

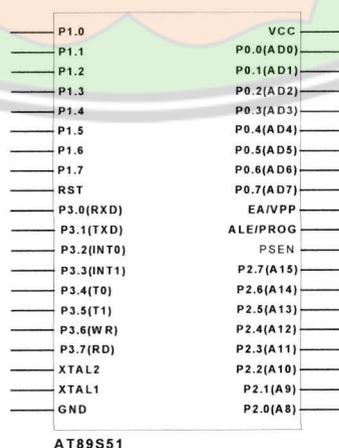
BAB II

LANDASAN TEORI

2.1. Perangkat Keras

2.1.1. Mikrokontroler AT89s51^[6]

Atmel 89s51 adalah mikrokontroler dengan arsitektur MCS51 dengan 4Kb Flash PEROM (*Programmable and Erasable Read Only Memory*) yaitu ROM yang dapat ditulis dan dihapus menggunakan perangkat *programmer*, memiliki 128 x 8 bit *Internal RAM*, 32 *Programable I/O lines* dan mempunyai 40 kaki, 32 kaki diantaranya adalah kaki untuk keperluan *port* paralel. Satu *port* paralel terdiri dari 8 kaki, dengan demikian 32 kaki tersebut membentuk 4 buah *port* paralel, yang masing-masing dikenal sebagai *port-0*, *port-1*, *port-2*, *port-3*. Nomor dari masing-masing jalur/kaki dari *port* paralel mulai dari 0 sampai 7. Jalur/kaki pertama *port-0* disebut sebagai P0,0 dan jalur terakhir untuk *port-3* adalah P3.7. Struktur kaki mikrokontroler dapat dilihat pada gambar 2.1.



Gambar 2.1. Struktur kaki mikrokontroler AT89S51^[1]

Adapun nama dan fungsi dari kaki-kaki (pin) pada mikrokontroler AT89S51 adalah sebagai berikut :

1. RST (*Reset*)

Masukan *reset* kondisi 1 selama 2 siklus mesin selama osilator bekerja akan mereset mikrokontroler yang bersangkutan ke alamat awal.

2. ALE/PROG (*Address Latch Enable*)

Keluaran ALE atau *Address Latch Enable* menghasilkan pulsa-pulsa untuk mengunci bit rendah (*low byte*). Selama mengakses memori eksternal. Kaki ini juga berfungsi sebagai masukan pulsa program (*the program pulse input*) atau PROG selama pemrograman *flash*. Pada operasi normal, ALE akan berpulsa dengan laju 1/6 dari frekuensi kristal dan dapat digunakan sebagai pewaktuan (*timing*) atau pendetakan (*clocking*) rangkaian eksternal, ada satu pulsa yang dilompati selama pengaksesan memori data eksternal. Jika dikehendaki operasi ALE bisa dimatikan dengan cara mengatur bit 0 dari SFR lokasi 8EH, jika isinya 1, ALE hanya akan aktif selama dijumpai intruksi MOVX atau MOVC. Selain itu, kaki ini akan secara lemah di *pulled-high*, mematikan bit ALE tidak akan ada efeknya jika mikrokontroler mengeksekusi program secara *external* bisa juga untuk memultiplexing antara *address* dan *address data bus*.

3. PSEN

Program Store Enable merupakan sinyal baca untuk memori program eksternal, saat mikrokontroler keluarga 51 menjalankan program dari memori eksternal, PSEN akan diaktifkan dua kali persiklus mesin, kecuali dua aktivasi PSEN dilompati (diabaikan) saat mengakses memori data eksternal.

4. EA/Vpp

External Acces Enable. EA harus selalu dihubungkan ke *ground*, jika mikrokontroler akan mengeksekusi program dari memori eksternal lokasi 0000H hingga FFFFH, selain itu EA harus dihubungkan ke Vcc agar mikrokontroler mengakses program secara *internal*. Kaki ini juga berfungsi menerima tegangan 12 Volt (Vpp) selama pemrograman *flash*.

5. *Port-0* , *Port1* , *Port2*, dan *Port3*, Berfungsi sebagai I/O data.

6. Vcc, Berfungsi sebagai suplai tegangan.

7. GND, *Ground* atau Pembedaan.

2.1.2. Struktur Memori^[6]

AT89S51 mempunyai struktur memori yang terdiri dari :

1. RAM *Internal*

RAM *Internal* memiliki memori sebesar 128 *byte* yang biasanya digunakan untuk menyimpan variabel atau data yang

bersifat sementara, dialamati oleh *RAM Address Register* (Register Alamat RAM).

RAM *Internal* terdiri atas :

a. *Register Banks*

AT89S51 memiliki delapan buah register yang terdiri dari R0 sampai R7 yang terletak pada alamat 00H hingga 07H pada setiap kali *reset*.

b. *Bit Addressable RAM*

RAM dengan alamat 20H hingga 2FH dapat diakses secara pengalamatan bit (*bit addressable*) sehingga hanya dengan sebuah instruksi saja setiap bit dalam area ini dapat di-set, *clear*, AND dan OR.

c. RAM Keperluan Umum

RAM keperluan umum dimulai dari alamat 30H hingga 7FH dan dapat diakses dengan pengalamatan langsung maupun tak langsung. Pengalamatan langsung dilakukan ketika salah satu *operand* merupakan bilangan yang dialamati. Sedangkan pengalamatan tak langsung pada lokasi dari RAM internal ini adalah akses data dari memori ketika alamat memori tersebut tersimpan dalam suatu register R0 atau R1 yang dapat digunakan sebagai *pointer* dari lokasi memori pada RAM internal.

d. *Special Function Register* (Register Fungsi Khusus)

Memori yang berisi register-register yang memiliki fungsi khusus yang disediakan oleh mikrokontroler, seperti *timer*, serial dan lain-lain. 89S51 memiliki 21 *Special Function Register* yang terletak pada alamat 80H hingga FFH. Salah satu contoh dari register fungsi khusus adalah *accumulator*, register ini terletak pada alamat E0H. Semua operasi aritmatika dan operasi logika dan proses pengambilan dan pengiriman data ke memori selalu menggunakan register ini.

e. *Flash PEROM*

Memori yang digunakan untuk menyimpan instruksi-instruksi MCS-51 dialamati oleh *Program Address Register* (Register Alamat Program). AT89S51 memiliki 4 Kb *Flash PEROM* yang menggunakan *Atmel's High-Density Non Volatile Technology*. Program yang ada pada *Flash PEROM* akan dijalankan jika pada saat sistem di-*reset*, pin EA/VP berlogika satu maka mikrokontroler aktif berdasarkan program yang ada pada *Flash PEROM*nya. Namun jika pin EA/VP berlogika nol, mikrokontroler aktif berdasarkan program yang ada pada memori eksternal.

2.1.2.1. Set Instruksi Program ASM51^[6]

Untuk dapat menjalankan mikrokontroler, pada *Flash PEROM*-nya harus diisi program yang kita buat dalam *format file* berextension *.HEX*. Program yang dibuat menggunakan bahasa *assembler* (*.ASM*). File *assembler* ini kemudian di-*compile* ke dalam *file* berbentuk *.HEX* oleh perangkat lunak yang disediakan oleh produsen, lebih dikenal dengan nama ASM51. Sintak instruksi program sumber *assembler* sebagai berikut :

```
Mnemonic      operand1,Operand2,Operand3
;Komentar
Contoh :      Mov    A, 10h      ;Copy data pada alamat
10h ke reg A
```

Mnemonic adalah kode program *assembler* yang akan dikerjakan oleh mikrokontroler. Kode-kode ini bergantung dari jenis mikrokontroler yang digunakan. Keluarga MCS-51 menggunakan kode-kode seperti : MOV, ADD INC, DEC CJNE, JMP, CALL, SUBB, SET, CLR dan lain-lain.

Operand ditulis setelah *mnemonic*, bisa berupa alamat atau data yang digunakan dalam instruksi yang bersangkutan. Jumlah *operand* yang mengikuti *mnemonic* bisa satu, dua ataupun tiga *operand* yang dipisahkan oleh tanda koma. Tetapi ada pula *mnemonic* yang tidak memerlukan *operand* seperti RET dan NOP pada MCS-51. Komentar harus selalu diawali dengan titik koma

(;). Komentar diperlukan untuk menjelaskan bagian dari program yang ditulis agar tidak membingungkan pembuat program itu sendiri.

2.1.2.2. Assembler Directive^[3]

Assembler Directive digunakan untuk mengatur kerja dari program *assembler*. *Mnemonic* untuk *Assembler Directive* tergantung pada program *assembler* yang dipakai. Ada beberapa *assembler directive* yang umum, yang sama untuk banyak macam program *assembler* diantaranya adalah :

1. ORG (*Origin*)

Digunakan untuk menyatakan lokasi memori tempat instruksi atau perintah yang ada di bawahnya disimpan.

2. EQU (*Equate*)

EQU digunakan untuk mendefinisikan sebuah simbol atau lambang *assembler* secara bebas.

3. DB (*Define Byte*)

Digunakan untuk memberikan nilai tertentu pada memori-program.

4. DW (*Define Word*)

Dipakai untuk memberi nilai 2 *byte* ke *memory-program* pada baris bersangkutan. *Assembler directive* ini biasa

dipakai untuk membentuk suatu tabel yang isinya adalah nomor-nomor *memory-program*.

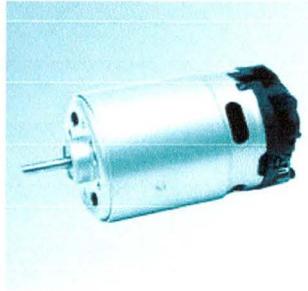
5. DS (*Define Storage*)

Assembler directive ini dipakai untuk membentuk variabel. Sebagai variabel tentu saja memori yang dipakai adalah *memory-data* (RAM) bukan *memory program* (ROM).

2.1.2. Aktuator^[4]

Aktuator merupakan perangkat elektromekanik yang menghasilkan daya gerakan. Dapat dibuat dari sistem motor listrik, sistem *pneumatic*, atau perangkat hidrolik. Untuk meningkatkan tenaga mekanik aktuator atau torsi gerakan dapat dipasang sistem *gearbox*, baik sistem *direct-gear* (sistem lurus, sistem *ohmic/worm-gear*, *planetary gear*, dsb) maupun *sprocket-chain* (gir-rantai, *gir-belt*, ataupun sistem *wire-roller*, dsb).

Aktuator dasar yang sering digunakan pada sistem robot sebagai penghasil gerakan yaitu motor DC magnet permanen. Motor DC (*direct current*) adalah peralatan elektromekanik dasar yang berfungsi untuk mengubah tenaga listrik menjadi tenaga mekanik. Secara umum, kecepatan putaran poros motor DC akan meningkat seiring dengan meningkatnya tegangan yang diberikan. Dengan demikian, putaran motor DC akan berbalik arah jika polaritas tegangan yang diberikan juga dirubah. Berikut ini adalah sebuah contoh motor DC-MP tanpa *gearbox* seperti yang terlihat pada gambar 2.2.



Gambar 2.2. Contoh Motor dc magnet permanen

Motor DC mempunyai tegangan kerja yang bervariasi. Ada yang memiliki tegangan kerja 3 V, 6 V, 12 V dan sebagainya. Motor DC biasanya masih bisa beroperasi bila diberikan tegangan yang lebih kecil atau lebih besar dari tegangan kerjanya, tetapi masing-masing mempunyai keuntungan dan kerugiannya.

Penggunaan motor *gear* DC sebagai komponen penggerak didasarkan selain atas putaran dan torsi yang lebih besar dibandingkan *stepper* motor maupun *servo* motor, juga didasarkan atas ketersediaannya di pasaran dengan variasi yang banyak dan harga yang murah, walaupun tidak dilengkapi dengan data-data spesifikasi/parameter motor secara lengkap.

Untuk men-*drive* motor DC digunakan *Smart Peripheral Controller/ SPC DC Motor*, yaitu *Driver* tipe mikrokontrol AT89C2051 dan IC L293D keluaran *innovatives* yang mampu mengendalikan motor

DC bertegangan kerja sampai 46 VDC dan arus total sampai 4A.

2.1.3. Catu Daya ^[4]

Catu daya memegang peranan yang sangat penting dalam hal perancangan sebuah *robot*. Tanpa bagian ini *robot* tidak akan berfungsi. Begitu juga bila pemilihan catu daya tidak tepat, maka *robot* tidak akan bekerja dengan baik.

Penentuan sistem catu daya yang akan digunakan ditentukan oleh banyak faktor, diantaranya :

1. Tegangan

Setiap aktuator tidak memiliki tegangan yang sama. Hal ini akan berpengaruh terhadap disain catu daya. Tegangan tertinggi dari salah satu aktuator akan menentukan nilai tegangan catu daya.

2. Arus

Arus memiliki satuan Ah (*Ampere-hour*). Semakin besar Ah, semakin lama daya tahan baterai bila digunakan pada beban yang sama.

3. Teknologi Baterai

Baterai isi ulang ada yang dapat diisi hanya apabila benar-benar kosong, dan ada pula yang dapat diisi ulang kapan saja tanpa harus menunggu baterai tersebut benar-benar kosong.

Baterai yang digunakan pada *mobile robot* ini adalah baterai berjenis *Nickel cadmium baterai*. Dimana *Nickel cadmium* (Ni-cd)

adalah jenis baterai yang umum dipakai pada mobil mainan (Tamiya). Hal ini tidak lain adalah karena jenis baterai ini merupakan jenis baterai yang dapat diisi ulang (*rechargeable*). Baterai ini memiliki tegangan kerja 1,2 Volt.

Baterai Ni-cd memiliki kelemahan yang dinamakan ‘*memory effect*’ yaitu kapasitas baterai akan berkurang jika baterai tersebut tidak dikosongkan secara penuh sebelum mengisinya kembali (*rechargeable*). Berikut ini adalah contoh sebuah baterai Ni-cd 700 mAh seperti yang terlihat dalam gambar 2.3.



Gambar 2.3. Baterai Ni-cd 700 mAh

Untuk pengisian baterai ini dapat dilakukan kapan saja, namun untuk hasil yang lebih baik, setelah beberapa kali pengisian, baterai ini harus dikosongkan terlebih dahulu sebelum diisi.

Selain jenis baterai Ni-cd, masih banyak lagi jenis baterai yang tersedia di pasaran dengan spesifikasi yang beragam dan dapat digunakan untuk catu daya pada sebuah robot. Diantaranya baterai Ni-

MH, *Alkaline, Lithium, Lead Acid* dan sebagainya, yang masing-masing mempunyai kelebihan dan kekurangannya.

2.2. Robot ^[7]

Kata robot berasal dari bahasa Czech, *robota*, yang berarti pekerja. Kata robot diperkenalkan dalam bahasa Inggris pada tahun 1921 oleh *Wright Karel Capek* dalam satu drama satiriknya, *R.U.R. (Rossum's Universal Robots)*. **Robot** adalah mesin hasil rakitan manusia yang bias bekerja tanpa mengenal lelah. Awalnya, robot diciptakan sebagai pengganti tenaga manusia. Namun, untuk jangka waktu kedepan, robot akan mengambil alih posisi manusia sepenuhnya dan bahkan mengganti ras manusia dengan beragam jenis robot (Fu, K. S., Gonzales, R. C. et.al, 1987:1 dan 4). Akhir-akhir ini telah banyak dilakukan penelitian dibidang robotika termasuk Negara kita (Indonesia) yang juga turut berperan dalam beberapa penelitian dan berinovasi dalam pembuatan robot (Suyadhi, 2008).

Kata Robotics juga berasal dari novel fiksi sains "runaround" yang ditulis oleh Isaac Asimov pada tahun 1942. Sedangkan pengertian robot secara tepat adalah *system* atau alat yang dapat berperilaku atau meniru perilaku manusia dengan tujuan untuk menggantikan dan mempermudah kerja/aktifitas manusia.

Untuk dapat diklasifikasikan sebagai robot, mesin harus memiliki dua macam kemampuan yaitu:

1. Bisa mendapatkan informasi dari sekelilingnya.
2. Bisa melakukan sesuatu secara fisik seperti bergerak atau memanipulasi objek.

Untuk dapat dikatakan sebagai robot sebuah *system* tidak perlu untuk meniru semua tingkah laku manusia, namun suatu sistem tersebut dapat mengadopsi satu atau dua saja sistem yang ada pada diri manusia saja sudah dapat dikatakan sebagai robot. Sistem yang diadopsi berupa sistem penglihatan (mata), sistem pendengaran (telinga) ataupun sistem gerak.

Sebuah robot dapat saja dibuat untuk berbagai macam aktifitas, namun sebuah robot harus dibuat dengan tujuan untuk kebaikan manusia. Ada hukum robotika yang perlu dipegang sebelum seseorang terjun dalam robotika, antara lain:

1. Robot tidak boleh menciderai manusia atau dalam keadaan tanpa aksi mengijinkan manusia mendekat untuk disakiti.
2. Robot harus menuruti perintah yang diberikan oleh manusia kecuali jika perintah tersebut bertentangan dengan hukum yang pertama.
3. Robot harus melindungi eksistensinya, selama tidak bertentangan dengan hukum pertama dan kedua.

2.2.1. Kegunaan dari robot

Sesuai dengan definisi robot di atas, maka terdapat banyak kegunaan dari robot. Secara umum kegunaan robot adalah untuk menggantikan kerja manusia yang membutuhkan ketelitian yang tinggi atau mempunyai resiko yang sangat besar atau bahkan mengancam keselamatan manusia. Sebagai contoh, seseorang yang bekerja di bagian *welding* di sebuah industri *assembling* kendaraan, akan mempunyai resiko kecelakaan kerja yang cukup tinggi. Maka untuk mengurangi resiko kerja tersebut perlu digunakan robot yang menggantikan kerja manusia di

bidang tersebut, sehingga resiko kecelakaan kerja dapat dikurangi bahkan dihilangkan.

Ada juga sebagian robot yang sengaja diciptakan untuk menemani manusia di dalam aktifitasnya. Robot-robot ini dapat disebut robot bermain. Robot ini diciptakan untuk membantu manusia yang mengalami kesepian diri sehingga dapat mempunyai teman. Robot-robot yang termasuk jenis ini termasuk antar lain *Battle Bots*, Robot *contesti*, Robot Anjing.

Namun secara garis besar robot dapat diklasifikasikan menjadi beberapa jenis antara lain:

- 1) Robot pribadi
- 2) Robot Industri
- 3) Robot Pendidikan
- 4) Robot antariksa
- 5) Robot transportasi
- 6) Robot perang
- 7) Robot kendali jarak jauh
- 8) Robot kedokteran
- 9) Robot riset
- 10) Robot bermain, dll



2.2.2. Robot Era Modern

Robot era modern kini lebih banyak dikembangkan untuk riset daripada penyerupaan manusia. Sehingga memiliki kecerdasan sendiri bahkan ada rencana untuk membuat robot yang bisa ditandingkan dengan kecerdasan menyerupai manusia.

2.2.3. Pembuatan Robot

Seperti yang telah diuraikan sebelumnya, robot dapat membantu meringankan pekerjaan manusia. Adapun tujuan pembuatan robot memang untuk kebaikan manusia. Oleh karena itu, sebelum terjun dalam bidang robotika, ada beberapa hal yang perlu diperhatikan, yaitu:

1. Robot tidak boleh menciderai manusia atau dalam keadaan tanpa aksi mengijinkan manusia mendekat untuk disakiti.
2. Robot harus menuruti perintah yang diberikan oleh manusia kecuali jika perintah tersebut bertentangan dengan hukum yang pertama.
3. Robot harus melindungi eksistensinya, selama tidak bertentangan dengan hukum pertama dan kedua.

Ada tiga tahapan pembuatan robot, yaitu:

1. Perencanaan, meliputi: pemilihan hardware dan design.
2. Pembuatan, meliputi pembuatan mekanik, elektronik, dan program.
3. Uji coba.

2.2.3.1. Perencanaan

Dalam tahap ini, akan ditentukan robot apa yang akan dibuat dan akan digunakan untuk apa. Adapun hal-hal yang perlu ditentukan dalam tahap ini:

1. Dimensi, yaitu panjang, lebar, tinggi, dan perkiraan berat dari robot.
2. Struktur material, apakah dari alumunium, besi, kayu, plastik, dan sebagainya.
3. Cara kerja robot, berisi bagian-bagian robot dan fungsi dari bagian-bagian itu. Misalnya lengan, konveyor, lift, power supply.
4. Sensor-sensor yang akan dipakai robot.
5. Mekanisme, bagaimana sistem mekanik agar robot dapat menyelesaikan tugas.
6. Metode pengontrolan, yaitu bagaimana robot dapat dikontrol dan digerakkan, mikroprosesor yang digunakan, dan blok diagram sistem.
7. Strategi untuk memenangkan pertandingan, jika memang robot itu akan diikuti lomba/kontes robot Indonesia/Internasional.

2.2.3.2. Pembuatan

Ada tiga pekerjaan yang harus dilakukan dalam tahap ini, yaitu pembuatan mekanik, elektronik, dan programming. Masing-masing membutuhkan spesialisasi yang berbeda-beda, yaitu:

1. **Spesialis Mekanik**, bidang ilmu yang cocok adalah teknik mesin dan teknik industri.
2. **Spesialis Elektronika**, bidang ilmu yang cocok adalah teknik elektro.
3. **Spesialis Programming**, bidang ilmu yang cocok adalah teknik informatika.

2.2.3.2.1. Pembuatan mekanik

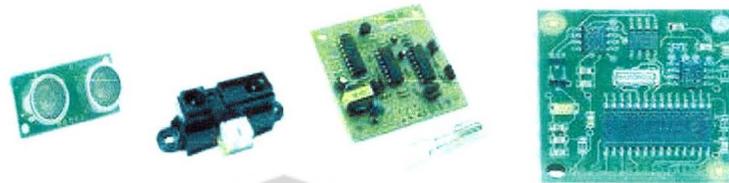
Setelah merancang gambaran garis besar bentuk robot, maka rangka dapat mulai dibuat. Satu ruas rangka terhubung satu sama lain dengan keling aluminium. Keling adalah semacam paku aluminium yang berguna untuk menempelkan lembaran logam dengan erat.

2.2.3.2.2. Pembuatan sistem elektronika

Bagian sistem elektronika dirancang sesuai dengan fungsi yang diinginkan. Misalnya untuk menggerakkan motor DC diperlukan h-bridge, sedangkan untuk menggerakkan relay diperlukan saklar transistor. Sensor-sensor yang akan digunakan dipelajari dan dipahami cara kerjanya, misalnya:

1. Sensor jarak, bisa menggunakan SRF04, GP2D12, atau merakit sendiri modul sensor ultrasonik atau inframerah.
2. Sensor arah, bisa menggunakan sensor kompas CMPS03 atau Dinsmore.
3. Sensor suhu, bisa menggunakan LM35 atau sensor yang lain.

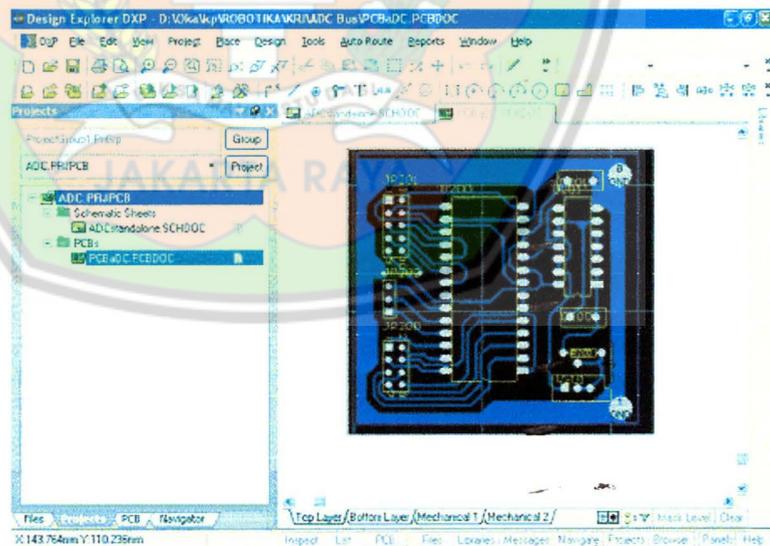
4. Sensor nyala api/panas, bisa menggunakan UVTron atau Thermopile.
5. Sensor line follower / line detector, bias menggunakan led & photo transistor.



Gambar 2.4. Sensor ultrasonik, inframerah, UVTron, dan kompas

Pembuatan sistem elektronika ini meliputi tiga tahap:

1. Design PCB, misalnya dengan program Altium DXP.
2. Pencetakan PCB, bisa dengan Proboard.
3. Perakitan dan pengujian rangkaian elektronika.



Gambar 2.5. Pembuatan Sistem Elektronika

2.2.3.2.3. Pembuatan Software/Program

Pembuatan software dilakukan setelah alat siap untuk diuji. Software ini ditanamkan (didownload) pada mikrokontroler sehingga robot dapat berfungsi sesuai dengan yang diharapkan.

Tahap pembuatan program ini meliputi:

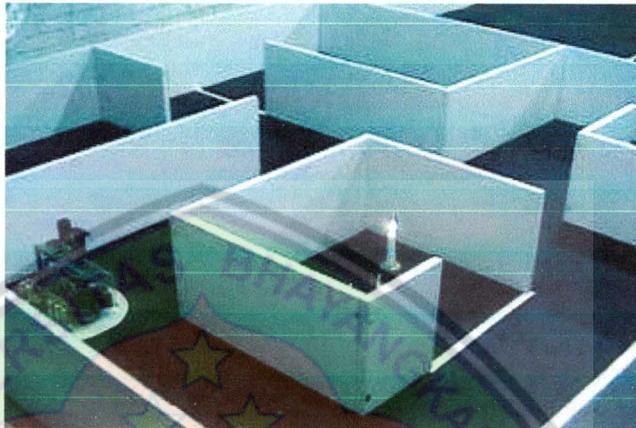
1. Perancangan Algoritma atau alur program. Untuk fungsi yang sederhana, algoritma dapat dibuat langsung pada saat menulis program. Untuk fungsi yang kompleks, algoritma dibuat dengan menggunakan flow chart.
2. Penulisan Program, dapat dilakukan dalam Bahasa C, Assembly, Basic, atau Bahasa yang paling dikuasai.
3. Compile dan download, yaitu mentransfer program yang kita tulis kepada robot.

2.2.3.3. Uji Coba

Setelah mendownload program ke mikrokontroler (otak robot) berarti telah siap dilakukan tahapan terakhir dalam membuat robot, yaitu uji coba. Pada proses ini, dilakukan pengujian terhadap alat yang dibuat, apakah sudah sesuai dengan spesifikasi yang diharapkan. Jika ternyata hasilnya tidak memuaskan, dapat dilakukan perancangan ulang.

Untuk KRCI (Kontes Robot Cerdas Indonesia), ujicoba dilakukan pada arena seluas sekitar 4×4 meter dan berbentuk seperti puzzle. Dalam arena KRCI ini diletakkan lilin-lilin yang harus dipadamkan oleh robot cerdas pemadam api. Untuk lomba robot KRI (Kontes Robot Indonesia),

dibutuhkan ruangan yang lebih besar, yaitu sekitar 15×15 meter. Dalam KRI 2008, masing-masing robot harus meraih target (bola/kubus) yang diletakkan di tempat yang tinggi, jadi sebuah robot harus bisa naik di atas robot yang lain untuk meraih target tersebut (seperti panjat pinang).



Gambar 2.6. Sebuah Robot Dalam Arena Pertandingan

2.2.4. Evolusi Robot Indonesia

Sampai saat ini, belum ada data yang dapat memberikan kepastian mengenai kapan robot mulai dikembangkan di Indonesia. Namun mulai tahun 80-an, kebijakan nasional dalam pengembangan riset teknologi telah memberikan dukungan pada litbang permesinan otomatis dalam rangka mencermati dan menunjang Sumber Daya Manusia Indonesia yang memiliki minat dan kemampuan untuk menguasai teknologi robot. Salah satu wujud konkretnya adalah dikembangkannya sejumlah laboratorium, seperti MEPPPO (Mesin Perkakas Teknik Produksi dan Otomatis) yang diprakarsai oleh BPPT bekerjasama dengan ITB, Industri strategis, serta LET (Laboratorium Elektronika Terapan) di LIPI.

Sejak dikembangkannya sejumlah laboratorium tersebut, beraneka macam permesinan otomatis/robot telah berhasil dikembangkan, diproduksi, serta dikomersilkan oleh berbagai industri, baik industri strategis maupun industri lainnya di Indonesia. Bahkan dalam pengembangan robot terbaru saat ini, telah dikembangkan jenis robot yang memiliki kemampuan untuk mengontrol seluruh sistem operasi suatu pabrik.

Sejak tahun 80an, pendayagunaan dan pemanfaatan permesinan otomatis telah dilakukan terutama melalui sejumlah industri strategis, diantaranya: PT PINDAD (sistem, peralatan, dll.), PT LEN Industri (IT, perangkat lunak, komputasi), PT Bharata dan PTBBI (pengecoran presisi untuk membuat bagian-bagian mesin), dan lain-lain. Disamping itu, PT DI dan PT PAL, yang merupakan pengguna mesin otomatis, telah menguasai pengetahuan mengenai operasionalisasi robot untuk teknologi pesawat terbang dan teknologi perkapalan.

Kontes Robot Indonesia pertama kali diselenggarakan oleh Depdiknas tahun 1990. Sebelas tahun berikutnya, tepatnya pada tahun 2001, salah satu perwakilan dari Indonesia, yaitu tim B-Cak dari PENS-ITS telah berhasil mencapai prestasi yang spektakuler, yakni dengan keluar sebagai Juara Pertama pada Asia Pasific Broadcasting (ABU) Robocon yang diselenggarakan di Tokyo.

Pada tahun 2001 juga, Kementerian Ristek bersama dengan Depdiknas telah mempromosikan juara Kontes Robot Indonesia dalam

pameran Ristek tahunan yaitu RITECH EXPO (Research, Inovation, Technology Expo) yang diselenggarakan di Balai Sidang Jakarta. Dalam pameran tersebut terlihat respon positif dan antusiasme dari masyarakat.

Menjelang Kontes Robot Indonesia 2004, Kementerian Ristek bekerjasama dengan Departemen Pendidikan Nasional - Fakultas Teknik Universitas Indonesia telah menyelenggarakan semiloka (seminar dan lokakarya) dengan tema "Peluang dan Tantangan Teknologi Robot di Indonesia". Semiloka ini diselenggarakan dengan tujuan mempertemukan pihak-pihak yang berkepentingan dalam rangka pengembangan teknologi robot, agar para stakeholders tersebut dapat saling berbagi informasi terbaru dan berbagi pemahaman mengenai isu-isu teknologi robot yang sedang berkembang saat itu. Sasaran yang ingin di capai dengan semiloka ini adalah terdifusinya teknologi robot ke kalangan masyarakat yang lebih luas. Yang menjadi sasaran dalam semiloka tersebut adalah difusi teknologi robot pada kalangan masyarakat yang lebih luas. Dengan diselenggarakannya seminar ini, diharapkan kalangan mahasiswa dapat memperoleh informasi mengenai kebijakan-kebijakan yang telah ditetapkan pemerintah serta kebutuhan industri dalam pemanfaatan dan pendayagunaan robot. Disisi lain, pihak industri bias mendapatkan informasi dan gambaran mengenai pemanfaatan dan pendayagunaan robot untuk keperluan dan kepentingan industry, serta prospek dan kemampuan yang para mahasiswa dalam mengembangkan teknologi robot.

2.3. Visual Basic 6.0 ^[2]

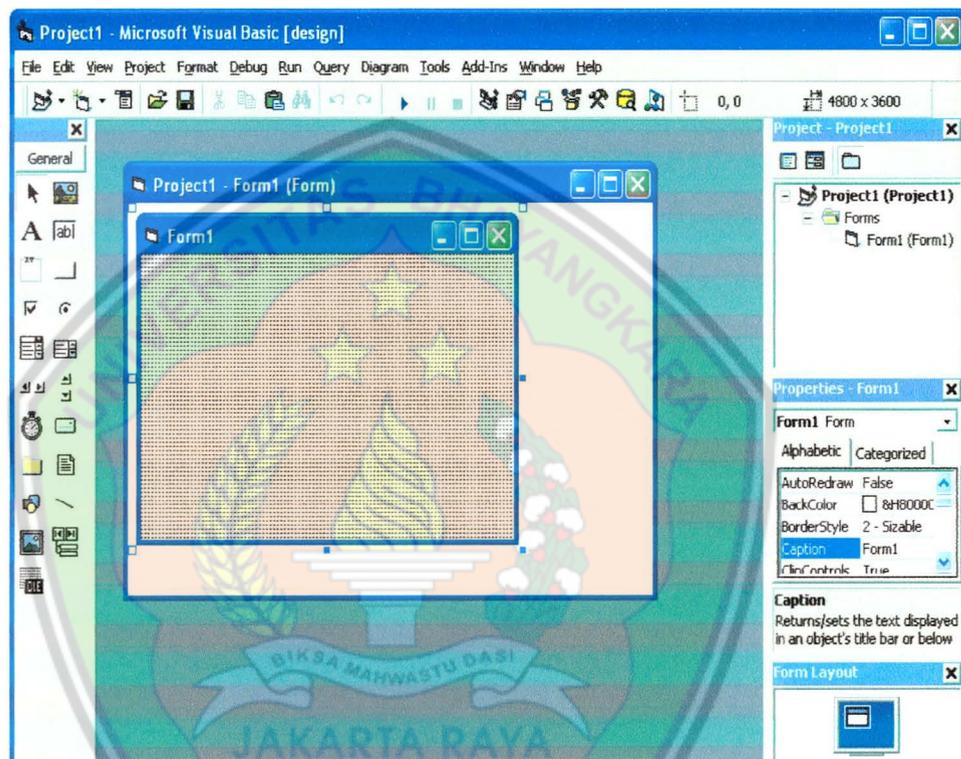
Visual Basic 6.0 adalah salah satu bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Bahasa pemrograman Visual Basic, yang dikembangkan oleh Microsoft sejak tahun 1991, merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (*Beginner's All-purpose Symbolic Instruction Code*) yang dikembangkan pada era 1950-an. Kata -visual- menunjukkan cara yang digunakan untuk membuat Graphical User Interface (GUI). Dengan cara ini, kita tidak perlu menuliskan instruksi pemrograman dalam kode-kode baris, tetapi secara mudah kita cukup melakukan drag dan drop objek-objek yang akan kita gunakan. Visual basic merupakan salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi.

2.3.1. Integrated Development Environment (IDE)

Begitu banyak Kepopuleran Visual Basic datang dari IDE nya, pada dasarnya kita bisa menuliskan program dengan banyak editor, diantaranya Notepad, tetapi kita tidak pernah merasa cukup dengan editor itu saja, kita harus memproses program tersebut dengan fasilitas lainnya, namun dengan IDE ini kita mendapatkan segala sesuatu kemudahan untuk membangun sebuah aplikasi yang baik, menuliskan program untuk aplikasi tersebut, melakukan testing dan pada akhirnya kita dapat menjadikan program tersebut menjadi sebuah program yang berbentuk file executable. Yang mana file ini akan berdiri sendiri tanpa harus

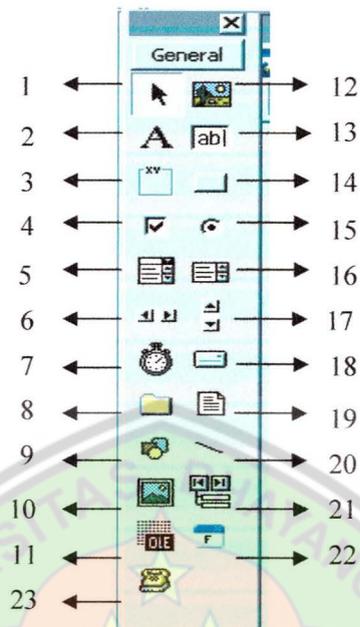
menggunakan IDE tersebut. Dan dapat menjualnya tanpa harus melakukan instalasi Visual basic.

Kita bisa melihat tampilan dari IDE Windows VB seperti gambar dibawah ini, dimana kita bisa melihat banyak fasilitas yang diberikan oleh IDE ini.



Gambar 2.7. Jendela IDE Visual Basic

2.3.2. Kontrol-kontrol Standar Visual Basic



Gambar 2.8. Kontrol-kontrol Visual Basic

Keterangan :

1. Pointer

Bukan merupakan obyek kontrol, digunakan hanya untuk memilih kontrol yang sudah ada di dalam form.

2. Label

Obyek kontrol yang digunakan untuk menampilkan teks yang tetap dan tidak dapat diubah oleh pemakai (*user*).

3. Frame

4. Check Box

Digunakan untuk menyediakan pilihan yang bisa dipilih lebih dari satu hal.

5. Combo Box

6. HScrollbar

Untuk membentuk *scroll* yang berdiri sendiri baik itu vertikal maupun horizontal.

7. Timer

Kontrol waktu ini dapat digunakan untuk menjalankan perintah setiap milisecond, second, menit dan juga setiap jam bergantung dari interval yang diatur di properties. Biasanya kontrol ini digunakan untuk menampilkan jam pada aplikasi ataw juga animasi.

8. Dir List Box

9. Shape

Digunakan untuk membuat bentuk seperti garis, persegi, bulatan, oval, dan lain-lain.

10. Image

Image merupakan obyek kontrol yang fungsinya hampir mirip dengan *picture box* bisa menampilkan dan menampung banyak image di dalam *picture box* itu sendiri.

11. OLE

Berfungsi untuk menjalankan program aplikasi yang lain ke dalam aplikasi yang kita buat (*Litbang, Hal 7, 2006*)

12. Picture Box

Obyek kontrol yang digunakan untuk menampilkan file gambar

13. Text Box

Obyek kontrol yang digunakan untuk menampung kata dengan nilai tipe data pengembaliannya adalah string.

14. Command Button

Obyek kontrol perintah yang berfungsi untuk mengerjakan perintah yang sudah diprogram sebelumnya.

15. Option Button

Obyek kontrol ini fungsinya hampir mirip dengan *check box*, yaitu menyediakan pilihan. Option button dapat juga menyediakan banyak pilihan tapi yang harus kita pilih hanya satu, sedangkan *check box* sebaliknya.

16. List Box

Kontrol ini digunakan untuk menampung sejumlah item dan user dapat memilih dari satu (bergantung pada *Property Multyselect*).

17. VScrollbar

18. DriveListBox

19. File List Box

20. Line

21. Data

Bentuk kontrol dari DAO (Data Access Object) digunakan untuk membuat suatu basis data atau database.

22. Visual Basic Skinner

23. MSComm Kontrol

2.3.3. Tipe Data dan Operator

2.3.3.1. Tipe Data

Dalam setiap bahasa pemrograman, sering kita temui adanya suatu pengelompokan tipe data. Tipe data di Visual Basic telah disediakan untuk ketepatan dalam memakai *resources* (sumber daya) oleh aplikasi yang kita buat sehingga akan dihasilkan program aplikasi yang efisien dan performance tinggi. Tipe-tipe data yang disediakan oleh Visual Basic adalah sebagai berikut :

Tabel 2.1. Tipe Data Visual Basic

Btpe	1 Byte	0 s/d 255
Boolean	2 Byte	True dan False
Integer	2 Byte	-32.768 s/d 32.768
Long	2 Byte	-2.147.783.684 s/d 2.147.783.687
Single	2 Byte	-3,402823 x 1038 s/d 3,402823 x 1038
Double	2 Byte	-1,797769313486232x10308 s/d 1,797769313486232x10308
Currency	2 Byte	-922.337.203.685.447.580 s/d 922.337.203.685.447.579
Date	2 Byte	Mulai tanggal 1 Jan 100 s/d 31 Des 9999
String	2 Byte	0 Hingga 2 Milyar
Variant	2 Byte	Sembarang tipe data (tipe data universal)
User Defined	-	Tergantung tipe yang akan dibuat

Sumber: Prasetya, Hal 25, 2004

Selain tipe data di atas, kita dapat membuat tipe data baru dengan mengambil salah satu atau beberapa tipe data standar yang tersedia. Sarana ini sangat bermanfaat apabila kita menangani

sekelompok data yang menyatu, tetapi berbagi ke dalam beberapa kategori data yang berbeda. Kita dapat membuat tipe data buatan menggunakan pernyataan “type”. Sebagai contoh, deklarasi berikut membuat tipe data buatan yang bernama mahasiswa yang akan menyimpan Nama, NIM, Tanggal lahir dan Alamat.

Type Mahasiswa

Nama as String*25

NIM as String*12

Tanggal_lahir as Date*10

Alamat as String

End Type

Contoh penggunaannya adalah sebagai berikut :

Dim Jimmy as Mahasiswa

Jimmy>Nama = “Jimmy P”

Jimmy.NIM = “060530320230”

Jimmy.Tanggal_lahir = “#9/04/1987#”

Jimmy.Alatmat = “Palembang”

Pada Visual Basic terdapat dua cara untuk mendeklarasikan sebuah variabel, yaitu dengan cara deklarasi eksplisit dan cara deklarasi implisit. Deklarasi eksplisit menggunakan pernyataan “DIM” diikuti nama dan tipe datanya, sedangkan deklarasi implisit menggunakan simbol di belakang nama variabel yang

merepresentasikan tipe data yang digunakan. Berikut ini adalah contoh penggunaan deklarasi eksplisit.

Dim Nama As String

Dim Angka As Integer

Dim Tanggal As Date

Dim Data_input As Variant

Deklarasi implisit digunakan dengan simbol yang merepresentasikan tipe data yang digunakan. Berikut adalah tabel simbol yang digunakan dan contoh pemakaiannya.

Tabel 2.2 Deklarasi Implisit

Type Data	Symbol Karakter	Pemakaian
Integer	%	Angka% = 100
Long Integer	&	Angka& =2147483647
Single	!	Angka! =2147482647000
Double	#	Konstanta Pi# = 3.1415926535
Currency	@	Saldo@ = 1000.50
String	\$	Nama\$ = "Jimmy"

Sumber: Prasetya, Hal 25, 2004

2.3.3.2. Operator-Operator

Operator Aritmetika

Operator aritmetika digunakan untuk operasi matematis terhadap nilai data. Simbol-simbol yang digunakan :

Tabel 2.3. Simbol-simbol Operator Aritmatika

Simbol	Operasi Matematis	Contoh
^	Pemangkatan	5^2 hasilnya 25
*	Perkalian	$5 * 2$ hasilnya 10
/	Pembagian (hasil pecahan)	$5 / 2$ hasilnya 2.5
\	Pembagian (hasil bulat)	$5 \setminus 2$ hasilnya 2
Mod	Sisa pembagian	$5 \text{ Mod } 2$ hasilnya 1
+	Penjumlahan	$5 + 2$ hasilnya 7
-	Pengurangan	$5 - 2$ hasilnya 3
&	Penggabungan string	$5 \& 2$ hasilnya 52

Sumber: Prasetya, Hal 10, 2004

Operator Relational atau Hubungan Perbandingan

Operator perbandingan digunakan untuk operasi yang membandingkan nilai data. Simbol-simbol yang digunakan :

Tabel 2.4. Simbol-simbol Operator Perbandingan

Simbol	Operasi Perbandingan	Contoh
<	Lebih kecil	$5 < 2$ hasilnya FALSE
>	Lebih besar	$5 > 2$ hasilnya TRUE
<=	Lebih kecil atau sama dengan	$5 <= 2$ hasilnya FALSE
>=	Lebih besar sama dengan	$5 >= 2$ hasilnya TRUE
=	Sama dengan	$5 = 2$ hasilnya FALSE
<>	Tidak sama dengan	$5 <> 2$ hasilnya TRUE

Sumber: Prasetya, Hal 10, 2004

Operator Logika atau Boolean

Operator logika digunakan untuk operasi yang membandingkan suatu perbandingan. Simbol-simbol yang digunakan :

Tabel 2.5. Simbol-simbol Operator Logika

Simbol	Operasi Logika	Contoh
Or	Atau	$(5 < 2)$ Or $(5 > 2)$ hasilnya TRUE
And	Dan	$(5 < 2)$ And $(5 > 2)$ hasilnya FALSE
Not	Tidak	$(5 < 2)$ Not $(5 > 2)$ hasilnya TRUE

Sumber: Prasetya, Hal 11, 2004

Variabel dan Deklarasi

Variabel

Salah satu hal yang paling atau sering kita ingat dan didengan apabila kita belajar matematika, nama variabel tidaklah asing kedengerannya. Variabel dalam matematika tidaklah lain, hanya suatu nilai yang hanya digantikan sementara dengan huruf-huruf atas nilai sebenarnya. Dalam pemrograman, ketika kita lihat dan perhatikan sejenak, variabel dalam pemrograman prinsipnya hampirlah sama dengan fungsi variabel yang ada pada matematika. Tetapi perbedaan mencolok antara pemrograman dengan matematika adalah variabel disini diartikan sebagai penampung sementara atas nilai yang telah dimasukkan ke variabel tadi. Dan luas tampungan tergantung dari tipe data yang digunakan atas

variabel tadi. Karena variabel kita itu diartikan data di dalam memori, dan perlu diketahui seberapa luas tampungannya. Untuk menentukan luas penampungannya, maka itu perlu di deklarasikan terlebih dahulu berdasarkan tipe data yang telah dijelaskan di atas. (Prasetya, Hal 13, 2004)

Berikut ini adalah aturan yang harus diperhatikan dalam pemberian nama sebuah variabel atau konstanta.

- ✓ Harus didahului dengan huruf
- ✓ Harus unik, tidak boleh sama, dan nama tersebut tidak digunakan oleh Visual Basic
- ✓ Maksimum 255 karakter
- ✓ Tidak boleh menggunakan karakter-karakter berikut ; “+”, “-”, “/”, “>”, “<”, “,”, “*”, dan lainnya (Prasetya, Hal 27, 2004)

Ruang Deklarasi dan Cara Deklarasi

Untuk mendeklarasikan variabel, kita membutuhkan tipe data sebagai ganti ruang variabel. Deklarasi bisa ditempatkan di general atau prosedur masing-masing “*event*” dari suatu obyek. Dan juga dibutuhkan *scope* (jangkauan) deklarasi. Cara dan formatnya adalah sebagai berikut :

<<Scope>> (nama variabel) as (tipe data)

Keterangan :

- Scope

Merupakan jangkauan berlakunya variabel dalam lingkungan *project*, scopenya ada 4, yaitu :

1. *Static*, dengan menggunakan scope ini isi variabel tidak akan hilang sampai program atau aplikasi itu ditutup. Hanya bisa digunakan untuk lingkungan prosedur.
2. *Dim*, kepanjangannya itu dimension, scope ini bisa kita gunakan untuk menyatakan tingkat penggunaan variabel baik lingkungan prosedur maupun lingkungan general, tapi hanya untuk satu form.
3. *Private*, pernyataan scope ini sangat jarang digunakan untuk tipe data standar, biasanya digunakan untuk deklarasi tipe obyek. Dan berlaku hanya dalam satu form sama seperti dim.
4. *Public*, beda public dari ketiga yang di atas adalah scope ini bisa membuat variabel yang kita gunakan akan dikenal untuk seluruh lingkungan aplikasi atau proyek yang kita buat.

➤ Nama Variabel

Untuk nama, terserah kita, asal jangan dimulai dengan karakter angka atau juga ada karakter operator yang telah di jelaskan di atas.

➤ Tipe data

Data yang akan kita gunakan sebagai format penampung variabel. (Prasetya, Hal 15, 2004)

2.3.4. Struktur Kontrol

Struktur kontrol di dalam bahasa pemrograman adalah perintah dengan bentuk (struktur) tertentu yang digunakan untuk mengatur (mengontrol) jalannya program.

Visual Basic mengenal 2 (dua) jenis struktur kontrol, sebagai berikut ;

1. Struktur Kontrol Keputusan

Digunakan untuk memutuskan kode program mana yang akan dikerjakan berdasarkan suatu kondisi.

Ada dua bentuk struktur kontrol keputusan, yaitu :

a. Struktur **If...Then**

Bentuk penulisannya (*syntax*) struktur **If ... Then** :

✓ If <kondisi> Then <ekspresi>

Bila <kondisi> bernilai "true" maka <kode program> akan dikerjakan

✓ If <kondisi> Then

<ekspresi 1>

Else

<ekspresi 2>

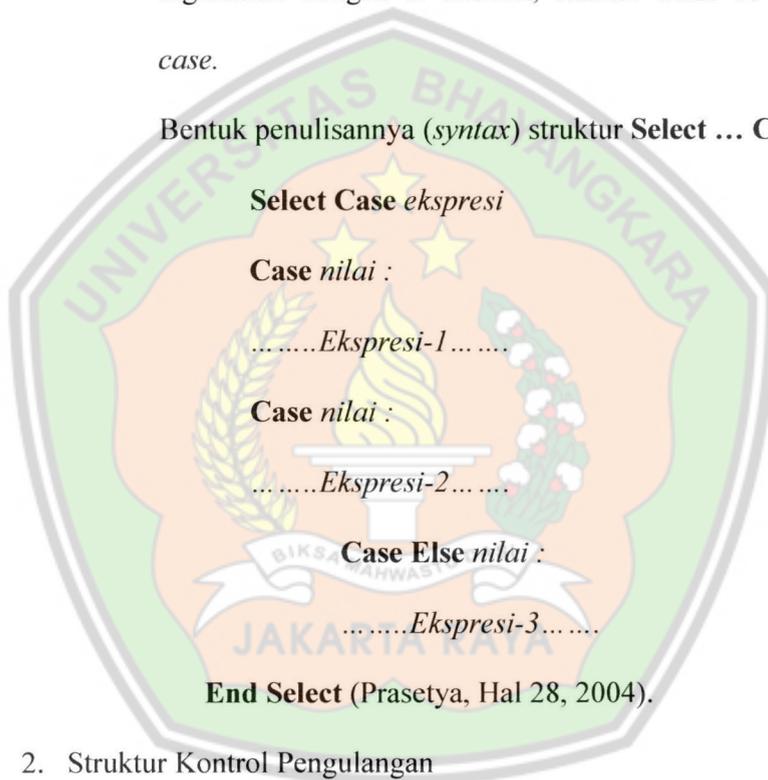
End If

Bila <kondisi> bernilai “true”, maka <ekspresi 1> akan dikerjakan, tetapi bila <kondisi> bernilai “false”, maka <ekspresi 2> yang akan dikerjakan.

b. Struktur **Select ... Case**

Digunakan untuk melakukan pemilihan kondisi yang jumlahnya banyak atau bertingkat tingkat. Walaupun bisa digunakan dengan *if* kondisi, namun tidak se-efektif *select case*.

Bentuk penulisannya (*syntax*) struktur **Select ... Case** :



2. Struktur Kontrol Pengulangan

Digunakan untuk melakukan pengulangan kode program. Ada dua bentuk struktur kontrol pengulangan (*looping*), yaitu :

a. Struktur For...Next

Bentuk penulisannya (*syntax*) struktur **For...Next** :

For <pencacah> = <awal> **to** <akhir> [step<langkah>]

<blok ekspresi>

Next <pencacah>

<pencacah> adalah variable (tipe:integer) yang digunakan untuk menyimpan angka pengulangan.

<awal> adalah nilai awal dari <pencacah>.

<akhir> adalah nilai akhir dari <pencacah>.

<langkah> adalah perubahan nilai <pencacah> setiap pengulangan. Sifatnya *optional* (boleh ditulis ataupun tidak). Bila tidak ditulis maka nilai <langkah> adalah 1

b. Struktur Do...Loop

Bentuk penulisan (*syntax*) struktur Do...Loop :

✓ Do While <kondisi>

<blok ekspresi>

Loop

<blok ekspresi> akan diulang selama <kondisi> bernilai "true". Pengulangan berhenti bila <kondisi> bernilai "false".

✓ Do Until <kondisi>

<blok ekspresi>

Loop

<blok ekspresi> akan diulang sampai <kondisi> bernilai “true”. Pengulangan berhenti bila <kondisi> bernilai “false”. (Prasetya, Hal 28, 2004).

2.3.5. Keuntungan menggunakan Visual Basic

1. Mendukung pembentukan program aplikasi berbasis windows (jendela-jendela) dengan menu yang menarik dan professional.
2. Jendela yang dihasilkan dapat dipindah-pindahkan (Moveable) dan diperbesar atau diperkecil (Resizable).
3. Microsoft Visual Basic secara otomatis dapat mengenal dan memanfaatkan mouse.

2.3.5. Istilah-istilah Visual Basic

1. Event-Driven, artinya setiap aplikasi yang ada digerakan oleh user.
2. GUI (Grafical Unit Interface) adalah pemrograman yang menggunakan tampilan grafik sebagai alat komunikasi dengan pemakainya.
3. DAO (Data Access Object) dukungan untuk memudahkan akses terhadap database lain.
4. OLE (Object Linking dan Embedding) kemudahan menghubungkan beberapa aplikasi dalam Windows -Active X aplikasi yang digunakan oleh pengguna internet.
5. OCX dukungan untuk menggunakan tolls tambahan.

6. DLL (Dynamic Link Library) adalah kumpulan library untuk menggabungkan program dalam microsoft Visual Basic dengan bahasa pemrograman lainnya.
7. DDE (Dynamic Data Exchange) dukungan kemudahan pertukaran dalam membangun program.

2.3.6. Jenis-jenis Aplikasi yang dapat dibuat oleh Visual Basic

Dengan Visual Basic 6 (VB6) kita bisa membangun database dengan mudah dan bisa membangun multi project secara simultan, bahkan membangun aplikasi database client server performa tinggi yang mampu :

1. Mengakses Microsoft SQL-Server.-mendukung Remote Automation dan DCOM.
2. Memakai visual Query dan rancangan koneksi jaringan.
3. Mendukung transact SQL debug serta Remote Data Objek 2.0

2.3.7. Pembuatan dan pengembangan Visual Basic

1. Design : Modus Desain (g) merupakan modus yang digunakan untuk mendesain sebuah aplikasi yaitu merancang tampilan form dan menuliskan rutin program.
2. Procedure : prosedur bisa berupa kelompok kode perhitungan atau pengolahan teks atau operasi database tertentu atau kelompok lainnya.

3. Procedure view dan tombol full modul view memudahkan pilihan untuk keseluruhan penampilan atau hanya procedure terpilih saja.
4. Drag dan drop diantara window code dan window watch.
5. Instant watch dalam mode debug bisa tampil sebagai tooltip.
6. Convert to runtime version : Modus Run-Time (8) merupakan modus saat aplikasi mengambil alih engontrolan program dengan cara Visual Basic menjalankan program yang telah didesain.

2.3.8. Pengaksesan Port Serial Dengan Control MSComm ^[6]

Kontrol MSComm menyediakan fasilitas komunikasi antara program aplikasi dengan *port* serial untuk mengirim dan menerima data melalui *port* serial. Setiap MSComm hanya menangani satu *port* serial sehingga jika ingin menggunakan lebih dari satu *port* serial, harus menggunakan MSComm sebanyak *port* serial yang dipakai.

Jumlah properti pada MSComm sangat banyak sehingga disini tidak akan dibahas secara keseluruhan, cukup disesuaikan dengan kebutuhan saja. Properti-properti yang sering dipakai adalah sebagai berikut:

CommPort : Untuk menentukan nomor *port* serial yang akan dipakai.

Setting : Untuk menset nilai *baudrate*, *parity*, jumlah bit data dan jumlah bit *stop*.

PortOpen : Untuk membuka dan menutup *port* serial yang dihubungkan dengan MSComm ini.

Output : Untuk menuliskan data *string* pada *buffer* kirim.

Berikut contoh properti tersebut untuk mengirim satu karakter dengan MSComm pada *port* Com 1:

```
Private Sub Form_Load ()  
MSComm1.ComPort = 1  
MSComm1.Settings = "9600,N,8,1"  
MSComm1.PortOpen = True  
MSComm1.Output = "w"  
MSComm1.PortOpen = False  
End Sub
```

Kode-kode program di atas akan melakukan aksi sebagai berikut:

1. *Port* serial yang digunakan adalah Com 1.
2. Seting MSComm dengan *baudrate* 9600, tanpa bit paritas, jumlah data 8 bit dan jumlah bit *stop* adalah 1 bit.
3. Membuka *Port* serial Com 1.
4. Mengirim satu karakter ("w").
5. Menutup kembali Com serial yang dipakai.

Event pada MSComm hanya mempunyai satu *event* saja yaitu *event OnComm* saja. *Event OnComm* dibangkitkan jika nilai properti dari *CommEvent* berubah yang mengindikasikan telah terjadi *event* pada *port* serial baik *event* komunikasi maupun *event error*. Berikut nilai-nilai dari

properti *CommEvent*, nilai-nilai properti ini tidak tersedia pada saat *design time* , tetapi hanya dapat dibaca pada saat *run time*.

comEvenFrame : *Hardware* mendeteksi adanya kesalahan *framing*.

comEvenRxParity : *Hardware* mendeteksi adanya kesalahan *parity*.

comEvenRxOver : *Buffer* penerima mengalami *over flow*, tidak ada ruang kosong lagi pada *buffer* penerima.

comEvenTxFull : *Buffer* pengiriman telah penuh.

comEvenOverrun : *Port* mengalami *Overrun*.

comEvenBreak : Sinyal *Break* dikirim.

comEvenDCB : Mendapatkan kembali *Device Control Clock* (DCB) dari *port* serial.

Nilai-nilai properti dari *event* komunikasi pada *CommEvent* sebagai berikut :

comEvSend : Jumlah karakter pada *buffer* kirim lebih sedikit dari pada nilai properti *Sthreshold*. *Event* ini akan dibangkitkan jika nilai properti *Sthreshold* tidak diisi "0"

comEvReceive : Telah diterima karakter sebanyak nilai properti *Rthreshold*. *Event* ini akan dibangkitkan terus menerus sampai data diambil dari *buffer* penerima menggunakan perintah *Input*. *Event* ini akan dibangkitkan jika nilai pada properti *Rthreshold* tidak diisi "0".

commEvCTS : Terjadi perubahan pada saluran *Clear to Send*.

comEvDSR : Terjadi perubahan pada saluran *Data Set Ready*.

comEvCD : Terjadi perubahan pada saluran *Carrier Detect*.

comEvRing : Terdeteksi adanya sinyal *Ring*.

comEvEOF : Karakter *End of File* diterima.



BAB III

SISTEM BERJALAN

3.1. Sistem yang Berjalan

Praktikum adalah subsistem dari perkuliahan yang merupakan kegiatan terstruktur dan terjadwal yang memberikan kesempatan bagi mahasiswa untuk mendapatkan pengalaman yang nyata dalam rangka meningkatkan pemahaman mahasiswa tentang teori atau agar mahasiswa menguasai keterampilan tertentu yang berkaitan dengan suatu pengetahuan atau suatu mata kuliah.

Praktikum yang dilakukan diprogram studi Teknik Informatika pada dasarnya sama dengan praktikum diprogram studi yang lainnya. Mahasiswa hanya diberikan simbol-simbol input-output mekanik yang telah terpadu pada processor ‘*OnBoard*’, tanpa mengetahui alur fisik pemrograman robot. Sehingga Mahasiswa dipresentasikan hanya awaal dan akhirnya saja.

Program studi Teknik Informatika menggunakan paket robot ‘*ROBOpro*’ berupa hardware dan software. Setiap Mahasiswa hanya 3 langkah menggunakan paket software yang akan di unggah ke Mainboard Robot berupa ;

a. Langkah Click and Drag

Metode Click and Drag yaitu cara pemrograman robot dengan hanya menggunakan software aplikasi yang telah disediakan dari *ROBOpro* tentunya kompatibel dengan dengan bergabai macam Mainboard robot sesuai penggunaan. Software aplikasi tersebut telah ‘*Graphical User Interface (GUI)*’ yang mengarahkan para Mahasiswa memanfaatkan simbol-simbol yang telah disediakan. Simbol tersebut berupa ‘*Start*’, ‘*End*’, ‘*Motor*’, ‘*Timer*’, ‘*Decision*’,

'Switch' dan 'alphabeth'. Tentunya sangat memudahkan Mahasiswa dalam menyusun langkah atau logika dalam pemrograman robot. Langkah atau Logika tersebut identik suatu flowchart dari proses tertentu.

Metode ini juga digunakan untuk menghubungkan simbol-simbol yang telah dipergunakan di layar area pemrograman sebagai bentuk alur proses. Satu garis hubungan mencerminkan suatu lompatan langkah proses. Satu simbol dapat mengeluarkan 1 (satu) hubungan dan menerima 1 (satu) hubungan. Untuk Simbol yang berfungsi sebagai desisi/keputusan dapat menerima dan mengeluarkan 2 (dua) hubungan.

Ada beberapa simbol yang membutuhkan nilai tertentu, bahkan penjelasan detail yang mencerikan sebagai sub-program. Untuk kebutuhan nilai User hanya Klik 2 (dua) kali, lalu muncul dialog kebutuhan nilai simbol tersebut. Sub-program dapat diwakili dengan symbol yang dibuat oleh User dan membuat detail programnya di layar area 'panel'.

b. Langkah Test

Metode ini adalah metode kedua yang digunakan saat praktikum bila perancangan pemrograman robot telah selesai. Untuk Test inipun telah disediakan oleh Robopro pada software aplikasinya dengan memanfaatkan simbol 'Test'. Sehingga Mahasiswa dapat melihat langsung hasil dari pembuatan pemrograman robot. Test ini tidak menunjukan kesalahan dari pemrograman. Para User yang dapat menilai pemrograman tersebut telah sesuai dengan keinginan atau belum berdasarkan penglihatan hasil pengujian.

c. Langkah Download

Setelah pemrograman dinyatakan selesai dan pengujian telah sesuai dengan yang diinginkan, maka selanjutnya isi program dapat diunggah ke Mainboard Robot melalui 'Downloader' dengan memanfaatkan symbol 'downloader'. Proses inilah sebagai bentuk penyelesaian kegiatan praktikum di Laboratorium Robotika sehingga program ditanamkan di Memori pada Mainboard Robot. Tentunya bila robot digunakan sudah memiliki program tersebut , sehingga dapat cerdas dalam melakukan fungsinya sesuai keinginan User.

3.1.1. Fasilitas yang Tersedia

Adapun fasilitas Laboratorium Robotika yang terdapat di Universitas Bhayangkara Jakarta Raya khususnya Fakultas Teknik adalah sebagai berikut:

a. Mainboard Robot

Berdasarkan data yang penulis peroleh, ada 3 tipe Mainboard Robot Robopro yang digunakan. Tipe tersebut tentunya mencerminkan sebuah fungsi tertentu.. Adapun fungsi tersebut datanya dapat dilihat sebagai berikut:

1. Tipe Main Board Serial No. 00103222, berfungsi untuk *Motor Switching*.
2. Tipe Main Board Serial No. 00014295, berfungsi untuk *Motor Driving*.

b. Personal Computer

Personal Computer (PC) adalah Hardware pendukung dari proses pemrograman dengan pemanfaatan software yang disediakan Robopro. Ada 15 personal computer dalam ruang Larboratorium Robotika. Pada Software Robopro kompatibel dengan Operating Sistem Windows.

3.2. Tempat Penelitian

Adapun tempat penelitian untuk tugas akhir ini penulis mengambil tempat di kampus Universitas Bhayangkara Jakarta Raya yang berada di wilayah Bekasi, tempat penelitian yang penulis lakukan adalah di Fakultas Teknik program studi Teknik Informatika.

Secara umum teknik informatika dibataskan sebagai kumpulan disiplin ilmu dan teknik yang secara khusus menangani masalah transformasi atau pengolahan fakta simbolik, dengan menggunakan komputer. Dalam melaksanakan tranformasi dan mengolah fakta simbolik itu ditempuh proses-proses logika yang pada tingkat paling tinggi. Ciri paling dominan informatika adalah logika dan sistematika.

Secara umum komponen dasar ilmu informatika adalah perangkat keras komputer, perangkat lunak, teori informatika, matematika komputasi dan aplikasi. Aplikasi informatika mencakup hampir semua bidang kehidupan manusia. Hal yang lebih dikemukakan dalam program studi informatika adalah penerapan dalam bisnis dan administrasi, aplikasi dalam industri. Dengan demikian secara garis besar kurikulum memberi bekal kepada para lulusan, agar mampu berperan sebagai pendorong dan pembuat perangkat lunak komputer yang memanfaatkan

metoda yang ada, secara sistematis untuk menyelesaikan masalah dengan bantuan komputer.

3.3. Teknik Pengumpulan Data

Teknik pengumpulan data yang penulis lakukan dalam penelitian tugas akhir ini adalah sebagai berikut:

3.3.1. Metode Observasi

Pengumpulan data melalui observasi langsung dengan pengamatan langsung adalah cara pengambilan data yang menggunakan mata tanpa ada pertolongan alat standard lain untuk keperluan tersebut. Observasi merupakan teknik pengumpulan data dengan cara terjun langsung ke lapangan guna meneliti objek atau masalah yang sedang diteliti serta mengumpulkan dokumen-dokumen pendukung tentang objek yang diamati.

Dalam hal ini, yang penulis lakukan selama observasi di Laboratorium Robotika Universitas Bhayangkara Jakarta Raya adalah melakukan pengamatan tentang sistem pembelajaran yang ada dan mengamati kendala-kendala yang ada dalam pembelajaran saat ini yang menggunakan metode konvensional.

3.3.2. Metode Wawancara

Interview atau wawancara dijelaskan oleh Subana (Riduwan, 2003:29) adalah suatu cara pengumpulan data yang digunakan untuk memperoleh informasi langsung dari sumbernya. Wawancara ini

digunakan bila ingin mengetahui hal-hal dari responden secara lebih mendalam serta jumlah responden sedikit. Ada beberapa faktor yang mempengaruhi arus informasi dalam wawancara, yaitu: pewawancara, responden, pedoman wawancara dan situasi wawancara. Pewawancara adalah petugas pengumpul informasi yang diharapkan dapat menyampaikan pertanyaan dengan jelas dan merangsang responden untuk menjawab semua pertanyaan dan mencatat semua informasi yang dibutuhkan dengan benar. Responden adalah pemberi informasi yang diharapkan dapat menjawab semua pertanyaan dengan jelas dan lengkap. Pedoman wawancara berisi tentang uraian penelitian yang biasanya dituangkan dalam bentuk daftar pertanyaan agar proses wawancara dapat berjalan dengan baik.

Dalam hal ini, penulis bertanya langsung kepada narasumber-narasumber yang terlibat dalam proses kegiatan praktikum robotika di Laboratorium Robotika Universitas Bhayangkara Jakarta Raya, seperti mahasiswa sebagai User/programmer.

3.3.3. Metode Kepustakaan

Studi Pustaka adalah teknik pengumpulan data dengan cara mengumpulkan bahan dan data-data dari berbagai jenis buku dengan pengarang yang berbeda sebagai referensi yang ada kaitannya dengan permasalahan yang sedang diteliti.

Dalam hal ini penulis mencari buku-buku referensi dengan cara pinjam/membaca di perpustakaan ataupun dengan browsing di Internet dan

mencari materi-materi yang akurat dari sumber-sumber yang dapat dipercaya. Tujuan dari studi pustaka ini adalah untuk mengetahui sampai kemana ilmu yang berhubungan dengan penelitian telah berkembang, sehingga situasi yang diperlukan dapat diperoleh. Selain itu dengan studi pustaka penulis juga mengumpulkan data-data yang menunjang untuk penulisan tugas akhir ini seperti mengumpulkan data tentang profil Fakultas Teknik program studi Teknik Informatika, meminta data-data dosen yang masih aktif, sistem penilaian dan data-data lain yang menunjang untuk penelitian tugas akhir ini.

3.4. Analisis Kebutuhan

Sebelum memutuskan untuk melaksanakan pembuatan aplikasi, terlebih dahulu haruslah mengetahui persiapan apa saja yang harus dilakukan, selain itu komponen apa saja yang dibutuhkan sehingga aplikasi yang akan dibuat dapat berjalan dengan baik.

Pada dasarnya terdapat 2 komponen penting yang perlu dipersiapkan yaitu:

a. Perangkat Keras:

- ✓ Mainboard Robot, yaitu DT-51 Low Cost Micro System (AT89s51) Ver. 2.0
- ✓ SPC DC Motor, berikut DC Motornya
- ✓ In System Programming (ISP) downloader
- ✓ Kabel DB-9

✓ Personal Computer (PC) minimum Pentium-3

b. Perangkat Lunak:

✓ ASM 51,

✓ OS. Windows XP

✓ Visual Basic 6.0

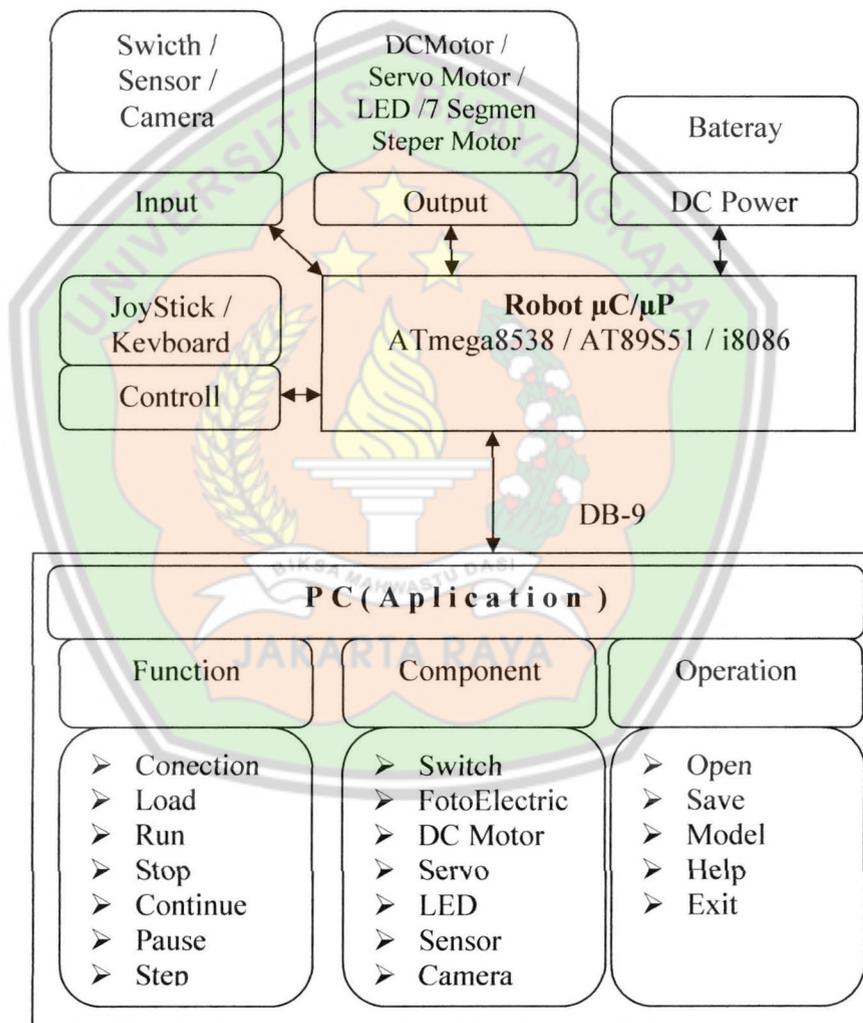
Untuk membuat aplikasi beberapa tahap harus dilalui yaitu:

1. Persiapan: proses persiapan perangkat keras, perangkat lunak, modul praktikum yang dipersiapkan sebelumnya.
2. Perancangan Aplikasi : proses penyusunan modul-modul 'sub-routine', koneksi dan download ke mainboard robot.
3. Uji coba: untuk beberapa saat, aplikasi harus melewati fase uji coba untuk mengetahui dan mengantisipasi segala kemungkinan kesalahan yang ada sebelum dipakai secara menyeluruh.
4. Pemakaian: pengguna secara menyeluruh aplikasi untuk menunjang praktikum robotika.

BAB IV

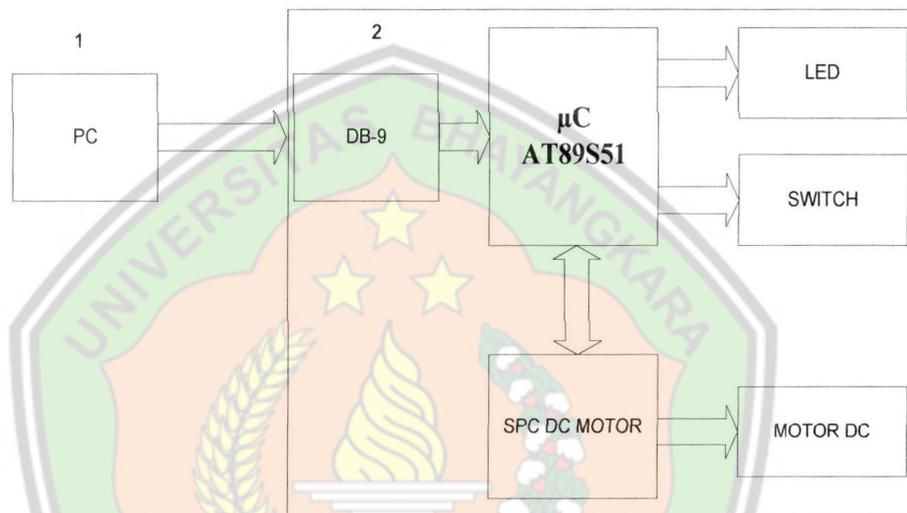
PERANCANGAN PROGRAM APLIKASI DAN PEMBAHASAN

Bab ini akan membahas tentang perancangan sistem yang terdiri dari perancangan perangkat keras dan perancangan perangkat lunak. Secara umum diagram blok dari sistem yang akan dibuat terlihat pada gambar 4.1.



Gambar 4.1. Gambaran Umum Aplikasi dan Robot

Pada Aplikasi ini, penulis merakit beberapa komponen yang sederhana dan dapat terlihat dengan jelas alur maupun jalur dari fungsi-fungsi setiap komponen yang akan dikendalikan dengan Personal Komputer. Processor robot ini menggunakan Mikrocontroll AT89S51. Salah satu bentuk tersebut diberi nama *robot*. Hal tersebut dituangkan dalam diagram di bawah ini ;



Gambar 4.2. Diagram blok robot

Obyek Alat yang dirancang terdiri dari dua bagian utama, yaitu *Personal Computer* (PC), dan *robot*.

3. PC

PC berfungsi untuk mengirimkan data yang digunakan untuk mengendalikan jalannya *robot*. PC ini akan mengontrol *robot* dengan memberikan data ke mikrokontroler melalui dengan kabel DB-9 (komunikasi serial RS232).

4. *Robot* terdiri dari:

a) *DB-9*

Sebagai alat komunikasi serial antara PC dengan *robot*.

b) Mikrokontroler AT89S51

Blok ini merupakan rangkaian sistem minimum dari AT89S51 yang berfungsi untuk mengendalikan *robot*.

c) Switch

Rangkaian ini merupakan rangkaian saklar pengganti sensor yang berfungsi untuk mengendalikan arah putaran motor DC.

d) LED (*Ligth Emitting Diode*)

Rangkaian ini merupakan rangkaian lampu kontrol yang berfungsi untuk mengetahui fungsi switch.

e) SPC DC Motor

Blok ini merupakan rangkaian *driver* motor DC yang berfungsi untuk mengendalikan motor DC.

4.1. Perancangan Perangkat Keras

Pada perancangan perangkat keras ini terdiri dari sebuah PC, DB-9 ke TTL menggunakan DT51 Low Cost Micro System Ver 2.0, SPC DC Motor, Switch, LED, serta DC Motor.

4.1.1. Rangkaian Mikrokontroler AT89S51

Penggunaan AT89S51 ditujukan untuk menerima data serial berupa karakter dari keluaran RS232, kemudian data tersebut diterjemahkan menjadi perintah untuk mengendalikan gerak robot.

Untuk data kendali robot dikirimkan langsung ke port yang tersambung ke SPC DC Motor, begitu juga dengan Switch dan LED data kendalinya langsung dikirimkan ke port yang tersambung ke masing-masing alat. Rangkaian mikrokontroler dapat dilihat pada gambar 4.3.

Pada robot ini akan dirancang kendali pergerakan yang terdiri dari :

a) Gerak Maju

Gerakan maju dilakukan dengan menggerakkan roda kiri dan roda kanan ke arah depan secara bersama-sama. Untuk dapat menggerakkan *robot* dengan arah maju, pada mikrokontroler diberikan instruksi sebagai berikut :

Mov p0,#2dh ; putaran motor arah maju

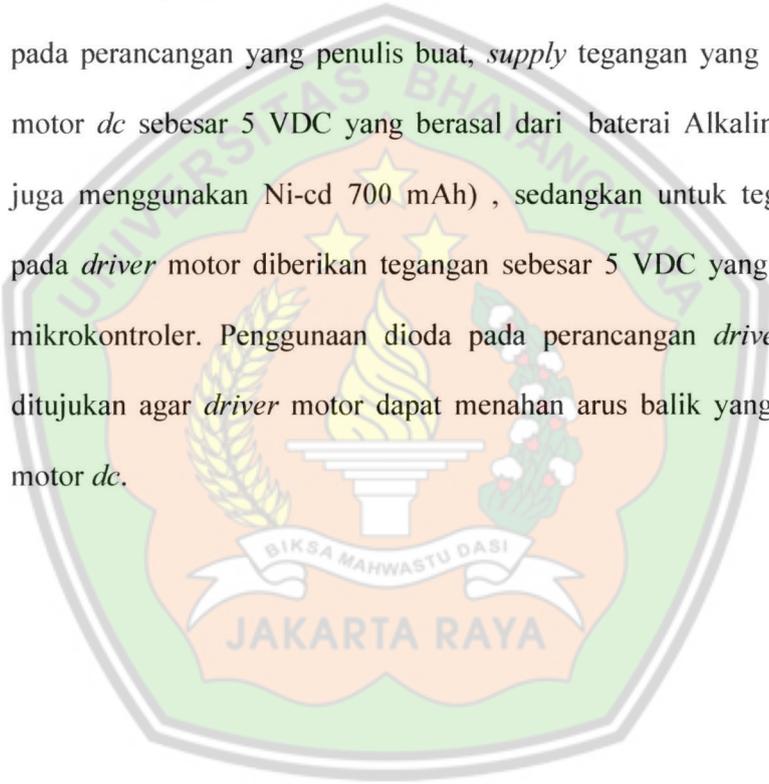
Data 2dh merupakan bilangan heksadesimal jika dikoversikan ke bilangan biner adalah 0010 1101b. Bilangan biner ini merupakan bit-bit yang dikeluarkan dari *port* 0 di mikrokontroler. Bit 1 artinya bit tersebut aktif dan mengeluarkan sinyal sedangkan bit 0 adalah bit tersebut tidak aktif dan tidak mengeluarkan sinyal. Untuk menggerakkan motor diberikan konfigurasi bit-bit yang harus dikeluarkan dari mikrokontroler dan masuk ke *driver* motor, konfigurasi pin-pin *driver* motor dapat dilihat pada tabel 4.1 dan tabel kebenarannya dapat dilihat pada tabel 4.2.

b) Belok Kiri dan Belok Kanan

Gerakan belok kiri dan belok kanan dilakukan dengan menggerakkan roda kiri dan kanan dengan arah yang berlawanan. Jika

Nilai input yang diberikan berasal dari mikrokontroler, sedangkan *port* yang digunakan adalah *port 0*. Pada perancangan pergerakan *robot* ini tidak digunakan PWM, karena sistem pergerakannya manual sehingga kontrol pergerakan secara penuh berada di tangan *user*. Rangkaian *driver* motor *dc* dapat dilihat pada gambar 4.3.

Tegangan yang dapat digunakan untuk mengendalikan *robot* bisa mencapai tegangan 46 VDC dan arus mencapai 2 A untuk setiap kanalnya, pada perancangan yang penulis buat, *supply* tegangan yang diberikan ke motor *dc* sebesar 5 VDC yang berasal dari baterai Alkaline 5V (dapat juga menggunakan Ni-cd 700 mAh) , sedangkan untuk tegangan *logic* pada *driver* motor diberikan tegangan sebesar 5 VDC yang berasal dari mikrokontroler. Penggunaan dioda pada perancangan *driver* motor ini ditujukan agar *driver* motor dapat menahan arus balik yang datang dari motor *dc*.



Tabel 4.1. Konfigurasi pin untuk menjalankan motor penggerak

No Pin	Nama	Keterangan Fungsi
5	Input 1 (A plus)	Untuk menentukan arah putaran motor A
6	Enable A	Untuk menon-aktifkan motor A
7	Input 2 (A min)	Untuk menentukan arah putaran motor A
10	Input 3 (B plus)	Untuk menentukan arah putaran motor B
11	Enable B	Untuk menon-aktifkan motor B
12	Input 4 (B min)	Untuk menentukan arah putaran motor B

Sumber : Innovative Electronic

Tabel 4.2. Tabel Pengaturan perputaran pada motor dc

Enable	IN+	IN-	Keadaan motor
0	X	X	Tidak berputar
1	0	0	Tidak berputar
	0	1	Berputar
	1	0	Berputar ke arah sebaliknya
	1	1	Tidak berputar

Sumber : Innovative Electronic

Tabel 4.3. Tabel kebenaran untuk dua motor

Gerak	A ^{enable}	A ^{plus}	A ^{min}	B ^{enable}	B ^{plus}	B ^{min}
Maju	H	L	H	H	H	L
Belok Kanan 45 derajat	H	L	H	H	L	H
Belok Kanan 90 derajat	H	L	H	H	L	H
Belok kiri 45 derajat	H	H	L	H	H	L
Belok kiri 90 derajat	H	H	L	H	H	L

Sumber : Innovative Electronic

Keterangan :

H=High L=Low

4.1.3. Kabel DB-9

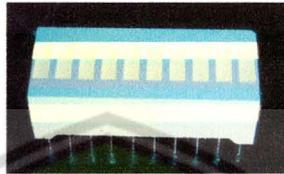
Kabel ini berfungsi untuk menghubungkan antara PC dengan Blok Microprocessor/Microcontroll. Kabel ini berfungsi untuk komunikasi secara serial antara alat dengan komputer.



Gambar 4.4. Kabel DB-9

4.1.4. Rangkaian LED (Ligth Emitting Diode)

Rangkaian LED ini sebagai indikasi berfungsinya switch sehingga dapat mengontrol putaran motor DC.



Gambar 4.5. LED bar

4.1.5. Rangkaian Switch (Saklar)

Sebagai penentu arah jalannya robot dengan mengatur putaran pada motor *dc* penulis menggunakan Switch sebagai pengganti *optocoupler* yang merupakan bagian dari *photo detector* yaitu rangkaian elektronika yang dapat mengubah besaran cahaya menjadi besaran listrik. Rangkaian ini hanya bekerja *on/off* dengan menyentuh tangkai switch tersebut. Sehingga hanya menyalurkan atau memutuskan bit dari microprocessor/mikrocontroll.



Gambar 4.6. Button Switch

Pada saat tungkai switch mengenai/menyentuh obyek benda maka *switch* akan bernilai low '0', tapi jika *switch* dalam keadaan normal dan robot dalam keadaan fungsi maka output bernilai high '1'.

4.2. Perancangan Perangkat Lunak

Setelah dalam pembuatan perangkat keras selesai, bagian yang paling penting dalam pembuatan tugas akhir ini yaitu dalam merancang suatu perangkat lunak. Dalam tugas akhir ini, perancangan perangkat lunak dibagi menjadi 2 bagian yaitu:

1. Perancangan perangkat lunak untuk PC.
2. Perancangan perangkat lunak untuk mikrokontroler.

4.2.1. Perancangan Perangkat Lunak untuk PC

Pada bagian ini menjelaskan tentang perancangan perangkat lunak untuk PC, yang berfungsi untuk mengirim data ke mikrokontroler. Untuk perangkat lunak disini menggunakan program *Visual Basic 6.0* yang memiliki tampilan *visual* dan grafis, sehingga akan lebih menarik untuk dilihat. Untuk program *Visual Basic* ini sebetulnya terbagi menjadi beberapa bagian, tapi disini akan dibahas secara langsung tentang perancangan perangkat lunak yang paling utama, dimana dapat menerima data dan kode dari pengirim.

Berikut ini adalah daftar kode karakter beserta penjelasan fungsinya seperti yang terlihat pada tabel 3.4.

Sintak pengiriman data :

Sintak : [Gerak] [Stop]

Keterangan :

Gerak dapat berupa maju, belok kanan 90°, belok kanan 45°, belok kiri 90°, serta belok kiri 45°.

Tabel 4.4. Kode karakter beserta fungsinya

Kode	Fungsi
w	Maju sejauh 20 cm
r	Belok kanan 90°
t	Belok kanan 45°
l	Belok kiri 90°
m	Belok kiri 45°
d	Mundur Sejauh 20 cm
s	Stop

Sumber : Penulis

Keterangan :

Pada perancangan ini gerak maju dan mundur dari *robot* ditentukan sejauh 20 cm, ini bukan merupakan suatu keharusan bahwa untuk gerak maju hanya dapat dilakukan sampai dengan jarak sejauh 20 cm. Begitu pula yang terjadi pada gerak belok pada *robot*, besar sudut pada gerak belok *robot* dapat diatur dengan cara switch pada *robot* yang tersentuh pada obyek, yaitu dengan mengatur arah putaran motor DC *robot* yang diinginkan, baik salah satu atau keduanya.

Tentunya pada saat perintah Motor DC berjalan dapat di interupsi bila ada switch yang tersentuh. Sehingga dapat mengubah atau mengendalikan *robot* agar tidak menabrak obyek yang dilaluinya.

Contoh format pengiriman data karakter :

(i). w s

Pengiriman perintah maju dan *stop* ke *robot*.

(ii). w r m s

Pengiriman perintah maju, belok kanan 90°, belok kiri 45°, serta *stop* ke *robot*.

(iii). w l r w m t s

Pengiriman perintah maju, belok kiri 90°, belok kanan 90°, maju, belok kiri 45°, belok kanan 45°, serta *stop* ke *robot*.

(iv). dan seterusnya,

Dengan catatan pengiriman perintah ke *robot* tidak terbatas, hanya saja mikrokontroler hanya dapat mengeksekusi maksimal 30 karakter data yang dikirimkan.

Dalam program ini dibuat suatu fasilitas komunikasi antara program aplikasi yang dibuat dengan *port serial*. Untuk mengirim dan menerima data melalui *port serial*. Setiap MScComm hanya menangani satu *port serial*. Untuk penulisan program dapat dilihat pada program dibawah ini:

Comm1.CommPort = 1 ; *port* yang digunakan adalah COM1

Comm1.Settings = "9600,n,8,1" ; *Setting Comm* adalah *baudrate* 9600, tanpa paritas, jumlah data 8 bit dan jumlah bit *stop* adalah 1 bit.

Comm1.PortOpen = True ; membuka *port* serial

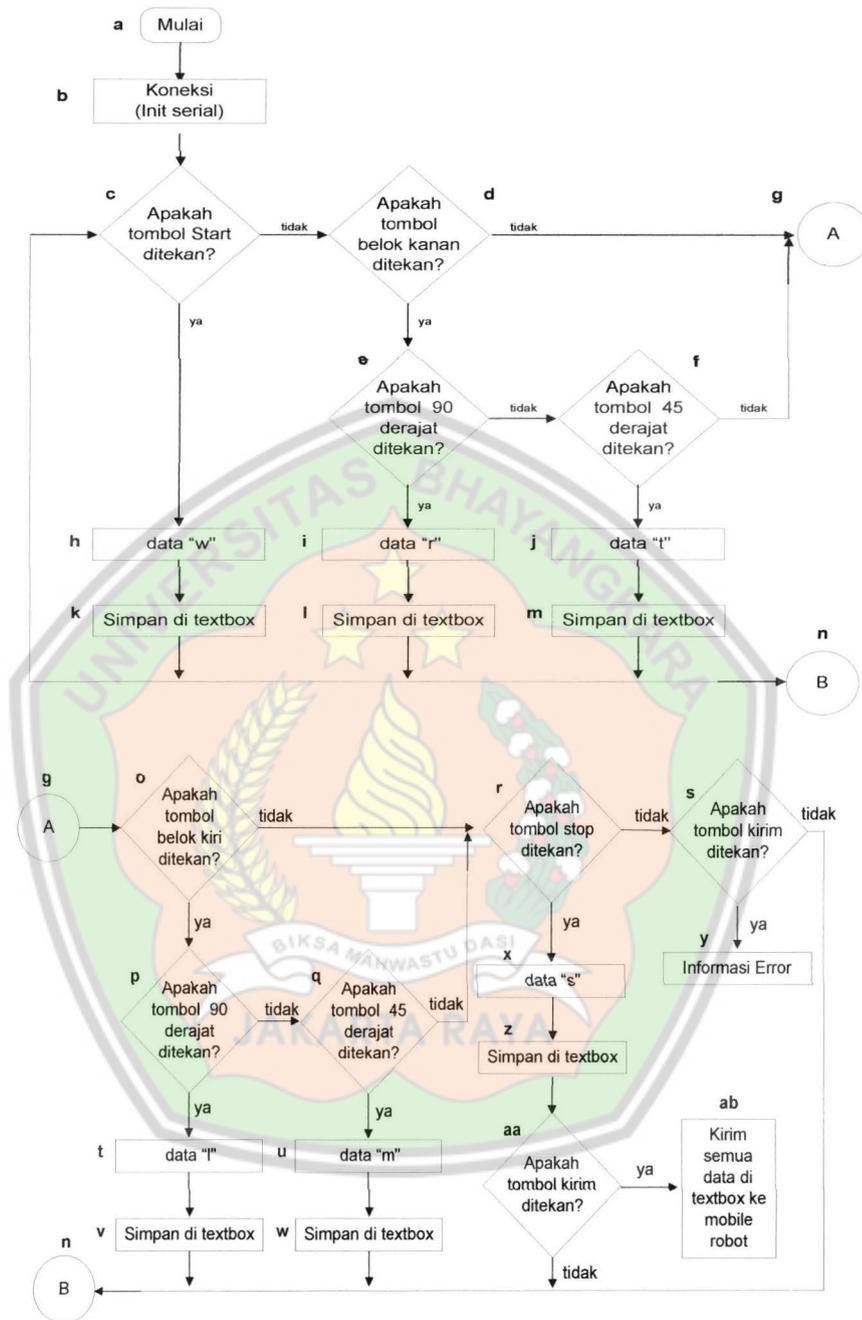
Berikut adalah contoh program di *Visual Basic 6.0* :

```
Private Sub Command1_Click()  
Timer1.Enabled = True  
Comm1.Output = "w" & krm_txt.Text  
End Sub
```

untuk penulisan program secara keseluruhan dapat dilihat pada bagian lampiran.



4.2.1.1. Diagram Alir Program Visual Basic



Gambar 4.7. Diagram alir program visual basic

Berikut ini adalah penjelasan diagram alir program pada visual

basic :

- a. Memulai awal program.
- b. Koneksi (inialisasi serial).
- c. Pengecekan apakah tombol maju ditekan, Jika tombol maju tidak ditekan lakukan pengecekan selanjutnya ke d, jika tombol maju ditekan maka lanjutkan ke h.
- d. Pengecekan apakah tombol kanan ditekan, Jika tombol kanan tidak ditekan lakukan pengecekan selanjutnya ke g, jika tombol kanan ditekan maka lanjutkan pengecekan kondisi ke e.
- e. Pengecekan apakah tombol kanan 90° ditekan, jika ditekan maka lanjutkan ke i, jika tidak maka lanjutkan ke f.
- f. Pengecekan apakah tombol kanan 45° ditekan, jika ditekan maka lanjutkan ke j, jika tidak maka lanjutkan ke g.
- g. Sebagai loncatan diagram alir dari d ke halaman berikutnya.
- h. Kirimkan data “w”, lanjutkan ke k.
- i. Kirimkan data “r”, lanjutkan ke l.
- j. Kirimkan data “t”, lanjutkan ke m.
- k. Data “w” yang dikirimkan di simpan di *textbox*.
- l. Data “r” yang dikirimkan di simpan di *textbox*.
- m. Data “t” yang dikirimkan di simpan di *textbox*.

- n. Sebagai loncatan diagram alir ke halaman berikutnya.
- o. Pengecekan apakah tombol kiri ditekan, jika tombol kiri ditekan maka lanjutkan pengecekan kondisi ke p, Jika tidak lakukan pengecekan selanjutnya ke r.
- p. Pengecekan apakah tombol kiri 90° ditekan, jika ditekan maka lanjutkan ke t, jika tidak maka lanjutkan ke q.
- q. Pengecekan apakah tombol kiri 45° ditekan, jika ditekan maka lanjutkan ke u, jika tidak maka lanjutkan ke r.
- r. Pengecekan apakah tombol *stop* ditekan, Jika tombol *stop* tidak ditekan lakukan pengecekan selanjutnya ke s, jika tombol *stop* ditekan maka lanjutkan ke x.
- s. Pengecekan apakah tombol kirim ditekan, jika tombol kirim ditekan maka lanjutkan ke y, Jika tidak lakukan pengecekan selanjutnya ke n.
- t. Kirimkan data “l”, lanjutkan ke v.
- u. Kirimkan data “m”, lanjutkan ke w.
- v. Data “l” yang dikirimkan di simpan di *textbox*.
- w. Data “m” yang dikirimkan di simpan di *textbox*.
- x. Kirimkan data “s”, lanjutkan ke z.
- y. Menampilkan informasi *error* yang berisi pesan “harus diakhiri dengan penekanan tombol *stop*”.
- z. Data “s” yang dikirimkan di simpan di *textbox*.

- aa. Pengecekan apakah tombol kirim ditekan, Jika tombol kirim tidak ditekan lakukan pengecekan kondisi selanjutnya ke n, jika tombol kirim ditekan maka lanjutkan ke ab.
- ab. Kirimkan semua data yang telah disimpan di *textbox* ke *robot*.

Diagram alir diatas adalah program untuk *visual basic*. Urutan langkah *robot* ditentukan oleh urutan penekanan tombol gerak (maju, belok kanan ada dua pilihan belok yaitu 90° dan 45°, begitu pula dengan belok kiri). Tombol kanan untuk menentukan belok kanan. Tombol kiri untuk menentukan belok kiri. Semua data akan disimpan di *textbox*. Untuk memasukan data ke *robot* melalui penekanan tombol kirim dan urutan langkah di *textbox* akan dimasukkan ke *robot*.

4.2.1.2. Perancangan Menu Utama

Perancangan menu utama berfungsi sebagai tampilan di PC dan untuk memasukkan data-data yang akan dijalankan oleh *robot*. Menu utama meliputi objek-objek seperti *form*, *button* (tombol), *textbox* (kotak isian) dan *label*. Rancangan menu utama dapat dilihat pada gambar 4.8.

Gambar 4.8. Rancangan menu utama

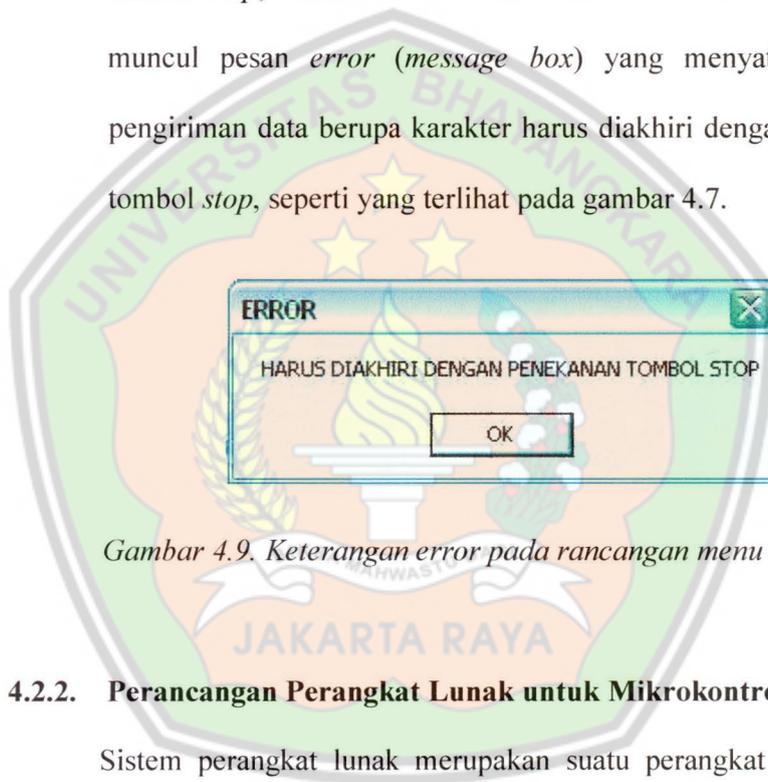
Berikut ini adalah penjelasan rancangan *menu* utama pada gambar 4.8 :

- 1) Merupakan *textbox* yang berisi urutan langkah pengiriman data, data berupa karakter yang akan dikirimkan ditampilkan pada *form* ini.
- 2) Merupakan *command button* untuk belok kiri 45°.
- 3) Merupakan *command button* untuk gerak maju.
- 4) Merupakan *command button* untuk belok kanan 45°.
- 5) Merupakan *command button* untuk belok kiri 90°.
- 6) Merupakan *command button* untuk *stop*.
- 7) Merupakan *command button* untuk belok kanan 90°.
- 8) Merupakan *command button* untuk mengirimkan semua data berupa karakter yang terdapat pada 1).

9) Merupakan *command button* untuk menghapus semua data berupa karakter yang terdapat pada 1).

10) Merupakan *command button* untuk keluar dari *form* utama.

Setelah memasukkan data berupa karakter kedalam *form menu* utama, harus diakhiri dengan penekanan tombol *stop*, jika *user* memasukkan data kemudian tidak diakhiri dengan penekanan tombol *stop*, setelah itu *user* menekan tombol kirim maka akan muncul pesan *error (message box)* yang menyatakan bahwa pengiriman data berupa karakter harus diakhiri dengan penekanan tombol *stop*, seperti yang terlihat pada gambar 4.7.



Gambar 4.9. Keterangan error pada rancangan menu utama

4.2.2. Perancangan Perangkat Lunak untuk Mikrokontroler

Sistem perangkat lunak merupakan suatu perangkat yang dibuat untuk mengendalikan sistem kerja dari masing-masing komponen, sistem ini dikendalikan oleh mikrokontroler yang juga mengendalikan perangkat keras dalam memberikan suatu *input* dan *output* data serta pertukaran informasi.

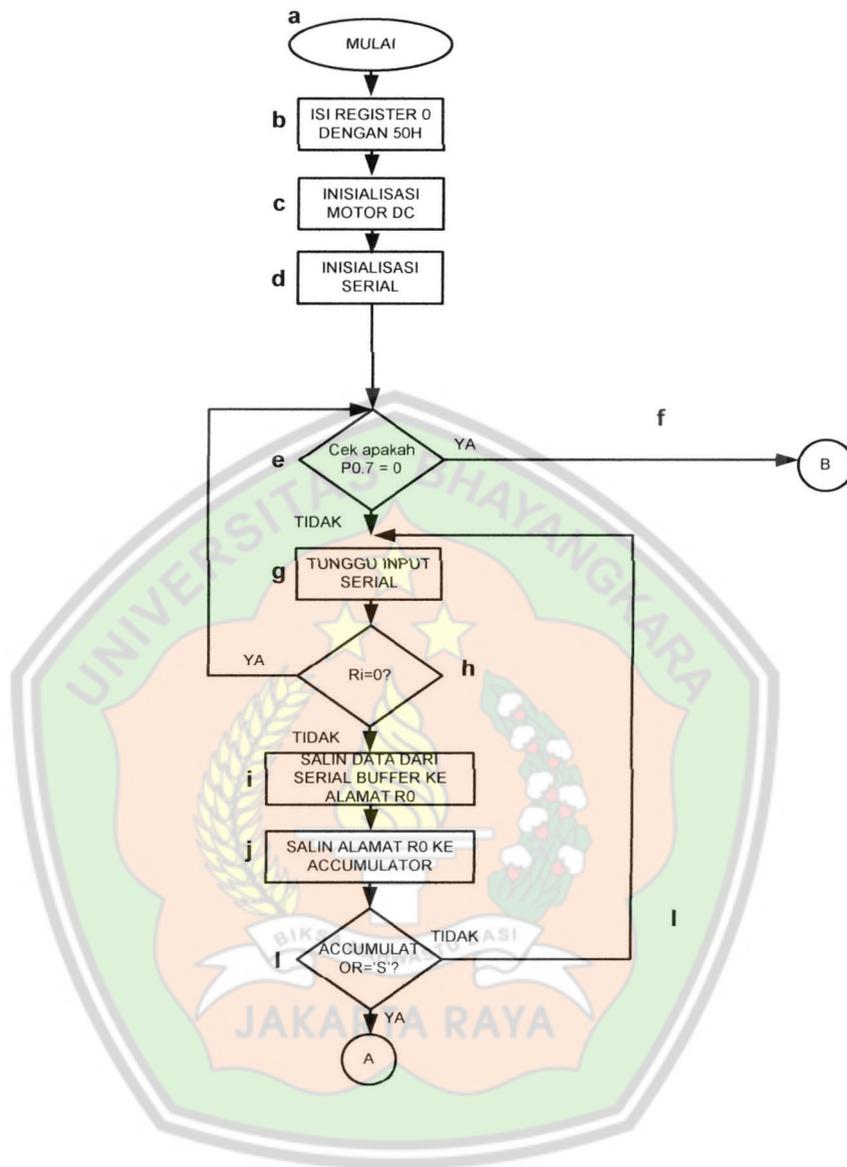
Pada gambar 3.12 adalah tahap perancangan perangkat lunak menggunakan bahasa *assembly* di mana hasilnya disimpan dalam bentuk *file* yang berekstensi *asm* (*.asm). *File* tersebut lalu di-*compile* menggunakan assembler ke dalam bentuk *file* dengan ekstensi *hex* (*.hex). *File* inilah yang nantinya akan dimasukkan ke dalam mikrokontroler menggunakan *downloader*.



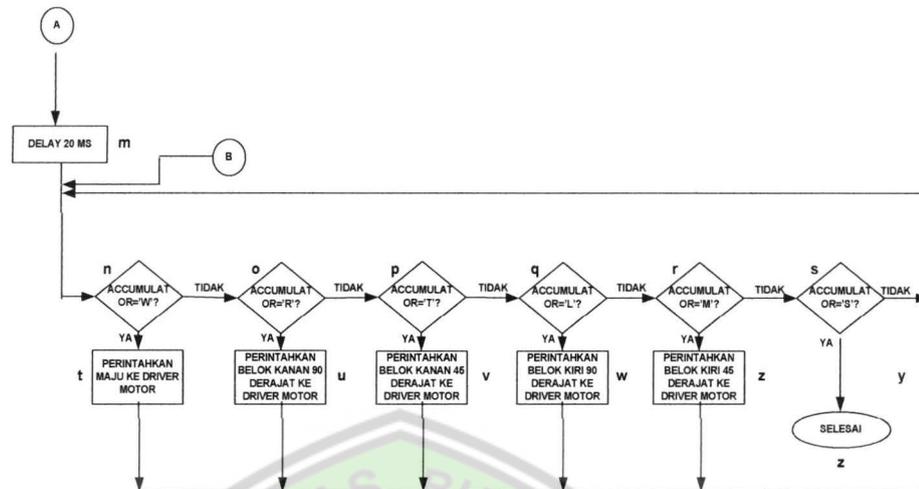
Gambar 4.10. Diagram proses memprogram mikrokontroler



4.2.2.1. Diagram Alir Program Mikrokontroler



Gambar 4.11. Diagram alir program mikrokontroler



Gambar 4.12. Diagram alir program mikrokontroler lanjutan

Berikut ini adalah penjelasan diagram alir program pada mikrokontroler :

- a. Memulai awal program.
- b. Pengisian register 0 (R0) dengan nilai 50H.
- c. Inisialisasi motor dc.
- d. Inisialisasi serial.
- e. Pengecekan apakah tombol *push button* yang dihubungkan ke P0.7 ditekan, jika *push button* ditekan maka lanjutkan ke g, jika tidak maka lanjutkan ke h.
- f. Tunggu apakah ada data serial yang masuk, lanjutkan ke i.
- g. Pengecekan apakah RI (*Receive Interupt Flag*) yaitu bit yang akan *diset* pada akhir penerimaan karakter bernilai 0, jika tidak

maka lanjutkan ke j, jika Ri masih bernilai 0 maka kembali ke f.

- h. Pindahkan data dari *serial buffer* ke alamat yang ditunjukkan oleh R0, yaitu alamat 50H, lanjutkan ke k.
- i. Salin data yang ada di alamat 50H ke *accumulator*.
- j. Pengecekan apakah isi dari *accumulator* = 's' jika ya maka lanjutkan ke o, jika tidak lanjutkan ke m.
- k. Sebagai loncatan diagram alir dari g ke halaman berikutnya.
- l. Sebagai loncatan diagram alir dari l ke halaman berikutnya.
- m. Beri waktu tunda selama 20 ms, lanjutkan ke r.
- n. Pengecekan apakah *accumulator* = 'w', jika ya ke t, jika tidak maka lanjutkan ke o.
- o. Pengecekan apakah *accumulator* = 'r', jika ya ke u, jika tidak maka lanjutkan ke p.
- p. Pengecekan apakah *accumulator* = 't', jika ya ke v, jika tidak maka lanjutkan ke q.
- q. Pengecekan apakah *accumulator* = 'l', jika ya ke w, jika tidak maka lanjutkan ke r.
- r. Pengecekan apakah *accumulator* = 'm', jika ya ke x, jika tidak maka lanjutkan ke s.
- s. Pengecekan apakah *accumulator* = '', jika ya lanjutkan ke y, jika tidak maka lanjutkan ke z.
- t. Memberikan perintah maju ke *driver* motor, lanjutkan ke y.

- u. Memberikan perintah belok kanan 90° ke *driver* motor, lanjutkan ke y.
- v. Memberikan perintah belok kanan 45° ke *driver* motor, lanjutkan ke y.
- w. Memberikan perintah belok kiri 90° ke *driver* motor, lanjutkan ke y.
- x. Memberikan perintah belok kiri 45° ke *driver* motor, lanjutkan ke y.
- y. Kembali ke permintaan perintah
- z. Selesai

Diagram alir diatas adalah program untuk mikrokontroler pada saat menerima data dari komputer dan mengeksekusinya. Untuk inisialisasi awal gerak *robot*, maka diberikan data awal pada program mikrokontroler berupa karakter-karakter seperti huruf 'w' untuk gerak maju, huruf 'r' untuk belok kanan 90° , huruf 't' untuk belok kanan 45° , huruf 'l' untuk belok kiri 90° , huruf 'm' untuk belok kiri 45° dan huruf 's' untuk berhenti. Agar data tersebut *valid* maka pada PC diinisialisasikan data awal yang sama pula. Data yang dikirim dari komputer akan ditampung dan disimpan di register 0 dengan alamat 50h, kemudian dilakukan pengecekan apakah *port* P0.7 ditekan. jika ditekan maka perintahkan mikrokontroler untuk kemudian data tersebut dieksekusi oleh

mikrokontroler yang akan menggerakkan *robot*, sedangkan jika *port* P0.7 tidak ditekan maka mikrokontroler akan menunggu inputan data serial baru dari PC untuk menggerakkan *robot* tersebut.

