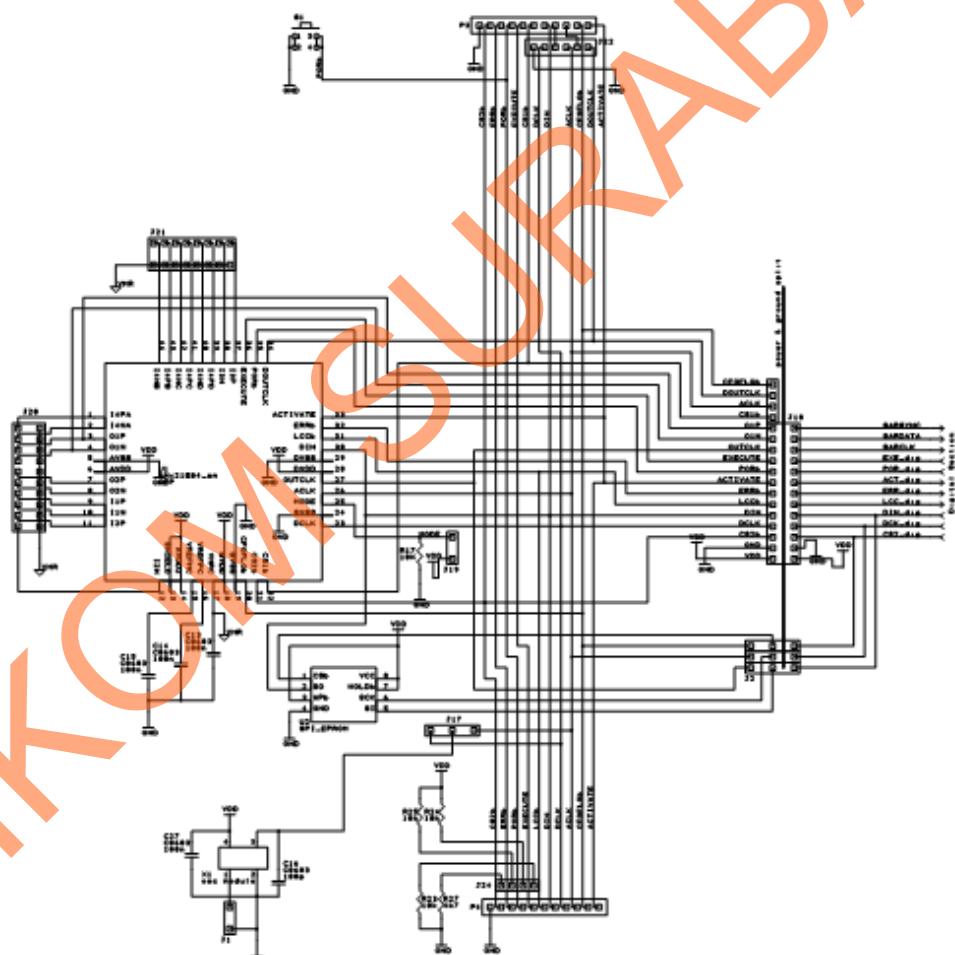


Gambar 3.10 Rangkaian modul Anadigm AN221K04 (part a) (Anadigm, 2003)

Pada penelitian ini menggunakan 1 buah modul *Anadigm AN221K04*, Di dalam modul tersebut terdapat *control PID* dan penghasil sinyal PWM. Adapun untuk alokasi port I/O bisa dilihat dari Tabel 3.2.

Tabel 3.2 Alokasi port I/O modul 1

Port	Alokasi
I2P	Jalur <i>Input</i> rangkaian <i>Set point</i>
I3P	Jalur <i>Input</i> dari sensor SHARP GP2D yang di tengah
O2P	Jalur <i>Output</i> PWM



Gambar 3.11 Rangkaian modul *Anadigm AN221K04* (part b) (*Anadigm*, 2003)

Pin VCC diberi masukan tegangan minimal 4 Volt sampai dengan maksimal 12.5 Volt dan tegangan optimal sekitar 5 Volt. Pin RESET berfungsi untuk masukan reset program secara otomatis atau manual.

Untuk melakukan proses *download* program dari komputer ke dalam *memory* program internal FPAA, dapat memilih antara Interface USB dan Serial RS232. Caranya dengan menghubungkan *jumper* atas dengan yang tengah jika menggunakan serial RS232 dan *jumper* tengah dan bawah untuk menggunakan serial RS232. Gambar 3.12 merupakan *jumper* yang dipasang untuk mengaktifkan *Interface* serial RS232.



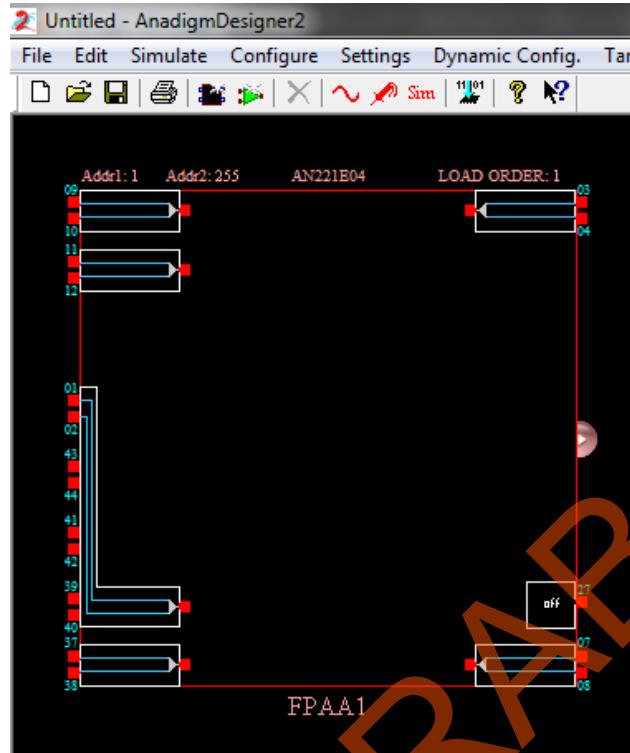
Gambar 3.12 Konfigurasi *jumper* Serial RS232 dan USB (Anadigm, 2003)

Program bantu yang terintegrasi untuk merancang rangkaian sekaligus melakukan *download* program adalah *Anadigm designer*. Merupakan perangkat lunak khusus yang digunakan untuk membuat rangkaian analog seperti *amplifier*, *integrator*, *differensiator*, *comparator*, dan *adder* dalam suatu chip IC. *Anadigm designer* dibuat oleh *Anadigm Inc* pada tahun 2004 dengan versi 2.2.7 (*Anadigm inc*, 2004:56). Dan yang akan digunakan adalah *Anadigm designer* Ver. 2.7.0.1.

Perangkat lunak ini mempunyai kelebihan antara lain :

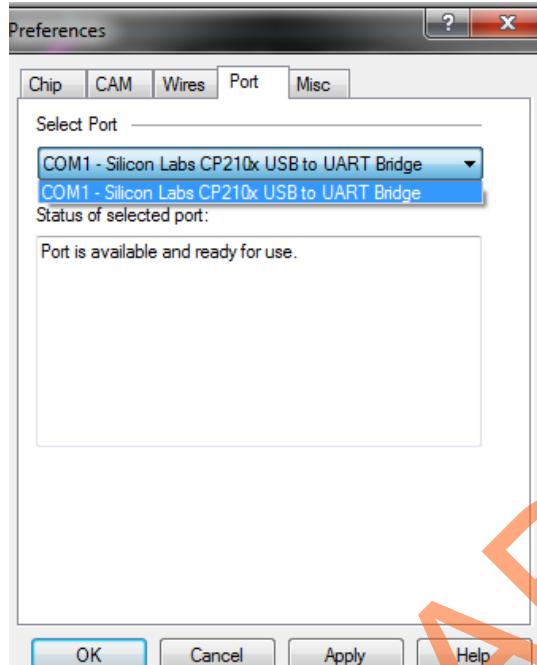
1. Mampu membuat beberapa rangkaian analog yang kompleks dengan cepat dan mudah.
2. Mampu untuk mengkonversi program menjadi bahasa C yang akhirnya dapat digunakan untuk keperluan program pada *microprocessor*.
3. Mampu melakukan simulasi keluaran rangkaian analog yang telah dibuat, sehingga program yang akan di transfer ke *device* FPAA benar-benar sesuai dengan yang dirancang.
4. Mampu membuat sistem filter dan PID (*Proportional Integrator* dan *Differensiator*) dengan mudah dan cepat. Tampilan awal *Anadigm* ditunjukkan pada Gambar 3.13.

Selain merancang, program *Anadigm designer* juga dapat disimulasikan sebelum di-*download* ke dalam *hardware* FPAA dengan cara klik menu *Simulate*, *Begin Simulation*, atau tekan *F5*. Kemudian sebelum *download* program, pastikan FPAA sudah terhubung dengan *interface* komputer. Dan samakan konfigurasi COM port pada *Anadigm designer* dengan COM port pada *Device Manager*.

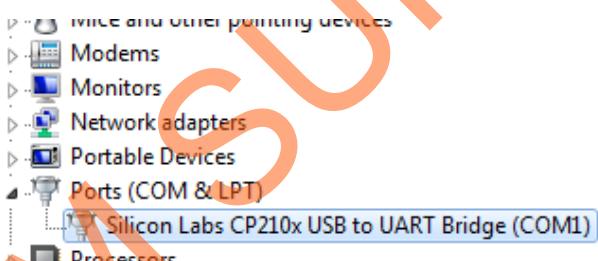


Gambar 3.13 Software *Anadigm designer*

Untuk *Anadigm designer* dengan cara klik menu *Setting* kemudian *Preferences Port*. Gambar 3.14 adalah tampilan pada *Anadigm designer* dan Gambar 3.15 pada *Device Manager* komputer.



Gambar 3.14 Tampilan setting COM Port pada Anadigm designer.



Gambar 3.15 Tampilan setting COM Port pada device manager.

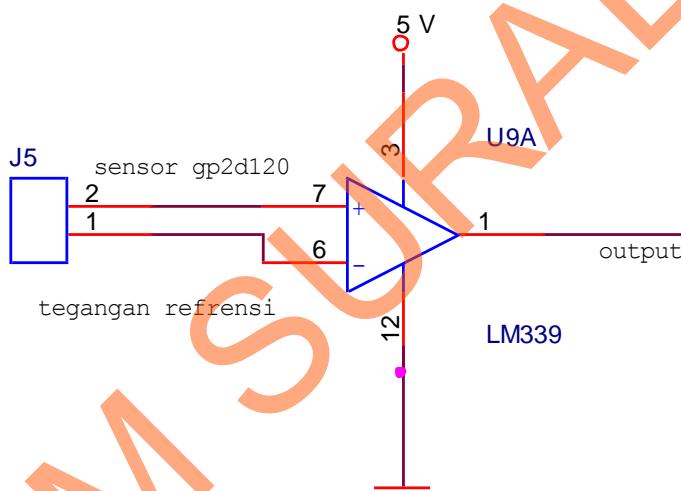
3.3.5 Rangkaian Comparator

Comparator merupakan sebuah *op-amp*. *Op-amp* tersebut akan membandingkan nilai tegangan pada kedua masukannya, apabila masukan (-) lebih besar dari masukan (+) maka, keluaran *op-amp* akan menjadi sama dengan $-V_{supply}$, apabila tegangan masukan (-) lebih kecil dari masukan (+) maka keluaran *op-amp* akan menjadi sama dengan $+V_{supply}$.

Jadi dalam hal ini jika V_{input} lebih besar dari V maka keluarannya akan menjadi $-V_{supply}$, jika sebaliknya, V_{input} lebih besar dari V maka keluarannya

akan menjadi $+V_{supply}$. Untuk *op-amp* yang sesuai untuk dipakai pada rangkaian *op-amp* untuk *comparator* biasanya menggunakan *op-amp* dengan tipe LM339 yang banyak di pasaran. Gambar 3.16 merupakan gambar rangkaian *comparator*.

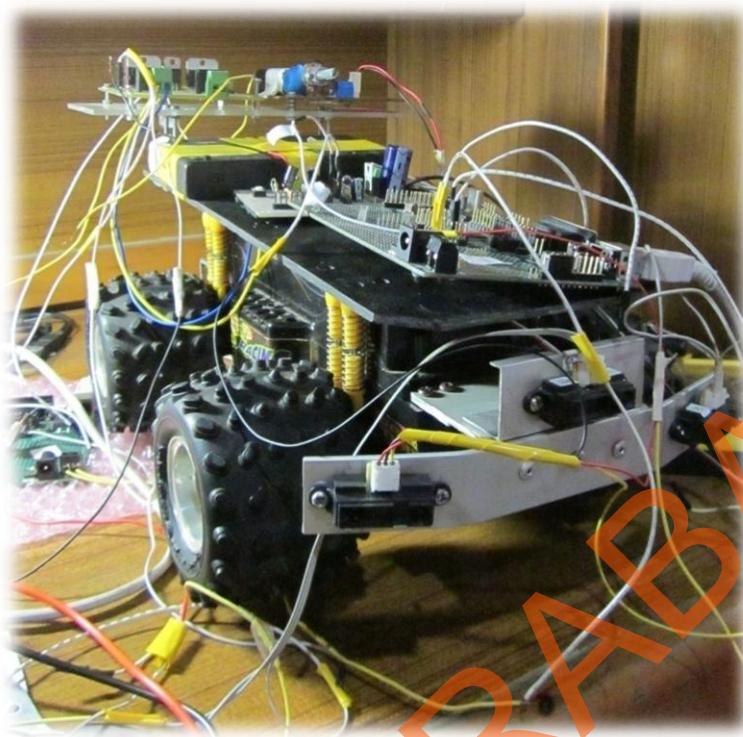
Pada penelitian ini penulis menggunakan tegangan referensi sebesar 0,322. Hal ini karena sensor sharp GP2D120 mendeteksi jarak 45 cm pada tegangan 0,322. Jika tegangan dari sensor lebih besar daripada tegangan referensi maka *output*-nya adalah $+V_{supply}$, dan jika tegangan sensor lebih kecil daripada tegangan refensi maka *output*-nya $-V_{supply}$.



Gambar 3.16 Rangkaian *comparator*

3.4 Perancangan Arsitektur Sistem

Perancangan arsitekturnya dapat dilihat seperti Gambar 3.17. Gambar 3.17 adalah gambar robot tampak depan. Sensor Sharp GP2D120 diletakkan di bagian depan (bemper). Untuk yang sebelah kiri dan kanan letaknya agak menyerong. Pada Gambar 3.18 merupakan gambar robot tampak samping.



Gambar 3.17 Robot tampak depan



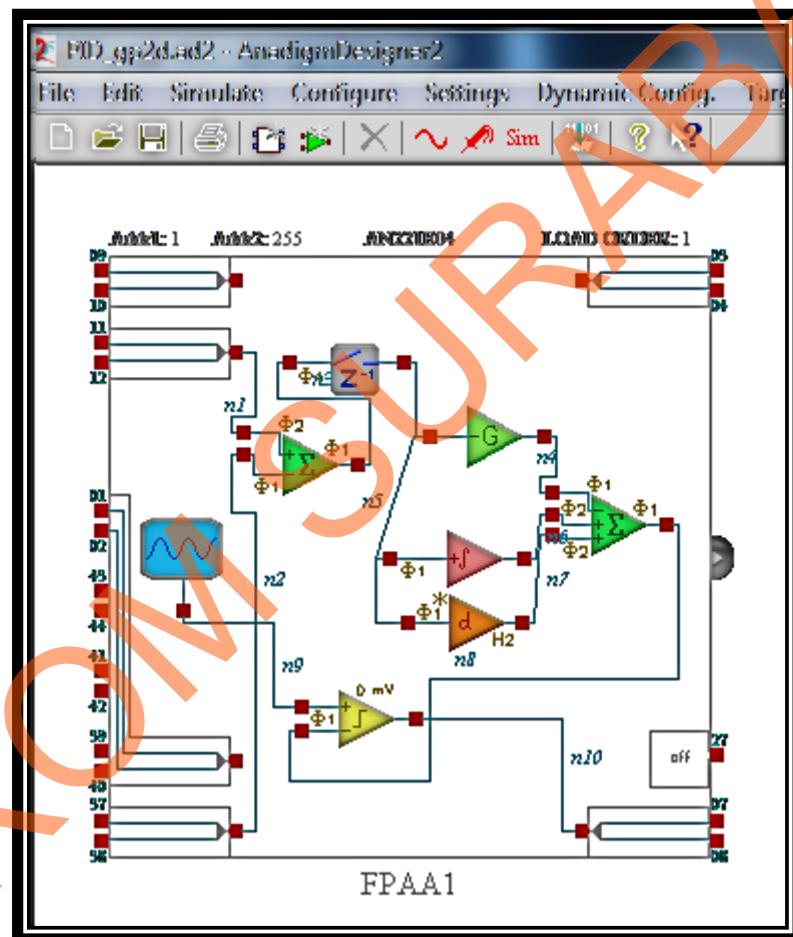
Gambar 3.18 Robot tampak samping

3.5 Perancangan Perangkat Lunak

Selain *hardware* yang diperlukan pada perancangan dan pembuatan pada penelitian ini juga diperlukan program pada *microcontroller*, komputer dan juga FPAA untuk dapat bekerja sesuai dengan fungsinya.

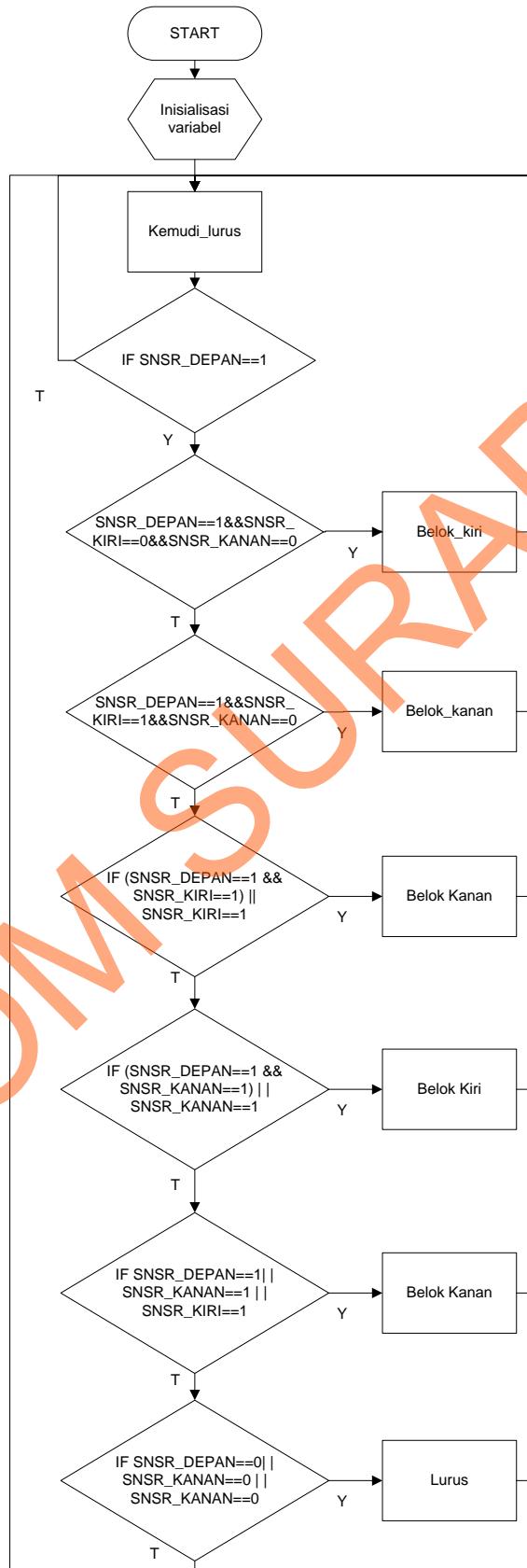
3.5.1 Perancangan Program pada *Anadigm AN221K04*

Perancangan program *Anadigm* dapat dilihat pada Gambar 3.19.



Gambar 3.19 Perancangan kontroler PID dan PWM

3.5.2 Flow Chart Program pada Microcontroller



Gambar 3.20 Flowchart microcontroller

Dari Gambar 3.20 merupakan diagram alir *microcontroller* dengan penjelasan sebagai berikut sensor kiri, kanan, dan depan akan terus secara kontinyu mendeteksi jarak. Sensor kiri, kanan dan depan terlebih dahulu masuk ke dalam *comparator* untuk mendapatkan nilai digital. Pada saat sensor kiri, kanan, dan depan tidak mendeteksi adanya halangan maka motor kemudi akan bergerak tepat di tengah-tengah. Pada saat sensor kanan tidak terdapat halangan, sensor kiri dan depan ada halangan maka akan belok kanan yaitu dengan motor kemudi berputar ke kiri. Pada saat sensor kanan dan depan terdapat halangan dan sensor kiri tidak ada halangan maka akan belok kiri yaitu dengan motor kemudi berputar ke kanan. Jika ketiga sensor mendeteksi adanya halangan maka akan belok ke kanan yaitu dengan motor kemudi berputar ke kiri.

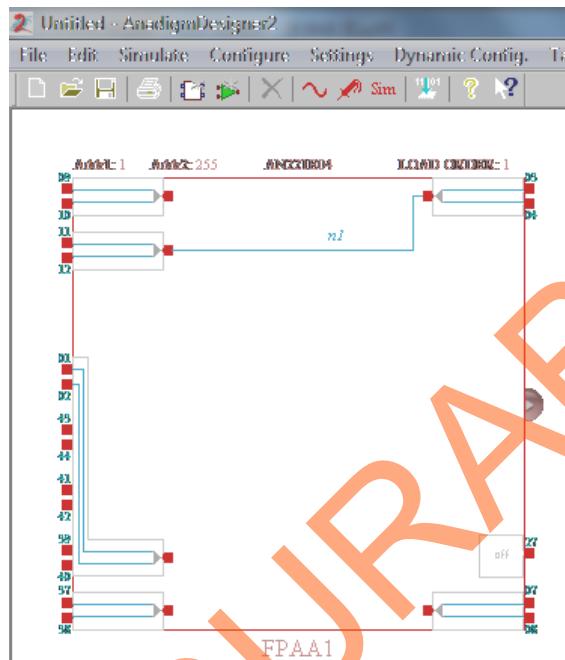
3.6 Metode Pengujian dan Evaluasi Sistem

Untuk mengetahui apakah aplikasi yang dibuat dapat berjalan sesuai dengan yang diharapkan, maka akan dilakukan pengujian dan evaluasi sistem untuk setiap tahapan-tahapan dalam pembuatan aplikasi. Dimulai dari pengujian pin I/O FPAA, pengujian *driver motor*, pengujian *microcontroller*, pengujian *comparator*, menggerakkan robot dengan menggunakan FPAA, menggerakkan kemudi menggunakan *microcontroller*, dan yang terakhir adalah pengujian sistem secara keseluruhan.

3.6.1 Pengujian PORT *Input* dan *Output* Modul FPAA AN221K04-V4

Pengujian pin I/O dilakukan untuk melihat apakah pin I/O dari FPAA berfungsi dengan baik. Cara pengujian untuk melihat pin I/O FPAA telah berfungsi dengan baik adalah dengan menghubungkan *input* dengan *output* seperti

Gambar 3.21. Kemudian ukur tegangan *output* menggunakan multimeter dan lihat apakah tegangan *output* sama dengan tegangan *input*, jika sama maka pin I/O bisa digunakan.



Gambar 3.21 Pengujian PIN I/O FPAA

3.6.2 Pengujian *Driver Motor*

Untuk pengujian *driver motor* dilakukan dengan cara menghubungkan *input 1* dengan tegangan *high* (5V) dan menghubungkan *input 2* dengan tegangan *low* (0V).

3.6.3 Pengujian Microcontroller

Pengujian ini dilakukan untuk mengetahui *microcontroller* telah berfungsi dengan baik. Pengujian dilakukan dengan cara men-*download* program untuk menghidupkan LED untuk setiap PORT I/O.

3.6.4 Pengujian Sensor SHARP GP2D120

Pengujian ini dilakukan untuk mengetahui sensor dapat berfungsi dengan baik. Pengujian yang dilakukan adalah dengan memberikan tegangan VCC pada pin 1 dan *ground* (GND) pada pin 2. Ukur pin 3 dengan menggunakan multimeter.

3.6.5 Pengujian Pergerakan Robot

Pengujian ini dilakukan untuk mengetahui pergerakan robot apakah sesuai dengan yang diharapkan. Pengujian ini dilakukan dengan men-*download* program dari FPAA untuk pengujian ini. Jika robot berhenti jika terdapat halangan berarti program telah berhasil.

3.6.6 Pengujian Pergerakan Kemudi

Pengujian ini dilakukan untuk melihat apakah kemudi berhasil berjalan sesuai dengan program yang di-*download*, kendali kemudi menggunakan motor *servo*. Pengujian ini dilakukan dengan menghubungkan sensor SHARP GP2D120 kiri, kanan, dan tengah dengan *comparator*. Hasil *output* dari *comparator* akan masuk ke PORT *input* *microcontroller*. Hubungkan motor servo ke pin I/O *microcontroller*. *Download* program untuk mengatur kendali kemudi dan lihat hasilnya.

3.6.7 Evaluasi Sistem Keseluruhan

Setelah melalui seluruh proses pengujian di atas maka perlu dilakukan pengujian sistem secara keseluruhan. Dimulai dari mengintegrasikan sensor PSD tengah dan setpoint dengan FPAA. Dari pembacaan sensor PSD tersebut akan menghasilkan sinyal PWM pada PORT *output* FPAA. Sinyal PWM yang dihasilkan FPAA akan dimanfaatkan oleh *driver motor* untuk mengatur kecepatan

motor. Pengaturan *set point* akan berpengaruh pada jarak yang diinginkan user untuk memperlambat motor pada jarak yang diinginkan *user*. Ketika sensor depan mendeteksi adanya halangan maka motor akan melambat dan akan berhenti pada jarak yang diatur pada *set point*. Sensor kiri, kanan, dan tengah diintegrasikan dengan *comparator*. Dari *comparator* akan menghasilkan tegangan digital. Tegangan digital ini yang digunakan *microcontroller* untuk mengatur motor *servo* kemudi. Motor kemudi akan bergerak ke kiri jika sensor kanan mendeteksi adanya halangan, dan bergerak ke kanan jika sensor kiri mendeteksi adanya halangan. Apabila sistem berjalan sesuai dengan langkah-langkah pengujian di atas, maka sistem yang dibuat sudah baik.