PERANCANGAN PERANGKAT LUNAK SISTEM KENDALI KECEPATAN MOTOR ARUS SEARAH PADA KONVEYOR MENGGUNAKAN MIKROKONTROLER AVR ATMEGA 8535

Surya Darma
Dosen Tetap Yayasan Program Studi Teknik Elektro
Fakultas Teknik Universitas Palembang
Email : suryadarma.stmt @ gmail.com

ABSTRAK

Salah satu penerapan sistem kendali dalam industri dapat kita temukan pada sistem konveyor. Konveyor merupakan salah satu peralatan yang banyak digunakan dalam bidang industri. Alat ini ini dapat mengangkut material curah dalam tonase yang besar pada jarak angkut yang lebih besar dan dengan ongkos angkut per ton yang lebih rendah dibandingkan dengan cara pengangkutan lainnya. Konveyor juga digunakan secara luas untuk pengangkutan berkapasitas kecil sampai sedang karena praktis dan ekonomis untuk mengangkut berbagai macam material.

Konveyor membutuhkan suatu perangkat lunak untuk mengendalikan sistem kerjanya, sistem kendali yang baik sangat diperlukan untuk menunjang kelancaran proses produksi dalam suatu industri.

Dalam penelitian ini penulis membuat perancangan perangkat lunak sistem kendali pada konveyor. Perangkat lunak (software) yang digunakan yaitu bahasa pemrograman Bascom AVR untuk memprogram mikrokontroler ATMega 8535 yang digunakan untuk menggerakkan driver motor.

Dari hasil pengujian didapatkan bahwa sistem kendali konveyor dengan menggunakan PWM (Pulse Width Modulation) berjalan dengan cukup baik pada masing – masing suplai feeder, terutama pada suplai feeder 210 gr/dtk dengan persentase kesalahan rata – rata 2,19 %. Pada suplai feeder 128 gr/dtk serta 170 gr/dtk dengan persentase kesalahan rata – rata 4,65 % dan 3,71 %.

I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi dan ilmu pengetahuan yang begitu pesat mempengaruhi segala aspek kehidupan manusia baik itu industri, ekonomi, pendidikan, kesehatan, keamanan dan sebagainya. Dalam bidang industri, sistem pengendalian semakin berperan penting dalam kehidupan manusia terutama dalam pengendalian proses dan pengendalian mesin – mesin produksi.

Salah satu penerapan sistem kendali dalam industri dapat kita temukan pada sistem konveyor. Sistem pengendalian konveyor biasanya mempunyai suatu *setting point* dimana alat ini akan bekerja sesuai dengan *setting point* tersebut, sehingga produksi yang berlangsung dalam industri tersebut akan berjalan dengan lancar. Di pabrik/industri, *setting point* ini biasanya ditetapkan dengan satuan Ton/Jam. Sehingga kecepatan putar motor penggerak *belt* konveyor akan disesuaikan demi mencapai *setting point* tersebut.

Mengingat pentingnya keberadaan alat transportasi ini dalam dunia industri dan semakin berkembangnya alat transportasi ini, maka diperlukanlah suatu sistem kendali yang baik sehingga produksi yang dihasilkan dapat berjalan dengan lancar. Berdasarkan latar belakang itulah, penulis mencoba merancang program mikrokontroler untuk mengendalikan kecepatan motor pada *prototype* konveyor dan penentuan kecepatan konveyornya berdasarkan *setpoint* yang telah ditentukan.

Berkaitan dengan itu aplikasi mikrokontroler dapat dan telah dikembangkan sebagai pengendali pada konveyor, seiring berkembang juga software yang dapat mengisi perintah ke dalam mikrokontroler tersebut. Diantaranya bahasa *Assembler*, bahasa C dan juga bahasa *Basic Compiler* (BASCOM).

Pengendalian yang digunakan dalam konveyor ini memanfaatkan PWM atau *Pulse width modulation* yang menunjukkan konsep dari penghasil pulsa sinyal digital secara cepat untuk mensimulasikan keluaran yang bervariasi. Metode ini lebih utama digunakan untuk menggerakkan motor, pemanas, atau lampu pada intensitas atau kecepatan yang bervariasi.

1.2 Perumusan Masalah

Pada penelitian ini penulis akan merancang program aplikasi mikrokontroler AVR ATmega 8535 dengan menggunakan bahasa pemrograman BASCOM AVR pada proses pengendalian kecepatan motor arus searah pada konveyor. Sehingga nantinya konveyor tersebut akan bergerak sesuai dengan *setting point* yang telah ditentukan.

Pada perancangan konveyor ini akan digunakan inputan-inputan untuk mikrokontroler yaitu sensor berat yang berfungsi sebagai inputan berat dari material yang akan dipasang di bawah *belt* konveyor, pada pengukuran sensor berat ini akan dilakukan pengubahan tegangan dari tegangan analog ke bilangan digital, sensor kecepatan berfungsi untuk mengetahui kecepatan konveyor, *keypad* berfungsi untuk menentukan *setpoint* yang diinginkan yang ditampilkan melalui LCD. Untuk menggerakkan konveyor digunakan motor arus searah, dan pengendalian sistem secara keseluruhan dilaksanakan oleh mikrokontroler AVR ATmega 8535.

1.3 Ruang Lingkup Penelitian

Untuk mempermudah dalam menjelaskan penelitian ini, maka penulis membatasi permasalahan yang ada. Batasan masalah pada penelitian ini adalah :

1. Mikrokontroler yang digunakan sebagai pengatur dari motor arus searah adalah AVR ATMega 8535
2. Bahasa program yang dipakai adalah bahasa BASCOM (*Basic Compiler*) AVR.
3. Pengaturan kecepatan konveyor dengan PWM (*Pulse Width Modulation*).

Untuk perangkat keras atau fisik dari *belt* konveyor itu sendiri tidak dibahas dalam penelitian ini.

1.4 Tujuan Penelitian

Adapun tujuan yang hendak dicapai dalam penulisan penelitian ini yaitu :

1. Mampu merancang perangkat lunak *(software)* yang dapat mengendalikan kecepatan konveyor
2. Mampu merancang dan membuat sistem kendali konveyor berdasarkan berat material yang terukur pada sensor berat.

1.5 Manfaat Penelitian

Manfaat penelitian ini adalah :

Memberikan informasi kepada para pembaca dalam mengembangkan alat transportasi dengan menggunakan suatu sistem kendali yang baik sehingga produksi yang dihasilkan dapat berjalan dengan lancar.

II. TINJAUAN PUSTAKA

2.1 Pengertian Konveyor

Konveyor adalah suatu sistem pengangkutan yang dapat diandalkan untuk mengangkut berbagai macam material, baik material tambang, ataupun bahan – bahan pada industri lainnya.

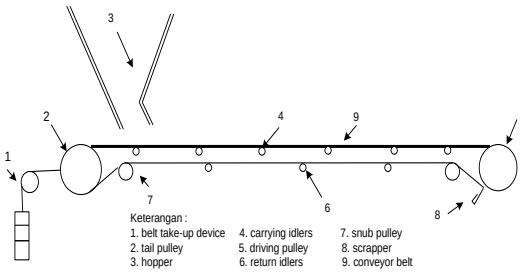
Belt konveyor merupakan jenis konveyor yang paling luas digunakan dibandingkan jenis konveyor lain. *Belt* konveyor ini dapat mengangkut material curah dalam tonase yang besar pada jarak angkut yang lebih besar, dan dengan ongkos angkut per ton yang lebih rendah dibandingkan dengan cara pengangkutan lainnya.

Meskipun demikian *belt* konveyor juga digunakan secara luas untuk pengangkutan berkapasitas kecil sampai sedang karena praktis dan ekonomis untuk mengangkut berbagai macam material. Disamping penggunaan utama sebagai alat transportasi material curah, belt konveyor dapat dimodifikasi untuk melaksanakan berbagai fungsi lainnya seperti penimbangan, *blending, sampling,* dan *stock pilling*.

Instalasi dari sistem *belt* konveyor terdiri dari komponen – komponen utama antara lain adalah :

- *Belt*
- *Terminal pulley*
- *Idler* penyangga belt berupa *"cylendrical metal roller"*
- *Take up (tensioning device)*
- Motor penggerak
- Perlengkapan pelindung dan bangunan penyangga
- *Scrapper*
- *Hopper*

Untuk lebih jelasnya, bagian – bagian pada konveyor dapat ditunjukkan pada gambar 2.1 berikut :



Gambar 2.1 Instalasi Belt Conveyor

2.2 Dasar – Dasar Mikrokontroler

Mikrokontroler merupakan sebuah sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu *chip* IC, sehingga sering disebut *single chip microcomputer*. Mikrokontroler merupakan sebuah sistem komputer yang mempunyai satu atau beberapa tugas yang sangat spesifik, berbeda dengan komputer yang memiliki beragam fungsi. Perbedaan yang lainnya adalah perbandingan RAM dan ROM yang sangat berbeda antara komputer dengan mikrokontroler. Dalam mikrokontroler ROM jauh lebih besar dibanding RAM, sedangkan dalam komputer RAM jauh lebih besar dibanding ROM.

Mikrokontroler biasanya dikelompokkan dalam satu keluarga, masing – masing mikrokontroler memiliki spesifikasi tersendiri namun kompatibel / cocok dalam pemrogramannya. Keluarga AVR merupakan salah satu dari keluarga mikrokontroler, AVR ini dapat dikelompokkan menjadi empat kelas yaitu :

1. Keluarga ATtiny
2. Keluarga AT90Sxx
3. Keluarga AT Mega
4. Keluarga AT86RFxx.

Pada dasarnya yang membedakan masing – masing kelas adalah memori, peripheral, dan fungsinya. Mikrokontroler yang digunakan dalam tugas akhir ini adalah mikrokontroler AVR jenis ATMega 8535, hal ini karena kelebihan yang dimilikinya, diantaranya sudah ada pengubah analog ke Digital (ADC internal) di dalam chip tersebut.

2.2.1 Arsitektur Mikrokontroler ATMega 8535

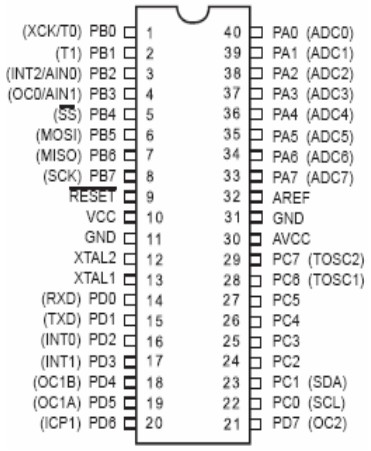
ATmega 8535 merupakan IC CMOS 8-bit yang memiliki daya rendah dalam pengoperasiannya dan berbasis pada arsitektur RISC (Reduced Instruction Set Computing). Mikrokontroler jenis ini dapat mengeksekusi satu instruksi dalam sebuah siklus *clock*, sehingga para perancang dapat mengoptimalkan penggunaan daya rendah dengan kecepatan yang tinggi.

Fitur – fitur yang terdapat pada ATmega 8535 yaitu:

a. 8 Kbyte *In-System Programmable flash* dengan kemampuan membaca ketika menulis.
b. 512 byte EEPROM
c. 512 byte SRAM
d. 32 *general purpose I/O*
e. 32 *general purpose register*
f. 3 buah *Timer/Counter* dengan mode compare
g. *Interrupt* internal dan eksternal
h. USART yang dapat diprogram
i. 8-*channel* ADC 10 bit
j. *Watchdog timer* yang dapat di program dengan osilator internal
k. Sebuah serial port SPI
l. 6 buah *mode power saving* yang dapat dipilih dengan *software*

2.2.2 Konfigurasi Pin

Pin adalah kaki fisik dari sebuah mikrokontroler. Masing-masing *pin* memiliki fungsi dan karakteristik tersendiri yang harus diperhatikan oleh *user*. Pin-pin pada ATmega8535 dengan kemasan 40-pin DIP (*dual in-line package*) ditunjukkan oleh gambar 2.2 berikut :



Gambar 2.2 Pin-pin ATmega8535 kemasan 40-pin

Penjelasan Pin

- VCC : Tegangan supply (5 Volt)
- GND : Ground
- Port A (PA7..PA0) : Port A berfungsi sebagai input analog ke ADC. Port A juga dapat berfungsi sebagai Port I/O 8 bit bidirectional, jika ADC tidak digunakan. Pin port dapat menyediakan resistor pull-up internal (dipilih untuk setiap bit).
- Port B (PB7..PB0) : Port B merupakan port I/O 8 *bidirectional* dengan resistor *pull-up* internal (dipilih untuk setiap bit).
- Port C (PC7..PCO) : Port C merupakan port I/O 8 *bidirectional* dengan resistor *pull-up* internal (dipilih untuk setiap bit)
- Port D (PD7..PDO) : Port D merupakan port I/O 8 *bidirectional* dengan resistor *pull-up* internal (dipilih untuk setiap bit)
- $\overline{\text{RESET}}$: Input *reset*. Level rendah pada pin ini selama lebih dari panjang pulsa minimum akan menghasilkan *reset*, walaupun clock sedang berjalan.
- XTAL1 : *Input* penguat osilator *inverting* dan input pada rangkaian operasi *clock* internal.
- XTAL2 : *Output* dari osilator *inverting*
- AVCC : AVCC adalah pin tegangan supply untuk port A dan ADC. Pin ini harus dihubungkan ke VCC walaupun ADC tidak digunakan. Jika ADC digunakan, maka pin ini harus dihubungkan ke VCC melalui *low pass filter*
- AREF : Adalah pin referensi tegangan analog untuk ADC

2.2.3 Fungsi Alternatif Port

Pengaktifan fungsi alternative port ini diatur oleh register SFIOR dengan menset Bit *PUD (Pull-Up disable)* *:*

a. Port A

Port A memiliki fungsi lain yaitu sebagai :

Tabel 2.1 Fungsi Alternatif Port A

Port Pin	Fungsi Alternatif
PA7	ADC7(ADC input channel 7)
PA6	ADC7(ADC input channel 6)
PA5	ADC7(ADC input channel 5)
PA4	ADC7(ADC input channel 4)
PA3	ADC7(ADC input channel 3)
PA2	ADC7(ADC input channel 2)
PA1	ADC7(ADC input channel 1)
PA0	ADC7(ADC input channel 0)

b. Port B

Port B memiliki fungsi lain yaitu sebagai :

Tabel 2.2 Fungsi Alternatif Port B

Port Pin	Fungsi Alternatif
PB7	SCK (SPI Bus Serial Clock)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB4	SS (SPI Slave Select Input)
PB3	AIN 1 (Analog Comparator Negative Input) OCO (Timer /Counter0 Output Compare Match Output)
PB2	AIN 0 (Analog Comparator Positive Input) INT 2 (External Interrupt 2 input)
PB1	T1 (Timer/Counter1 External Counter Input)

PB0	T0 (Timer/Counter0 External Counter Input) XCK (JSART External Clock Input/Output)

c. Port C

Fungsi lain dari Port C adalah :

Tabel 2.3 Fungsi Alternatif Port C

Port Pin	Fungsi Alternatif
PC7	TOSC2 (Timer Osilator Pin 2)
PC6	TOSC1 (Timer Osilator Pin 1)
PC1	SDA (Two-wire Serial Bus data Input/Output line)
PC0	SCL (Two-wire Serial Bus Clock Line)

d. Port

Fungsi lain dari Port D adalah :

Tabel 2.4 Fungsi Alternatif Port D

Pin Port	Fungsi Alternatif
PD7	OC2 (Timer/Counter2 Output Compare Match Output)
PD6	ICP1 (Timer/Counter1 Input Capture Pin)
PD5	OC1A (Timer/Counter1 Output Compare A Match Output)
PD4	OC1B (Timer/Counter1 Output Compare B Match Output)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

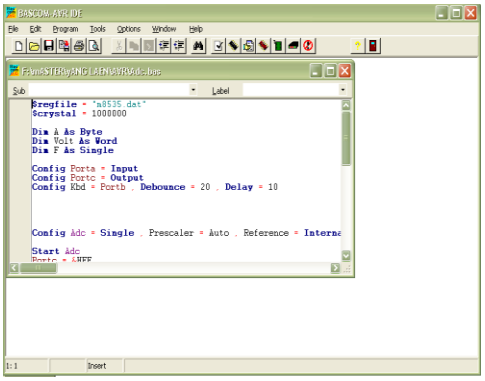
2.2.4 Bahasa Pemrograman Mikrokontroler

Secara umum bahasa yang digunakan untuk pemrograman mikrokontroler adalah bahasa tingkat rendah yaitu bahasa *asembly*. Karena banyaknya hambatan dalam penggunaan bahasa *assembly* ini maka mulai dikembangkan kompiler/penerjemah untuk bahasa tingkat tinggi. Untuk AVR bahasa tingkat tinggi yang banyak dikembangkan antara lain BASIC dan Bahasa C.

Pada perancangan alat kendali ini penulis menggunakan kompiler BASCOM-AVR *(Basic Compiler-AVR).*

2.3 BASCOM AVR

Bascom AVR *(Basic Compiler)* merupakan software dengan menggunakan bahasa basic yang dibuat untuk melakukan pemrograman chip – chip mikrokontroler tertentu, salah satunya ATmega 8535. Interface dari Bascom AVR dapat dlihat pada gambar 2.3 berikut



Gambar 2.3 Interface BASCOM AVR

Keterangan lengkap ikon – ikon dari program BASCOM AVR dapat dlihat pada tabel 2.5 berikut:

Tabel 2.5 Keterangan tombol - tombol program BASCOM AVR

Icon	Nama	Fungsi	Shortcut
	File New	Membuat file baru	Ctrl+N
	Open File	Untuk membuka file	Ctrl+N
	File Close	Untuk menutup program yang di buka	Ctl+O
	File Save	Untuk menyimpan file	Ctrl+S
	Save As	Menyimpan dengan nama yang lain	-
	Print Preview	Untuk melihat tampilan sebelum dicetak	-
	Print	Untuk mencetak dokumen	Ctrl+P
	Exit	Untuk keluar dari program	-
	Program Compile	Untuk mengkompile program yang dibuat, outoutnya bs berupa*.hex,*.bin, dll	F7
	Syntax check	Untuk memeriksa kesalahan bahasa	Ctrl+F7
	Show result	Untuk menampilkan hasil kompilasi program	Ctrl+W

Untuk menu *show result* informasi yang akan ditampilkan berupa :

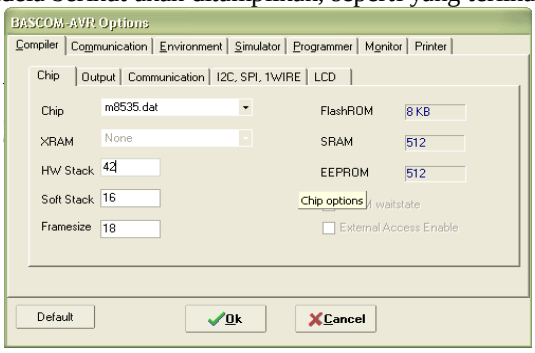
Tabel 2.6 Info hasil kompilasi program

Info	Keterangan
Compiler	Versi dari compiler yang digunakan
Processor	Menampilkan target processor yang dipilih
Date and Time	Tanggal dan waktu kompilasi
Baud Rate and Xtal	Baudrate yang dipilih dan kristal yang digunakan uP.
Error	Error nilai Baud yang di set dengan nilai baud sebenarnya
Flash Used	Persentase Flash ROM yang terisi program
Stack	Lokasi awal stack pointer

Start	memori
RAM Start	Lokasi awal eksternal RAM
LCD Mode	Mode LCD yang digunakan 4 bit atau 8 bit

2.3.1 Compiler

BASCOM AVR menyediakan pilihan untuk memodifikasi pilihan – pilhan pada kompilasi. Dengan memilih menu kompiler maka jendela berikut akan ditampilkan, seperti yang terlihat pada gambar 2.4 berikut :



Gambar 2.4 Jendela Output

Keterangan dari pilihan tersebut adalah sebagai berikut :

Tabel 2.7 Keterangan menu pilihan

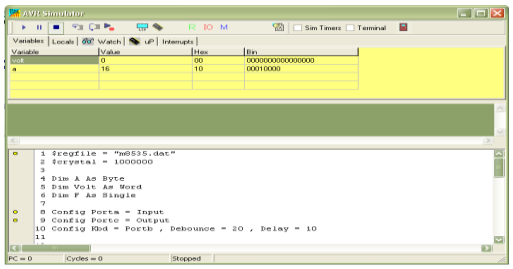
TAB Menu	Option	Keterangan
Chip	Chip	Mikrokontroller yang digunakan, sebagai contoh m8535.dat untuk ATMEGA 8535
	XRAM	Jika menggunakan external RAM nilai ini dapat ditampilkan
	HW Stack	Stack memory hardware, setiap Gosub membutuhkan 2 byte. Jika menggunakan interupsi, naokkan nilainya
	Soft Stack	Stack

		software, nilai defaultnya 8
	FlashROM	Nilai flashROM chip yang dipilih
	SRAM	Nilai RAM internal chip yang dipilih
	EEPROM	Nilai EEPROM chip yang dipilih
Output		File output yang akan dihasilkan dalam proses proses kompilasi
Communication	Baudrate 0	Nilai Baudrate yang digunakan dalam komunikasi serial
	Frekuensi	Nilai osilator yang digunakan
	Error	Error antara baudrate yabg dipilh dengan nilai sebenarnya. Hal ini tergantung dari osilator yang dipilih
I2C,SPI, 1 wire	SDA	Pin yang berfungsi untuk data serial dalam komunikasi I2C
	SCl	Pin yang berfungsi untuk data clock dalam komunikasi I2C
	1Wire	Pin yang digunakan

		untuk komuniksi serial sinkron
	SPI	Pin yang digunakan untuk komuniksi serial sinkron
LCD		Pemilihan port yang digunakan untuk tampilan LCD, jenis LCD

2.3.2 Program Simulasi

BASCOM AVR menyediakan pilihan yang dapat mensimulasikan program. Agar dapat menjalankan simulator ini, file DBG dan OBJ harus dipilh pada menu *Options Compiler Outputs.*
Tampilan program simulasi adalah sebagai berikut :



Gambar 2.5 Interface Simulator BASCOM AVR

Tekan tombol  untuk memulai simulasi. Dan untuk memberhentikan simulasi atau menahan proses simulasi gunakan tombol di sebelahnya. Layar biru ditengah merupakan simulasi layar monitor keetika menggunakan komunikasi serial.

Untuk dapat mengamati perubahan – perubahan nilai register atau variabel selama program berjalan, simulator ini menyediakan beberapa jendela, antara lain :

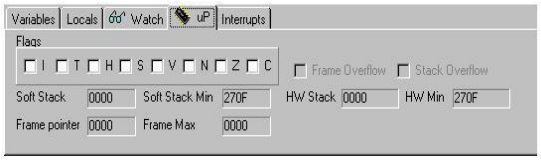
- Variabel

Jendela ini berisi tabel yang berfungsi untuk mengamati nilai – nilai variabel yang digunakan dalam program yang sedang disimulasikan. Untuk menambahkan variabel klik ganda pada kolom variabel maka daftar variabel akan ditampilkan, klik variabel yang ingin diamati.

Gambar 2.6 Jendela Variabel

- uP (Mikroprosessor)

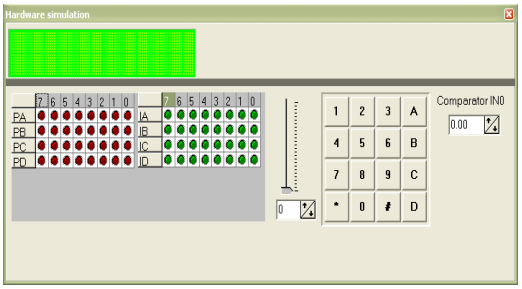
Jendela ini menunjukan status Register (SREG) dari mikroprosessor, software stack, hardware stack dan pointer frame.



Gambar 2.7 Jendela status register mikroprosessor

- Simulasi Hardware

  Untuk dapat melihat perubahan data pada tiap *port* atau ketika kita ingin memberikan input pada pin – pin tertentu dari mikrokontroler, maka gunakan tombol LCD untuk menampilkan jendela seperti yang ditunjukkan pada gambar 2.8 berikut :



Gambar 2.8 Jendela simulasi Hardware

2.4 Dasar Pemograman dalam BASCOM

2.4.1 Karakter dalam 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 2.9 berikut :

Tabel 2.8 Karakter-karakter Spesial pada BASCOM

Karakter	Nama
	Blank atau spasi
‘	Apostrophe
*	Asteriks atau simbol perkalian
+	Simbol Pertambahan (Plus Sign)
,	Comma
-	Simbol Pengurangan (Minus Sign)
.	Period (decimal point)
/	Slash (division symbol) will be handled as \
:	C olon
“	Double Quotation mark
;	Semicolon
<	Less than
=	Equal sign (assigment symbol or relation operator)
>	Greater than
\	Backslash (interger/word division symbol)

2.4.2 Variabel

Variabel dalam sebuah pemrograman berfungsi sebagai tempat penyimpanan data atau penampung data sementara.

Dalam BASCOM ada beberapa aturan dalam penamaan sebuah variabel:

1. Nama variabel maksimum terdiri atas 32 karakter.
2. Karakter biasa berupa angka atau huruf.
3. Nama variabel harus dimulai dengan huruf.
4. Variabel tidak boleh menggunkan kata-kata yang digunakan oleh BASCOM sebagai perintah, pernyataan, internal register dan nama operator (AND, OR, DIM, dan lainnya).

Sebelum variabel digunakan maka variabel tersebut harus dideklarasikan terlebih dahulu, dalam BASCOM ada beberapa cara untuk mendeklarasikan sebuah variabel. Yang pertama dengan menggunakan pernyataan **"DIM"** diikuti nama tipe datanya, contoh pendeklarasikan menggunakan DIM sebagai berikut :

Dim nama **as** byte
Dim tombol1 **as** interger
Dim tombol2 **as** word
Dim Kas **as** string*10

Untuk mempercepat pendeklarasian sebuah variabel yang banyak adalah:

Dim nama **as** byte, tombol1 **as** interger
Dim tombol2 **as** bit, tombol4 **as** word
Dim kas **as** string*10

Cara lain untuk mendeklarasikan sebuah variabel dengan menggunakan **DEFINT, DEFBIT, DEFBYTE** dan / atau **DEFWORD.** Sebagai contoh:

DEFBYTE nama
DEFINT tombol1
DEFWORD tombol2 ; tombol3 ; tombol4

Deklarasi diatas berarti nama tipe datanya adalah byte, tombol1 tipe datanya adalah integer, dan tombol2, tombol3, dan tombol4 tipe datanya adalah word.

2.4.3 Tipe Data

Setiap variabel dalam BASCOM memiliki tipe data yang menunjukkan daya tampung variabel tersebut, hal ini berhubungan dengan penggunaan memori dari mikrokontroller.
Berikut ini adalah tipe data pada BASCOM berikut keterangannya.

Tabel 2.9 Tipe Data pada BASCOM

Tipe Data	Ukuran (Byte)	Range
Bit	1/8	-
Byte	1	0 – 255
Integer	2	(-32,768) – (+32,767)
Word	2	0 – 65535
Long	4	(-2147483648) – (+2147483647)
Single	4	-
String	s/d 254 byte	-

2.4.4 Alias

Dengan menggunakan ALIAS sebuah variabel yang sama dapat diberikan nama yang lain, tujuannya untuk mempermudah proses pemograman biasanya ALIAS digunakan untuk mengganti nama variabel yang telah baku seperti port mikrokontroller.

Tombol1 **alias** PortA.1
Sensor1 **alias** PortA.2
Motor2 **alias** PortA.3

Dengan deklarasi seperti diatas maka perubahan pada sensor1 akan mengubah kondisi dari PortA.2. Selain mengganti nama port ALIAS juga dapat digunakan untuk mengakses bit tertentu dari sebuah variabel yang teleh dideklarasikan.

2.4.5 Konstanta

Dalam BASCOM selain variabel dikenal juga konstanta, konstanta ini juga merupakan variabel. Dengan konstanta, kode program yang dibuat akan lebih mudah dibaca dan dapat mencegah kesalahan penulisan pada program. Dengan konstanta akan lebih mudah menulis phi daripada menulias 3,14159867. Sama seperti variabel agar konstanta ini bisa dikenali oleh program maka harus dideklarasikan terlebih dahulu. Berikut adalah cara pendeklarasikan sebuah konstanta:

Dim A **as Const** 5
Dim B1 **as const** &B1001
Cara lain yang paling mudah:
Const Cbyte = &HF
Const Cint = -1000
Const Csingle = 1.1
Const Cstring = “test”

2.4.6 Array

Dengan array bisa digunakan sekumpulan variabel dengan nama dan tipe yang sama, untuk mengakses variabel tertentu dalam array tersebut harus menggunakan indeks. Indeks ini harus berupa angka dengan tipe data byte, integer atau word. Hal ini berarti nilai maksimum sebuah indeks adalah sebesar 65535.
Proses pendeklarasian sebauh array hampir sama dengan variabel namun perbedaannya kita juga mengikutkan elemennya. Berikut contoh pemakaian array:

```
Dim kelas(10) as byte
Dim C as Integer
For C = 1 to 10
        A(C) = c
        PortB = a(c)
Next
```

Program diatas membuat array dengan nama “kelas” berisi 10 elemen (1-10) dan kemudian seluruh elemennya diisikan dengan nilai c yang berurutan. Untuk membacanya menggunakan indeks dimana elemen tersebut disimpan, pada program diatas elemen-elemen array-nya dikeluarkan ke port B dari mikrokontroller.

2.4.7 Operasi-operasi dalam BASCOM

Operasi-operasi dalam BASCOM bertujuan untuk mendapatkan, memodifikasi atau menggabungkan suatu informasi ke dalam sebuah pernyataan yang akan dibuat sesuai dengan kebutuhan. Adapun operasi-operasi dalam BASCOM-AVR adalah sebagai berikut:

- **Operator Aritmatika**
  Digunakan dalam perhitungan, yang termasuk operator aritmatika ialah + (tambah), - (Kurang), / (bagi), dan * (kali).
- **Operator Relasi**
  Digunakan untuk membandingkan nilai sebuah angka, hasilnya dapat digunakan untuk membuat keputusan dengan program yang dibuat. Yang termasuk operator relasi adalah:

Tabel 2.10 Operator Relasi pada BASCOM

Operator	Relasi	Pernyataan
=	Sama Dengan	X =Y
<>	Tidak	X <> Y

	sama dengan	
<	Lebih kecil dari	X < Y
>	Lebih besar dari	X > Y
<=	Lebih kecil atau sama dengan	X <= Y
>=	Lebih besar atau sama dengan	X >= Y

- **Operator Logika**
  Digunakan untuk menguji sebuah kondisi atau untuk memanipulasi bit dan operasi boelan. Dalam BASCOM ada empat buah operator logika yaitu **AND, OR, NOT dan XOR**. Operator logika ini juga bisa digunakan untuk menguji sebuah byte dengan pola bit tertentu, sebagai contoh:

```
Dim A as Byte
A = 63 and 19
PRINT A
A = 10 or 9
PRINT A
Output
16
11
```

- **Operasi Fungsi**
  Digunakan untuk melengkapi operator yang sederhana

2.4.8 Kontrol Program

Keunggulan sebuah program terletak pada kontrol program ini. Kontrol program merupakan kunci dari kehandalan program yang dibuat termasuk juga pada rule evaluation pada logika samar. Kontrol program dapat mengendalikan alur dari sebuah program dan menentukan apa yang harus dilakukan oleh sebuah program ketika menemukan suatu kondisi tertentu. Kontrol program ini melipuiti kontrol pertimbangan kondisi dan keputusan, kontrol pengulangan serta kontrol alternatif. BASCOM menyediakan beberapa kontrol program yang sering digunakan untuk menguji sebuah kondisi, perulangan dan pertimbangan sebuah keputusan. Berikut ini beberapa kontrol program yang sering digunakan dalam pemrograman dengan BASCOM.

Berikut adalah beberapa kontrol program yang sering digunakan dalam pemograman dengan BASCOM:

1. **IF… THEN**

   Dengan pernyataan ini kita dapat menguji sebuah kondisi tertentu dan kemudian menentukan tindakan yang sesuai dengan kondisi yang diinginkan. Sintak penulisannya adalah sebagai berikut:

   IF <Syarat Kondisi> **THEN** <Pernyataan>

   Sintak diatas digunakan jika hanya ada satu kondisi yang diuji dan hanya melakukan satu tindakan. Jika melakukan lebih dari satu tindakan maka sintaknya harus ditulis sebagai berikut:

```
IF <Syarat kondisi> THEN
    <Pernyataan ke-1>
    <Pernyataan ke-2>
    <Pernyataan ke-n>
END IF
```

2. SELECT… CASE

Perintah ini akan mengeksekusi beberapa blok pernyataan tergantung dari nilai variabelnya. Perintah ini mirip dengan perintah **IF... THEN**, namun perintah ini memiliki kelebihan yaitu kemudahan pada penulisannya. Sintaknya adalah sebagai berikut:

SELECT CASE Variabel
    CASE test1 : statement
    CASE test2 : statement
    CASE ELSE : statement
END SELECT

3. DO… LOOP

Perintah **Do... Loop** digunakan untuk mengulangi sebuah blok pernyataan terus menerus. Untuk membatasi pengulangannya dapat ditambahkan sebuah syarat kondisi agar perulangan berhenti dan perintahnya menjadi **Do... loop Until**. Sintak penggunaan perintah ini adalah sebagai berikut:

Do
<Blok pernyataan>
Loop

Yang menggunakan perintah Do Loop Until

Do
<Blok pernyataan>
Loop Until <syarat kondisi>

4. FOR… NEXT

Perintah ini digunakan untuk mengeksekusi sebuah blok pernyataan secara berulang. Perintah ini hampir sama dengan perintah **Do... Loop,** namun pada perintah **For... Next** ini nilai awal dan akhir perulangan serta tingkat kenaikan atau turunnya bisa ditentukan.

Penggunaannya sebagai berikut:

For var = start **To/Downto** end **[Step value]**
    <Blok pernyataan>
Next

Untuk menaikan nilai perulangan gunakan **To** dan untuk menurunkan gunakan **Downto.** Tingkat kenaikan merupakan pilihan, jadi bisa digunakan ataupun tidak. Jika nilai kenaikan tidak ditentukan maka secara otomatis BASCOM akan menentukan nilainya yaitu 1.

5. EXIT

Perintah ini digunakan untuk keluar secara langsung dari blok program **For... Next, Do... Loop, Sub... Endsub, While... Wend.** Sintak penulisannya adalah sebagai berikut:

Exit [Do] [For] [While] [Sub]

Sintak selanjutnya setelah **EXIT** bisa bermacam-macam tergantung perintah exit itu berada dalam perintah apa. Jika dalam perintah **Do... Loop** maka sintaknya menjadi **Exit Do.**

6. GOSUB

Dengan perintah **GOSUB** program akan melompat ke sebuah label dan akan menjalankan program yang ada dalam rutin tersebut sampai menemui perintah Return. Perintah Return akan mengembalikan program ke titik setelah perintah **Gosub.**

2.5 Analog to Digital Converter (ADC)

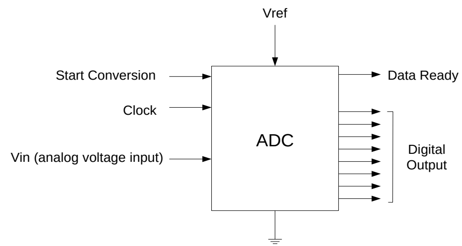
ADC merupakan sebuah sirkuit yang dapat mengubah sebuah tegangan analog menjadi tegangan digital. Sebuah jenis ADC terdiri dari sebuah *Integrated Circuit* (IC) dengan beberapa komponen pendukung.

Pada proses pengubahan ADC, hardware membutuhkan waktu untuk pengubahannya dalam satuan mikro detik. Waktu pengubahan yang dibutuhkan tergantung dari tipe ADC, penerimaan frekuensi clock dan nomor bit yang akan diubah. Gambar 2.10 berikut menunjukkan sebuah diagram blok untuk 8 bit. Masukan Vin dapat berada diantara O V dan Vref. Bila Vin adalah 0 Vdc, keluarannya adalah 00000000. Bila Vin adalah Vref, keluaran adalah 11111111 (255 desimal). Untuk tegangan masukan antara 0 V & Vref peningkatan output linear dengan Vin. Dengan demikian dapat dituliskan hubungan antara tegangan input dengan output digitalnya adalah :

$$\frac{Output}{V_{in}} = \frac{255}{Vref}$$

(Untuk 8 bit)

$$\text{Output} = \frac{V_{in} x255}{Vref} \quad \text{.. (2.1)}$$



Gambar 2.9 Blok Diagram sebuah Analog to-digital converter (ADC)

2.6 Sensor

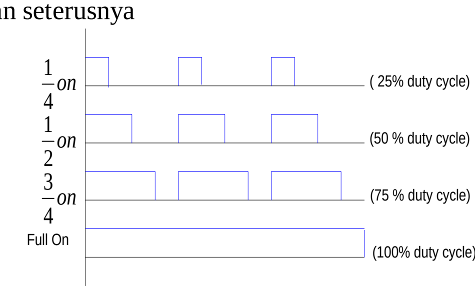
Sensor adalah alat untuk mendeteksi / mengukur sesuatu yang digunakan untuk mengubah variasi mekanis, magnetis, panas, sinar dan kimia menjadi tegangan dan arus listrik. Sensor itu sendiri terdiri dari *transduser* dengan atau tanpa penguat/pengolah sinyal yang terbentuk dalam satu sistem pengindera. Dalam lingkungan sistem pengendali dan robotika, sensor memberikan kesamaan yang menyerupai mata, pendengaran, hidung, lidah yang kemudian akan diolah oleh kontroler sebagai otaknya.

Agar sensor dapat bekerja lebih baik dan tepat haruslah memiliki persyaratan sebagai berikut :

1. Kepekaan, yaitu sensor harus dipilih sedemikian rupa pada nilai-nilai masukan yang ada dapat diperoleh keluaran yang cukup besar
2. Stabilitas waktu, yaitu untuk menentukan masukan tertentu, sensor harus dapat memberikan keluaran yang tetap nilainya dalam waktu yang lama.

2.7 Pulse Width Modulation (PWM)

Pulse width modulation atau modulasi lebar pulsa merupakan sebuah pendekatan untuk mengendalikan torsi dan kecepatan motor arus searah. Tenaga yang disuplai ke motor adalah dalam bentuk sinyal gelombang kotak dari magnet konstan dengan merubah lebar pulsa atau *duty cycle*. Pada gambar 2.11 berikut menunjukkan bentuk gelombang untuk 4 kecepatan yang berbeda. Untuk kecepatan yang paling rendah, tenaga disuplai hanya untuk seperempat dari waktu cycle (*duty cycle of* 25%). Untuk 50% *duty cycle* (hidup pada setengah waktu), motor akan bergiliran pada setengah kecepatan dan seterusnya



Gambar 2.10 Bentuk gelombang PWM.

Duty cycle adalah lamanya waktu terbentuknya pulsa tinggi dalam satu periode

$$duty\,cycle = \frac{t_{high}}{t_{high} + t_{low}} x100\% \quad \text{..(2-2)}$$

Duty cycle menggambarkan pulsa keluaran. Ia akan memberikan nilai dalam bentuk persentase. Persentase akan menunjukkan berapa persen cycle keluaran pada posisi tinggi. Misalnya nilai *duty cycle* 80 %, nilai t_{high} adalah 80% dari T dan t_{Low} adalah 20%.

2.7.1 PWM pada Mikrokontroler ATMega 8535

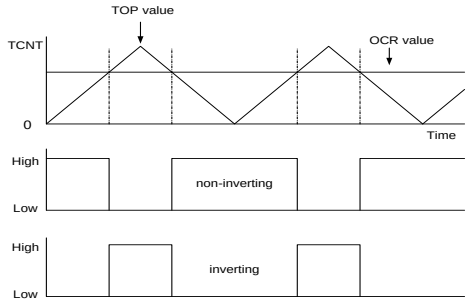
PWM adalah salah satu keunggulan Timer/Counter pada ATMega8535. Ketiga jenis Timer/Counter pada ATMega8535 yaitu Timer/Counter 0, Timer Counter 1 dan Timer/Counter 2 dapat menghasilkan pulsa PWM

Untuk memahami penggunaan PWM, diambil contoh pemakaian Timer/Counter 0 sebagai PWM. PWM adalah Timer mode Output Compare yang canggih. Mode PWM Timer juga dapat mencacah turun yang berlawanan dengan mode Timer lainnya yang hanya mencacah naik. Pada mode PWM tersebut, Timer mencacah naik hingga mencapai nilai TOP, yaitu 0xFF untuk PWM 8 bit. Sebagai penggunaan mode PWM Timer/Counter 0, keluaran sinyal PWM terletak pada pin OCO. Ketika nilai TCNT0 sama dengan nilai OCRO, maka output pada OCO akan berlogika nol atau satu, tergantung pada mode pemilihan PWM. Pemilihan mode PWM disetting melalui bit COM01 dan bit COM00 pada register TCCR0 yang konfigurasinya seperti tabel berikut :

Tabel 2.11 Konfigurasi Bit COM01 dan COM00 Compare Output Mode Phase Correct PWM

COM01	COM00	Description
0	0	Normal port operation, 0C0 disconnected
0	1	Reserved
1	0	Clear 0C0 on Compare Match when up-counting. Set 0C0 on Compare Match when down-counting.
1	1	Set 0C0 on Compare Match when up-counting. Clear 0C0 on Compare Match when down-counting.

Dari tabel diatas dapat diketahui saat COM00 clear dan COM01 set, pin OCO clear saat timer mencacah diatas Compare Match dan pin OCO set saat timer mencacah dibawah Compare Match atau *non-inverting* PWM. Kebalikannya, saat COM00 set dan COM01 set, maka pin OCO set saat timer mencacah diatas compare Match dan pin OCO clear saat timer mencacah di bawah Compare Match atau disebut juga *inverting* PWM. Agar lebih jelas dapat dilihat pada gambar 2.12 berikut :

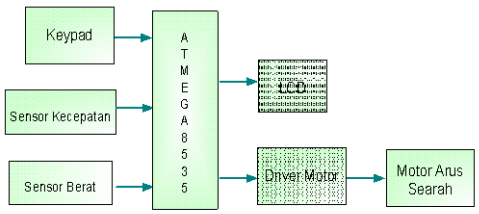


Gambar 2.11 Pulsa PWM Inverting dan Non-Inverting

III. PERANCANGAN PERANGKAT LUNAK

3.1 Perancangan Sistem

Sistem kerja konveyor pengangkut bahan material ini secara umum dapat dilihat pada gambar 3.1 berikut :

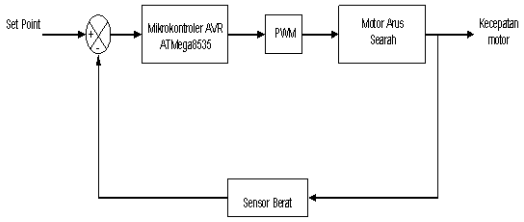


Gambar 3.1 Sistem kerja konveyor

Pada perancangan perangkat lunak sistem kendali kecepatan motor arus searah pada konveyor ini digunakan :

1. Mikrokontroler AVR ATMega 8535, hal ini di karenakan pada mikrokontroler jenis ini sudah tersedia internal PWM dan 8 channel ADC 10 bit yang dibutuhkan dalam pengendalian kecepatan motor arus searah pada konveyor berdasarkan berat material yang terukur pada sensor berat.
2. Sebagai inputan digunakan keypad untuk menentukan setpoint, sensor kecepatan untuk mengetahui kecepatan motor dan sensor berat untuk mendeteksi berat material pada *belt.*
3. ADC *(Analog to Digital Converter)* digunakan agar mikrokontroler dapat membaca berat material pada sensor berat yaitu dengan mengubah tegangan
   analog menjadi bilangan digital yang akan tampil pada LCD.
4. Pengendalian yang digunakan menggunakan PWM yang menunjukkan konsep dari penghasil pulsa sinyal digital secara cepat untuk mensimulasikan keluaran yang bervariasi.

Gambar 3.2 berikut menunjukkan diagram blok sistem kendali konveyor :



Gambar 3.2 Diagram blok sistem

Sistem pengendalian ini akan terdapat 2 inputan, yaitu :

1. Kecepatan dengan satuan cm/dtk
2. Berat dengan satuan gr/cm

Dengan pengubahan satuan, maka akan didapatkan suatu *setting point* yang ditetapkan dengan satuan gr/dtk

Konveyor akan bergerak sesuai dengan *setting point* tersebut, mikrokontroler akan membandingkan antara setpoint yang ditentukan dengan setpoint berdasarkan masukan sensor berat, bila *setpoint* yang ditentukan lebih besar dari yang terbaca di mikrokontroler, maka mikrokontroler akan mempercepat motor arus searah dengan pengontrolan PWM untuk mencapai *setpoint* yang ditentukan tersebut dan sebaliknya bila *setpoint* yang ditentukan lebih kecil dari yang terbaca di mikrokontroler, maka mikrokontroler akan memperlambat motor arus searah sehingga *setpoint* yang ditentukan dapat dicapai.

3.2 Perancangan Perangkat Lunak

3.2.1 Pemograman Mikrokontroler

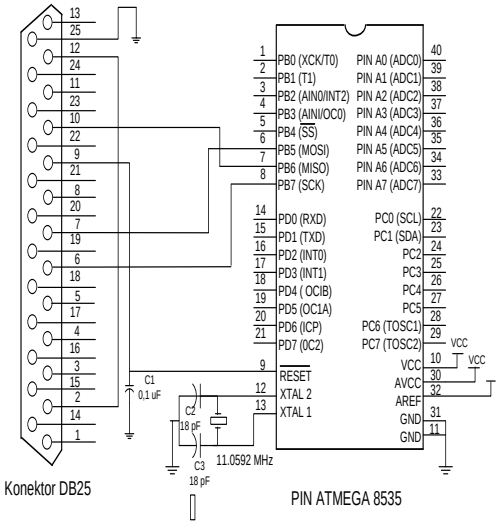
Program pada suatu Mikrokontroler yang disimpan dalam PEROM atau EPROM adalah bahasa mesin, yang berupa kode-kode instruksi. Kode-kode tersebut ditulis dalam sistem bentuk hexadesimal.

Salah satu keunggulan mikrokontroler ATMega8535 adalah kemudahan dalam men-*downlad*-kan program (pemrograman flash memori) yang bisa dilakukan secara langsung pada sistem, yaitu dengan memanfaatkan pin – pin pada mikrokontroler. Ada dua mode pemrograman flash memori (Memori data/ROM) yaitu mode paralel dan mode serial. Pemrograman flash memori pada tugas akhir ini dilakukan dengan menggunakan paralel mode.

Tabel 3.1 Koneksi pin pemrograman Mikrokontroler

Pin DB25 Paralel Port	Pin Mikrokontroler
Pin 6	Pin 8
Pin 7	Pin 6
Pin 9	Pin 9
Pin 10	Pin 7
Pin 25	Pin 31 (Ground)

Untuk lebih jelasnya, rangkaian downloader ATMega 8535 dapat dilihat pada gambar 3.3 berikut :



Gambar 3.3 Rangkaian Downloader ATMega 8535

Keterangan pin :

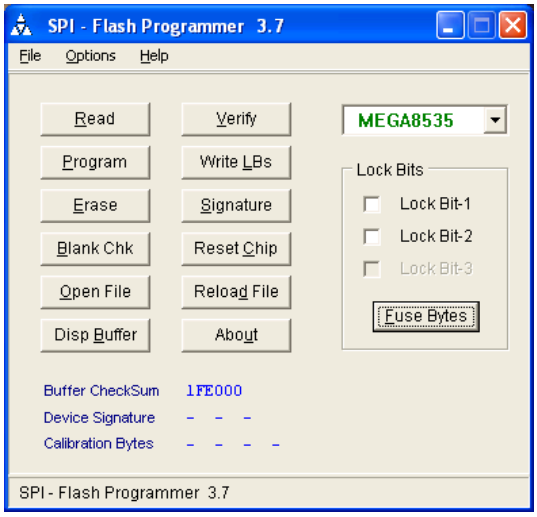
- Pin 6 : MOSI (Master Out Slave In), jalur data serial dari PC ke chip
- Pin 7 : MISO (Master In Slave Out), jalur data serial dari chip ke PC
- Pin 8 : SCK (Serial Clock) : detak yang mengatur aliran data
- Pin 9 : reset

Ketika proses download dilakukan, chip harus diberi supply tegangan 5 volt dan koneksi ground yang baik. Pemograman memanfaatkan keadaan pada saat RESET status hi. Pada awal proses *download*, pin RST diset *high*, kemudian instruksi *Programming Enable* dieksekusi terlebih dahulu sebelum operasi yang lain dapat dilanjutkan.

3.2.2 Program Downloader

Program (Software) downloader merupakan jenis perangkat lunak di computer yang digunakan untuk memasukkan perintah yang dibuat dengan bahasa BASCOM dari computer ke mikrokontroler. Dalam tugas akhir ini jenis program downloader yang digunakan, yaitu SPI PGM 3.7.

Software ini digunakan untuk menuliskan program ke dalam mikrokontroler ATMEGA8535. Software ini merupakan software freeware yang dibuat dan dikembangkan oleh Muhammad Asim Khan dari Pakistan. Tampilan programnya diperlihatkan oleh gambar 3.4 berikut:



Gambar 3.4 Software ISP-Flash Programer 3.7

- Read : Berfungsi untuk membaca software yang sudah ada pada mikrokontroler.
- Program : Berfungsi untuk memasukkan program dalam file.hex kedalam chip memori mikrokontroler.
- Erase : Berfungsi untuk menghapus semua alamat yang ada pada memori mikrokontroler.
- Blank Chk : Berfungsi Untuk menguji kosong atau tidaknya memori pada Mikrokontroler.
- Open File : Berfungsi untuk membuka dan mencari file .bas yang dibutuhkan untuk memprogram mikrokontroler.
- Disp Buffer : Menunjukkan isi dari program yang ditampilkan dalam bentuk pengalamatan.
- Verify : Berfungsi untuk memverifikasi program yang sudah ada pada memori Mikrokontroler.
- Signature : Berfungsi untuk menguji koneksi/hubungan antara mikrokontroler sama komputer (telah atau belum terhubung).
- Reset Chip : Berfungsi untuk mereset chip mikrokontroler.
- Reload File : Berfungsi untuk mereload (membuka kembali file .bas terakhir yang dibuka) secara cepat/instan.
- About : Keterangan tentang seri program dan pembuatnya.
- Fuse Bytes : Berfungsi untuk menetukan nilai oscilator (Xtal) yang akan digunakan pada Mikrokontroler.
- MEGA8535 : Merupakan menu untuk memilih chip mikrokontroler yang digunakan (jenisAVR, yaitu AT90Sxx, ATTiny, dan ATMega).

3.3 Pengalamatan input dan output

Pada sistem kendali konveyor ini digunakan 3 buah input dan 2 buah output. Tabel 3.2 menunjukkan pengalamatan input dan output yang digunakan.

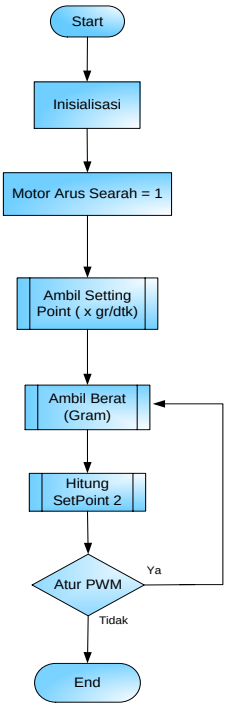
Tabel 3.2 Pengalamatan input dan output

Input	Port
Sensor Berat	PA.0
Sensor kecepatan	PB.0
Keypad	PC.0 - PC.7
Output	**Port**
Driver motor	PD.5
LCD	PA.2 - PA.7

3.4 Perancangan program utama

Pada perancangan perangkat lunak ini akan dibuat dua jenis program, yaitu program utama dan program rutin/subprogram

Program utama merupakan program yang dipakai untuk mengendalikan konveyor secara keseluruhan sedangkan program rutin atau cabang merupakan program yang dibuat spesifik untuk masing – masing sensor yaitu untuk sensor berat dan sensor kecepatan, pemrograman keypad serta pemrograman PWM (pulse width modulation). Gambar 3.5 berikut menunjukkan flowchart program utama pengendalian konveyor:

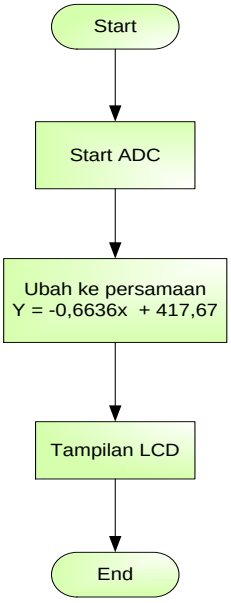


Gambar 3.5 Diagram alir sistem kendali konveyor

3.5 Program rutin

3.5.1 Diagram alir pembacaan sensor berat

Pengukuran berat material pada sensor berat menghasilkan tegangan yang merupakan bilangan analog, agar dapat terbaca pada mikrokontroler, maka tegangan ini harus diubah terlebih dahulu menjadi bilangan digital sesuai persamaan 2.1 (perhitungan terlampir). Gambar 3.6 berikut menunjukkan flowchart untuk pembacaan sensor berat :

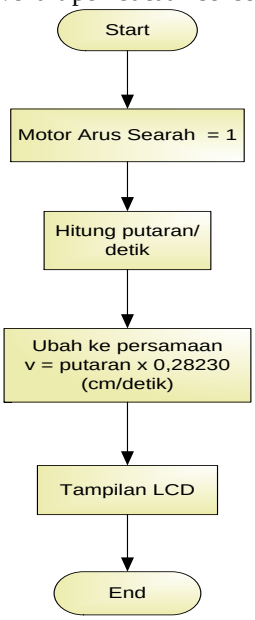


Gambar 3.6 Diagram alir pembacaan sensor berat

3.5.2 Diagram alir pembacaan sensor kecepatan

Untuk membaca kecepatan *belt* konveyor maka perlu dicari terlebih dahulu nilai konstanta perkalian yang akan dimasukkan ke dalam *software* Bascom (perhitungan terlampir).

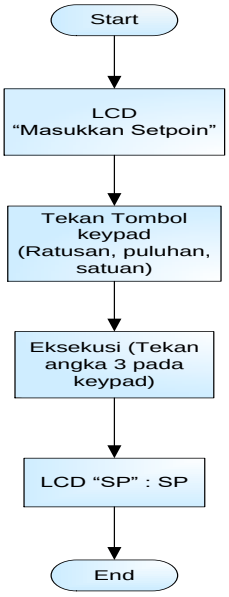
Gambar 3.7 berikut menunjukkan flowchart pembacaan sensor kecepatan :



Gambar 3.7 Diagram alir pembacaan sensor kecepatan

3.5.3 Diagram alir *setting point*

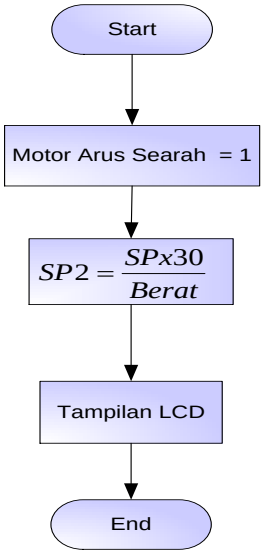
Untuk menentukan setting point pada konveyor, maka dilakukan terlebih dahulu pembuatan diagram alir untuk pemrograman pada keypad sesuai pada gambar 3.8 berikut :



Gambar 3.8 Diagram alir penentuan setting point

3.5.4 Diagram alir perhitungan perbandingan *setpoint (Setpoint 2)*

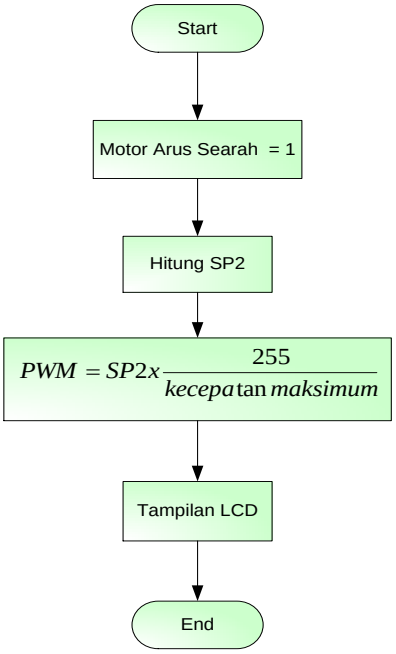
Setpoint 2 merupakan hubungan antara setpoint yang ditetapkan dengan berat yang terbaca per satuan panjang pada sensor berat. Untuk lebih jelasnya, diagram alir perhitungan setpoint 2 dapat dilihat pada gambar 3.9 berikut :



Gambar 3.9 Diagram alir perhitungan setpoint 2

3.5.5 Pemrograman kendali PWM

Untuk menentukan PWM yang digunakan, terlebih dahulu ditentukan kecepatan maksimum motor (dalam satuan cm/dtk). Setelah didapatkan, maka penentuan PWM yang digunakan dapat diketahui dengan mengalikan setpoint 2 dengan pembagian antara PWM maksimum dengan kecepatan maksimum motor. Gambar 3.10 menunjukkan flowchart kendali PWM :



Gambar 3.10 Diagram alir kendali PWM

3.6 Listing Program

3.6.1 Program sensor kecepatan

```
$regfile = "m8535.dat"
$crystal = 8000000
```

```
Dim A As Byte
Dim Volt As Word , V As Single
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db7 = Porta.7 , Db6 = Porta.6 , Db5 = Porta.5,
Config Lcdpin = Pin , Db4 = Porta.4 , E = Porta.3 , Rs = Porta.2
Config Timer1 = Pwm , Pwm = 8 , Compare A Pwm = Clear Down , Prescale = 256
Cursor Off
Config Timer0 = Counter , Edge = Falling
Portb.0 = 1
Tcnt0 = 0

Do
Tcnt0 = 0
Pwm1a = 128
Wait 1
Volt = Tcnt0
V = Volt * 0.28232
Locate 1 , 1
Lcd Volt
Locate 2 , 1
Lcd V
Loop
End
```

3.6.2 Program sensor berat

Listing untuk pembacaan nilai digital pada kecepatan penuh dan kecepatan setengah (dengan PWM 50%) sebagai berikut :

```
$regfile = "m8535.dat"
$crystal = 8000000
Cls

Dim A As Byte
Dim Volt As Word , Gram As Single

Config Lcd = 16 * 2
Config Lcdpin = Pin , Db7 = Portc.5 , Db6 = Portc.4 , Db5 = Portc.3,
Config Lcdpin = Pin , Db4 = Port C.2 , E = Portc.1 , Rs = Portc.0
Cursor Off
Config Porta.0 = Input

Config Adc = Single , Prescaler = Auto , Reference = Internal
Main:
'Memulai mengambil bilangan digital
Start Adc
Waitms 1
Volt = Getadc(0)
Stop Adc

Waitms 10
Locate 1 , 2
Lcd Volt
Wait 2
Cls
Goto Main
```

Setelah dilakukan uji coba linearitas sensor berat, maka akan didapatkan suatu persamaan untuk menentukan berat material dalam satuan gram, sehingga listing program pembacaan sensor berat menjadi :

```
$regfile = "m8535.dat"
$crystal = 8000000

Dim A As Byte
Dim Adc1 As Word , Adc2 As Word , Volt As Word , Gram As Single

Config Lcd = 16 * 2
Config Lcdpin = Pin , Db7 = Porta.7 , Db6 = Porta.6 , Db5 = Porta.5,
Config Lcdpin = Pin , Db4 = Porta.4 , E = Porta.3 , Rs = Porta.2
Config Porta.0 = Input
Cls
Config Adc = Single , Prescaler = Auto , Reference = Internal
Main:
Start Adc
Waitms 1
Volt = Getadc(0)
Stop Adc
Waitms 10
Locate 1 , 2
Gram = 0,6636 * Volt                ‘Konversi bilangan digital ke berat
Gram = 417,97 - Gram
Lcd Gram
Wait 2
Cls
Goto Main
```

3.6.3 Program penentuan setpoint

Sebelum digunakan, keypad terlebih dahulu dikalibrasi sesuai dengan listing program berikut :

```
$regfile = "m8535.dat"
$crystal = 8000000

Dim A As Byte
Cls
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db7 = Porta.7 , Db6 = Porta.6 , Db5 = Porta.5,
Config Lcdpin = Pin , Db4 = Porta.4 , E = Porta.3 , Rs = Porta.2
Config Kbd = Portc ,
Cursor Off

Do
A = Getkbd()
Locate 1 , 1
Lcd A
Loop
End
```

Setelah kalibrasi selesai, maka keypad dapat digunakan untuk menekan tombol penentuan *setting point* sesuai listing program berikut :

```
$regfile = "m8535.dat"
$crystal = 8000000

Dim A As Byte , Pul As Byte , Rs As Byte , Sat As Byte , Sp As Word
```

```
Dim Volt As Word , V As Byte , Nilai As Byte

Cls
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db7 = Porta.7 , Db6 = Porta.6 , Db5 = Porta.5,
Config Lcdpin = Pin , Db4 = Porta.4 , E = Porta.3 , Rs = Porta.2
Config Kbd = Portc ,

Cursor Off

Do
Locate 1 , 1
Lcd "Masukan Setpoint"
Cursor Blink
Locate 2 , 1
Do
A = Getkbd()
Gosub Ubah
Rs = Nilai
Loop Until A <> 16
Lcd Rs
Wait 1
Do
A = Getkbd()
Gosub Ubah
Pul = Nilai
Loop Until A <> 16
Lcd Pul
Wait 1
Do
A = Getkbd()
Gosub Ubah
Sat = Nilai
Loop Until A <> 16
Lcd Sat
Wait 1
Cursor Noblink
Do
A = Getkbd()

Loop Until A = 3
Sp = Rs * 100
Pul = Pul * 10
Pul = Pul + Sat
Sp = Sp + Pul
Cls
Locate 1 , 1
Lcd "SP:" ; Sp
Wait 1
Loop
End

Ubah:
Select Case A
```

```
Case 0 : Nilai = 1
Case 4 : Nilai = 2
Case 8 : Nilai = 3
Case 1 : Nilai = 4
Case 5 : Nilai = 5
Case 9 : Nilai = 6
Case 2 : Nilai = 7
Case 6 : Nilai = 8
Case 10 : Nilai = 9
Case 7 : Nilai = 0
Case 11 : Nilai = "#"
Case 3 : Nilai = "*"
Case 12 : Nilai = "A"
Case 13 : Nilai = "B"
Case 14 : Nilai = "C"
Case 15 : Nilai = "D"
End Select
Return
End
```

3.6.4 Program keseluruhan

```
$regfile = "m8535.dat"
$crystal = 8000000

Dim A As Byte , Pul As Byte , Rs As Byte , Sat As Byte , Sp As Word , Sp2 As Word
Dim Sped As Word , V As Single , Nilai As Byte , Error As Integer
Dim Adc1 As Word , Adc2 As Word , Volt2 As Word , Gram2 As Single , Pwm3 As Word
Dim Speed As Word , Volt As Word , Gram As Word

Cls
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db7 = Porta.7 , Db6 = Porta.6 , Db5 = Porta.5,
Config Lcdpin = Pin , Db4 = Porta.4 , E = Porta.3 , Rs = Porta.2
Config Kbd = Portc ,
Config Timer1 = Pwm , Pwm = 8 , Compare A Pwm = Clear Down , Prescale = 256
Config Pina.0 = Input
Cursor Off
Config Timer0 = Counter , Edge = Falling
Config Adc = Single , Prescaler = Auto , Reference = Internal
Portb.0 = 1
Tcnt0 = 0
Cursor Off

Gosub Mula_mula
Wait 2
Gosub Ambil_sp
Wait 1
Do
Gosub Ukur_berat
Gosub Hitung_sp2
Gosub Ukur_speed
Gosub Kendali
Loop

'Perintah untuk memulai proses
```

```
Mula_mula:
Lcd "press D to Start"
Do
A = Getkbd()
Loop Until A = 15
Pwm1a = 192
Return

‘Perintah untuk memulai memasukkan setpoint
Ambil_sp:
Locate 1 , 1
Lcd "Masukkan SetPoint"
Cursor Blink
Locate 2 , 1
Waitms 500
Do
A = Getkbd()
Gosub Ubah
Rs = Nilai
Loop Until A <> 16
Lcd Rs
Waitms 500
Do
A = Getkbd()
Gosub Ubah
Pul = Nilai
Loop Until A <> 16
Lcd Pul
Waitms 500
Do
A = Getkbd()
Gosub Ubah
Sat = Nilai
Loop Until A <> 16
Lcd Sat
Waitms 500
Cursor Noblink
Do
A = Getkbd()
Loop Until A = 3
Sp = Rs * 100
Pul = Pul * 10
Pul = Pul + Sat
Sp = Sp + Pul
Cls
Locate 1 , 1
Lcd "SP:" ; Sp
Return

‘Perintah untuk mengendalikan kecepatan konveyor
Kendali:
Pwm3 = Sp2 * 4.4
If Pwm3 <= 100 Then Pwm3 = 100
If Pwm3 >= 255 Then Pwm3 = 255
Pwm1a = Pwm3
```

```
Waitms 200
Locate 2 , 12
Lcd Pwm3
Return

‘Perhitungan perbandingan setpoint
Hitung_sp2:
Sp2 = Sp * 30
Sp2 = Sp2 / Gram
Return

‘Perhitungan kecepatan belt konveyor
Ukur_speed:
Tcnt0 = 0
Wait 1
Sped = Tcnt0
V = Sped * 0.28232
Speed = V
Locate 1 , 9
Lcd "v:" ; Speed ; "cm/s"

‘Pengukuran berat material
Ukur_berat:
Start Adc
Waitms 1
Adc1 = Getadc(0)
Stop Adc
Waitms 10
Gram2 = Adc1 * 0.6636
Gram = 417,67 - Gram2
If Gram > 64000 Then Gram = 0
Locate 2 , 1
Lcd "B: " ; Gram ; "Gr"
Return
Ubah:
Select Case A
Case 0 : Nilai = 1
Case 4 : Nilai = 2
Case 8 : Nilai = 3
Case 1 : Nilai = 4
Case 5 : Nilai = 5
Case 9 : Nilai = 6
Case 2 : Nilai = 7
Case 6 : Nilai = 8
Case 10 : Nilai = 9
Case 7 : Nilai = 0
Case 11 : Nilai = "#"
Case 3 : Nilai = "*"
Case 12 : Nilai = "A"
Case 13 : Nilai = "B"
Case 14 : Nilai = "C"
Case 15 : Nilai = "D"
End Select
Return
```

IV. PENGUJIAN DAN ANALISA SOFTWARE

4.1 Umum

Suatu sistem harus dilakukan pengujian sebelum di aplikasikan. Pengujian dilakukan dengan tujuan untuk menguji spesifikasi yang telah dibuat sehingga dapat dilakukan perbaikan apabila hasil yang diperoleh tidak sesuai dengan spesifikasi yang diharapkan. Pengujian yang dilakukan terdiri dari :

1. Pengujian sistem minimum
2. Pengujian suplai *feeder*
3. Pengujian keseluruhan

4.2 Pengujian Sistem Minimum ATMega 8535

Untuk mengetahui apakah rangkaian mikrokontroler berfungsi dengan baik atau tidak, maka digunakan pengujian pada minimum sistemnya. Proses pengujian sistem minimum ini dilakukan dengan menjalankan program SPI-Flash program. Dengan menjalankan SPI-Flash program, kita dapat mengetahui respon dari mikrokontroler ketika meng-klik *command"signature"*. Setelah mikrokontroler dapat direspon oleh ISP-Flash program, kita telah dapat memulai mendownload program ke rangkaian minimum sistem ATMega 8535.

Gambar 4.1 menunjukkan tampilan hasil pengujian ATMega 8535 menggunakan SPI *Flash programmer.*

Gambar 4.1 Pengujian ATMEGA 8535 menggunakan SPI Flash Programmer

4.3 Pengujian suplai *feeder*

Feeder sebagai penyuplai material ke belt konveyor mempunyai satuan berat per satuan waktu. Berdasarkan uji coba dengan waktu (t = 5 detik), didapatkan besarnya suplai *feeder* seperti yang ditunjukkan pada tabel 4.1 berikut :

Tabel 4.1 Pengujian suplai *feeder*

Bukaan garis pada penutup feeder	Percobaan 1	Percobaan 2	Percobaan 3	Rata-rata (Gram)	t (s)	Gram/dtk
3	645	635	640	640	5	128
4	860	850	855	856,67	5	171
5,5	1200	1200	1200	1200	5	240

4.4 Pengujian keseluruhan

Setpoint memegang peranan penting dalam pengendalian konveyor ini, setelah nilai setpoint ditentukan, maka konveyor akan berusaha mencapai setpoint tersebut berdasarkan masukan sensor berat sehingga PWM akan bekerja sesuai dengan rumus yang telah ditentukan sebelumnya.

Sebelum menentukan nilai setpoint, terlebih dahulu harus dicari berat material yang terukur pada sepanjang sensor berat pada masing – masing suplai *feeder*. Pengujian dilakukan dengan menghidupkan motor terlebih dahulu, kemudian penutup pada *feeder* dibuka sesuai dengan yang diinginkan, setelah material berjalan sepanjang *belt*, motor dimatikan, kemudian ukur berat material sepanjang sensor berat (30 cm), hasilnya seperti yang ditunjukkan pada tabel 4.2 berikut :

Tabel 4.2 Pengukuran berat material per satuan panjang sensor berat

Bukaan garis pada penutup feeder	Suplai *feeder* (Gr/dtk)	Panjang pengukuran Sensor berat (cm)	Berat material (gram)
3	128	30	70
4	171	30	90
5	210	30	115

Nilai *setpoint* mempunyai batasan sesuai dengan suplai *feeder* masing – masing, dengan asumsi kecepatan maksimum $\pm$ 58 cm/dtk dan kecepatan setengah penuh (PWM 50%) adalah $\pm$ 30 cm/dtk, maka perhitungan batasan nilai *setpoint* tersebut adalah sebagai berikut :

a. Suplai *feeder* 128 gram/detik

- Setpoint minimum

  $= \frac{70\ gram}{30\ cm}\ x\ 29\ cm/dtk$

  = 67,67 gr/dtk
- Setpoint maksimum

$$= \frac{70\ gram}{30\ cm}\ x\ 58\ cm/dtk$$

= 135,3 gr/dtk

b. Suplai *feeder* 171 gram/detik

- Setpoint minimum

$$= \frac{90\ gram}{30\ cm}\ x\ 29\ cm/dtk$$

= 87 gr/dtk

- Setpoint maksimum

$$= \frac{90\ gram}{30\ cm}\ x\ 58\ cm/dtk$$

= 174 gr/dtk

b. Suplai *feeder* 210 gram/detik

- Setpoint minimum

$$= \frac{115\ gram}{30\ cm}\ x\ 29\ cm/dtk$$

= 111,2 gr/dtk

- Setpoint maksimum

$$= \frac{115\ gram}{30\ cm}\ x\ 58\ cm/dtk$$

= 222,3 gr/dtk

Setelah *range setpoint* didapatkan, maka dalam penentuan nilai masukkan *setpoint*nya pada masing – masing suplai *feeder* harus berada pada *range*/batasannya. Jika nilai *setpoint* yang ditentukan diluar batasan tersebut maka sistem kendali konveyor tidak akan berjalan dengan baik.

4.4.1 Pengujian pada suplai *feeder* 128 gram/detik

Pengujian ini bertujuan untuk mengetahui apakah sistem kendali menggunakan PWM pada konveyor ini berjalan dengan baik atau tidak dengan membandingkan setpoint yang ditentukan dengan setpoint yang tercapai.

Pengujian diawali dengan menghidupkan motor kemudian memasukkan nilai *setpoint* dengan menekan tombol keypad, lalu penutup pada *feeder* dibuka pada suplai 128 gr/dtk kemudian tekan tombol eksekusi, waktu yang diberikan 20 detik, dalam selang waktu tersebut, kecepatan konveyor akan menyesuaikan demi mencapai *setpoint* tersebut dengan memperlambat atau mempercepat kecepatan konveyor sesuai dengan masukan dari sensor berat. Hasilnya kemudian ditimbang lalu dibagi dengan waktu yang diberikan sehingga didapatkan nilai setpoint yang tercapai.

Tabel 4.3 berikut memperlihatkan pengujian berat material pada setpoint yang berbeda dan tabel 4.4 menunjukkan perbandingan antara *setpoint* yang ditentukan dengan *setpoint* yang tercapai.

Tabel 4.3 Pengujian berat material dengan setpoint yang berbeda

Percobaan ke	t (s)	Berat Material (Gram)			
		70 gr/dtk	90 gr/dtk	110 gr/dtk	130 gr/dtk
1	30	2350	2750	3400	3800
2	30	2355	2850	3100	3700
3	30	2250	2800	3350	3750
4	30	2300	2850	3200	3650
5	30	2350	2800	3250	3700
Berat Total		11605	14050	16300	18600
Berat Rata - Rata		2321	2810	3260	3720

Tabel 4.4 Perbandingan antara setpoint yang ditentukan dengan setpoint yang tercapai

Percobaan ke	t (s)	Setpoint (Gr/dtk)			
		70	90	110	130
1	30	78,3	91,6	113,3	126,7
2	30	78,5	95	103,3	123,3
3	30	75	93,3	111,67	125
4	30	75,67	95	106,67	121,7
5	30	78,3	93,3	108,3	123,3
Setpoint Total		385,8	468,2	543,24	620
Setpoint Rata - Rata		77,15	93,64	108,65	124

Untuk mengetahui baik tidaknya sistem kendali konveyor ini, maka perlu dicari persentase kesalahan pada masing – masing *setpoint* yang ditentukan dengan nilai *setpoint* yang tercapai sesuai dengan persamaan sebagai berikut :

- Setpoint 70 gr/dtk
  % kesalahan

$$= \left| \frac{SP_{ideal} - SP_{rata-rata}}{SP_{rata-rata}} \right| x\,100\%$$

$$= \left| \frac{70 - 77{,}15}{77{,}15} \right| x\,100\% = 9{,}27\ \%$$

- Setpoint 90 gr/dtk
  % kesalahan

$$= \left| \frac{SP_{ideal} - SP_{rata-rata}}{SP_{rata-rata}} \right| x\,100\%$$

$$= \left| \frac{90 - 93{,}64}{93{,}64} \right| x\,100\% = 3{,}88\ \%$$

- Setpoint 110 gr/dtk
  % kesalahan

$$= \left| \frac{SP_{ideal} - SP_{rata-rata}}{SP_{rata-rata}} \right| x\,100\%$$

$$= \left| \frac{110 - 108{,}65}{108{,}65} \right| x\,100\% = 1{,}24\ \%$$

- Setpoint 130 gr/dtk
  % kesalahan

$$= \left| \frac{SP_{ideal} - SP_{rata-rata}}{SP_{rata-rata}} \right| x\,100\%$$

$$= \left|\frac{130-124}{124}\right| x\,100\% = 4{,}84\ \%$$

Persentase Kesalahan rata – rata

$$= \frac{9{,}27\% + 3{,}88\% + 1{,}24\% + 4{,}84\%}{4} = 4{,}65\ \%$$

Berdasarkan hasil perhitungan diatas dapat dinyatakan bahwa sistem kendali konveyor pada suplai *feeder* 128 gr/dtk ini cukup baik dengan tingkat kesalahan antara 1,24 % sampai 9,27 % danp ersentase kesalahan rata – rata sebesar 4,65 %. Persentase kesalahan terkecil pada *setpoint* 110 gr/dtk dengan 1,24%, sedangkan persentase kesalahan terbesar pada *setpoint* 70 gr/dtk dengan 9,27 %. Hal ini dapat disebabkan karena *setpoint* 70 gr/dtk merupakan setpoint yang mendekati *setpoint* minimum, kendali konveyor belum begitu stabil pada *setpoint* ini sehingga sering terjadi penumpukan material setelah waktu 20 detik percobaan.

4.4.2 Pengujian pada suplai feeder 170 gram/detik

Tujuan pengujian dan prosedur percobaan pada suplai *feeder* 170 gr/dtk ini sama dengan suplai *feeder* 128 gr/dtk, hanya berbeda pada besarnya bukaan penutup *feeder* yaitu pada suplai 170 gr/dtk.

Tabel 4.5 berikut memperlihatkan pengujian berat material pada setpoint yang berbeda dan tabel 4.6 menunjukkan perbandingan antara *setpoint* yang ditentukan dengan *setpoint* yang tercapai.

Tabel 4.5 Pengujian berat material dengan setpoint yang berbeda

Percobaan ke	t (s)	Berat Material (Gram)				
		90 gr/dtk	110 gr/dtk	130 gr/dtk	150 gr/dtk	170 gr/dtk
1	20	1950	2260	2630	3100	3300
2	20	1900	2300	2730	3200	3350
3	20	1850	2300	2700	3150	3200
4	20	1900	2200	2650	3100	3250
5	20	1950	2250	2550	3200	3350
Berat Total		9550	11310	13260	15750	16450
Berat Rata - Rata		1910	2262	2652	3150	3290

Tabel 4.6 Perbandingan antara setpoint yang ditentukan dengan setpoint yang tercapai

Percobaan ke	t (s)	Setpoint (Gr/dtk)				
		90	110	130	150	170
1	20	97,5	113	131,5	155	165
2	20	95	115	136,5	160	167,5
3	20	92,5	115	135	157,5	160
4	20	95	110	132,5	155	162,5
5	20	97,5	112,5	127,5	160	167,5
Setpoint Total		477,5	565,5	663	787,5	822,5
Setpoint Rata - Rata		95,5	113,1	132,6	157,5	164,5

Persentase kesalahan pada masing – masing setpoint pada suplai feeder 171 gr/dtk dapat dihitung sesuai dengan persamaan berikut :

- Setpoint 90 gr/dtk

  % kesalahan

$$= \left|\frac{SP_{ideal} - SP_{rata-rata}}{SP_{rata-rata}}\right| x\,100\%$$

$$= \left|\frac{90-95{,}5}{95{,}5}\right| x\,100\% = 5{,}76\ \%$$

- Setpoint 110 gr/dtk
  % kesalahan

$$= \left| \frac{SP_{ideal} - SP_{rata-rata}}{SP_{rata-rata}} \right| x\,100\%$$

$$= \left| \frac{110 - 113{,}1}{113{,}1} \right| x\,100\% = 2{,}74\ \%$$

- Setpoint 130 gr/dtk
  % kesalahan

$$= \left| \frac{SP_{ideal} - SP_{rata-rata}}{SP_{rata-rata}} \right| x\,100\%$$

$$= \left| \frac{130 - 132{,}6}{132{,}6} \right| x\,100\% = 1{,}96\ \%$$

- Setpoint 150 gr/dtk
  % kesalahan

$$= \left| \frac{SP_{ideal} - SP_{rata-rata}}{SP_{rata-rata}} \right| x\,100\%$$

$$= \left| \frac{150 - 157{,}5}{157{,}5} \right| x\,100\% = 4{,}76\ \%$$

- Setpoint 170 gr/dtk
  % kesalahan

$$= \left| \frac{SP_{ideal} - SP_{rata-rata}}{SP_{rata-rata}} \right| x\,100\%$$

$$= \left| \frac{170 - 164{,}5}{164{,}5} \right| x\,100\% = 3{,}34\ \%$$

Persentase kesalahan rata – rata $= \dfrac{5{,}76\% + 2{,}74\% + 1{,}96\% + 4{,}76\% + 3{,}34\%}{5}$

$= 3{,}71\ \%$

Berdasarkan hasil perhitungan diatas dapat dinyatakan bahwa sistem kendali konveyor pada suplai *feeder* 171 gr/dtk ini cukup baik dengan tingkat kesalahan antara 1.96 % sampai 5,76 % dan persentase kesalahan rata – rata sebesar 3,71 %. Persentase kesalahan terkecil pada *setpoint* 130 gr/dtk dengan 1,96 %, sedangkan persentase kesalahan terbesar pada *setpoint* 90 gr/dtk dengan 5,76 %.

Sama halnya dengan suplai *feeder* 128 gr/dtk, persentase kesalahan terbesar pada suplai 171 gr/dtk ini juga berada pada *setpoint* yang mendekati *setpoint* minimum, yaitu pada *setpoint* 90 gr/dtk, kendali konveyor belum begitu stabil pada *setpoint* ini sehingga sering terjadi penumpukan material setelah waktu 20 detik percobaan.

4.4.3 Pengujian pada suplai *feeder* 210 gram/detik

Tujuan pengujian dan prosedur percobaan pada suplai *feeder* 210 gr/dtk ini sama dengan suplai *feeder* 128 gr/dtk dan suplai *feeder* 171 gr/dtk, hanya berbeda pada besarnya bukaan penutup *feeder* yaitu pada suplai 210 gr/dtk.

Tabel 4.7 berikut memperlihatkan pengujian berat material pada setpoint yang berbeda dan tabel 4.8 menunjukkan perbandingan antara *setpoint* yang ditentukan dengan *setpoint* yang tercapai.

Tabel 4.7 Pengujian berat material dengan setpoint yang berbeda

Percobaan ke	t (s)	Berat Material (Gram)					
		120 gr/dtk	140 gr/dtk	160 gr/dtk	180 gr/dtk	200 gr/dtk	220 gr/dtk
1	20	2300	2900	3300	3500	3850	4200
2	20	2500	2850	3400	3650	3900	4250
3	20	2400	2800	3250	3700	3950	4300
4	20	2450	2900	3300	3550	3900	4350
5	20	2500	2850	3350	3700	3850	4300
Berat Total		12150	14300	16600	18100	19450	21400
Berat Rata - Rata		2430	2860	3320	3620	3890	4280

Tabel 4.8 Perbandingan antara setpoint yang ditentukan dengan setpoint yang tercapai

Percobaan ke	t (s)	Setpoint (Gr/dtk)					
		120	140	160	180	200	220
1	20	115	145	165	175	192,5	210
2	20	125	142,5	170	182,5	195	212,5
3	20	120	140	162,5	185	197,5	215
4	20	122,5	145	165	177,5	195	217,5
5	20	125	142,5	167,5	185	192,5	215
Setpoint Total		607,5	715	830	905	972,5	1070
Setpoint Rata - Rata		121,5	143	166	181	194,5	214

Persentase kesalahan pada masing – masing *setpoint* pada suplai *feeder* 210 gr/dtk dapat dihitung sesuai dengan persamaan berikut :

- Setpoint 120 gr/dtk

  % kesalahan

$$= \left| \frac{SP_{ideal} - SP_{rata-rata}}{SP_{rata-rata}} \right| x\,100\%$$

$$= \left| \frac{120 - 121{,}5}{121{,}5} \right| x\,100\% = 1{,}23\ \%$$

- Setpoint 140 gr/dtk

  % kesalahan

$$= \left| \frac{SP_{ideal} - SP_{rata-rata}}{SP_{rata-rata}} \right| x\,100\%$$

$$= \left| \frac{140-143}{143} \right| x\,100\% = 2{,}1\ \%$$

- Setpoint 160gr/dtk
  % kesalahan

$$= \left| \frac{SP_{ideal} - SP_{rata-rata}}{SP_{rata-rata}} \right| x\,100\%$$

$$= \left| \frac{160-166}{166} \right| x\,100\% = 3{,}61\ \%$$

- Setpoint 180 gr/dtk
  % kesalahan

$$= \left| \frac{SP_{ideal} - SP_{rata-rata}}{SP_{rata-rata}} \right| x\,100\%$$

$$= \left| \frac{180-181}{181} \right| x\,100\% = 0{,}55\ \%$$

- Setpoint 200 gr/dtk
  % kesalahan

$$= \left| \frac{SP_{ideal} - SP_{rata-rata}}{SP_{rata-rata}} \right| x\,100\%$$

$$= \left| \frac{200-194{,}5}{194{,}5} \right| x\,100\% = 2{,}83\ \%$$

- Setpoint 220 gr/dtk
  % kesalahan

$$= \left| \frac{SP_{ideal} - SP_{rata-rata}}{SP_{rata-rata}} \right| x\,100\%$$

$$= \left| \frac{220-214}{214} \right| x\,100\% = 2{,}8\ \%$$

Persentase kesalahan rata – rata $= \frac{1{,}23\% + 2{,}1\% + 3{,}61\% + 0{,}55\% + 2{,}83\% + 2{,}8\%}{5}$

= 2,19 %.

Berdasarkan hasil perhitungan diatas dapat dinyatakan bahwa sistem kendali konveyor pada suplai *feeder* 210 gr/dtk ini cukup baik dengan tingkat kesalahan antara 0,55 % sampai 3,61 %. Persentase kesalahan terkecil pada *setpoint* 180 gr/dtk dengan 0,55 %, sedangkan persentase kesalahan terbesar pada *setpoint* 160 gr/dtk dengan 3,61 %. Pada suplai *feeder* ini tidak terdapat perbedaan tingkat kesalahan yang jauh dari masing – masing *setpoint*. Hal ini disebabkan karena *setpoint* terkecil yang diambil adalah 120 gr/dtk tidak terlalu mendekati *setpoint* minimum sehingga penumpukan material pada *setpoint* ini tidak terjadi.

4.5 Analisa hasil pengujian

Berdasarkan percobaan yang telah di lakukan terhadap kendali konveyor ini, maka dapat dikatakan bahwa secara umum sistem kendali konveyor ini berjalan dengan cukup baik pada masing – masing suplai *feeder*, baik 128 gr/dtk, 171 gr/dtk dan 210 gr/dtk, hal ini dapat ditunjukkan dari persentase tingkat kesalahan rata – ratanya ada masing – masing suplai *feeder* dibawah 5%, Tingkat kesalahan yang terjadi dapat disebabkan banyak hal, antara lain disebabkan oleh faktor kemampuan dari pembacaan sensor berat, kekurangtelitian pembacaan waktu dan kekurangan dari mekanik konveyor itu sendiri.

Apabila pada percobaan dilakukan penentuan *setpoint* diatas *setpoint* maksimum, maka *setpoint* yang terbaca tetaplah *setpoint* maksimum yang ditentukan berdasarkan rumus sebelumnya, hal ini dikarenakan kecepatan maksimum belt konveyor adalah 58 cm/dtk. Pemberian *duty cycle* di bawah 50% sangat berpengaruh pada putaran motor. Suplai tegangan dari sumber daya ke motor sangat kecil antara 0 V sampai 4 V, mempengaruhi putaran *belt* pada konveyor sehingga terjadi slip antara belt dan puley, akibatnya material yang dikirim melalui *feeder*, menumpuk dan membuat *belt* tidak bisa berjalan.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan pengujian terhadap sistem yang telah dibuat maka dapat diambil kesimpulan sebagai berikut :

1. Fasilitas *timer/counter* untuk pemanfaatan pengendalian PWM dan channel ADC yang terdapat pada mikrokontroler ATMega 8535 dapat diaplikasikan dengan baik pada pengendalian konveyor tanpa memerlukan prosedur yang rumit.
2. Penerapan sistem PWM dalam pengaturan kecepatan motor arus searah perlu memperhatikan besar kristal atau frekuensi yang dipakai pada mikrokontroler, *fuse bytes* pada ISP Programmernya, serta *prescaler* yang digunakan, jika tidak sesuai maka motor tidak akan berjalan dengan halus atau tersentak – sentak.
3. Sistem kendali konveyor ini berjalan dengan cukup baik pada masing – masing suplai *feeder*, terutama pada suplai *feeder* 210 gr/dtk dengan persentase kesalahan rata – rata 2,19 %. Pada suplai *feeder* 128 gr/dtk serta 170 gr/dtk dengan persentase kesalahan rata – rata 4,65 % dan 3,71 %.
4. Semakin besar suplai material ke *belt* (selama dalam kapasitas *belt*), maka semakin baik sistem kendali yang berjalan, hal ini dapat dilihat dari kecilnya persentase kesalahan pada suplai *feeder* 210 gr/dtk.

5.2 Saran

1. Untuk bahasa pemrograman pada perkembangan lebih lanjut bisa dicoba menggunakan bahasa pemrograman yang lain seperti bahasa C, dan lain – lain.
2. Setelah aplikasi dilapangan kemungkinan program sederhana bisa menjadi kompleks dan panjang, karena mikrokontroler memiliki kapasitas memori yang terbatas sehingga hanya cocok digunakan untuk mengontrol sistem- sistem yang sederhana , maka perlu dicoba pengendali yang lain misalnya PLC.

DAFTAR PUSTAKA

1. Budiharto, Widodo & Rizal, Gamayel. 2007. *Belajar Sendiri 12 Proyek Mikrokontroler untuk Pemula*. Jakarta : Elex Media Komputindo.
2. Budiharto, Widodo & Togo Jefri. 2007. *12 Proyek Sistem Akuisisi Data*. Jakarta : Elex Media Komputindo.
3. Datasheet ATMEGA 8535. 2006. Atmel Corporation
4. Delmar. *Modern Control Technology Components and Systems.* 2nd edition. Kilian

5. Husein Alam, Syarbini & Rusydi Suwardi, Fuad. 2001. *Peralatan dan Pengangkutan Tambang Bawah Tanah*. Edisi Kedua. Teknik Pertambangan UNSRI.
6. Manual Book *Schenck*
7. Ogata. Katsuhito. 1997. *Teknik Kontrol Automatik*. Jakarta: Erlangga.
8. Petruzella, Frank D. 2001. *Elektronik Industri*. (Diterjemahkan oleh Sumanto). Yogyakarta : Andi.
9. Sigit, Riyanto. 2007. *Robotika, Sensor dan Aktuator*. Yogyakarta : Graha Ilmu.
10. Tutorial Mikrokontroler MCS 51 Atmel ISP
11. Utami, Ema & Sukrisno. 2005. *10 Langkah Belajar Logika & Algoritma Mengguakan Bahasa C & C++ di GNU/Linux*. Yogyakarta : Andi.
12. Wardhana, Lingga. 2006. *Belajar Sendiri Mikrokontroler AVR Seri ATMega 8535 Simulasi, Hardware & Aplikasi*. Yogyakarta : Andi.
13. Web: http://www.AllDataSheet.com
14. Web: http://en.wikipedia.org/