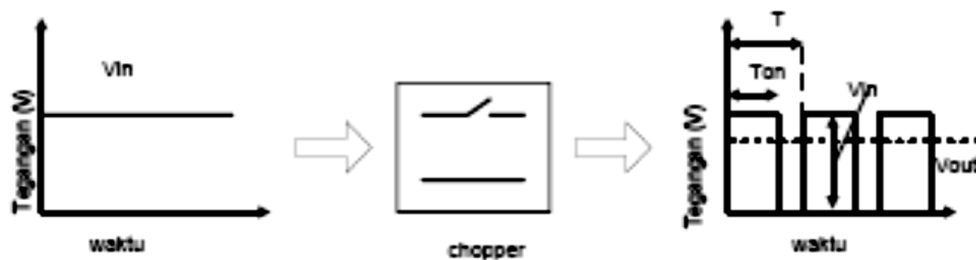


II. TINJAUAN PUSTAKA

A. Konverter DC ke DC Tipe Pensaklaran

Terdapat dua macam cara untuk mengolah daya yaitu tipe linear dan tipe pensaklaran (*switching*). Dalam tugas akhir ini akan digunakan cara yang kedua dalam pengolahan daya yaitu menggunakan konverter DC ke DC tipe pensaklaran.



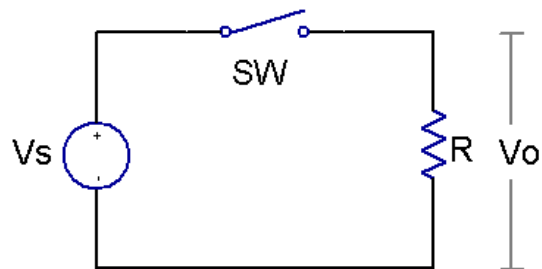
Gambar 1. Proses perubahan tegangan arus searah ke arus searah dengan konverter DC ke DC (*chopper*).

Konverter DC ke DC atau biasa disebut *chopper* diperlukan untuk mengubah tegangan DC tetap menjadi tegangan DC yang bersifat variabel. Seperti halnya pada transformator yang berfungsi menaikkan atau menurunkan tegangan AC, konverter DC ke DC dapat digunakan untuk menaikkan atau menurunkan tegangan DC. Rangkaian konverter DC ke DC memerlukan suatu sumber tegangan DC yang konstan dan tegangan keluaran yang bervariasi dengan

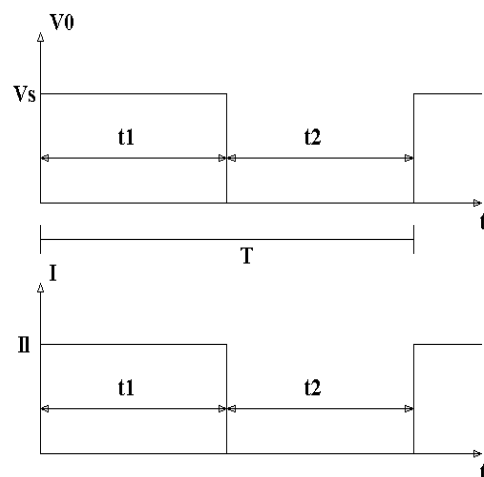
mengatur waktu pensaklaran pada transistor yang dihubungkan ke beban. Cara kerjanya adalah suatu tegangan arus searah konstan di *switch on* dan *off* untuk mereduksi harga mula-mula tegangan yang digunakan pada beban.

1) *Chopper Step-Down*

Prinsip kerja *chopper step-down* dapat dijelaskan melalui gambar 2 sebagai berikut.



a). Rangkaian *Chopper step-down*



b). Bentuk gelombang tegangan dan arus

Gambar 2. *Chopper step-down*

Ketika saklar SW ditutup selama waktu t_1 , tegangan masukan V_s muncul melalui beban. Bila saklar *off* selama waktu t_2 , tegangan yang melalui beban adalah nol. Bentuk gelombang untuk tegangan keluaran dan arus beban ditunjukkan pada gambar 2. (b). Saklar *chopper* dapat diimplementasikan dengan menggunakan sebuah BJT daya, MOSFET daya, IGBT atau *thyristor* komutasi paksa.

Tegangan keluaran rata-rata dari *chopper* diberikan oleh

$$V_o = \frac{1}{T} \int_0^{t_1} V_0 dt = \frac{t_1}{T} V_s = f t_1 V_s = k V_s \quad 2-2$$

Dan arus beban rata-rata, $I_a = V_a/R = kV_s/R$, dengan T adalah periode *chopping*, $k = t_1/T$ adalah *duty cycle chopper*, dan f adalah frekuensi *chopping*. Nilai rms tegangan keluaran ditentukan oleh

$$V_0 = \left(\frac{1}{T} \int_0^{kT} V_0^2 dt \right)^{1/2} = \sqrt{k} V_s \quad 2-3$$

Dengan mengasumsikan bahwa tidak ada rugi-rugi pada *chopper* maka daya masukan pada *chopper* sama dengan daya keluaran yang diberikan dengan persamaan

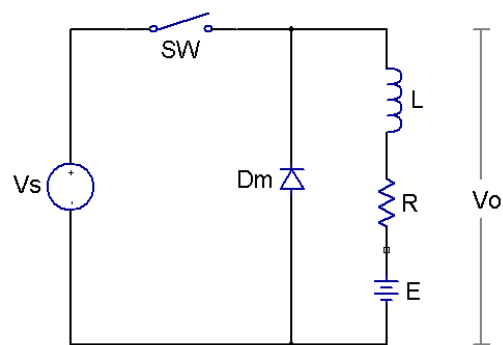
$$P_i = \frac{1}{T} \int_0^{kT} V_0 i dt = \frac{1}{T} \int_0^{kT} \frac{V_0^2}{R} dt = k \frac{V_s^2}{R} \quad 2-4$$

Resistansi masukan efektif yang dilihat dari sumber

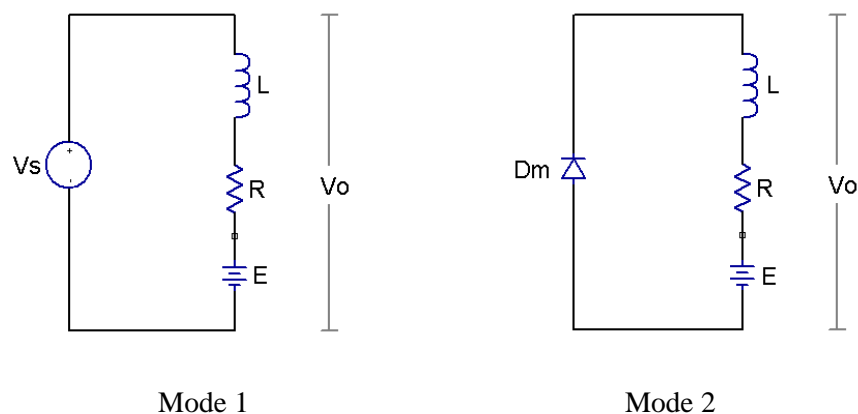
$$R_i = \frac{V_s}{I_a} = \frac{V_s}{kV_s/R} = \frac{R}{k} \quad 2-5$$

Duty cycle (k) dapat divariasikan dari 0 sampai 1 dengan variasi menurut t_1 , T atau f , maka tegangan keluaran V_0 dapat divariasikan dari 0 sampai V_s dengan mengatur k .

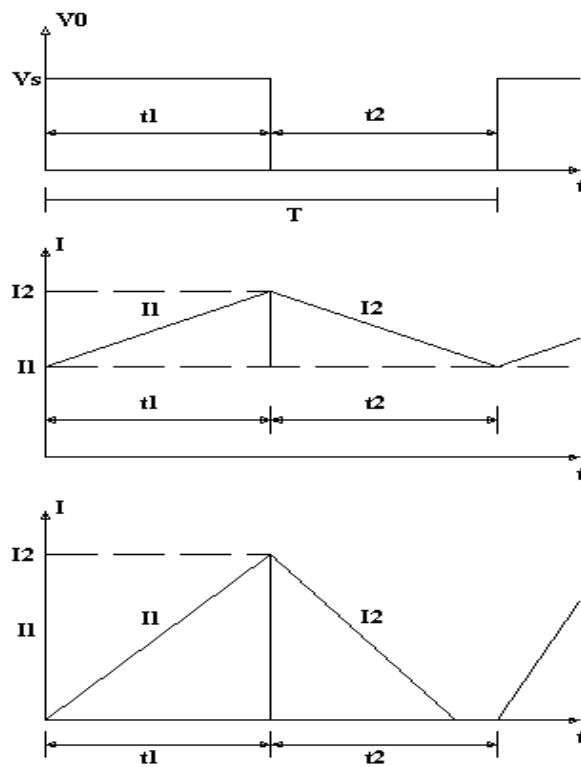
Chopper dengan beban RL ditunjukkan pada gambar 3. Cara kerja *chopper* dapat dibagi menjadi dua mode. Selama mode 1, *chopper* akan *on* dan arus akan mengalir ke beban. Selama mode 2, *chopper* akan *off* dan arus beban terus mengalir melalui dioda freewheeling D_m . Rangkaian ekuivalen untuk mode-mode ini ditunjukkan pada gambar 4.(a). Bentuk gelombang arus beban dan tegangan keluaran ditunjukkan pada gambar 4.(b).



3. Chopper step-down dengan beban RL



(a) Rangkaian ekuivalen



(b) Bentuk gelombang

Gambar 4. Rangkaian ekivalen *chopper step-down* dan bentuk gelombang untuk beban RL

Arus beban untuk mode 1 dapat ditentukan melalui

$$V_s = Ri_1 + L \frac{di_1}{dt} + E \quad 2-6$$

Penyelesaian persamaan 2-6 dengan arus mula $i_1(t=0) = I_1$ menghasilkan arus beban

$$i_1(t) = I_1 e^{-tR/L} + \frac{V_s + E}{R} (1 - e^{-tR/L}) \quad 2-7$$

Mode ini berlaku untuk $0 \leq t \leq t_1 (= kT)$; dan pada akhir mode ini, arus beban menjadi

$$i_1(t = t_1 = kT) = I_2 \quad 2-8$$

Arus beban untuk mode 2 dapat ditentukan melalui

$$0 = Ri_2 + L \frac{di_2}{dt} + E \quad 2-9$$

Dengan arus mula $i_2(t = 0) = I_2$ dan mendefinisikan arus mula (misalnya, $t = 0$) pada permulaan mode 2 kita dapatkan

$$i_2(t) = I_2 e^{-tR/L} - \frac{E}{R} (1 - e^{-tR/L}) \quad 2-10$$

Mode ini berlaku untuk $0 \leq t \leq t_2 [= (1 - k) T]$ dan pada akhir mode ini arus beban menjadi

$$i_2(t = t_2) = I_3 \quad 2-11$$

Diakhir mode 2, *chopper* menjadi *on* kembali pada siklus berikutnya setelah waktu $T = 1/f = t_1 + t_2$.

Dalam kondisi keadaan tunak, $I_1 = I_3$. arus *ripple* beban puncak ke beban dapat ditentukan dari persamaan 2-7, 2-8, 2-10, 2-11. dari persamaan 2-7 dan 2-8 memberikan nilai

$$I_2 = I_1 e^{-RT/L} + \frac{V_s - E}{R} (1 - e^{-RT/L}) \quad 2-12$$

Dari persamaan 2-10 dan 2-11 memberikan nilai I_3

$$I_3 = I_1 = I_2 e^{-(1-k)TR/L} - \frac{E}{R} (1 - e^{-(1-k)TR/L}) \quad 2-13$$

Arus *ripple* puncak-ke puncak adalah

$$\Delta I = I_2 - I_1 \quad 2-14$$

Dan setelah penyederhanaan menjadi

$$\Delta I = \frac{V_s}{R} \frac{1 - e^{-kTR/L} + e^{-TR/L} - e^{-(1-k)TR/L}}{1 - e^{-TR/L}} \quad 2-15$$

Kondisi untuk *ripple* maksimum

$$\frac{d(\Delta I)}{dk} = 0 \quad 2-16$$

Menghasilkan $e^{-kTR/L} - e^{-(1-k)TR/L} = 0$ atau $-k = -(1-k)$ atau $k = 0,5$. Arus *ripple* puncak ke puncak maksimum (pada $k = 0,5$) adalah

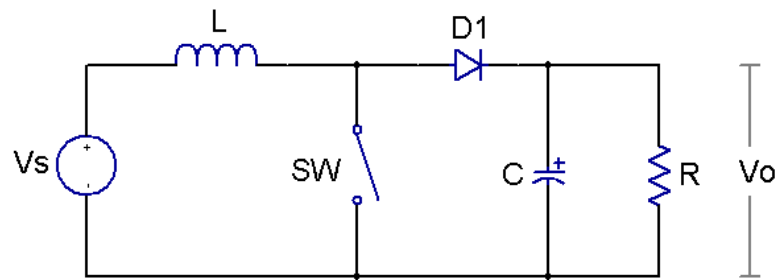
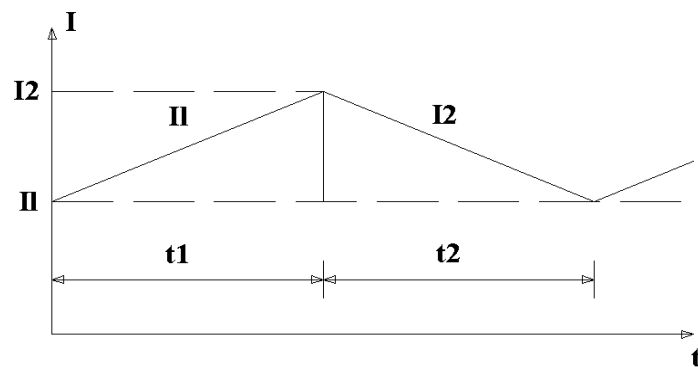
$$\Delta I_{max} = \frac{V_s}{R} \tanh \frac{R}{4fL} \quad 2-17$$

Untuk $4fL \gg R$, $\tanh \theta \approx \theta$ dan arus *ripple* maksimum dapat diaproksimasi menjadi

$$\Delta I_{max} = \frac{V_s}{4fL} \quad 2-18$$

2. Chopper Step-Up

Chopper dapat digunakan untuk menaikkan tegangan DC. susunan kerja untuk operasi *chopper step-up* ditunjukkan pada gambar 5 sebagai berikut.

a). Rangkaian *Chopper step-up*

b). Bentuk gelombang arus

Gambar 5. *Chopper step-up*

Bila saklar SW ditutup selama waktu t_1 , arus induktor menjadi naik dan energi akan tersimpan pada induktor (L). Bila saklar dibuka selama waktu t_2 , energi yang tersimpan pada induktor akan dipindahkan ke beban melalui dioda (D_1) dan arus induktor menjadi jatuh. Dengan asumsi arus yang mengalir adalah tetap, bentuk gelombang untuk induktor ditunjukkan pada gambar 5.(b).

Bila *chopper* di-on-kan, tegangan yang melalui induktor adalah

$$V_L = L \frac{di}{dt}$$

Dan ini akan menimbulkan arus *ripple* puncak ke puncak pada induktor

$$\Delta I = \frac{V_s}{L} t_1 \quad 2-20$$

Sehingga tegangan keluaran instantaneous adalah

$$V_0 = V_s + L \frac{\Delta I}{t_2} = V_s \left(1 + \frac{t_1}{t_2} \right) = V_s \frac{1}{1-k} \quad 2-21$$

B. Jenis-jenis Transistor Daya yang Digunakan sebagai Pensaklaran

1. BJT (*Bipolar Junction Transistor*)

Sebuah transistor bipolar dibentuk dengan menambahkan sebuah *p*-kedua atau wilayah-*n* ke diode sambungan *pn*. Dengan dua wilayah-*n* dan satu wilayah-*p*, ini dikenal dengan transistor NPN. Sedangkan transistor yang terbentuk dari dua wilayah-*p* dan satu wilayah-*n* disebut transistor PNP. Tiga terminalnya disebut sebagai kolektor, emiter dan basis. Sebuah transistor bipolar memiliki dua sambungan yaitu sambungan kolektor-basis (CBJ) dan sambungan basis-emiter (BEJ)

a) Karakteristik Tunak

Ada tiga wilayah operasi dari transistor, yaitu: *cutoff*, aktif dan saturasi. Pada wilayah *cutoff* transistor mati atau arus basis tidak mencukupi untuk mengaktifkan transistor dan kedua sambungan memiliki bias mundur. Pada wilayah aktif, transistor berlaku seperti penguat, dimana arus kolektor dikuatkan oleh sebuah gain dan tegangan kolektor emiter menurun terhadap arus basis. CBJ bias mundur, dan BEJ bias maju. Sedangkan pada wilayah saturasi arus

basis tinggi sehingga tegangan kolektor-emiter rendah, dan transistor berlaku sebagai saklar. Kedua sambungan (CBJ dan BEJ) dalam keadaan bias maju.

Model sebuah transistor NPN ditunjukkan pada gambar dibawah

Pada transistor NPN berlaku persamaan

$$I_E = I_C + I_B$$

Arus basis secara efektif adalah arus masukan dan arus kolektor adalah arus keluaran, rasionya disebut beta arus β

$$\beta = \frac{I_C}{I_B}$$

Arus kolektor memiliki dua komponen, yaitu: pertama karena arus basis dan kedua karena arus bocor CBJ

$$I_C = \beta I_B + I_{CEO}$$

Dengan I_{CEO} arus bocor kolektor-emiter dengan basis rangkaian terbuka dan dapat diabaikan dibandingkan dengan βI_B .

$$I_E = I_B(1 + \beta) + I_{CEO}$$

Arus kolektor dapat dinyatakan dengan

$$\alpha = \frac{\beta}{\beta + 1}$$

Atau

$$\beta = \frac{\alpha}{1 - \alpha}$$

b) Batasan Pensaklaran

Breakdown kedua adalah merupakan fenomena yang terdapat pada transistor bipolar. Fenomena ini bersifat merusak, *breakdown* kedua dihasilkan dari aliran arus ke bagian kecil basis, menghasilkan *hot spot local*. Jika energy di hot spot tersebut mencukupi akan mengakibatkan panas lokal berlebih yang dapat merusak transistor. *Breakdown* kedua terjadi saat kombinasi tertentu antara tegangan, arus dan waktu.

Area operasi aman bias maju, selama penyalaan dan kondisi nyala, temperature sambungan rata-rata dan *breakdown* kedua membatasi kemampuan menanggung daya dari sebuah transistor. Area operasi aman bias maju biasanya mengindikasikan batas I_C dan V_{CE} transistor.

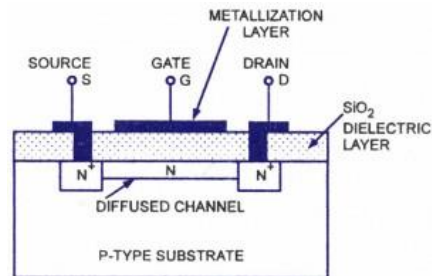
Area operasi aman bias balik, selama pematian sebuah arus tinggi dan tegangan tinggi harus diatasi oleh transistor. Tegangan kolektor emiter harus dijaga pada tingkat yang aman dengan katalain harus dibawah nilai arus kolektor yang ditentukan. Biasanya pembuat mencantumkan nilai I_C dan V_{CE} untuk bias mundur pematian

2. MOSFET (*Metal-Oxide Semiconductor Field Effect Transistor*)

MOSFET daya merupakan komponen yang dikendalikan oleh tegangan dan memerlukan arus masukan yang kecil. Kecepatan *switching* sangat tinggi dan memiliki orde nanodetik. MOSFET daya akan banyak diterapkan pada persoalan-persoalan *converter* frekuensi tinggi daya rendah. MOSFET tidak memiliki masalah fenomena *breakdown* ke dua seperti pada BJT, akan tetapi MOSFET memiliki persoalan pengosongan muatan elektrostatis dan memerlukan penanganan yang hati-hati. Sebagai tambahan, MOSFET lebih sulit diproteksi terhadap kondisi hubung singkat.

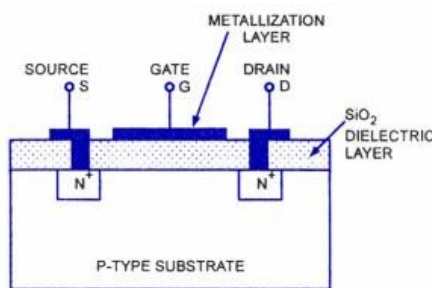
MOSFET memiliki dua tipe yaitu: MOSFET deplesi dan MOSFET enhancement. MOSFET tipe deplesi kanal-*n* dibentuk dari substrat silikon tipe-*p* yang diperlihatkan pada gambar 6. Dengan dua silikon yang disopping n^+ dengan berat agar memiliki resistansi hubungan yang rendah. Gerbang diisolasi dari kanal dengan lapisan tipis oksida. Tiga terminal disebut gerbang, drain, sumber. Substrate biasanya dihubungkan dengan sumber.. tegangan gerbang ke sumber V_{GS} negative, banyak dari elektron-elektron pada daerah kanal-*n* akan tersingkir dari suatu daerah deplesi akan terbentuk dibawah lapisan oksida, dengan menghasilkan kanal efektif yang lebih lebar dan resistansi yang tinggi dari drain ke sumber, R_{DS} . Jika V_{GS} dibuat cukup negatif, kanal akan terdepleksi penuh, yang menghasilkan R_{DS} yang tinggi dan tidak akan ada arus mengalir dari drain ke sumber, $I_{DS} = 0$. Nilai V_{GS} ketika hal ini terjadi disebut tegangan *pinch-off*, V_p . padahal V_{GS}

dibuat positif, kanal menjadi lebih lebar dan I_{DS} akan meningkat karena reduksi dari R_{DS} . Dengan MOSFET tipe deplesi kanal- p , polaritas V_{DS} , I_{DS} dan V_{GS} akan terbalik.



N-Channel DE-MOSFET Structure

a) MOSFET deplesi kanal- n



N-Channel E-MOSFET Structure

b) MOSFET enhancement kanal- n

Gambar 6. MOSFET tipe deplesi dan enhancement kana- n

MOSFET tipe enhacement tidak memiliki kanal- n fisik, seperti pada gambar 6. b). Jika V_{GS} positif, suatu tegangan enduksi akan menarik elektron dari substract- p dan mengumpulkannya pada permukaan dibawah lapisan oksida. V_{GS} lebih besar atau sama dengan nilai yang dikenal sebagai tegangan threshold, V_T , jumlah elektron yang terakumulasi akan cukup untuk membentuk kanal- n virtual dan arus mengalir dari drain ke sumber.

3. IGBT (*Insulated Gate Bipolar Transistor*)

IGBT mengkombinasikan keunggulan-keunggulan dari BJT dan MOSFET. Suatu IGBT mempunyai impedansi masukan yang tinggi dan kehilangan pada saat hidup yang rendah seperti BJT. Akan tetapi tidak memiliki breakdown kedua seperti pada BJT. Dengan merancang struktur chips, resistansi drain ke sumber ekivalent R_{DS} dikendalikan untuk berperilaku seperti transistor.¹

C. Mikrokontroler

Mikrokontroler adalah prosesor mikro yang yang terdiri CPU ditambah dengan RAM, ROM, I/O ports, dan *timer* yang jumlahnya tetap dan dikemas dalam satu chip. Mikrokontroler merupakan salah satu bagian dasar dari suatu sistem komputer. Meskipun mempunyai bentuk yang jauh lebih kecil dari suatu komputer pribadi dan komputer mainframe, mikrokontroler dibangun dari elemen-elemen dasar yang sama. Secara sederhana, komputer akan menghasilkan *output* spesifik berdasarkan *input* yang diterima dan program yang dikerjakan. Seperti umumnya komputer, mikrokontroler adalah alat yang mengerjakan instruksi-instruksi yang diberikan kepadanya. Artinya, bagian terpenting dan utama dari suatu sistem terkomputerisasi adalah program itu sendiri yang dibuat oleh seorang programmer. Program ini menginstruksikan komputer untuk melakukan jalinan yang panjang dari aksi-

¹ Rasid, Muhammad H. 1999. *Elektronika Daya jilid 1*. PT. Prenhallindo. Jakarta.

aksi sederhana untuk melakukan tugas yang lebih kompleks yang diinginkan oleh programmer.

Terdapat perbedaan yang signifikan antara mikrokontroler dan mikroprosesor. Perbedaan utama dapat dilihat dari dua faktor utama yaitu arsitektur perangkat keras dan aplikasi masing-masing. Ditinjau dari segi arsitekturnya, mikroprosesor hanya merupakan *single chip* CPU, sedangkan mikrokontroler dalam IC-nya selain CPU juga terdapat divais lain yang memungkinkan mikrokontroler berfungsi sebagai suatu *single chip computer*. Dalam sebuah IC mikrokontroler telah terdapat ROM, RAM, EPROM, *serial interface* dan *parallel interface*, *timer*, *interrupt controller*, konverter analog ke digital, dan lainnya (bergantung fitur yang melengkapi mikrokontroler tersebut). Sedangkan dari segi aplikasinya, mikroprosesor hanya berfungsi sebagai *central processing unit* yang menjadi otak komputer, sedangkan mikrokontroler, dalam bentuknya yang mungil, pada umumnya ditujukan untuk melakukan tugas-tugas yang berorientasi kontrol pada rangkaian yang membutuhkan jumlah komponen minimum dan biaya rendah.

Mikrokontroler AVR (*Alf and Vegard's Risc processor*) merupakan mikrokontroler keluaran Atmel, yang mempunyai arsitektur RISC 8 bit, di mana semua instruksi dikemas dalam kode 16-bit dan sebagian besar instruksi dieksekusi dalam 1 siklus *clock*, berbeda dengan instruksi MCS51 yang membutuhkan 12 siklus *clock*. Tentu saja itu terjadi karena kedua jenis mikrokontroler tersebut memiliki arsitektur yang berbeda. AVR berteknologi RISC (*reduced Instruction set computing*), sedangkan seri MCS51

berteknologi CISC (*complex Instruction set computing*) Secara umum, AVR dapat dikelompokkan menjadi 4 kelas, yaitu keluarga Attiny, keluarga AT90Sxx, keluarga Atmega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, *peripheral*, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, mereka bisa dikatakan hampir sama [Lingga Wardhana, 2006].

1. ATMega8535

ATMega8535 memiliki fitur-fitur sebagai berikut:

- Saluran I/O sebanyak 32 buah yaitu *port A*, *port B*, *port C*, dan *port D*.
- ADC 10 bit sebanyak 8saluran.
- Tiga buah *Timer/Counter* dengan kemampuan pembanding.
- CPU yang terdiri atas 32 buah register.
- *Watchdog Timer* dengan osilator internal.
- SRAM sebesar 512byte.
- Memori *Flash* sebesar 8 KB dengan kemampuan *Read While Write*.
- Unit interupsi internal dan eksternal.
- *Port* antarmuka SPI.
- EEPROM sebanyak 512 byte yang dapat diprogram saat operasi.
- Antarmuka komparator analog.
- *Port* USART komunikasi serial.

Kapabilitas detail dari ATmega8535 adalah sebagai berikut:

- Sistem mikroprosesor 8 bit berbasis RISC dengan kecepatan maksimal 16 MHz.
- Kapabilitas memori *flash* 8 Kb, SRAM sebesar 512 byte, dan EEPROM sebesar 512 *byte*.
- ADC *internal* dengan fidelitas 10 bit sebanyak 8 *channel*.
- *Port* komunikasi serial (USART) dengan kecepatan 2,5 Mbps.
- Enam pilihan *mode sleep* menghemat penggunaan daya listrik.

Dalam penelitian ini dibutuhkan lebih dari satu buah saluran I/O yang dihubungkan dengan rangkaian konverter DC ke DC. Sehingga mikrokontroler yang digunakan harus mempunyai fitur dan port I/O yang memadai. Atas dasar ini maka digunakan mikrokontroler ATmega8535 sebagai pengendali utama.²

2. Bahasa C

a. Format Penulisan Bahasa C

Untuk mengenal lebih jauh tentang bahasa C, kita terlebih dahulu harus mengenal struktur atau format penulisan program. Sehingga kita memiliki gambaran tentang bagian-bagian dalam pembuatan program.

² Wardhana, Lingga. 2006. *Belajar Sendiri Mikrokontroler Seri ATmega8535*. Penerbit Andi. Yogyakarta.

Bagian Komentar Keterangan Program

Komentar digunakan untuk memberikan keterangan pada program agar mudah dibaca dan akan diabaikan oleh *compiler*.

Cara kerja:

```
/*.....*/ untuk komentar bentuk paragraf
// untuk komentar bentuk per baris (sebelum enter)
```

Bagian preprocessor

Preprocessor `#include` biasanya digunakan untuk menyertakan *file header(.h)* atau *file library*. *File include* berguna untuk memberitahu *compiler* agar membaca file yang di-include-kan lebih dahulu agar mengenali definisi-definisi yang digunakan dalam program agar tidak dianggap *error*.

Cara penulisan:

`#include <.....>` untuk lokasi standar *file* yang telah di-*setting* oleh *tools* biasanya pada *folder include* atau *folder direktori compiler*.

`#include "....."` untuk lokasi file yang kita tentukan sendiri

File header io.h adalah *file* yang berisi segala informasi atau definisi tentang register-register fungsi khusus (SFR) dan bit-bit atau pin-pin mikrokontroler AVR.

Bagian deklarasi variable global

Variable global dideklarasikan di luar semua fungsi termasuk fungsi utama dan letaknya harus di atas. Sifat variabel global yaitu dapat diakses oleh semua pernyataan dalam program.

Bagian *Prototype* Fungsi

Berfungsi untuk mendeklarasikan fungsi yang di bawah fungsi “*main*”. Jika kita membuat fungsi di atas fungsi “*main*” maka kita tidak usah mendeklarasikan fungsi tersebut langsung tulis saja seperti fungsi “*x_pangkat_y*”.

Bagian Fungsi Utama/main

Fungsi utama adalah fungsi pertama yang akan dieksekusi dengan urutan dari atas ke bawah dan akan loncat-loncat bergantung pada instruksi loncatan “*goto*” atau instruksi panggilan fungsi atau terjadi interupsi apabila interupsi diaktifkan.

Bagian Subprogram fungsi fungsi yang telah di-*prototype*-kan ditulis di bawah fungsi main. *Prototype* fungsi berguna untuk memudahkan programmer dalam penulisan program yang besar. Jika kita membuat sebuah fungsi tanpa kita *prototype*kan maka harus ditulis diatas fungsi main dan ini menyulitkan untuk dibaca dan diperbarui sehingga kita butuh *prototype* fungsi.

b. Komentar

Komentar tidak akan dikompilasi oleh *compiler*. Komentar berguna bagi kita untuk membantu programmer mengingat kembali fungsi tiap pernyataan-pernyataan program.

c. Preprocessor

Preprocessor #include

Biasanya digunakan untuk menyertakan *file header* (.h) atau *file library*.

Preprocessor #define

Digunakan untuk mendefinisikan konstanta atau makro.

Preprocessor #if-#endif

Digunakan untuk mengetes ekspresi yang valid untuk mengolah kode program dibawahnya hingga #endif.

Preprocessor #if-#else-#endif

Digunakan untuk mengetes ekspresi yang *valid* untuk mengolah kode program di bawahnya atau jika tidak valid maka kode program di bawah #else hingga #endif.

Preprocessor #ifdef-#endif

Digunakan untuk mencari tahu apakah *indetifief* sudah didefinisikan? Jika ya akan mengeksekusi program di bawahnya.

Preprocessor #ifndef-#endif

Digunakan untuk mencari tahu apakah indetifier sudah didefinisikan? Jika belum maka akan mengeksekusi program dibawahnya.

Preprocessor #undef

Digunakan untuk menghilangkan *identifier* yang telah kita definisikan dengan #define, sehingga dapat didefinisikan ulang.

Pengenal (*inditifier*)

Pengenal digunakan untuk member nama variable, fungsi, konstanta dan lain-lain. Bahasa C bersifat *case sensitive* (huruf capital dan kecil dianggap berbeda). Konstruksi pengenal: huruf angka garis bawah (_). Tiap pengenal bisa menggunakan ketiga hal tersebut dengan catatan tidak boleh diawali dengan angka.

Variable

Variable adalah tempat untuk menyimpan dan mengakses data yang mewakili memori dalam mikrokontroller. Variable harus dideklarasikan dengan tipe data bersama nama variabel yang akan digunakan.. tiap tipe data mempunyai jangkauan bilangan yang dapat disimpan, hal ini akibat dari byte memori yang dipesan dan bentuk bilangan bertanda atau tidak. Misalnya *unsigned char* oleh *compiler* disediakan 1 byte memori ram sehingga hanya bisa menampung bilangan dari 0 sampai dengan 255 sedangkan jika bertanda -128 sampai dengan 127.

Larik (*Array*)

Array adalah variable yang berisi sekumpulan data yang mempunyai tipe data yang sama berbentuk matriks tunggal atau matrik multidimensional.

Operator

Operator adalah karakter-karakter khusus yang digunakan untuk manipulasi variable. Operator aritmatika, penugasan (*assignment*), logika dan bit, relasi, pinter.³

³ Winoto, Ardi. 2008. *Mikrokontroller AVR Atmega8/32/16/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR*. Informatika. Bandung.