BAB I
PENDAHULUAN

Pada bab ini akan diuraikan mengenai latar belakang, identifikasi masalah, tujuan, pembatasan masalah, spesifikasi alat, dan sistematika penulisan.

1.1. Latar Belakang

Seiring dengan kemajuan budaya masyarakat yang menuntut mobilitas serta didukung dengan berbagai perkembangan teknologi komunikasi dan otomasi, memungkinkan untuk menciptakan perangkat yang dapat mendukung kinerja manusia dalam melaksanakan proses pekerjaan agar lebih praktis dan efisien. Salah satu caranya adalah dengan meminimalkan tingkat keberadaan manusia di lokasi proses pekerjaan tersebut berlangsung dan membuat suatu instrumen berupa robot yang dapat dimanfaatkan untuk melakukan pekerjaan tersebut. Untuk mengendalikan robot diperlukan pengendali sebagai otak dari robot tersebut.

Pada teknologi robot saat ini permasalahan *interfacing* untuk mengendalikan robot dapat menjadi persoalan tersendiri. Sedangkan penggunaan media kabel untuk mengendalikan robot dirasakan kurang efektif, dengan keterbatasan dari panjang kabel itu sendiri. Oleh karena itu, media *wireless* dapat menjadi pilihan yang lebih efektif. Untuk mengendalikan robot secara jarak jauh dengan media *wireless* diperlukan suatu modul *wireless interface* sehingga terjadi komunikasi antara user dengan robot yang dikendalikan tersebut.

1.2. Identifikasi Masalah

Permasalahan pada Tugas Akhir ini adalah bagaimana merancang dan merealisasikan suatu sistem *interface* untuk mentransmisikan sinyal kendali dari komputer ke robot dan dari robot ke komputer melalui gelombang radio 2.4 GHz.

1.3. Tujuan

Tujuan dari Tugas Akhir ini adalah merancang dan merealisasikan suatu sistem *interface* untuk mentransmisikan sinyal kendali dari komputer ke robot dan dari robot ke komputer melalui gelombang radio 2.4 GHz.

1.4. Pembatasan Masalah

Adapun pembatasan masalah dalam tugas akhir ini dalah sebagai berikut :

1. Sistem *Interface* memiliki 30 *Input*/*Output* digital dan yang digunakan pada Tugas Akhir ini adalah sebanyak 4 *Input* serta 8 *Output.*
2. Sistem *Interface* digunakan untuk mengendalikan mekanik lengan robot buatan Fischertechnic yang memiliki empat buah motor DC.
3. Sistem *interface* tidak menggunakan fasilitas keamanan pada komunikasi data.
4. Protokol yang digunakan pada sistem *interface* adalah protokol TCP/IP.
5. Perancangan dan realisasi sistem *interface* tidak menggunakan prosedur / metode *handshaking (flow control).*

1.5 Spesifikasi Alat

Alat – alat yang digunakan antara lain :

1. PC/Notebook
2. WiFi Card/WiFi Adapter.
3. Accesss Point D-Link DI-524.
4. Kabel Ethernet konektor RJ-45
5. Ethernet to Serial Gateway (EGSR-7150MJ).
6. Mikrokontroler AVR ATMega 16.

1.6 Sistematika Penulisan

Sistematika penulisan dalam Tugas Akhir ini adalah sebagai berikut :

Bab I Pendahuluan

Menguraikan tentang latar belakang, perumusan masalah, tujuan, pembatasan masalah, spesifikasi alat yang akan digunakan dalam Tugas Akhir ini.

Bab II Landasan Teori

Pada bab ini akan dibahas tentang teori – teori yang digunakan dalam perancangan dan pembuatan perangkat keras dan lunak meliputi ATMega 16, bahasa C, visual basic, komunikasi data, windows socket (winsock), ethernet to serial gateway (EGSR-7150MJ).

Bab III Perancangan dan Pembuatan Perangkat Keras dan Lunak

Pada bab ini akan dibahas secara lengkap tentang perancangan dan pembuatan perangkat keras dan perangkat lunak sistem *interface* pada tugas akhir ini.

Bab IV Pengujian dan Analisa Data

Pada bab ini akan dibahas tentang pengujian dan analisis dari sistem *interface* yang telah dibuat.

Bab V Penutup

Bab ini berisi tentang kesimpulan dan saran berdasarkan hasil analisis sistem *interface* yang telah dibuat.

BAB II
LANDASAN TEORI

Pada bab ini akan dibahas tentang teori dasar mikrokontroler AVR ATMega 16, bahasa C, visual basic, komunikasi data, windows socket (winsock), ethernet to serial gateway (EGSR-7150MJ).

2.1 Mikrokontroler AVR ATMega 16

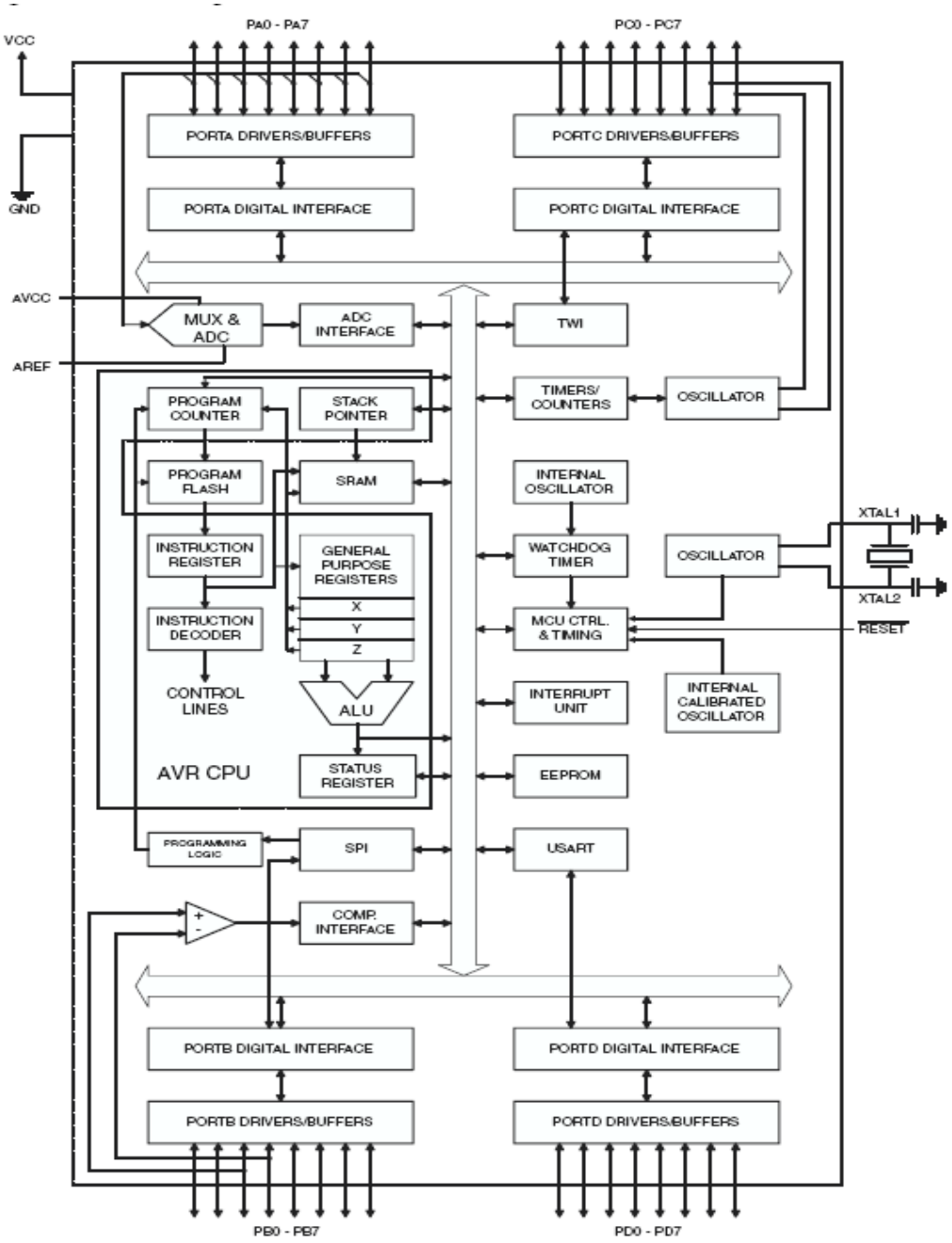
Mikrokontroler adalah suatu chip yang dapat digunakan sebagai pengontrol utama sistem elektonika. Hal ini dikarenakan di dalam chip tersebut sudah ada unit pemroses, memory ROM (*Read Only Memory*), RAM (*Random Access Memory*), *Input – Output,* dan fasilitas pendukung lainnya.

Saat ini, sudah banyak pemula dan praktisi yang beralih ke mikrokontroler AVR, dibandingkan mikrokontroler keluarga MCS51. Keterbatasan pada mikrokontoler tersebut (resolusi, memory, dan kecepatan) menyebabkan banyak orang beralih ke mikrokontoler AVR. Hal ini dikarenakan kelebihan yang dimilikinya, diantaranya pada beberapa tipe AVR sudah terdapat ADC internal, ukuran memory yang lebih besar, kecepatan eksekusi instruksi, dan dukungan software yang dapat digunakan di dalam chip tersebut.

AVR dapat dikelompokan menjadi empat kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATMega, dan keluargaAT86RFxx. Pada dasarnya yang membedakan masing – masing kelas adalah memory, peripheral dan fungsinya. Mikrokontroler yang digunakan dalam Tugas Akhir ini adalah ATMega 16.

2.1.1 Arsitektur AVR ATMega 16

Mikrokontroler yang digunakan dalam Tugas Akhir ini adalah ATMega 16. AVR ATMega 16 merupakan seri Mikrokontroler CMOS 8-bit dengan 16 Kbyte programable flash yang berbasis pada arsitektur RISC (Reduced Instruction Set Computers). Kebanyakan instruksi dikerjakan pada satu siklus clock, ATMega 16 mempunyai *throughput* mendekati 1 MIPS per MHz. Blok diagram dari mikrokontroler dapat dilihat pada Gambar 2.1



Gambar 2.1 Blok Diagram ATMega 16

AVR mempunyai 16K bytes *of In-System Programmable Flash on chip* yang membolehkan memori program untuk diprogram ulang. Selain itu AVR ATmega 16 memiliki beberapa keistimewaan sebagai berikut :

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution

- 32 x 8 General Purpose Working Registers
- Fully Static Operation
- Up to 16 MIPS Throughput at 16 MHz
- On-chip 2-cycle Multiplier

• Nonvolatile Program and Data Memories

- 16K Bytes of In-System Self-Programmable Flash Endurance: 10,000 Write/Erase Cycles
- Optional Boot Code Section with Independent Lock Bits In-System Programming by On-chip Boot ProgramTrue Read-While-Write Operation
- 512 Bytes EEPROM Endurance: 100,000 Write/Erase Cycles
- 1K Byte Internal SRAM
- Programming Lock for Software Security

• Peripheral Features

- Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
- Real Time Counter with Separate Oscillator
- Four PWM Channels
- 8-channel, 10-bit ADC
  - 8 Single-ended Channels
  - 7 Differential Channels in TQFP Package Only
  - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
- Byte-oriented Two-wire Serial *Interface*
- Programmable Serial USART
- Master/Slave SPI Serial *Interface*
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator

• Special Microcontroller Features

- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated RC Oscillator
- External and Internal Interrupt Sources

- Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standbyand Extended Standby

• I/O and Packages

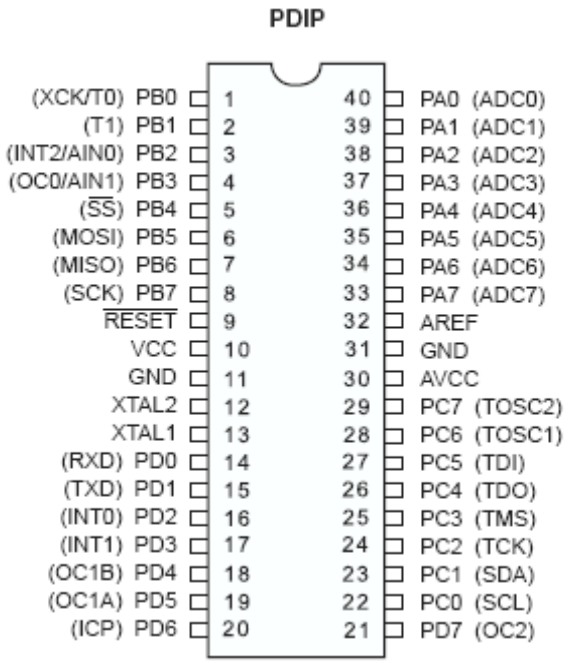
- 32 Programmable I/O Lines
- 40-pin PDIP, 44-lead TQFP, and 44-pad MLF

• Operating Voltages

- 2.7 - 5.5V for ATmega16L
- 4.5 - 5.5V for ATmega16

2.1.2 Deskripsi Pin/Kaki pada AVR ATMega 16

Konfigurasi *Pinout* pada mikrokontroler AVR ATMega 16 dengan kemasan 40-Pin DIP (*Dual In line Package*) dapat dilihat pada Gambar 2.2.



Gambar 2.2 Konfigurasi Pin ATMega 16

Berikut adalah susunan pin/kaki dari ATMega 16.

- VCC *(Digital power supply)* merupakan pin masukan positif catu daya.
- GND *(ground)* sebagai pin Ground.

- Port A (PA7..PA0)

  Port A berfungsi sebagai *input* analog pada A/D Konverter. Port A juga berfungsi sebagai suatu Port I/O 8-bit dua arah, jika A/D Konverter tidak digunakan. Pin - pin Port dapat menyediakan resistor *internal pull-up* (yang dipilih untuk masing-masing bit).

- Port B (PB7..PB0)

  Port B adalah suatu Port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit).

- Port C (PC7..PC0)

  Port C adalah suatu Port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit).

- Port D (PD7..PD0)

  Port D adalah suatu Port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Pada Port D juga terdapat fasilitas komunikasi serial dan PWM yang disediakan oleh mikrokontroler ATMega 16.

- RESET *(Reset input)* merupaka pin yang digunakan untuk me-reset mikrokontoler.
- XTAL1 *(Input Oscillator)* dan XTAL2 *(Output Oscillator)* sebagai pin masukan clock eksternal.
- AVCC adalah pin penyedia tegangan untuk port A dan A/D Konverter (ADC)
- AREF adalah pin masukan tegangan referensi analog untuk A/D konverter (ADC) pada Port A.

2.1.3 Port Sebagai *Input*/*Output* Digital

ATMega 16 mempunyai empat buah port yang bernama PortA, PortB, PortC, dan PortD. Keempat port tersebut merupakan jalur *bi-directional* dengan pilihan *internal pull-up*. Tiap port mempunyai tiga buah register bit, yaitu DDxn, PORTxn, dan PINxn. Huruf 'x' mewakili nama huruf dari port sedangkan huruf 'n' mewakili nomor bit. Bit DDxn terdapat pada I/O address DDRx, bit PORTxn terdapat pada I/O address PORTx, dan bit PINxn terdapat pada I/O address PINx. Bit DDxn dalam regiter DDRx (Data Direction Register) menentukan arah pin. Bila DDxn diset 1 maka Px berfungsi sebagai pin *output*. Bila DDxn diset 0 maka Px berfungsi sebagai pin *input*. Bila PORTxn diset 1 pada saat pin terkonfigurasi sebagai pin *input*, maka resistor pull-up akan diaktifkan. Untuk mematikan resistor pull-up, PORTxn harus diset 0 atau pin dikonfigurasi sebagai pin *output*. Pin port adalah tri-state setelah kondisi reset. Bila PORTxn diset 1 pada saat pin terkonfigurasi sebagai pin *output* maka pin port akan berlogika 1. Saat PORTxn diset 0 pada saat pin terkonfigurasi sebagai pin *output* maka pin port akan berlogika 0. Lebih detil mengenai port ini dapat dilihat pada manual datasheet dari IC ATMega 16. Konfigurasi Pin Port dapat dilihat pada Tabel II.1

Tabel II.1 Konfigurasi Pin Port

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

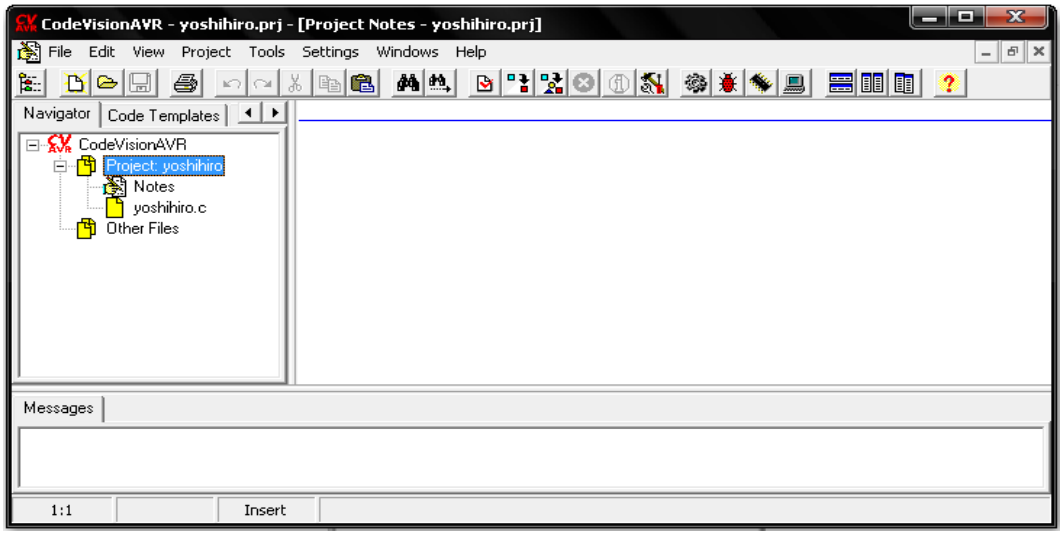
2.1.4 Pemrograman pada AVR ATMega 16

Pemrograman yang digunakan untuk mengisi program pada mikrokontroler AVR ini digunakan CodeVision AVR. CodeVisionAVR merupakan *software C- cross compiler*, program dapat ditulis menggunakan bahasa-C. Dengan menggunakan pemrograman bahasa-C diharapkan waktu disain *(developing time)* akan menjadi lebih singkat. Setelah program dalam bahasa-C

ditulis dan dilakukan kompilasi tidak terdapat kesalahan (error) maka proses download dapat dilakukan. Mikrokontroler AVR mendukung sistem download secara *ISP (In-System Programming).*

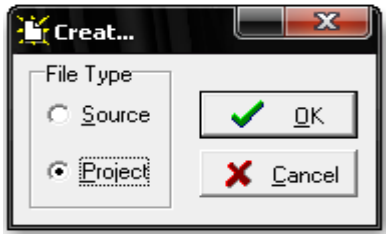
Selain itu, pada CodeVision AVR ini bisa ditentukan port-port dari mikrokontroler AVR yang berfungsi sebagai *input* maupun *output*, serta bisa juga ditentukan tentang penggunaan fungsi-fungsi internal dari AVR. Sebelum menentukan port-port dan fungsi-fungsi internal yang akan digunakan, harus ditentukan terlebih dahulu mikrokontroler yang akan dipakai. Masing - masing mikrokontroler mempunyai perbedaan dalam fungsi-fungsi internal. Cara memulai project baru pada CodeVision AVR adalah sbb :

1. Jalankan software CodeVision AVR seperti pada Gambar 2.3.



Gambar 2.3 CodeVision AVR

2. Buat project baru kemudian pilih File, New, lalu pilih Project seperti pada Gambar 2.4

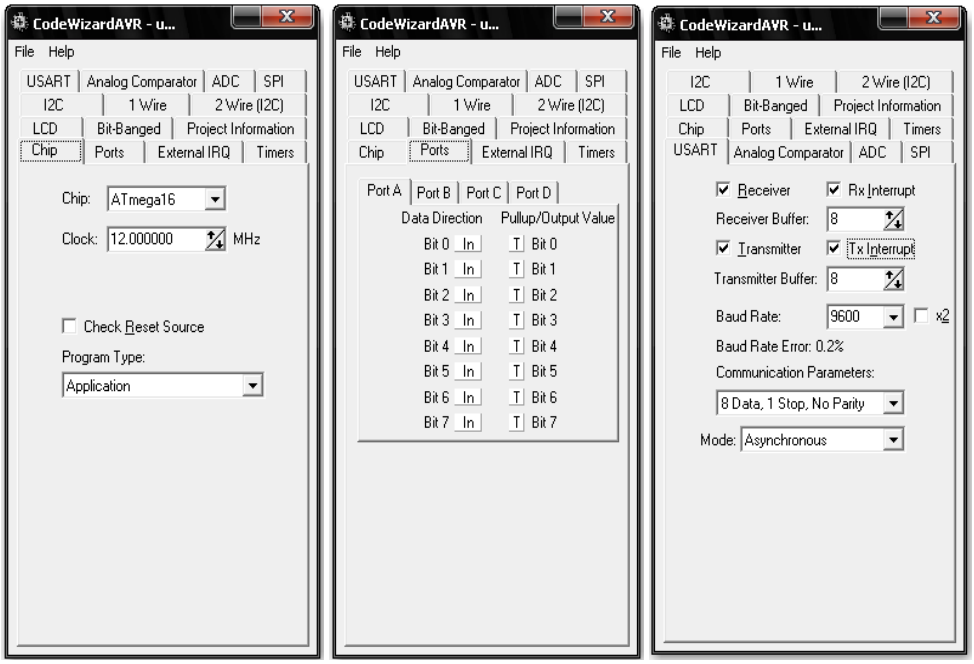


Gambar 2.4 Create Project Baru

3. Ketika muncul kotak dialog apakah akan menggunakan codevision wizard dalam project baru. Pilih Yes seperti terlihat pada Gambar 2.5.

Gambar 2.5 Pilihan Untuk Menggunakan Codewizard AVR

4. Lalu selanjutnya pada kotak codewizard, terdapat pilihan untuk menentukan mikrokontroler yang akan dipakai. Selain itu juga dapat menentukan port – port dan fungsi – fungsi internal mikrokontoller yang akan digunakan seperti : komunikasi serial USART, LCD, PWM dan lain – lain seperti yang terlihat pada Gambar 2.6.



Gambar 2.6 Konfigurasi Pada Codewizard AVR

2.2 Bahasa C

Akar bahasa C adalah dari bahasa BCPL yag dikembangkan oleh Martin Richards pada tahun 1967. Bahasa ini memberikan ide kepada Ken Thompson yang kemudian mengembangkan bahasa yang disebut bahasa B pada tahun 1970.

Perkembangan selanjutnya dari bahasa B adalah bahasa C oleh Dennis Ritchie sekitar tahun 1970-an di Bell Telephone Laboratories Inc.

C adalah bahasa yang standar, artinya suatu program yang ditulis dengan versi bahasa C tertentu akan dapat dikompilasi dengan versi bahasa C yang lain dengan sedikit modifikasi.

Beberapa alasan bahasa C banyak digunakan, diantaranya adalah sebagai berikut :

1. Bahasa C tersedia hampir di semua jenis komputer.
2. Kode bahasa C sifatnya portabel.
3. Bahasa C hanya menyediakan sedikit kata – kata kunci.
4. Proses *executable program* bahasa C lebih cepat.
5. Dukungan Pustaka yang banyak.
6. C adalah bahasa yang terstruktur.
7. Selain bahasa tingkat tinggi, C juga dianggap sebagai bahasa tingkat Menengah.
8. Bahasa C adalah kompiler .

2.2.1 Struktur Penulisan Bahasa C

Untuk dapat memahami cara suatu program ditulis, maka struktur dari program harus dimengerti terlebih dahulu. Tiap bahasa komputer mempunyai struktur program yang berbeda. Struktur dari program memberikan gambaran secara luas, bagaimana bentuk dari program secara umum.

Struktur dari program C dapat dilihat sebagai kumpulan dari sebuah atau lebih fungsi – fungsi. Fungsi pertama yang harus ada di program C sudah ditentukan namanya, yaitu bernama main(). Suatu fungsi di program C dibuka dengan kurung kurawal ({) dan ditutup dengan kurung kurawal tertutup (}). Diantara kurung kurawal dapat dituliskan statemen – statemen program C. Berikut ini adalah struktur dari program C.

```
main()
{
    statemen – statemen;
}
Fungsi_Fungsi_Lain()
{
    statemen-statemen;
}
```

Fungsi Utama

Fungsi – fungsi lain yang ditulis oleh pemrograman komputer

2.2.2 Dasar – Dasar Pemrograman C

Dasar – dasar pemrograman C sangat diperlukan dalam pemrograman mikroontroler. Oleh karena itu akan dijelaskan secara ringkas dasar – dasar pemrograman C.

2.2.2.1 Tipe Data Dasar

Data merupakan suatu nilai yang biasa dinyatakan dalam bentuk konstanta atau variabel. Konstanta menyatakan nilai yang tetap, sedangkan variabel menyatakan nilai yang dapat diubah – ubah selama eksekusi berlangsung.

Data berdasarkan jenisnya dapat dibagi menjadi lima kelompok, yang dinamakan sebagai tipe data dasar. Kelima tipe data dasar adalah :

1. Bilangan bulat (integer)
2. Bilangan real presisi-tunggal
3. Bilangan real-presisi ganda
4. Karakter
5. Tak bertipe

Tabel II.2 Ukuran Memori Untuk Tipe Data

Tipe	Total Bit	Kawasan	Keterangan
char	8	-128 s/d 127	Karakter
int	32	-2147483648 s/d 2147483647	Bilangan Integer
float	32	1.7E-38 s/d 3.4E+38	Bil. Real presisi-tunggal
double	64	2.2E-308 s/d 1.7E+308	Bil. Real presisi-ganda

2.2.2.2 Operator

Operator adalah suatu tanda atau simbol yang digunakan untuk suatu operasi tertentu. Berikut adalah operator pada bahasa C.

- Operator Aritmatika yang tergolong operator binary :

1. * Perkalian
2. : Pembagian
3. + Penjumlahan
4. - Pengurangan
5. % Sisa Pembagian

- Operator Aritmatika yang tergolong operator unary :

1. - Tanda Minus
2. + Tanda Plus

2.2.2.3 Menampilkan Data

Untuk menampilkan data ke layar. C menyediakan beberapa fungsi yaitu :

- Fungsi *putchar ( )*

  Fungsi *putchar( )*digunakan khusus untuk menampilkan sebuah karakter di layar. Penampilan karakter tidak diakhiri dengan perpindahan baris.

- Fungsi *printf ( )*

  Fungsi *printf()* merupakan fungsi yang paling umum digunakan dalam menampilkan data. Berbagai jenis data dapat ditampilkan ke layar dengan memakai fungsi ini.

2.2.2.4 Memasukkan Data

Data dapat dimasukkan lewat keyboard saat eksekusi berlangsung. Untuk keperluan ini, C menyediakan sejumlah fungsi, diantaranya adalah *scanf()* dan *getchar().*

- Fungsi *scanf ( )*

  Fungsi *scanf()* merupakan fungsi yang dapat digunakan untuk memasukkan berbagai jenis data.

- Fungsi *getchar ( )*

  Fungsi *getchar()* digunakan khusus untuk menerima masukan berupa sebuah karakter dari keyboard.

2,3 Visual Basic

Pada tugas akhir ini, bahasa pemrograman yang digunakan untuk aplikasi *user interface* pada server (komputer) adalah Visual Basic 6.0. Visual Basic 6.0 merupakan bahasa pemrograman yang dapat digunakan untuk menyusun dan membuat program aplikasi yang terdapat pada lingkungan sistem operasi Windows. Program aplikasi dapat berupa program database, program grafis, dan lain sebagainya. Pada Visual Basic 6.0 sudah terdapat komponen – komponen yang sangat membantu pembuatan program aplikasi.

2.3.1 Menu

Visual basic mempunyai beberapa fasilitas yang dapat digunakan untuk membantu user dalam membuat program aplikasi, salah satunya adalah menu. Visual Basic mempunyai tigabelas menu yang dapat digunakan dan masing – masing menu mempunyai fungsi yang berbeda-beda. Menu tersebut dapat dilihat pada Gambar 2.7.

Gambar 2.7 Menu Visual Basic

2.3.2 Toolbar

Pada prinsipnya toolbar berfungsi sama dengan menu, hanya saja berbeda tampilan. Untuk dapat menggunakan fasilitas menu, harus mengklik menu yang dipilih terlebih dahulu dan selanjutnya mengklik submenu yang akan digunakan. Pada toolbar, Anda cukup tinggal mengklik icon yang terdapat pada masing-masing toolbar. Untuk dapat menambah toolbar pada Visual Basic, klik menu **View | Toolbars.** Selanjutnya ada pilihan untuk menambah toolbar, diantaranya Debug, Edit, Form Editor, Standard, dan Customize. Pada submenu Customize terdapat pilihan untuk mengatur toolbar yang akan digunakan. Bentuk salah satu toolbar terlihat pada Gambar 2.8.

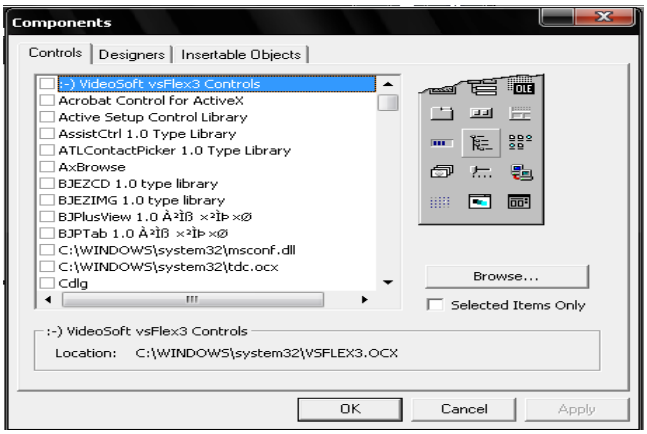
Gambar 2.8 Toolbar Visual Basic

2.3.3 Toolbox

Toolbox merupakan tempat kontrol-kontrol yang akan digunakan untuk membantu pembuatan program aplikasi. Secara default, pada toolbox hanya terdapat kontrol-kontrol seperti pada Gambar 2.9.

Gambar 2.9 Toolbox Visual Basic

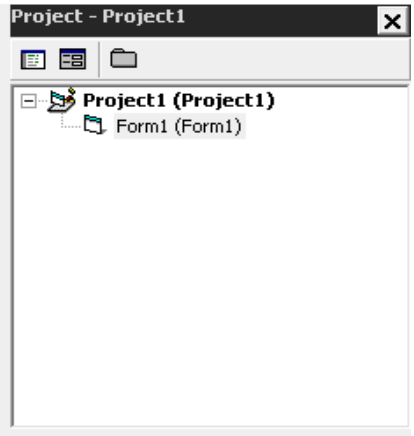
Tapi kontrol-kontrol pada toolbox dapat ditambah sesuai dengan selera. Untuk dapat menambah kontrol-kontrol pada toolbox, lakukan dengan mengklik kanan bagian toolbox, lalu mengklik Components. Selanjutnya akan terlihat tampilan seperti Gambar 2.10. Selanjutnya pilih komponen kontrol yang akan ditambahkan sampai tanda √ muncul pada kotak pilihan dan klik **OK**.



Gambar 2.10 Komponen Objek Kontrol

2.3.4 Project Explorer

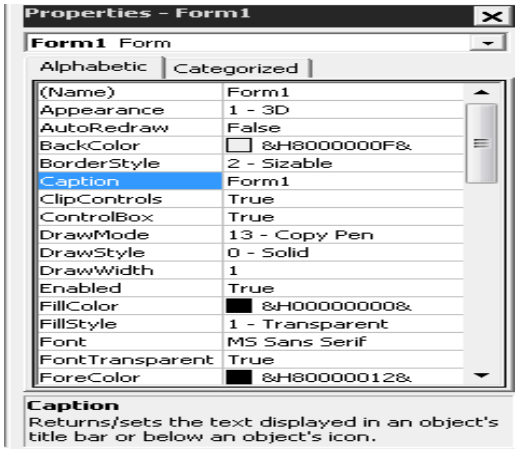
Project Explorer merupakan tempat yang digunakan untuk melihat daftar forms, modules, class modules, dan designers seperti pada Gambar 2.11 . Di samping itu, Project Explorer dapat digunakan untuk menambah forms, modules, class modules, dan designers dengan mengklik kanan bagian Project Explorer dan memilih **Add** serta format yang akan ditambahkan.



Gambar 2.11 Project Explorer

2.3.5 *Properties* Window

Properties Window merupakan tempat yang digunakan untuk mengatur properti dari setiap objek kontrol. Pada *Properties* Window ini semua objek kontrol dapat diatur sesuai dengan program aplikasi yang akan dibuat seperti yang terlihat pada Gambar 2.12.



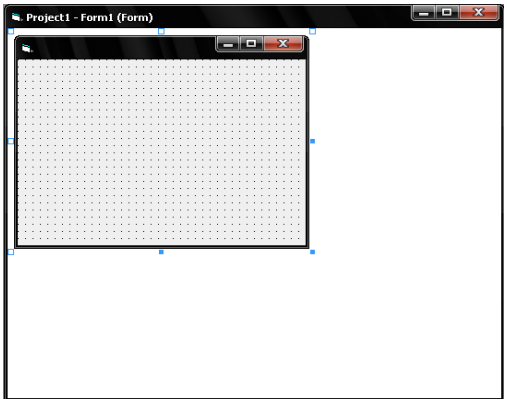
Gambar 2.12 *Properties* Window

2.3.6 Form Layout Window

Form Layout Window ini berfungsi untuk melihat atau mengetahui posisi tampilan form saat dieksekusi atau program dijalankan. Untuk dapat mengubah ukuran form, gunakan fasilitas *Properties* Windows, lalu ubah ukuran tampilan form sesuai denga selera. Untuk mengubah posisi tampilan saat program dijalankan , klik layar pada form layout window dan atur sesuai dengan keinginan.

2.3.7 Form Objek

Form Objek digunakan untuk menempatkan atau meletakkan objek dari kontrol-kontrol yang akan digunakan untuk merancang dan membuat program aplikasi. Untuk dapat menampilkan form objek ini, klik form pada Project Explorer, klik kanan form tersebut, lalu pilih **View Object**. Tampilan form objek dapat dilihat pada Gambar 2.13.



Gambar 2.13 Tampilan Form Objek

2.3.8 Form Kode

Form kode digunakan sebagai tempat untuk menulis kode-kode program aplikasi. Untuk dapat menampilkan form kode ini, klik form pada Project Explorer, lalu klik kanan form tersebut dan pilih **View Code** seperti pada Gambar 2.14.

Gambar 2.14 Tampilan Form Kode

.

2.4 Komunikasi Data

Berbagai macam protokol digunakan untuk berkomunikasi atau saling mengirim data antar device. Protokol adalah tata cara berkomunikasi, hal ini diperlukan untuk standarisasi pertukaran data agar antar kedua atau lebih device dapat saling berkomunikasi dengan baik.

Jika suatu device seperti PC misalnya ingin bertukar data dengan device lainnya, maka harus ditentukan dahulu media apa yang akan digunakan dan protokol apa yang dipakai. Pada proyek ini proses pertukaran data antara PC/Notebook dengan Ethernet to Serial Gateway menggunakan fasilitas winsock (Windows Socket) dengan protokol yang digunakan adalah protokol TCP/IP. Sedangkan pertukaran data antara ethernet to serial converter dengan mikrokontroler adalah komunikasi serial.

2.4.1 TCP/IP

TCP/IP (*Transmission Control Protocol*/ *Internet Protocol*) adalah standar komunikasi data yang dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam jaringan Internet. Dalam arti yang sederhana , TCP/IP (Transmission Control Protocol / Internet Protocol) adalah nama keluarga protokol jaringan. Protokol dalam arti sebenarnya adalah sekelompok aturan yang harus diikuti oleh perusahaan -perusahaan dan produk-produk agar produk tersebut bisa kompatibel satu dengan yang lainnya. Suatu protokol menentukan

cara suatu software berkomunikasi dengan software lain, juga menentukan cara setiap bagian dari keseluruhan paket mengatur perjalanan informasinya.

Protokol TCP/IP dikembangkan pada akhir dekade 1970-an hingga awal 1980-an sebagai sebuah protokol standar untuk menghubungkan komputer-komputer dan jaringan untuk membentuk sebuah jaringan yang luas (WAN). TCP/IP merupakan sebuah standar jaringan terbuka yang bersifat independen terhadap mekanisme transport jaringan fisik yang digunakan, sehingga dapat digunakan di mana saja. Protokol ini menggunakan skema pengalamatan yang sederhana yang disebut sebagai alamat IP (*IP Address*) yang mengizinkan hingga beberapa ratus juta komputer untuk dapat saling berhubungan satu sama lainnya di Internet.

TCP/IP sebenarnya adalah dua macam protokol yang berbeda. Tidak seperti dengan anggapan kebanyakan orang, istilah TCP/IP mengacu kepada seluruh keluarga protokol yang dirancang untuk mentransfer informasi sepanjang jaringan. TCP/IP dirancang untuk menjadi komponen perangkat lunak dari suatu jaringan. Semua bagian di dalam keluarga TCP/IP memiliki tugas sendiri, misalnya mengirim e-mail, mentransfer file , menyediakan layanan login jarak jauh (remote login) dan menangani informasi routing jaringan.

Protokol TCP (*Transmission Control Protocol*) bertanggung jawab untuk memecah / membagi informasi ke dalam beberapa paket, sedangkan IP (*Internet Protocol*) bertanggung jawab untuk mengirimkan (mentransfer) paket-paket tersebut sesuai tujuannya. Kemudian TCP (*Transmission Control Protocol*) bertugas menyatukan kembali paket-paket tersebut sesuai dengan urutan yang benar.

Layanan dalam TCP/IP yang berbeda dikelompokkan menurut fungsi-fungsinya. Protokol-prorokol transport mengendalikan pergerakan data antara dua mesin, dan mencakup:

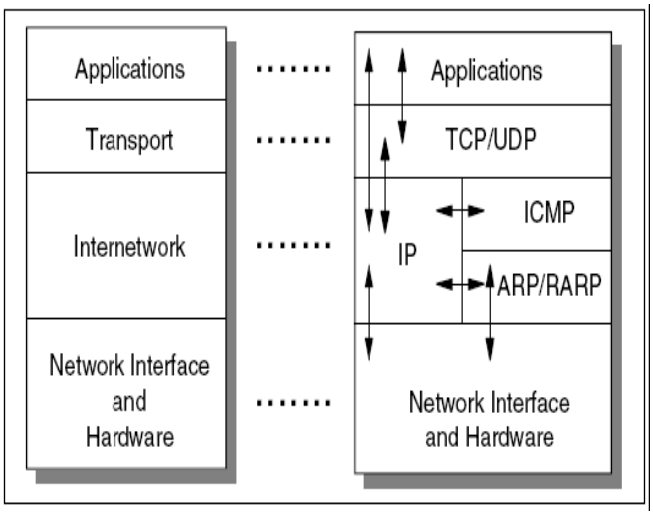
1. TCP (Transmission Control Protokol)

   Protokol ini bersifat *connection-based*, artinya kedua mesin pengrim dan penerima tersambung dan berkomunikasi satu sama lain sepanjang waktu.

2. UDP (User Datagram Protokol)

   Protokol ini bersifat *connectionless* (tanpa koneksi), artinya data dikirim tanpa kedua mesin penerima dan pengirim saling berhubungan. Hal ini sama seperti mengirim surat lewat kantor pos, surat dikirim oleh pengirim namun ia tidak pernah bisa mengetahui apakah surat tersebut sampai di tujuan atau tidak.

Seperti pada perangkat lunak, TCP/IP dibentuk dalam beberapa lapisan (*layer*). Dengan dibentuk dalam layer, akan mempermudah untuk pengembangan dan pengimplementasian. Sehingga antar layer dapat berkomunikasi ke atas maupun ke bawah dengan suatu penghubung *interface*. Tiap-tiap layer memiliki fungsi dan kegunaan yang berbeda dan saling mendukung layer diatasnya. Pada protokol TCP/IP dibagi menjadi 4 layer, tampak seperti Gambar 2.15.



Gambar 2.15 Layer TCP/IP

Layer Aplikasi (Aplications)

Layer aplikasi digunakan pada program untuk berkomunikasi menggunakan TCP/IP. Contoh aplikasi antara lain Telnet dan File Transfer Protocol (FTP). *Interface* yang digunakan untuk saling berkomunikasi adalah nomer port dan socket.

Layer Transport

Layer transport memberikan fungsi pengiriman data secara *end-to-end* ke sisi remote. Aplikasi yang beragam dapat melakukan komunikasi secara serentak

(*simultaneously*). Protokol pada layer transport yang paling sering digunakan adalah Transmission Control Protocol (TCP), TCP memberikan fungsi pengiriman data secara *connection-oriented*, pencegahan duplikasi data, *congestion control* dan *flow control*. Protokol lainnya adalah User Datagram Protocol (UDP), UDP memberikan fungsi pengiriman *connectionless* dengan jalur yang tidak reliabel. UDP banyak digunakan pada aplikasi yang membutuhkan kecepatan tinggi dan dapat mentoleransi terhadap kerusakan data.

Layer Internetwork

Layer Internetwork biasa disebut juga layer internet atau layer network, dimana memberikan "vitual network" pada internet. Internet Protocol (IP) adalah protokol yang paling penting. IP memberikan fungsi routing pada jaringan dalam pengiriman data. Protokol lainnya antara lain : IP, ICMP, IGMP, ARP, RARP.

Layer Network *Interface*

Layer network *interface* disebut juga layer link atau layer datalink, yang merupakan perangkat keras pada jaringan. Contoh : IEEE802.2, X.25, ATM, FDDI, dan SNA.

2.4.2 *Wireless* LAN (WLAN)

Jaringan lokal nirkabel atau WLAN adalah suatu jaringan area lokal nirkabel yang menggunakan gelombang radio sebagai media tranmisinya. Link terakhir yang digunakan adalah nirkabel, untuk memberi sebuah koneksi jaringan ke seluruh pengguna dalam area sekitar. Area dapat berjarak dari ruangan tunggal ke seluruh kampus. Tulang punggung jaringan biasanya menggunakan kabel, dengan satu atau lebih titik akses jaringan menyambungkan pengguna nirkabel ke jaringan berkabel.

LAN nirkabel adalah suatu jaringan nirkabel yang menggunakan frekuensi radio untuk komunikasi antara perangkat komputer dan akhirnya titik akses yang merupakan dasar dari transiver radio dua arah yang tipikalnya bekerja di bandwith 2,4 GHz (802.11b, 802.11g) atau 5 GHz (802.11a). Kebanyakan peralatan mempunyai kualifikasi Wi-Fi, IEEE 802.11b atau akomodasi IEEE 802.11g dan menawarkan beberapa level keamanan seperti WEP dan WPA.

2.4.2.1 *Wireless Fidelity* (*WiFi*)

WiFi merupakan kependekan dari *Wireless Fidelity*, memiliki pengertian yaitu sekumpulan standar yang digunakan untuk Jaringan Lokal Nirkabel (*Wireless Local Area Networks* - WLAN) yang didasari pada spesifikasi IEEE 802.11. Standar terbaru dari spesifikasi 802.11a atau b, seperti 802.16 g, saat ini sedang dalam penyusunan, spesifikasi terbaru tersebut menawarkan banyak peningkatan mulai dari luas cakupan yang lebih jauh hingga kecepatan transfernya.

Awalnya *WiFi* ditujukan untuk pengunaan perangkat nirkabel dan Jaringan Area Lokal (LAN), namun saat ini lebih banyak digunakan untuk mengakses internet. Hal ini memungkinkan seseorang dengan komputer dengan kartu nirkabel (*wireless card*) atau *personal digital assistant* (PDA) untuk terhubung dengan internet dengan menggunakan titik akses (atau dikenal dengan *hotspot*) terdekat.

WiFi dirancang berdasarkan spesifikasi standar IEEE 802.11. Sekarang ini ada beberapa variasi dari standar IEEE 802.11. Spesifikasi standar IEEE 802.11 dapat dilihat pada Tabel II.3.

Tabel II.3 Spesifikasi Standar IEEE 802.11

Standar / Protokol	Frekuensi	Data Rate
IEEE 802.11	2.4 GHz	2 Mbps
IEEE 802.11a	5 GHz	54 Mbps
IEEE 802.11a 2X	5 GHz	108 Mbps
IEEE 802.11b	2.4 GHz	11 Mbps
IEEE 802.11b+	2.4 GHz	22 Mbps
IEEE 802.11g	2.4 GHz	54 Mbps
IEEE 802.11n	2.4 GHz	120 Mbps

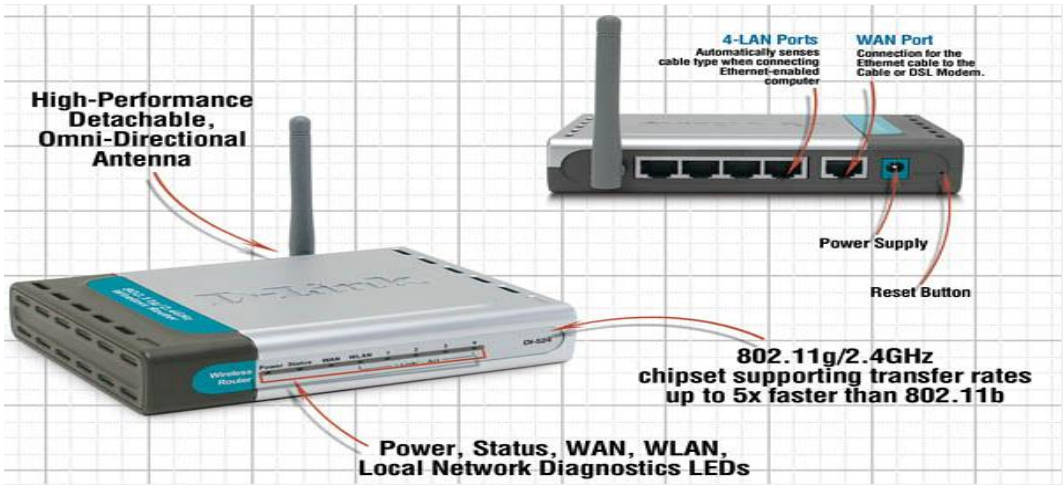
Infrastruktur *WiFi* IEEE 802.11 pada dasarnya mempunyai jumlah channel yang sangat terbatas sekali. Pada jaringan yang sangat padat, tidak semua channel dapat digunakan sekaligus untuk mengurangi interferensi di infratruktur WiFi. Pembagian channel pada WiFi dapat dilihat pada Tabel II.4.

Tabel II.4 Pembagian Channel Frekuensi *WiFi*

Channel	Frekuensi	Channel	Frekuensi
1	2412 MHz	8	2447 MHz
2	2417 MHz	9	2452 MHz
3	2422 MHz	10	2457 MHz
4	2427 MHz	11	2462 MHz
5	2432 MHz	12	2467 MHz
6	2437 MHz	13	2472 MHz
7	2442 MHz	14	2477 MHz

2.4.2.2 Perangkat *Wireless*

Perangkat *wireless* yang digunakan pada Tugas Akhir ini adalah Access Point D-Link pada sisi client yang merupakan jembatan antara komunikasi wired dan *wireless*. Sedangkan pada sisi server menggunakan *WiFi* adapter ataupun notebook yang sudah terintegrasi dengan *WiF*i card di dalamnya.



Gambar 2.16 Access Point D-Link

2.4.3 Komunikasi Serial

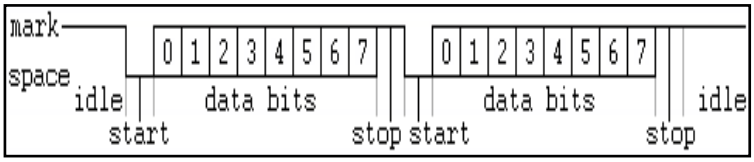
Komunikasi serial adalah suatu proses pengiriman data secara *sequential* atau satu persatu melalui sebuah kanal informasi. Pada dasarnya ada dua jenis komunikasi data serial, yaitu komuniksi data serial sinkron dimana pengiriman clock dilakukan secara bersamaan dengan data serial dan komunikasi serial asinkron dimana pengiriman clock dilakukan dalam dua tahap, yaitu pada saat data dikirimkan dan saat data diterima. Komunikasi serial mempunyai kecepatan

transfer data yang rendah tetapi cocok untuk komunikasi jarak jauh. Pada PC dan Mikrokontroler ATMega 16 sudah terdapat port serial yang membuat jenis komunikasi ini umum digunakan.

Serial port pada PC dan Mikrokontroler memungkinkan untuk melakukan komunikasi secara full duplex yang berarti dapat berkomunikasi secara dua arah dimana mengirim dan menerima data dapat dilakukan secara bersamaan. Tetapi ada beberapa device yang hanya support untuk half duplex (satu arah) saja.

Komunikasi serial mempunyai parameter yang harus ditentukan yaitu :

- Baud Rate atau kecepatan dari transmisi data
- Data bit
- Parity bit yang terdiri dari even dan odd, parity bit biasanya digunakan untuk error detection
- Stopbit



Gambar 2.17 Format Data Pada Komunikasi Serial

2.4.3.1 Standar RS-232

RS-232 (Recomended Standard 232) adalah salah satu standar dalam melakukan komunikasi serial yang dikeluarkan oleh Electronic Industries Alliance (EIA) yang mencakup :

- Karakteristik sinyal seperti level tegangan, signal rate, timing dan slew-rate , dll
- *Interfacing* konektor seperti identifikasi pin- pin dari konektor
- Fungsi dari setiap circuit pada konektor
- Standar subset dari *interface* circuit untuk aplikasi telekomunikasi

Device yang menggunakan serial dibedakan atas dua kategori yaitu DCE (Data Communication Equipment) dan DTE (Data Terminating Equipment). DCE adalah device seperti modem. Sedangkan DTE adalah komputer atau terminal data lainnya. Kabel RS 232 dibedakan atas 2 tipe yaitu D- Type 25 pin *connector* dan D-Type 9 pin *connector*.

2.5 Windows Socket Control (Winsock)

Sebelum membahas tentang Winsock Control, perlu mengetahui lebih dahulu tentang socket. Socket merupakan jembatan yang menghubungkan suatu aplikasi berbasis Internet dengan lapisan TCP/IP pada sistem operasi.

Sebuah socket umumnya digunakan pada aplikasi yang menyangkut perpindahan data melalui jaringan komputer. Socket menyediakan jalur untuk mentransfer data ke tujuan. Tedapat juga dua pasang socket, yaitu digunakan untuk proses pengiriman data dan yang digunakan untuk proses penerimaan data.

Pada aplikasi client/server, socket digunakan dalam implementasi program sisi client maupun sisi server. Saat client mengirimkan request, socket pengiriman ada pada sisi client, sementara socket penerimaan ada pada sisi server. Pada saat server mengirimkan response, socket pengiriman ada pada sisi server, sementara socket penerimaan ada pada sisi client.

Sebuah socket dilengkapi dengan alamat, yang terdiri atas alamat IP tujuan dan nomor port. Nomor port merupakan bilangan bulat yang digunakan untuk membedakan layanan-layanan yang berjalan pada komputer server yang sama. Pengguna layanan menggunakan nomor port ini menghubungi komputer server. Beberapa layanan yang umum digunakan pada Internet telah memiliki nomor port standard.Ada dua mode operasi yang bisa dilakukan Winsock, yaitu :

- sckTCPProtocol

  Pada mode ini, Winsock menggunakan protocol TCP sehingga koneksi yang dibangun bersifat *connection oriented.*
- sckUDPProtocol

  Pada mode ini, Winsock menggunakan protocol UDP sehingga koneksi yang dibangun bersifat *connectionless.*

2.5.1 *Properties* Winsock Control

Winsock memiliki beberapa *properties* yang digunakan dalam pemrograman Winsock Control. Berikut *properties* yang dimiliki oleh Winsock Control :

- Property BytesReceived

  Property ini memberitahu jumlah byte yang terdapat pada buffer *receiver*.
- Property LocalHostName

  Property ini merupakan nama dari local host pada sistem.
- Property LocalIP

  Property ini merupakan alamat IP dari local host pada sistem.
- Property LocalPort

  Property ini merupakan nomor port pada local host.
- Property Protocol

  Property ini merupakan jenis protocol komunikasi yang terdiri dari TCP dan UDP.
- Property RemoteHost

  Property ini merupakan alamat IP dari *remote host* pada sistem.
- Property RemotePort

  Property RemotePort merupakan nomor port pada *remote host*.

2.5.2 Metode dari Winsock Control

Metode adalah fungsi pradefinisi yang digunakan untuk melakukan berbagai tugas dalam winsock control. Beberapa metoda penting yang digunakan dalam winsock control yaitu :

- Accept Method

  Accept Method digunakan hanya untuk aplikasi TCP server. Server menerima permintaan koneksi dari client. Untuk accept method ini, server harus dalam keadaan listening.
- Close Method

  Close method mengakhiri koneksi TCP dari kedua aplikasi yaitu server dan client.
- GetData Method

  GetData Method adalah metode mencari kembali (*retrieves*) aliran blok data dari buffer dan menyimpannya dalam sebuah variable dengan berbagai tipe.

- Listen Method

  Listen Method hanya terdapat pada aplikasi server. Dalam hal ini server menunggu untuk permintaan koneksi TCP dari client.
- SendData Method

  SendData method adalah mengirimkan data ke *remote computer*.
- Connect Method

  Connect method adalah meminta koneksi pada computer tujuan *remote computer*.

2.5.3 Event pada Winsock Control

Beberapa event penting dalam Winsock Control yaitu :

- Close Event

  Close Event terjadi ketika *remote computer* menutup koneksi.
- Connect Event

  Connect Event terjadi setelah terkoneksi dengan *remote computer*.
- ConnectionRequest Event

  ConnectionRequest Event terjadi ketika server menerima koneksi dari client.
- DataArrival Event

  DataArrival Event terjadi ketika data baru tiba.
- SendComplete Event

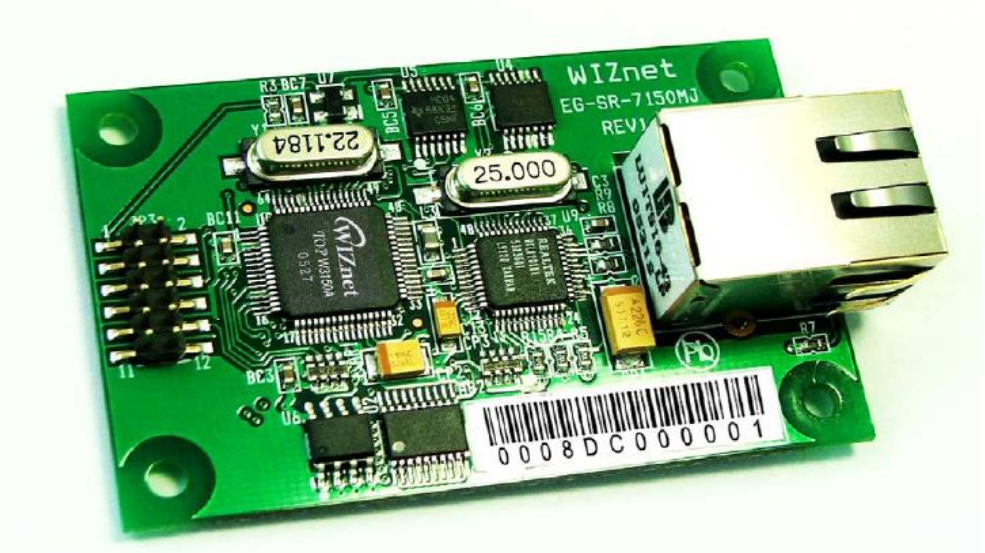
  SendComplete Event terjadi ketika operasi pengiriman data telah selesai dilakukan.
- SendProgress Event

  SendProgress Event terjadi ketika data sedang dalam proses pengiriman.
- Error Event

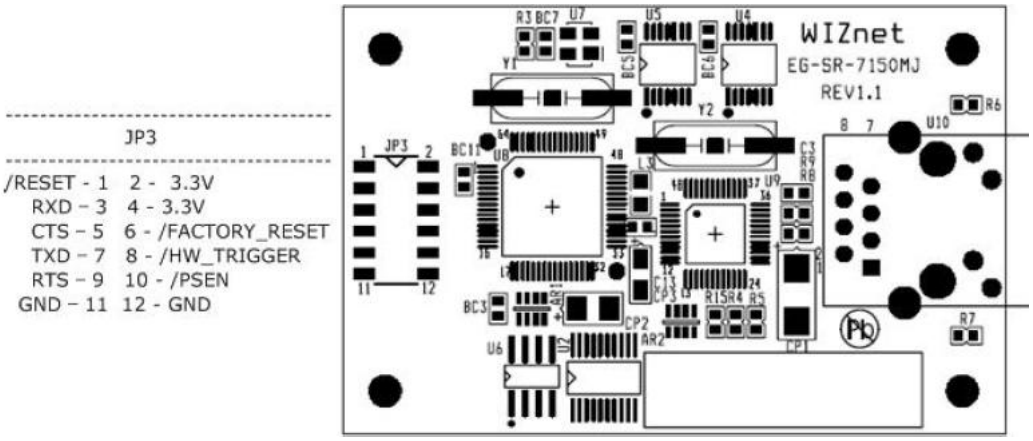
  Error Event terjadi proses seperti koneksi, pengiriman data dan penerimaan data gagal dilakukan atau terjadi error.

2.6 EGSR-7150MJ (Ethernet to Serial Gateway)

EGSR-7150MJ adalah suatu modul gateway yang mengubah tipe data serial ke tipe data TCP/IP dan sebaliknya atau lebih dikenal sebagai ethernet to serial gateway. Bentuk EGSR-7150MJ dapat dilihat pada Gambar 2.20. Sedangkan konfigurasi pin dari EGSR-7150MJ dapat dilihat pada Gambar 2.21. Dengan EGSR-7150MJ yang tersambung dengan konektor RJ-45, user bisa lebih mudah dan lebih cepat untuk membuat antarmuka melalui Ethernet. EGSR-7150MJ memiliki program Configuration Tool yang digunakan untuk melakukan konfigurasi TCP/ IP dan serial.



Gambar 2.18 Modul EGSR-7150MJ



Gambar 2.19 Dimensi dan Konfigurasi Pin Pada EGSR-7150MJ

Untuk menggunakan Ethernet to Serial gateway EGSR-7150MJ maka harus dilakukan konfigurasi pada EGSR-7150MJ. Berikut konfigurasi yang dilakukan pada EGSR-7150MJ.

1. Search

   Fungsi search digunakan untuk mencari semua modul EGSR-7150MJ yang berada pada suatu jaringan..

2. Setting

   Fungsi setting digunakan apabila kita telah selesai melakukan konfigurasi.

3. IP Configuration Method

   Fungsi IP configuration method memiliki 2 pilihan yaitu :

   - Static : IP address dapat diatur secara manual oleh user.
   - DHCP : IP , subnet ,dan gateway address ditentukan / diberikan oleh DHCP server.

4. Operation Mode

   Operation Mode terdiri dari 3 pilihan yaitu :

   - TCP server : Pada mode operasi ini, IP address , subnet , dan gateway dikonfigurasi sebagai server. EGSR-7150MJ menunggu untuk dihubungi oleh komputer host. Membolehkan komputer host untuk mengadakan koneksi dan menerima data dari peralatan serial.
   - TCP client : Pada mode operasi ini, IP address, subnet, dan gateway diset sebagai client. EGSR-7150MJ dalam keadaan active open untuk mengadakan koneksi TCP ke komputer host.
   - UDP : Pada mode ini, prosedur TCP/IP tidak diperlukan.

   Pilihan diatas tergantung pada jenis protokol yang digunakan pada operasi sistem.

5. Serial Command Method

   Serial command method terdiri dari 2 pilihan yaitu :

   - H/W Trigger
   - S/W Trigger

6. Delimiter

   Dengan menu delimiter ini dapat ditentukan bagaimana data dipaketkan dan dikirim melalui Ethernet.

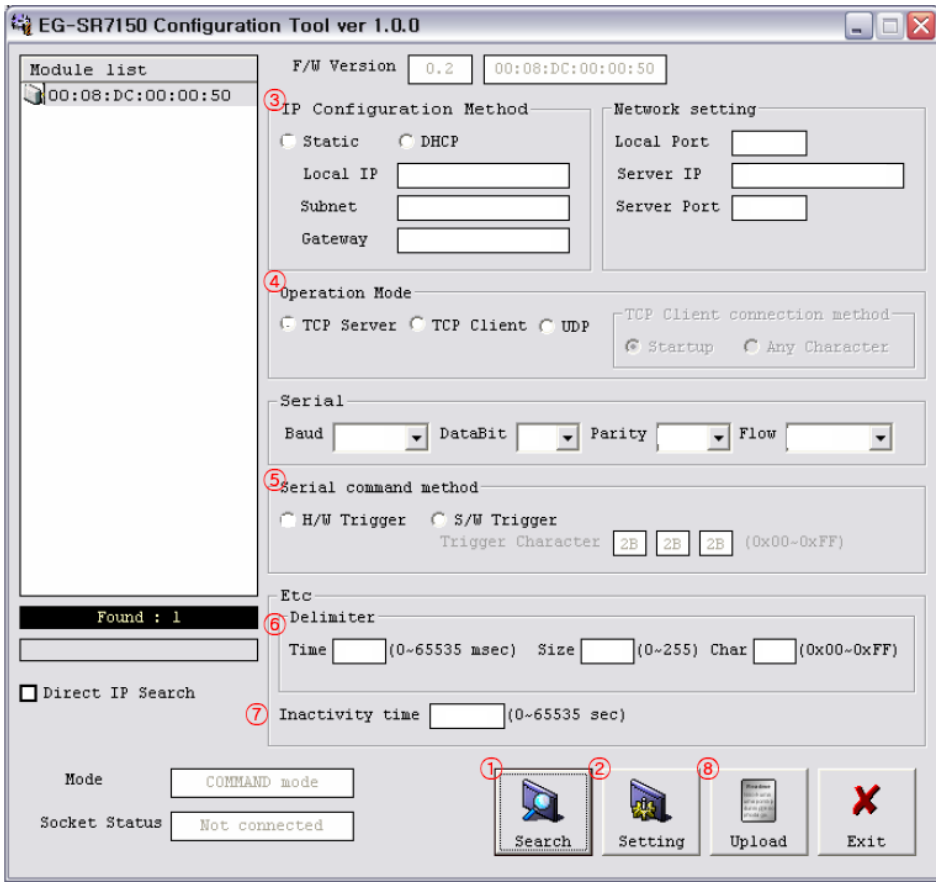
7. Inactivity Time

   Setelah koneksi terjadi, jika tidak ada lagi data yang ditransmisikan dalam suatu waktu maka didefisikan sebagai “inactivity time” dan koneksi secara otomatis akan ditutup.

8. Upload

   Menu ini digunakan untuk mengupload firmware melalui jaringan.

Gambar tampilan konfigurasi dari EGSR-7150MJ (configuration tool EGSR-7150MJ) yang digunakan untuk melakukan konfigurasi dapat dilihat pada Gambar 2.20.



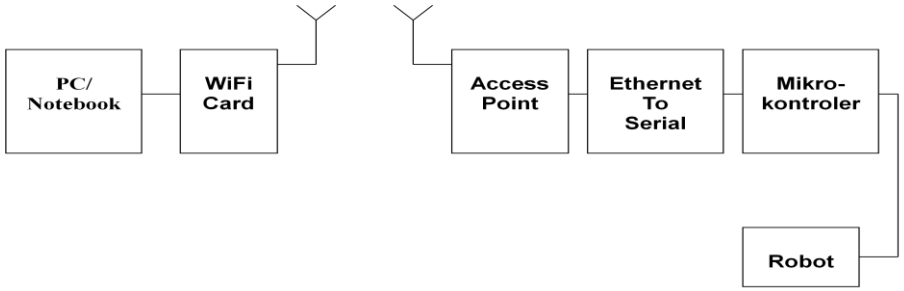
Gambar 2.20 *Configuration Tool* EGSR-7150MJ

Adapun beberapa fitur utama yang terdapat pada EGSR-7150MJ adalah:

- *Ready-to-go Serial to Ethernet gateway* module yang terhubung dengan konektor RJ-45.
- Mendukung dalam komunikasi serial.
- Stabilitas dan keandalan yang tinggi dengan menggunakan Chip W3150A WIZnet, dan dilengkapi dengan *fully-hardwired* TCP/IP Stack.
- Mudah dalam konfigurasi.
- 10/100Mbps Ethernet *interface*, Max. 230Kbps Serial *interface*
- RoHS *compliant*

BAB III
PERANCANGAN DAN REALISASI PERANGKAT KERAS DAN LUNAK

Pada bab ini akan dibahas perancangan dan realisasi perangkat keras dan lunak yang meliputi perakitan rangkaian mikrokontroler untuk menerima/mengirimkan data serta mengendalikan robot, pemrograman pada mikrokontroler serta pembuatan program aplikasi pengendali robot.



Gambar 3.1 Blok Diagram Sistem Pengendalian Robot

Pada blok diagram tersebut dapat diamati bahwa PC/Notebook mengendalikan Mikrokontroler. PC/Notebook dihubungkan dengan WiFi Card sebagai transmiter yang kemudian mentransmisikan sinyal kendali pada frekuensi 2.4 GHz ke Access Point sebagai receiver. Sinyal yang diterima oleh Access Point langsung dikirim ke Ethernet to Serial Gateway untuk selanjutnya dihubungkan dengan Mikrokontroler. Kemudian Mikrokontroler akan mengendalikan mekanik lengan robot sesuai dengan sinyal kendali yang dikirimkan dari PC/ Notebook. Selain itu Mikrokontroler akan mengirimkan data dari sensor yang berupa limit-switch ke PC/Notebook bila mekanik lengan robot tersebut menekan sensor limit-switch yang terdapat pada ujung bagian mekanik lengan robot.

3.1 Perancangan dan Perakitan Perangkat Keras.

Dalam tugas akhir ini, digunakan mikrokontroler AVR ATMega 16 yang akan mengendalikan mekanik lengan robot dan untuk komunikasi serial

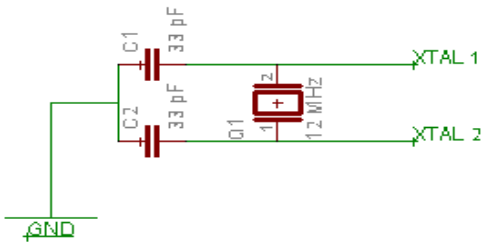
memanfaatkan pin Tx (Transmiter) dan pin Rx (*Receiver*) yang terdapat pada mikrokontroler. Dalam membuat rangkaian mikrokontroler diperlukan pemahaman mengenai sistem minimum dari mikrokontroler yang akan direalisasi. Sistem rangkaian yang akan dirancang diusahakan menggunakan rangkaian yang seringkas mungkin (sesuai dengan keperluan perancangan) dan dengan sistem penataan kabel yang baik.

ATMega 16 mempunyai rangkaian eksternal yang relatif sedikit dibanding dengan mikrokontoler yang lain. Rangkaian eksternal yang dibutuhkan adalah sebagai berikut :

- Rangkaian *Clock Generator*.
- Rangkaian *Regulator*.
- Rangkaian *Interfacing* ke rangkaian luar (*input*/*output*).
- Rangkaian driver motor pengendali robot.
- Rangkaian *interfacing* mikrokontroler dengan EGSR-7150MJ (Ethernet to serial converter)

3.1.1 Rangkaian *Clock Generator*

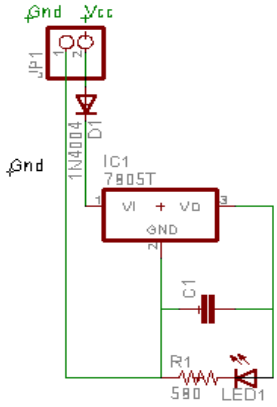
Mikrokontroler ATMega 16 memiliki osilator internal (*on chip oscillator*) yang dapat digunakan sebagai sumber *clock* bagi CPU. Untuk mengunakan osilator internal diperlukan sebuah kristal antara pin XTAL1 dan XTAL2 serta kapasitor ke ground. Untuk kristal dapat digunakan frekuensi dari 6 sampai 12 MHz. Namun pada rangkaian tugas akhir ini digunakan kristal dengan frekuensi 12 MHz. Sedangkan untuk kapasitor digunakan kapasitor dengan nilai 33 pF. Rangkaian *clock generator* dapat dilihat pada Gambar 3.1.



Gambar 3.1 Rangkaian *Clock Generator*

3.1.2 Rangkaian *Regulator*

Rangkaian *regulator* yang digunakan untuk memberi tegangan yang stabil pada mikrokontroler, dan mempunyai arus yang cukup sehingga tidak terjadi drop tegangan pada mikrokontroler dan rangkaian yang lain. Supply tunggal yang dibutuhkan mikrokontroler sebesar +5Volt. Pada rangkaian ini menggunakan regulator 7805 yang berfungsi menurunkan tegangan dari 12 Volt atau 9 Volt menjadi 5 Volt. Selain itu rangkaian regulator menggunakan kapasitor 10 uF. Rangkaian *regulator* dapat dilihat pada Gambar 3.2.



Gambar 3.2 Rangkaian *Regulator*

3.1.3 Rangkaian *Interfacing* ke *Input/Output*

Rangkaian I/O dari mikrokontroler mempunyai kontrol *directional* yang tiap bit dapat dikonfigurasikan secara individual, maka dalam pengkonfigurasian I/O yang digunakan ada yang berupa operasi port ada pula yang dikonfigurasi tiap bit I/O. Berikut ini akan diberikan konfigurasi dari I/O mikrokontroler tiap bit yang ada pada masing - masing port yang terdapat pada mikrokontroler.

- Port A

  Port A yang terdapat pada mikrokontroler difungsikan sebagi *input* dari sensor limit-switch.

- Port B

  Port B yang terdapat pada mikrokontroler difungsikan sebagai *output* untuk megendalikan motor DC pada robot. Dalam hal ini Port B akan mengeluarkan bit – bit logic yang akan dihubungkan dengan driver motor L293D.

- Port C

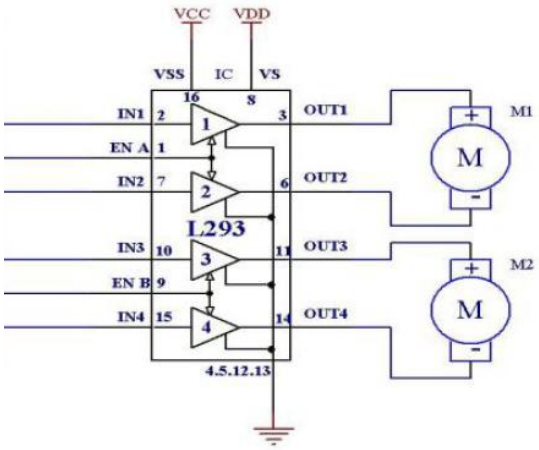
  Port C pada mikrokontroler tidak digunakan.

- Port D

  Port D pada mikrokontroler difungsikan sebagai pengirim (Tx) dan penerima (Rx) data serial. Selain itu digunakan juga sebagai pengendali motor servo dengan menggunakan fasilitas PWM pada mikrokontroler.

3.1.4 Rangkaian *Driver* Motor Pengendali Robot.

Dalam menggerakkan sebuah motor DC, diperlukan *driver* untuk mengendalikan motor DC sehingga motor DC dapat berputar dengan 2 arah putaran yang berlawanan. Dalam hal ini digunakan IC L293D sebagai *driver* motor. Dengan *driver* motor ini, mengendalikan motor dapat lebih mudah karena hanya memberikan *input* berupa bit – bit logika pada port *output* mikrokontroler. Gambar *driver* motor L293D dapat dilihat pada Gambar 3.3.

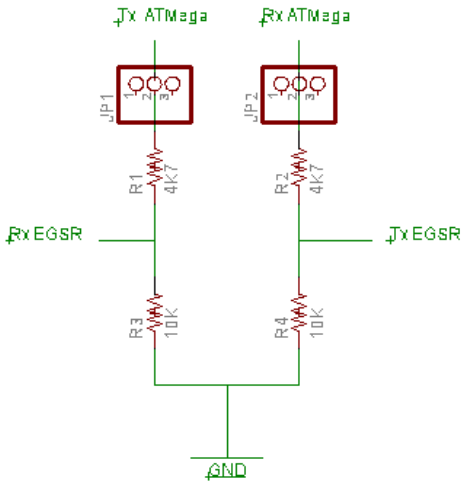


Gambar 3.3 *Driver* motor L293D

3.1.5 Rangkaian *Interfacing* Dengan EGSR-7150MJ (Ethernet to Serial Gateway)

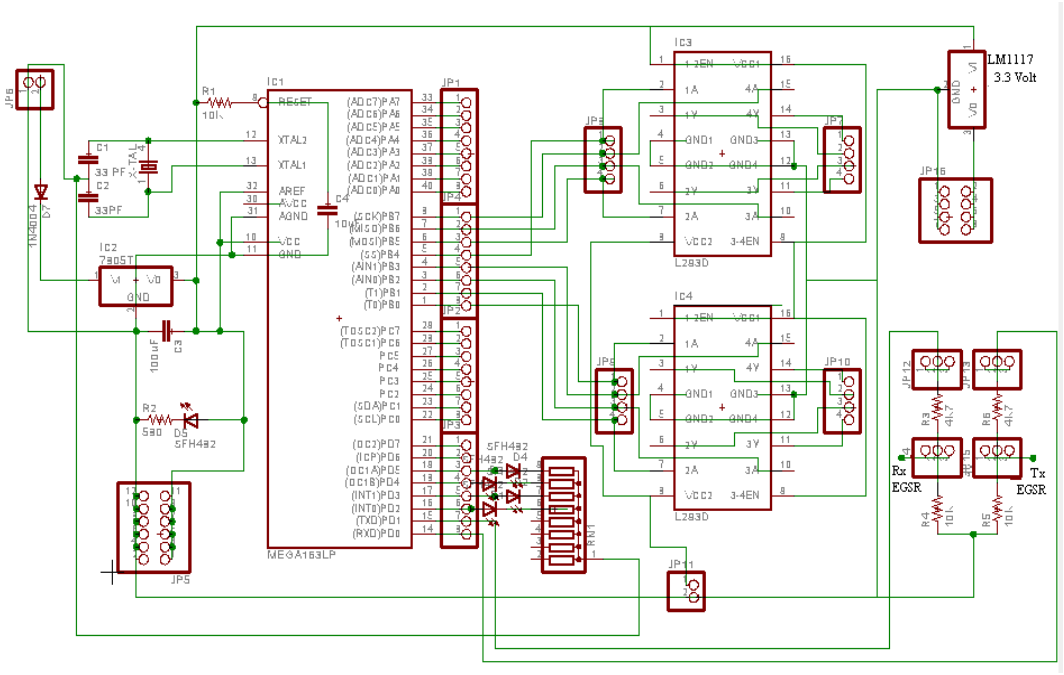
Agar komunikasi antar data serial dan data TCP/IP dapat terjadi maka port serial pada mikrokontroler harus terhubung dengan EGSR-7150 MJ. Pada rangkaian ini menggunakan resistor 4700 ohm dan 10000 ohm untuk menghindari kelebihan tegangan pada EGSR-7150 MJ. Hal ini disebabkan karena *level* tegangan pada EGSR-7150 MJ adalah 3.3 volt sedangkan *level* tegangan pada port

mikrokontroler adalah 5 volt. Rangkaian *interfacing* mikrokontroler dan EGSR 7150 MJ dapat dilihat pada Gambar 3.4.



Gambar 3.4 Rangkaian *Interfacing* Mikrokontroler dan EGSR-7150MJ

Secara keseluruhan rangkaian lengkap dari perangkat keras sistem yang dibuat adalah seperti Gambar 3.5.



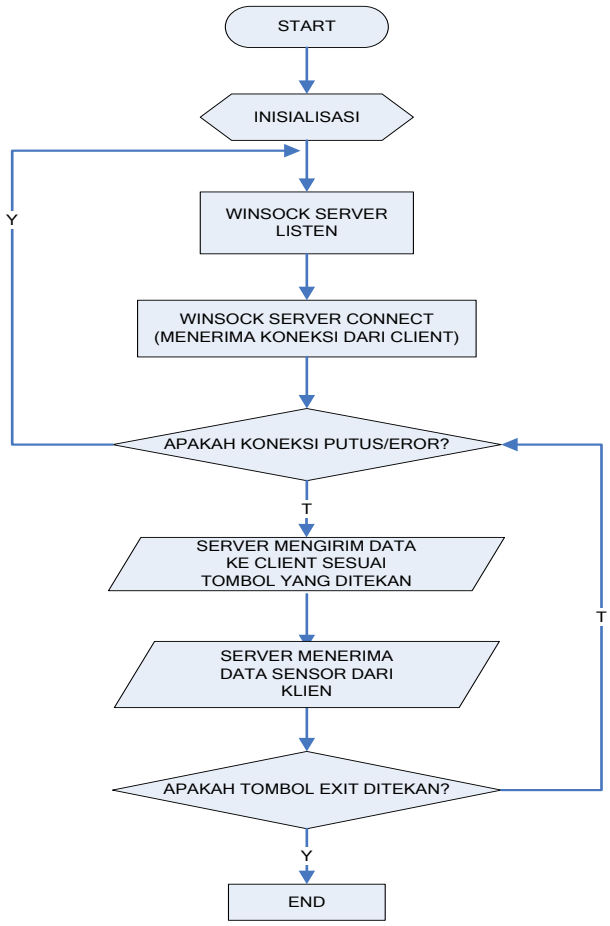
Gambar 3.5 Rangkaian Skematik *Interface* Pengendali Robot.

3.2 Perancangan dan Realisasi Perangkat Lunak.

Perancangan dan pembuatan perangkat lunak dilakukan pada server (komputer) dan client (robot).

3.2.1 Perancangan dan Realisasi Perangkat Lunak pada Server

Perancangan dan realisasi perangkat lunak pada server yaitu pada komputer / notebook sebagai pengendali robot.



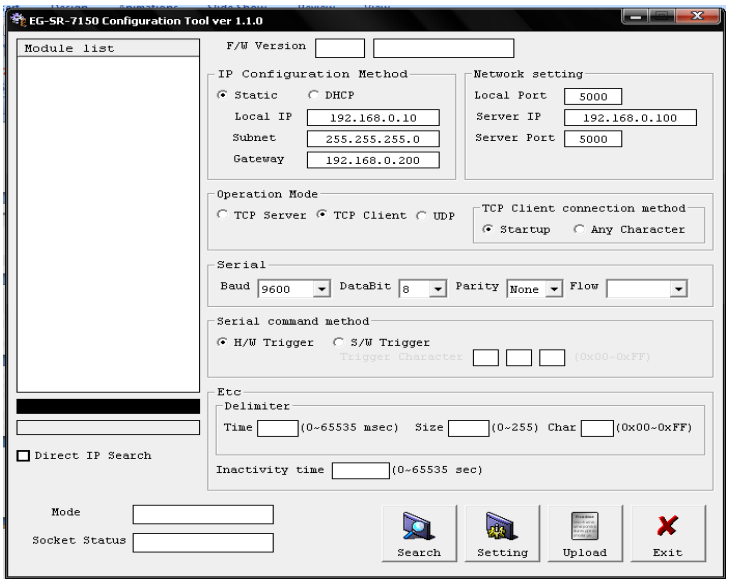
Gambar 3.6 Diagram Alir pada Server (Komputer)

Pada diagram alir dari Gambar 3.6 dapat dilihat bahwa setelah melakukan inisialisasi yaitu konfigurasi pada EGSR-7150MJ maka program pengendali robot dapat dijalankan. Pada program server (komputer) ketika tombol start ditekan maka server (komputer) berada dalam keadaan *listen* , artinya server (komputer) siap untuk menerima koneksi dari client (robot) yang menghubunginya. Apabila koneksi yang dilakukan putus atau error maka harus dilakukan inisialisasi kembali atau konfigurasi ulang. Jika koneksi tidak putus atau error maka server (komputer) dapat mengirim sinyal kendali /data karakter ke client (robot).

Setelah itu, server (komputer) siap menerima sinyal bila mekanik lengan robot menekan sensor limit-switch. Lalu jika tombol exit pada server (komputer) maka proses selesai.

3.2.1.1 Konfigurasi pada EGSR-7150MJ

Pada server (komputer), terdapat *configuration tool* EGSR-7150MJ yang merupakan *configuration tool* bawaan dari modul EGSR-7150MJ yang digunakan untuk mengkonfigurasi alamat IP, port dan baud rate yang digunakan untuk komunikasi antara server dan client. Gambar tampilan konfigurasi pada EGSR-7150MJ dapat dilihat seperti pada Gambar 3.7.



Gambar 3.7 Konfigurasi EGSR-7150MJ (Client) pada Server

Konfigurasi yang dilakukan pada EGSR-7150MJ adalah sebagai berikut:

- IP configuration Method : Static

  Dipilih Static agar alamat IP yang telah diset tidak berubah – ubah. Artinya alamat IP akan tetap pada nomor yang telah kita set secara manual.

- Local IP

  Local IP adalah alamat IP yang digunakan oleh EGSR-7150MJ (Client) yaitu 192.168.0.10. Pada konfigurasi ini menggunakan alamat IP kelas C

dengan panjang 32 bit. Dalam format ini setiap 4 byte ditulis dalam bilangan decimal, mulai dari 0 sampai 255.

- Subnet

  Subnet dibutuhkan bila terdapat pembagian jaringan. Pada konfigurasi ini tidak dibutuhkan pembagian jaringan sehingga Subnet pada server dan client diset sama yaitu 255.255.255.0.

- Gateway

  Gateway digunakan bila pada jaringan menggunakan gateway. Pada konfigurasi ini gateway diset sesuai dengan alamat Access Point yaitu 192.168.0.200.

- Local Port

  Local port adalah nomor port pada EGSR-7150MJ (client) yang akan digunakan untuk komunikasi. Nomor port merupakan bilangan bulat yang digunakan untuk membedakan layanan – layanan yang berjalan pada komputer server yang sama. Port pada server dan client harus diset pada angka yang sama Dan pada konfigurasi ini local port diset pada nomor port 5000 sesuai dengan *default port* dari EGSR-7150MJ

- Server IP

  Server IP adalah alamat IP dari komputer (server) yang akan melakukan komunikasi. Pada konfigurasi ini alamat IP diset 192.168.0.100.

- Server Port

  Server Port adalah nomor port dari komputer yang akan digunakan untuk komunikasi. Dan pada konfigurasi ini server port diset pada nomor port sama dengan nomor port EGSR-7150MJ (client) yaitu 5000.

- Operation Mode

  Operation Mode adalah mode operasi yang digunakan oleh EGSR-7150MJ. Pada konfigurasi ini EGSR-7150MJ difungsikan sebagai client, maka operation mode diset sebagai TCP Client.

- TCP Client connection method

  Pada konfigurasi ini TCP Client connection method diset startup yang artinya bila EGSR diaktifkan maka EGSR akan langsung mencari koneksi

pada server sesuai dengan alamat server yang diset pada konfigurasi tersebut.

- Baud

  Baud adalah data rate yang akan digunakan untuk komunikasi serial asinkron dengan mikrokontroler. Pada konfigurasi ini baud diset 9600 bps.

- Databit

  Data bit adalah jumlah bit untuk satu data. Pada komunikasi serial asinkron ini jumlah bit data dalah 8 bit.

- Parity

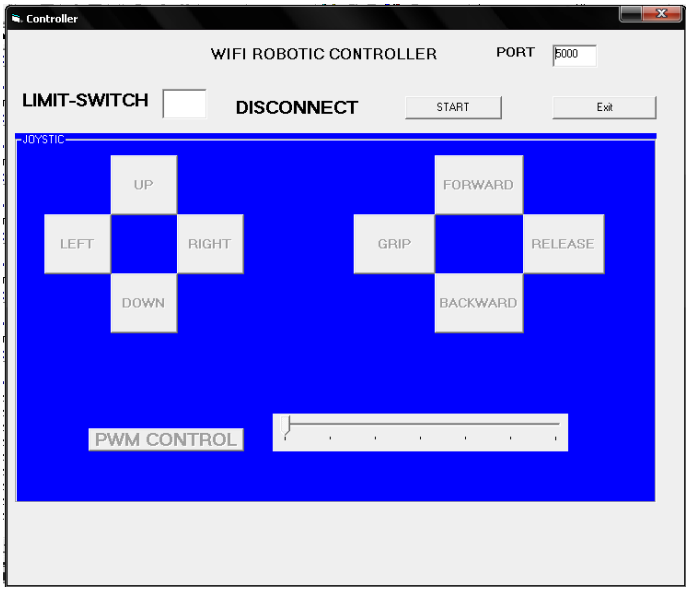
  Pada konfigurasi ini tidak menggunakan parity.

- Flow control

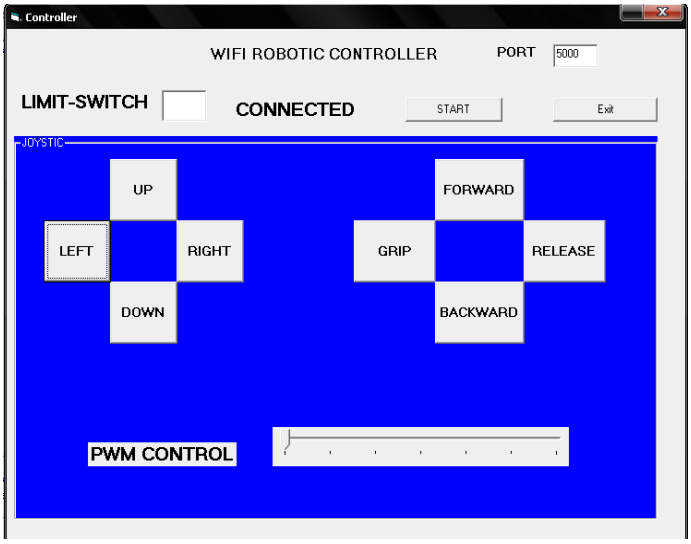
  Pada konfigurasi ini tidak menggunakan flow control.

3.2.1.2 Realisasi Program Aplikasi *User Interface* Pengendali Robot

Pada server dibuat suatu aplikasi *user interface* berbasis Visual Basic yang digunakan untuk mengirimkan perintah / sinyal kendali pada mekanik lengan robot yang dikendalikan tersebut. Pada aplikasi ini menggunakan fasilitas winsock pada Visual Basic sebagai penghubung dengan client.



Gambar 3.8 Tampilan Program Pengendali Robot Sebelum Connect



Gambar 3.9 Tampilan Program Pengendali Robot pada Saat Connect

Pada tampilan program pengendali robot seperti yang terlihat pada Gambar 3.9, server (komputer) akan mengirimkan sinyal kendali berdasarkan pada tombol yang ditekan. Dan client akan memvisualisasikan karakter tersebut dengan gerakan mekanik lengan robot. Adapun keterangan gerakan lengan robot sesuai dengan data karakter yang dikirimkan dapat dilihat pada Tabel III.1.

Tabel III.1 Keterangan Gerakan Lengan Robot Sesuai Karakter yang Dikirimkan

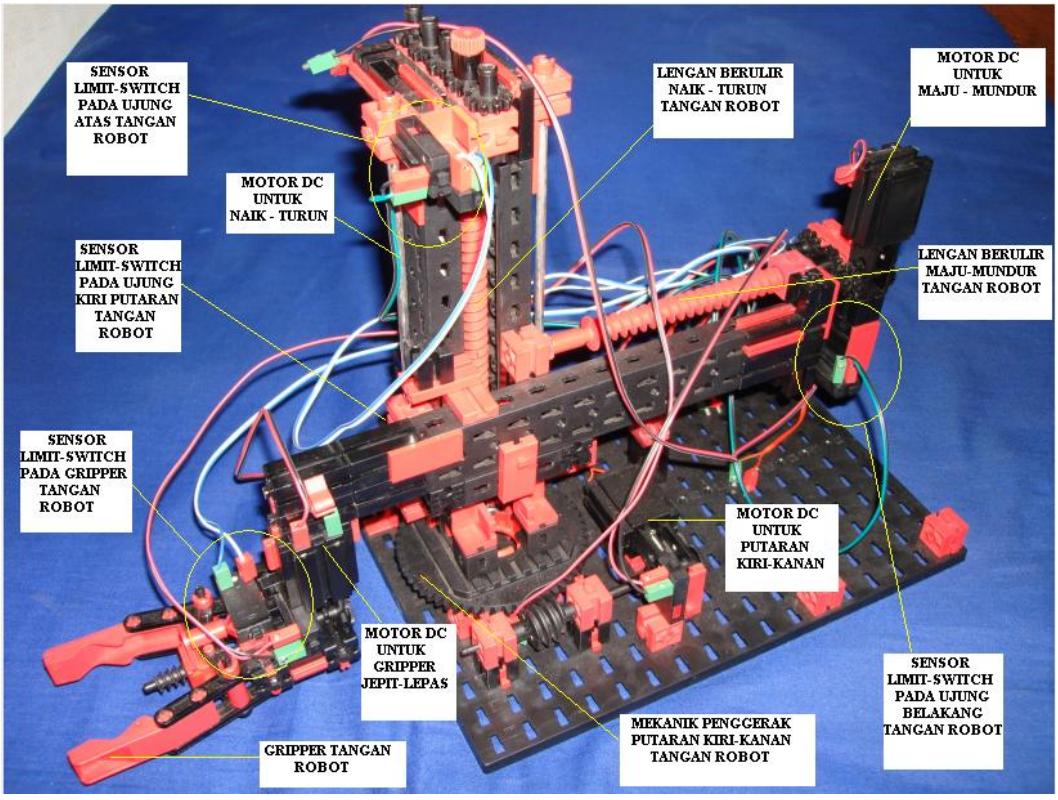
No	Tombol Yang Ditekan Pada Server	Data Karakter Yang Dikirim	Keterangan Gerakan Mekanik Lengan Robot
1	UP	w	Lengan Robot Bergerak Naik
2	DOWN	s	Lengan Robot Bergerak Turun
3	LEFT	a	Lengan Robot Berputar ke Kiri
4	RIGHT	d	Lengan Robot Berputar ke Kanan
5	FORWARD	e	Lengan Robot Bergerak Maju
6	BACKWARD	x	Lengan Robot Bergerak Mundur
7	GRIP	q	Lengan Robot Menjepit
8	RELEASE	z	Lengan Robot Melepas

Pada robot terdapat sensor limit-switch. Jika mekanik lengan robot menekan sensor limit-switch yang terdapat pada robot.. Maka robot akan

mengirimkan sinyal karakter ke server yang mengindikasikan bahwa posisi lengan robot sudah berada pada posisi maksimal dari ujung bagian robot. Keterangan sensor limit-switch dapat dilihat pada Tabel III.2 sedangkan mekanik lengan robot dapat dilhat pada Gambar 3.10.

Tabel III.2 Keterangan Sensor Limit - Switch

No	Data Karakter Yang dikirim	Keterangan Sensor Limit - Switch
1	1	Lengan robot menekan sensor limit-switch pada ujung atas mekanik lengan robot.
2	2	Lengan robot menekan sensor limit-switch pada ujung gripper mekanik lengan robot.
3	3	Lengan robot menekan sensor limit-switch pada ujung belakang mekanik lengan robot.
4	4	Lengan robot menekan sensor limit-switch pada ujung kiri mekanik lengan robot.



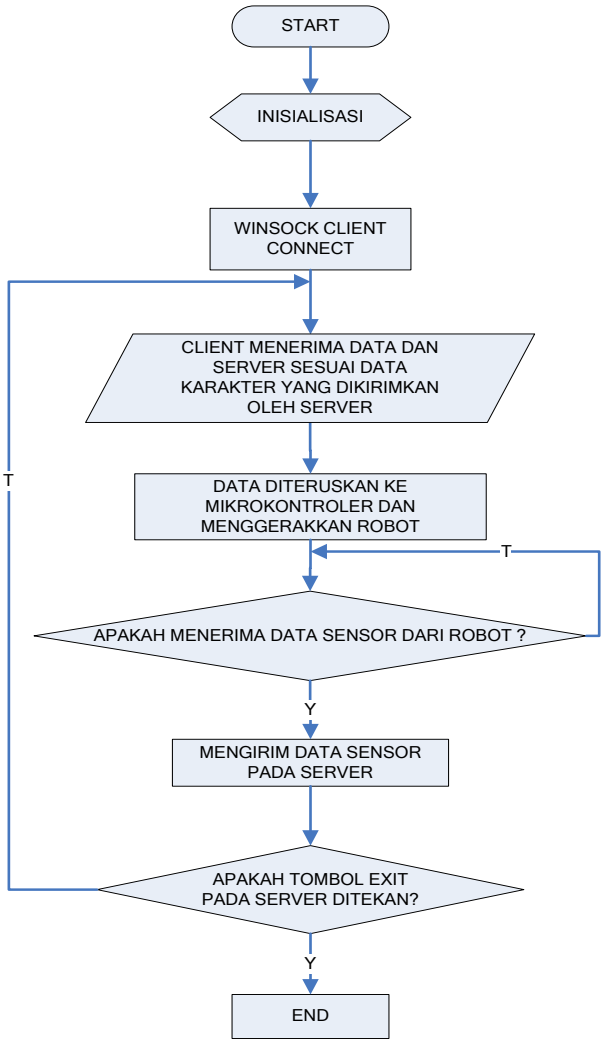
Gambar 3.10 Gambar dan Keterangan Mekanik Lengan Robot

3.2.2 Perancangan dan Realisasi Perangkat Lunak Pada Client.

Perancangan dan realisasi perangkat lunak pada client (robot) dilakukan pada mikrokontroler ATMega 16. Pemrograman pada mikrokontroler dilakukan dengan menggunakan CodeVision AVR yang ditulis meggunakan bahasa C.

Program pada mikrokontroler terdiri dari :

1. Program untuk menerima data karakter yang dikirim dari server (komputer) dan selanjutnya akan menggerakkan robot sesuai dengan karakter yang diterima oleh mikrokontroler.
2. Program untuk mengirim data karakter ke server (komputer) bila mekanik lengan robot menekan sensor limit-switch.



Gambar 3.11 Diagram Alir Pada Client

Pada diagram alir dari Gambar 3.11 dapat dilihat bahwa setelah server (komputer) dan client terkoneksi, maka client siap untuk menerima sinyal kendali yang dikirimkan oleh server. Dan sinyal kendali yang diterima client akan diteruskan ke mikrokontroler untuk menggerakkan robot. Selain itu, apabila mekanik lengan robot menekan sensor limit-switch, maka client akan mengirimkan sinyal pada server. Setelah itu, jika tombol exit pada server ditekan maka proses selesai.

BAB IV
Pengujian dan Analisa Sistem

Pada bab ini akan dibahas pengujian sistem yang terdiri dari pengujian pengiriman data dari notebook ke robot, pengujian pengiriman data dari robot ke notebook, pengujian pengiriman data dari notebook ke mikrokontroler untuk mengendalikan motor servo dan pengujian pengendalian robot untuk melakukan proses *"Pick and Place"*

4.1 Pengujian Pengiriman Data dari Komputer ke Robot.

Pengujian pengiriman data ke robot yaitu komunikasi dengan mikrokontroler di robot yang akan men*driver* motor untuk menggerakkan robot. Sehingga pada tahap ini visualisasi data akan langsung ditunjukkan dengan pergerakan robot. Pengujian ini dilakukan pada lokasi *outdoor* dan *indoor*. Lokasi outdoor yang dimaksud adalah area terbuka pada jalan raya dengan lebar 4 meter pada suatu perumahan. Sedangkan lokasi indoor yang dimaksud adalah area tertutup pada sebuah gedung yang terdiri dari beberapa ruangan, dalam hal ini adalah labotarorium instrumentasi yang terletak di lantai 3 gedung teknik. Pengujian pada tahap ini berjalan dengan baik, robot dapat bergerak sesuai dengan data yang dikirimkan dengan waktu pengiriman data yang bisa dikatakan real time karena kurang dari 1 detik. Dengan data yang didapati sistem komunikasi program antara server dan client berjalan dengan baik..

Tabel IV.1 Pengujian Pengiriman Data ke Robot pada Lokasi *Outdoor*

No	Jarak Transmisi (meter)	(S/N) (dB)	Data yang dikirim	Data yang diterima	Indikator Gerakan Robot	Waktu Transfer (ms)
1	1	-52	w	w	Naik	<1000
2	10	-64	s	s	Turun	<1000
3	20	-72	a	a	Kiri	<1000
4	30	-71	d	d	Kanan	<1000
5	40	-75	q	q	Jepit	<1000
6	50	-77	z	z	Lepas	<1000
7	60	-80	e	e	Maju	<1000
8	70	-84	x	x	Mundur	<1000
9	80	-85	w	w	Naik	<1000

10	90	-82	s	s	Turun	<1000
11	100	-88	a	a	Kiri	<1000
12	110	-83	d	d	Kanan	<1000
13	120	-90	q	q	Jepit	<1000
14	130	-89	e	e	Maju	<1000
15	140	-91	x	x	Mundur	<1000
16	150	-90	w	w	Naik	<1000
17	160	-85	s	s	Turun	<1000
18	170	-92	a	-	Diam	-
19	180	-94	d	-	Diam	-

Pada saat pengujian pengiriman data ke robot pada lokasi *outdoor*, terdapat beberapa kondisi meliputi :

1. Tinggi access point adalah 1.6 meter dari tanah.
2. S/N diukur dengan menggunakan software netstumbler.
3. Pada jarak 0 – 100 meter berada pada kondisi Line Of Sight (LOS).
4. Waktu transfer data diukur dengan menggunakan stopwatch.

Pada pengujian pengiriman data ke robot pada lokasi *outdoor*, dapat disimpulkan bahwa sinyal dapat diterima dengan baik oleh robot sampai jarak 160 meter dalam waktu yang dapat dikatakan *real time* karena kurang dari 1 detik.

Tabel IV.2 Pengujian Pengiriman Data ke Robot pada Lokasi *Indoor*

No	Jarak Transmisi (meter)	(S/N) (dB)	Data yang dikirim	Data yang diterima	Indikator Gerakan Robot	Waktu Transfer (ms)
1	1	-24	w	w	Naik	<1000
2	2	-28	s	s	Turun	<1000
3	5	-32	a	a	Kiri	<1000
4	10	-38	d	d	Kanan	<1000
5	15	-44	w	w	Naik	<1000
6	20	-48	s	s	Turun	<1000
7	25	-54	a	a	Kiri	<1000
8	30	-62	d	d	Kanan	<1000
9	40	-78	w	-	Diam	-
10	50	-92	s	-	Diam	-

Pada saat pengujian pengiriman data ke robot pada lokasi *indoor*, terdapat beberapa kondisi meliputi :

1. Tinggi access point adalah 1.4 meter dari lantai
2. Pada saat pengujian terdapat 4 access point yang sedang aktif dalam area pengujian.
3. S/N diukur dengan menggunakan software netstumbler.
4. Pada jarak 0 sampai 15 meter berada pada kondisi Line Of Sight (LOS).
5. Waktu transfer data diukur dengan menggunakan stopwatch.

Pada pengujian pengiriman data ke robot pada lokasi *indoor*, dapat disimpulkan bahwa sinyal dapat diterima dengan baik oleh robot sampai jarak 30 meter dalam waktu yang dapat dikatakan *real time* karena kurang dari 1 detik.

4.2 Pengujian Pengiriman Data dari Robot ke Komputer.

Pengujian pengiriman data ke robot dilakukan pada 2 lokasi yaitu pada lokasi *outdoor* dan lokasi *indoor*. Pengujian pada tahap ini berjalan dengan baik, robot akan mengirimkan data ke komputer/notebook bila mekanik lengan robot menekan sensor limit-switch yang terdapat pada robot. Pengujian pengiriman data karakter dari robot ke komputer/notebook dilakukan dengan beberapa kondisi yang sama pada saat pengiriman data dari komputer/notebook ke robot. Sehingga kesimpulan yang didapat dari pengujian pengiriman data sensor dari robot ke komputer/notebook sama dengan pengiriman data dari komputer / notebook ke robot.

Tabel IV.3 Pengujian Pengiriman Data ke Komputer pada Lokasi *Outdoor*

No	Jarak Transmisi (meter)	S/N (dB)	Data Karakter yang dikirim	Data Karakter yang diterima	Waktu Transmisi (ms)
1	1	-52	1	1	<1000
2	10	-64	2	2	<1000
3	20	-72	3	3	<1000
4	30	-71	4	4	<1000

5	40	-75	1	1	<1000
6	50	-77	2	2	<1000
7	60	-80	3	3	<1000
8	70	-84	4	4	<1000
9	80	-85	1	1	<1000
10	90	-82	2	2	<1000
11	100	-88	3	3	<1000
12	110	-83	4	4	<1000
13	120	-90	1	1	<1000
14	130	-89	2	2	<1000
15	140	-91	3	3	<1000
16	150	-90	4	4	<1000
17	160	-85	1	1	<1000
18	170	-92	2	-	-
19	180	-94	3	-	-

Tabel IV.4 Pengujian Pengiriman Data ke Komputer pada Lokasi *Indoor*

No	Jarak Transmisi (meter)	S/N (dB)	Data karakter yang dikirim	Data Karakter yang diterima	Waktu transmisi (ms)
1	1	-24	1	1	<1000
2	2	-28	2	2	<1000
3	5	-32	3	3	<1000
4	10	-38	4	4	<1000
5	15	-44	1	1	<1000
6	20	-48	2	2	<1000
7	25	-54	3	3	<1000
8	30	-62	4	4	<1000
9	40	-78	1	-	-
10	50	-92	2	-	-

4.3 Pengujian Pengiriman Data dari Komputer ke Mikrokontroler untuk Mengendalikan Motor Servo HS-325HB.

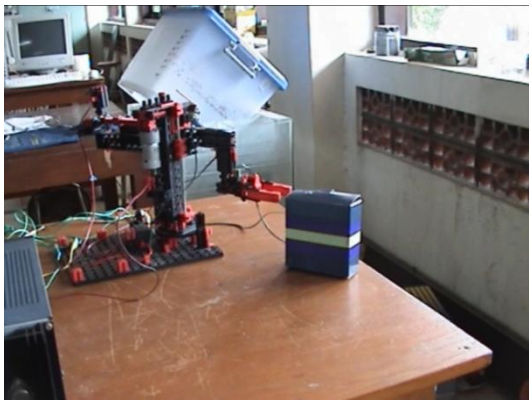
Pada tahap ini dilakukan pengujian pengiriman data ke mikrokontroler unutk mengendalikan motor servo dengan jarak penegendalian sejauh 2 meter dan S/N sebesar -30 dB. Sudut motor servo akan bergerak sesuai dengan data karakter yang dikirimkan. Pengujian pengiriman data ke mikrokontroler untuk mengendalikan motor servo ini berjalan dengan baik, tetapi masih terdapat kesalahan pada sudut motor servo. Hal ini disebabkan oleh ketidakakuratan sudut pada motor servo itu sendiri. Pengukuran sudut motor servo dilakukan dengan menggunakan busur derajat. Pengujian pengiriman data untuk mengendalikan motor servo dapt dilihat pada Tabel IV.5.

Tabel IV.5 Pengujian Pengiriman Data ke Mikrokontroler untuk Mengendalikan Motor Servo HS-325HB

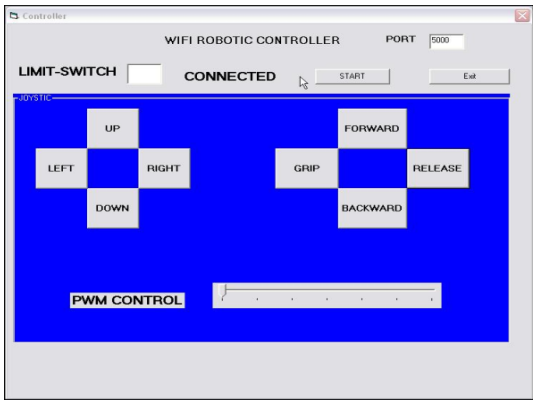
No	Data karakter yang dikirim	Data karakter yang diterima	Sudut Acuan servo	Sudut Motor Servo	Waktu transfer (ms)
1	p	p	0°	2°	<1000
2	o	o	30°	31°	<1000
3	i	i	60°	62°	<1000
4	u	u	90°	93°	<1000
5	y	y	120°	122°	<1000
6	t	t	150°	154°	<1000
7	r	r	180°	184°	<1000

4.4 Pengujian Pengendalian Robot untuk Melakukan Proses "*Pick and Place*".

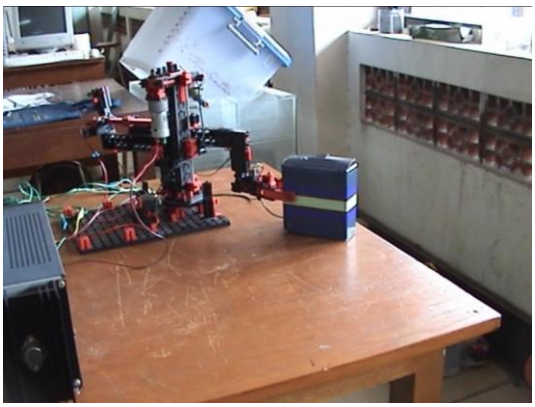
Pada tahap akhir dilakukan pengujian pengendalian pada robot untuk melakukan proses *"Pick and Place"*. Hasilnya robot dapat melakukan proses *"Pick and Place"* dengan baik. Proses pengujian robot dapat dilihat pada Gambar 4.1.



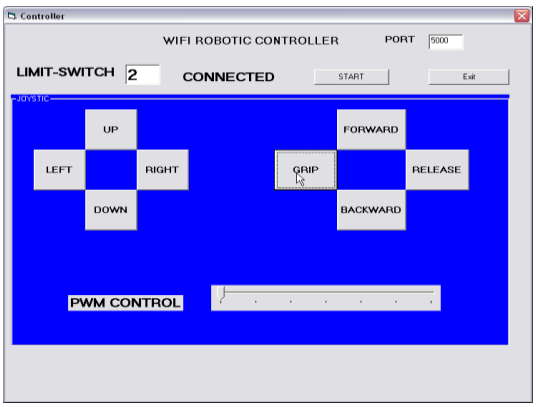
Posisi Awal



Server Connected

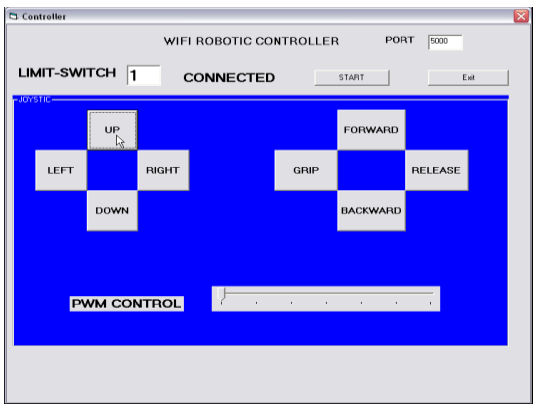


Posisi Saat Menjepit Benda

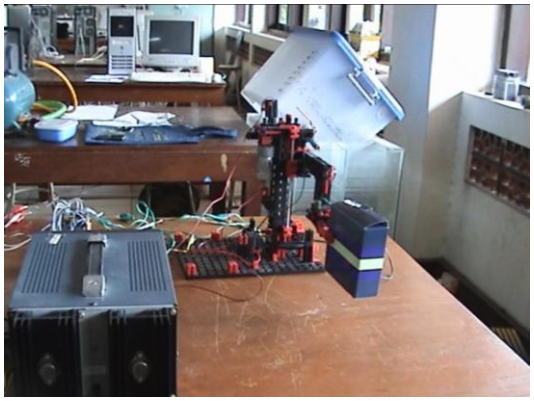


Tombol *Grip* Ditekan

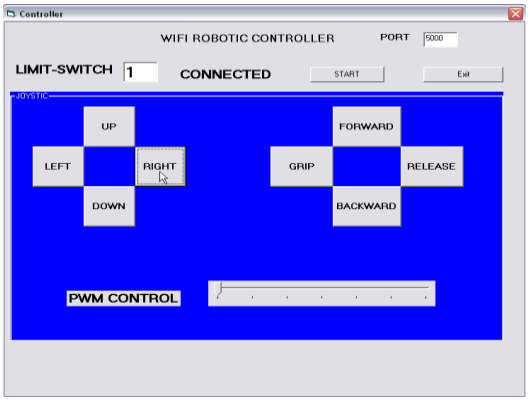
Posisi Saat Mengangkat Benda



Tombol *Up* Ditekan

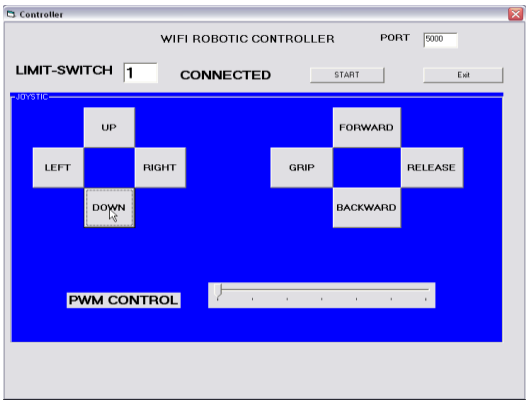


Posisi Saat Memindahkan Benda



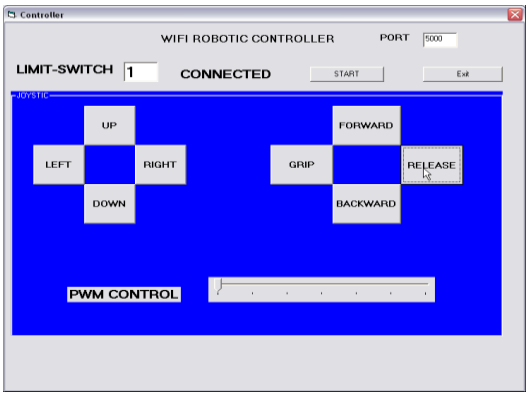
Tombol *Right* Ditekan

Posisi Saat Menurunkan Benda



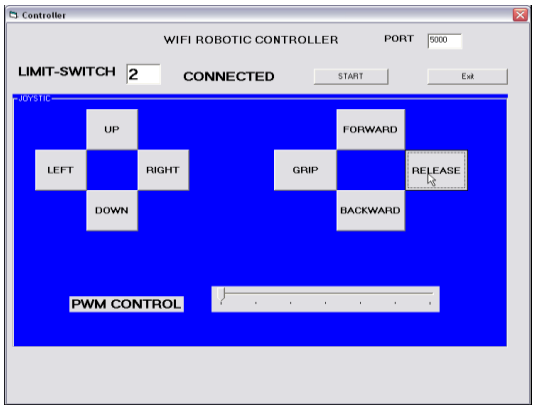
Tombol *Down* Ditekan

Posisi Saat Meletakkan Benda



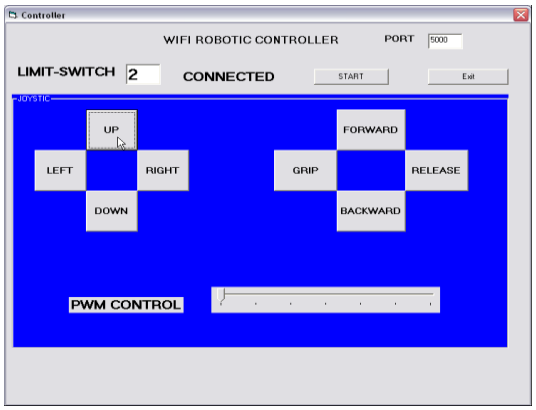
Tombol Release Siap Ditekan

Posisi Saat Melepaskan Benda



Tombol Release Ditekan

Posisi Akhir



Tombol *Up* Ditekan

Gambar 4.1 Pengujian Pengendalian Robot untuk Melakukan Proses "Pick and Place"

BAB V
KESIMPULAN DAN SARAN

V.1 Kesimpulan

Dari analisa dan pengujian yang dilakukan pada sistem, dapat ditarik kesimpulan yaitu :

- Sistem komunikasi *WiFi* yang dibuat dapat digunakan untuk mengendalikan robot dengan baik.
- Kondisi sinyal pada *WiFi* mempengaruhi jangkauan pengendalian robot. Berdasarkan pengujian, jangkauan (*coverage area*) pengendalian robot bisa mencapai ±160 meter pada area terbuka (*outdoor*) dan bisa mencapai ±30 meter pada area tertutup/gedung (*indoor*).
- Komunikasi antara pengirim dan penerima adalah 2 arah dengan waktu transfer data yang masih dapat dikategorikan real time karena waktu transfer data yang cepat (beberapa ms).

V.2 Saran

Pada tugas akhir ini masih terdapat beberapa kekurangan sehingga perlu dilakukan pengembangan. Beberapa saran tentang Tugas Akhir ini adalah :

- *Interface* dapat diaplikasikan untuk komunikasi data yang memerlukan *bandwidth* yang besar mengingat bandwidth frekuensi *WiFi* yang cukup besar.
- Sebaiknya dilakukan proses *flow control* pada *receiver* untuk menghindari *overflow* data yang diterima.
- Sebaiknya lama waktu kerja motor dapat diatur secara otomatis pada server.

DAFTAR PUSTAKA

1. Andi Nalwan, Paulus, Teknik Antarmuka dan Pemrograman Mikrokontroler AT89C51, PT Elex Media Komputindo, Jakarta, 2003.
2. Budiharto Widodo, Rizal Gamayel, Belajar Sendiri 12 Proyek Mikrokontroler untuk Pemula, PT Elex Media Komputindo, Jakarta, 2006.
3. Hartono Jogiyanto, Konsep Dasar Pemrograman Bahasa C, ANDI, Yogyakarta, 1999.
4. Newman Frans, Aplikasi Internet dengan Visual Basic 6.0, PT Elex Media Komputindo, Jakarta, 2001.
5. Prasetya Retna, Edi Widodo Catur, Teori dan Praktek *Interfacing* Port Paralel dan Port Serial Komputer dengan Visual Basic 6.0, ANDI ,Yogyakarta, 2004.
6. Stallings William, Dasar – Dasar Komunikasi Data, Prentice Hall.Inc, New Jersey, 1996.
7. Suhata, ST, VB sebagai Pusat Kendali Peralatan Elektonik, PT Elex Media Komputindo, Jakarta, 2003.
8. Tanenbaum S Andrew, Computer Networks, Prentice Hall.Inc, New Jersey, 1996.
9. Thayer Rob, Visual Basic 6 Unleashed, Sams, Indiana, 2001.
10. W Purbo, Onno, Internet *Wireless* dan Hotspot, PT Elex Media Komputindo, Jakarta, 2005.
11. www.atmel.com
12. www.digi-ware.com
13. www.dlink.com
14. www.netstumbler.com
15. www.*wireless*applications.com
16. www.fischertechnic.com

LAMPIRAN

Lampiran A : Listing Program Aplikasi Pengendali Robot Pada Server.

```
Private Sub Command1_Click()
Unload CONCENTRATOR
End Sub
Private Sub Command2_Click()
Winsock1.Close
Winsock1.Listen
End Sub
Private Sub Command3_Click()
Winsock1.SendData ("w")
End Sub
Private Sub Command4_Click()
Winsock1.SendData ("s")
End Sub
Private Sub Command5_Click()
Winsock1.SendData ("a")
End Sub
Private Sub Command6_Click()
Winsock1.SendData ("d")
End Sub
Private Sub Command7_Click()
Winsock1.SendData ("q")
End Sub
Private Sub Command8_Click()
```

```
Winsock1.SendData ("z")
End Sub
Private Sub Command9_Click()
Winsock1.SendData ("x")
End Sub
Private Sub Command10_Click()
Winsock1.SendData ("e")
End Sub
Private Sub Form_Load()
Command3.Enabled = False
Command4.Enabled = False
Command5.Enabled = False
Command6.Enabled = False
Command7.Enabled = False
Command8.Enabled = False
Command9.Enabled = False
Command10.Enabled = False
Slider1.Enabled = False
Winsock1.LocalPort = Text1.Text
End Sub
Private Sub Slider1_Scroll()
   If Slider1.Value <= 10 Then
      Slider1.Value = 0
      Winsock1.SendData ("p")
   ElseIf Slider1.Value > 11 And Slider1.Value <= 30 Then
      Slider1.Value = 30
      Winsock1.SendData ("o")
   ElseIf Slider1.Value > 30 And Slider1.Value <= 60 Then
      Slider1.Value = 60
      Winsock1.SendData ("i")
   ElseIf Slider1.Value > 60 And Slider1.Value <= 90 Then
      Slider1.Value = 90
```

```
        Winsock1.SendData ("u")
    ElseIf Slider1.Value > 90 And Slider1.Value <= 120 Then
        Slider1.Value = 120
        Winsock1.SendData ("y")
    ElseIf Slider1.Value > 120 And Slider1.Value <= 150 Then
        Slider1.Value = 150
        Winsock1.SendData ("t")
    ElseIf Slider1.Value > 150 And Slider1.Value <= 180 Then
        Slider1.Value = 180
        Winsock1.SendData ("r")
    End If
End Sub
Private Sub Text1_Change()
Winsock1.LocalPort = Text1.Text
End Sub
Private Sub Timer1_Timer()
Winsock1.SendData ("0")
End Sub
Private Sub Winsock1_Connect()
Label3.Caption = "CONNECTED"
Command3.Enabled = True
Command4.Enabled = True
Command5.Enabled = True
Command6.Enabled = True
Command7.Enabled = True
Command8.Enabled = True
Command9.Enabled = True
Command10.Enabled = True
Slider1.Enabled = True
Timer1.Enabled = True
End Sub
Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
```

```
Label3.Caption = "CONNECTED"
Command3.Enabled = True
Command4.Enabled = True
Command5.Enabled = True
Command6.Enabled = True
Command7.Enabled = True
Command8.Enabled = True
Command9.Enabled = True
Command10.Enabled = True
Slider1.Enabled = True
If Winsock1.State <> sckClosed Then Winsock1.Close
Winsock1.Accept requestID
Winsock1.SendData "Connection Succes"
End Sub
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Dim dat As String
Winsock1.GetData dat, vbString
Text2.Text = dat
End Sub
Private Sub Winsock1_Error(ByVal Number As Integer, Description As String,
ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String,
ByVal HelpContext As Long, CancelDisplay As Boolean)
MsgBox "Error : " & Err.Description
Winsock1.Close
Winsock1.Listen
End
End Sub
```

Lampiran B : Listing Program CodeVision AVR Mikrokontroler ATMega 16 Pada Client.

```
This program was produced by the
CodeWizardAVR V1.25.3 Professional
Automatic Program Generator
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 11/15/2007
Author  : F4CG
Company : F4CG
Comments:


Chip type           : ATmega16
Program type        : Application
Clock frequency     : 12.000000 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 256
*****************************************************/

#include <mega16.h>
#include<delay.h>
char temp;
eeprom int ax=0;
int a;

#define RXB8 1
```

```
#define TXB8 0
#define UPE 2
#define OVR 3
#define FE 4
#define UDRE 5
#define RXC 7

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<OVR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// USART Receiver buffer
#define RX_BUFFER_SIZE 8
char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE<256
unsigned char rx_wr_index,rx_rd_index,rx_counter;
#else
unsigned int rx_wr_index,rx_rd_index,rx_counter;
#endif

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;
```

```
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
   {
   rx_buffer[rx_wr_index]=data;
   if (++rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
   if (++rx_counter == RX_BUFFER_SIZE)
      {
      rx_counter=0;
      rx_buffer_overflow=1;
      };
   };
a = 1;
}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter==0);
data=rx_buffer[rx_rd_index];
if (++rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
#asm("cli")
--rx_counter;
#asm("sei")
return data;
}
#pragma used-
#endif
```

```
// USART Transmitter buffer
#define TX_BUFFER_SIZE 8
char tx_buffer[TX_BUFFER_SIZE];

#if TX_BUFFER_SIZE<256
unsigned char tx_wr_index,tx_rd_index,tx_counter;
#else
unsigned int tx_wr_index,tx_rd_index,tx_counter;
#endif

// USART Transmitter interrupt service routine
interrupt [USART_TXC] void usart_tx_isr(void)
{
if (tx_counter)
   {
   --tx_counter;
   UDR=tx_buffer[tx_rd_index];
   if (++tx_rd_index == TX_BUFFER_SIZE) tx_rd_index=0;
   };
}

#ifndef _DEBUG_TERMINAL_IO_
// Write a character to the USART Transmitter buffer
#define _ALTERNATE_PUTCHAR_
#pragma used+
void putchar(char c)
{
while (tx_counter == TX_BUFFER_SIZE);
#asm("cli")
if (tx_counter || ((UCSRA & DATA_REGISTER_EMPTY)==0))
   {
   tx_buffer[tx_wr_index]=c;
```

```
  if (++tx_wr_index == TX_BUFFER_SIZE) tx_wr_index=0;
  ++tx_counter;
  }
else
  UDR=c;
#asm("sei")
}
#pragma used-
#endif

// Standard Input/Output functions
#include <stdio.h>

// Declare your global variables here

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTA=0x00;
DDRA=0xFF;

// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTB=0x00;
```

```
DDRB=0xFF;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTD=0x00;
DDRD=0xFF;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 1500.000 kHz
// Mode: CTC top=OCR1A
// OC1A output: Set
// OC1B output: Discon.
// Noise Canceler: Off
```

```
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0xC0;
TCCR1B=0x0A;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0xD8;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x4D;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;


// Watchdog Timer initialization
// Watchdog Timer Prescaler: OSC/2048k
#pragma optsize-
WDTCR=0x1F;
WDTCR=0x0F;
#ifdef _OPTIMIZE_SIZE_
#pragma optsize+
#endif
```

```
// Global enable interrupts
#asm("sei")

DDRD.1=1;
DDRA = 0x00;
PORTA = 0xFF;

while (1)
      {
      // Place your code here
if(a ==1)
{
 temp = getchar();          //ambil karakter dari komputer
 a=0;
};

 if (PINA.0 == 0)           //cek sensor limit switch
 {
 putchar('1');
};
 if (PINA.1 == 0)
 {
 putchar('2');
};
 if (PINA.2 == 0)
 {
 putchar('3');
};
 if (PINA.3 == 0)
 {
 putchar('4');
};
```

```
 if (PINA.4 == 0)
 {
 putchar('5');
};
  switch (temp)
      {
      case 'w':PORTB = 0x01;
      delay_ms (500);
      break;
      case 's':PORTB = 0x02;
      delay_ms (500);
      break;
      case 'a':PORTB = 0x04;
      delay_ms (500);
      break;
      case 'd':PORTB = 0x08;
      delay_ms (500);
      break;
      case 'q':PORTB = 0x10;
      delay_ms (500);
      break;
      case 'z':PORTB = 0x20;
      delay_ms (500);
      break;
      case 'e':PORTB = 0x40;
      delay_ms (500);
      break;
      case 'x':PORTB = 0x80;
      delay_ms (500);
      break;
     case ‘p’: OCR1A = 1C2;
     break;
```

```
    case ‘o’: OCR1A = 2A3;
    break;
    case ‘i’: OCR1A = 384;
    break;
    case ‘u’: OCR1A = 465;
    break;
    case ‘y’: OCR1A = 546;
    break;
    case ‘t’: OCR1A = 627;
    break;
   case ‘r’: OCR1A = 708;
    break;
     }
     PORTB = 0x00;
      temp = 0;
     };

}
```

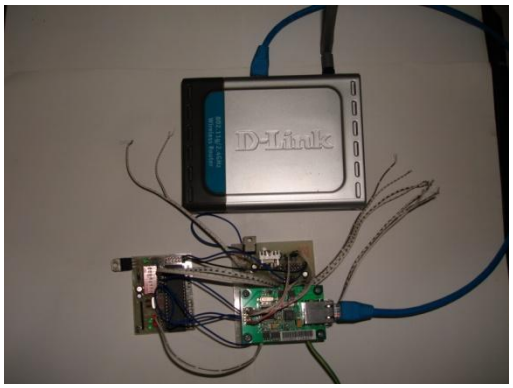
LAMPIRAN C : Spesifikasi Access Point D-Link DI-524

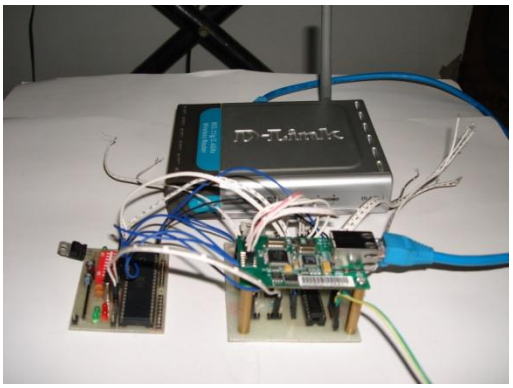
Specifications	
Standards	• IEEE 802.11g • IEEE 802.11b • IEEE 802.3 • IEEE 802.3u
Wireless Signal Rates* with Automatic Fallback	• 54Mbps • 48Mbps • 36Mbps • 24Mbps • 18Mbps • 12Mbps • 11Mbps • 9Mbps • 6Mbps

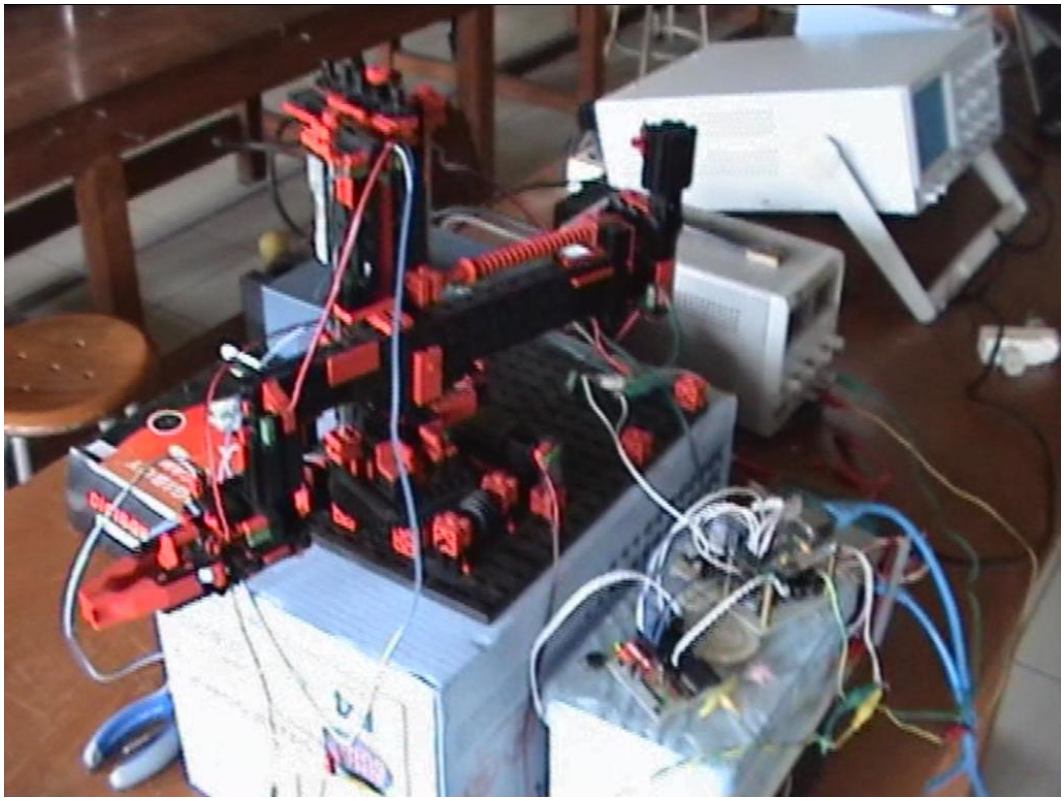
	• 5.5Mbps • 2Mbps • 1Mbps
Security	• 802.1X • 64-, 128-bit WEP • WPA — Wi-Fi Protected Access (WEP with TKIP, MIC, IV Expansion, Shared Key Authentication)
Modulation Technology	• Orthogonal Frequency Division Multiplexing (OFDM)
Receiver* Sensitivity	• 54Mbps OFDM, 10% PER,-68dBm) • 48Mbps OFDM, 10% PER,-68dBm) • 36Mbps OFDM, 10% PER,-75dBm) • 24Mbps OFDM, 10% PER,-79dBm) • 18Mbps OFDM, 10% PER,-82dBm) • 12Mbps OFDM, 10% PER,-84dBm) • 11Mbps CCK, 8% PER,-82dBm) • 9Mbps OFDM, 10% PER,-87dBm) • 6Mbps OFDM, 10% PER,-88dBm) • 5.5Mbps CCK, 8% PER,-85dBm) • 2Mbps QPSK, 8% PER,-86dBm) • 1Mbps BPSK, 8% PER,-89dBm)
VPN Pass Through/Multi-Sessions	• PPTP • L2TP • IPSec
Device Management	• Web-Based – Internet Explorer v6 or later; Netscape Navigator v6 or later; or other Java- enabled browsers. • DHCP Server and Client
Advanced Firewall Features	• NAT with VPN Pass-through (Network Address Translation) • MAC Filtering • IP Filtering • URL Filtering • Domain Blocking • Scheduling
Wireless Signal Range*	• *Indoor*s: Up to 328 ft (100 meters) • *Outdoor*s: Up to 1312 ft (400 meters)
Wireless Frequency Range	• 2.4GHz to 2.462GHz
Wireless Transmit Power	• 15dBm ± 2dBm
External Antenna Type	• Single detachable reverse SMA
Operating Temperature	• 32°F to 131°F (0°C to 55°C)
Humidity	• 95% maximum (non-condensing)
Safety & Emissions	• FCC
LEDs	• Power

	• Status • WAN • WLAN (Wireless Connection) • LAN (10/100)
Dimensions	• L = 5.6 inches (142mm) • W = 4.3 inches (109mm) • H = 1.2 inches (31mm)
Weight	• 0.49lbs (22g)
Warranty	• 3 Year

LAMPIRAN D : FOTO ALAT DAN ROBOT

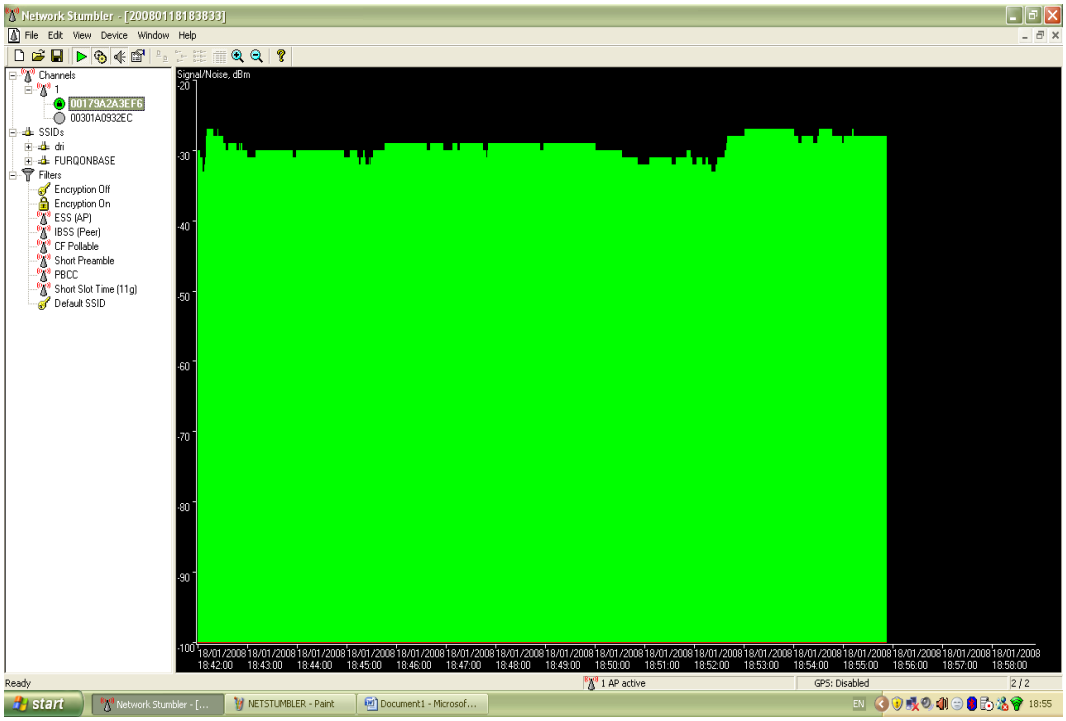






LAMPIRAN E : FOTO PENGUJIAN SISTEM

LAMPIRAN F : *SOFTWARE* NETSTUMBLER



Untuk melakukan site survey, cukup mudah dengan NetStumbler. NetStumbler akan jalan dan otomatis mencari/*scanning* frekuensi untuk melihat Access Point yang sedang beroperasi pada frekuensi *WiFi*. NetStumbler akan melaporkan :

- MAC address Access Point dan frekuensinya.
- Channel yang digunakan Access Point.
- ESSID Access Point.
- Nama Access Point.
- Kekuatan Sinyal yang diterima Access Point. Terdiri dua warna yaitu hijau yang mengindikasikan baik dan kuning yang mengindikasikan kurang baik.
- Signal to Noise Ratio (SNR).

NetStumbler akan memperlihatkan rekaman grafik dari Signal to Noise Ratio (SNR) sebagai fungsi waktu. Kita dapat menyimpan hasil laporan sinyal untuk perioda tertentu.