

BAB II

TINJAUAN PUSTAKA

Untuk lebih memahami sistem kerja pada Laporan Akhir yang dibuat oleh penulis, maka terlebih dahulu harus dapat kita mengetahui teori - teori dasar dari rangkaian atau pun komponen - komponen yang berhubungan dengan alat yang di buat. Pada tinjauan pustaka ini maka penulis akan membahas komponen ataupun program yang digunakan pada alat yang dibuat oleh penulis.

2.1 Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer kecil yang lengkap (*“special purpose computers”*) di dalam satu IC yang berisi CPU, memori, timer, saluran komunikasi serial dan paralel, Port *input/output*, ADC, mikrokontroler digunakan untuk suatu tugas dan menjalankan suatu program. Mikrokontroler lebih dari sekedar sebuah mikroprosesor karena sudah terdapat atau berisikan ROM (*Read-Only Memory*), RAM (*Read-Write Memory*), beberapa bandar masukan maupun keluaran, dan beberapa *peripheral* seperti pencacah/pewaktu, ADC (*Analog to Digital converter*), DAC (*Digital to Analog converter*) dan serial komunikasi. (*Heri Andrianto, 2012*)

Pada saat ini penggunaan mikrokontroler dapat kita temui pada berbagai peralatan, misalnya peralatan yang terdapat dirumah seperti telepon digital, *microwave oven*, televisi, mesin cuci, sistem keamanan rumah, PDA, dll. Keuntungan menggunakan mikrokontroler yaitu harganya murah, dapat diprogram berulang – ulang kali, dan juga dapat kita program sesuai dengan keinginan kita. (*Heri Andrianto, 2012*)

Adapun kelebihan dari mikrokontroler adalah sebagai berikut :

- a. Penggerak pada mikrokontroler menggunakan bahasa pemrograman *assembly* dengan berpatokan pada kaidah digital dasar sehingga pengoperasian sistem menjadi sangat mudah dikerjakan sesuai dengan logika sistem (bahasa *assembly* ini mudah dimengerti karena

menggunakan bahasa *assembly* aplikasi dimana parameter input dan output langsung bisa diakses tanpa menggunakan banyak perintah).

- b. Desain bahasa *assembly* ini tidak menggunakan begitu banyak syarat penulisan bahasa pemrograman seperti huruf besar dan huruf kecil untuk bahasa *assembly* tetap diwajibkan. Mikrokontroler tersusun dalam satu *chip* dimana prosesor, memori, dan I/O terintegrasi menjadi satu kesatuan kontrol sistem sehingga mikrokontroler dapat dikatakan sebagai komputer mini yang dapat bekerja secara inovatif sesuai dengan kebutuhan sistem.
- c. Sistem *running microcontroller* berdiri sendiri tanpa tergantung dengan komputer sedangkan parameter komputer hanya digunakan untuk download perintah instruksi atau program. Langkah-langkah untuk download komputer dengan mikrokontroler sangat mudah digunakan karena tidak menggunakan banyak perintah. Pada mikrokontroler tersedia fasilitas tambahan untuk pengembangan memori dan I/O yang disesuaikan dengan kebutuhan sistem.
- d. Harga *microcontroller* lebih murah dan mudah didapat. (Syahrul, 2012)

2.2 Mikrokontroler AVR ATmega 16

AVR adalah mikrokontroler RISC (*Reduce Instruction Set Compute*) 8 bit berdasarkan arsitektur Harvard, yang dibuat oleh Atmel pada tahun 1996. AVR mempunyai kepanjangan *Advanced Versatile RISC* atau *Alf and Vegard's Risc processor* yang berasal dari nama dua mahasiswa *Norwegian Institute of Technology* (NTH), yaitu Alf – Egil Bogen and Vegard Wollan.

AVR memiliki keunggulan dibandingkan dengan mikrokontroler lain, keunggulan mikrokontroler AVR yaitu AVR memiliki kecepatan eksekusi program yang lebih cepat karena sebagaimana besar instruksi di eksekusi dalam 1 siklus *clock*, lebih cepat dibandingkan dengan mikrokontroler MCS₅₁ yang memiliki arsitektur CISC (*Complex Instruction Set Computer*) dimana mikrokontroler MCS₅₁ membutuhkan 12 siklus *clock* untuk mengeksekusi 1 instruksi.

Selain itu, mikrokontroler AVR memiliki fitur yang lengkap (ADC internal, EEPROM Internal, *Timer/Counter*, *Watchdog Timer*, PWM, Port I/O, komunikasi serial, komparator, I₂C, dll), sehingga dengan fasilitas yang lengkap ini, programmer dan desainer dapat menggunakannya untuk berbagai aplikasi sistem elektronika seperti robot, otomasi industri, peralatan telekomunikasi, dan berbagai keperluan lain. Pemrograman mikrokontroler AVR dapat menggunakan *low level language (assembly)* dan *high level language (C, Basic, Pascal, JAVA, dll)*. Tergantung *compiler* yang digunakan.

Bahasa *Assembler* mikrokontroler AVR memiliki kesamaan instruksi, sehingga jika pemrograman satu jenis mikrokontroler AVR sudah dikuasai. Maka akan dengan mudah menguasai pemrograman keseluruhan mikrokontroler jenis AVR, namun bahasa *assembly* relatif lebih sulit dipelajari dari pada bahasa C, untuk pembuatan suatu proyek yang besar akan memakan waktu yang lama, serta penulisan pemrogramannya akan panjang.

Sedangkan bahasa C memiliki keunggulan dibandingkan bahasa *assembly* yaitu independent terhadap *hardware* serta lebih mudah untuk menangani *project* yang besar. Bahasa C memiliki keuntungan - keuntungan yang yang dipunyai oleh bahasa mesin (*assembly*), hampir semua operasi yang dapat dilakukan oleh bahasa mesin, dapat dilakukan oleh bahasa C dengan penyusunan program yang lebih sederhana dan mudah. Bahasa C sendiri sebenarnya terletak di antara pemrograman tingkat tinggi dan *assembly*. (*Agfianto Eko Putra, 2010*)

Mikrokontroler ATmega 16 adalah suatu arsitektur Harvard yang memisahkan memori program dari memori data, baik bus alamat maupun bus data, sehingga pengaksesan program dan data dapat dilakukan secara bersamaan (*concurrent*). (*Eko Putra, 2005:2*)

ATMEGA16 mempunyai *throughput* mendekati 1 *Millions Instruction Per Second (MIPS)* per MHz, sehingga membuat konsumsi daya menjadi rendah terhadap kecepatan proses eksekusi perintah.

Beberapa keistimewaan dari AVR ATMEGA16 antara lain:

- a. Mikrokontroler AVR 8 *bit* yang memiliki kemampuan tinggi dengan konsumsi daya rendah.

- b. Arsitektur RISC dengan *throughput* mencapai 16 MIPS pada frekuensi 16MHz
- c. Memiliki kapasitas *Flash* memori 16 Kbyte, EEPROM 512 Byte dan SRAM 1 Kbyte
- d. Saluran I/O sebanyak 32 buah, yaitu *Port A*, *Port B*, *Port C* dan *Port D*
- e. CPU yang terdiri dari 32 buah *register*
- f. Unit interupsi dan eksternal
- g. *Port* USART untuk komunikasi serial
- h. Fitur *peripheral*
 - 1) Tiga buah *Timer/Counter* dengan kemampuan perbandingan (*compare*)
 - 2) Dua buah *Timer/Counter* 8 bit dengan *Prescaler* terpisah dan *Mode Compare*
 - 3) Satu buah *Timer/Counter* 16 bit dengan *Prescaler* terpisah, *Mode Compare* dan *Mode Capture*
 - 4) *Real Time Counter* dengan *Oscillator* tersendiri
 - 5) Empat kanal PWM
 - 6) 8 kanal ADC
 - 7) 8 *Single-ended Channel* dengan keluaran hasil konversi 8 dan 10 resolusi (register ADCH dan ADCL)
 - 8) 7 *Differential Channel* hanya pada kemasan *Thin Quad Flat Pack* (TQFP)
 - 9) 2 *Differential Channel* dengan *Programmable Gain*
 - 10) Antarmuka *Serial Peripheral Interface* (SPI) Bus
 - 11) *Watchdog Timer* dengan *Oscillator Intern.*
 - 12) *On-chip Analog Comparator*
- i. *Non-volatile program memory.* (Sholihul Hadi, 2008:1-2)

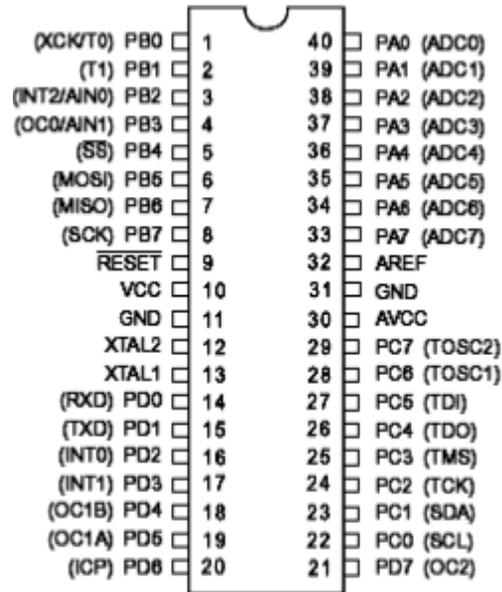
Features ATMega16 yaitu :

- 1. *High-performance, Low-power* Atmel[®] AVR[®] 8-bit *Microcontroller*
- 2. *Advanced RISC Architecture*
 - a. 131 *Powerful Instructions – Most Single-clock Cycle Execution*

- b. *32 x 8 General Purpose Working Registers*
 - c. *Fully Static Operation*
 - d. *Up to 16 MIPS Throughput at 16 MHz*
 - e. *On-chip 2-cycle Multiplier*
3. *High Endurance Non-volatile Memory segments*
- a. *16 Kbytes of In-System Self-programmable Flash program memory*
 - b. *512 Bytes EEPROM*
 - c. *1 Kbyte Internal SRAM*
 - d. *Write/Erase Cycles: 10,000 Flash/100,000 EEPROM*
 - e. *Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾*
 - f. *Optional Boot Code Section with Independent Lock Bits*
In-System Programming by On-chip Boot Program True Read-While-Write Operation
 - g. *Programming Lock for Software Security*
4. *JTAG (IEEE std. 1149.1 Compliant) Interface*
- a. *Boundary-scan Capabilities According to the JTAG Standard*
 - b. *Extensive On-chip Debug Support*
 - c. *Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface.*
5. *Peripheral Features*
- a. *Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes*
 - b. *One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode*
 - c. *Real Time Counter with Separate Oscillator*
 - d. *Four PWM Channels*
 - e. *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
 - f. *Byte-oriented Two-wire Serial Interface*
 - g. *Programmable Serial USART*

- h. *Master/Slave SPI Serial Interface*
 - i. *Programmable Watchdog Timer with Separate On-chip Oscillator*
 - j. *On-chip Analog Comparator*
6. *Special Microcontroller Features*
- a. *Power-on Reset and Programmable Brown-out Detection*
 - b. *Internal Calibrated RC Oscillator*
 - c. *External and Internal Interrupt Sources*
 - d. *Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby*
7. *I/O and Packages*
- a. *32 Programmable I/O Lines*
 - b. *40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF*
8. *Operating Voltages*
- a. *2.7V - 5.5V for ATmega16L*
 - b. *4.5V - 5.5V for ATmega16*
9. *Speed Grades*
- a. *0 - 8 MHz for ATmega16L*
 - b. *0 - 16 MHz for ATmega16*
10. *Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L*
- a. *Active: 1.1 mA*
 - b. *Idle Mode: 0.35 mA*
 - c. *Power-down Mode: < 1 μ A*
- (*Heri Andrianto, 2012*)

2.2.1 Konfigurasi Pin ATmega 16



Gambar 2.1 Konfigurasi Pin ATmega 16
(Sumber: Data Sheet ATmega16, 2013, Diakses: 5 Mei 2014)

Konfigurasi pena (*pin*) mikrokontroler ATmega16 dengan kemasan 40 pena. Dari gambar tersebut dapat terlihat ATmega16 memiliki 8 pena untuk masing-masing bandar A (*Port A*), bandar B (*Port B*), bandar C (*Port C*), dan bandar D (*Port D*).

- VCC merupakan pin yang berfungsi sebagai masukan catu daya.
- GND merupakan pin *ground*
- Port A (PA0..PA7) merupakan pin input / output dua arah dan pin inputan ADC.
- Port B (PB0..PB7) merupakan pin input / output dua arah dan pin yang memiliki fungsi khusus sebagai berikut :

Tabel 2.1 Fungsi Khusus Port B

PIN	Fungsi Khusus
PB7	SCK (SPI Bus <i>Serial Clock</i>)
PB6	MISO (SPI Bus <i>Master Input/Slave Output</i>)
PB5	MOSI (SPI Bus <i>Master Output/Slave Input</i>)

PB4	SS (<i>SPI Slave Select Input</i>)
PB3	AIN 1 (<i>Analog Comparator Negative Input</i>)OC0 (<i>Timer/Counter 0 Output Compare Match Output</i>)
PB2	AIN0 (<i>Analog Comparator Positive Input</i>)INT2 (<i>Eksternal Interrupt 2 Input</i>)
PB1	T1 (<i>Timer Counter 1 Eksternal Counter Input</i>)
PB0	T0 T1 (<i>Timer/Counter 0 Eksternal Counter Input</i>)XCK (<i>USART Eksternal Clock Input/Output</i>)

- e. Port C (PC0..PC7) merupakan pin input output dua arah dan pin fungsi khususnya yaitu :

Tabel. 2.2 Fungsi Khusus Port C

PIN	Fungsi Khusus
PC7	TOSC2 (<i>Timer Oscillator Pin2</i>)
PC6	TOSC1 (<i>Timer Oscillator Pin1</i>)
PC5	TDI (<i>JTAG Test Data In</i>)
PC4	TDO (<i>JTAG Test Data Out</i>)
PC3	TMS (<i>JTAG Test Mode Select</i>)
PC2	TCK (<i>JTAG Test Clock</i>)
PC1	SDA (<i>Two Wire Serial Bus Data Input/Output Line</i>)
PC0	SCL (<i>Two Wire Serial Bus Clock Line</i>)

- f. PORT D (PD0..PD7) memiliki pin input output dua arah dan pin khusus sebagai berikut :

Tabel 2.3 Fungsi Khusus Port D

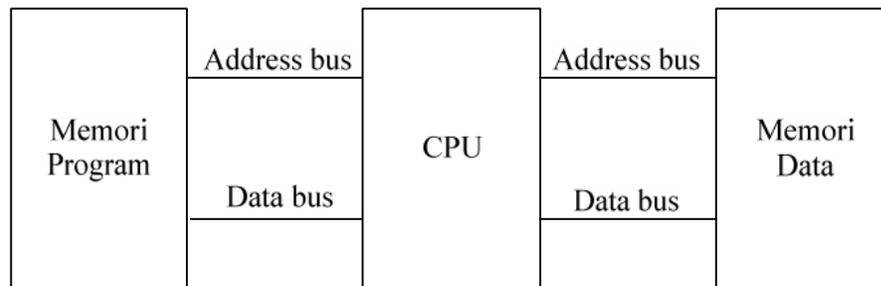
PIN	Fungsi Khusus
PD7	OC2 (<i>Timer/Counter 2 Output Compare Match Output</i>)
PD6	ICP (<i>Timer/Counter 1 Input Capture Pin</i>)

PD5	OC1A (<i>Timer/Counter 1 Output Compare A Match Output</i>)
PD4	OC1B (<i>Timer/Counter 1 Output Compare B Match Output</i>)
PD3	INT1 (<i>Eksternal Interrupt 1 Input</i>)
PD2	INT0 (<i>Eksternal Interrupt 0 Input</i>)
PD1	TXD (<i>USART Output Pin</i>)
PD0	RXD (<i>USART Input Pin</i>)

(Sumber: Pemrograman Mikrokontroler AVR ATmega 16 Menggunakan Bahasa C (Code Vision AVR), 2012:Hal. 9-11)

- g. RESET : pin ini digunakan untuk mereset ATmega 16
 - h. XTAL1 dan XTAL2 merupakan pin untuk input clock eksternal
 - i. AVCC : merupakan pin input untuk tegangan ADC
 - j. AREF : merupakan pin input untuk tegangan referensi ADC
- (Heri Andrianto, 2012:9-11)

2.2.2 Arsitektur ATmega 16

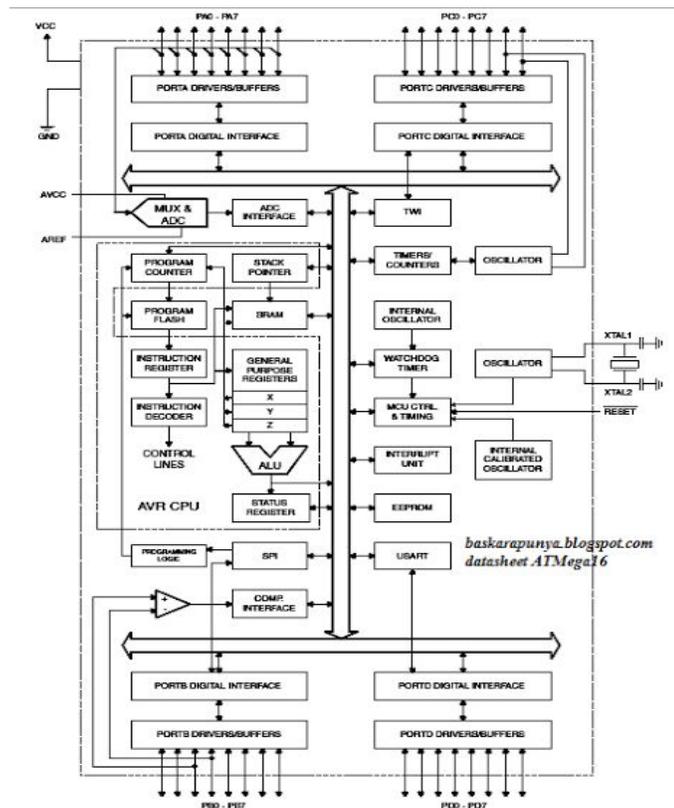


Gambar 2.2 Arsitektur Harvard

(Sumber: Pemrograman Mikrokontroler AVR ATmega16, 2013)

Dari gambar diatas, AVR menggunakan arsitektur Harvard dengan memisahkan antara memori dan bus untuk program dan data untuk memaksimalkan kemampuan dan kecepatan. Instruksi dalam memori program di eksekusi dengan *pipelining singel level*. Dimana ketika satu instruksi di eksekusi, instruksi berikutnya diambil dari memori program. Konsep ini mengakibatkan instruksi di eksekusi setiap *clock cycle*. CPU terdiri dari 32x8 – bit *general*

purpose register yang dapat diakses dengan cepat dalam satu *clock cycle*, yang mengakibatkan operasi *Arithmetic Logic Unit (ALU)* dapat dilakukan dalam satu *cycle*. Pada operasi ALU, dua *operand* berasal dari *register*, kemudian operasi di eksekusi dan hasilnya disimpan kembali ke *register* dalam satu *clock cycle*. Operasi aritmatika *logic* pada ALU akan mengubah bit – bit yang terdapat pada *Status Register (SREG)*. Proses pengambilan instruksi dan pengeksekusian instruksi berjalan secara *parallel*. (Syahrul, 2012)



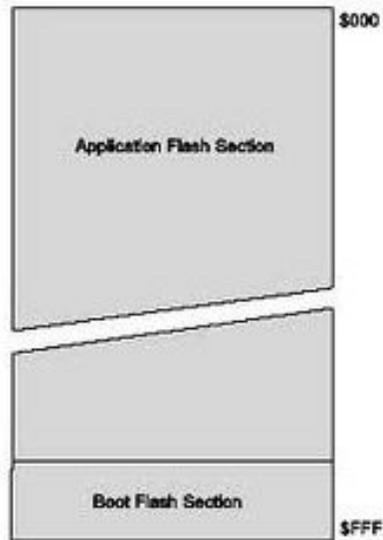
Gambar 2.3 Blog Diagram ATmega16

(Sumber: Data Sheet ATmega16, 2013, Diakses: 5 Mei 2014)

2.2.3 Peta Memori AVR ATmega16

Arsitektur AVR mempunyai dua memori utama, yaitu memori data dan memori program. Selain itu, ATmega16 memiliki memori EEPROM untuk menyimpan data. ATmega16 memiliki 16K *byte one-chip In – System Reprogrammable Flash* memory untuk menyimpan program. Karena semua

instruksi AVR memiliki format 16 atau 32 bit, *Flash* diatur dalam 8K x 16 bit. Untuk keamanan program, memori program, *flash* dibagi kedalam dua bagian, yaitu bagian program *Boot* dan aplikasi. *Bootloader* adalah program kecil yang bekerja pada saat *start up time* yang dapat memasukan seluruh program aplikasi ke dalam memori prosesor. (*Ardi Winoto, 2010*)



Gambar 2.4 Peta Memori Program AVR ATmega16

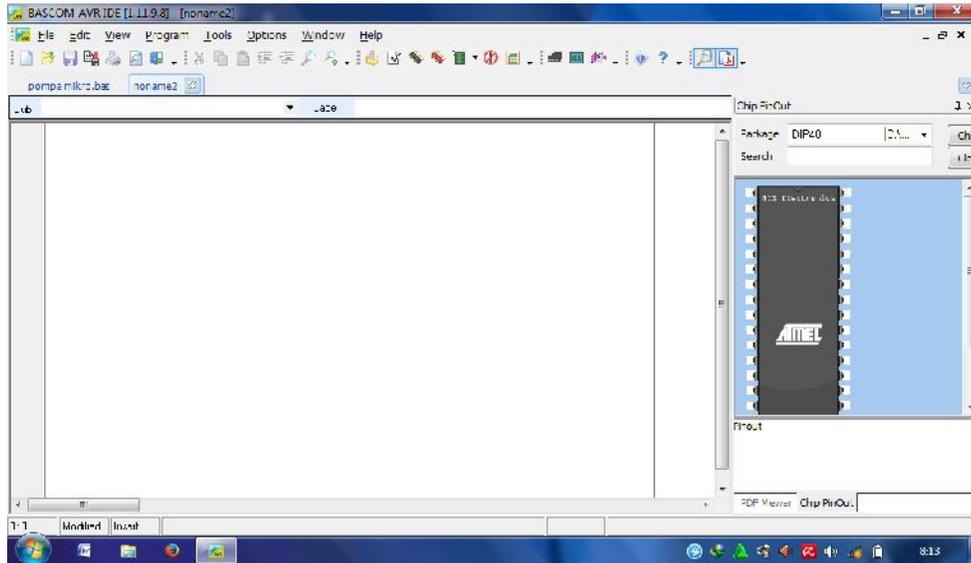
(Sumber: Data Sheet ATmega16, 2013, Diakses: 6 Mei 2014)

2.3 *Basic Compiler* AVR (BASCOS AVR)

Software yang digunakan untuk pemrograman mikrokontroler AVR ATmega16 adalah BASCOM - AVR singkatan dari *Basic Compiler*. Pemrograman menggunakan BASCOM - AVR adalah salah satu dari sekian banyak Bahasa BASIC untuk pemrograman mikrokontroler, misalnya Bahasa *Assembly*, Bahasa C, dan lain - lain. Penulis menggunakan Bahasa *BASIC* BASCOM - AVR karena penggunaannya mudah dalam penulisannya, ringkasan, cepat dimengerti bagi pemula, dan tidak kalah dengan *BASIC* lainnya.

BASCOS - AVR adalah salah satu *tool* untuk pengembangan / pembuatan program untuk kemudian ditanamkan dan dijalankan pada mikrokontroler terutama mikrokontroler keluarga AVR. BASCOM AVR juga bisa disebut sebagai IDE (*Integrated Development Environment*) yaitu lingkungan kerja yang

terintegrasi, karena disamping tugas utama (meng*compile* kode program menjadi file HEX / bahasa mesin), BASCOM - AVR juga memiliki kemampuan / fitur lain yang berguna sekali. (*Ardi Winoto, 2010*)

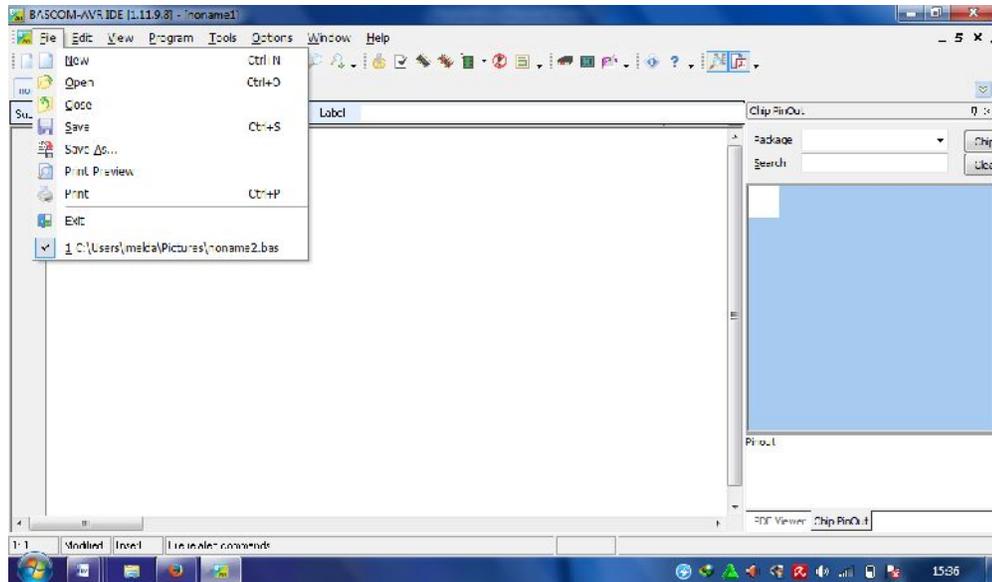


Gambar 2.5 Tampilan Jendela BASCOM - AVR

Tabel 2.4 Info Show Result

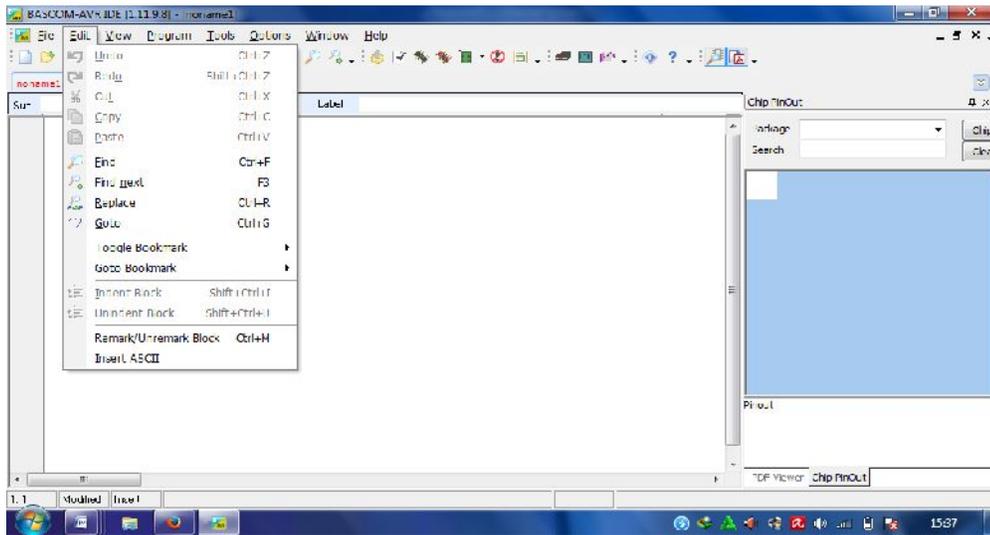
Info	Keterangan
<i>Compiler</i>	Versi dari <i>compiler</i> yang digunakan
<i>Processor</i>	Menampilkan target <i>processor</i> yang dipilih
<i>Date and Time</i>	Tanggal dan waktu kompilasi
<i>Baud rate dan xtal</i>	<i>Baudrate</i> yang dipilih dan kristal yang digunakan uP
<i>Error</i>	<i>Error</i> nilai Baud yang diset dengan nilai Baud sebenarnya
<i>Flash Used</i>	Persentase Flash ROM yang terisi program
<i>Stack Start</i>	Lokasi awal <i>stack pointer</i> memori
RAM Start	Lokasi awal RAM
LCD Mode	<i>Mode</i> LCD yang digunakan, 4 bit, 8 bit

2.3.1 Bagian – bagian dari Tampilan Jendela BASCOM –AVR



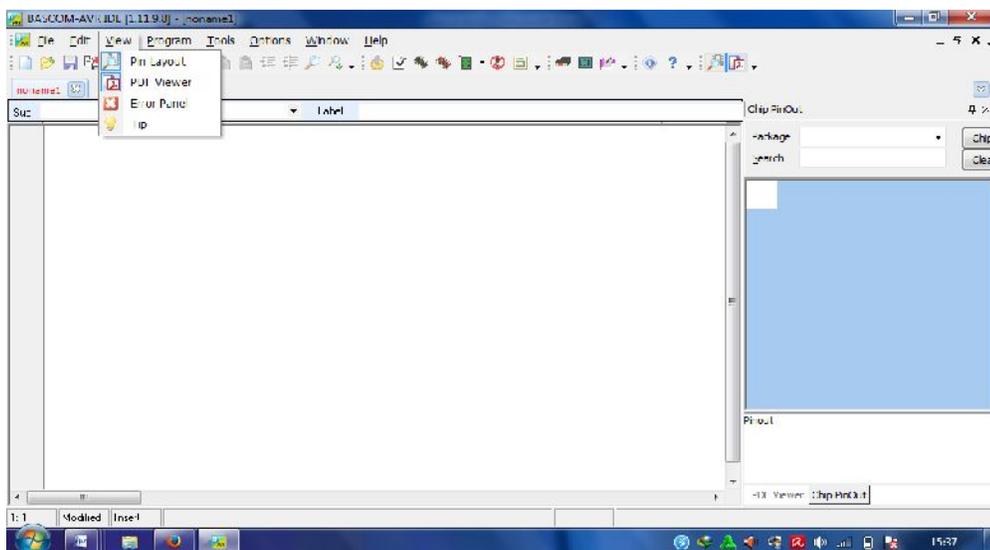
Gambar 2.6 Halaman Menubar Pada File

- a. **New**, digunakan untuk membuat project baru atau membuat file program baru.
- b. **Open**, digunakan untuk membuka *project* atau *file* program yang pernah dibuat.
- c. **Save**, digunakan untuk menyimpan *project* atau menyimpan *file* program.
- d. **Save As**, digunakan digunakan untuk menyimpan *project* atau menyimpan *file* dengan nama yang berbeda dari sebelumnya.
- e. **Print Preview**, digunakan untuk melihat hasil cetakan *print out* dari sintaks penulisan program.
- f. **Print**, digunakan untuk mencetak *file* program.
- g. **Exit**, digunakan untuk keluar dari BASCOM AVR



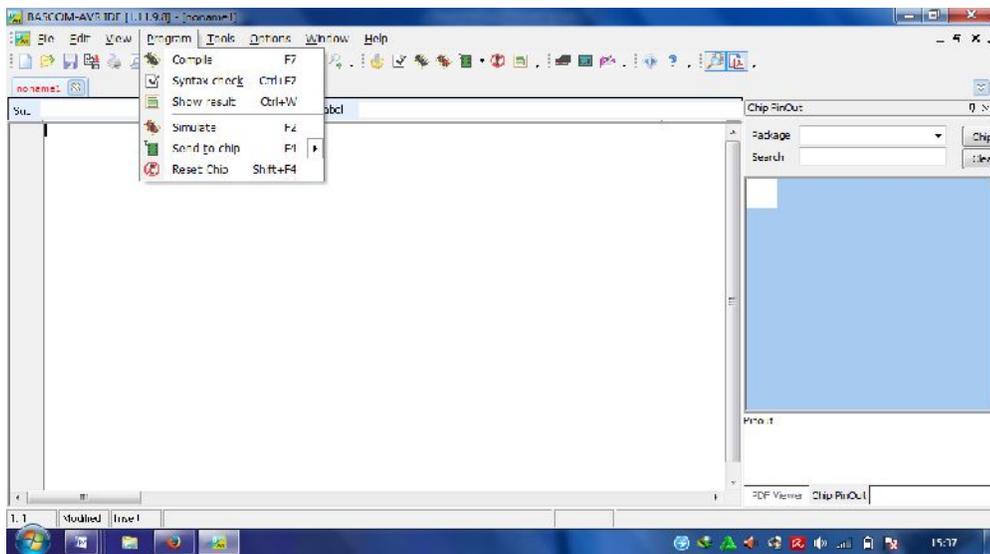
Gambar 2.7 Halaman Menubar Pada Edit

- a. **Undo**, digunakan untuk kembali ke langkah sebelumnya.
- b. **Redo**, kebalikan dari *undo*.
- c. **Cut**, digunakan untuk mengkopi dan menghapus teks sekaligus
- d. **Copy**, digunakan untuk mengkopi teks.
- e. **Paste**, digunakan untuk menyalin bagian yang telah dikopi.
- f. **Find**, digunakan untuk mencari teks yang diinginkan.
- g. **Find next**, sama halnya dengan *find* hanya saja berikutnya.



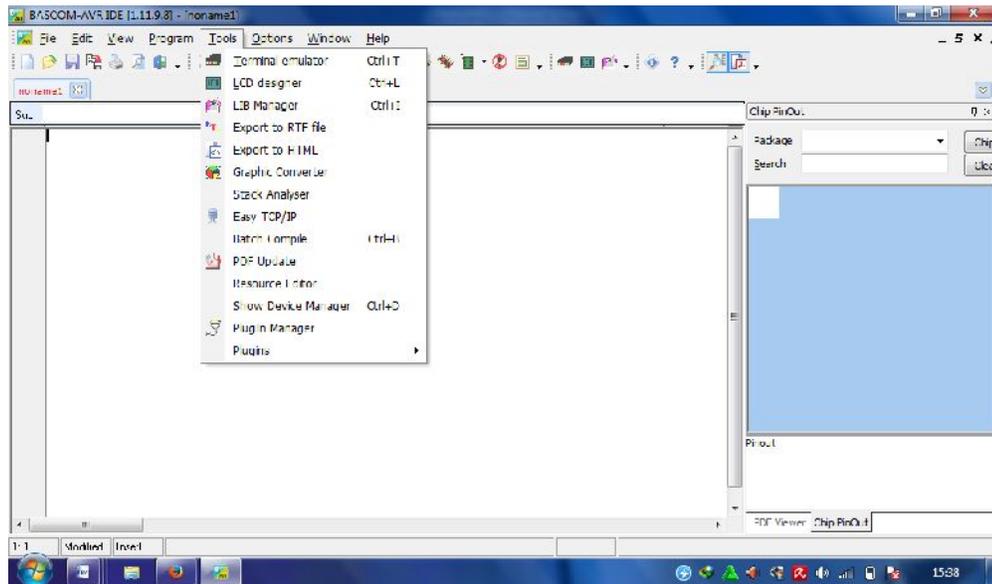
Gambar 2.8 Halaman Menubar Pada View

- a. **Pin Layout**, digunakan untuk menampilkan susunan pin / *chip pinout* yang dapat kita lihat pada gambar diatas.
- b. **Error panel**, yaitu panel untuk menampilkan apabila terjadi kesalahan / *error* pada saat kita membuat suatu program.
- c. **Tip**, digunakan sebagai petunjuk kita dalam melakukan suatu program.



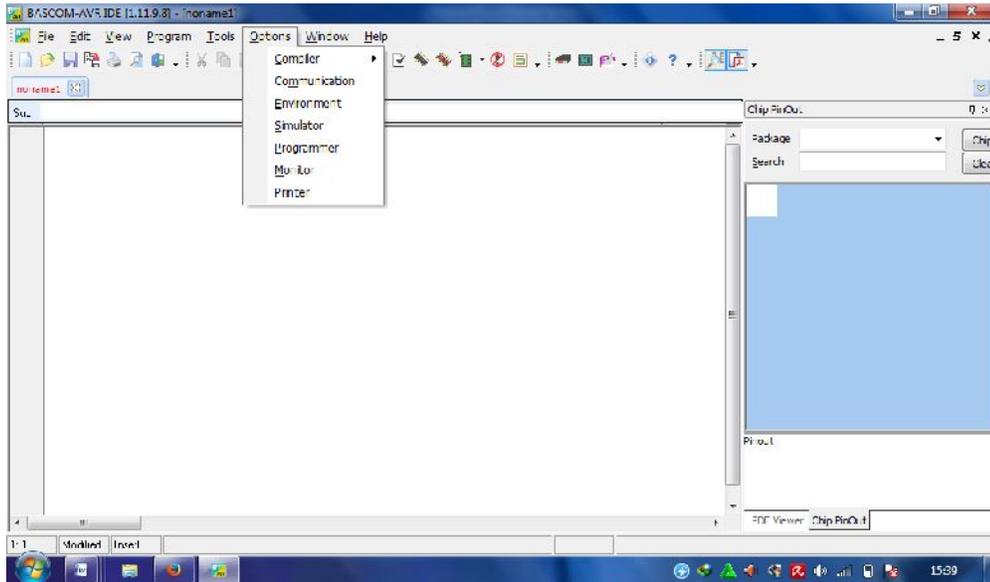
Gambar 2.9 Halaman Editor Menu Pada Program

- a. **Compile**, digunakan untuk mengkompile program. Proses ini akan menghasilkan file berektension *.hex
- b. **Syntax check**, digunakan untuk memeriksa apakah terjadi kesalahan pada penulisan program atau tidak.
- c. **Show result**, digunakan untuk melihat hasil *report* dan *error* dari penulisan program.
- d. **Simulate**, digunakan untuk mensimulasikan program.
- e. **Send to chip**, digunakan untuk mengirim file *.hex ke dalam *chip* mikrokontroler (mendownload program mikrokontroler).



Gambar 2.10 Halaman Editor *Menubar Pada Tools*

- a. **Terminal emulator**, digunakan untuk simulasi komunikasi serial dengan komputer (RS232) hampir sama dengan *Hypert Terminal* yang dimiliki oleh *Windows*.
- b. **Lcd designer**, digunakan untuk mendesain karakter LCD yang diinginkan.
- c. **Library Manager**, digunakan untuk *library* yang terdapat pada BASCOM AVR
- d. **Export to RTF**, digunakan untuk mengkonversi penulisan program pada RTF (*Rich Text Format*).
- e. **Graphic Converter**, digunakan untuk mengkonversi gambar ke LCD yang menjang RGB (*high* kualitas LCD).
- f. **Stack Analyser**, digunakan untuk menganalisa *stack* program.
- g. **PlugIn Manager**, digunakan untuk mengatur plugin yang ada.



Gambar 2.11 Halaman *Menubar* Pada *Options*

- a. **Compiler**, digunakan untuk *mensetting chip, output, communication, I2C* dan LCD.
- b. **Communication**, digunakan untuk *mensetting komunikasi mikrokontroler*.
- c. **Simulator**, digunakan untuk *mensetting simulasi pada BASCOM AVR*.
- d. **Programmer**, digunakan untuk *mensetting downloader programmer* yang akan digunakan.
- e. **Monitor**, untuk *mensetting tampilan*.
- f. **Printer**, digunakan untuk *mensetting printer* yang digunakan.

2.3.2 Karakter Pada BASCOM

Dalam program BASCOM, karakter dasarnya terdiri atas karakter alphabet (A-Z dan a-z), karakter numeric (0-9) dan karakter spesial seperti yang ditunjukkan pada tabel di bawah ini.

Tabel 2.5 Karakter-karakter spesial pada BASCOM

Karakter	Nama
	<i>Blank</i> atau spasi
'	Apostrophe

*	Asteriks atau simbol perkalian
+	Simbol Pertambahan (<i>Plus Sign</i>)
,	<i>Comma</i>
-	Simbol Pengurangan (<i>Minus Sign</i>)
.	<i>Period (decimal point)</i>
/	<i>Slash (division symbol) will be handled as \</i>
:	<i>Colon</i>
“	<i>Double Quotation mark</i>
;	<i>Semicolon</i>
<	<i>Less than</i>
=	<i>Equal sign (assignment symbol or relation operator)</i>
>	<i>Greater than</i>
\	<i>Backslash (integer/word division symbol)</i>

2.3.3 Tipe Data BASCOM

Setiap variabel dalam BASCOM memiliki tipe data yang menunjukkan daya tampungnya. Hal ini berhubungan dengan penggunaan memori mikrokontroler. Berikut ini adalah tipe data pada BASCOM berikut keterangannya.

Tabel 2.6 Tipe Data BASCOM

Tipe Data	Ukuran (byte)	Range
Bit	1/8	0-1
<i>Byte</i>	1	0 sampai 255
<i>Interger</i>	2	-32,768 sampai 32,767
<i>Word</i>	2	0 sampai 65535
<i>Long</i>	4	-2147483648 sampai 2147483647

<i>Single</i>	4	-
<i>String</i>	Hingga 254 byte	-

2.3.4 Macam – macam perintah BASCOM

IF – THEN

Perintah *IF – THEN* digunakan untuk menguji suatu keadaan benar atau salah dan menentukan tindakan yang sesuai dengan keinginan.

Perintahnya:

```

If <keadaan> Then <Perintah>      '1 baris perintah
End IF
If <keadaan> Then                  'lebih dari 1 baris perintah
    <Perintah_1>
    <Perintah_2>
    <Perintah_n>
End If

```

IF – THEN-ELSE

Perintah *IF – THEN – ELSE* digunakan untuk menguji dua keadaan (benar ataupun salah) dan menentukan tindakan yang sesuai dengan keinginan.

Perintahnya:

```

If <keadaan> Then
    <Perintah_1>
Else
    <Perintah_2>
End IF

```

IF – THEN-ELSEIF

Perintah *IF – THEN – ELSEIF* digunakan untuk menguji lebih dari satu keadaan dan menentukan tindakan sesuai dengan keinginan.

Perintahnya:

```
If <keadaan_1> Then
    <Perintah_1>
Elseif <keadaan_2>Then
    <Perintah_2>
Elseif <Keadaan_3>
    .....
End If
```

SELECT – CASE

Perintah *SELECT – CASE* digunakan untuk pengujian keadaan yang banyak sehingga penulisan menjadi lebih sederhana.

Perintahnya:

```
Select case <Nama_variabel>
    Case 1 : <perintah_1>
    Case 2 : <Perintah_2>
    Case 3 : <Perintah_3>
    .....
End Select
```

DO – LOOP

Perintah *DO – LOOP* merupakan perintah untuk perulangan yang digunakan untuk melakukan perulangan program selama suatu kondisi telah terpenuhi.

Perintahnya:

```

Do
If <keadaan_1> Then
    <Perintah_1>
Elseif <keadaan_2>Then
    <Perintah_2>
Elseif <Keadaan_3>
.....
End If

```

Perintah *FOR – NEXT* merupakan perintah untuk perulangan yang digunakan untuk melakukan perulangan sesuai dengan jumlah dan tingkat peluangannya.

Perintahnya:

```

For <Variabel=Nilai_awal> To <Nilai_akhir> <selisi_pertambahan>
    <Pertanyaan>
Next

```

WHILE – WEND

Perintah *WHILE – WEND* merupakan perintah untuk perulangan yang akan melakukan perulangan apabila keadaan yang diminta telah terpenuhi.

Perintahnya:

```
While <keadaan>
    <Perintah>
Wend
```

EXIT

Perintah *EXIT* merupakan perintah untuk mengakhiri perulangan *DO – LOOP*, *FOR – NEXT*, *WHILE – WEND*.

Perintahnya:

```
<Pernyataan>
EXIT.....
```

(Sumber: Pemrograman Mikrokontroler AVR ATMega 16 Menggunakan Bahasa C (Code Vision AVR), 2013)

2.4 Pompa Air

Pompa adalah suatu alat atau mesin yang digunakan untuk memindahkan cairan dari suatu tempat ke tempat yang lain melalui suatu media perpipaan dengan cara menambahkan energi pada cairan yang dipindahkan dan berlangsung secara terus menerus.

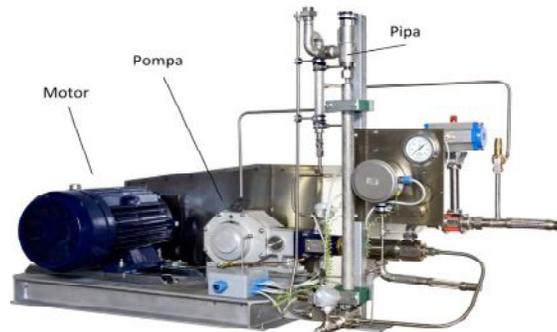
Pompa beroperasi dengan prinsip membuat perbedaan tekanan antara bagian masuk (*suction*) dengan bagian keluar (*discharge*). Dengan kata lain, pompa berfungsi mengubah tenaga mekanis dari suatu sumber tenaga (penggerak) menjadi tenaga kinetis (kecepatan), dimana tenaga ini berguna untuk mengalirkan cairan dan mengatasi hambatan yang ada sepanjang pengaliran.

Pompa juga dapat digunakan pada proses - proses yang membutuhkan tekanan *hidraulik* yang besar. Dalam operasi, mesin-mesin peralatan berat membutuhkan tekanan *discharge* yang besar dan tekanan isap yang

rendah. Akibat tekanan yang rendah pada sisi isap pompa maka fluida akan naik dari kedalaman tertentu, sedangkan akibat tekanan yang tinggi pada sisi *discharge* akan memaksa fluida untuk naik sampai pada ketinggian yang diinginkan. Dalam aplikasi kehidupan sehari - hari banyak sekali aplikasi yang berkaitan dengan pompa. Contoh pompa yang ditemui dalam kehidupan sehari - hari antara lain pompa air, pompa diesel, pompa hydram, pompa bahan bakar dan lain-lain. Dari sekian banyak pompa yang ada tentunya mempunyai prinsip kerja dan kegunaan yang berbeda-beda, penggerak merubah energi listrik menjadi energi mekanik yang diperlukan untuk menggerakkan pompa. Energi ditransmisi ke pompa oleh suatu *belt* ke *pully* penggerak pompa. Aksi kerja pompa dimaksud adalah terjadinya kerja pemompaan yang dilakukan oleh pompa *reciprocating* untuk satu siklus gerak bolak - balik batang plunger silinder.

Pompa menggerakkan energi mekanik sebagai berikut :

- a. Untuk menggerakkan atau mengalirkan cairan yang diproses melalui pompa pada kapasitas cairan yang diperlukan.
- b. Untuk memindahkan energi kedalam cairan yang diproses, yang terlihat dengan bertambahnya tekanan cairan pada lubang keluar pompa.



Gambar 2.12 Konstruksi Pompa

(Sumber: Putra R. 2012, Diakses: 5 Mei 2014)

2.5 Sensor Air

Rangkaian kontrol ketinggian air / sensor air otomatis berfungsi untuk mengatur proses pengisian air pada bak penampungan (tandon) dengan cara mendeteksi level air pada posisi sebelum penuh dan posisi sebelum habis.

Rangkaian kontrol ketinggian air otomatis ini akan mengontrol mesin pompa air untuk hidup pada saat air sebelum habis dan akan mematikan mesin pompa air pada saat bak air penuh. Rangkaian kontrol ketinggian air ini sering juga disebut dengan rangkaian kontrol mesin pompa air. Rangkaian kontrol ketinggian air diatas membaca level air pada bak penampungan air menggunakan sensor berupa plat logam yang diletakan pada bak air. Ada 2 sensor ketinggian air yang dipasang pada bak air tersebut, yaitu sensor level air posisi penuh yang berfungsi untuk mendeteksi level ketinggian air maksimum pada bak penampungan air. Dan sensor level air posisi habis yang berfungsi untuk mendeteksi posisi air minimum yang diperbolehkan. Sensor level air penuh diletakan di sisi atas bak penampungan air dan sensor level air minimal diletakkan di posisi minimal. Kemudian air bak penampungan tersebut dihubungkan ke *ground* dengan memasang plat hingga dasar penampungan agar sensor level air dapat bekerja. (Putra, 2012)

2.6 Shower Air

Shower adalah alat mandi yang praktis. Ada banyak manfaat mandi dengan menggunakan *shower*. Pertama, itu menunjukkan kepedulian kita terhadap lingkungan dengan memanfaatkan air secara maksimal. Karena penggunaan air dengan *shower* saat mandi jauh lebih hemat daripada kita menggunakan gayung.

Manfaat kedua, dengan menggunakan air kita dapat memperoleh lebih banyak ion negatif yang bermanfaat bagi kulit kita. Dalam udara bebas, sebenarnya ada ion negatif dan positif yang bersatu. Dengan adanya semprotan air, maka ion negatif dan ion positif terpecah. Ion positif yang lebih berat, membuat ion ini cepat jatuh sedangkan ion negatif dapat melayang di udara. Itulah sebabnya kita merasa lebih nyaman dan segar saat berada di dekat air terjun atau saat menghirup udara setelah hujan karena ada banyaknya ion negatif di udara. (fardams, 2011)