

## BAB II TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Pemodelan dan simulasi telah banyak dilakukan untuk mengetahui performa maksimal berbagai macam kendaraan mulai dari kendaraan konvensional, kendaraan bertenaga listrik sampai kendaraan bertenaga surya, dan masih banyak yang lainnya. Ada beberapa penelitian terdahulu yang bisa dijadikan referensi untuk penelitian tentang pemodelan dan simulasi kendaraan ini.

Pertama pada tahun 2009 Grundit dan Jansson [6] melakukan pemodelan pada *hybrid electric vehicle* menggunakan *MATLAB/Simulink* dan mensimulasikan performanya untuk *Shell Eco Marathon 2009*. Pada penelitian ini sistem pada *hybrid vehicle* dibagi menjadi tiga subsistem yang terdiri dari *environment*, *smarter driver*, dan *smarter vehicle*. Subsistem *environment* memodelkan *drive cycle* yang direpresentasikan dengan vektor kecepatan dan kemiringan jalan. Subsistem *smarter driver* memberikan informasi mengenai kecepatan kendaraan yang sekaligus menjadi *input* untuk subsistem *smarter vehicle*. Pada subsistem *smarter vehicle* merepresentasikan blok diagram untuk gaya traksi kendaraan dan *drivetrain*. Pemodelan kendaraan didapatkan dari data-data kendaraan kemudian diinterpretasikan ke dalam bentuk *Simulink*, setelah itu disimulasikan dan dapat diketahui hasilnya. Dari data-data hasil simulasi dapat dianalisis dan dievaluasi apakah model yang dibangun sudah sesuai serta sudah dapat diaplikasikan dalam keadaan sebenarnya.

Selanjutnya S. Kaloko dkk. [7] membuat model *electrical vehicle* menggunakan *software MATLAB/Simulink* untuk mendapatkan daya terbaik dalam sistem manajemen energi. Pada penelitian ini dimodelkan mobil listrik jenis *urban concept* yang sesuai dengan keadaan geografis Indonesia. Komponen-komponen dari kendaraan ini meliputi motor listrik sebagai penggerak utama, dan baterai sebagai sistem penyimpanan energi. Motor listrik digunakan sebagai penggerak karena memiliki banyak keuntungan antara lain, lebih kuat, lebih murah, dapat digunakan untuk berkendara dengan kecepatan konstan, inersia besar, dan tidak membutuhkan perawatan secara rutin. Model dari motor listrik lebih *flexible* dibandingkan dengan mesin konvensional. Sedangkan baterai yang digunakan

masih menggunakan *Lead Acid Battery*, karena mudah ditemukan dengan harga yang relatif murah. Ini alasan mengapa tipe baterai jenis ini dipilih selain karena tipe baterai ini dapat digunakan untuk menghidupkan mesin yang membutuhkan arus yang tinggi. *Output* dari *Lead Acid Battery* berupa tegangan arus DC, sedangkan motor yang digunakan adalah motor induksi, sehingga diperlukan konverter daya untuk mengkonversi *output* dari baterai yang kemudian diteruskan ke motor induksi. Teknologi yang digunakan pada konverter daya adalah PWM *Converter*. Berdasarkan dari variabel dan parameter dari kendaraan yang kemudian dimodelkan menggunakan *MATLAB/Simulink* maka didapat *output* dari simulasi berupa daya dan arus *input* dari motor. Dari persamaan-persamaan yang diperoleh dari perhitungan dinamika kendaraan dapat dimodelkan kedalam *software MATLAB/Simulink* yang merepresentasikan kendaraan. *Input* yang diperlukan adalah model dari akselerasi, kecepatan, dan kemiringan jalan yang mengeluarkan *output* berupa daya yang dikeluarkan motor. Berdasarkan dari data-data parameter dan variabel dari kendaraan dapat diketahui juga gaya yang dibutuhkan untuk menggerakkan *prototype* kendaraan.

Menurut penelitian yang dilakukan oleh Salem [8] sistem pada mobil listrik dibagi menjadi dua subsistem, yaitu sistem kelistrikan pada kendaraan dan sistem dinamika kendaraan, keduanya dimodelkan berdasarkan semua gaya yang bekerja pada kendaraan. Kedua subsistem dipasangkan dengan kecepatan putaran roda melalui karakteristik motor listrik dan gaya-gaya yang bekerja meliputi gaya traksi, dan torsi. Sistem kontrol yang tepat terdiri dari dua *loop*, kontrol kecepatan dan kontrol arus didesain, diuji, dan diverifikasi menggunakan *software MATLAB/Simulink*. Pemodelan dan solusi kontrol untuk mobil listrik yang kemudian disimulasi menggunakan *MATLAB/Simulink*. Simulasi menggunakan *open loop* model dengan *step input* akan menghasilkan kecepatan linier, kecepatan angular, percepatan linier, arus, dan torsi.

## **2.2 Pemodelan dan Simulasi**

Model merupakan representasi sistem dalam kehidupan nyata yang menjadi fokus perhatian dan menjadi pokok permasalahan. Pemodelan dapat didefinisikan sebagai proses pembentukan model dari sistem tersebut dengan menggunakan bahasa formal tertentu. Simulasi adalah sebuah model matematika yang

menjelaskan tingkah laku sebuah sistem dalam beberapa waktu dengan mengobservasi tingkah laku sebuah model matematika [9]. Dalam melakukan simulasi terdapat beberapa langkah yang perlu dilakukan antara lain :

1. Pendefinisian sistem yaitu penentuan batasan sistem dan identifikasi variabel yang signifikan.
2. Formulasi model yaitu merumuskan hubungan antar komponen-komponen model.
3. Pengambilan data yaitu identifikasi data yang diperlukan oleh model sesuai dengan tujuan pembuatan model.
4. Pembuatan model dalam penyusunan model perlu disesuaikan dengan jenis bahasa simulasi yang akan digunakan.
5. Verifikasi model yaitu proses pengecekan terhadap model apakah sudah bebas dari *error*.
6. Validasi model merupakan proses pengujian terhadap model apakah model yang dibuat sudah sesuai dengan sistem nyatanya [9].

### **2.3 Mobil Listrik**

Mobil listrik merupakan kendaraan yang menggunakan satu atau lebih motor listrik sebagai penggerak utama kendaraan. Adapun kendaraan yang menggunakan penggabungan sistem antara motor listrik dan *internal combustion engine* disebut sebagai mobil *hybrid*. Untuk saat ini, mobil listrik mempunyai jarak tempuh yang lebih rendah apabila dibandingkan dengan mobil *hybrid*. Tidak hanya jarak tempuh, akan tetapi performa mesin juga kendaraan *hybrid* memiliki kemampuan yang lebih baik dan mendekati kemampuan dari kendaraan ICE.

#### **2.3.1. Jenis Kendaraan Mobil Listrik Secara Umum**

Di masa yang akan datang, akan muncul berbagai macam tipe kendaraan mobil listrik, hal ini tentunya dikarenakan hasil dari produksi massal mobil listrik nantinya, yang berasal dari asosiasi produsen mobil dan usaha tim riset dari seluruh dunia. Mobil listrik dibagi menjadi dua tipe utama, yaitu: *All Electric Vehicle* atau yang biasa disebut *Battery Electric Vehicle* (BEV) dan *Hybrid Electric Vehicle* (HEV) yang menggunakan kombinasi kinerja antara motor listrik dan ICE sebagai tenaga pendorongnya.

### 2.3.2. Sejarah *All Electric* atau *Battery Electric Vehicle* (BEV)

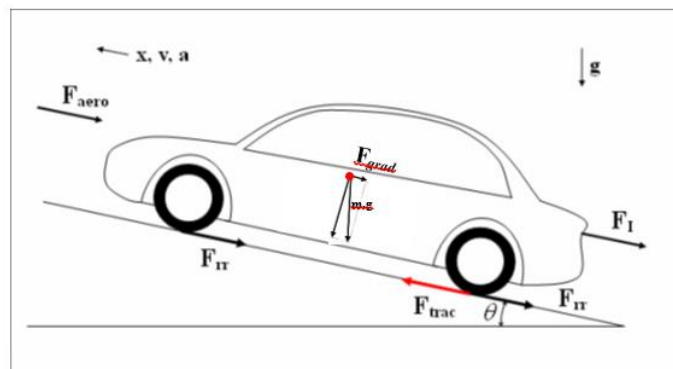
BEV telah ditemukan sejak tahun 1834. Pada akhir dari abad ke-19, sejumlah perusahaan memproduksi kendaraan listrik di Amerika, Inggris, dan Perancis. Akan tetapi, dikarenakan lemahnya perkembangan teknologi baterai dan juga perkembangan dari ICE yang lebih signifikan, kendaraan listrik mulai pudar kepopulerannya pada tahun 1930. Namun, pada tahun 1970, dimotori oleh isu energi krisis global, kendaraan listrik mulai populer kembali. Saat ini, BEV lebih digunakan untuk kendaraan-kendaraan dengan basis kendaraan kecil dengan batas tempuh yang pendek dikarenakan oleh keterbatasan kemampuan baterai [10].

### 2.3.3. Sejarah dari *Hybrid Electric Vehicle* (HEV)

Pada tahun 1898, Doktor Jerman bernama Dr. Ferdinand Porsche membuat mobil pertamanya, yang dinamai *Lohner Electric Chalse*. Sedangkan mobil kedua *Porsche* itu sendiri adalah sebuah mobil *hybrid*, dengan menggunakan mesin ICE untuk menyuplai listrik pada motor-motor listrik yang terdapat pada hub ban. Dengan menggunakan tenaga baterai tanpa mesin ICE itu sendiri, kendaraan ini sudah bisa mencapai jarak sekitar 65 km. Pada tahun 1900 an, perusahaan-perusahaan mobil di Amerika telah membuat 1681 mesin dengan tenaga *steam*, 1575 mobil listrik dan 936 mobil dengan tenaga mesin petrol [10].

## 2.4 Dinamika Kendaraan

Untuk mengembangkan model dan simulasi digunakan persamaan-persamaan dinamika kendaraan. Gaya-gaya yang bekerja pada sebuah kendaraan yang sejajar dengan arah akselerasi dapat dijabarkan pada **Gambar 2.1**. Dalam keadaan diam di *track* lurus dan datar saat mesin dihidupkan, gaya traksi ( $F_{trac}$ ) dari *drivetrain* akan dihambat oleh gaya inersia ( $F_{inertia}$ ) dari kendaraan, *aerodynamic force* ( $F_{aero}$ ), *rolling resistance force* ( $F_{rr}$ ), dan *gravitational force* ( $F_{grad}$ ).



**Gambar 2.1** Gaya yang Bekerja Pada Mobil [11]

### 2.4.1 Gaya Hambat Aerodinamika (*Aerodynamic Force*)

*Aerodynamic force* merupakan gaya yang menghambat gaya dorong kendaraan karena arah dari *aerodynamic force* berlawanan dengan arah kecepatan kendaraan. Gaya hambat aerodinamis yang dialami kendaraan dapat dihitung melalui persamaan (2.1).

$$F_{aero} = \frac{1}{2} \cdot \rho \cdot C_D \cdot A \cdot v^2 \quad (2.1)$$

Dimana:

$F_{aero}$  = aerodynamic force (N)

$\rho$  = massa jenis udara (kg/m<sup>3</sup>)

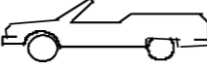






$C_D$  = koefisien *drag*

$A$  = luas area depan kendaraan (m<sup>2</sup>)

$v$  = kecepatan relatif angin terhadap kendaraan (m/s)

Nilai dari koefisien *drag* ( $C_D$ ) didapat dengan melakukan simulasi CFD (*Computational Fluid Dynamics*), Beberapa nilai  $C_D$  untuk berbagai bentuk penampang kendaraan ditunjukkan pada **Tabel 2.1**.

**Tabel 2.1** *Coefficient Drag* Berbagai Bentuk Kendaraan [12]

<i>Vehicle Type</i>	<i>Coefficient of Aerodynamic Resistance</i>
 <i>Open convertible</i>	0.5-0.7
 <i>Van body</i>	0.5-0.7
 <i>Ponton body</i>	0.4-0.55
 <i>Wedge-shaped body; headlamps and bumpers are integrated into the body, covered underbody, optimized cooling air flow</i>	0.3-0.4
 <i>Headlamp and all wheels in body, covered underbody</i>	0.2-0.25
 <i>K-shaped (small breakway section)</i>	0.23
 <i>Optimum streamlined design</i>	0.15-0.2
<i>Trucks, road trains</i>	0.8-1.5
<i>Buses</i>	0.6-0.7
<i>Streamlined buses</i>	0.3-0.4
<i>Motorcycles</i>	0.6-0.7

### 2.4.2 Gaya Hambat Rolling (*Rolling Resistance*)

*Rolling resistance* merupakan gaya hambat yang terjadi karena adanya gesekan antara ban dengan jalan. *Rolling resistance* dari kendaraan dapat dihitung dengan persamaan (2.2).

$$F_{rr} = C_{rr} \cdot m \cdot g \cdot \cos \theta \quad (2.2)$$

Dimana:

$F_{rr}$  = *rolling resistance* (N)

$C_{rr}$  = koefisien *rolling resistance*

$m$  = massa kendaraan (kg)

$g$  = gravitasi (9,81 m/s<sup>2</sup>)

$\theta$  = sudut kemiringan (°)

Koefisien *rolling resistance* ( $C_{rr}$ ) merupakan fungsi dari bahan ban, struktur ban, suhu ban, tekanan angin ban, geometri tapak, kekasaran jalan, bahan jalan, dan ada tidaknya cairan di jalan. Nilai dari koefisien *rolling resistance* ( $C_{rr}$ ) pada berbagai jalan dapat dilihat pada **Tabel 2.2**.

**Tabel 2.2** *Rolling Resistance Coefficient* Berbagai Permukaan [12]

<i>Conditions</i>	<i>Rolling Resistance Coefficient</i>
<i>Car tires on concrete or asphalt</i>	0,013
<i>Car tires on rolled gravel</i>	0,02
<i>Tar macadam</i>	0,025
<i>Unpaved road</i>	0,05
<i>Field</i>	0,1–0,35
<i>Truck tires on concrete or asphalt</i>	0,006–0,01
<i>Wheels on rail</i>	0,001–0,002

### 2.4.3 Gravitational Force

*Gravitational force* adalah gaya hambat yang terjadi karena adanya sudut tanjakan pada jalan. Dengan adanya gaya hambatan tanjakan maka beban kendaraan akan bertambah akibat pengaruh gaya gravitasi bumi. Besar gaya hambat akibat sudut tanjak dapat dihitung dengan persamaan (2.3).

$$F_{grad} = m \cdot g \cdot \sin \theta \quad (2.3)$$

Dimana:

$F_{grad}$  = *gravitational force* (N)

$m$  = massa kendaraan (kg)

- $g$  = gravitasi (9,81 m/s<sup>2</sup>)  
 $\theta$  = sudut kemiringan (°)

#### 2.4.4 Akselerasi Kendaraan

Akselerasi memiliki pengaruh besar pada performa kendaraan. Untuk melakukan akselerasi, diperlukan gaya lebih karena harus melawan gaya inersia kendaraan. Gaya inersia karena akselerasi ini dipengaruhi oleh besar akselerasi ( $a$ ) dan massa kendaraan. Besar gaya inersia dapat dihitung dengan persamaan (2.4).

$$F_{inertia} = m \cdot a \quad (2.4)$$

Dimana:

$$F_{inertia} = \text{inertia force (N)}$$

$$m = \text{massa kendaraan (kg)}$$

$$a = \text{percepatan (m/s}^2\text{)}$$

Pada sebuah kendaraan yang dipacu dengan akselerasi tertentu, gaya traksi yang harus dicapai oleh mesin dapat dijabarkan sebagai penjumlahan dari gaya-gaya hambat, dan inersia akibat percepatan. Daya traksi merupakan perkalian gaya dengan kecepatan. Besar gaya traksi dan daya traksi kendaraan dapat dihitung dengan persamaan (2.5) dan (2.6).

$$F_{trac} = F_{aero} + F_{rr} + F_{grad} + F_{inertia} \quad (2.5)$$

$$P_{trac} = (F_{aero} + F_{rr} + F_{grad} + F_{inertia}) \cdot v \quad (2.6)$$

Dimana:

$$F_{trac} = \text{traction force (N)}$$

$$F_{aero} = \text{aerodynamic force (N)}$$

$$F_{rr} = \text{rolling resistance force (N)}$$

$$F_{grad} = \text{gravitational force (N)}$$

$$F_{inertia} = \text{inertia force (N)}$$

$$v = \text{kecepatan kendaraan (m/s)}$$

#### 2.4.5 Kondisi Kecepatan Konstan

Pada kondisi kecepatan konstan, gaya inersia diabaikan karena percepatan bernilai nol. Besar daya traksi kendaraan ketika berada dalam kecepatan konstan dapat dihitung dengan persamaan (2.7).

$$P_{trac} = F_{trac} \cdot v \quad (2.7)$$

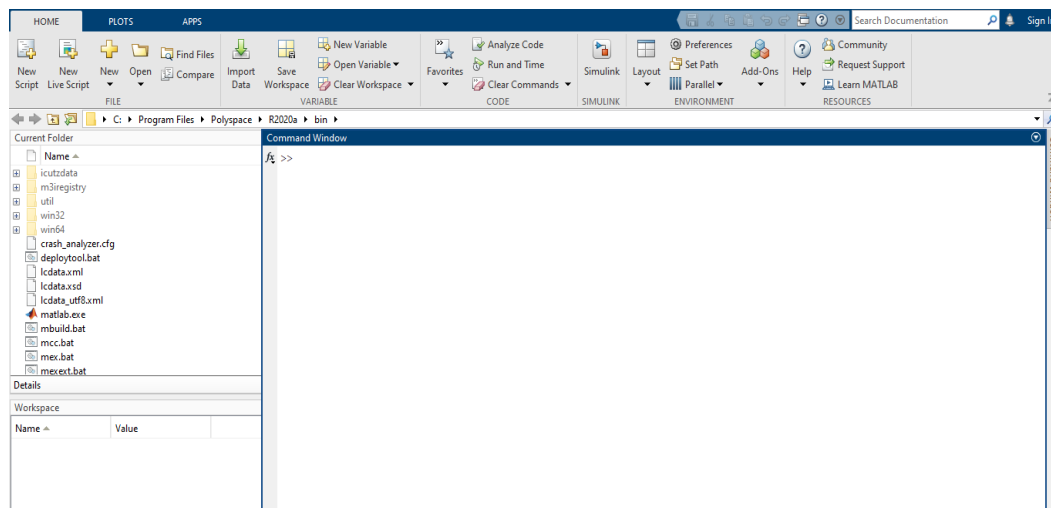
#### 2.4.6 Kondisi Deselerasi

Kendaraan dapat berada dalam kondisi deselerasi baik saat mesin dimatikan maupun saat mesin hidup. Saat kondisi mesin mati, mesin tidak tersambung dengan roda penggerak sehingga tidak menimbulkan kerugian daya karena *engine break*. Untuk dapat terus bergerak dan melawan gaya hambat, kendaraan hanya digerakkan oleh inersia kendaraan itu sendiri. Besar daya traksi kendaraan ketika berada dalam kondisi deselerasi dapat dihitung dengan persamaan (2.8).

$$m a = F_{aero} + F_{rr} + F_{grad} - F_{trac} \quad (2.8)$$

### 2.5 MATLAB/Simulink

*MATLAB* merupakan singkatan dari *Matrix Laboratory* yang berarti bahasa pemrograman tingkat tinggi dengan kinerja tinggi untuk komputasi masalah teknik. *MATLAB* mengintegrasikan komputasi, visualisasi, dan pemrograman dalam sebuah lingkungan tunggal. *MATLAB* memberikan sistem interaktif yang menggunakan konsep *array/matrix* sebagai variabel elemennya tanpa membutuhkan pendeklarasian *array*. **Gambar 2.3** merupakan tampilan awal dari *software MATLAB*.



**Gambar 2.3** Tampilan Awal *MATLAB* R2020a

*MATLAB* dikembangkan oleh *Mathwork* pada tahun 1970. Aplikasi *MATLAB* ini banyak digunakan dalam bidang yang membutuhkan perhitungan matematika yang rumit, dimana seluruh operasi perhitungan dalam *MATLAB*

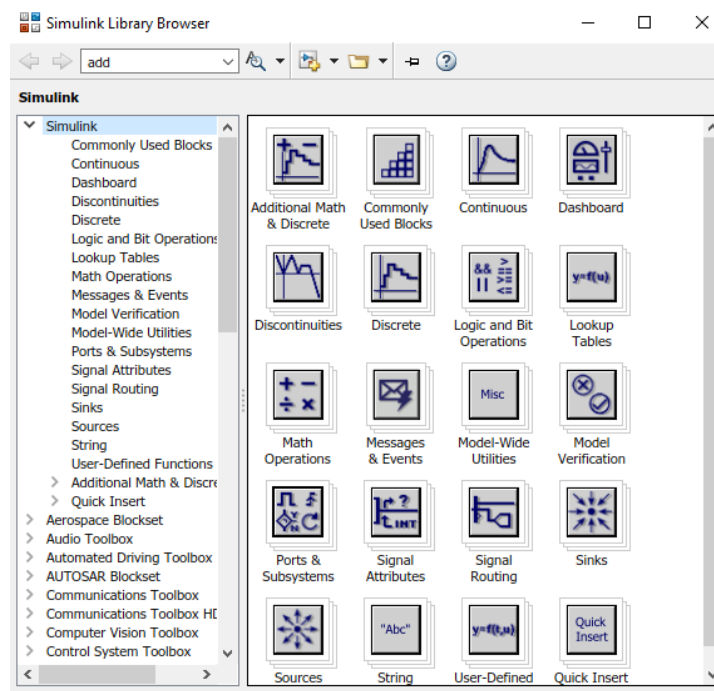


berupa operasi matrik. *MATLAB* dapat menampilkan hasil perhitungan dalam bentuk *plot* grafik. Pada perangkat lunak *MATLAB* terdapat beberapa bagian penting yang digunakan dalam menjalankan program, yaitu:

1. *Command window* digunakan untuk mengetik fungsi yang diinginkan.
2. *Command history* berfungsi agar fungsi yang telah digunakan sebelumnya dapat digunakan kembali.
3. *Workspace* digunakan untuk membuat variabel yang ada dalam *MATLAB*.

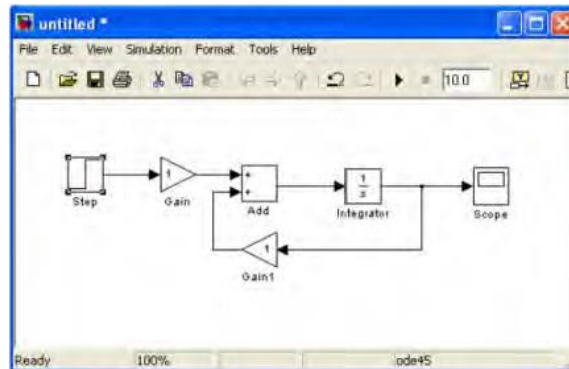
*Simulink* adalah sebuah kumpulan aplikasi dalam *MATLAB* untuk melakukan *modeling*, simulasi, dan untuk melakukan analisis dinamik pada suatu sistem. Program *Simulink* memudahkan *user* untuk membuat suatu simulasi lebih interaktif. Tiruan sistem diharapkan mempunyai perilaku yang sangat mirip dengan sistem fisik. Jika digunakan dengan benar, simulasi akan membantu proses analisis dan desain sistem.

*Simulink* dalam *MATLAB* juga dapat menunjukkan performa sistem dalam bentuk dua ataupun tiga dimensi. Dalam perancangannya *user* menjadi mudah karena adanya blok-blok diagram yang dapat dengan mudah diatur sedemikian rupa sesuai dengan model matematis dari sistem atau *plant* yang akan dikendalikan. Tampilan kotak dialog *Library Browser* dari *Simulink* dapat dilihat pada **Gambar 2.5**.



**Gambar 2.5** Kotak Dialog *Simulink Library Browser*

Membuat model pada *Simulink* dilakukan dengan cara membuat model baru pada *library browser*. Kemudian memasang blok-blok yang diinginkan dan menyambungkannya dengan konektor. Contoh sederhana dari blok diagram dapat dilihat pada **Gambar 2.6**.



**Gambar 2.6** Contoh Blok *Simulink*

Setelah membuat blok, langkah selanjutnya adalah dengan mengedit *properties* dari masing-masing blok. Setelah itu dilakukan *configuration parameters* untuk mengatur simulasi yang sesuai model. Langkah terakhir adalah dengan menjalankan simulasi sehingga akan didapatkan hasil yang diinginkan.