

BAB II

DASAR TEORI

2.1 Tinjauan Pustaka

Pada penelitian terkait perancangan AGV *forklift* dilakukan studi literatur terlebih dahulu untuk mendapatkan *research gap* dari penelitian yang telah ada sebelumnya. Berikut merupakan hasil dari *research gap* yang telah dilakukan:

Tabel 2. 1 Tinjauan Pustaka (Sumber: Analisis Penulis,2026)

No	Penulis	Judul	Hasil
1	Kurniawan, (2025)	Implementasi Simulasi Metode <i>Field Oriented Control</i> (Foc) Untuk Pengaturan Kecepatan Konstan Motor BLDC Pada Beban Statis	Metode FOC unggul dari segi respons, kestabilan, dan efisiensi, serta cocok untuk sepeda listrik modern dengan kebutuhan performa tinggi. PWM tetap menjadi solusi praktis untuk produk dengan keterbatasan biaya. Penelitian ini menyarankan pengembangan lebih lanjut untuk implementasi FOC secara nyata di perangkat keras sepeda listrik [6].
2	Sholikha et al., (2022)	PENGATURAN KECEPATAN MOTOR BLDC SENSORLESS MENGGUNAKAN FOC	Metode FOC cukup presisi pada kecepatan tinggi, pada penelitian selanjutnya diharapkan dapat melanjutkan untuk menerapkannya ke motor sebenarnya [7].

3	Putranto, (2021)	PERBANDINGAN ANTARA <i>FIELD ORIENTED CONTROL</i> MENGUNAKAN SINUSOIDAL <i>PULSE WIDTH MODULATION</i> (SPWM) DAN <i>SPACE VECTOR PULSE WIDTH MODULATION</i> (SVPWM) DALAM PENGENDALIAN <i>BRUSHLESS DIRECT CURRENT MOTOR</i> (BLDC)	Berdasarkan hasil simulasi menggunakan MATLAB, FOC tanpa maupun terdapat beban memiliki kemampuan yang baik. Namun, pengembangan selanjutnya dapat membuat rancang bangun pengendali motor BLDC [8] .
4	Jian et al., (2024)	<i>BLDC Motor Zero-Speed Startup Position Sensorless FOC Control</i>	Hasil penelitian berdasarkan simulasi online menunjukkan FOC realible dalam mengontrol BLDC, namun penelitian yang dilakukan masih secara online dan pengembangan selanjutnya dapat diterapkan pada desain BLDC [9].
5	Sharma, (2024)	<i>Implementation of Sensor-less FOC Multi-Motor Control of BLDC for Vehicle Applications</i>	Penelitian menunjukkan FOC merupakan solusi yang menjanjikan dan hemat biaya, penelitian dilakukan telah pada BLDC akan tetapi belum diterapkan dengan pengujian membawa beban [10].

6	Abirami, M., & Balakrishnan, P. (2023)	<i>DTC-FOC Hybrid Controller to Control the Speed and Torque of BLDC Motor</i>	Berdasarkan penelitian ini dihasilkan berhasil mengurangi <i>ripple</i> dan respon cepat. Namun, pengujiannya masih berbasis simulink belum diterapkan pada BLDC secara nyata [11].
7	Pradhana et al., (2024)	Desain dan Implementasi Pengaturan Kecepatan Motor BLDC Dengan Metode FOC Menggunakan Fuzzy	Hasil pengujian simulasi dengan metode FOC dan kontrol fuzzy pada motor BLDC dapat memberikan hasil eror yang kecil terhadap setting point. hasil yang didapatkan dengan setting point terendah. Namun pengujian dilakukan pada MATLAB [12].

Berdasarkan *research gap* yang telah dilakukan didapatkan hasil bahwa telah banyak penelitian sebelumnya terkait control kecepaataan menggunakan metode pengendalian FOC. Akan tetapi, hanya sebatas dilakukan padaa simulasi *online*. Belum terdapat penerapan FOC secara langsung pada *hardware*.

2.2 Autonomous Guided Vehicle (AGV)

Pada dunia industri khususnya logistik, penggunaan *Autonomous Guided Vehicle (AGV)* telah banyak digunakan dalam membantu proses operasional industri. AGV dimanfaatkan karena dapat meningkatkan efisiensi pada operasional logistik, mengurangi risiko kecelakaan kerja akibat *human error*, terutama dalam pemindahan beban berat di area gudang serta dapat mengurangi biaya operasional perusahaan jangka panjang [13].



Gambar 2. 1 AGV [14]

Autonomous Guided Vehicle (AGV) merupakan robot otonom pemindah barang pada proses *material handling*. AGV dinilai paling efektif karena tidak membutuhkan pengemudi dalam menjalankannya [15]. AGV dirancang untuk distribusi barang dari satu titik ke titik lainnya secara teratur dan terarah [16]. Melalui hal itu, pada penerapan AGV tidak dibutuhkan tenaga manusia, karena robot mampu mengetahui titik mana yang akan dituju secara akurat. Seiring dengan perkembangan teknologi saat ini, penggunaan AGV bertransformasi menjadi *smart AGV*, dimana menggunakan sistem navigasi tidak lagi bergantung pada jalur tetap seperti *magnetic tape* atau *line follower*, akan tetapi telah memanfaatkan teknologi navigasi berbasis sensor seperti LiDAR, kamera, serta integrasi *Simultaneous Localization and Mapping (SLAM)*. Terdapat tiga jenis AGV yang dibedakan berdasarkan fungsinya yaitu *towing vehicle*, *unit load vehicle* dan *pallet truck*. *Towing vehicle* merupakan jenis AGV pertama yang dipakai untuk menarik jenis *trailer*, selanjutnya *unit load vehicle* yang merupakan tipe AGV untuk pengangkatan dan pemindahan barang secara otomatis dan yang terakhir *pallet truck* yang difungsikan untuk dapat mengangkat palet dari lantai ke arah tempat yang tinggi.

2.3 Kinematika Ackerman

Kinematika Ackerman merupakan salah satu metode gerak kendaraan beroda, dimana cara kerjanya menggunakan perubahan sudut roda depan untuk mendapatkan gerakan belok, sedangkan roda belakang tetap pada posisi sejajar dengan sumbu longitudinal kendaraan. Kinematika Ackerman mempunyai prinsip apabila kendaraan belok, seluruh roda berpotongan pada titik yang sama yaitu Instantaneous Center of Rotation (ICR). Hal itu menyebabkan roda depan bagian dalam mempunyai sudut kemudi lebih besar disbanding roda depan bagian luar. Hal tersebut bertujuan agar mengurangi terjadinya selip dan menghasilkan gerakan kendaraan yang stabil [17]. Kinematika Ackerman banyak diterapkan pada kendaraan otonom karena mempunyai karakteristik gerak seperti kendaraan konvensional. Model Ackerman dapat direpresentasikan kedua roda depan sebagai roda kemudi dan kedua roda belakang sebagai roda penggerak.

2.4 *Brushless Direct Current (BLDC)*

Motor *Brushless Direct Current* (BLDC) merupakan jenis motor listrik *synchronous* yang umum dipakai di industri. Pada dasarnya BLDC berbeda dengan motor DC, dimana BLDC tidak membutuhkan sikat (*brushless*) pada sistem komutasi melainkan memakai komutasi elektronik.

Tabel 2. 2 Spesifikasi BLDC

(Sumber: berdasarkan spesifikasi Changzhou Hanghong model HH06502)

Parameter	Spesifikasi
Jenis motor	BLDC hub motor
Model rujukan	HH06502
Tegangan nominal	36 V
Daya nominal	300 W
Kecepatan tanpa beban	550-600 rpm
Arus tanpa beban	$\leq 0,9$ A
Diameter roda	6,5 inci
Penggunaan	Hoverboard / swing car

Secara teknis BLDC lebih unggul dibandingkan motor DC karena BLDC dinilai memiliki efisiensi tinggi, kekuatan torsi serta saat dioperasikan memiliki tingkat kebisingan yang rendah [18].

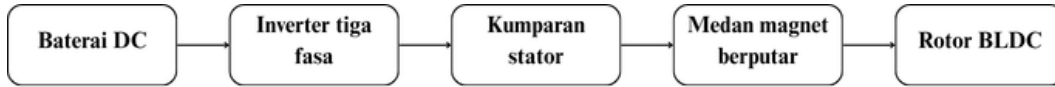


Gambar 2. 2 Roda BLDC [19]

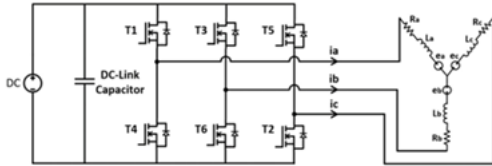
Motor BLDC sering disebut sebagai motor magnet tetap yang memerlukan suplai tegangan DC melalui semikonduktor selama mengubah arah rotasi dalam menggerakkan motor. Secara arsitektur motor BLDC terdiri dari rotor berbentuk magnet permanen sehingga kutubnya tetap sedangkan stator berbentuk belitan kawat email sehingga kutub magnet dapat berubah tergantung polaritas arus belitan stator yang diberikan. Masing masing memiliki jumlah stator duabelas sedangkan pada rotor berjumlah delapan kutub magnet permanen. Pada pengaturan putaran motor BLDC membutuhkan keterangan untuk mengetahui letak rotor. Berdasarkan hal itu dibutuhkan sensor pendeteksi posisi rotor yaitu *hall effect sensor* yang merupakan sensor untuk mendeteksi posisi magnet [20].

2.4.1 Prinsip Kerja Motor BLDC

Motor BLDC bekerja berdasarkan hubungan medan magnet stator dengan medan magnet pada rotor. Cara kerjanya yaitu sumber tegangan DC baterai akan diubah menjadi tegangan fasa U, V, dan W oleh inverter tiga fasa. Selanjutnya MOSFET pada inverter diaktifkan mengikuti urutan tertentu sehingga membuat medan magnet stator berputar. Sementara itu, rotor magnet permanen mengikuti perubahan arah medan magnet tersebut sehingga menghasilkan gerakan berputar. Proses pemberian tegangan pada kumparan BLDC dilakukan secara elektronik melalui inverter, sedangkan pergantian fasa ditentukan berdasarkan posisi rotor [21].



Gambar 2. 3 Prinsip Kerja Motor BLDC (Sumber: Analisis Penulis, 2026)



Gambar 2. 4 Rangkaian Inverter 3 Fasa [22]

Pada BLDC yang berputar, posisi rotor bisa diketahui melalui tiga sensor Hall yang menghasilkan kombinasi sinyal digital. Pada penerapan metode FOC, arus tiga fasa ditransformasikan menjadi arus sumbu d dan q . Arus I_d berhubungan dengan fluks magnet, sedangkan I_q menjadi arus utama pembentuk torsi. Kedua arus tersebut akan diatur secara terpisah, sehingga FOC menghasilkan perubahan torsi yang lebih halus disbanding dengan komutasi *Six-Step* [23].

2.4.2 Model Elektrik Motor BLDC

Model elektrik motor BLDC menjelaskan hubungan antara tegangan fasa, arus, resistansi, induktansi, dan gaya gerak listrik atau *back electromotive force* (back-EMF). Pada setiap fasa stator dapat direpresentasikan sebagai rangkaian resistansi dan induktansi yang dihubungkan dengan sumbu *back-EMF*. Setiap tiga fasa digunakan untuk menganalisis perubahan arus, tegangan, torsi, dan kecepatan motor [21]. Bentuk umum persamaan tegangan satu fasa dinyatakan melalui Persamaan (2.1)

$$v_k = R_s i_k + L_s \frac{di_k}{dt} + e_k \quad (2.1)$$

Keterangan:

v_k = tegangan fasa-ke k (V)

R_s = resistansi kumparan stator per fasa (Ω)

i_k = arus fasa-ke k (A)

L_s = induktansi stator per fasa (H)

e_k = *back-EMF* fasa-ke k (V)

k = fasa a, b, c

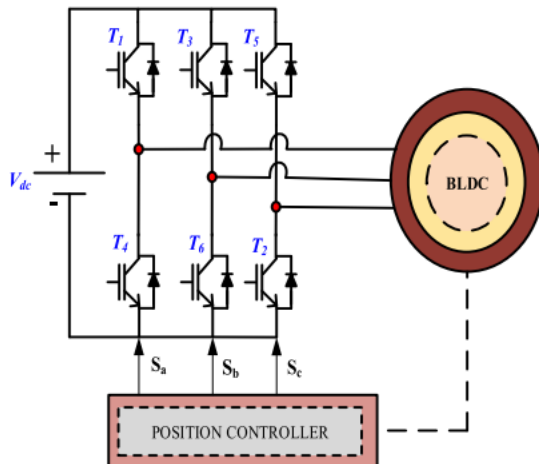
Persamaan tersebut menunjukkan bahwa tegangan yang diberikan oleh inverter digunakan untuk mengendalikan perubahan arus pada induktansi, *back-EMF* akibat perputaran rotor. Nilai *back-EMF* akan meningkat seiring dengan peningkatan kecepatan sudut rotor dan dapat dinyatakan melalui Persamaan (2.2)

$$e = K_e \omega_m \quad (2.2)$$

Keterangan:

K_e : konstanta *back-EMF*

ω_m : sebagai kecepatan sudut mekanis rotor (rad/s)



Gambar 2. 5 Controller Motor BLDC [23]

Sementara itu, daya listrik motor dihitung menggunakan Persamaan (2.3)

$$P_{in} = V_{dc} I_{dc} \quad (2.3)$$

Keterangan:

P_{in} : daya listrik (W)

V_{dc} : tegangan baterai atau DC-link

I_{dc} : arus baterai (A)

2.4.3 Hubungan Torsi dan Dinamika Rotor

Torsi elektromagnetik motor BLDC diperoleh dari fluks magnet permanen rotor dengan arus stator. Pada metode FOC, arus sumbu q digunakan sebagai pembentuk torsi, sedang sumbu d pada umumnya dijaga mendekati nol untuk motor magnet permanen. Hubungan torsi dan arus dapat dinyatakan melalui Persamaan (2. 4)

$$T_e \approx K_t I_q \quad (2. 4)$$

Keterangan:

T_e = torsi elektromagnetik motor (N.m)

K_t = konstanta torsi motor (N.m/A)

I_q = arus sumbu q (A)

Apabila konstanta torsi motor belum diketahui, torsi mekanis dapat diperkirakan dari daya dan kecepatan putar menggunakan Persamaan (2.5)

$$T_m = \frac{P_m}{\omega_m} \quad (2.5)$$

Kecepatan sudut didapatkan dari kecepatan motor dalam rpm melalui Persamaan (2. 6)

$$\omega_m = \frac{2\pi n}{60} \quad (2. 6)$$

Dengan demikian:

$$T_m = \frac{60P_m}{2\pi n} \quad (2.7)$$

Keterangan:

T_m : torsi mekanis (N.m)

P_m : daya motor (W)

n : kecepatan rotor (rpm)

Motor dengan daya 300 W, torsi tidak dapat ditentukan hanya berdasarkan nilai daya karena nilai kecepatan motor juga diperlukan. Perubahan kecepatan rotor karena selisih antara torsi elektromagnetik dan torsi beban dinyatakan melalui persamaan dinamika mekanis:

$$T_e - T_L = J \frac{d\omega_m}{dt} + B\omega_m \quad (2.8)$$

Dengan T_L sebagai torsi beban (N.M), J sebagai momen inersia rotor dan beban (kg.m^2), serta B sebagai koefisien (N.m.s/rad). Persamaan tersebut menjelaskan apabila torsi beban bertambah, motor membutuhkan torsi elektromagnetik yang lebih besar untuk mempertahankan kecepatan. Apabila peningkatan T_e belum mencukupi, kecepatan rotor akan menurun [21][23].

2.5 Hall Effect Sensor

Hall effect sensor jenis sensor posisi yang seringkali dipakai pada sistem penggerak motor BLDC maupun *Permanent Magnet Synchronous Motor* (PMSM) untuk mengetahui posisi rotor berdasarkan perubahan medan magnet yang dihasilkan oleh magnet permanen rotor. Sensor ini memiliki prinsip kerja berdasarkan pada sinyal Hall, dimana tampak tegangan Hall di suatu material semikonduktor yang dialiri arus listrik dan diberikan medan magnet tegak lurus terhadap arah arus. Perubahan medan magnet yang disebabkan oleh pergerakan rotor akan menghasilkan perubahan sinyal digital yang menunjukkan posisi sudut rotor [24].

Pada penerapannya di BLDC tiga fasa, biasanya dipakai tiga *Hall effect sensor* yang diletakkan dengan jarak 120° electrical untuk mendapatkan keterangan posisi rotor secara terpisah. Gabungan keluaran ketiga sensor tersebut dihasilkan enam kondisi logika yang menggambarkan enam bagian komutasi dalam satu siklus elektrik. Keterangan tersebut dipakai oleh pengendali untuk menentukan urutan sakelar inverter sehingga arus tiga fasa bisa diberikan sesuai posisi rotor [25]. Berbeda dengan pemakaian *encoder* atau *resolver*, sensor Hall mempunyai kelebihan berupa ukuran yang lebih kecil, biaya rendah, dan rangkaian pengimplementasian yang lebih sederhana [24].

Akan tetapi, pemakaian sensor Hall mempunyai keterbatasan karena resolusi posisi yang didapatkan sangat rendah. Pada sistem BLDC/PMSM, transformasi sensor sekadar memberikan keterangan posisi sekitar 60° elektrik di setiap bagian sehingga diperlukan perkiraan tambahan untuk mendapatkan posisi rotor dengan resolusi yang lebih tinggi pada pengaplikasian kendali presisi seperti FOC [24]. Selain itu juga, ketidaktepatan pemakaian sensor Hall dapat menimbulkan ketidakseimbangan waktu komutasi, distorsi arus fasa, dan peningkatan *torque ripple* yang disebabkan oleh perubahan interval konduksi inverter yang kurang simetris [26]. Oleh karena itu, sensor Hall masih banyak digunakan untuk solusi yang efektif pada sistem

pengendalian motor karena mampu memberikan keterangan posisi yang lengkap dengan biaya yang rendah dan kemampuan yang baik, terutama pada pengaplikasian industry yang membutuhkan sisrem penggerak yang responsif [25].

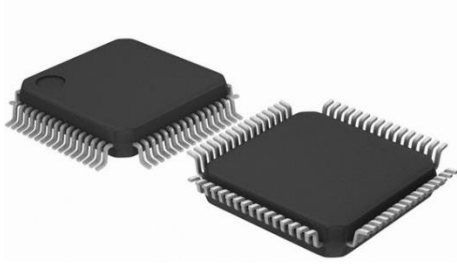
2.6 *Controller Hoverboard* STM32F103RCT6

Controller hoverboard merupakan jenis unit pengendali daya yang mengintegrasikan mikrokontroller, inverter tiga fasa, rangkaian pengukuran arus dan tegangan, sensor posisi rotor, dan antarmuka komunikasi.

Tabel 2. 3 Fungsi Peripheral STM32F103RCT6 pada Controller Hoverboard

Bagian	Fungsi pada controller
TIM1 dan TIM8	Menghasilkan enam sinyal PWM komplementer untuk inverter motor kanan dan kiri
ADC1 dan ADC2	Mengubah sinyal analog sensor arus dan tegangan DC-link menjadi data digital
DMA	Memindahkan hasil konversi ADC ke memori tanpa pemindahan berulang oleh CPU
GPIO dan sensor Hall	Menyediakan informasi sektor atau posisi listrik rotor
USART	Menerima perintah kecepatan atau kemudi dan mengirimkan umpan balik

Jenis mikrokontroller yang banyak digunakan pada *mainboard* adalah STM32F103RCT6 yang termasuk dari STM32F103 dengan prosesor inti ARM Cortex-M3 dan mempunyai kemampuan pemrosesan hingga 72 MHz. Jenis mikrokontroller ini memiliki memori Flash hingga 256 Kb dan SRAM sebesar 48 Kb, dan dilengkapi dengan ADC 12-bit, serta dua *advanced-control* timer TIM1 dan TIM8 yang dirancang sebagai kendali motor [27]. Kedua timer yang ada tersebut menjadi pendukung untuk pembangkitan PWM komplementer dua kanal motor. Sementara itu, ADC serta DMA akan menyediakn jalur akuisisi arus sebagai umpan balik pengendali.



Gambar 2. 6 STM32F103RCT6 [27]

Firmware pada github EFeru menjelaskan bahwa *mainboard* hoverboard berbasis STM32F103RCT6 mampu menjalankan komutasi konvensional, kendali sinusoidal serta FOC. Pada metode FOC, sistem dapat dijalankan dalam mode tegangan, kecepatan, atau torsi serta dilengkapi dengan pembatas arus dan kecepatan [28]. Pada penelitian ini, *controller* sebagai pengendali *low level* yang menerima perintah dari STM32F411CEU6, membaca arus dan posisi rotor, menjalankan loop FOC, menghasilkan PWM untuk inverter.

2.6.1 Inverter Tiga Fasa dan PWM Komplementer

Inverter tiga fasa berfungsi dalam mengubah tegangan DC baterai menjadi tegangan tiga fasa U, V, dan W untuk mengendalikan motor BLDC. Setiap kaki inverter terdiri dari MOSFET *high-side* dan *low-side* yang tidak boleh aktif secara bersamaan, oleh karena itu menggunakan PWM komplementer dan menyisipkan *dead-time* di antara proses untuk menonaktifkan satu MOSFET dan pengaktifan MOSFET lainnya. *Dead-time* berfungsi untuk mencegah kondisi *shoot-through* yang dapat menyebabkan *short* pada satu kaki inverter. STM32F103RCT6 menyediakan keluaran komplementer, *dead-time* yang dapat diprogram, *brake* input, dan mode PWM *edge-aligned* maupun *center-aligned* pada TIM1 dan TIM8 [27][29].

Mode *center-aligned* sesuai untuk pengendalian motor karena pencacah timer bergerak naik dan turun dalam satu periode. Pola ini menghasilkan tepi pensaklaran yang menyediakan titik sampling konsisten terhadap periode PWM [30]. Jika clock timer dinyatakan sebagai f_{TIM} , nilai *prescaler* sebagai PSC , dan nilai *auto-reload register* sebagai ARR , frekuensi PWM *center-aligned* dihitung dengan Persamaan (2. 9)

$$f_{PWM} = \frac{f_{TIM}}{2(PSC + 1)(ARR + 1)} \quad (2. 9)$$

Keterangan:

f_{PWM} = frekuensi sinyal PWM yang dihasilkan (Hz)

f_{TIM} = frekuensi clock timer (Hz)

PSC = nilai prescales timer

ARR = Auto Reload Register

Konfigurasi clock, PSC dan ARR menghasilkan frekuensi PWM yang sama pada TIM1 dan TIM8.

2.6.2 ADC dan Pengukuran Arus

Metode FOC membutuhkan arus fasa aktual sebagai respon balik loop arus. Tegangan keluaran *current-sense* amplifier dibaca oleh ADC 12-bit, sehingga nilai digital berada pada rentang 0 sampai 4095. Sebelum digunakan, nilai ADC perlu dikoreksi menggunakan offset yang diperoleh ketika arus motor bernilai nol. Offset diperlukan karena rangkaian pengondisi sinyal umumnya menggeser titik nol ke tegangan tertentu agar arus dua arah tetap dapat dibaca oleh ADC. Hubungan antara data ADC dan tegangan masukan ditunjukkan oleh Persamaan (2.10), sedangkan konversi tegangan menjadi arus ditunjukkan oleh Persamaan (2.11)

$$V_{ADC} = \frac{N_{ADC} \times V_{REF}}{4095} \quad (2.10)$$

$$I_{phase} = \frac{(N_{ADC} - N_{offset}) V_{REF}}{4095 G R_{shunt}} \quad (2.11)$$

Keterangan:

N_{ADC} = nilai hasil konversi

N_{offset} = nilai ADC ketika arus motor nol

V_{REF} = tegangan referensi ADC

G = penguatan rangkaian *current-sense* amplifier

I_{phase} = arus fasa motor dalam ampere

R_{shunt} = nilai resistansi shunt dalam ohm

4095 = nilai maksimum ADC 12-bit

Selanjutnya untuk sistem tiga fasa tanpa jalur netral, jumlah arus direpresentasikan sama dengan hukum Kirchoff seperti pada Persamaan (2.12). Apabila dua arus fasa terukur, arus ketiga dapat direkonstruksi menggunakan Persamaan (2.13).

$$i_u + i_v + i_w = 0 \quad (2.12)$$

$$i_w = -(i_u + i_v) \quad (2.13)$$

Keterangan:

I_u : arus fasa U

I_v : arus fasa V

I_w : arus fasa W

Pengukuran arus yang cepat dan akurat merupakan bagian penting dari regulasi arus pada penggerak AC [31].

2.6.3 Sinkronisasi Timer-ADC-DMA

Sinkronisasi Timer, ADC, dan DMA memungkinkan proses pengambilan data berlangsung secara teratur dan konsisten. Timer menentukan waktu pengambilan sampel sesuai dengan periode PWM, trigger dari Timer memulai konversi ADC. Setelah konversi selesai, DMA memindahkan hasil pembacaan ke buffer memori secara otomatis. Mekanisme ini membantu mengurangi perubahan waktu sampling serta mengurangi beban CPU karena pemindahan data tidak perlu dilakukan secara manual [30]. Pada pengendalian dua motor, TIM1 dan TIM8 membutuhkan basis waktu yang konsisten sehingga pembaruan duty cycle dan pembacaan arus kedua motor menjadi lebih akurat. Salah satu timer dapat digunakan sebagai master dan timer lainnya sebagai slave melalui jalur trigger internal. ADC1 dan ADC2 kemudian dikonfigurasi untuk memperoleh pasangan sampel pada kondisi yang sama, sedangkan DMA menyimpan urutan kanal ke dalam buffer yang tetap.

2.6.4 Komunikasi Serial *Controller*

Komunikasi serial digunakan sebagai jalur pertukaran data antara Jetson Orin Nano dan controller hoverboard. Jetson Orin Nano mengirimkan perintah gerak menuju controller melalui komunikasi UART dengan level logika TTL, sedangkan controller mengirimkan kembali data telemetry untuk proses pemantauan. Pada sisi controller, komunikasi tersebut menggunakan peripheral USART3 pada STM32F103RCT6 melalui pin PB10 sebagai TX dan PB11 sebagai RX.

Peripheral USART3 dapat bekerja dengan mekanisme interrupt dan DMA sehingga proses penerimaan data serial dapat dilakukan secara efisien tanpa mengganggu eksekusi loop kendali motor secara langsung [30][27].

2.7 *Field-Oriented Control (FOC)*

Field Oriented Control (FOC) atau biasa disebut sebagai *vector control*, merupakan metode pengendalian motor yang memisahkan komponen arus stator menjadi dua sumbu ortogonal pada kerangka acuan yang berputar bersama rotor. Komponen arus sumbu langsung (I_d) berkaitan dengan pembentukan fluks, sedangkan komponen arus sumbu kuadratur (I_q) digunakan dalam pembentukan torsi elektromagnetik. Pemisahan tersebut memungkinkan pengaturan fluks dan torsi dilakukan secara terpisah sehingga pengendalian motor BLDC dapat dilakukan menggunakan pengendali arus yang lebih [12][32].

Pada penelitian ini, tahapan FOC menggunakan arus fasa yang dihasilkan dari rangkaian *current sensing* pada *controller* hoverboard. Proses konversi data ADC menjadi arus telah dijelaskan pada Persamaan (2. 10) dan Persamaan (2.11). Sementara itu, hubungan jumlah arus tiga fasa dan rekonstruksi arus fasa dinyatakan pada Persamaan (2.12) dan Persamaan (2.13). Pada algoritma FOC secara berurutan arus tiga fasa diproyeksikan dalam kerangka $\alpha - \beta$ melalui transformasi Clarke, selanjutnya diputar ke kerangka $d - q$ melalui transformasi Park menggunakan sudut elektrik rotor. Selisih antara arus referensi dan arus aktual diproses oleh pengendali PI arus untuk menghasilkan tegangan referensi pada sumbu d dan q . Pada prosesnya tegangan dibatasi sesuai kemampuan *DC-link*, kemudian dikembalikan lagi kedalam kerangka $\alpha - \beta$ dan tiga fasa melalui inverse Park dan inverse Clarke. Tahapan akhir dalam algoritma FOC yaitu *Space Vector Pulse Width Modulation* (SVPWM) membentuk *duty cycle* inverter tiga fasa [32][33]. Pada sistem kendali kecepatan, loop kecepatan berada di luar loop arus dan menghasilkan referensi i_q , sedangkan i_d tetap bernilai 0.

2.7.1 *Transformasi Clarke*

Transformasi *Clarke* digunakan untuk mengubah arus tiga fasa i_a, i_b, i_c , stasioner menjadi koordinat dua fasa ortogonal i_α, i_β pada kerangka stasioner $\alpha - \beta$. Pada tahapan ini, arus masukan bukan lagi data ADC, tetapi arus fasa yang telah diproses dengan Persamaan (2. 10) - (2.13). Bentuk dari transformasi *clarke* dapat dinyatakan melalui Persamaan (2.14)

$$i_{\alpha} = \frac{2}{3} \left(i_a - \frac{1}{2} i_b - \frac{1}{2} i_c \right) \quad (2.14)$$

$$i_{\beta} = \frac{2}{3} \left(\frac{\sqrt{3}}{2} i_b - \frac{\sqrt{3}}{2} i_c \right)$$

Keterangan:

i_{α} : arus pada fasa A motor (A)

i_b : arus pada fasa B motor (A)

i_c : arus pada fasa C motor (A)

i_{α} : komponen arus pada sumbu α dalam kerangka koordinat stasioner (A)

i_{β} : komponen arus pada sumbu β dalam kerangka koordinat stasioner (A)

kedua komponen arus tersebut digunakan sebagai masukan transformasi Park.

2.7.2 Transformasi Park

Pada transformasi Park akan memutar vektor arus dari kerangka stasioner $\alpha - \beta$ ke kerangka $d - q$ yang bergerak sinkron dengan sudut elektrik rotor θ_e . Pada tahap ini dihasilkan komponen i_d dan i_q yang relatif konstan pada kondisi yang konsisten sehingga dapat dikendalikan menggunakan control PI [32][34]. Hubungan transformasi Park ditunjukkan pada Persamaan (2.15)

$$\begin{aligned} i_d &= i_{\alpha} \cos \theta_e + i_{\beta} \sin \theta_e \\ i_q &= -i_{\alpha} \sin \theta_e + i_{\beta} \cos \theta_e \end{aligned} \quad (2.15)$$

di mana θ_e adalah posisi sudut elektrik rotor yang dipakai oleh algoritma FOC dan harus konsisten dengan urutan fasa serta arah putaran motor. Pada *controller* yang memakai Hall effect sensor, keterangan posisi rotor pada firmware dipakai untuk menentukan sudut elektrik yang dibutuhkan pada proses transformasi. Kesalahan arah, urutan Hall, atau offset susut menyebabkan i_d dan i_q tidak terpisah secara benar dan teratur.

2.7.3 Pengendali Arus d-q

Pengendalian arus $d - q$ merupakan tahapan pada metode FOC untuk membandingkan arus actual hasil transformasi Park dengan nilai referensinya. Nilai referensi i_d ditetapkan nol, sedangkan i_q merupakan pembentuk torsi dan diperoleh dari loop kecepatan. Error arus pada masing-masing sumbu dinyatakan oleh Persamaan (2.16)

$$\begin{aligned} e_d &= i_d - i_d \\ e_q &= i_q - i_q \end{aligned} \quad (2.16)$$

Nilai eror yang dihasilkan diproses oleh control PI untuk menghasilkan referensi tegangan v_d dan v_q . Bentuk umum dari control PI arus dinyatakan melalui Persamaan (2.17)

$$\begin{aligned} v_d &= K_{pd}e_d + K_{id} \int e_d dt \\ v_q &= K_{pq}e_q + K_{iq} \int e_q dt \end{aligned} \quad (2.17)$$

Parameter K_{pd} dan K_{pq} merupakan gain proporsional, sedangkan K_{id} dan K_{iq} merupakan gain integral di masing-masing sumbu.

2.7.4 Pembatasan Vektor Tegangan

Tegangan referensi hasil control PI tidak dapat dinaikkan tanpa batas karena inverter hanya memperoleh power dari tegangan DC-link. Besar vector tegangan referensi pada kerangka $d - q$ dapat dihitung melalui Persamaan (2.18)

$$V^2 = \sqrt{(v_d)^2 + (v_q)^2} \quad (2.18)$$

Pada SVPWM, batas vector tegangan dapat dinyatakan terhadap tegangan DC-link. Jika V^2 melebihi batas yang dipakai pada firmware, komponen v_d^2 dan v_q^2 disesuaikan secara proporsional agar arah vektor tetap sama. Bentuk umum pembatas ditunjukkan oleh Persamaan (2.19)

$$k_v = \min\left(1, \frac{V_{max}}{V^2}\right) \quad (2.19)$$

Nilai V_{max} pada penerapannya harus mengikuti batas modulasi yang benar-benar dipakai oleh firmware, bukan ditentukan dari asumsi.

2.7.5 Inverse Park dan Iverse Clarke

Setelah tegangan pada sumbu $d - q$ diperoleh, v_d dan v_q ditransformasikan kembali ke kerangka stasioner $\alpha\beta$ melalui inverse Park. Transformasi ini menggunakan sudut elektrik rotor yang sama dengan transformasi Park sehingga konvensi tanda harus dipertahankan secara konsisten. Persamaan inverse Park ditunjukkan pada Persamaan (2.20)

$$\begin{aligned} V_\alpha &= V_d \cos \theta_e - V_q \sin \theta_e \\ V_\beta &= V_d \sin \theta_e + V_q \cos \theta_e \end{aligned} \quad (2.20)$$

Komponen V_α dan V_β selanjutnya dikonversi menjadi referensi tegangan tiga fasa melalui inverse Clarke. Bentuk persamaan inverse Clarke ditunjukkan oleh Persamaan (2.21)

$$\begin{aligned} V_a &= V_\alpha \\ V_b &= -\frac{1}{2}V_\alpha + \frac{\sqrt{3}}{2}V_\beta \\ V_c &= -\frac{1}{2}V_\alpha - \frac{\sqrt{3}}{2}V_\beta \end{aligned} \quad (2.21)$$

Hasil dari inerse Clarke adalah referensi tegangan tiga fasa yang belum menjadi sinyal MOSFET. Nilai tersebut masih harus diproses oleh modulator untuk menghasilkan *duty cycle* pada masing-masing inverter.

2.7.6 *Space Vector PWM (SVPWM)*

Space Vector Pulse Width Modulation (SVPWM) digunakan untuk membentuk *duty cycle* inverter tiga fasa dari vektor tegangan referensi. Pada implementasinya, SVPWM dapat direpresentasikan menggunakan penambahan komponen common-mode terhadap tiga referensi fasa. Nilai maksimum dan minimum dari V_a , V_b , dan V_c digunakan untuk mendapatkan tegangan offset sebagaimana dalam Persamaan(2.22) [32] [33].

$$V_{off} = \frac{\max(V_a, V_b, V_c) + \min(V_a, V_b, V_c)}{2} \quad (2.22)$$

Setelah nilai offset diperoleh, referensi fasa dapat disesuaikan terhadap tegangan DC-link untuk membentuk *duty cycle*. Bentuk persamaan umumnya dapat dinyatakan oleh Persamaan (2.23)

$$d_x = \frac{1}{2} + \frac{V_x - V_{off}}{V_{dc}} \quad (2.23)$$

Nilai *duty cycle* selanjutnya dipakai pada timer PWM komplementer.

2.7.7 *Arsitektur Cascade Kecepatan-Arus*

Sistem penggerak AGV menggunakan struktur cascade, merupakan loop kecepatan sebagai loop luar dan loop arus $d - q$ sebagai loop dalam. Loop kecepatan akan membandingkan kecepatan referensi n^2 dengan kecepatan aktual n . Error kecepatan dinyatakan pada Persamaan (2.24)

$$e_n = n^2 - n \quad (2.24)$$

Error kecepatan diproses oleh pengendali PI untuk menghasilkan referensi arus torsi i_q . Bentuk umum pengendali PI kecepatan ditunjukkan oleh Persamaan (2. 25)

$$i_q = K_{pn}e_n + K_{in} \int e_n dt \quad (2. 25)$$

Nilai referensi i_q selanjutnya dibatasi sesuai kemampuan motor dan *controller* sebelum menjadi masukan loop arus. Penurunan kecepatan akibat kenaikan beban akan meningkatkan error kecepatan dan selanjutnya loop luar menaikkan i_q agar loop arus menghasilkan torsi yang diinginkan.

2.8 Hubungan Beban, Torsi dan Arus Motor BLDC

Pada sistem penggerak AGV *forklift*, motor BLDC digunakan sebagai aktuator yang menghasilkan gaya dorong melalui roda penggerak. Torsi yang dihasilkan motor berhubungan dengan gaya dorong pada permukaan roda melalui jari-jari roda. Hubungan antara torsi motor dengan gaya dorong dapat dirumuskan sebagai berikut:

$$T = \frac{F_r}{n} \quad (2.26)$$

Dengan:

T = torsi yang dibutuhkan setiap roda penggerak (Nm)

F = gaya dorong (N)

r = jari-jari roda (m)

n = jumlah roda penggerak

Pada motor BLDC, hubungan antara torsi elektromagnetik dengan arus motor dinyatakan melalui *torque constant* sebagai berikut:

$$T_e = K_t I_q \quad (2.27)$$

Keterangan:

T_e = torsi elektromagnetik motor (Nm)

K_t = konstanta torsi motor (Nm/A)

I_q = komponen arus sumbu q

Pada algoritma FOC, arus I_q sebagai pembentukan torsi elektromagnetik sehingga perubahan kebutuhan torsi akibat perubahan beban dapat diamati melalui perubahan nilai I_q

2.9 Parameter Evaluasi Kinerja Sistem

Evaluasi kinerja sistem penggerak motor BLDC berbasis FOC dilakukan untuk mengetahui kemampuan sistem dalam menerapkan referensi pengendalian serta mempertahankan respon yang stabil. Parameter evaluasi pada penelitian ini difokuskan pada karakteristik respon arus dan kecepatan motor meliputi *rise time*, *settling time*, *overshoot*, *steady-state error*, *Root Mean Square Error* (RMSE), *speed drop*, dan *recovery time*. Parameter tersebut dipakai untuk mengevaluasi performa *inner-loop* arus FOC serta respon sistem penggerak terhadap perubahan setpoint [35].

Rise time merupakan waktu yang diperlukan sistem untuk mencapai rentang tertentu dari nilai akhir setelah diberikan perubahan referensi[32]. Pada penelitian ini, *rise time* digunakan untuk mengetahui kecepatan respon awal sistem dalam mengikuti perubahan setpoint. Persamaan *rise time* ditunjukkan sebagai berikut:

$$t_r = t_{90\%} - t_{10\%} \quad (2.28)$$

Sementara itu, *settling time* merupakan waktu yang diperlukan oleh sistem untuk mencapai kondisi yang stabil dan tetap berada pada batas toleransi terhadap nilai akhir [32]. Parameter ini digunakan untuk mengetahui kestabilan respon setelah terjadi perubahan referensi. Persamaan *settling time* sebagai berikut:

$$e(t) = r(t) - y(t) \quad (3.29)$$

Parameter lain yang digunakan untuk mengevaluasi besarnya penyimpangan maksimum terhadap nilai referensi adalah *overshoot* [32]. *Overshoot* menunjukkan kemampuan sistem dalam merespon perubahan referensi serta menjadi indikator adanya respon yang terlalu agresif. Persamaan *overshoot* sebagai berikut:

$$Overshoot(\%) = \frac{y_{max} - y_{ss}}{y_{ss}} \times 100\% \quad (2.30)$$

Selain itu, untuk mengetahui selisih antara nilai referensi dengan nilai aktual ketika sistem telah mencapai kondisi stabil digunakan *steady-state error* [32]. Nilai error yang kecil menunjukkan kemampuan sistem dalam melakukan tracking terhadap referensi. Berikut persamaan *steady-state error*:

$$e_{ss} = |r - y_{ss}|$$

Evaluasi tracking menggunakan error antara kecepatan referensi dan kecepatan aktual. Besarnya error rata-rata absolut dinyatakan sebagai persentase terhadap setpoint. Selain itu digunakan Root Mean Square Error (RMSE) untuk menunjukkan besarnya penyimpangan keseluruhan, overshoot untuk melihat nilai puncak di atas setpoint, dan persentase sampel yang berada di dalam pita $\pm 10\%$ setpoint untuk menunjukkan konsistensi tracking. Untuk sinkronisasi dua motor digunakan rata-rata selisih absolut ERPM kiri dan kanan

$$RMSE = \sqrt{\text{mean}[(ERPM_{\text{aktual}} - ERPM_{\text{ref}})^2]} \quad (2.31)$$

Pada pengujian respons kecepatan terhadap perubahan beban, parameter *speed drop* dan *recovery time* digunakan untuk mengetahui kemampuan sistem mempertahankan performa ketika terjadi gangguan.

$$Speed Drop(\%) = \frac{\omega_{ref} - \omega_{min}}{\omega_{ref}} \times 100\% \quad (2.32)$$

Speed drop menunjukkan penurunan kecepatan akibat perubahan beban, sedangkan *recovery time* menunjukkan waktu yang diperlukan sistem untuk kembali mendekati nilai referensi setelah gangguan terjadi.