

## ROBOT BERODA YANG BERJALAN MENURUT JALUR YANG DIGAMBAR PADA *PERSONAL DIGITAL ASSISTANT*

Michael Yohannes<sup>1)</sup>, Peter Rhatodirdjo Angka<sup>2)</sup>  
E-mail: keonk10@gmail.com, peter@mail.wima.ac.id

### ABSTRAK

*Dengan berkembangnya teknologi dan rumitnya jalur yang harus dilalui oleh robot, maka dibuatlah Robot Beroda Yang Berjalan Menurut Jalur Yang Digambar Pada Personal Digital Assistant (PDA). Tujuan dari pembuatan alat tersebut untuk mempermudah operator dalam menentukan jalur yang harus dilalui oleh robot. Cara kerja robot adalah membaca titik yang digambar oleh operator, lalu PDA akan menghitung jarak jalur yang akan dilalui, setelah itu PDA berkomunikasi dengan mikrokontroler untuk berjalan sesuai dengan jalur tersebut. Cara menggambar jalur pada layar PDA adalah dengan menyentuh suatu titik pada layar sentuh, lalu program akan menempatkan titik pada daerah yang disentuh untuk kemudian dihubungkan secara otomatis dengan titik jalur sebelumnya, sehingga terbentuklah jalur, dan proses itu akan berulang-ulang sampai operator menekan tombol "selesai". Setelah jalur selesai digambar, maka titik jalur yang digambar pertama kali dianggap sebagai titik START dan yang digambar terakhir sebagai titik END. PDA berkomunikasi dengan mikrokontroler secara 2 arah (kirim-terima) dengan menggunakan rangkaian transmisi data infra merah. Untuk mengenali jarak yang telah dilalui robot, maka digunakan sensor optocoupler pada kedua roda belakang. Sensor optocoupler ini juga digunakan pada saat memutar haluan robot sesuai dengan sudut yang dikehendaki. Aplikasi robot ini pada dunia nyata antara lain robot pembawa barang.*

**Kata Kunci :** robot, jalur, PDA, mikrokontroler, optocoupler

### PENDAHULUAN

Robot semakin memegang peranan penting dalam kehidupan manusia. Kemudahan dalam mengoperasikan, biaya operasional yang relatif murah dibanding menggaji pegawai, kecepatan serta keakuratan dalam pekerjaan menjadikan robot semakin menjadi primadona dalam dunia industri, film, medis, dan banyak bidang lainnya. Biasanya arah gerak robot, diprogram terlebih dahulu ataupun diatur oleh komputer.

Namun seiring dengan semakin rumitnya jalur yang harus dilalui oleh robot, maka dibuatlah "Robot Beroda Yang Berjalan Menurut Jalur Yang Digambar Pada *Personal Digital Assistant*". Adapun tipe *Personal Digital Assistant* yang digunakan adalah tipe iPAQ 1940 yang mudah dijumpai di pasaran Indonesia. Penggunaan *Personal Digital Assistant* dalam robot ini dimaksudkan agar robot dapat memiliki kemampuan komputasional yang lebih canggih, dan rumit, karena adanya komputer yang terintegrasi. Layar sentuh yang terdapat pada *Personal Digital Assistant* juga memudahkan operator dalam menampilkan data-data yang dikehendaki, serta menerima *input* dari pengguna.

Diharapkan dengan terwujudnya peralatan ini, maka akan ada alternatif lain dalam pengendalian robot. Apabila semula robot yang menggunakan otak berupa mikrokontroler saja, namun kini dapat juga dibantu dengan menggunakan komputer genggam (*Personal Digital Assistant*), sehingga memungkinkan robot melakukan perhitungan komputasional, dan dapat diintegrasikan dengan piranti digital lainnya. Selain itu dengan adanya robot ini, maka operator robot dapat lebih mudah dalam memasukkan jalur yang harus dilalui. Robot ini juga dapat dikembangkan sebagai robot kompetisi dalam Kontes Robot Indonesia (KRI) ataupun Kontes Robot Cerdas Indonesia (KRCI).

### TINJAUAN PUSTAKA

#### Mikrokontroler ATmega32

Mikrokontroler yang digunakan dalam pembuatan peralatan ini adalah AVR Atmega32 produk keluaran dari *ATMEL* yang merupakan mikrokontroler dengan arsitektur *Reduced Instruction Set Computer (RISC)* yang lebih sederhana perintah intruksinya dibandingkan dengan keluarga 80C51 yang menggunakan arsitektur *Complex Instruction Set Computer (CISC)* dengan perintah intruksi yang kompleks di dalam eksekusi program<sup>[1]</sup>.

Mikrokontroler *Alf and Vegard's Risc Processor (AVR)* memiliki konsep yang hampir

<sup>1)</sup> Mahasiswa di Fakultas Teknik Jurusan Teknik Elektro Universitas Katolik Widya Mandala Surabaya

<sup>2)</sup> Staf pengajar di Fakultas Teknik Jurusan Teknik Elektro Universitas Katolik Widya Mandala Surabaya

sama dengan mikrokontroler keluarga 8051, yaitu memiliki 8 pin dalam 1 *port*-nya yang dapat digunakan sebagai *input* atau *output digital* dan penggunaan pin (*MISO*, *MOSI*, *SCK*, *RESET*) untuk memasukkan file dalam bentuk heksadesimal dari komputer ke mikrokontroler.

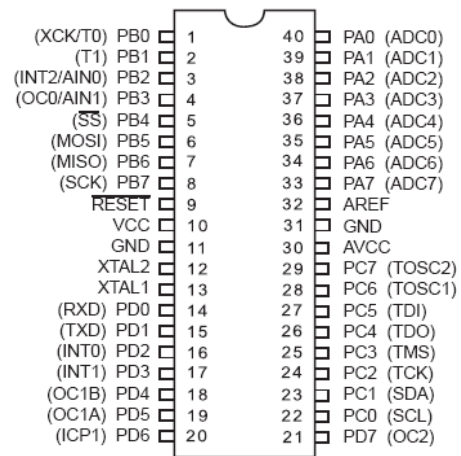
Mikrokontroler AVR memiliki arsitektur *RISC* 8-bit, semua instruksi dikemas dalam kode 16-bit (16-bit *word*) dan sebagian besar instruksi dieksekusi dalam satu siklus *clock*. Secara umum yang membedakan dengan keluarga 80C51 adalah kecepatan (AVR lebih cepat dalam mengeksekusi instruksi program), memori (memori AVR lebih besar), periferan dan fungsinya (penambahan *ADC* internal, komunikasi serial *I2C*).

Pemrograman AVR tergolong mudah karena pemrograman AVR dapat dilakukan dengan menggunakan teknik *In System Programming (ISP)* di mana kode *HEX* dapat langsung di *download* pada mikrokontroler di dalam rangkaian aplikasi (rangkaiannya yang diinginkan). Pengisian program pada AVR juga dapat dilakukan dengan komunikasi paralel menggunakan perangkat *downloader* dengan memakai 74HC244 (*octal buffer*).

Mikrokontroler AVR Atmega32 memiliki fasilitas antara lain:

- 32K Bytes *In-System Programmable (ISP) Flash Memori*.
- Kisaran operasi 4,5V sampai 5,5V.
- Operasi Secara penuh Statis: 0 Hz ke 16 MHZ.
- 32 x 8-bit *General Purpose Register*.
- 512 Byte *EEPROM*.
- 1 K Byte *internal SRAM*.
- *Timer/Counter* 2 x 8-bit.
- 3 buah pin *interrupt (INT0, INT1, INT2)*.
- *Output PWM* 4 kanal.
- 8 x 10-bit *ADC*.
- *Master/Slave serial interface*.
- 32 pin *input/output*.

Pada Gambar 1 disajikan pin mikrokontroler ATmega32.



Gambar 1. Pin Mikrokontroler ATmega32

### Fungsi Pin ATmega32

Konfigurasi pin dari mikrokontroler AVR Atmega 32 sebagai berikut:

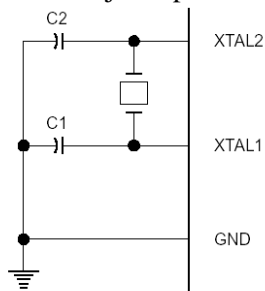
- a) Pin 1 sampai 8 (*Port B0-B8*) merupakan *port* paralel 8 bit dua arah (*bidirectional*) yang dapat digunakan untuk berbagai keperluan (*general purpose*), selain itu mempunyai fungsi khusus *T0 (Timer 0)*, *T1 (Timer 1)*, *INT2 (Interrupt 2)* dan *ISP* pada *MOSI*, *MISO*, dan *SCK*.
- b) Pin 9 (*reset*) adalah *input reset* (aktif *Low*). Pulsa transisi dari tinggi ke rendah akan me-*reset* program mikrokontroler. Pin ini akan melakukan *reset* secara manual.
- c) Pin 10 (*Vcc*), tegangan positif 4,5 V sampai dengan 5,5 V.
- d) Pin 11 (*Gnd*), merupakan tegangan *ground*;
- e) Pin 12 (*X\_tall 1*), adalah *input* bagi *inverting oscillator amplifier* dan *input* bagi *clock internal*.
- f) Pin 13 (*X\_tall 2*), adalah *Output inverting oscillator amplifier*.
- g) Pin 14 sampai 21 (*Port D0* sampai *D8*) merupakan *port* paralel 8 bit dua arah (*bidirectional*) yang dapat digunakan untuk berbagai keperluan (*general purpose*) selain itu mempunyai fungsi khusus *RXD-TXD* (serial), *INT0 (interrupt 0)* dan *INT1 (interrupt 1)*.
- h) Pin 22 sampai 29 (*Port C0* sampai *C8*) merupakan *port* paralel 8 bit dua arah (*bidirectional*) yang dapat digunakan untuk berbagai keperluan (*general purpose*).
- i) Pin 33 sampai 40 (*Port A7* sampai *A0*) merupakan *port* paralel 8 bit dua arah (*bidirectional*) yang dapat digunakan untuk

berbagai keperluan (*general purpose*), *port A* dapat digunakan sebagai fungsi *ADC converter*.

- j) Pin 30 (*Avcc*) merupakan tegangan eksternal untuk *port A* sebagai fungsi *ADC converter*.
- k) Pin 32 (*Aref*) merupakan tegangan referensi untuk *port A*, bila digunakan sebagai fungsi pengubah *analog* ke digital (*ADC converter*).
- l) Pin 22 dan pin 23 merupakan pin *SDA-SCL* untuk komunikasi serial *Inter Interface Circuit (I2C)*.

### Rangkaian Osilator

Frekuensi kerja dari mikrokontroler *Atmega32* tergantung dari besarnya frekuensi kristal yang dipakai dan dipasang pada kaki *XTAL1* dan *XTAL2*, dalam hal ini memakai frekuensi 4 Mhz. Selain itu, dapat juga dipakai sumber *clock* eksternal yang dipasang pada kaki *XTAL1*, di mana kaki *XTAL2* tidak digunakan. Rangkaian osilator disajikan pada Gambar 2.



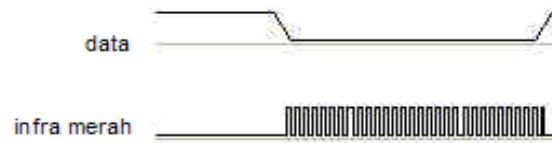
Gambar 2. Rangkaian Osilator

### Transmisi Data Inframerah

Transmisi data inframerah digunakan untuk komunikasi antara mikrokontroler dengan *PDA iPAQ 1940*<sup>[2,3]</sup>. Pengirim dan penerima dihubungkan dengan mikrokontroler *ATMega16*. Penggunaan inframerah dikarenakan *PDA iPAQ 1940* tidak mendukung adanya koneksi *RS232*, sehingga harus digunakan alternatif lain dalam komunikasi dengan *ATMega32* yaitu dengan inframerah. Untuk pengiriman data, data tersebut harus dimodulasikan dengan gelombang 38 kHz sebelum ditransmisikan. Gambar dari sinyal data inframerah disajikan pada Gambar 3.

Sinyal data inframerah tersebut dapat diterima oleh penerima dengan menggunakan demodulator inframerah yang banyak dijumpai di pasaran. Sinyal frekuensi 38 kHz yang

diterima oleh dioda penerima infra merah lalu dikembalikan lagi menjadi bentuk data.



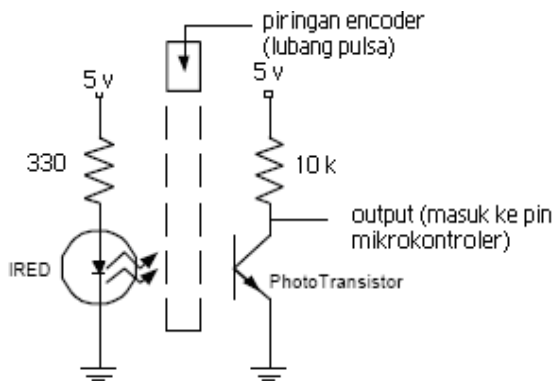
Gambar 3. Bentuk Sinyal Inframerah

### Transmisi Data dengan Port Serial

Pada mikrokontroler *ATMega32* terdapat pin yang dapat mentransmisikan dan menerima data secara serial. Pin ini adalah pin *RXD* dan *TXD*. Pin *RXD* digunakan untuk menerima data serial, sedangkan *TXD* digunakan untuk mengirim data serial. Adapun *baudrate* yang digunakan adalah 9.600 bps untuk mengirim, dan 2.400 bps untuk menerima. Mode transmisi serial yang digunakan adalah asinkron karena tidak adanya *clock* yang ikut ditransmisikan.

### Optocoupler

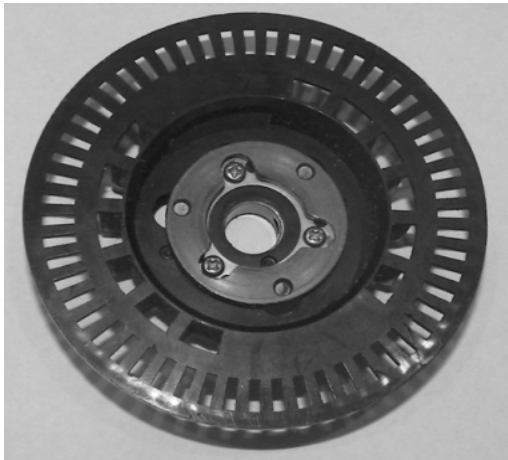
*Optocoupler* berasal dari kata *Opto* (optik) dan *Coupler* (penghubung). Jadi *optocoupler* adalah suatu komponen penghubung (*coupling*) yang bekerja berdasarkan cahaya atau optik. *Optocoupler* adalah gabungan sebuah *LED* (inframerah) pada sisi *input* dan sebuah *photodiode* atau *phototransistor* pada sisi *output* dalam satu kemasan (komponen dalam satu kemasan untuk sinyal *input* dan sinyal *output* melalui satu medium). Tegangan *output* akan berubah bila terjadi perubahan pada pancaran cahaya *LED* (infra merah). Perubahan tersebut disebabkan karena terhalangnya sinar *LED* (infra merah) yang mengenai *photodiode*. *Optocoupler* membutuhkan catu daya sama dengan *LED* yaitu arus sekitar 30 mA agar dapat beroperasi. Bentuk *output* berupa sinyal pulsa, bila tidak terhalang akan berkondisi 1, bila terhalang benda akan berlogika 0. Rangkaian *optocoupler* disajikan pada Gambar 4 sebagai berikut:



Gambar 4. Rangkaian Optocoupler

### Encoder

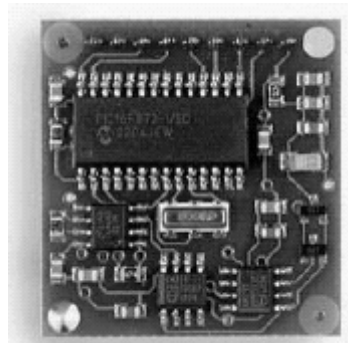
Encoder merupakan piringan yang terbuat dari plastik berbentuk lingkaran yang diletakkan pada poros motor. Pada piringan tersebut terdapat sebanyak 54 lubang yang mengelilingi pinggir lingkaran sebagai obyek pulsa dari *optocoupler*. Makin banyak lubang, maka pembacaan akan makin sering dan jika dikonversi ke RPM akan didapat hasil yang makin tinggi responnya. Gambar *encoder* disajikan pada Gambar 5.



Gambar 5. Encoder 54 Lubang

### Sensor kompas

*Devantech magnetic compass CMPS03* biasa digunakan pada robot untuk tujuan mengemudikan robot<sup>[4]</sup>. *Devantech magnetic compass CMPS03* menggunakan sensor medan magnet tipe Philips KMZ51 untuk mendeteksi medan magnet bumi. Bentuk *Devantech magnetic compass CMPS03* disajikan pada Gambar 6.



Gambar 6. Bentuk *Devantech CMPS03*

Spesifikasi secara umum dari *Devantech magnetic compass CMPS03* adalah sebagai berikut:

- Tegangan : 5 V;
- Arus : 20 mA;
- Resolusi : 0,1 derajat;
- Keakuratan : 3-4 derajat;
- Keluaran 1 : *Timing pulse* berkisar 1 sampai 37ms;
- Keluaran 2 : *I2C Interface*, 0-255 dan 0-3599;
- Kecepatan SCL: 1 MHz;
- Berat : 0,03 gram;
- Ukuran : 32 mm x 35mm.

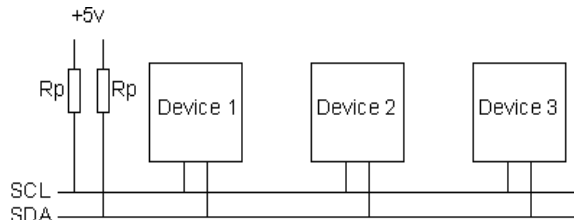
### I2C bus

Keluaran sensor kompas yang digunakan pada alat ini adalah melalui hubungan *I2C*. *I2C* adalah singkatan dari *Inter Integrated Circuit Bus*. *I2C* adalah hubungan serial 2 jalur yang telah distandarkan oleh *Philips Semiconductor* pada tahun 1980-an. *I2C* pertama kali dibuat dengan tujuan memberikan kemudahan untuk menghubungkan *CPU* ke *peripheral chips* pada *TV*.

Bentuk fisik dari *bus* terdiri dari 2 jalur dan sebuah sambungan *ground*. Jalur-jalur itu adalah *SDA* dan *SCL*, di mana keduanya merupakan *bidirectional* (memiliki 2 arah). *SCL* (*Serial Clock Line*) adalah *clock line*. *SCL* digunakan untuk menyamakan semua pengiriman data pada *I2C bus*. *SDA* (*Serial Data Line*) adalah *data line*. Jalur pada *SCL* dan *SDA* dihubungkan ke semua peralatan pada *I2C bus*, di sana jalur ketiga terhubung ke *ground* atau 0 volt.

Kedua jalur *SCL* dan *SDA* adalah “*open drain driver*” artinya *chip* dapat menghasilkan keluaran rendah (*output low*), tetapi *chip* tidak

dapat menghasilkan keluaran tinggi (*output high*). Untuk mengatasi kekurangan dalam menghasilkan keluaran tinggi harus diberikan resistor *pull-up* yang dihubungkan ke sumber tegangan 5 Volt. Dengan demikian, jalur *SDA* dan *SCL* harus diberi resistor *pull-up* untuk seluruh *I2C bus*, tidak untuk setiap peralatan. Rangkaian *I2C Bus* disajikan pada Gambar 7.



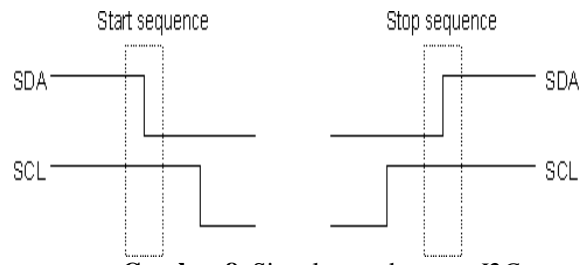
Gambar 7. *I2C Bus*

Nilai dari resistor tidak ditentukan. Nilai resistor berkisar antara 1k8 (1800 ohms) sampai 47k (47000 ohms). Nilai-nilai: 1k8, 4k7 dan 10k adalah nilai-nilai yang umum digunakan, tetapi nilai resistor pada ketiga kisaran tersebut seharusnya dapat bekerja dengan baik. Jika tidak terdapat resistor *pull up*, jalur *SCL* dan *SDA* akan mengambang dan *I2C bus* tidak akan mampu bekerja.

**Protokol *I2C***

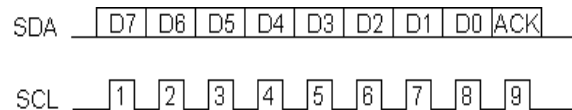
Peralatan pada *I2C bus* adalah kepala (*master*) atau bawahan (*slave*). *Master* selalu menggerakkan *SCL clock line*. *Slave* adalah peralatan yang merespon perintah-perintah dari *master*. Sebuah *slave* tidak dapat melakukan pengiriman melalui *I2C bus*, kecuali jika *master* memintanya. Jika terdapat banyak *slave* pada *I2C bus*, *master* hanya berjumlah satu. Pada alat ini, *master* adalah mikrokontroler dan yang bertindak sebagai *slave* adalah sensor kompas.

Ketika *master* memberikan perintah pada *slave*, maka akan terjadi *start sequence* pada *I2C bus*. *Start sequence* adalah satu dari dua deret tanda khusus untuk *I2C bus*, dan lainnya menjadi *stop sequence*. Ketika data dikirim, *SDA* harus tetap stabil dan tidak mengubah *SCL* menjadi *high*. *Start* dan *stop sequence* adalah tanda untuk mengawali dan mengakhiri koneksi ke *slave*. Sinyal *start* dan *stop sequence* disajikan pada Gambar 8.



Gambar 8. Sinyal *start* dan *stop I2C*

Data dikirim pada *sequences* dari 8 bit. Bit-bit ditempatkan pada *SDA line* dimulai dengan *Most Significant Bit (MSB)*. Pada saat pengiriman data *SCL line* membentuk pulsa *high* kemudian menjadi *low*. Untuk setiap 8 bit yang dikirim, peralatan akan menerima respon dari *slave* (sinyal *ACK = acknowledge*) jadi total 9 *SCL clock pulse*. Jika peralatan menerima respon dari *slave* berupa *ACK* dengan logika *low*, berarti *slave* siap untuk menerima data selanjutnya, tanpa harus diberikan sinyal *stop sequence* terlebih dulu. Jika peralatan menerima respon dari *slave* berupa *ACK* dengan logika *high*, berarti *slave* siap untuk menerima data selanjutnya setelah diberikan sinyal *stop sequence* terlebih dulu. Pengiriman data 8 bit oleh *SDA line* dan *SCL clock* disajikan pada Gambar 9.

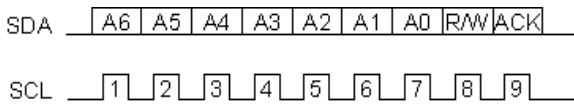


Gambar 9. Sinyal *SDA* dan *SCL* untuk data

**Pengalamanatan *I2C***

Semua alamat-alamat *I2C* adalah 7 bit atau 10 bit. Penggunaan 10 bit alamat sangat jarang dipergunakan. Semua *module* dan *chip-chip* umumnya menggunakan 7 bit untuk alamat, artinya dapat memiliki 128 peralatan pada *I2C bus*, dengan 7 bit diperoleh dari 0-127. Ketika mengirim 7 bit alamat, maka akan selalu mengirim 8 bit. Bit ekstra digunakan untuk memberitahukan *slave* jika *master* menulis untuk *slave* (*write*) atau membaca (*read*) dari *slave*. Jika bit *R/W* nol berarti *master* menulis untuk *slave*. Jika bit *R/W* satu itu artinya *master* membaca dari *slave*. Tujuh (7) bit alamat ditempatkan pada bagian atas *Most Significant Bit (MSB)*, 8 byte terkirim, dan *Read/Write (R/W)* bit pada *Least Significant Bit (LSB)*.

Pengalamatan pada *I2C bus* disajikan pada Gambar 10.



**Gambar 10.** Sinyal *SDA* dan *SCL* untuk alamat

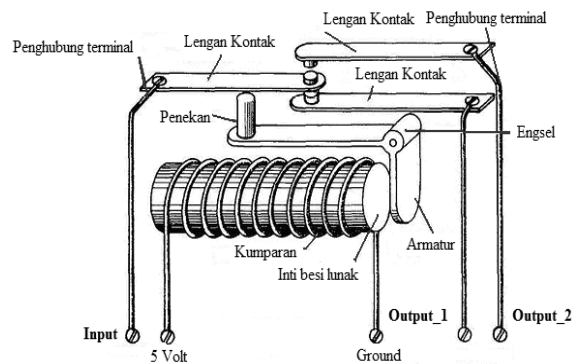
Sensor kompas *Devantech CMPS03* memiliki alamat  $0xC0$  ( $C0$  heksa). Pada *CMPS03* alamat  $C0$  heksa digunakan untuk menulis dan alamat  $C1$  heksa untuk membaca. Perbedaan alamat ini dikarenakan bit *R/W*. Untuk register-register fungsi yang terdapat pada *CMPS03* dapat dilihat pada Tabel 1.

**Tabel 1.** Register Fungsi *CMPS03*

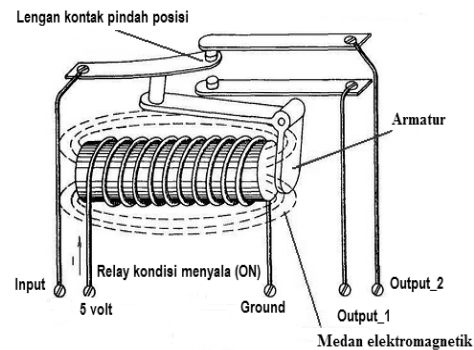
Register	Fungsi
0	<i>Software Revision Number</i>
1	<i>Compass Bearing as a byte, i.e. 0-255 for a full circle</i>
2,3	<i>Compass Bearing as a word, i.e. 0-3599 for a full circle.</i>
4,5	<i>Internal Test - Sensor1 difference signal - 16 bit word</i>
6,7	<i>Internal Test - Sensor2 difference signal - 16 bit word</i>
8,9	<i>Internal Test - Calibration value 1 - 16 bit word</i>
10,11	<i>Internal Test - Calibration value 2 - 16 bit word</i>
12	<i>Unused - Read as Zero</i>
13	<i>Unused - Read as Zero</i>
14	<i>Unused - Read as Undefined</i>
15	<i>Calibrate Command - Write 255 to calibrate</i>

### Relay

*Relay* merupakan komponen elektromagnetik yang dapat mengubah posisi lengan kontak input yang terhubung lengan *output\_1* pindah ke lengan kontak *output\_2* dengan memberikan medan elektromagnetik yang dibentuk dari kumparan kawat teraliri tegangan dan arus. Komponen *relay* terdiri dari: kumparan kawat, inti besi lunak, lengan kontak, penekan, engsel, dan armatur. Struktur *relay* dalam kondisi tidak teraliri arus listrik disajikan pada Gambar 11. Struktur *relay* dalam kondisi teraliri arus listrik disajikan pada Gambar 12.



**Gambar 11.** *Relay* Pada Kondisi Tidak Berarus Listrik

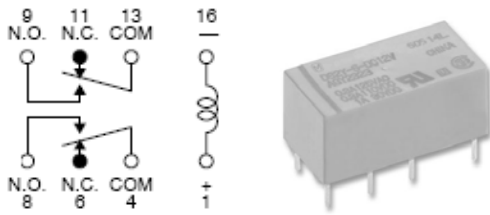


**Gambar 12.** *Relay* Pada Kondisi Berarus Listrik

Cara kerja *relay* adalah sebagai berikut: kawat kumparan dari *relay* yang teraliri arus listrik akan menimbulkan medan elektromagnetik dan membuat inti besi lunak termagnetisasi, akibatnya armatur akan tertarik oleh inti besi (kutub magnet yang sama), armatur akan menggerakkan penekan (pada penekan terdapat engsel untuk memudahkan pergerakan) untuk mendorong lengan kontak *input* yang tersambung dengan lengan kontak *output\_1* ke lengan kontak *output\_2*.

Ada 3 jenis kondisi *relay* yaitu:

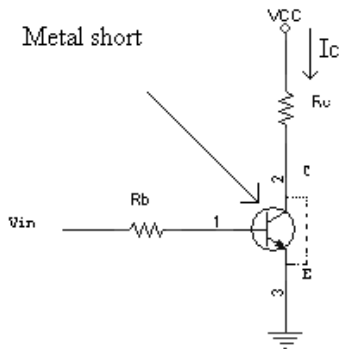
1. Normal terbuka (*Normally Open*), kondisi di mana *relay* tidak teraliri arus listrik pada kawat kumparan hubungan terbuka.
2. Normal tertutup (*Normally Close*), kondisi di mana *relay* tidak teraliri arus pada kawat kumparan hubungan tertutup.
3. *Change Over (CO)*, kondisi di mana *relay* memiliki sepasang lengan kontak *NO* dan *NC* dengan sebuah tuas kontak. Saat kawat kumparan dialiri arus listrik, kondisi *NO* akan tertutup dan *NC* akan terbuka atau sebaliknya sebagaimana ditunjukkan pada Gambar 13 berikut.



Gambar 13. Skema dan Bentuk Relay

**Transistor**

Pada rangkaian *driver motor*, transistor berfungsi sebagai saklar, daerah kerjanya berada pada daerah *cutoff* atau berada pada daerah saturasi. Bias yang digunakan adalah bias basis. Pada transistor, yang dimaksud dengan titik saturasi adalah arus maksimum kolektor yang dapat dilewatkan. Titik saturasi didapat dengan cara menghubungkan secara singkat antara kolektor dan *emitor* seperti yang ditunjukkan pada Gambar 14 berikut.

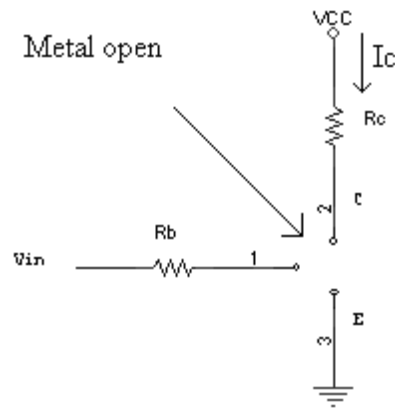


Gambar 14. Transistor Saat Keadaan Saturasi

Pada saat transistor dalam keadaan saturasi/jenuh adalah tegangan  $V_{ce} \sim 0,1$ , maka tegangan  $V_{cc}$  akan melewati resistor  $R_c$ , sehingga didapat arus  $I_c$  (arus  $I_c$  adalah arus  $I_c$  saturasi) dengan rumus:

$$I_c(sat) = V_{cc} / R_c \quad (1)$$

Pada kondisi di mana terjadi *cutoff*, didapat tegangan  $V_{ce}$  maksimum yaitu sama dengan tegangan yang berada pada kolektor, seperti yang ditunjukkan pada Gambar 15 berikut.



Gambar 15. Transistor Saat Keadaan Cutoff

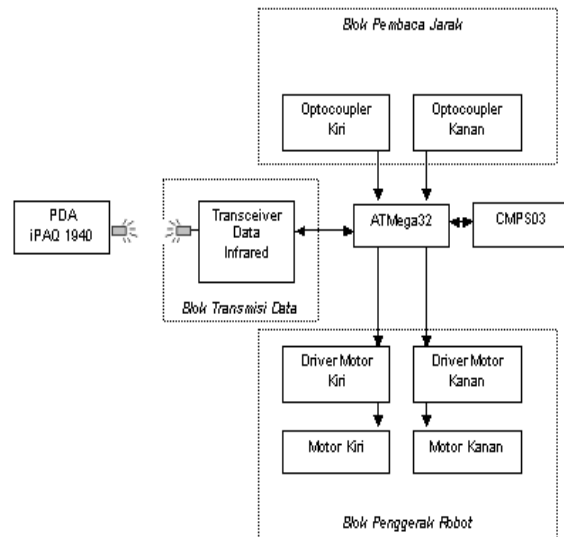
Dengan demikian pada saat keadaan *cutoff* dapat dituliskan persamaan sebagai berikut:

$$V_{ce}(cutoff) = V_{cc} \quad (2)$$

**METODE PENELITIAN**

**Perancangan Sistem**

Dalam merancang suatu alat, langkah pertama yang dilakukan adalah membuat diagram blok yang menggambarkan kerja masing-masing bagian dari alat tersebut secara garis besar. Diagram blok dari robot beroda yang berjalan menurut jalur yang digambar pada *personal digital assistant* sebagaimana disajikan pada Gambar 16.



Gambar 16. Diagram Blok Robot Beroda Yang Berjalan Menurut Jalur Yang Digambar Pada PAD

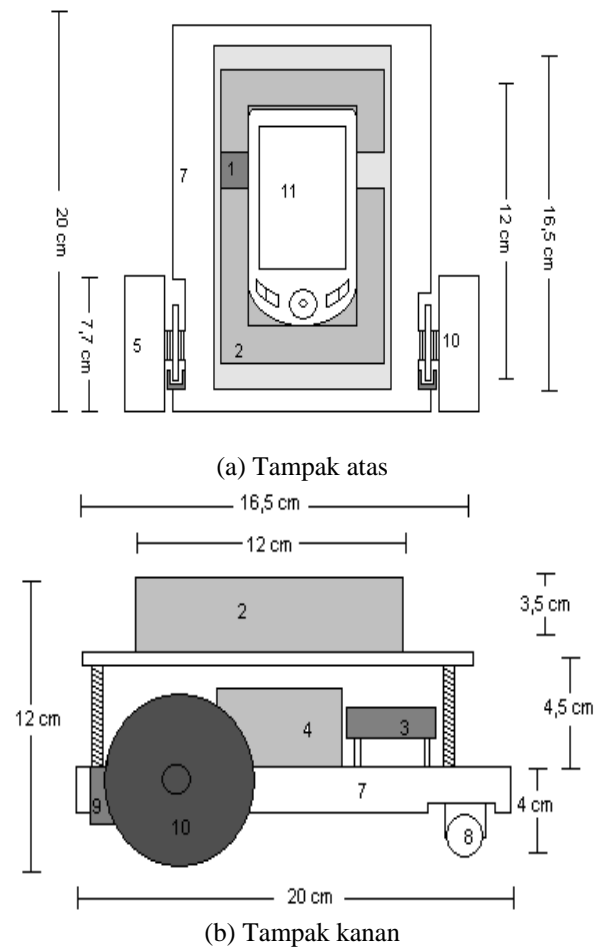
Penjelasan diagram blok pada Gambar 16 adalah sebagai berikut:

1. *PDA* iPAQ 1940  
Bagian ini digunakan sebagai otak dari robot. *PDA* berfungsi menerima *input* berupa jalur yang diberikan oleh operator, lalu bersama dengan mikrokontroler mengkoordinasikan gerak motor kanan dan kiri. Selain itu *PDA* iPAQ 1940 juga digunakan untuk menghitung sudut putaran yang diperlukan.
2. *Transceiver Data Infrared*  
Bagian ini adalah *transmitter* (pengirim)-*receiver* (penerima) data melalui inframerah. Bagian ini digunakan sebagai jembatan komunikasi antara mikrokontroler AT89S51 dengan *PDA* iPAQ 1940.
3. *ATMega32*  
*ATMega32* adalah mikrokontroler yang berfungsi menjalankan motor kiri atau kanan sesuai dengan perintah yang diterima dari *PDA* iPAQ 1940. Selain itu juga berfungsi untuk mengirim data yang diterima dari *optocoupler* kiri dan *optocoupler* kanan ke *PDA* iPAQ 1940.
4. *Optocoupler Kiri*  
*Optocoupler* kiri berfungsi sebagai sensor pengukur jarak yang telah ditempuh oleh roda kiri. Pada rangkaian ini digunakan sensor SHARP GP1A51HR<sup>[5]</sup> yang membaca lubang yang ada pada piringan yang menempel pada sumbu roda kiri.
5. *Optocoupler Kanan*  
*Optocoupler* kanan berfungsi sebagai sensor pengukur jarak yang telah ditempuh oleh roda kanan. Pada rangkaian ini digunakan sensor SHARP GP1A51HR yang membaca lubang yang ada pada piringan yang menempel pada sumbu roda kanan.
6. *Driver Motor Kiri*  
*Driver Motor Kiri* adalah rangkaian yang berfungsi mengaktifkan atau mematikan atau mengatur arah maju-mundur pada motor *DC* kiri.
7. *Driver Motor Kanan*  
*Driver motor* kanan adalah rangkaian yang berfungsi mengaktifkan atau mematikan atau mengatur arah maju-mundur pada motor *DC* kanan.
8. *Motor Kiri*  
*Motor Kiri* adalah motor *DC* kiri yang menyalakan atau mematikan atau mengatur maju-mundur sesuai dengan keluaran dari *driver* motor kiri.

9. *Motor Kanan*  
*Motor* kanan adalah motor *DC* kanan yang menyalakan atau mematikan atau mengatur gerak maju-mundur sesuai dengan keluaran dari *driver* motor kanan.
10. *CMPS03*  
*CMPS03* berfungsi sebagai pengukur sudut digital.

### Perancangan Bentuk Alat

Dengan mengacu pada pertimbangan bahwa bodi robot harus berdimensi kompak dan ringan agar lincah bermanuver, maka alat dibuat dengan menggunakan bahan berupa aluminium. Bentuk alat yang dibuat disajikan pada Gambar 17.



**Gambar 17.** Bentuk alat yang dibuat



### Perancangan Perangkat Keras

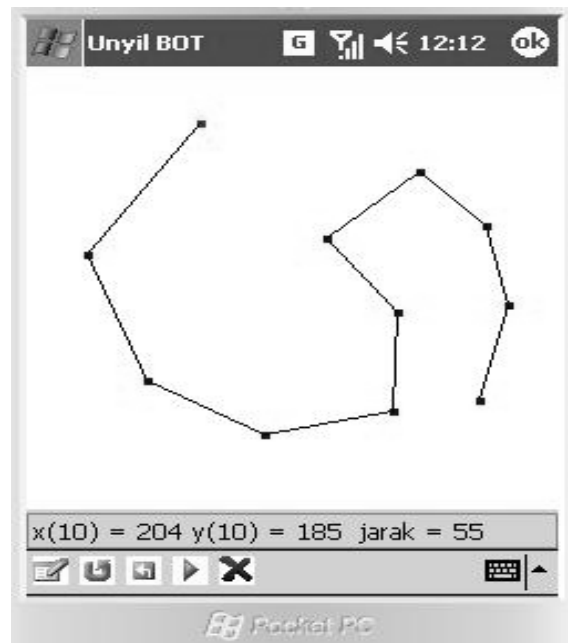
Komponen-komponen yang digunakan dalam pembuatan robot beroda yang berjalan menurut jalur yang digambar pada *PAD* disajikan pada Tabel 2 sebagai berikut:

**Tabel 2.** Komponen-komponen Robot Beroda Yang Berjalan Menurut Jalur Yang Digambar Pada *PAD* Secara Umum

No	Jenis Rangkaian	Keterangan
1	<i>Minsys</i> Mikrokontroler	ATMega32
2	<i>Driver</i> Motor Kanan	Transistor dan <i>relay</i>
3	<i>Driver</i> Motor Kiri	Transistor dan <i>relay</i>
4	<i>Optocoupler</i> Kanan	<i>Sensor Sharp</i> OPIC GP1A51HR
5	<i>Optocoupler</i> Kiri	<i>Sensor Sharp</i> OPIC GP1A51HR
6	<i>Transceiver</i> inframerah	<i>Transmitter</i> menggunakan IC 74HCT132 dan transistor CN538, <i>Receiver</i> menggunakan <i>receiver</i> infra merah dengan demodulator

### Perancangan Perangkat Lunak

Perangkat lunak yang digunakan untuk memprogram ATMega32 adalah CodeVision AVR dari ATMEL, sedangkan untuk memprogram *PDA* iPAQ 1940 digunakan *Microsoft Visual Studio.NET Compact Framework*<sup>[6]</sup>. Tampilan program dari perangkat lunak disajikan pada Gambar 18.



**Gambar 18.** Tampilan Program Dari Perangkat Lunak Penggerak Robot

### Cara Kerja Robot

- Operator mula-mula menggambar jalur pada layar *PDA* iPAQ 1940 dengan cara menggambar titik jalur pada layar, lalu menggambar titik jalur lainnya. Antar titik jalur tersebut dengan titik jalur sebelumnya akan terhubung secara otomatis dengan sebuah garis. Proses tersebut akan berulang-ulang sampai operator menekan tombol "SELESAI". Titik jalur yang pertama kali dianggap sebagai titik "START" dan titik jalur yang terakhir kali tergambar dianggap sebagai titik "END";
- Program lalu menghitung jarak yang harus ditempuh oleh robot untuk mencapai tiap-tiap titik jalur. Apabila ada jalur yang berupa tikungan, maka program akan menghitung sudut putaran yang harus diambil oleh robot untuk memutar bodi;
- Robot lalu mulai berjalan sesuai dengan jalur yang telah digambar sepanjang  $x$  sentimeter, dimana  $x$  adalah panjang jalur yang harus dilalui oleh robot yang telah dihitung oleh *PDA* iPAQ 1940. Proses tersebut akan berulang-ulang sampai kesemua jalur telah selesai dilewati (robot mencapai titik *END*).
- Robot lalu berhenti bekerja.

## HASIL PENELITIAN DAN PEMBAHASAN

### Hasil Penelitian

#### 1. Uji Coba Alat di Medan Aspal (kasar) dan Bebas Gangguan Magnetik

Untuk mengetahui kinerja dari alat yang telah dibuat perlu dilakukan pengujian di medan kasar berupa aspal yang bebas dari gangguan elektromagnetik. Hasil pengujian tersebut dalam bentuk hubungan antara jarak yang diinginkan terhadap jarak sebenarnya ditempuh oleh alat (robot) sebagaimana disajikan pada Tabel 3 dan hubungan antara sudut yang diinginkan terhadap sudut aktual yang ditempuh oleh alat (robot) sebagaimana disajikan pada Tabel 4.

**Tabel 3.** Hubungan antara jarak yang diinginkan terhadap jarak sebenarnya yang ditempuh oleh alat (robot) di medan aspal dan bebas gangguan magnetik

No	Jarak yang diinginkan (cm)	Jarak sebenarnya (cm)	Selisih (cm)
1	50	51	1
2	50	52	2
3	50	52	2
4	50	51	1
5	50	50	0
Rata-rata total selisih ( <i>error</i> )= 6/5			1,2 cm

**Tabel 4.** Hubungan antara sudut yang diinginkan terhadap sudut sebenarnya yang ditempuh oleh alat (robot) di medan aspal dan bebas gangguan magnetik

No	Sudut yang diinginkan (derajat)	Sudut sebenarnya (derajat)	Selisih (derajat)
1	90	95	5
2	90	102	12
3	90	98	8
4	90	96	6
5	90	99	9
Rata-rata total selisih ( <i>error</i> )= 40/5			8 derajat

#### 2. Uji Coba Alat di Medan Aspal (kasar) dan Terdapat Gangguan Magnetik

Untuk mengetahui efek dari gangguan magnetik terhadap alat ini. Hasil pengujian tersebut juga disajikan dalam bentuk hubungan antara jarak yang diinginkan terhadap jarak sebenarnya ditempuh oleh alat (robot) sebagaimana disajikan pada Tabel 5 dan hubungan antara sudut yang diinginkan terhadap sudut aktual yang ditempuh oleh alat (robot) sebagaimana disajikan pada Tabel 6.

**Tabel 5.** Hubungan antara jarak yang diinginkan terhadap jarak sebenarnya yang ditempuh oleh alat (robot) di medan aspal (kasar) dan terdapat gangguan magnetik

No	Jarak yang diinginkan (cm)	Jarak sebenarnya (cm)	Selisih (cm)
1	50	52	2
2	50	51	1
3	50	52	2
4	50	51	1
5	50	51	1
Rata-rata total selisih ( <i>error</i> )=7/5			1,4 cm

**Tabel 6.** Hubungan antara sudut yang diinginkan terhadap sudut sebenarnya yang ditempuh oleh alat (robot) di medan aspal (kasar) dan terdapat gangguan magnetik

No	Sudut yang diinginkan (derajat)	Sudut sebenarnya (derajat)	Selisih (derajat)
1	90	290	200
2	90	250	160
3	90	284	194
4	90	305	215
5	90	218	128
Rata-rata total selisih ( <i>error</i> )= 897/5			179,4 (derajat)

#### 3. Uji Coba Alat di Medan Lantai Marmer (licin) dan Bebas Gangguan Magnetik

Selain medan aspal, juga perlu dilakukan pengujian di medan licin berupa lantai marmer yang bebas dari gangguan elektromagnetik. Hasil pengujian tersebut juga disajikan dalam bentuk hubungan antara jarak yang diinginkan terhadap jarak sebenarnya ditempuh oleh alat (robot) sebagaimana disajikan pada Tabel 7 dan hubungan antara sudut yang diinginkan terhadap sudut aktual yang ditempuh oleh alat (robot) sebagaimana disajikan pada Tabel 8.

**Tabel 7.** Hubungan antara jarak yang diinginkan terhadap jarak sebenarnya yang ditempuh oleh alat (robot) di medan lantai marmer dan bebas gangguan magnetik

No	Jarak yang diinginkan (cm)	Jarak sebenarnya (cm)	Selisih (cm)
1	50	54	4
2	50	55	5
3	50	54	4
4	50	55	5
5	50	55	5
Rata-rata total selisih ( <i>error</i> )= 23/5			4,6 cm

**Tabel 8.** Hubungan antara sudut yang diinginkan terhadap sudut sebenarnya yang ditempuh oleh alat (robot) di medan lantai marmer dan bebas gangguan magnetik

No	Sudut yang diinginkan (derajat)	Sudut aktual (derajat)	Selisih (derajat)
1	90	100	10
2	90	107	17
3	90	103	13
4	90	106	16
5	90	101	11
Rata-rata total selisih ( <i>error</i> )= 67/5			13,4 derajat

#### 4. Uji Coba Alat di Medan Lantai Marmer (licin) dan Terdapat Gangguan Magnetik

Untuk lantai marmer yang licin dan terdapat gangguan magnetik, hasil pengujian alat disajikan pada Tabel 9 dan 10 sebagai berikut:

**Tabel 9.** Hubungan antara jarak yang diinginkan terhadap jarak sebenarnya yang ditempuh oleh alat (robot) di medan lantai marmer dan terdapat gangguan magnetik

No	Jarak yang diinginkan (cm)	Jarak sebenarnya (cm)	Selisih (cm)
1	50	56	6
2	50	55	5
3	50	55	5
4	50	56	6
5	50	54	4
Rata-rata total selisih ( <i>error</i> )= 26/5			5,2 cm

**Tabel 10.** Hubungan antara sudut yang diinginkan terhadap sudut sebenarnya yang ditempuh oleh alat (robot) di medan lantai marmer dan terdapat gangguan magnetik

No	Sudut yang diinginkan (derajat)	Sudut aktual (derajat)	Selisih (derajat)
1	90	309	219
2	90	207	117
3	90	293	203
4	90	6	84
5	90	349	259
Rata-rata total selisih ( <i>error</i> )= 882/5			176,4 derajat

Dari uji coba di atas dapat dirangkum kesalahan pengukuran dari alat yang disajikan pada Tabel 11 dan 12 sebagai berikut:

**Tabel 11.** Hubungan antara kondisi uji coba alat terhadap rata-rata kesalahan untuk jarak yang ditempuh oleh alat

Medan	Rata-rata kesalahan (cm)
Kasar, bebas gangguan	1,2
Kasar, ada gangguan	1,4
Licin, bebas gangguan	4,6
Kasar, ada gangguan	5,2
Total rata-rata kesalahan	12,4

**Tabel 12.** Hubungan antara kondisi uji coba alat terhadap rata-rata kesalahan untuk sudut yang ditempuh oleh alat

Medan	Rata-rata kesalahan (derajat)
Kasar, bebas gangguan	8
Kasar, ada gangguan	179,4
Licin, bebas gangguan	13,4
Kasar, ada gangguan	176,4
Total rata-rata kesalahan	377,2

#### Pembahasan

Dari hasil pengujian di atas, terdapat sejumlah kesalahan, baik pada pengukuran jarak maupun pada sudut putaran. Hal-hal yang dapat menyebabkan terjadinya kesalahan tersebut antara lain:

- Penggunaan motor *DC* yang bekas, sehingga mempengaruhi kecepatan motor kiri dan kanan yang berbeda.
- Permukaan roda yang licin, sehingga membuat titik pengereman ideal menjadi bergeser.
- Permukaan lantai yang licin, sehingga membuat roda kurang mendapatkan *grip* (tekanan) ke bawah.
- Adanya gangguan elektromagnetik dari lingkungan sekitar (adanya *speaker*, komputer, dan sebagainya) yang mengganggu sensor kompas dalam melakukan pembacaan sudut.

#### KESIMPULAN

Dari hasil penelitian dan pembahasan dapat ditarik kesimpulan sebagai berikut:

- Penggunaan sensor kompas dapat lebih meningkatkan ketelitian robot dalam melakukan putaran dengan sudut tertentu.
- Dari pengujian sistem secara keseluruhan yang dilakukan, diperoleh rata-rata kesalahan dalam jarak tempuh yaitu sebesar 12,4 cm dan rata-rata kesalahan dalam

melakukan putaran yaitu sebesar 377,2 derajat.

#### **DAFTAR PUSTAKA**

- [1] Atmel, *ATMega32 Mikrocontroller Datasheet*
- [2] Anonim, *Komunikasi Inframerah dengan DST51*, <http://www.delta.electronic.com/Design/Anote/Infrared.html>, diakses 20 Juli 2007
- [3] Anonim, *iPAQ Headquarters*, <http://www.ipaqhq.com>, diakses 20 Juli 2007
- [4] Anonim, *GP1A51HR DataSheet*, Sharp
- [5] Anonim, *Magnetic Compass CMPS03 Datasheet*, Devantech
- [6] Anonim, *Open Source Dot Net Compact Framework Library*, <http://opennetcf.org/library/>, Open Source diakses 20 Juli 2007