

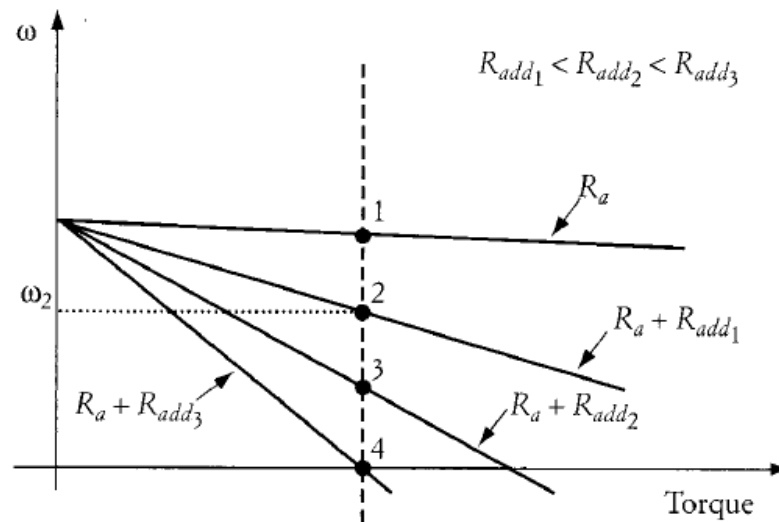
## BAB II

### LANDASAN TEORI

#### 2.1 Perkembangan Kontrol Motor DC

Salah satu keunggulan dari motor dc adalah kecepatannya dapat dengan mudah dikontrol sesuai kebutuhan dengan menggunakan metode sederhana. Pengaturan kecepatan motor dc dapat dilakukan dengan tiga cara (Muchlas, Sunardi and Antoro, 2015), yaitu:

1. Mengubah tahanan rangkaian jangkar.



Gambar 2.1 Penambahan Resistansi Armature Pada Kecepatan  
(El-Sharkawi, 2000)

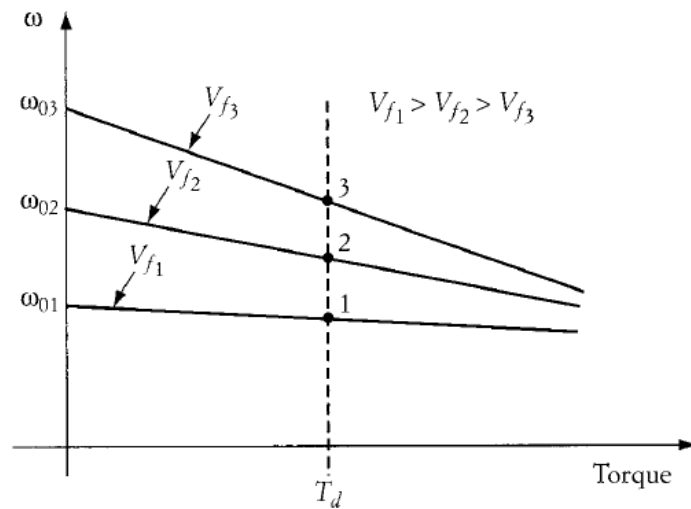
Gambar 2.1 menjelaskan bahwa penambahan resistansi akan mempengaruhi pada kecepatan motor yang dihasilkan. Ketika resistansi dimasukkan ke sirkuit *armature*, penurunan kecepatan  $\Delta\omega$  meningkat dan kecepatan motor berkurang (El-Sharkawi, 2000).

Pengaturan resistansi dalam pengendalian kecepatan motor dc memiliki kekurangan dalam hal efisiensi, pengaturan yang sangat rumit serta memberikan kecepatan yang rendah dibawah kecepatan normal. Metode ini kurang ekonomis, efisien dan tidak banyak digunakan dalam pengendalian kecepatan motor dc. Sehingga mereka menggunakan metode lain, salah satu teknik yang sangat banyak digunakan adalah modulasi lebar pulsa (PWM) (Yadav, Kanodia and Kumari, 2014).

*Effect Of Pulse Width Modulation On DC Motor Speed*, menjelaskan mengenai tugas PWM yang digunakan untuk memvariasikan kecepatan motor dengan mengendalikan tegangan terminal motor. Selain itu penggunaan mikrokontroler PIC dan konverter *buck* DC-DC dalam melakukan sistem pengontrol kecepatan motor DC (Tatenda Katsambe, Luckose and Shahabuddin, 2017).

Selain itu, pengaturan tahanan jangkar ini jarang dipakai karena penambahan tahanan seri terhadap tahanan jangkar menimbulkan rugi panas yang besar (Haryanto and Hidayat, 2012).

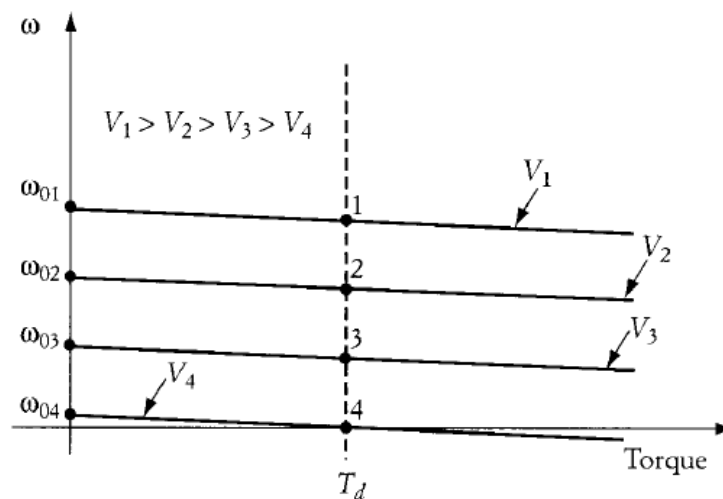
2. Mengubah fluks magnet.



Gambar 2.2 Penambahan Tegangan Medan (Fluks Medan) Pada Kecepatan  
(El-Sharkawi, 2000)

Gambar 2.2 merupakan grafik dari penambahan fluks medan (atau tegangan medan), yang dapat berpengaruh pada kecepatan motor yang dihasilkan. Ketika tegangan medan dikurangi, maka itu akan mengurangi nilai fluks  $\Phi$ , dan kecepatan motor meningkat (El-Sharkawi, 2000).

3. Mengubah tegangan jangkar.



Gambar 2.3 Penambahan Tegangan Pada Kecepatan  
(El-Sharkawi, 2000)

Gambar 2.3 menjelaskan mengenai grafik pemanbahan tegangan yang dapat mempengaruhi kecepatan motor yang dihasilkan. Mengurangi tegangan jangkar  $V$  dari motor mengurangi kecepatan motor (El-Sharkawi, 2000).

Pengaturan tegangan jangkar ini, dilakukan dengan cara mengatur tegangan jangkar motor sehingga kecepatan motor dapat diatur. Kecepatan motor akan bertambah jika menaikkan sumber tegangan dan kecepatan motor akan berkurang jika menurunkan sumber tegangan. Cara ini menghasilkan pengaturan kecepatan motor yang sangat halus dan banyak digunakan dalam kehidupan sehari-hari (Haryanto and Hidayat, 2012).

Salah satu cara mencapai kecepatan dengan teknik pengontrolan tegangan dapat dicapai dengan konverter *full-bridge* (Ranadev and Chakrasali, 2014), metode *switching* dalam pengaturan tegangan digunakan dalam pengaturan kecepatan motor DC.

Selain penggunaan konverter dalam pengaturan tegangan, perlu adanya suatu pengendalian kecepatan yang dapat mengendalikan kecepatan motor DC. Fungsi pengendalian dalam sebuah sistem adalah meningkatkan performa sistem, meminimalkan *error* sistem dan menjaga stabilitas sistem agar sesuai dengan tujuan pengendaliannya. Keluaran sebuah sistem merupakan nilai konstan, oleh karena itu perlu dikendalikan agar nilai yang sesuai kriteria memiliki *error* yang minimal. Sistem kendali atau pengendalian bekerja dalam menjaga nilai *error* untuk selalu mendekati nilai 0 (Siregar, 2018). Beberapa metode yang dapat digunakan untuk mengendalikan kecepatan motor DC seperti *Propotional Integral Derivative* (PID),

Fuzzy Logic Controller, Jaringan Saraf (Antar, Allu and Ali, 2013), Model Predictive Control (MPC) dan lain-lain (Dani *et al.*, 2017).

PI (*Propotional Integral*) kontrol, ataupun PID (*Propotional Integral Derivative*) menjadi salah satu pengendalian yang banyak digunakan sampai saat ini (Dani *et al.*, 2017). Salah satu contoh penggunaan pengontrol ini, yaitu terdapat dalam jurnal *Starting Analysis of Separately Excited DC Motor by PI Controller* dengan motor dc jenis penguat terpisah (*separately excited*) (Itankar, Porate and Singh, 2016). Serta dalam jurnal *Speed Control Of DC Motor Using Chopper*, penggunaan PI dalam motor dc jenis penguat terpisah (*separately excited*) serta penerapan *Chopper* sebagai konverter untuk mengatur nilai tegangan (Sethy and Sinha, 2013).

*Model Predictive Control* (MPC) merupakan strategi pengendalian lanjutan (Abirami *et al.*, 2014). *Model Predictive Control* (MPC) memiliki akurasi dan respon yang cepat dalam pengendalian dibandingkan *PID Controller* dalam proses pada sistem (Jichkar and Sondkar, 2017). Salah satu penggunaan MPC yaitu terdapat dalam karya tulis *Model Predictive Control Of DC Motor Model In MATLAB*, dengan penggunaan motor dc (Khanna and Gaur, 2017).

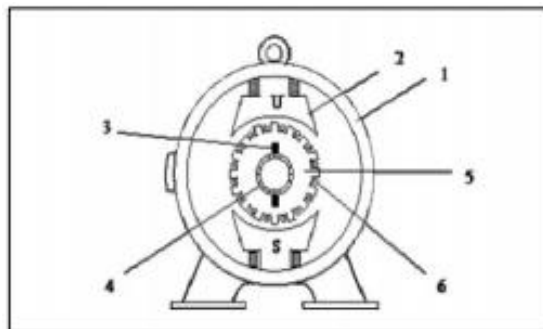
Pengendalian kecepatan motor dc ini, dilakukan dengan menggunakan motor dc jenis penguat terpisah (*separately excited*), penggunaan konverter turunan *buck* yaitu *full-bridge chopper* sebagai pengatur tegangan yang nantinya akan dikendalikan oleh kontrol kecepatan dengan menggunakan metode *Model Predictive Control* (MPC).

## 2.2 Motor DC

Motor DC (*Direct Current*) adalah peralatan elektromagnetik dasar yang berfungsi untuk mengubah tenaga listrik menjadi tenaga mekanik yang desain awalnya diperkenalkan oleh Michael Faraday lebih dari seabad yang lalu (Hidayati and Prasetyo, 2018). Beberapa aplikasi motor DC seperti pada pabrik kertas, *rolling mills*, winder tambang, kerekan, mesin cetak, peralatan mesin, traksi, pabrik tekstil, *excavator*, dan *crane* (Vadapalli *et al.*, 2015).

### 2.2.1 Bagian-bagian Motor DC

Bagian-bagian motor DC secara umum (Sukarasa, 2016), digambarkan seperti gambar 2.4.



Gambar 2.4 Bagian-bagian Motor DC  
(Sukarasa, 2016)

#### 1. Badan Mesin

Badan mesin berfungsi sebagai tempat mengalirnya fluks magnet yang terbuat dari bahan feromagnetik yang dihasilkan oleh kutub magnet. Badan mesin juga untuk meletakkan alat-alat tertentu, sehingga harus terbuat dari bahan yang benar-benar kuat seperti dari besi tuang dan plat campuran baja.

## 2. Inti kutub magnet dan belitan penguat magnet

Inti kutub magnet dan belitan penguat magnet mampu mengalirkan arus listrik sehingga ada elektromagnetik.

## 3. Sikat-sikat

Sikat-sikat berfungsi sebagai jembatan bagi aliran arus jangkar dengan bebas dan juga memegang peranan penting untuk terjadinya proses komutasi.

## 4. Komutator

Komutator berfungsi sebagai penyearah mekanik yang akan dipakai bersama-sama dengan sikat. Sikat-sikat ditempatkan sedemikian rupa sehingga komutasi terjadi pada saat sisi kumparan berbeda.

## 5. Jangkar

Jangkar dibuat dari bahan ferromagnetik dengan maksud memperbesar induksi magnetiknya untuk menghasilkan ggl induksi yang besar.

## 6. Belitan jangkar

Belitan jangkar adalah bagian yang sangat penting pada mesin arus searah dengan berfungsi untuk tempat timbulnya tenaga putar motor.

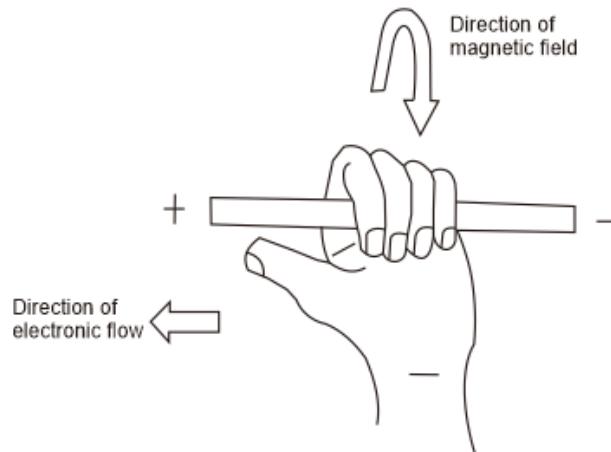
### 2.2.2 Prinsip Kerja Motor DC

Terdapat dua prinsip dasar yang melatarbelakangi kerja motor DC, yaitu:

#### 1. Adanya aliran arus yang melewati sebuah konduktor atau penghantar.

Dimana, akan timbul adanya medan magnet yang mengelilingi penghantar tersebut. Arah garis gaya magnet (*fluks magnet*) ini sesuai kaidah tangan kiri yang ditunjukkan pada gambar 2.5. Ibu jari menandakan arah arus elektron

yang mengalir dan jari-jari menunjukkan arah dari garis gaya magnet (*fluks*) yang mengelilingi penghantar.



Gambar 2.5 Kaidah Tangan Kiri

(Zamroni and Moediyono, 2010) “Telah Diolah Kembali”

2. Gaya pada penghantar bergerak dalam medan magnet. Besarnya gaya yang didesakkan untuk menggerakkan berubah sebanding dengan kekuatan medan magnet, besarnya arus yang mengalir pada penghantar, dan panjang penghantar. Gaya tersebut sering disebut gaya Lorentz. Sesuai dengan rumus:

$$F = B i l \quad (2.1)$$

Dimana:

F = Gaya (Newton)

B = Kuat medan magnet (Tesla)

I = Arus yang mengalir (Ampere)

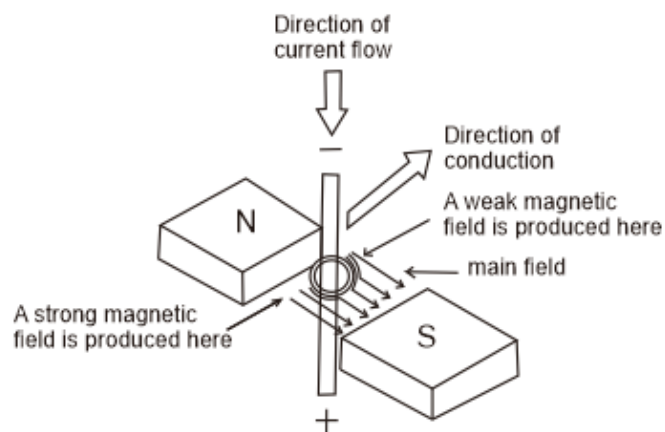
L = Panjang penghantar (meter)

Arah dari garis gaya magnet tergantung dari arah arus yang mengalir pada kumparan dan arah dari garis-garis fluks magnet antara dua kutub. Sebagaimana diilustrasikan pada gambar 2.6. Medan magnet mengembang



diantara dua kutub dari magnet permanen atau induksi elektromagnet. Ketika penghantar berarus ditempatkan diantara dua kutub magnet, maka menghasilkan pembengkokan garis gaya.

Sehingga, di satu sisi memusatkan kedua medan magnet menimbulkan medan magnet yang kuat dan disisi lain berlawanan menimbulkan medan magnet yang lemah. Garis gaya magnet yang kuat cenderung lurus keluar dan menekan kearah garis gaya magnet yang lemah. Dan menyebabkan penghantar tersebut berputar berlawanan arah jarum jam.

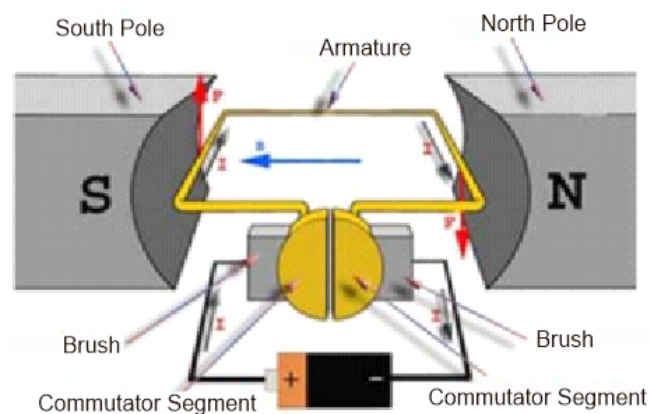


Gambar 2.6 Interaksi Penghantar Berarus Diantara Medan Magnet  
(Zamroni and Moediyono, 2010) “Telah Diolah Kembali”

Ketika kawat konduktor dialiri arus, akan dihasilkan gaya gerak listrik karena konduktor bergerak didalam kumparan medan magnet. Untuk menghasilkan energi mekanik sempurna, maka tegangan gerak yang disebabkan reaksi lawan harus lebih kecil. Dengan mengalirkan arus pada kumparan jangkar yang terlindung dari medan, maka motor akan berputar, sehingga terjadilah gaya yang akan memutar jangkar (*armature*) motor.

Interaksi dua buah medan yang dihasilkan oleh bagian jangkar dan bagian medan (*field*) dari motor DC menghasilkan putaran pada motor DC. Gambar 2.7,

menunjukkan kumparan yang dihubungkan ke sumber arus searah, sedangkan magnet permanen adalah bagian dari jangkar. Bagian jangkar ini tidaklah harus dalam magnet permanen, karena bisa juga berbentuk lilitan yang akan menjadi elektomagnet jika memperoleh sumber arus searah. Jika motor DC berjenis jangkar lilitan, maka harus ada dua sumber arus searah, satu untuk jangkar dan satu lagi untuk medan. Komutator adalah bagian lain yang tidak kalah penting pada motor DC, yang berpasangan dengan cincin belah (*gap rings*). Pasangan ini menjadikannya lilitan medan berputar dengan arah arus tetap, yang biasanya disebut konverter mekanik.

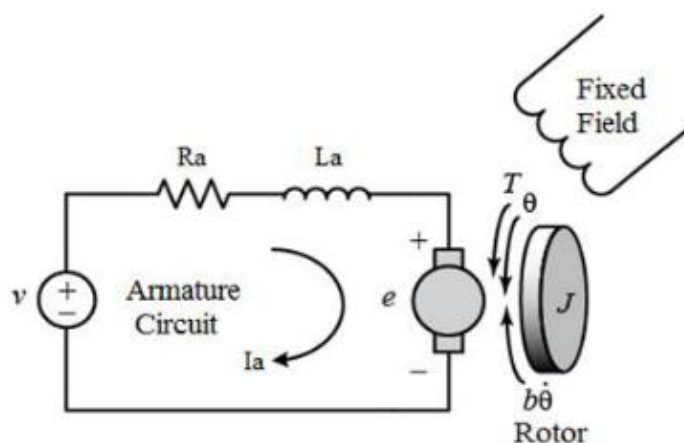


Gambar 2.7 Prinsip Kerja Motor DC  
(Sukarasa, 2016) “Telah Diolah Kembali”

### 2.2.3 Motor DC Penguat Terpisah

Motor DC Penguat terpisah merupakan salah satu jenis dari motor listrik. Seperti namanya, dalam kasus motor DC penguat terpisah pasokan diberikan secara terpisah ke lilitan medan dan angker. Fakta pembeda utama dalam jenis motor DC ini adalah arus jangkar tidak mengalir melalui belitan medan, karena belitan medan

diberi energi dari sumber eksternal arus DC yang terpisah seperti ditunjukkan pada gambar berikut.



Gambar 2.8 Rangkaian Ekivalen Motor DC Penguat Terpisah  
(Özer, Kivrak and Oğuz, 2017)

Gulungan medan memasok fluks medan ke *armature*. Ketika tegangan DC diterapkan ke motor, arus diumpankan ke gulungan anker melalui sikat dan komutator. Karena rotor ditempatkan di medan magnet dan membawa arus juga. Jadi motor akan mengembangkan ggl kembali dan torsi untuk menyeimbangkan torsi beban pada kecepatan tertentu.

#### 2.2.4 Rumus Matematis Motor DC Penguat Terpisah

Ketika motor dc penguat terpisah tereksitasi oleh arus medan  $I_f$  dan arus jangkar  $I_a$  mengalir dalam sirkuit, motor mengembangkan EMF dan torsi untuk menyeimbangkan torsi beban pada kecepatan tertentu. Arus medan  $I_f$  tidak tergantung pada arus jangkar  $I_a$ . Setiap belitan disediakan secara terpisah. Setiap perubahan arus jangkar tidak berpengaruh pada arus medan.  $I_f$  umumnya jauh lebih kecil dari  $I_a$ .

Dalam gambar 2.8, anggaplah  $V_a$  tegangan jangkar dalam volt,  $I_a$  arus jangkar dalam ampere,  $E_g$  ggl dalam volt,  $L_a$  induktansi jangkar dalam Henry,  $R_a$  resistansi jangkar dalam ohm.

Persamaan jangkar ditunjukkan di bawah ini:

$$V_a = E_g + I_a R_a + L_a \frac{dI_a}{dt} \quad (2.2)$$

Sekarang persamaan torsi akan diberikan oleh:

$$T_m = J \frac{d\omega}{dt} + B\omega + T_l \quad (2.3)$$

Dimana,  $T_l$  adalah torsi beban dalam Nm,  $T_m$  adalah torsi mekanik dalam Nm,  $J$  adalah momen inersia dalam  $\text{kg} / \text{m}^2$ ,  $B$  adalah koefisien gesek motor dan  $\omega$  adalah kecepatan sudut dalam rad / detik.

Dengan asumsi tidak adanya (dapat diabaikan) dari gesekan pada rotor motor, maka akan menghasilkan  $B = 0$ . Oleh karena itu, persamaan torsi baru akan diberikan oleh:

$$T_m = J \frac{d\omega}{dt} + T_l \quad (2.4)$$

Persamaan untuk ggl kembali:

$$E_g = K \Phi \omega \quad (2.5)$$

$$T_m = K \Phi i_a \quad (2.6)$$

Atau bisa juga menggunakan (Mawardi and Lianda, 2018):

$$T = \frac{5250 \times HP}{n} \quad (2.7)$$

Dimana HP merupakan daya motor (HP),  $T$  adalah torsi motor (Nm),  $n$  adalah kecepatan putar motor (RPM) dan 5250 merupakan konstanta.

Maka rumus kecepatannya:

Substitusi persamaan (2) (5) didapat

$$\omega = \frac{(V_a - i_a R_a)}{K \Phi} \quad (2.8)$$

Sekarang, dari persamaan di atas jelas bahwa kecepatan motor DC tergantung pada tegangan yang diberikan pada arus jangkar, resistansi jangkar dan fluks medan. Jadi, ada tiga cara mengendalikan kecepatan motor DC yaitu kontrol tegangan jangkar, kontrol resistansi jangkar dan kontrol fluks medan (Sethy and Sinha, 2013).

#### 2.2.5 Pengendalian Kecepatan Motor DC Penguat Terpisah Dengan Pengaturan Tegangan

Salah satu metode pengendalian kecepatan motor dc penguat terpisah yaitu dengan cara pengaturan tegangan. Jika tegangan jangkar dari motor dc penguat terpisah yang sedang berjalan pada kecepatan tunak diturunkan, maka torsi motor akan turun. Torsi motor akan lebih kecil daripada torsi beban, maka kecepatan motor akan turun sampai torsi beban sama dengan torsi motor. Dan sebaliknya, jika tegangan jangkar dari motor dc yang sedang berjalan pada kecepatan tunak dinaikkan, maka torsi motor akan naik. Dengan begitu kecepatan motor akan naik sampai torsi motor sama dengan torsi beban (Suyono, 2014).

Saat menambah dan mengurangi kecepatan motor dc, tegangan jangkar harus dinaik atau turun-kan dengan perubahan (*step*) yang kecil. Perubahan yang besar secara langsung dapat mengakibatkan aliran arus pada jangkar motor yang dapat merusak komutator atau menurunkan umur motor.

Karena tegangan jangkar tidak boleh dicatu dengan tegangan yang melebihi rating motor tersebut, maka metode ini hanya digunakan untuk mendapatkan operasi motor dibawah karakteristik kecepatan-torsi normalnya.

### 2.3 DC Chopper

Menurut Jaafer Sadiq Jaafer dan Mohammed Chessab Mahdi dalam karya tulisnya *Speed control of separately excited dc motor using chopper* menjelaskan bahwa sebuah *chopper* adalah daya perangkat elektronik statis yang mengubah tegangan *input* DC tetap ke tegangan *output* DC variabel. Sebuah *chopper* dapat dianggap sebagai DC setara dengan transformator ac karena mereka berperilaku dalam cara yang identik. *Chopper* melibatkan satu konversi tahap, ini adalah sistem *chopper* lebih efisien menawarkan kontrol halus, efisiensi tinggi (Jaafer and Mahdi, 2013). Karena konversi satu tahap, *chopper* lebih efisien dan sekarang digunakan di seluruh dunia untuk sistem transit cepat, dalam kerekan laut, mobil troli, pengangkut ranjau dan truk *forklift* dll. Mobil listrik masa depan kemungkinan akan menggunakan *chopper* untuk kontrol kecepatan dan pengereman mereka. Sistem *chopper* menawarkan kontrol yang mulus, efisiensi tinggi, respons yang lebih cepat, dan fasilitas regenerasi (Sethy and Sinha, 2013).

DC *chopper* atau konverter DC-DC dapat digunakan sebagai regulator *mode switching* untuk mengubah tegangan dc, yang biasanya tidak diatur menjadi tegangan keluaran dc yang bisa diatur. Regulasi ini biasanya dicapai oleh PWM pada frekuensi tetap dan driver *switching* biasanya BJT, MOSFET, atau IGBT

(Lopa *et al.*, 2016). Perangkat semikonduktor daya MOSFET dipilih karena kecepatan *switching* yang tinggi dan penurunan 0,5 hingga 2,5 volt.

Secara umum, ada tiga rangkaian (topologi) dasar konverter dc-dc, yaitu *buck*, *boost*, dan *buck-boost*. Rangkaian lain biasanya mempunyai kinerja mirip dengan topologi dasar ini sehingga sering disebut sebagai turunannya. Contoh dari konverter dc-dc yang dianggap sebagai turunan rangkaian *buck* adalah *forward*, *push-pull*, *half-bridge*, dan *full-bridge*. Contoh dari turunan rangkaian *boost* adalah konverter yang bekerja sebagai sumber arus. Contoh dari turunan rangkaian *buck-boost* adalah konverter *flyback* (Dahono, 2009).

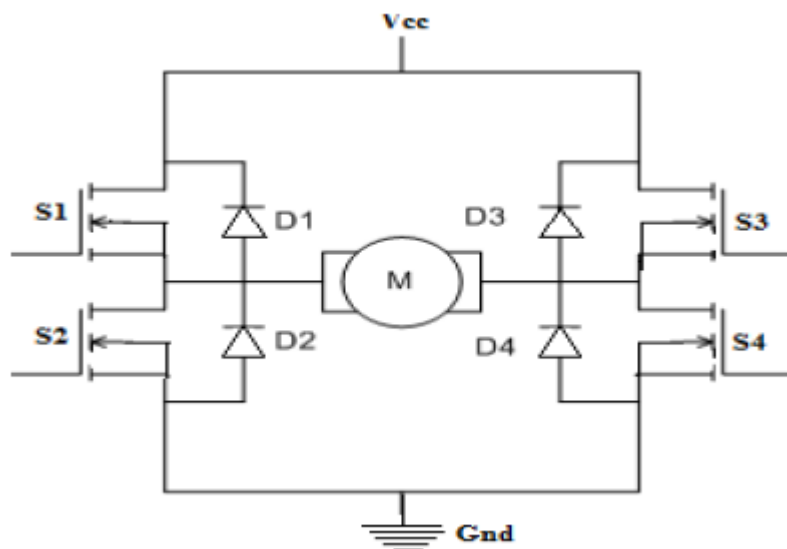
Salah satu topologi yang digunakan adalah turunan konverter *buck* yaitu *full-bridge*. Konverter jenis *buck* merupakan jenis konverter yang banyak digunakan dalam industri catu-daya. Konverter ini akan mengkonversikan tegangan dc masukan menjadi tegangan dc lain yang lebih rendah (konverter penurun tegangan).

Dalam pengendalian kecepatan dengan pengontrolan tegangan jangkar, dimana nilai dari tegangan jangkar tidak boleh dicatu dengan tegangan yang melebihi rating dari motor maka konverter jenis *buck* sangatlah cocok untuk digunakan.

### 2.3.1 Full-bridge Chopper

*Full-bridge chopper* (Gambar 2.9) merupakan salah satu jenis konverter DC-DC, ketika S1 dan S4 dihidupkan, maka arus akan mengalir dari tegangan  $V_{cc}$  mengalir dari kiri motor ke kanan motor (*Forward Motoring*) searah jarum jam. Ketika S2 dan S3 dihidupkan, kebalikannya akan memberi energi motor untuk

bergerak ke arah sebaliknya (*Reverse Motoring*). Tegangan yang muncul di terminal motor bervariasi dengan mengubah waktu hidup-mati sinyal modulasi lebar pulsa. Operasi motor untuk kondisi sakelar yang berbeda ditunjukkan pada Tabel 2.1.



Gambar 2.9 Rangkaian *Full-bridge* Chopper  
(Ranadev and Chakrasali, 2014)

Tabel 2.1 Operasi Motor untuk Kondisi Switch yang Berbeda

Kondisi Switch	S1	S2	S3	S4
<i>Forward Motoring</i>	On	Off	Off	On
<i>Reverse Motoring</i>	Off	On	On	Off

(Ranadev and Chakrasali, 2014)

### 2.3.2 Keuntungan *Full-bridge Chopper*

Masalah utama yang dihadapi konverter *buck* lainnya, seperti konverter *forward* adalah penggunaan trafo yang kurang efisien. Penggunaan trafo kurang efisien karena trafo dimagnetisasi secara tak simetris (gelombang tegangan trafo

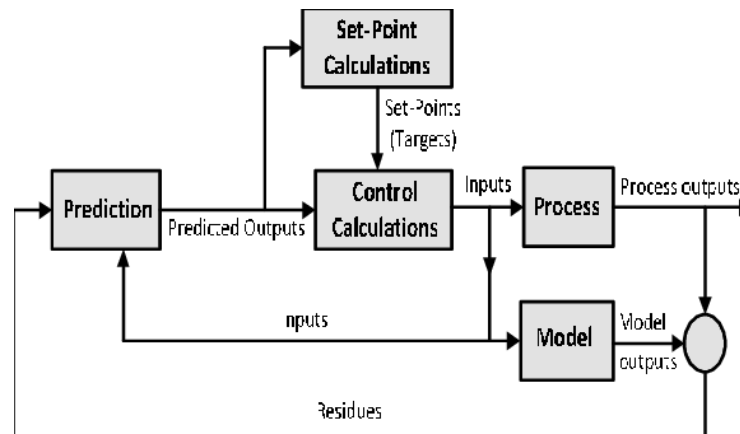


bukan gelombang bolak-balik). Untuk mengatasi masalah ini, dapat menggunakan topologi setengah-jembatan (*half-bridge*). Kelemahan utama dari topologi ini yaitu tidak cocoknya untuk dioperasikan dalam mode arus terkendali. Inilah alasan utama mengapa topologi ini tidak banyak digunakan.

Untuk mengatasi masalah pada konverter setengah-jembatan, bisa menggunakan topologi jembatan-penuh (*full-bridge*). Konverter jembatan-penuh merupakan gabungan dua konverter setengah-jembatan. Mode kerja konverter jembatan-penuh bisa dirancang agar bekerja dalam mode pensaklaran lunak (*soft switching*). Pada mode kerja ini, pembukaan dan penutupan saklar selalu terjadi saat tegangan pada saklar sama dengan nol. Akibatnya, rugi-rugi daya pensaklaran (rugi-rugi daya yang terjadi selama proses penutupan dan pembukaan saklar) bisa ditekan menjadi sangat rendah (Dahono, 2009).

#### **2.4 Model Predictive Control (MPC)**

*Model Predictive Control* (MPC) adalah kendali model terprediksi. Sesuai namanya, sistem kendali ini menggunakan hasil prediksi dalam mengeluarkan kontrol *input*. Kontrol *input* ini merupakan kontrol optimal untuk pengendalian berdasarkan hasil prediksi *plant* beberapa waktu ke depan. Teknik ini sudah banyak diaplikasikan dalam industri-industri dan awalnya diterapkan pada proses kimia.



Gambar 2.10 Blok Diagram Model Predictive Control

MPC adalah sebuah teknik kendali untuk sistem waktu diskrit. Persamaan *state space plant*/proses dalam waktu diskrit adalah:

$$x(k + 1) = Ax(k) + Bu(k) \quad (2.9)$$

$$y(k) = Cx(k) \quad (2.10)$$

*Cost function* yang mewakili perilaku yang diinginkan dari sistem perlu didefinisikan. Fungsi ini mempertimbangkan referensi, status masa depan, dan aktuasi masa depan:

$$g = y(k) - y(k + 1) \quad (2.11)$$

MPC mengoptimalkan masalah dengan meminimalkan *cost function*, optimisasi masalah diselesaikan kembali setiap pengambilan sampel, menggunakan data yang diukur baru dan memperoleh urutan baru dari tindakan optimal setiap waktu, hal ini disebut juga *receding horizon strategy*.

Model predictive control (MPC) memiliki kemampuan untuk memprediksi peristiwa masa depan dan mengambil tindakan kontrol berdasarkan tujuan minimalisasi fungsi (YARAMASU and WU, 2017). Pada kendali kecepatan motor

dc, metode MPC melibatkan arus prediksi. Berikut merupakan persamaan arus prediksinya.

$$i_{(k+1)} = - \frac{((speed)*k*fl*Ts)+(Ts*(armature\ voltage))}{R_a} \quad (2.12)$$

Setelah arus prediksi ditemukan, dilakukan optimasi *cost function*. Optimasi *cost function* dilakukan untuk meminimalisir *error* antara referensi dengan prediksi. Perhitungan *cost function* yaitu dengan menggunakan kecepatan referensi dan arus prediksi, sehingga optimasi *cost function* dapat dipahami sebagai berikut.

$$g = |(speed) - i_{(k+1)}| \quad (2.13)$$

*Cost function* digunakan untuk mengendalikan lebar PWM yang akan mengatur seberapa besar tegangan jangkar diberikan pada motor dc. Dengan cara seperti ini, kecepatan motor dc dapat berputar dengan kecepatan yang konstan.

## 2.5 MATLAB dan Simulink

### 2.5.1 MATLAB

#### 2.5.1.1 Sekilas Tentang MATLAB

MATLAB merupakan perangkat lunak yang digunakan untuk pemrograman, analisis, serta komputasi teknis dan matematis berbasis matriks. MATLAB adalah singkatan dari *Matrix Laboratory* karena mampu menyelesaikan masalah perhitungan dalam bentuk matriks. MATLAB versi pertama dirilis pada tahun 1970 oleh Cleve Moler. Pada awalnya, MATLAB didesain untuk menyelesaikan masalah-masalah persamaan aljabar linear. Seiring berjalannya waktu, program ini terus mengalami perkembangan dari segi fungsi dan performa komputasi (Tjolleng, 2017).

Bahasa pemrograman yang kini dikembangkan oleh *MathWorks Inc.* menggabungkan proses pemrograman, komputasi, dan visualisasi melalui lingkungan kerja yang mudah digunakan. MATLAB juga memiliki keunggulan umum lainnya, seperti analisis dan eksplorasi data, pengembangan algoritma, pemodelan dan simulasi, visualisasi plot dalam bentuk 2D dan 3D, hingga pengembangan aplikasi antar muka grafis. Dalam ruang lingkup perguruan tinggi, MATLAB digunakan sebagai alat pembelajaran pemrograman matematika, teknik, dan sains pada level pengenalan dan lanjutan, sedangkan dalam dunia industri, MATLAB dipilih sebagai alat penelitian, pengembangan, dan analisis produk industri.

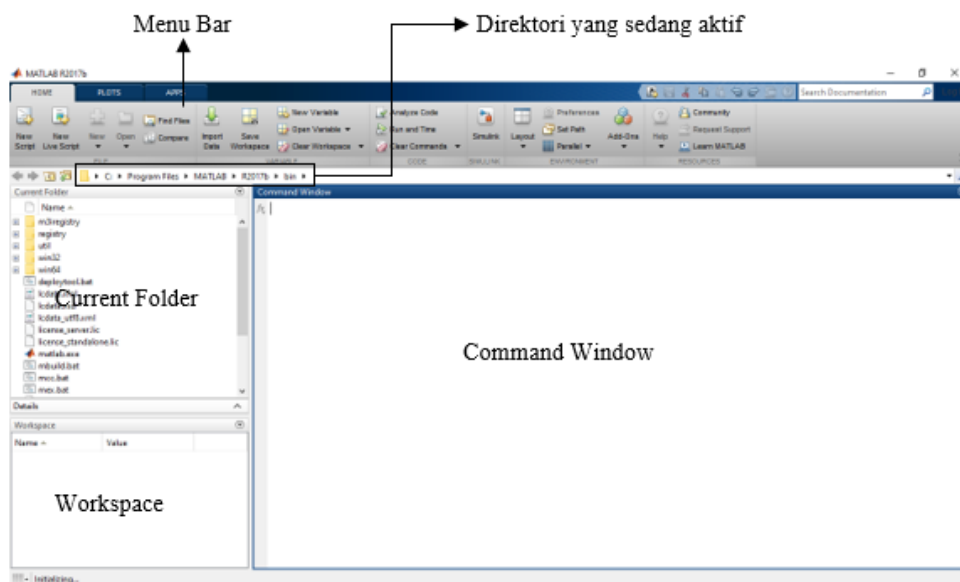
MATLAB dapat dioperasikan pada sistem operasi Windows, Linux, maupun macOS. Selain itu, MATLAB juga bisa dihubungkan dengan aplikasi atau bahasa pemrograman eksternal lainnya, seperti C, Java, .NET, dan Microsoft Excel. Dalam MATLAB tersedia pula kotak kakas (*toolbox*) yang dapat digunakan untuk aplikasi-aplikasi khusus, seperti pengolahan sinyal, sistem kontrol, logika fuzzy, jaringan saraf tiruan, optimasi, pengolahan citra digital, bioinformatika, simulasi, dan berbagai teknologi lainnya (Tjolleng, 2017).

#### 2.5.1.2 Lingkungan Kerja MATLAB

Setelah melakukan instalasi software MATLAB di komputer, ikon MATLAB akan muncul di desktop seperti pada gambar 2.11. Untuk memulai program, kita tinggal meng-klik ikon tersebut sebanyak dua kali. Selanjutnya, akan muncul tampilan layar utama MATLAB seperti pada gambar 2.12 berikut ini.



Gambar 2.11 ikon MATLAB



Gambar 2.12 Tampilan layar utama MATLAB

(Tjolleng, 2017) “Telah Diolah Kembali”

Secara umum, layar utama MATLAB memuat tiga bagian penting berikut:

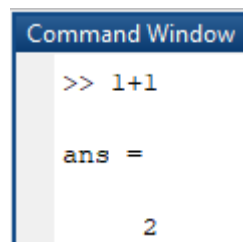
1. *Current Folder* (folder terkini), untuk mengakses berbagai file dalam direktori terkini yang sedang digunakan.
2. *Command window* (jendela perintah), untuk menuliskan perintah dalam bentuk sintaks program. Perintah yang dituliskan dapat berupa perhitungan sederhana, pemanggilan fungsi, demo program, dan lain sebagainya. Setiap penulisan perintah selalu diawali dengan prompt “>>”.
3. *Workspace* (jendela ruang kerja), untuk mengeksplorasi data yang telah dibuat atau diimpor dari file lain.

### 2.5.1.3 Prosedur Pemrograman MATLAB

Pemrograman MATLAB dapat dilakukan dengan dua cara, yaitu menggunakan jendela perintah atau menggunakan skrip M-File. Perhatikan contoh pembuatan program MATLAB dengan dua cara tersebut berikut ini.

#### 1. Menggunakan Jendela Perintah

Untuk membuat program dengan cara ini, kita hanya perlu mengetikkan perintah pada jendela perintah lalu menekan Enter. Selanjutnya, perintah tersebut akan secara langsung dieksekusi baris per baris oleh MATLAB. Simaklah contoh pada gambar 2.13 berikut.



```
Command Window
>> 1+1
ans =
    2
```

Gambar 2.13 Contoh Menggunakan Jendela Perintah *Command Window*

Perintah di atas akan ditampilkan dalam variabel `ans` sebagai singkatan dari `answer` atau jawaban. Jika kita mengakhiri perintah dengan tanda `(;)` maka MATLAB akan melakukan perhitungan tanpa menampilkan hasil pada jendela perintah. Untuk memanggil kembali perintah-perintah sebelumnya, gunakanlah tombol up dan down –arrow (`↑` dan `↓`) pada keyboard.

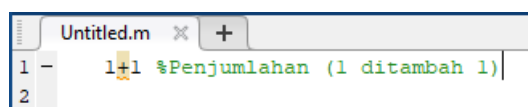
Selain itu, MATLAB juga menggunakan tanda `“%”` untuk menambahkan keterangan atau komentar terhadap suatu sintaks dalam program. Saat menjalankan program, MATLAB akan mengabaikan tanda `“%”` beserta semua keterangan yang mengikutinya. Perhatikanlah contoh penulisan komentar berikut.

```
>> 1+1 %Penjumlahan (1 ditambah 1)
ans =
     2
```

Untuk membersihkan jendela perintah, kita menuliskan perintah `clc`. Jika ingin membersihkan jendela ruang kerja, kita menuliskan perintah `clear`. Selanjutnya, untuk mengakhiri proses yang sedang berjalan pada MATLAB, bisa dilakukan dengan menuliskan perintah `exit` atau `quit` pada jendela perintah atau dengan meng-klik menu `exit` pada bagian kanan atas pada `menu bar`.

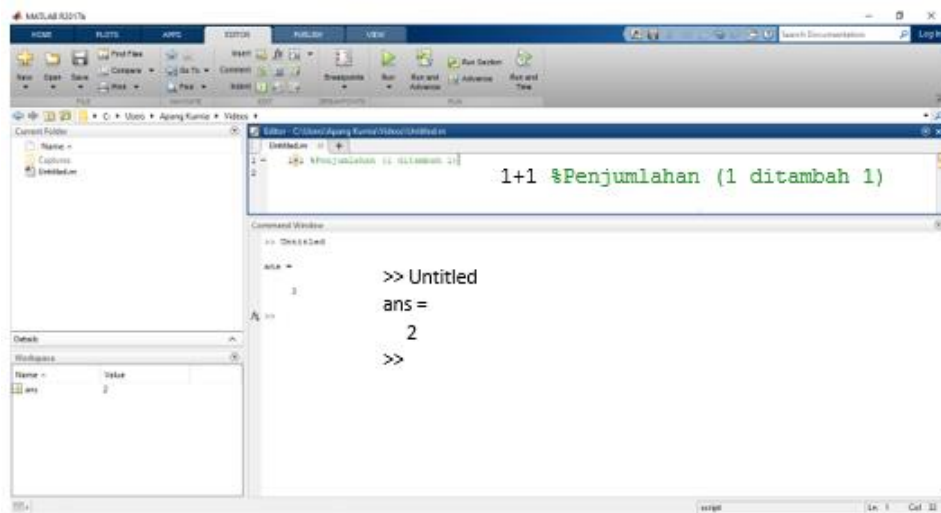
## 2. Menggunakan Skrip M-File

Penggunaan jendela perintah saat pemrograman terasa kurang efisien jika kita ingin melakukan suatu perintah berulang-ulang. Menjalankan perintah MATLAB dengan M-File memberi kemudahan, yaitu perintah-perintah tersebut dapat disimpan, dimodifikasi, dan dijalankan kapan saja kita inginkan. Untuk membuat skrip M-File, pilihlah Home → New Script pada `menu bar` atau tuliskan perintah edit di jendela perintah. Jendela editor akan muncul, lalu tulislah perintah yang ingin dijalankan. Penulisan perintah di jendela editor tidak diawali dengan prompt “>>”.



Gambar 2.14 Contoh Menggunakan Skrip M-File

Selanjutnya, simpanlah perintah tersebut dalam folder yang diinginkan dengan memilih ikon Save pada `menu bar`. File tersebut akan tersimpan dalam format MATLAB codes file, yaitu (\*.m) seperti contoh pada gambar 2.14. Terakhir, pilih Run pada `menu bar` dan program akan langsung dijalankan oleh MATLAB, seperti pada Gambar 2.15.



Gambar 2.15 Pemrograman dengan skrip M-File  
(Tjolleng, 2017) “Telah Diolah Kembali”

## 2.5.2 Simulink

### 2.5.2.1 Sekilas Tentang Simulink

Simulink<sup>®</sup> adalah paket perangkat lunak untuk pemodelan, simulasi, dan analisis sistem dinamis (The MathWorks Inc, 2004). Simulink merupakan ekstensi grafis ke MATLAB untuk pemodelan dan simulasi sistem. Dalam Simulink, sistem digambar di layar sebagai diagram blok. Banyak elemen diagram blok tersedia, seperti fungsi transfer, penjumlahan persimpangan, serta perangkat *input* dan *output* virtual seperti generator fungsi dan osiloskop. Perangkat virtual ini akan memungkinkan Anda untuk melakukan simulasi model yang akan Anda buat. Simulink terintegrasi dengan MATLAB dan data dapat dengan mudah ditransfer antar program.



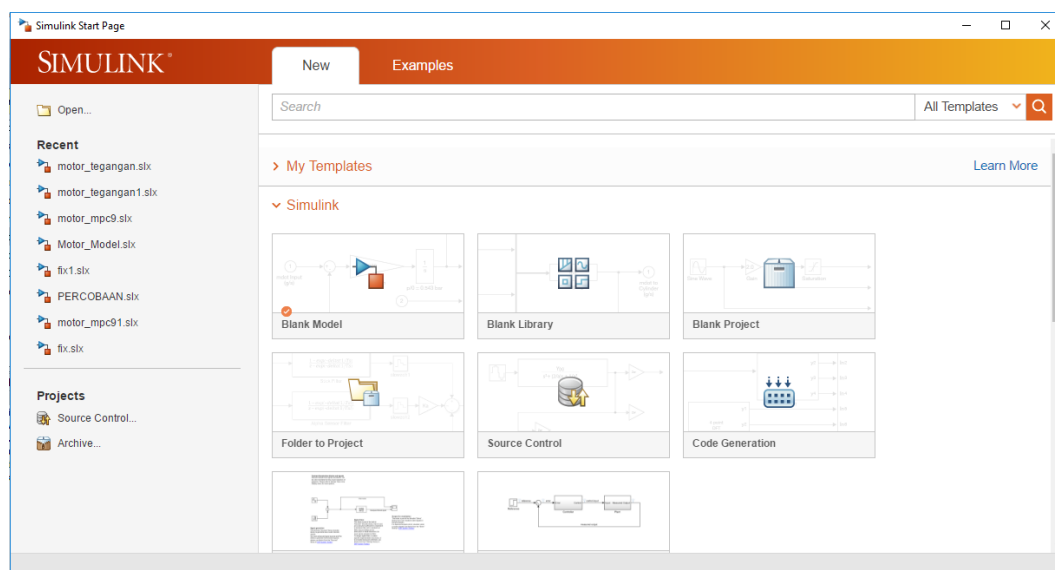
### 2.5.2.2 Lingkungan Kerja Simulink

Untuk menjalankan Simulink, terlebih dahulu harus memanggil MATLAB, dan pastikan Simulink terinstall pada sistem. Terdapat dua acara untuk masuk pada lingkungan kerja Simulink. Pertama bisa menggunakan *command window* dengan mengetik `simulink`. Cara kedua yaitu dengan mengklik ikon Simulink pada *menu bar*, berikut merupakan letak ikon Simulink pada gambar 2.16.



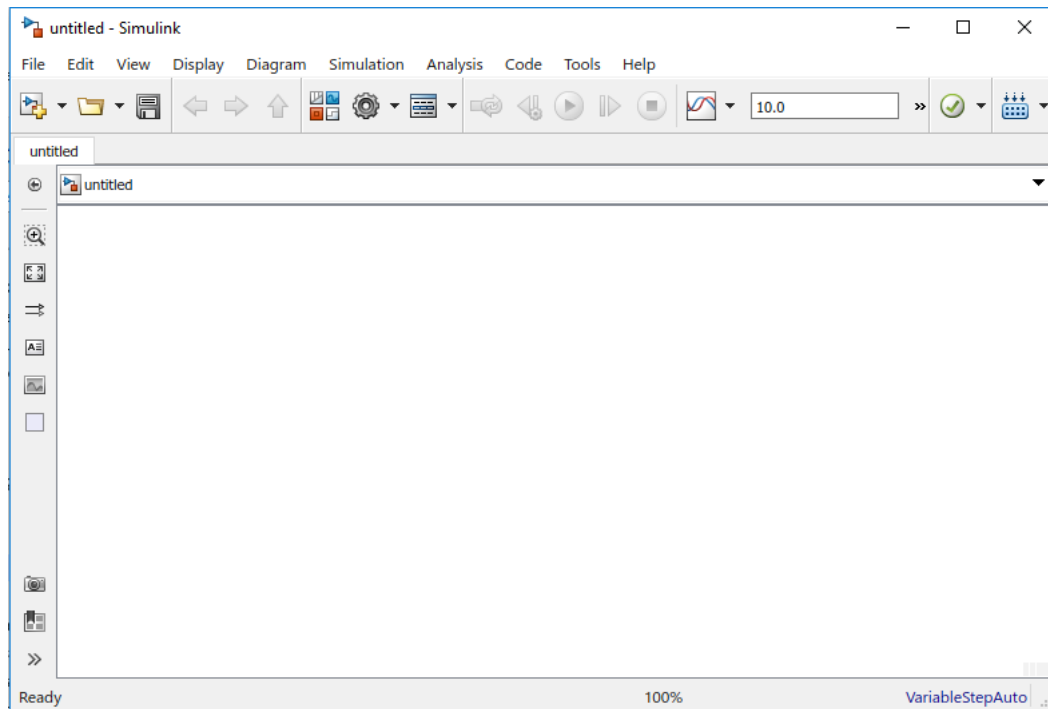
Gambar 2.16 Posisi dari ikon Simulink  
(Sianipar, 2015) “Telah Diolah Kembali”

Setelah perintah Simulink dieksekusi, maka akan muncul tampilan *simulink start page* seperti gambar 2.17 dan pilih *blank model* untuk memulai kerja dengan keadaan kerja kosong.



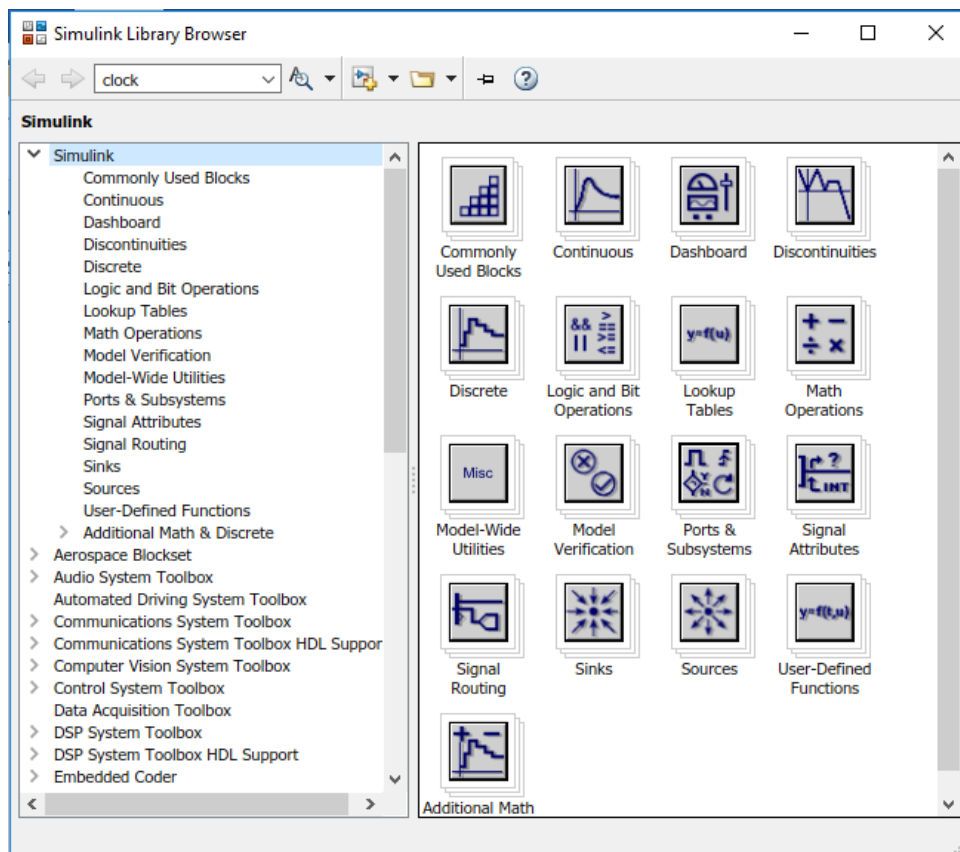
Gambar 2.17 Tampilan *Simulink Start Page*

### 2.5.2.3 Prosedur Pemodelan Simulink



Gambar 2.18 Tampilan Kerja Simulink

Gambar 2.18 merupakan tampilan kerja Simulink, dimana tampilan ini adalah tempat blok diagram dirangkai dan di simulasikan. Untuk memulai sebuah pemodelan di Simulink, tentu perlu adanya blok komponen, maka klik ikon *library browser* pada *menu bar* Simulink dan muncul tampilan seperti gambar 2.19. Pada gambar tersebut, sisi kiri dikenal dengan *tree pane*, untuk menampilkan semua pustaka Simulink yang terinstall. Sisi kanannya dikenal sebagai *contents pane*, untuk menampilkan blok-blok yang ada di dalam pustaka yang terseleksi pada *tree pane*. Setelah tampil *library browser*, maka pilih blok komponen yang dibutuhkan.



Gambar 2.19 Simulink *Library Browser*

(Sianipar, 2015) “Telah Diolah Kembali”

## 2.6 Daya Listrik

Daya Listrik atau dalam Bahasa Inggris disebut dengan Electrical Power adalah jumlah energi yang diserap atau dihasilkan dalam sebuah sirkuit/rangkaian (Weedy *et al.*, 2012). Sumber energi seperti tegangan listrik akan menghasilkan daya listrik sedangkan beban yang terhubung dengannya akan menyerap daya listrik tersebut (Adiyansa and Haryudo, 2013). Berikut merupakan rumus daya nyata atau aktif.

$$P = V \times I \quad (2.14)$$

Dengan efisiensi

$$\eta = \left| \frac{P_{out}}{P_{in}} \times 100\% \right| \quad (2.15)$$

Dimana:

- $P$  = Daya Aktif (W)
- $V$  = Tegangan (V)
- $I$  = Arus (A)
- $\eta$  = Efisiensi (%)
- $P_{in}$  = Daya Aktif Masuk (W)
- $P_{out}$  = Daya Aktif Keluar (W)

