

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Pembentukan Karakter**

Setiap karakter yang akan ditampilkan pada *display* adalah tulisan berjalan dari kanan ke kiri. Untuk menampilkan sebuah huruf pada *display*, maka kolom-kolom tempat tampilnya huruf tersebut harus dihidupkan secara bergantian dalam waktu yang cepat, sehingga tampak bahwa sepertinya kolom-kolom tersebut hidup secara bersamaan.

#### **II.2. Rotating Display**

Pembangunan dioda pemancar cahaya berputar (LED) *display*, yang terdiri dari hanya tujuh lampu LED. Prinsipnya adalah bahwa LED ditempatkan pada lengan vertikal berputar, dan dinyalakan dan dimatikan tepat pada saat yang tepat untuk meyakinkan bahwa mata anda melihat tujuh dengan 35 dot matriks dari lampu LED membentuk teks. LED yang dinyalakan dan dimatikan oleh mikrokontroler yang berputar bersama dengan lampu LED. (<http://www.scribd.com/doc/89767769/Rotating-Led-Project-Report>)

#### **II.3. Robotik**

Keunggulan dalam teknologi robotik tak dapat dipungkiri telah lama dijadikan ikon kebanggaan negara-negara maju di dunia. Kecanggihan teknologi yang dimiliki, gedung-gedung tinggi yang mencakar langit, tingkat kesejahteraan

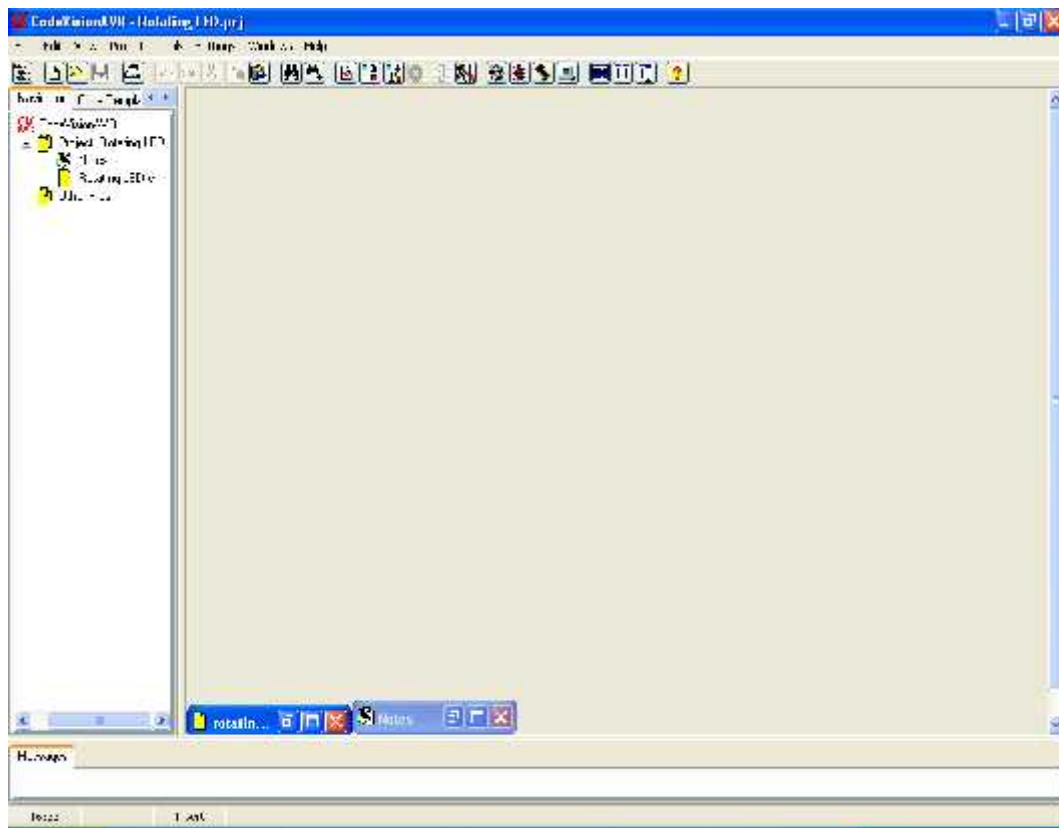
rakyatnya yang tinggi, kota-kotanya yang modern, belumlah terasa lengkap tanpa popularitas kepiawaian dalam dunia robotik.

Kata robot yang berasal dari bahasa Czech, *robota*, yang berarti pekerja, mulai menjadi populer ketika seorang penulis berbangsa Czech(Ceko), Karl Capek, membuat pertunjukan dari lakon komedi yang ditulisnya pada tahun 1921 yang berjudul RUR(Rossum's Universal Robot). Berceritakan tentang mesin yang menyerupai manusia, tapi mampu bekerja terus menerus tanpa lelah. Gaung popularitas istilah robot ini kemudian memperoleh sambutan dengan diperkenalkannya robot Jerman dalam film *Metropolis* tahun 1926 yang sempat dipamerkan dalam *New York World's Fair* 1939. Film ini mengisahkan tentang robot berjalan mirip manusia beserta hewan peliharaannya. Kembali atas jasa insan film, istilah robot ini makin populer dengan lahirnya robot C3PO dalam film *Star Wars* pertama pada tahun 1977.

Menurut Fu, et al. (1987) penelitian dan perkembangan pertama yang berbuah produk robotik dapat dilacak mulai dari tahun 1940-an ketika Argonne National *Laboratories* di Oak Ridge, Amerika, memperkenalkan sebuah mekanisme robotik yang dinamai *master-slave manipulator*. Robot ini digunakan untuk menangani material radioaktif. Kemudian produk robot komersial pertama diperkenalkan oleh Unimation Incorporated, Amerika, pada tahun 1950-an. hingga belasan tahun kemudian langkah komersial ini telah diikuti oleh perusahaan-perusahaan dibelahan dunia lain. (Endra Pitowarno ; 2006 ;1-2)

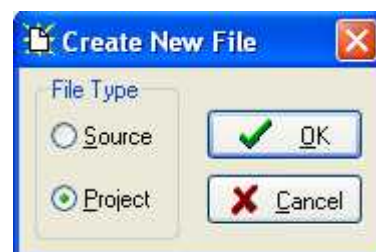
## II.4. Codevision AVR C Compiler

*Codevision AVR C Compiler*, Pemrograman mikrokontroler AVR lebih mudah dilakukan dengan bahasa pemrograman C. Salah satu *software* pemrograman AVR mikrokontroler adalah *CodevisionAVR C Compiler* yang selanjutnya dalam pembahasan disebut *cvAVR*. Pada *cvAVR* terdapat *code wizard* yang sangat membantu dalam proses inialisasi register dalam mikrokontroler dan untuk membentuk fungsi-fungsi interrupt. Pada *code wizard* untuk membuat inialisasi cukup dengan meng klik atau memberi tanda *check* sesuai *property* dari desain yang dikehendaki setelah itu *register* yang ter *inisialisasi* dapat dilihat melalui program *preview* atau melalui *generate and save*. Dengan menggunakan pemrograman bahasa-C diharapkan waktu desain (*deleloping 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*). Gambar Tampilan *CodeVisionAVR* adalah dapat dilihat pada gambar II.1 sebagai berikut :



Gambar II.1 Tampilan CodeVision AVR  
(Sumber : <http://www.musbikhin.com/codevision-avr-c-compiler>)

Untuk memulai bekerja dengan *CodeVisionAVR* pilih pada *menu File -> New*. Maka akan muncul kotak dialog dapat dilihat pada gambar II.2 sebagai berikut :



Gambar II.2. Tampilan project baru  
(Sumber : <http://www.musbikhin.com/codevision-avr-c-compiler>)

Pilih *Project* kemudian tekan *OK*, maka akan muncul kotak dialog dapat dilihat pada gambar II.3 berikut ini.



Gambar II.3. Tampilan code wizard AVR  
(Sumber : <http://www.musbikhin.com/codevision-avr-c-compiler>)

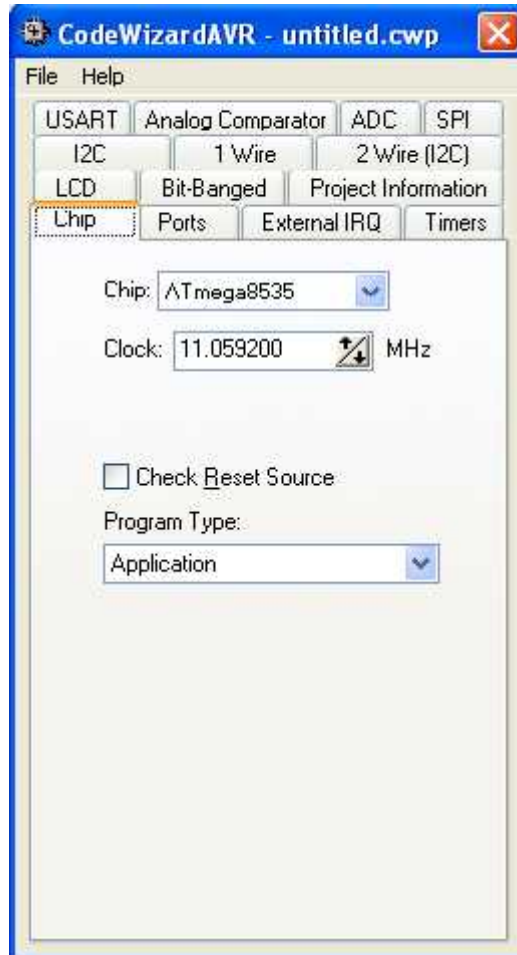
Pilih *Yes* untuk menggunakan *CodeWizardAVR*.

*CodeWizardAVR* digunakan untuk membantu dalam men-*generate* program, terutama dalam *konfigurasi chip* mikrokontroler, baik itu konfigurasi *Port*, *Timer*, penggunaan fasilitas-fasilitas seperti LCD, interrupt, dan sebagainya. *CodeWizardAVR* ini sangat membantu programmer untuk *setting chip* sesuai keinginan.

#### II.4.1. Pemilihan Chip dan Frekuensi Kristal

Langkah pertama dalam menggunakan *cvAVR* adalah membentuk sebuah *project* baru, dengan *click create new project* maka akan muncul pertanyaan apakah anda ingin memanfaatkan bantuan *code wizard*, pilih saja ok maka anda akan masuk pada *code wizard*. Langkah pertama yang harus dilakukan pada *code wizard* adalah memilih jenis chip mikrokontroler yang digunakan dalam *project*

dan *frekwensi xtall* yang digunakan. Pemilihan *chip* dan *frekwensi xtall* dapat dilihat seperti pada gambar II.4 berikut.

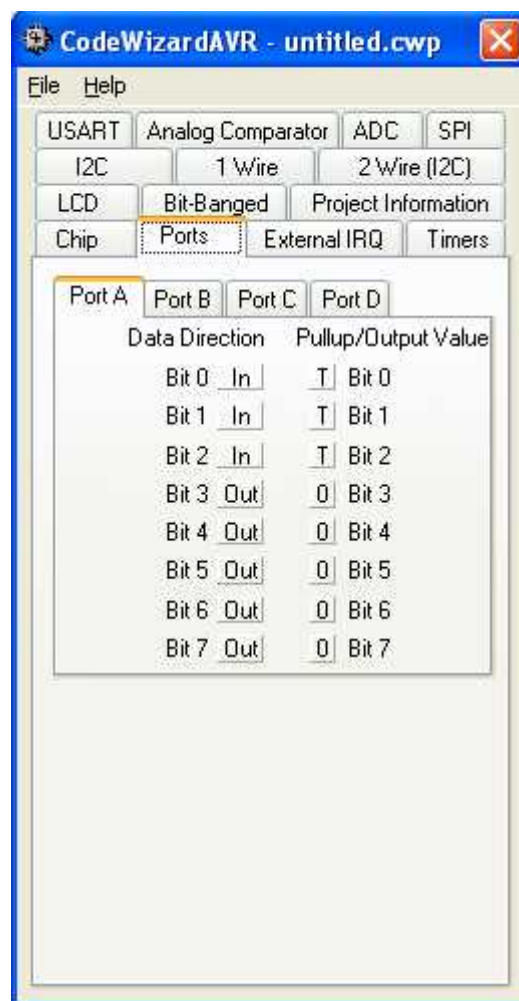


Gambar II.4. Tampilan pemilihan chip dan frekuensi Kristal  
(Sumber : <http://www.musbikhin.com/codevision-avr-c-compiler>)

#### II.4.2 Inisialisasi Port I/O

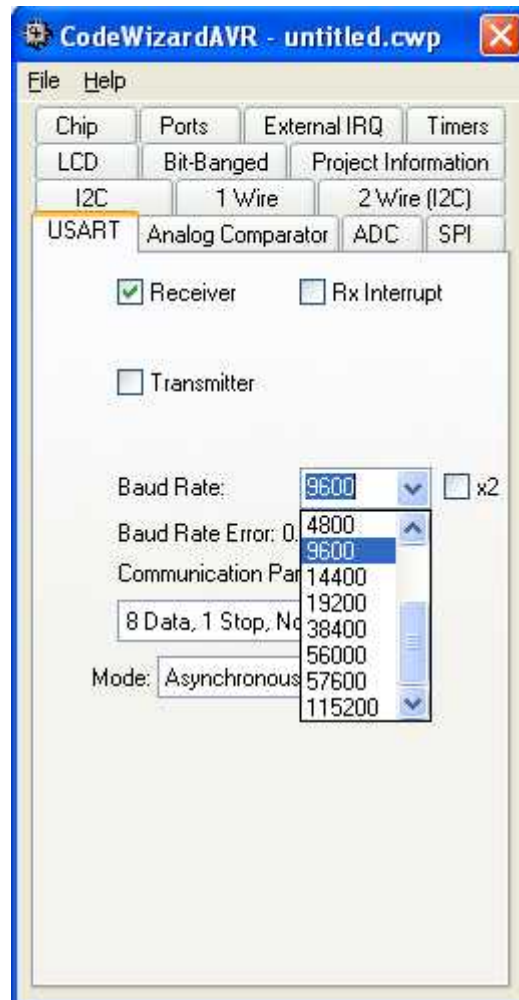
*Inisialisasi Port* berfungsi untuk memilih fungsi *port* sebagai *input* atau sebagai *output*. Pada *konfigurasi port* sebagai *output* dapat dipilih pada saat awal setelah *reset* kondisi *port* berlogika 1 atau 0, sedangkan pada *konfigurasi port* sebagai *input* terdapat dua pilihan yaitu kondisi pin *input toggle state* atau *pull-up*, maka sebaiknya dipilih *pull up* untuk memberi *default* pada *input* selalu berlogika 1.

setiap *port* berjumlah 8 bit, konfigurasi dari *port* dapat diatur sesuai dengan kebutuhan. Pengaturan konfigurasi dapat dilakukan perbit, jadi dalam satu *port* dapat difungsikan sebagai *input* dan *output* dengan nilai *default* nya berbeda-beda. Gambar II.5. menunjukkan *setting* konfigurasi pada *port* a dengan kombinasi *input* dan *output* yang berbeda-beda *default*nya.



Gambar II.5. Tampilan pemilihan port I/O  
(Sumber : <http://www.musbikhin.com/codevision-avr-c-compiler>)

### II.4.3 Inisialisasi Serial untuk mode RX interrupt



Gambar II.6. Tampilan inisialisasi serial

(Sumber : <http://www.musbikhin.com/codevision-avr-c-compiler>)

Untuk selanjutnya fasilitas-fasilitas lainnya dapat *disetting* sesuai kebutuhan dari pemrograman. Setelah selesai dengan *CodeWizardAVR*, selanjutnya pada menu *File*, pilih *Generate, Save and Exit* dan simpan pada direktori yang diinginkan.

## **II.5. Aktuator**

Aktuator adalah bagian yang berfungsi sebagai penggerak dari perintah yang diberikan oleh input. Aktuator biasanya merupakan peranti elektro mekanik yang menghasilkan daya gerakan.

### **II.5.1. Motor DC**

Motor arus searah (DC) adalah suatu mesin yang berfungsi untuk mengubah tenaga listrik arus searah menjadi gerak atau energi mekanik. Konstruksi dasar motor DC terdiri dari 2 bagian utama , yaitu rotor dan stator. Rotor adalah bagian yang berputar atau *armature*, berupa koil dimana arus listrik dapat mengalir. Stator adalah bagian yang tetap dan menghasilkan medan magnet dari koilnya.

### **II.5.2. Prinsip Kerja Motor DC**

Prinsip kerja motor DC adalah jika ada kumparan dilalui arus maka pada kedua sisi kumparan akan bekerja gaya Lorentz. Aturan tangan kiri dapat digunakan untuk menentukan arah gaya Lorentz, di mana gaya jatuh pada telapak tangan, jari-jari yang direntangkan menunjukkan arah arus, ibu jari yang direntangkan menunjukkan arah gaya. Kedua gaya yang timbul merupakan sebuah kopel. Kopel yang dibangkitkan pada kumparan sangat tidak teratur karena kopel itu berayun antara nilai maksimum dan nol. Kumparan-kumparan tersebut dihubungkan dengan lamel tersendiri pada komulatur sehingga motor arus searah tidak berbeda dengan generator arus searah.

## II.6. Mikrokontroler

Pemrograman mikrokontroller merupakan dasar dari prinsip pengontrolan kerja robot, di mana orientasi dari penerapan mikrokontroler adalah untuk mengendalikan suatu sistem berdasarkan informasi input yang diterima, yang kemudian diproses oleh mikrokontroler, dan dilakukan aksi pada bagian output sesuai program yang telah ditentukan sebelumnya.

Mikrokontroler merupakan pengontrol utama perangkat elektronika saat ini, termasuk robot tentunya. Mikrokontroler yang terkenal dan mudah didapatkan di Indonesia saat ini ialah 89S51, AVR Atmega 8535, Atmega16, Atmega32, dan Atmega128. Beberapa merek lain yang terkenal, misalnya PIC 16F877 dan *Basic Stamp 2*.

Mikrokontroler memiliki beberapa port yang digunakan sebagai I/O (*input/output*). Gambar ini merupakan susunan kaki standar 40 pin DIP mikrokontroler AVR Atmega 8535. Dapat dilihat pada gambar II.7 berikut ini. (Widodo Budiharto ; 2010 ;77-78)



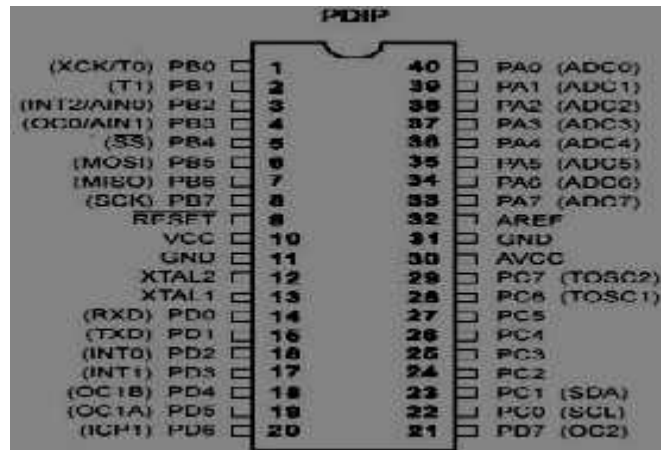
Gambar II.7. Mikrokontroler Atmega8535  
(Sumber : <http://jogjarobotika.com>)

### II.6.1. Konfigurasi Pin ATmega8535

Secara umum konfigurasi dan fungsi pin ATmega8535 dapat dijelaskan sebagai berikut

1. **VCC** *Input* sumber tegangan (+)
2. **GND** *Ground* (-)
3. **Port A (PA7 ... PA0)** Berfungsi sebagai *input analog* dari ADC (*Analog to Digital Converter*). *Port* ini juga berfungsi sebagai *port I/O* dua arah, jika ADC tidak digunakan.
4. **Port B (PB7 ... PB0)** Berfungsi sebagai *port I/O* dua arah. Port PB5, PB6 dan PB7 juga berfungsi sebagai MOSI, MISO dan SCK yang dipergunakan pada proses *downloading*. Fungsi lain *port* ini selengkapnya bisa dibaca pada buku petunjuk "AVR ATmega8535".
5. **Port C (PC7 ... PC0)** Berfungsi sebagai *port I/O* dua arah. Fungsi lain *port* ini selengkapnya bisa dibaca pada buku petunjuk "AVR ATmega8535".
6. **Port D (PD7 ... PD0)** Berfungsi sebagai *port I/O* dua arah. *Port* PD0 dan PD1 juga berfungsi sebagai RXD dan TXD, yang dipergunakan untuk komunikasi serial. Fungsi lain *port* ini selengkapnya bisa dibaca pada buku petunjuk "AVR ATmega8535".
7. **RESET** *Input reset*.
8. **XTAL1** *Input* ke *amplifier inverting osilator* dan *input* ke *sirkuit clock internal*.
9. **XTAL2** *Output* dari *amplifier inverting osilator*.
10. **AVCC** *Input* tegangan untuk *Port A* dan ADC.

## 11. AREF Tegangan referensi untuk ADC.



Gambar II.8 Konfigurasi Pin ATmega8535  
(Sumber : <http://digilib.itelkom.ac.id> )

## II.6.2. Arsitektur Mikrokontroler

Untuk mengoptimalkan performa dan paralelisme, mikrokontroler modern umumnya menggunakan arsitektur Harvard (bus untuk memori dan program dan data terpisah). Jika dilihat lebih detail, pada bagian pemroses mikrokontroler ini terdapat unit CPU utama untuk memastikan eksekusi program. CPU juga dapat mengakses memori, melakukan kalkulasi, pengontrolan, dan penanganan intrupsi, sehingga dapat dihasilkan performa yang tinggi. Hal ini karena instruksi pada memori program dieksekusi dengan *single level pipelining*, sehingga pada saat sebuah instruksi dieksekusi maka instruksi berikutnya dapat diakses dari memori program. Konsep ini memungkinkan instruksi-instruksi dieksekusi pada setiap siklus *clock*. (Widodo Budiharto ; 2010 ;79-80)

*Fast-access Register File* berisi 32x8 bit register keperluan umum dengan waktu akses hanya membutuhkan sebuah siklus detak. Teknik ini membuat operasi ALU (*Arithmetic Logic Unit*) hanya membutuhkan 1 siklus. Lebih dahsyat

lagi, 6 dari 32 *register* dapat digunakan sebagai 3buah 16 bit *indirect address register pointer* untuk pengalamatan ruang data. Satu *address pointer* dapat digunakan sebagai *address pointer* untuk *look up table* pada memori flash program. Setiap alamat memori program berisi instruksi sebesar 16 atau 32 bit. (Widodo Budiharto ; 2010 ;80)

## **II.7. Rangkaian Sistem Minimum AVR 8535**

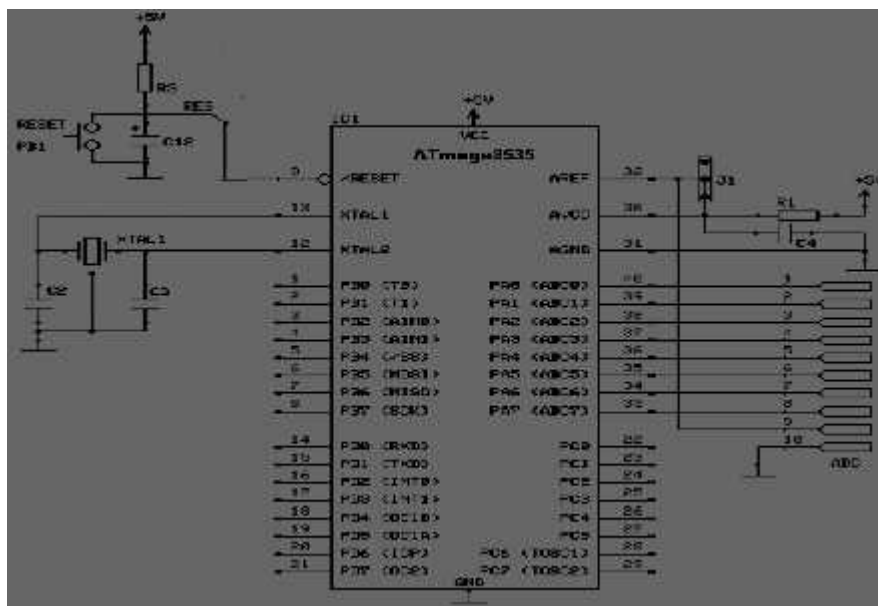
Sistem minimum (sismin) mikrokontroler adalah rangkaian elektronik minimum yang diperlukan untuk beroperasinya IC mikrokontroler. Sismin ini kemudian bisa dihubungkan dengan rangkaian lain untuk menjalankan fungsi tertentu. Di keluarga mikrokontroler AVR, seri 8535 adalah salah satu seri yang sangat banyak digunakan.

Untuk membuat rangkaian sismin Atmel AVR 8535 diperlukan beberapa komponen yaitu:

1. IC mikrokontroler ATmega8535
2. 1 XTAL 4 MHz atau 8 MHz (XTAL1)
3. 3 kapasitor kertas yaitu dua 22 pF (C2 dan C3) serta 100 nF (C4)
4. 1 kapasitor elektrolit 4.7 uF (C12) 2 resistor yaitu 100 ohm (R1) dan 10 Kohm (R3)
5. 1 tombol reset pushbutton (PB1)

Selain itu tentunya diperlukan power suply yang bisa memberikan tegangan 5V DC.

Rangkaian sistem minimum ini sudah siap untuk menerima sinyal analog (fasilitas ADC) di *port* A. Rangkaian dapat dilihat pada gambar II.9 berikut ini. (<http://www.mikron123.com/index.php/Tutorial-AVR/Rangkaian-Sistem-Minimum-AVR-8535.html>)



Gambar II.9. Sistem Minimum AVR 8535

(Sumber : <http://www.mikron123.com/index.php/Tutorial-AVR/Rangkaian-Sistem-Minimum-AVR-8535.html>)