

BAB II DASAR TEORI

2.1 Tinjauan Pustaka

Setelah melakukan studi literatur terhadap topik terkait baik dari jurnal dan *paper* nasional maupun internasional, didapati beberapa penelitian yang menjadi inspirasi oleh penulis. Berikut beberapa penelitian tersebut.

Tabel 2. 1 Tinjauan Pustaka

NO	Judul Penelitian	Keterangan	Target Pembaharuan
1	Sistem Kendali Kecepatan Motor BLDC Menggunakan PWM Berbasis Mikrokontroler Arduino Uno (Astuti dan Masdi, 2022)	Pengaturan kecepatan dilakukan menggunakan keypad untuk mengatur besaran nilai <i>Duty Cycle</i> yang mengirimkan data berbentuk ADC (<i>Analog to Digital Converter</i>) untuk mengontrol kecepatan motor <i>BLDC</i> [6].	Penelitian ini hanya mengontrol kecepatan motor BLDC secara <i>open loop</i> saja sehingga perlu penambahan kontrol <i>close-loop</i> .
2	<i>Control BLDC Motor Speed using PID Controller</i> (Mahmud et al., 2020)	Simulasi desain kontrol motor BLDC menggunakan <i>software</i> Matlab/Simulink. Lalu membandingkan kontrol PID dengan kontrol PI dan <i>fuzzy</i> untuk mengontrol kecepatan motor BLDC. Hasilnya kontroler PID lebih baik dari pada PI dan <i>fuzzy</i> dari segi <i>overshoot</i> dan <i>settling time</i> [5].	Penelitian ini terbatas pada analisa berbasis Matlab/Simulink saja tanpa diterapkan secara langsung pada perangkat keras. Sehingga perlu pengimplementasian secara langsung.

Tabel 2. 2 Lanjutan Tinjauan Pustaka

NO	Judul Penelitian	Keterangan	Target Pembaharuan
3	Alat Penstabil Kecepatan Motor BLDC Menggunakan Kontrol PID (Sutedjo <i>et al.</i> , 2023)	Menstabilkan kecepatan BLDC dengan kontrol PID dan Komutasi <i>Six step</i> , nilai <i>set point</i> yang dipakai ada di 500 RPM dengan $K_p = 4.9505$, $K_i = 0.266156$, $K_d = 18.4158$ [8].	Parameter respon waktu seperti, rise time, settling time, overshoot, dan steady-state error tidak di cantumkan dalam jurnal.
4	Efisiensi Penggunaan Motor Bldc Terhadap Beban Dan Kecepatan Sepeda Motor Listrik (Putro dan Ma'arif, 2024)	Melakukan analisis pada penggunaan BLDC 3000W di sepeda motor listrik untuk mengetahui bagaimana beban dan kecepatan mempengaruhi konsumsi energi dan performa motor. [9].	Penelitian hanya untuk pengujian kecepatan pada variasi beban secara <i>open-loop</i> tanpa menggunakan metode tambahan seperti <i>close-loop</i> .
5	Alat Kendali Kecepatan Motor Pada Penggerak Depan Sepeda Listrik (Fitriyah <i>et al.</i> , 2020)	Pengujian pada pengendalian kecepatan motor BLDC 350 Watt 36 Volt menggunakan throttle didapati kenaikan kecepatan pada motor berbanding lurus dengan kenaikan <i>throtle</i> gas [10].	Pengontrolan kecepatan dilakukan secara <i>open loop</i> nilai PWM ditentukan oleh <i>throttle</i> sehingga tidak ideal apabila terjadi gangguan.

Pada penelitian ini, untuk penggerak sepeda listrik memakai motor BLDC 800 watt yang telah dilengkapi oleh 3 buah sensor *hall effect* sebagai penentu posisi rotor dan membaca RPM dari motor. Untuk bagian catu daya memakai baterai LifePo4 48V berkapasitas 12Ah yang cukup besar untuk menjalankan motor. Lalu untuk sistem kontrolnya memakai mikrokontroler STM32 yang di tanamkan sistem kontrol PID dan komutasi *Six step*.

Sistem ini menggunakan *Inverter 3 fasa half Bridge* yang dibuat dengan menyusun 6 buah MOSFET IRFP250 untuk melakukan *switching* dengan urutan sesuai dengan komutasi *Six step* agar bisa membentuk gelombang AC untuk mengontrol motor BLDC, IC IR2110 digunakan sebagai sakelar untuk memerintahkan *gate* dari MOSFET agar terbuka atau menutup sehingga proses *switching* bisa dilakukan. STM32F401 sendiri digunakan sebagai otak yang mengontrol sistem *switching* dan membaca masukan dari *hall* sensor sehingga kecepatan dari BLDC dapat kontrol lewat PWM. Apabila terdapat selisih antara nilai PWM yang di berikan dengan kecepatan aktual, maka PID akan bekerja dengan memperbaiki *error* yang terjadi supaya kontrol dari motor BLDC bisa menjadi lebih *smooth* dan minim gangguan.

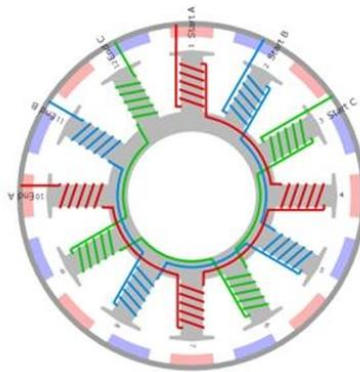
2.2 Motor BLDC 800 Watt

Motor *Brushless Direct Current* (BLDC) merupakan jenis motor listrik arus DC yang bekerja tanpa menggunakan *brush* atau sikat dan komutator mekanik. Motor BLDC bekerja dengan menggunakan prinsip gaya tarik antara dua buah magnet dengan kutub yang berlawanan dan gaya tolak antar kutub yang sama [11]



Gambar 2. 1 Motor BLDC 800 Watt[12]

Prinsip kerja dari Motor BLDC didasarkan pada interaksi medan magnet antar rotor dan stator. Pada saat arus listrik mengalir ke lilitan stator sesuai urutan tertentu, maka akan timbul medan magnet yang akan menarik atau mendorong magnet yang menempel pada rotor sehingga nantinya akan terbentuk sebuah gerak rotasi. Urutan pergantian aliran listrik pada lilitan stator disebut dengan komutasi. Motor BLDC tidak memiliki *brush* yang umumnya digunakan untuk menghantarkan arus listrik dari stator ke rotor, sehingga untuk komutasinya dapat dilakukan dengan menggunakan *hall* sensor [11]. Dapat dilihat pada gambar 2.2 kalau rotor, stator, dan *hall* sensor telah digabung menjadi satu kesatuan di dalam motor BLDC di atas dan di *seal* dengan rapat agar komponen di dalamnya terlindungi dari pengotor dari luar.



Gambar 2. 2 Konstruksi motor BLDC [13]

Konstruksi dari motor BLDC dapat dilihat di gambar 2.3 yang memperlihatkan stator yang berisi kumparan dan rotor yang dipenuhi oleh magnet permanen. Tiap stator yang terhubung dengan fasa yang sama diletakkan saling berjauhan 90° , lalu magnet yang terdapat pada rotor dipasang secara berlawanan kutub berdampingan, hal tersebut dilakukan supaya timbul beda potensi dari magnet. Beda potensi magnet nantinya akan membuat rotor berputar mengikuti lilitan stator yang di aliri arus listrik. Putaran tersebut yang akan menghasilkan torsi supaya motor BLDC dapat berputar secara terus menerus.

Tabel 2. 3 Spesifikasi Motor BLDC Bosch [12]

Spesifikasi	BLDC Bosch	Satuan
Daya maksimal	800	Watt
Tegangan Sumber	48 – 64	V
Arus Maksimal	20	A
Kecepatan Putar Maksimal	1000	RPM
Ukuran Roda	10	Inchi
Kecepatan Nominal	40	Km/h
Beban Maksimal	180	Kg
Torsi maksimum	80	Nm
Tipe Sensor Rotor	<i>Hall</i>	-
Efisiensi Daya	82	%
Suhu Operasi	-20 - 45	°C

Dapat dilihat dari tabel 2.2 kalau motor BLDC yang dipakai memiliki torsi dan kecepatan yang cukup tinggi untuk mendayai sebuah sepeda listrik. Terdapat 3 buah kabel fasa untuk menyalurkan arus AC hasil *Switching* agar dapat menghasilkan medan magnet pada stator. Kabel kuning (*U Phase*), Kabel Biru (*V Phase*), dan Kabel hijau (*W Phase*). Selain kabel fasa, terdapat juga 5 buah kabel *hall* sensor yang berfungsi untuk mendeteksi posisi rotor untuk tujuan komutasi, yakni 2 buah kabel untuk *power* dan 3 kabel data untuk *hall* sensor [12]. Pada penelitian ini, Motor BLDC digunakan sebagai penggerak dari sepeda listrik yang nantinya akan di kontrol lewat komutasi *Six step*.

2.2.1 Persamaan pada motor BLDC

Pada motor listrik, energi kinetik merepresentasikan energi yang dimiliki saat berada dalam kondisi bergerak. Energi kinetik dihasilkan dari konversi energi listrik dari baterai yang dipakai untuk menggerakkan massa total sepeda

motor listrik, meliputi massa pengendara dan bobot sepeda, hingga mencapai kecepatan tertentu. Energi kinetik di jabarkan pada persamaan 2.1 [14].

$$Ek = \frac{1}{2} \times m \times v^2 \quad (2.1)$$

Karena sepeda listrik bergerak saat beroperasi, maka akan timbul daya yakni konsumsi energi terhadap waktu. dengan membagi energi kinetik dengan waktu operasi, kita dapat memperoleh berapa daya yang dikonsumsi oleh sepeda listrik, untuk persamaan daya dapat dilihat pada rumus 2.2 [14].

$$P = \frac{E}{t} \quad (2.2)$$

Setelah memperoleh variabel waktu dan daya, kita dapat menghitung nilai variabel percepatannya. Percepatan merupakan perubahan kecepatan per satuan waktu. persamaan percepatan dapat dilihat pada persamaan 2.3 [9]

$$a = \frac{v}{t} \quad (2.3)$$

Selanjutnya kita dapat menghitung berapa nilai torsi yang dihasilkan oleh motor BLDC 800 watt dengan menggunakan data jari-jari roda motor dan akselerasi yang didapat sebelumnya. Torsi merupakan besaran gaya putar yang dihasilkan oleh poros motor BLDC untuk menggerakkan roda. Persamaan torsi dapat diperoleh dengan persamaan 2.4 [9].

$$\tau = m \times a \times r \quad (2.4)$$

RPM (*Rotations Per Minute*) merupakan variabel yang merepresentasikan frekuensi putaran poros motor dalam satu menit, atau lebih mudahnya supaya motor BLDC dapat berjalan dengan kecepatan tertentu, roda harus berputar

sekian kali dalam rentang waktu satu menit. Variabel yang mempengaruhi besarnya RPM adalah kecepatan dan diameter roda penggerak. Untuk menentukan RPM dapat menggunakan persamaan 2.5 [9].

$$\text{Kecepatan Putar} = \frac{v \times 60}{\pi \times D} \quad (2.5)$$

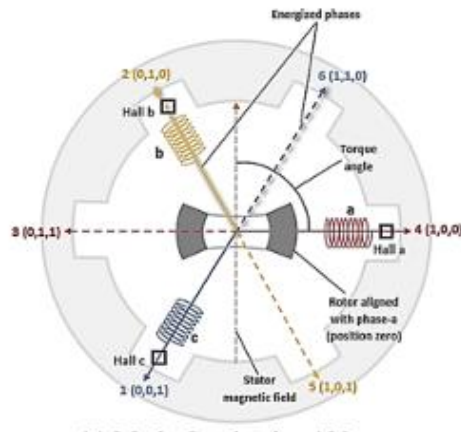
2.3 Sensor *Hall*

Sensor *hall* adalah sensor magnetik yang bekerja berdasarkan efek *hall*, yakni fenomena terbentuknya tegangan transversal pada saat medan magnet dialiri arus listrik [15]. Sensor *hall* merupakan komponen yang krusial dalam sistem pengontrolan motor BLDC yang memiliki fungsi untuk mendeteksi posisi rotor secara *real time* dengan membaca perubahan medan magnet yang dihasilkan oleh magnet yang terpasang pada rotor. Hal tersebut sangat penting karena komutasi *six step* membutuhkan posisi rotor agar dapat menentukan urutan *switching* MOSFET pada *driver* motor supaya arus mengalir ke stator yang tepat [15].

Pada penerapannya, motor BLDC umumnya dilengkapi dengan tiga buah sensor Hall yang diposisikan saling berjauhan sejauh 120 derajat elektrik pada bagian stator. Saat rotor berputar, interaksi kutub magnet akan membuat ketiga sensor tersebut menghasilkan variasi sinyal keluaran digital—berlogika *High* atau *Low*—yang membentuk enam pola kombinasi 3-bit berbeda dalam satu siklus penuh. Pola sinyal digital inilah yang secara terus-menerus diakuisisi oleh mikrokontroler melalui pin *interrupt* sebagai acuan mutlak untuk mengeksekusi perpindahan fasa secara sinkron, sehingga motor dapat menghasilkan torsi pergerakan yang presisi dan optimal.



Gambar 2. 3 Sensor *Hall* [16]



Gambar 2. 4 Posisi *hall* sensor pada motor BLDC [8]

Sensor *hall* sendiri memiliki 3 buah pin sesuai dengan gambar 2.4. Pin tersebut yakni VCC, GND, dan Output. Keluaran dari sensor ini berupa tegangan DC dengan rating sesuai dengan tipe sensor yang digunakan. Pada motor BLDC, sensor ini digunakan sebanyak tiga buah dan diletakkan terpisah disamping dari tiap fasa yang ada di stator sesuai dengan gambar 2.5 . Pemosisian tersebut dilakukan agar deteksi posisi rotor akurat. bentuk sinyal keluaran sensor *hall* pada BLDC berbentuk sinyal digital dengan enam kombinasi sinyal logika pada saat rotor berputar satu siklus [15]. Untuk logika yang dihasilkan dapat dilihat pada tabel 2.4 berikut.

Tabel 2. 4 Kombinasi sinyal logika *Hall* sensor [8]

Pola	<i>Hall A</i>	<i>Hall B</i>	<i>Hall C</i>
1	1	0	0
2	1	1	0
3	0	1	0
4	0	1	1
5	0	0	1
6	1	0	1

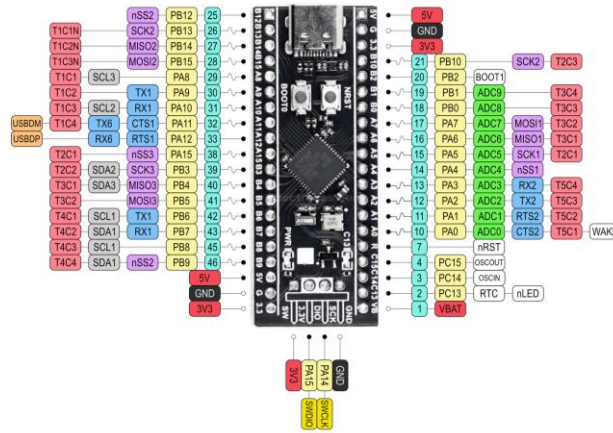
Tabel 2. 5 Spesifikasi *hall* sensor SS41 [16]

Spesifikasi	<i>Hall</i> Sensor SS41	Satuan
Tegangan Sumber	4,5 – 24	V
Jumlah Pin	3	-
Suplai Arus	50	mA
Tegangan saturasi	0.5	V
Output <i>rise time</i>	1.5	μ s
Output <i>fall time</i>	1.5	μ s
Suhu Operasi	-40 – 125	°C
Tegangan keluaran	4,5 – 24	V

Berdasarkan tabel spesifikasi *hall* sensor 2.5 dapat dilihat kalau sensor ini memiliki spesifikasi yang mumpuni dan respon yang cepat pada saat pembacaan medan magnet perubahan medan magnet sehingga cocok digunakan untuk menentukan posisi rotor motor BLDC yang membutuhkan *Switching* cepat untuk mengejar RPM tinggi.

2.4 STM32F401CCU6

STM32F401 adalah mikrokontroler yang diproduksi oleh STMicroelectronics dengan seri F4 yang dirancang untuk memenuhi kebutuhan performa tinggi. STM32F4 menggunakan *chipset* ARM Cortex-M4 yang memiliki frekuensi hingga 84MHz. Chip tersebut sudah dilengkapi fitur *floating-point unit* (FPU) serta *digital signal processing* (DSP) untuk perhitungan matematika kompleks [17]. Keberadaan fitur FPU berfungsi sebagai sistem kendali kecepatan *closed-loop*, karena memungkinkan kalkulasi algoritma *Proportional Integral Derivative* (PID) yang melibatkan bilangan desimal dapat dieksekusi secara langsung tanpa membebani kinerja memori. Selain itu, frekuensi komputasi yang tinggi dipadukan dengan kapabilitas DSP-nya menjadikan mikrokontroler ini sangat ideal untuk menangani pembacaan *interrupt* dari sensor Hall secara *real-time*.



Gambar 2. 5 STM32F401 Development Board Pinout [17]

Chipset STM32F401CCU6 memiliki tegangan operasi sebesar 1,7 – 3,6V dan tegangan input yang dibutuhkan board sebesar 5V. *Board* ini dibekali dengan *flash memory* sebesar 512 KB dan SRAM sebesar 96 KB yang sangat memadai untuk menyimpan data variabel maupun mengeksekusi instruksi program yang kompleks secara cepat. STM32F401 sendiri memiliki berbagai fitur periferal yang mumpuni, seperti *Timer* tingkat lanjut (*advanced-control timers*), *output* PWM beresolusi tinggi, serta dukungan ADC 12-bit dengan 10 *channel*. Selain itu modul ini juga compatible dengan berbagai *communication interfaces* seperti SPI, I2C, USART, dan SDIO. Kelengkapan fitur tersebut menjadikannya sangat ideal untuk diaplikasikan pada sistem kendali waktu nyata (*real-time*).

Tabel 2. 6 Spesifikasi STM32F401 Development Board [17]

Spesifikasi	STM32F401	Satuan
Chipset	ARM Cortex-M4 32-bit	-
Frekuensi maksimal	84	Mhz
Flash Memory	512	KB

Tabel 2. 7 lanjutan Spesifikasi STM32F401

Spesifikasi	STM32F401	Satuan
SRAM	96	KB
Tegangan Operasi	1,7 – 3,6	V
Tegangan Sumber	5	V
GPIO	37	-
Timers	8	Channel
ADC	12	Bit
Channel ADC	10	Channel
Komunikasi Interface	USART, SPI, I2C, SDIO	-
Suhu Operasi	−40 to +85	°C
Debugger Program	ST-LINK	-

STM32F401 Development Board menggunakan *Chip* yang ARM Cortex-M4 yang merupakan *upgrade* dari seri M3, dimana *chipset* ini dirancang untuk sistem *embended* yang membutuhkan performa tinggi, terutama pada sistem kendali secara *real time* dan pemrosesan sinyal digital. Sistem ini banyak digunakan pada aplikasi pengontrolan *driver* motor, *Inverter*, sistem alarm, dan berbagai sistem otomatis lain [17].

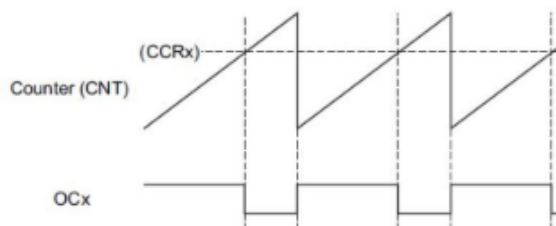
Pada sistem kendali motor, Pin *General Purpose Input Output* (GPIO) yang digunakan adalah digital input, Timer, USART, SWD, dan ADC. Timer pada STM32 digunakan untuk menghasilkan sinyal PWM yang nantinya akan digunakan sebagai sinyal kontrol untuk proses *switching* MOSFET. untuk mencegah adanya *short* pada saat proses *switching*, STM32F401 memanfaatkan fitur dead-time insertion yang terdapat pada *advance control timer*, fitur ini berguna untuk memberikan jeda waktu singkat antara memutus aliran pada satu sisi MOSFET dan menyalakan MOSFET di sisi lainnya. Sehingga dengan adanya fitur tersebut proses *switching* MOSFET jadi lebih aman [18].

Kecepatan motor BLDC dikendalikan menggunakan *duty cycle* PWM yang diberikan pada *gate* MOSFET. STM32F401 akan menghitung nilai *duty cycle* berdasarkan algoritma kontrol yang digunakan. Nantinya, nilai *duty cycle* akan terus diperbarui pada register timer yang dipakai. Semakin besar nilai *Duty Cycle*, maka semakin besar juga nilai PWM untuk mengendalikan kecepatan motor [19]. Jika PWM semakin besar maka tegangan rata-rata yang masuk ke lilitan motor akan lebih besar, hal tersebut membuat motor berputar dengan lebih kencang.

Interrupt adalah sebuah mekanisme yang memungkinkan mikrokontroler untuk berhenti sementara dalam mengeksekusi tugas utama yang sedang berjalan guna merespon tugas lainnya [20]. Pada sistem kontrol motor BLDC dengan STM32F401, *interrupt* memiliki fungsi untuk menjaga ketepatan waktu pada komutasi *six step*. Setiap kali terdapat perubahan medan magnet pada rotor, sensor *hall* akan membaca perubahan sinyal tersebut dan akan memicu *interrupt* yang menyebabkan STM32F401 segera memperbarui status *switching* MOSFET sesuai dengan tabel komutasi *six step*.

2.5 Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) merupakan teknik kontrol sinyal digital untuk menghasilkan efek sinyal analog dengan cara mengatur rasio waktu kondisi ON dan OFF dalam satu periode frekuensi. PWM dapat dianggap sebagai kebalikan dari sinyal ADC (*analog to digital converter*) yang mengonversikan sinyal analog menjadi digital. Teknik kontrol ini bekerja dengan cara mengubah lebar dari pulsa sinyal (*Duty Cycle*) berdasarkan waktu aktifnya [21]. Untuk bentuk sinyal dari PWM pada STM32 dapat dilihat pada gambar 2.7 .



Gambar 2. 6 Bentuk pulsa PWM pada STM32 [22]

Pada pengaturan PWM di STM32, terdapat dua komponen yang menentukan nilai dari PWM yang dihasilkan, yakni ARR (*Auto-Reload Register*) dan CCR (*Capture Compare Register*). ARR berfungsi sebagai penentu periode atau frekuensi sinyal PWM, register ini akan menetapkan nilai maksimal yang dapat dipakai oleh *counter* sebelum di atur ulang kembali ke nilai nol. Semakin besar nilai ARR, maka siklus gelombang akan semakin lama sehingga frekuensi PWM akan semakin mengecil [23]

$$ARR = \left(\frac{Freq_{time}}{Freq_{PWM} \cdot (Prescaler + 1)} \right) - 1 \quad (2.6)$$

CCR digunakan untuk menentukan nilai *duty cycle* dengan cara membandingkan nilainya terhadap nilai *counter*. Saat nilai counter berada dibawah nilai CCR, output PWM akan berada pada logika tertentu (bisa *High* atau *Low* tergantung konfigurasinya), dan status akan berubah ketika nilai *counter* telah melampaui nilai CCR tersebut [23].

$$CCR = \frac{Duty}{100} \cdot ARR \quad (2.7)$$

Dalam sistem pengontrolan kecepatan motor BLDC, PWM digunakan pada STM32F401 untuk mengontrol *switching* dari MOSFET lewat IC IR2110. *Switching* yang dilakukan akan berakibat pada kecepatan putaran motor BLDC. Semakin tinggi *duty cycle* yang dihasilkan, maka PWM juga akan semakin tinggi sehingga kecepatan motor juga akan meningkat.

2.6 Integrated Circuit IR2110

IC IR2110 merupakan *high-side* dan *low-side gate* driver yang sering digunakan pada sistem kontrol motor BLDC. IC ini berfungsi sebagai pengendali *switching* MOSFET dengan cara memberikan tegangan pada *gate* nya. Pada sistem pengendali motor BLDC, IC ini digunakan untuk mengontrol proses *switching* MOSFET pada tiap fasa motor sesuai dengan urutan komutasi *six step*. Sinyal PWM dari mikrokontroler

high-side secara penuh, meskipun kaki *source* dari MOSFET tersebut tidak terhubung langsung dengan *ground*.

Tabel 2. 8 Spesifikasi IR2110 [24]

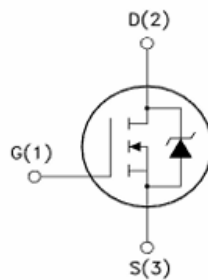
Spesifikasi	IR2110	Satuan
High Side Floating Supply Voltage	500	V
Tegangan Supply Logic	10 – 20	V
Arus Puncak Operasi	± 2	A
Frekuensi Switching	500	kHz
Turn-On Delay	≈ 120	Ns
Turn-Off Delay	≈ 94	Ns
Tegangan input logika	≥ 3	V
Suhu Operasi	-40 - 125	°C

2.7 IRFP250

MOSFET IRFP250 merupakan MOSFET bertipe N-Channel yang sering digunakan dalam proses *switching* pada beban dengan tegangan dan arus yang besar [26]. IRFP250 memiliki 3 buah pin yakni *drain* yang terhubung dengan beban atau catu daya, *Source* yang dihubungkan dengan *ground* rangkaian, lali *gate* yang berfungsi sebagai saklar yang dihubungkan dengan sinyal kontrol. Prinsip kerjanya yakni MOSFET akan menghubungkan arus dari *drain* ke *Source* apabila tegangan yang masuk pada *gate* sudah memenuhi nilai *threshold* yang ditentukan. IRFP250 sering digunakan sebagai sakelar daya pada rangkaian *inverter* baik *half bridge* maupun *half Bridge*. Pada aplikasi pengontrolan motor BLDC, MOSFET ini berfungsi sebagai komponen *switching* yang mengalirkan maupun memutus arus listrik dari baterai ke lilitan motor BLDC sesuai urutan komutasinya.



Gambar 2. 9 MOSFET IRFP250 [27]



Gambar 2. 10 Rangkaian skematik MOSFET [27]

Dari Gambar 2.10 tentang rangkaian skematik internal MOSFET dapat dilihat kalau cara kerja MOSFET sendiri mirip dengan sakelar dimana *gate* berfungsi sebagai pengontrolnya. Bila *gate* diberi tegangan *threshhold*, maka source dan drain akan menyambung sehingga arus listrik bisa lewat. Lalu terdapat pengaman berupa dioda zener untuk menghentikan arus balik. Hal tersebut sangat bermanfaat untuk menyaring *back EMF* yang ditimbulkan oleh motor BLDC saat beroperasi.

Tabel 2. 9 Spesifikasi IRFP250 [28]

Spesifikasi	IRFP250	Satuan
Max Drain Source Voltage	200	V
<i>Gate</i> Source Voltage	12	V
Arus Drain suhu 25°C	33	A
Arus Drain suhu 100°C	20	°C
Suhu Operasi	-40 - 150	°C
Frekuensi <i>Switching</i> maksimal	30	kHz

Dari tabel 2.9 spesifikasi IRFP250 dapat dilihat kalau mosfet ini memiliki tegangan *drain to source* yang tergolong tinggi hingga menyentuh angka 200 V, selain itu IC ini mempunyai frekuensi *switching* hingga 20 kHz yang membuatnya dapat melakukan *switching* cepat pada saat pengendalian kecepatan motor BLDC [28]. Namun perlu diingat juga kalau semakin tinggi frekuensi *switching* nya akan membuat suhu dari MOSFET akan naik dan berakibat buruk ke efisiensi dayanya.

2.8 TVS Dioda P6KE68A

Transient Voltage Suppressor (TVS) diode merupakan komponen proteksi yang berfungsi untuk melindungi rangkaian elektronik dari lonjakan tegangan sesaat (*transient voltage*) yang dapat disebabkan oleh *switching* beban induktif, pelepasan muatan elektrostatik, maupun *back Electromotive Force* (Back-EMF). TVS diode bekerja dengan cara tetap berada pada kondisi impedansi tinggi selama tegangan operasi normal, kemudian berubah menjadi impedansi rendah pada saat terjadi lonjakan tegangan yang melebihi nilai breakdown-nya sehingga energi berlebih dapat dialirkan ke ground dan tidak merusak komponen yang dilindungi [29]. Bentuk dari TVS dioda dapat dilihat pada gambar 2.12



Gambar 2. 11 TVS Dioda [29]

Tabel 2. 10 Spesifikasi TVS Dioda P6KE68A [29]

Spesifikasi	P6KE68A	Satuan
Daya Kerja Maksimal	600	Watt
Arus Kerja Maksimal	6,5	A
Breakdown Voltage	64,6 – 71,4	V
Max Clamping voltage	92	V

Berdasarkan Tabel 2.10 tentang spesifikasi dioda TVS yang dipakai, dapat dilihat kalau P6KE68A memiliki breakdown voltage di atas nilai tegangan supply baterai yakni sekitar 53V apabila baterai terisi maksimal, sehingga komponen tersebut tidak akan aktif apabila sistem berjalan normal, namun apabila terdapat *spike* atau lonjakan tegangan yang tinggi, TVS dioda akan aktif untuk mengalirkan spike tersebut ke ground sehingga tidak akan merusak rangkaian. TVS dioda P6KE68A memiliki respon yang sangat cepat hingga pikodetik sehingga bagus untuk sistem proteksi.

2.9 ELCO Low ESR

Elco atau yang biasa disebut kapasitor elektrolit merupakan salah satu komponen pasif yang sering digunakan pada sistem elektronika daya sebagai penyimpan energi sementara dan peredam fluktuasi tegangan [30]. Pada sistem kelistrikan daya tinggi seperti sistem inverter, Elco ditempatkan pada DC bus setelah baterai dan sebelum MOSFET inverter untuk menjaga kestabilan tegangan suplai. Kapasitor ini berfungsi sebagai penyerap arus ripple yang timbul akibat proses *switching* MOSFET serta menyediakan energi sesaat pada waktu perubahan beban mendadak. Dengan adanya kapasitor DC bus, tegangan pada sistem menjadi lebih stabil. Bentuk dari ELCO ini dapat dilihat pada gambar 2.13 .



Gambar 2. 12 ELCO Low ESR 100V 470uF [31]

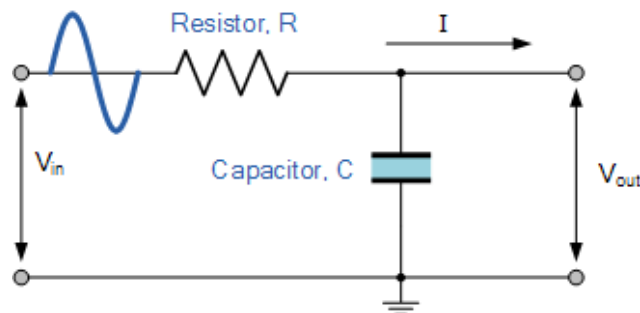
Tabel 2. 11 Spesifikasi ELCO Low ESR [31]

Spesifikasi	ELCO	Satuan
Kapasitansi	470	uF
Tegangan Kerja	6,3 - 100	V
Suhu Operasi	-40 - 105	°C
Toleransi kapasitansi	±20	%

Berdasarkan Tabel 2.11 dapat dilihat kalau ELCO yang digunakan pada sistem ini memiliki kapasitansi yang cukup besar di angka 470 uF serta tegangan kerja maksimal di angka 100V. Angka tersebut cocok dengan tegangan bus yang digunakan dari baterai yakni 48V sehingga ELCo dapat bekerja dengan baik untuk menjaga kestabilan tegangan suplai. Selain itu, komponen ini memiliki *Equivalent Series Resistance* (ESR) yang rendah sehingga dapat merespon perubahan arus dengan lebih cepat dan menghasilkan rugi daya yang lebih kecil dibanding tipe *non Low ESR*.

2.10 RC Low Pass Filter

RC Low Pass Filter merupakan rangkaian elektronika pasif yang tersusun dari resistor (R) dan kapasitor (C) yang memiliki fungsi untuk melewatkan sinyal dengan frekuensi rendah dan meredam sinyal dengan frekuensi tinggi. Pada sistem kendali motor BLDC, RC Low Pass Filter banyak digunakan sebagai rangkaian pengkondisi sinyal (*signal conditioning*) untuk mengurangi gangguan EMI (electromagnetic interference) dan noise yang muncul akibat proses *switching inverter* maupun PWM pada driver motor [32].



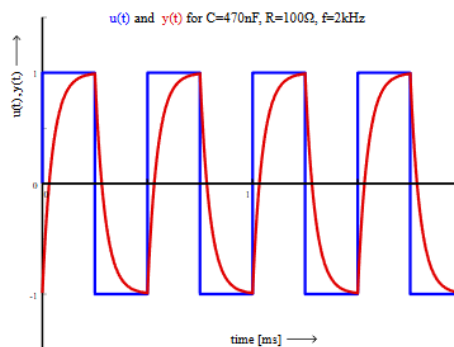
Gambar 2. 13 Rangkaian RC Filter [33]

Pada rangkaian Low Pass Filter, Resistor dan kapasitor memiliki fungsinya masing-masing. Resistor berguna untuk membatasi arus yang mengalir menuju kapasitor sehingga mempengaruhi seberapa cepat kapasitor mengisi dan mengosongkan muatan, sedangkan kapasitor berfungsi untuk menyimpan muatan listrik sementara dan mengalirkan sinyal berfrekuensi tinggi ke *ground* sehingga membuat tegangan

keluaran menjadi lebih halus. Low Pass filter bekerja dengan melewatkan sinyal yang frekuensinya dibawah frekuensi *cut-off* nya. Untuk menentukan frekuensi *cut-off* dapat menggunakan rumus 2.8 dibawah ini .

$$f_c = \frac{1}{2\pi RC} \quad (2.8)$$

Pada persamaan 2.8 dapat dilihat kalau frekuensi *cut-off* dipengaruhi oleh nilai resistor dan kapasitor yang digunakan. Semakin besar nilai resistor dan kapasitornya, maka semakin kecil nilai frekuensi sinyal yang dilewatkan begitu pula sebaliknya. Untuk contoh perbandingan sinyal antara sebelum dan sesudah terkena RC filter dapat dilihat pada gambar 2.13 berikut.



Gambar 2. 14 Perbandingan sinyal keluaran RC Low Pass Filter [33]

2.11 LifePo4 48V 12Ah

Baterai *Lithium Iron Phosphate* (LiFePO₄) merupakan jenis baterai *rechargeable* atau dapat diisi ulang yang bahan utama dalam pembuatannya berupa *Lithium Iron Phosphate* sebagai katoda dan karbon grafit sebagai anodanya [34]. Baterai ini dikenal sebagai baterai dengan teknologi *Lithium-ion* yang paling stabil dan aman dikarenakan memiliki struktur kimia yang sangat kuat dan tahan terhadap panas. Hal tersebut membuat risiko meledak atau terbakar akibat suhu tinggi menjadi lebih rendah apabila dibandingkan dengan varian Lithium-ion yang lain.

Tegangan nominal baterai LiFePO₄ berbeda-beda tergantung jumlah *cell* yang dimiliki. Untuk menentukan tegangan nominalnya, dapat digunakan rumus 2.7.

$$V_{nom} = N_{cell} \times 3,2 \quad (2.9)$$

Kapasitas yang dapat disimpan oleh baterai LiFePO4 di pengaruhi oleh besaran arus yang dapat disimpan serta waktu operasi dari baterai tersebut, untuk menentukan nilai kapasitas baterai LiFePO4 dapat menggunakan persamaan 2.8.

$$Batt\ Capacity = I_{load} \times Hours \quad (2.10)$$

Untuk bentuk fisik dari baterai LiFePO4 dapat dilihat pada gambar 2.16 berikut. Namun bentuk baterai yang digunakan pada sistem bisa jadi berbeda dari gambarnya.



Gambar 2. 15 Baterai LiFePO4 48V 12Ah [35]

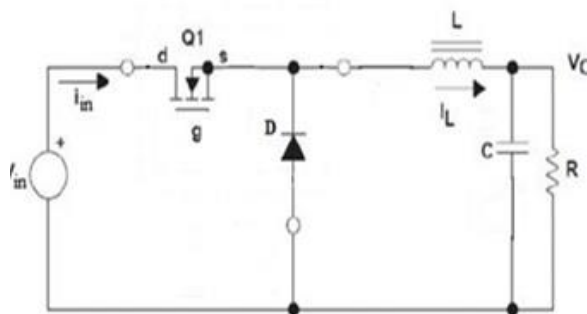
Tabel 2. 12 Spesifikasi Baterai LiFePO4 48V 12Ah [35]

Spesifikasi	LiFePO4	Satuan
Tegangan	48	V
Kapasitas	14	Ah
Daya Simpan	576	Wh
Tegangan cut-off	36,8	V
Siklus pengisian	3000	-
Arus Output Maksimal	14	A

Dapat dilihat dari tabel 2.12 tentang spesifikasi baterai LiFePO₄, kalau baterai jenis ini memiliki kapasitas dan tegangan yang cukup tinggi untuk mendayai motor BLDC, selain itu LiFePO₄ juga memiliki Siklus pengecasan yang cukup tinggi sehingga kapasitas baterai tidak cepat menurun pada saat dilakukan pengecasan berulang kali. Hal tersebut cukup penting karena sepeda listrik akan sering di *charger* agar bisa digunakan secara terus-menerus.

2.12 Buck Converter LM2596

Buck converter berbasis LM2596 merupakan rangkaian konverter DC-to-DC yang berfungsi untuk menurunkan tegangan. Berdasarkan gambar 2.9 Konverter ini bekerja dengan prinsip *switching* cepat dari IC LM2596, dioda, induktor, dan kapasitor untuk menurunkan tegangan dari sumbernya. Modul ini menggunakan teknik PWM internal untuk mengendalikan durasi dari aliran arus listrik yang masuk ke induktor, kemudian menyaring energi listrik tersebut menjadi tegangan output yang stabil. Saat transistor aktif, energi tersimpan ke dalam induktor. Sedangkan saat transistor sedang non-aktif, maka induktor melepaskan energi melalui dioda ke beban. Rangkaian dioda zener digunakan untuk mempertahankan nilai tegangan output, sedangkan rangkaian induktor-kapasitor bertugas untuk memfilter riak tegangan sehingga tegangan keluaran menjadi lebih stabil [36]. Untuk rangkaian dari *buck converter* LM2596 dapat dilihat pada gambar 2.16 dan spesifikasinya dimuat pada tabel 2.14.



Gambar 2. 16 Rangkaian Buck Converter [36]

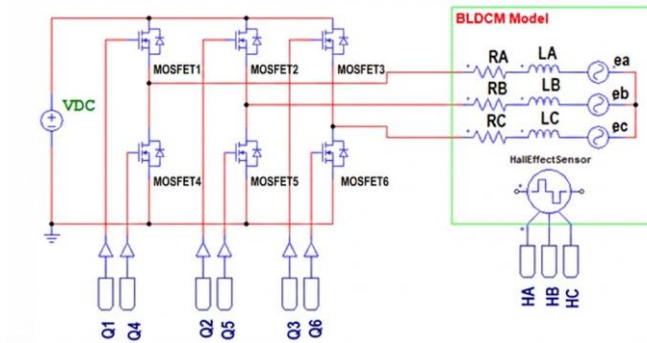
Tabel 2. 13 Spesifikasi Buck Converter LM2596 [37]

Spesifikasi	Buck Converter LM2596	Satuan
<i>Input Voltage</i> Maksimal	40	V
<i>Output Voltage</i> Maksimal	37	V
Arus Maksimal	3	A
Frekuensi <i>switching</i>	150	kHz
Suhu Operasi	-40 - 125	°C

Pada Sistem Pengontrolan motor BLDC. *Buck converter* LM2596 digunakan untuk mendayai mikrokontroler STM32F401 yang membutuhkan tegangan sumber minimal 3,3V DC agar dapat beroperasi. Modul ini mengambil daya dari baterai lalu menurunkannya menjadi 3,3V hingga 5V.

2.13 Inverter 3 Fasa *Half Bridge*

Inverter merupakan suatu rangkaian elektronika yang berfungsi untuk mengonversi sumber tegangan DC menjadi tegangan AC. *Inverter* disuplai dari sumber tegangan DC dimana output tegangan dan arusnya AC nya mempunyai frekuensi dan amplitudo yang dapat di variasikan. Rangkaian *inverter* 3 fasa terdiri dari 6 buah komponen semikonduktor sebagai sakelar, seperti MOSFET atau IGBT yang disusun sejumlah 3 pasang lengan (*leg*), masing-masing pasang mewakili satu fasa keluaran. Tiap lengan mempunyai *High-Side* dan satu *Low-Side* yang bekerja secara komplementer [38]. Keluaran dari tiap fasanya akan menjadi daya untuk mengontrol kecepatan dari motor BLDC. Dalam pengoperasiannya pada sistem penggerak motor BLDC, pensaklaran keenam MOSFET tersebut tidak dilakukan secara serentak, melainkan diatur secara sekuensial mengikuti pola komutasi *six-step*. Mikrokontroler akan menyuntikkan sinyal kendali berupa *Pulse Width Modulation* (PWM) ke *gate* masing-masing MOSFET untuk mengatur besaran tegangan fasa dan frekuensi keluaran sesuai dengan putaran rotor. Bentuk rangkaian inverter 3 fasa dapat dilihat pada gambar 2.17.

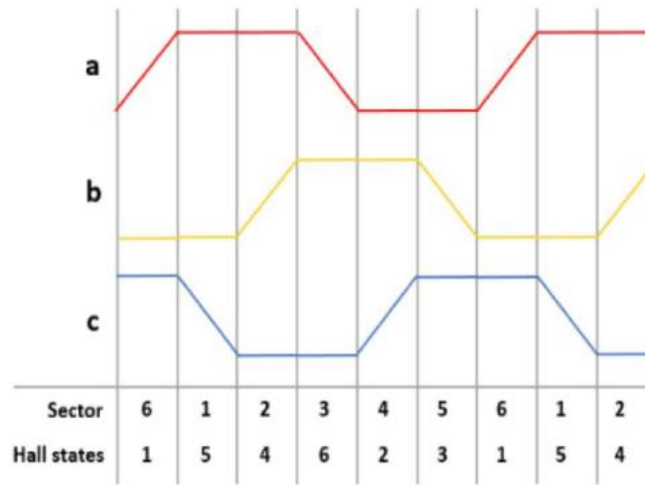


Gambar 2. 17 Skematik rangkain *inverter* 3 fasa pada BLDC motor [22]

Pada sistem kontrol motor BLDC, *inverter* 3 fasa berjenis *half bridge* berperan sebagai penghasil tegangan dan arus 3 fasa untuk menciptakan medan magnet rotasi pada Stator agar rotor dari BLDC dapat bergerak dan menghasilkan torsi. Nominal tegangan fasa dan frekuensi keluaran dari *inverter* dapat diatur nilainya menggunakan PWM dari mikrokontroler. Namun sebelum itu, PWM dari mikrokontroler akan di besarkan dengan IC 2110 agar tegangannya dapat memenuhi tegangan *threshold* yang dibutuhkan oleh *gate* agar MOSFET bisa aktif dan menghasilkan listrik 3 fasa.

2.14 Komutasi Six-Steps

Komutasi *six step* adalah metode kontrol untuk menggerakkan motor BLDC dengan cara mengaktifkan dua dari tiga fasa motor pada setiap langkahnya. Secara umum, metode ini membagi siklus putaran 360° menjadi 6 langkah yang tiap langkahnya berjarak 60° . Pada tiap langkah, satu fasa dihubungkan ke kutub positif (*High-Side*), satu fasa ke kutub negatif (*Low-side*), sedangkan satu fasa lainnya dibiarkan mengambang atau tidak terkoneksi. Proses *switching* yang berurutan tersebut menciptakan medan magnet pada stator yang akan menarik magnet permanen pada rotor sehingga dapat berputar. Metode ini juga sering disebut metode *trapezoidal*, dikarenakan pada satu waktu hanya ada dua fasa yang aktif sehingga menghasilkan bentuk gelombang trapesium seperti gambar 2.18 [39].



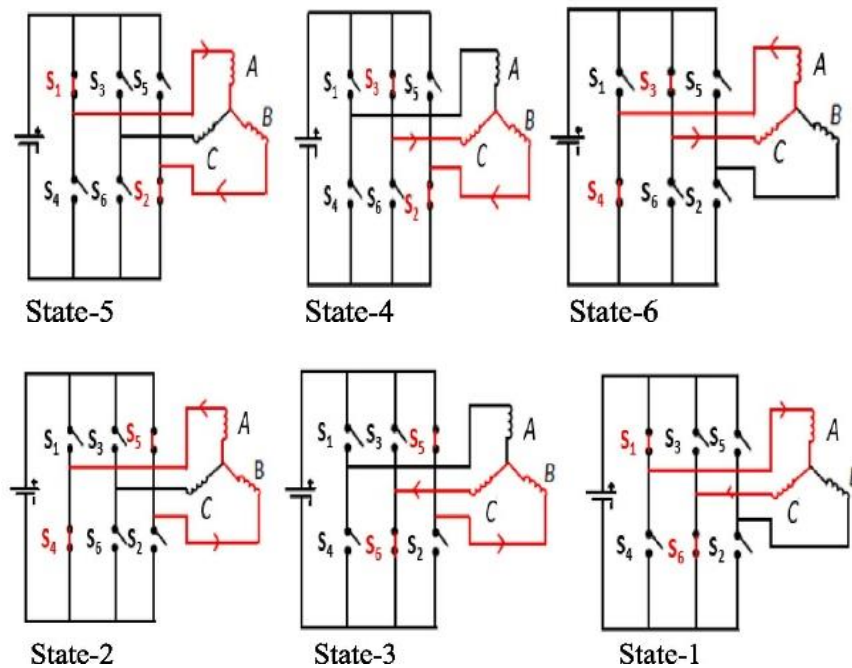
Gambar 2. 18 Bentuk gelombang arus dan tegangan komutasi six steps [40]

Pada Sistem pengontrolan BLDC, komutasi *six step* akan bekerja berdasarkan data pembacaan posisi rotor yang didapat dari *hall* sensor. Tiga buah *hall* sensor dipasang dengan jarak 120° dalam motor. Data posisi rotor akan diteruskan ke mikrokontroler untuk diolah lalu mikrokontroler akan mengirimkan PWM ke fasa tertentu supaya daya tarik medan magnet berpindah dari fasa satu ke fasa lainnya yang membuat rotor akan terus berputar. Urutan *Switching* pada komutasi six steps dapat dilihat pada tabel 2.15. MOSFET pada rangkaian *inverter* akan dikontrol berdasarkan urutan *switching* tersebut dengan mengikuti syarat mutlak, yakni hanya dua fasa yang aktif secara bersamaan dengan kutub yang berbeda, sedangkan satu fasa lainnya dibiarkan dalam kondisi mengambang (*floating*). Pada keadaan tersebut, arus listrik akan mengalir dari sumber tegangan melewati MOSFET *High-side* pada fasa pertama, masuk menembus lilitan stator di dalam motor, dan mengalir kembali ke *ground* melalui MOSFET *Low-side* pada fasa kedua. Aliran arus yang presisi inilah yang membangkitkan fluks magnetik pada stator untuk berinteraksi dengan magnet permanen rotor, sehingga menghasilkan torsi putar yang optimal. Seiring dengan berputarnya poros motor, perubahan kombinasi logika dari ketiga sensor Hall akan secara terus-menerus memicu mikrokontroler untuk mengeksekusi transisi ke langkah komutasi berikutnya setiap

pergerakan 60 derajat elektrik. Siklus pergantian fasa stator yang berlangsung sinkron dan berkesinambungan ini memastikan motor BLDC dapat berputar secara responsif dan terhindar dari kondisi *stalling* (berhenti mendadak) saat diberi beban. Untuk MOSFET yang aktif pada state tertentu dapat dilihat di gambar 2.19 .

Tabel 2. 14 State pada komutasi Six Step [39]

State	Hall			Position	Sector	The active line		Switch (On)
	H_a	H_b	H_c			Positive	Negative	
5	1	0	0	(-30, 30)	1	A	C	S_1 and S_2
4	1	1	0	(30, 90)	2	B	C	S_3 and S_2
6	0	1	0	(90, 150)	3	B	A	S_3 and S_4
2	0	1	1	(150, 210)	4	C	A	S_5 and S_4
3	0	0	1	(210, 270)	5	C	B	S_5 and S_6
1	1	0	1	(270, 330)	6	A	B	S_1 and S_6



Gambar 2. 19 Kondisi pensaklaran MOSFET berdasarkan state [39]

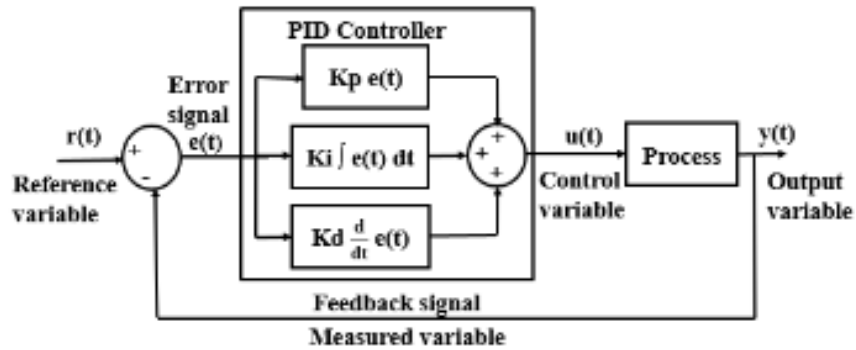
Dapat dilihat pada tabel 2.15 berisi urutan komutasi *six-step* yang menjadi dasar dalam pengoperasian sistem kendali motor BLDC. Pada tabel tersebut, penentuan *state* atau kondisi komutasi sangat bergantung pada kombinasi logika dari ketiga sensor Hall (*Hall A*, *Hall B*, dan *Hall C*) yang berfungsi mendeteksi posisi rotor pada setiap interval 60 derajat elektrik (dibagi ke dalam enam sektor). Setiap kombinasi sinyal sensor ini akan menentukan fasa mana yang akan dialiri arus positif dan fasa mana yang menjadi jalur yang terhubung dengan *Ground*. Sebagai contoh, pada *State 5* dengan pembacaan sensor Hall 100 yang berada di sektor 1 (rentang -30 hingga 30 derajat), fasa A bertindak sebagai kutub positif dan fasa C sebagai kutub negatif. Oleh karena itu, mikrokontroler akan memerintahkan pensaklaran sehingga sakelar (MOSFET) S1 dan S2 menjadi aktif (*ON*).

Prinsip kerja dari tabel tersebut dijelaskan pada Gambar rangkaian *inverter* 2.21, yang menunjukkan arah aliran arus pada masing-masing *state*. Pada topologi *inverter* tiga fasa ini, sistem kontrol hanya akan mengaktifkan dua dari enam MOSFET pada satu waktu tertentu, sementara satu fasa sisanya berada dalam kondisi tidak terhubung (*floating*). Mengacu pada gambar, saat *State-5* berlangsung, arus searah mengalir dari sumber tegangan melewati MOSFET S1 (sisi *high-side* fasa A), masuk melalui kumparan stator, dan kembali ke *ground* melalui MOSFET S2 (sisi *low-side* fasa C). Transisi ini kemudian berganti secara berurutan mengikuti putaran rotor, mulai dari *State-5*, berlanjut ke *State-4* (di mana S3 dan S2 aktif), *State-6* (S3 dan S4 aktif), dan seterusnya hingga *State-1* (S1 dan S6 aktif). Perpindahan *state* yang presisi pada rangkaian *inverter* inilah yang menghasilkan medan magnet putar secara sinkron untuk menjaga kelancaran torsi pada motor.

2.15 Proportional Integral Derivative (PID)

Proportional Integral Derivative (PID) merupakan salah satu metode pengontrolan yang paling sering digunakan karena strktur yang sederhana, mudah untuk diimplementasikan, serta mampu menghasilkan respon time yang baik. Terhitung 97% industri yang bergerak dalam bidang proses telah menggunakan PID

dalam komponen pengontrolannya [41]. Prinsip kerja dari kontrol PID adalah dengan menghitung nilai *error* dari sebuah sistem, yakni selisih antara nilai referensi (*set point*) dan nilai keluaran sistem (output) lalu membuat nilai tersebut menjadi seminimal mungkin supaya sistem dapat mempertahankan kondisi yang diinginkan [42].



Gambar 2. 20 Diagram Blok Kontrol PID [42]

Kontrol PID terdiri dari 3 buah komponen kontrol yakni kontrol *Proportional*, *integral*, dan *Derivative*. Ketiga komponen ini memungkinkan sistem kontrol untuk merespons perubahan secara cepat, presisi, mengurangi *Steady State error*, dan meningkatkan kestabilan sistem. Rumus untuk persamaan PID dapat ditulis sebagai berikut :

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2.11)$$

Berdasarkan persamaan 2.11, tiap parameter kontrol memiliki fungsinya masing masing, parameter K_p merupakan konstanta untuk menentukan besarnya respon sistem sebanding dengan *error* yang dihasilkan, parameter K_i merupakan konstanta integral yang berfungsi untuk mengakumulasikan error terhadap waktu dan fungsi utamanya untuk menghilangkan *steady state error*. Sedangkan parameter K_d merupakan konstanta untuk meredam laju keluaran kontrol berdasarkan kecepatan perubahan

error, fungsi utama dari parameter ini untuk meredam *overshoot* pada saat waktu transien.

Penentuan nilai parameter K_p , K_i , dan K_d pada pengendali PID dapat dilakukan menggunakan metode Ziegler–Nichols berdasarkan respon *open-loop* sistem [43]. Metode ini dilakukan dengan memberikan masukan berupa sinyal langkah (step input) pada sistem *open-loop* lalu mengamati kurva respon untuk mendapatkan garis singgung pada respon keluarannya. Berdasarkan kurva dan garis singgung nantinya akan didapati karakteristik respon sistem seperti waktu tunda *dead time* (L), dan konstanta waktu *time constant* (T), yang selanjutnya digunakan untuk menentukan parameter awal pengendali PID sesuai aturan penalaan Ziegler–Nichols. Untuk melakukan *tuning* parameter PID yang akan di gunakan, dapat menggunakan metode Ziegler–Nichols 1 dengan aturan *tuning* seperti pada tabel 2.15 Dibawah ini.

Tabel 2. 15 Perhitungan parameter PID dengan Ziegler Nicole 1

Metode Kontrol	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PD	$1.2 \frac{T}{L}$	$2 L$	$0.5 L$

Berdasarkan tabel 2.15, untuk menentukan parameter kontrol PID dapat digunakan persamaan seperti pada rumus 2. Berikut

$$K_p = 1.2 \frac{R}{L} \quad (2.12)$$

$$T_i = 2 L \quad (2.13)$$

$$T_d = 0.5 L \quad (2.14)$$

Berdasarkan Persamaan 2.11 hingga 2.13 diperoleh L sebagai *dead time*, T sebagai *time constant*, sedangkan T_i digunakan untuk mendapatkan kecepatan integral guna mengeliminasi *Steady-state error*. Nilai K_i dapat diperoleh dengan menggunakan rumus sebagai berikut.

$$K_i = \frac{K_p}{T_i} \quad (2.15)$$

Dengan K_p merupakan konstanta proporsional, K_i merupakan konstanta integral dan T_i merupakan waktu integral. Setelah itu parameter K_d dapat di tentukan dengan menggunakan nilai T_d dengan menggunakan rumu sebagai berikut.

$$K_d = K_p \cdot T_d \quad (2.16)$$

Dengan K_p merupakan konstanta proporsional, K_d merupakan konstanta derivative dan T_d merupakan waktu derivative. Setelah mendapatkan semua parameter K_p , K_i , dan K_d , Sistem PID dapat di operasikan dengan baik dan optimal.

2.16 Exponential Moving Average (EMA)

Exponential Moving Average (EMA) merupakan salah satu metode pemfilteran sinyal yang memberikan pembobotan secara eksponensial, di mana titik data yang paling baru memiliki tingkat signifikansi yang dapat diatur lebih tinggi maupun lebih rendah dibandingkan data masa lampau. Dalam konteks pemrosesan sinyal digital dan sistem kendali, algoritma EMA sering diimplementasikan sebagai *low-pass filter* pada bagian *software*. Penggunaan metode ini bertujuan untuk meredam fluktuasi acak atau *noise* berfrekuensi tinggi pada data masukan seperti fluktuasi pembacaan sensor tanpa menimbulkan waktu tunda (*phase delay*) komputasi yang terlalu besar [44]. Hal ini

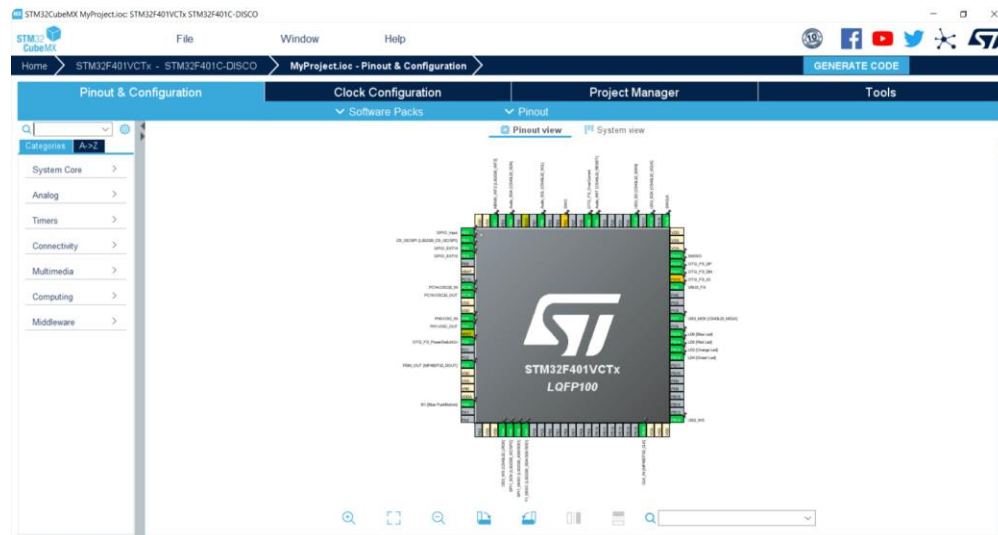
menjadikan algoritma EMA sangat efisien untuk diterapkan secara *real-time* di dalam mikrokontroler untuk menjaga stabilitas keluaran dari sensor yang dinamis seperti pembacaan kecepatan oleh *hall* sensor. Untuk persamaan dari EMA dapat ditulis sebagai berikut.

$$St_1 = Xt.\alpha + (1 - \alpha)St_0 \quad (2.17)$$

Dimana St_1 merupakan perhiperhitungan saat ini, Xt merupakan data terbaru yang masuk atau jika dalam sistem saya yakni kecepatan aktual RPM saat ini, lalu α merupakan *smoothing constant* atau konstanta pengali untuk pembobotan RPM aktual terbaru yang nolainya dapat di atur dari 0 hingga 1, dan St_0 merupakan data RPM sebelumnya. Pada sistem ini nilai α yang digunakan adalah 0,2 agar nilai RPM sebelumnya memiliki pembobotan yang lebih besar di banding RPM terbaru, tujuannya supaya nilai RPM tidak langsung anak atau turun drastis pada saat terjadi *noise* agar kalkulasi PID dapat berjalan dengan baik.

2.17 STM32CubeMX

STM32CubeMX merupakan *software* konfigurasi yang dirancang oleh STMicroelectronics untuk memudahkan inisialisasi mikrokontroler berbasis ARM cortex-M *series*. Software ini membuat pengguna mikrokontroler tipe STM32 dapat mengatur konfigurasi *hardware* melalui antarmuka visual, mulai dari pemetaan pin yang ingin digunakan, pengaturan struktur *clock*, maupun inisialisasi periferal internal tanpa harus menulis kode register secara manual. Dengan membuat kode inisialisasi berbasis pustaka HAL (*Hardware Abstraction Layer*) secara otomatis, membuat konfigurasi sistem menjadi lebih efisien dan minim kesalahan [45].



Gambar 2. 21 Tampilan STM32Cube MX [45]

Pada penelitian ini, STM32CubeMX digunakan untuk mengkonfigurasi Timer meliputi ARR dan CCR untuk menghasilkan tiga buah sinyal PWM yang saling berhubungan. Selain itu, konfigurasi digital input dan *interrupt* dipakai untuk membaca masukan dari 3 buah *hall* sensor agar dapat mendeteksi posisi rotor secara presisi.

2.18 Arm Keil MDK

Arm Keil uVision adalah sebuah *software* yang dirancang khusus untuk pengembangan aplikasi terkhusus mikrokontroler yang emakai arsitektur ARM. Secara general, dalam *software* ini sudah mencakup ekosistem seperti editor kode, manajemen file kodingan, *compiler* C/C++, serta sudah dilengkapi fitur untuk debug kodingan ke mikrokontroler secara langsung [46]. Dalam *software* ini, kita bisa menuliskan kodingan baik itu berupa text code maupun operasi aritmatika yang nantinya akan di eksekusi oleh mikrokontroler berarsitektur ARM yang kita pakai.

Dalam penelitian ini, *software* keil uVision digunakan untuk menuliskan kodingan tentang algoritma kontrol *Close-loop* untuk kontrol PID dengan elakukan *tuning* parameter kontrol seperti Ki, Kp, dan Kd serta menghitung selisih antara pembacaan RPM aktual yang di dapat dari pembacaan *interrupt hall* sensor dengan nilai RPM

setpoint yang diberikan pada kodingan. Nantinya, hasil perhitungan PID akan digunakan untuk menyesuaikan nilai register CCR dan ARR secara otomatis pada Timer STM32, sehingga nilai *duty cycle* PWM dapat digunakan untuk mengatur kecepatan motor BLDC lewat IC IR2110 dan MOSFET IRFP250.

2.19 Karakteristik Respon Waktu Sistem kontrol

Kinerja suatu sistem kontrol *Close-loop*, seperti pada pengaturan kecepatan motor listrik, dievaluasi berdasarkan pengamatan terhadap respons peralihan (*transient response*) dan *steady-state response*. Karakteristik ini umumnya diamati dengan memberikan sinyal uji berupa masukan sinyal uji maupun menguji sistem secara langsung. Parameter-parameter utama yang digunakan untuk mengukur spesifikasi unjuk kerja sistem dalam domain waktu meliputi *rise time*, maksimum *overshoot*, *settling time*, dan *steady-state error* [47].

a. *Rise Time* (t_r)

Rise time merupakan waktu yang dibutuhkan oleh respons sistem untuk bergerak dari persentase batas bawah menuju persentase batas atas dari nilai akhir keadaan tunaknya [47]. Pada sistem orde satu yang tidak berosilasi (seperti karakteristik dasar motor), *rise time* umumnya diukur dari 10% hingga 90% dari nilai akhirnya. Rise time dapat dihitung menggunakan persamaan 2.18 berikut.

$$t_r = t_{90\%} - t_{10\%} \quad (2.18)$$

Di mana $t_{90\%}$ adalah waktu saat respons mencapai 90% dari nilai *steady-state*, dan $t_{10\%}$ adalah waktu saat respons mencapai 10% dari nilai *steady-state*. Parameter ini mengindikasikan seberapa cepat sistem merespons perintah dari pengontrol.

b. *Overshoot* (%OS)

Overshoot merupakan nilai puncak tertinggi dari kurva respons yang melampaui nilai keadaan tunaknya. Parameter ini sering dinyatakan dalam bentuk persentase terhadap nilai akhir [47]. *Overshoot* merupakan indikator tingkat kestabilan relatif sistem, nilai *overshoot* yang terlalu besar dapat menyebabkan tekanan mekanis atau menjadi indikasi adanya daya yang berlebih pada perangkat elektronik dan mekanis. Presentase *overshoot* dapat ditulis menggunakan rumus 2.19 berikut

$$\%OS = \left(\frac{C_{max} - C_{ss}}{C_{ss}} \right) \times 100\% \quad (2.19)$$

Dimana C_{max} merupakan nilai output tertinggi dari sistem sedangkan C_{ss} merupakan nilai output tertinggi saat sistem telah mencapai kondisi tunak.

c. *Settling Time* (t_s)

Settling Time adalah waktu yang dibutuhkan oleh kurva respons sistem untuk mencapai dan tetap berada di dalam suatu rentang nilai toleransi yang telah ditentukan (umumnya 5%) di sekitar nilai keadaan tunaknya [47]. Parameter ini menunjukkan durasi keseluruhan yang dibutuhkan sistem kontrol hingga fluktuasi kecepatan motor benar-benar mereda. Persamaan *settling time* dapat dituliskan dengan persamaan 2.20 berikut

$$t_s = t \text{ saat } |C(t) - C_{ss}| \leq 5\% \times C_{ss} \quad (2.20)$$

Di mana $C(t)$ nilai keluaran sistem pada waktu t . Sehingga t_s merupakan rentang waktu pada saat sistem mulai stabil dengan toleransi nilai error 5% dari *steady-state*.

d. *Steady-State Error* (e_{ss})

Steady-State Error merupakan selisih antara nilai referensi yang diinginkan (*setpoint*) dengan nilai aktual keluaran sistem setelah respons transien

menghilang sepenuhnya dan sistem telah stabil. Parameter ini merepresentasikan tingkat akurasi statis dari sistem kendali [47]. Dalam domain waktu, *Steady-State Error* dapat ditulis menggunakan rumus 2.21 berikut.

Berdasarkan rumus 2.21, *Steady-State Error* dapat dihitung dengan mengurangi nilai *set point* dengan nilai saat sistem dalam kondisi Tunak

$$e_{ss} = |SP - C_{ss}| \quad (2.21)$$