

**SKRIPSI**  
**SISTEM KENDALI MOTOR BLDC DENGAN HALL SENSORS**  
**BERBASIS CPLD**



*Disusun dalam rangka memenuhi salah satu persyaratan untuk menyelesaikan  
program Strata Satu Departemen Teknik Elektro Fakultas Teknik*

*Universitas Hasanuddin*

*Makassar*

Disusun Oleh:

**MUHAMMAD FAJRI SACHRUDDIN**

**D411 16 304**

**DEPARTEMEN ELEKTRO FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2021**

**LEMBAR PENGESAHAN TUGAS AKHIR**

**SISTEM KENDALI MOTOR BLDC DENGAN HALL SENSORS  
BERBASIS CPLD**

Disusun oleh:

Muhammad Fajri Sachruddin          D411 16 304

Disusun dalam rangka memenuhi salah satu persyaratan untuk menyelesaikan  
Program Strata Satu Program Studi Teknik Elektro  
Departemen Teknik Elektro Fakultas Teknik Universitas Hasanuddin

Makassar, 5 Januari 2022

Disahkan oleh:

Pembimbing I

Pembimbing II

**Prof. Dr-Ing Faizal Arya Samman, S.T., M.T.**

NIP. 19750605 200212 1 004

**Dr. Ir. H. Rhiza S. Sadjad, MSEE.**

NIP. 19570906 198203 1 004

Mengetahui,

Ketua Departemen Elektro  
Fakultas Teknik Universitas Hasanuddin

**Dr. Eng. Ir. Dewiani, M.T.**

NIP. 19691026 199412 2 001

## ABSTRAK

Pada penelitian ini menampilkan perancangan dan implementasi dari pengendali motor *brushless direct current* (BLDC) menggunakan *complex programmable logic device* (CPLD). Implementasi dari sistem ini menggunakan teknik PWM dengan mengubah *duty cycles* yang diberikan pada inverter tiga fasa. Posisi rotot ditentukan menggunakan sensor hall yang digunakan untuk sinkronisasi sinyal control PWM. Sistem control ditulis menggunakan Bahasa *Verilog Hardware Description Language* dan diverifikasi menggunakan simulasi pada ModelSim-Altera. Perancangan eksperimen dilakukan untuk menguji performa motor BLDC menggunakan control PWM. Algoritma control diimplementasikan menggunakan perangkat MAX II EPM240T100C5 pada motor BLDC 350W 36V. Jumlah *logic elements* yang digunakan sekitar 133 dari 240 *LEs* yang tersedia. PWM yang dibangkitkan memiliki frekuensi 20KHz dengan lebar *duty cycles* yang dapat diatur dan berkomutasi secara berurutan sesuai komutasi *six-step* dengan hasil uji kecepatan hingga 600 RPM dengan kondisi tanpa beban.

*Kata kunci: CPLD; PWM; BLDC; Hall sensor; ADC.*

## ABSTRACT

This research presents the design and implementation of speed control for a brushless direct current (BLDC) motor using a complex programmable logic device (CPLD). Implementation of speed control is using a PWM technique by varying duty cycles applied to a three-phase inverter. Rotor position determines by hall sensors which are used as references to synchronize the PWM control signals. The control model is written using Verilog Hardware Description Language (HDL) and verified by simulation using ModelSim-Altera. An experimental setup is built to test the performance of the BLDC Motor under the PWM control. The control algorithm is implemented using Max II EPM240T100C5 devices on a 350W 36V rated BLDC motor. The number of the used logic elements (LEs) of the CPLD is about 133 of 240 LEs. PWM with controllable duty cycle generated in this system for having working frequency about 20KHz and commutate sequentially according to six-step commutation up to 600 RPM on no-load condition.

*Keywords: CPLD; PWM; BLDC; Hall sensor; ADC.*

## KATA PENGANTAR

*Bismillahirrahmanirrahim*, Dengan menyebut nama Allah yang Maha Pengasih lagi Maha Penyayang.

Puji syukur kehadiran Allah SWT atas rahmat dan karunia-Nya, penulis dapat menyelesaikan penyusunan skripsi yang berjudul “Sistem Kendali Motor BLDC dengan Hall Sensors Berbasis CPLD” untuk memenuhi persyaratan kurikulum sarjana strata-1 (S-1) pada Departemen Teknik Elektro, Fakultas Teknik, Universitas Hasanuddin, Gowa.

Walaupun demikian, dalam skripsi ini masih terdapat banyak kekurangan dan masih jauh dari kata sempurna yang disebabkan oleh keterbatasan kemampuan penulis. Oleh karena itu, saran dan kritik yang bersifat membangun sangat dibutuhkan dari semua pihak yang mana dapat membuat skripsi ini lebih baik di waktu yang akan datang.

Penulis menyadari bahwa penyusunan skripsi ini dari awal hingga selesai dapat terlaksana karena adanya bantuan, motivasi, serta bimbingan dari berbagai pihak. Maka, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Teristimewa kepada kedua orang tua penulis, Bapak Sachruddin Djafar dan Ibu Hartin Silawaty, yang menjadi sumber semangat dan motivasi yang dengan luar biasanya selalu mendukung baik dalam dukungan moral maupun materi dan tanpa henti selalu mendoakan keberhasilan penulis.
2. Bapak Prof. Dr-Ing Faizal Arya Samman, S.T., M.T. selaku pembimbing 1 dan Bapak Dr. Ir. H. Rhiza S. Sadjad, MSEE. selaku Pembimbing 2 yang telah meluangkan waktu di tengah kesibukannya selama penulis melaksanakan penelitian serta memberikan bimbingan, saran, dukungan, dan motivasinya dalam penyusunan skripsi ini.

3. Bapak Muh. Anshar, S.T., M.Sc. (Research), Ph.D dan Ibu Dr. Hj. A. Ejah Umraeni Salam, S.T., M.T. selaku dosen penguji yang telah memberikan kritik dan saran dalam penyusunan tugas akhir ini.
4. Ibu Dr. Eng. Ir. Dewiani, M.T. selaku Ketua Departemen Teknik Elektro, Fakultas Teknik, Universitas Hasanuddin.
5. Bapak/Ibu Dosen Fakultas Teknik Departemen Teknik Elektro atas bimbingan, arahan, didikan, dan motivasi yang telah diberikan selama ini.
6. Seluruh staf dan karyawan Fakultas Teknik Universitas Hasanuddin atas segala bantuannya selama penulis menempuh perkuliahan terutama kepada staf S1 Teknik Elektro.
7. Mutiah Rayhana, selaku sahabat yang selalu memotivasi dan mendorong serta selalu meluangkan waktu untuk membantu penulis dalam penyelesaian skripsi ini.
8. Dan kepada keluarga besar saya, teman, rekan penelitian, dan berbagai pihak yang tidak dapat disebutkan satu persatu, penulis ucapkan terima kasih atas tiap bantuan dan doa yang diberikan.

Makassar, 5 Januari 2022  
Penulis,

Muhammad Fajri Sachruddin

## DAFTAR ISI

<b>LEMBAR PENGESAHAN .....</b>	<b>ii</b>
<b>ABSTRAK .....</b>	<b>iii</b>
<b>ABSTRACT .....</b>	<b>iv</b>
<b>KATA PENGANTAR.....</b>	<b>v</b>
<b>DAFTAR ISI.....</b>	<b>vii</b>
<b>DAFTAR GAMBAR.....</b>	<b>ix</b>
<b>DAFTAR TABEL .....</b>	<b>xi</b>
<b>BAB 1 PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Deskripsi dan Rumusan Masalah .....	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Penulisan.....	4
<b>BAB 2 TINJAUAN PUSTAKA.....</b>	<b>6</b>
2.1 Motor BLDC .....	6
2.2 Inverter Tiga Fasa.....	11
2.3 Complex Programmable Logic Device .....	13
2.4 Komunikasi SPI.....	16
2.5 Pulse Width Modulation.....	17
<b>BAB 3 METODOLOGI PENELITIAN.....</b>	<b>18</b>
3.1 Gambaran Umum .....	18
3.2 Perancangan Sistem.....	20
3.2.1 Perancangan Kode Verilog .....	21
3.2.2 Perancangan Perangkat Keras.....	24
3.2.3 Perancangan Rangkaian Penyesuai Tegangan.....	26
3.3 Skenario Pengujian.....	29

3.3.1	Skenario Pengujian Sistem Kendali motor BLDC .....	29
3.3.2	Skenario Pengujian Karakteristik Kecepatan Motor BLDC Terhadap Beban.....	30
3.3.3	Skenario Rangkaian Rangkaian Penyesuai Tegangan .....	30
<b>BAB 4</b>	<b>HASIL DAN PEMBAHASAN .....</b>	<b>31</b>
4.1	Pengujian Sistem Kendali motor BLDC .....	31
4.1.1	Hasil Simulasi Verilog HDL.....	31
4.1.2	Hasil Implementasi .....	32
4.2	Hasil Pengujian Karakteristik kecepatan Motor BLDC terhadap Perubahan Beban.....	35
4.3	Hasil Pengujian Rangkaian Penyesuai Tegangan Secara Simulasi....	37
<b>BAB 5</b>	<b>PENUTUP .....</b>	<b>40</b>
5.1	Kesimpulan.....	40
5.2	Saran.....	40
<b>DAFTAR PUSTAKA</b>	<b>.....</b>	<b>42</b>
<b>LAMPIRAN</b>	<b>.....</b>	<b>45</b>

## DAFTAR GAMBAR

<b>Gambar 2.1</b>	Gaya yang dihasilkan oleh kawat berarus dalam medan magnet .....	7
<b>Gambar 2.2</b>	Gelombang motor BLDC. ....	9
<b>Gambar 2.3</b>	Gambar kontruksi motor BLDC [7]. ....	10
<b>Gambar 2.4</b>	Komutasi six-step motor BLDC. ....	11
<b>Gambar 2.5</b>	Topologi Umum Inverter Tiga Fasa .....	13
<b>Gambar 2.6</b>	Lini waktu perkembangan teknologi digital [17] .....	14
<b>Gambar 2.7</b>	Struktur dari complex programmable logic device (CPLD) [16]....	15
<b>Gambar 2.8</b>	Block diagram SPI.....	16
<b>Gambar 2.9</b>	Sinyal pulse width modulation .....	17
<b>Gambar 3.1</b>	Diagram alir tahapan penelitian.....	19
<b>Gambar 3.2</b>	Diagram blok sistem kontrol motor BLDC .....	20
<b>Gambar 3.3</b>	RTL Viewer dari desain Verilog HDL .....	21
<b>Gambar 3.4</b>	Komunikasi SPI MCP3008 .....	22
<b>Gambar 3.5</b>	Diagram alir kontrol BLDC.....	23
<b>Gambar 3.6</b>	Diagram blok perancangan perangkat keras.....	24
<b>Gambar 3.7</b>	Implementasi Perangkat Keras .....	25
<b>Gambar 3.8</b>	Karakteristik Rangkaian .....	27
<b>Gambar 3.9</b>	Rangkaian Penyesuai Tegangan .....	28
<b>Gambar 3.10</b>	Skenario pengujian dengan beban .....	30
<b>Gambar 4.1</b>	Hasil Simulasi pada Modelsim-Altera.....	31
<b>Gambar 4.2</b>	Perangkat Uji Coba.....	33
<b>Gambar 4.3</b>	Sinyal kendali motor BLDC pada duty cycle 50%.....	34
<b>Gambar 4.4</b>	Sinyal pada Fasa Motor BLDC.....	35

<b>Gambar 4.5</b> Grafik hubungan tegangan potentiometer terhadap kecepatan motor BLDC .....	36
<b>Gambar 4.6</b> Sinyal pada Fasa Motor BLDC.....	37
<b>Gambar 4.7</b> Sinyal keluaran rangkaian penyesuai tegangan .....	38
<b>Gambar 4.8</b> Grafik hubungan tegangan dua polaritas terhadap kecepatan motor BLDC .....	39

## DAFTAR TABEL

<b>Tabel 2.1</b> Perbandingan motor listrik pada kendaraan listrik.....	8
<b>Tabel 3.1</b> Clockwise Six-step Commutation.....	26

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan zaman saat ini terjadi di segala bidang di dunia ini, termasuk pada perkembangan kendaraan listrik. Pada teknologi kendaraan bermotor saat ini mulai terjadi transformasi dari pembakaran energi fosil menuju kendaraan listrik. [1] Pada implementasi kendaraan listrik, terdapat berbagai hambatan yang ditemui dan salah satunya yaitu pada sistem kendali [2], [3] sehingga banyak penelitian serta publikasi terkait untuk menghadapi masalah ini.

Motor listrik merupakan salah satu bagian yang penting pada kendaraan listrik. Terdapat berbagai macam tipe motor listrik dan salah satu motor yang banyak digunakan pada kendaraan listrik yaitu motor brushless DC (BLDC). Motor BLDC memiliki kelebihan dibanding motor dengan sikat dan motor induksi, antara lain memiliki efisiensi yang tinggi, kemudahan perawatan, serta memiliki tingkat densitas torsi yang tinggi [3]–[7]. Berdasarkan kelebihan-kelebihan inilah motor BLDC banyak digunakan pada aplikasi yang membutuhkan traksi seperti pada kendaraan listrik. Selain itu juga banyak digunakan dibidang bidang medis, penerbangan dan otomasi industri. Pada pengaplikasian untuk beberapa hal, motor DC memiliki kekurangan dibandingkan BLDC yaitu menimbulkan percikan serta masa pakai sikat [8]. Meskipun demikian pada motor BLDC dibutuhkan komutator elektronik, tidak seperti motor DC yang menggunakan komutator mekanik.

Pada motor BLDC, sistem komutasinya perlu diatur secara elektronik karena pada stator motor perlu dinyalakan-dimatikan secara berurutan dan teratur. Komutasi elektronik ini berdasarkan posisi dari magnet permanen pada rotor [4]. Posisi rotor dapat diketahui menggunakan sensor maupun secara sensorless. Pada metode sensorless dapat digunakan secara universal akan tetapi terdapat kekurangan tingkat kompleksitas algoritma yang tinggi, serta permulaan dan komutasi kecepatan rendah yang menyulitkan [4], [9]. Untuk penggunaan sensor dapat menggunakan sensor hall-effect yang lebih mudah digunakan, akan tetapi menaikkan ongkos produksi serta memiliki resolusi yang rendah meski beberapa penelitian terkait melakukan publikasi untuk menghadapi masalah tersebut [10], [11]. Pada penelitian ini digunakan sensor hall-effect yang terdapat pada motor BLDC.

Sistem pengendalian motor BLDC dapat dilakukan dengan beberapa cara, biasanya menggunakan Digital Signal Processors (DSPs) pada sebuah mikrokontroler. Selain menggunakan mikrokontroler, FPGA serta CPLD juga dapat digunakan yang merupakan rangkaian terintegrasi yang diprogram menggunakan hardware description language (HDL). Implementasi pada CPLD memungkinkan dilakukannya simulasi terhadap rangkaian logika serta penggunaan HDL memungkinkan diimplementasikan pada *system-on-chip* untuk produksi massal. Sehingga pada penelitian ini digunakan CPLD sebagai otak untuk implementasi komutasi elektronik. Dari perancangan yang dilakukan dapat digunakan untuk melihat karakteristik motor BLDC untuk pengembangan sistem kendali yang lebih baik.

Dari permasalahan tersebut maka dilakukan penelitian tugas akhir dengan judul **“SISTEM KENDALI MOTOR BLDC DENGAN HALL SENSORS BERBASIS CPLD”**.

## **1.2 Deskripsi dan Rumusan Masalah**

Dalam pengendalian motor BLDC, komutasinya diatur secara elektronik berdasarkan posisi dari magnet permanen pada rotor. Untuk mengetahui posisi rotor maka dapat digunakan sensor hall-effect pada motor BLDC sehingga dapat melakukan komutasi pada motor BLDC.

Berdasarkan deskripsi tersebut peneliti merumuskan pokok permasalahan, yaitu perancangan sistem pengendalian motor BLDC dengan memanfaatkan sensor hall-effect berbasis CPLD.

## **1.3 Tujuan Penelitian**

Tujuan yang hendak dicapai dari penelitian ini antara lain:

1. Merancang dan mengimplementasikan sistem kendali motor BLDC menggunakan CPLD dan hall sensor.
2. Mengetahui karakteristik kecepatan motor BLDC terhadap kondisi beban.

## **1.4 Batasan Masalah**

Dalam melaksanakan penelitian ini, permasalahan yang akan dibahas dibatasi dengan ketentuan berikut:

1. Pengujian dan implementasi CPLD sebagai pembangkit sinyal PWM untuk menggerakkan motor BLDC dengan metode *six-step commutation*.
2. Implementasi diuji dengan menggunakan sumber tegangan 36 VDC yang terdapat pada laboratorium elektronika dan divais.
3. Pengambilan data karakteristik motor BLDC berfokus pada hubungan tegangan potentiometer dan kecepatan motor.

## **1.5 Sistematika Penulisan**

Adapun sistematika penulisan dari penelitian ini adalah sebagai berikut:

### **BAB 1 PENDAHULUAN**

Bab ini berisi uraian tentang latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, dan sistematika penulisan.

### **BAB 2 TINJAUAN PUSTAKA**

Bab ini menjelaskan teori-teori pendukung materi penelitian yang diambil dari berbagai sumber ilmiah yang digunakan dalam penulisan laporan tugas akhir ini.

### **BAB 3 METODOLOGI PENELITIAN**

Bab ini membahas tentang metode penelitian yang digunakan dalam tugas akhir ini.

### **BAB 4 HASIL DAN PEMBAHASAN**

Bab ini menampilkan data dan pembahasan hasil prototipe sistem yang telah dibuat.

## **BAB 5 PENUTUP**

Bab ini berisi tentang kesimpulan yang didapatkan berdasarkan penelitian dan saran untuk pengembangan penelitian kedepan.

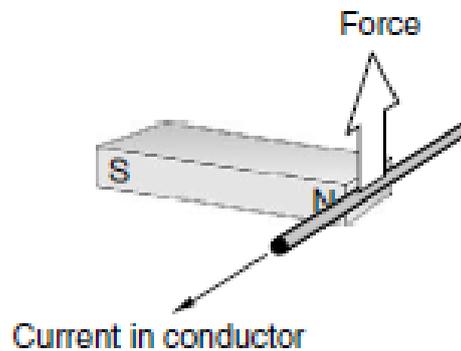
## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Motor BLDC**

Mesin listrik merupakan alat yang dapat mengubah energi mekanik menjadi energi listrik dan sebaliknya. Alat yang dapat mengubah energi listrik menjadi energi mekanik dikenal sebagai motor. Alat yang berfungsi sebaliknya, mengubah energi mekanik menjadi energi listrik disebut generator atau dinamo. Motor listrik sendiri dapat ditemukan pada berbagai aplikasi seperti pada peralatan rumah tangga contohnya kipas angin, mesin cuci, pompa air dan penyedot debu, hingga kepada dunia otomotif [2], [8].

Motor listrik bekerja dari gaya yang timbul dari konduktor berarus yang diletakkan pada suatu medan magnet seperti yang diilustrasikan gambar 2.1. Untuk membuat gaya yang besar, dibutuhkan medan magnet yang kuat dan berinteraksi dengan banyak konduktor berarus. Untuk itu secara umum prinsip kerja motor yaitu: Arus Listrik dalam Medan Magnet akan memberikan gaya, jika kawat yang membawa arus dibengkokkan menjadi sebuah loop maka kedua sisi loop, yaitu pada sudut kanan medan magnet, akan mendapatkan gaya pada arah yang berlawanan. Pasangan gaya menghasilkan tenaga putar / torque untuk memutar kumparan. Motor-motor memiliki beberapa loop pada dinamonya untuk memberikan tenaga putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan elektromagnetik yang disebut kumparan medan [2].



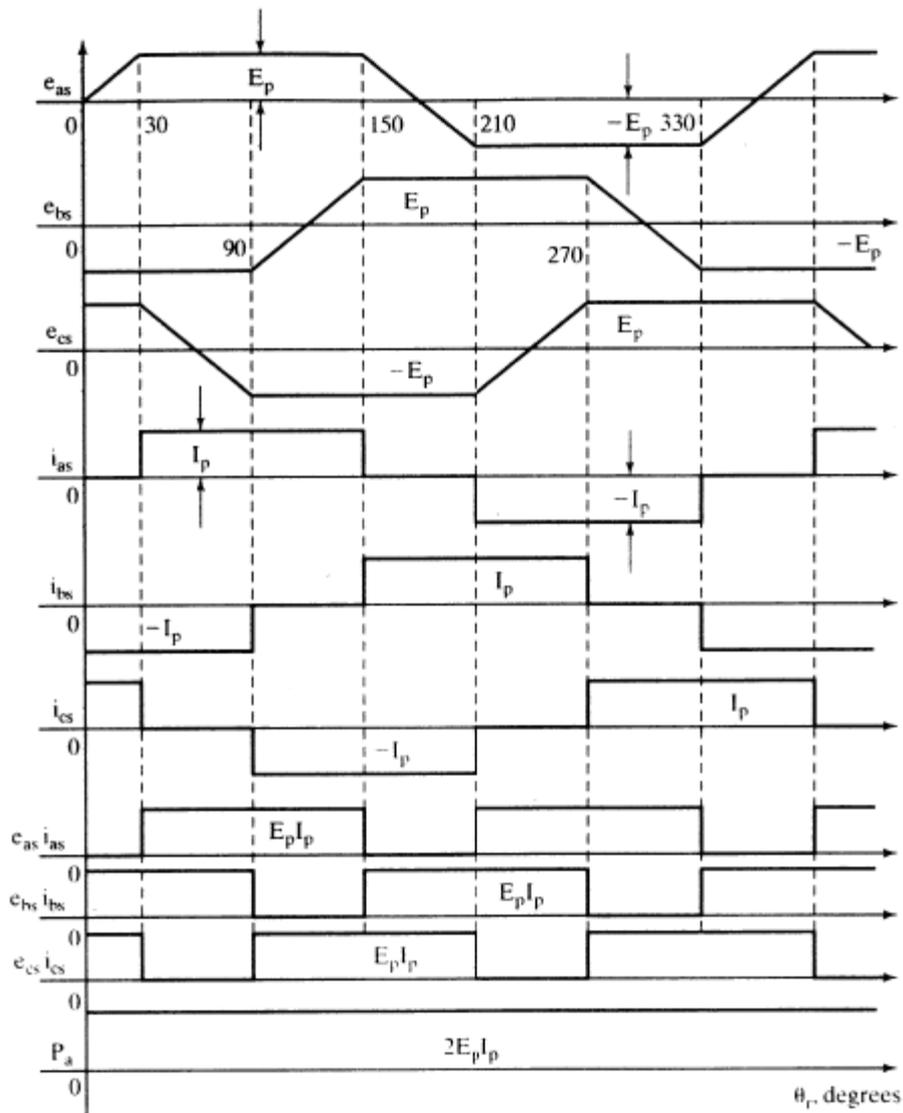
**Gambar 2.1** Gaya yang dihasilkan oleh kawat berarus dalam medan magnet

Motor Listrik sendiri memiliki beragam konfigurasi dan tipe, pada pembahasan ini kita membagi menjadi dua jenis motor, yaitu Motor AC dan Motor DC. Motor AC pun terbagi menjadi dua jenis, yaitu Motor Sinkron dan Motor Induksi. Motor Induksi terbagi lagi menjadi dua macam yaitu Motor Induksi 1 Fasa dan Motor Induksi 3 Fasa. Sedangkan Motor DC terbagi menjadi dua jenis yaitu Separately Excited dan Self Excited. Motor DC Self Excited terbagi menjadi 3 buah macam yaitu SE Seri, Campuran, dan Shunt. Dengan beragamnya motor listrik ini, tabel 2.1 menyajikan perbandingan terhadap aplikasi di dunia otomotif [7].

**Tabel 2.1** Perbandingan motor listrik pada kendaraan listrik

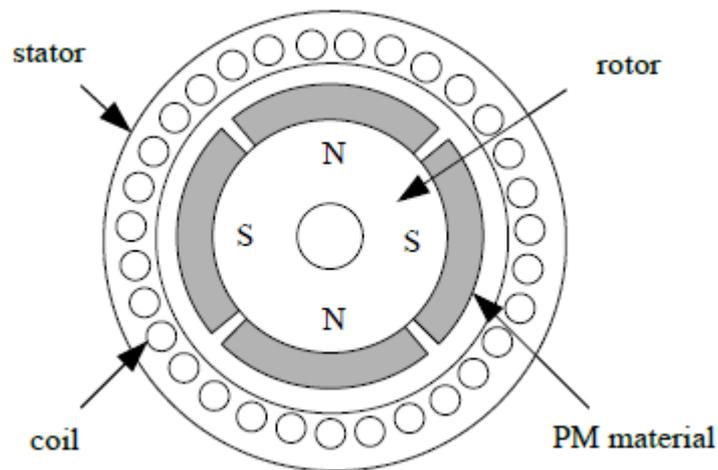
Motor type	DC motor	Induction motor	PM motor	Switched reluctance motor
Performance index				
Power density	Low	Intermediate	High	Very high
Peak efficiency (%)	<90	90–95	95–97	<90
Load efficiency (%)	80–87	90–92	85–97	78–86
Controllability	Simple	Complex	Hard for field-weakening	Complex
Reliability	Normal	Good	Excellent	Good
Heat dissipation	Bad	Bad	Good	Good
Size & weight	Big, Heavy	Normal, Normal	Small, Light	Small, Light
High-speed performance	Poor	Excellent	Good	Excellent
Construction	Slightly worse	Better	Slightly better	Excellent
Cost of motor (\$/kW)	10	8–10	10–15	6–10
Cost of controller	Low	High	High	Normal
Combination property	Slightly worse	Normal	Excellent	Better

Motor BLDC atau motor *brushless DC* merupakan motor yang secara definisi merupakan motor yang memiliki rotor berupa permanen magnet dan tipe motor *self-synchronous* yang dikendalikan menggunakan komutator elektronik. Beberapa mengenal dan mengategorikan motor ini sama dengan motor PMSM, namun juga ada yang menganggap motor ini berbeda dikarenakan beda gelombang yang dimiliki. BLDC memiliki gelombang *Back EMF* (BEMF) berbentuk trapezoid dan untuk motor PMSM memiliki gelombang sinusoidal [7]. Pada gambar 2.2 dapat dilihat bentuk gelombang dari motor BLDC [12].



**Gambar 2.2** Gelombang motor BLDC.

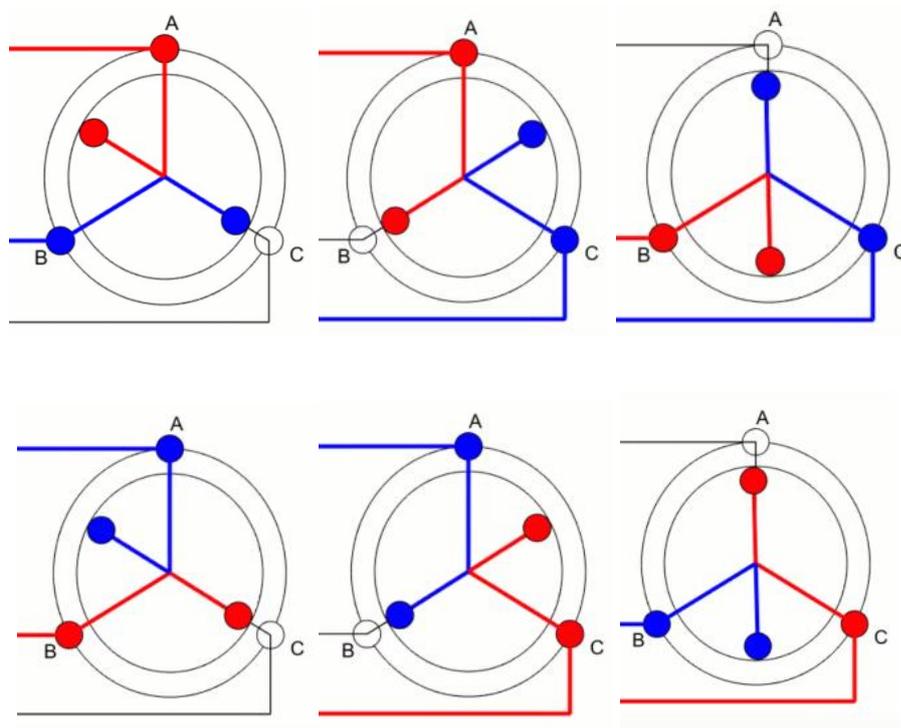
Konstruksi motor BLDC terdiri dari dua komponen utama, yaitu rotor dan stator. Gambar 2.3 menunjukkan konstruksi dasar motor BLDC yang memiliki permanen magnet material pada rotornya dan *winding* pada stator. *Winding* pada motor BLDC dapat dihubung dengan koneksi Why ataupun Delta. Pemilihan koneksi yang umum digunakan berupa Why yang dihubungkan simetris pada titik netral dan memiliki performa dan biaya yang lebih baik [7].



**Gambar 2.3** Gambar konstruksi motor BLDC [7].

Untuk melakukan komutasi pada motor BLDC kumparan pada stator diberikan arus tiga fasa sehingga menimbulkan medan magnet pada inti besi. Medan magnet inilah yang akan saling menghasilkan gaya tolak menolak terhadap rotor yang menyebabkan motor berputar. Untuk melakukan komutasi yang terus menerus, polaritas kumparan perlu diubah setiap saat sesuai posisi rotor-stator [13].

Dalam komutasi pada motor BLDC, terdapat dua mode konduksi dari fasa motor BLDC yaitu mode dua fasa dan mode tiga fasa. Mode yang pertama ialah mode konduksi dua fasa, yang pada prinsipnya melakukan konduksi dua kumparan dari motor pada satu waktu. Waktu konduksi ditentukan dari posisi rotor. Dalam komutasinya, motor BLDC berkomutasi  $60^\circ$  sudut elektrik. Pada mode ini terdapat enam keadaan magnet dan dua kumparan yang bekerja pada setiap posisinya, ini juga dikenal sebagai six-step commutation. Waktu arus yang mengalir pada kumparan secara terus menerus ialah  $120^\circ$  sudut elektrik [7].



**Gambar 2.4** Komutasi six-step motor BLDC.

## 2.2 Inverter Tiga Fasa

Inverter merupakan rangkaian elektronika daya yang digunakan untuk mengkonversikan atau mengubah tegangan searah (DC) ke tegangan bolak-balik (AC). Untuk penggunaan dalam penggerak motor AC dan uninterruptible ac power supply digunakan switch-mode inverter untuk menghasilkan sebuah keluaran tegangan bolak-balik berupa sinusoidal yang besar dan frekuensinya dapat dikendalikan. Ada beberapa jenis inverter yang dibagi berdasarkan kriterianya, antara lain adalah:[8], [14]

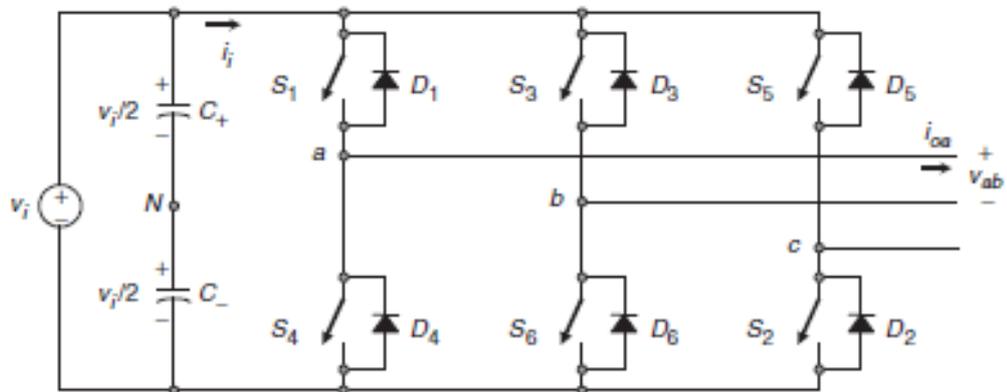
- a. Berdasarkan Jumlah fasanya: yaitu inverter satu-fasa dan banyak-fasa, misalnya tiga-fasa yang selama ini merupakan jenis inverter yang

digunakan untuk menginterkoneksi keluarannya ke jala-jala sistem tenaga listrik (grid).

- b. Berdasarkan sumber masukan DC-nya: yaitu Inverter sumber tegangan (VSI – Voltage Source Inverter), dan Inverter sumber arus (CSI – Current Source Inverter).
- c. Berdasarkan metode bentuk sinyal pengaturannya: yaitu gelombang persegi, pulse amplitudo modulation (PAM) dan pulse width modulation (PWM).
- d. Berdasarkan bentuk gelombang keluarannya: yaitu gelombang persegi, persegi berundak dan sinusoidal

Pemakaian peralatan tersebut dipilih didasarkan pada jenis penerapannya. Inverter biasanya memakai sinyal kontrol modulasi lebar pulsa (PWM) untuk menghasilkan tegangan keluaran bolak-balik. Sebuah Inverter dikategorikan sebagai jenis inverter sumber tegangan (Voltage fed Inverter) jika masukannya berupa sumber tegangan, dan dikategorikan sebagai inverter sumber arus (Current Fed Inverter) jika sumber dayanya berupa sumber arus[14].

Dalam pengaplikasian pada uninterruptible ac power supply dan penggerak motor ac, umumnya inverter tiga fasa digunakan untuk menyuplai beban tiga fasa. Ini dimungkinkan untuk menyuplai beban tiga fasa menggunakan tiga inverter satu fasa secara terpisah dengan keluaran yang terpisah 120 derajat, akan tetapi hal ini membutuhkan masukan yang berbeda[14]. Untuk itu dalam inverter tiga fasa memiliki topologi yang secara umum sebagai berikut: [15]



**Gambar 2.5** Topologi Umum Inverter Tiga Fasa

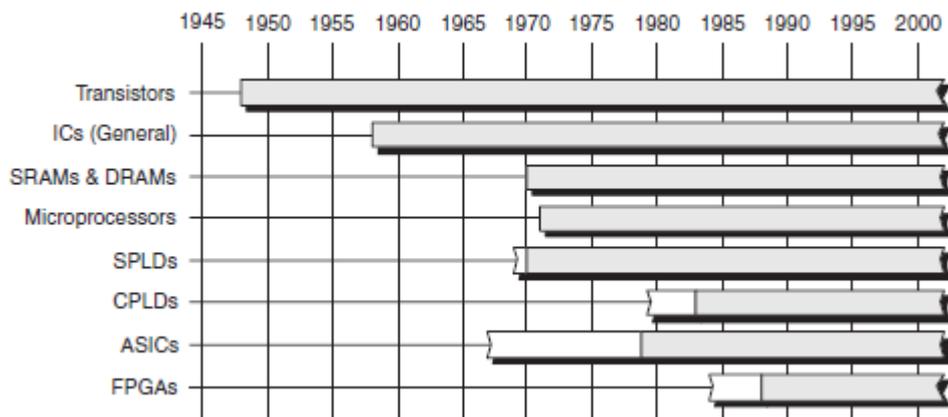
Inverter tiga fasa ini tersusun dari dapat terdiri enam buah switch, dapat berupa power MOSFET, IGBT dan lainnya yang dirangkai secara berpasangan. Agar dapat menghasilkan tegangan tiga fasa, masing-masing MOSFET harus diberikan sinyal kontrol yang sesuai.

### 2.3 Complex Programmable Logic Device

Dalam perkembangan dunia perangkat keras yang digunakan dalam membangun perangkat komputasi, di dalamnya terdapat rangkaian logika dan perangkat keras yang terlibat disebut sebagai perangkat keras digital. Hal tersebut diturunkan dari informasi yang direpresentasikan pada elektronik.

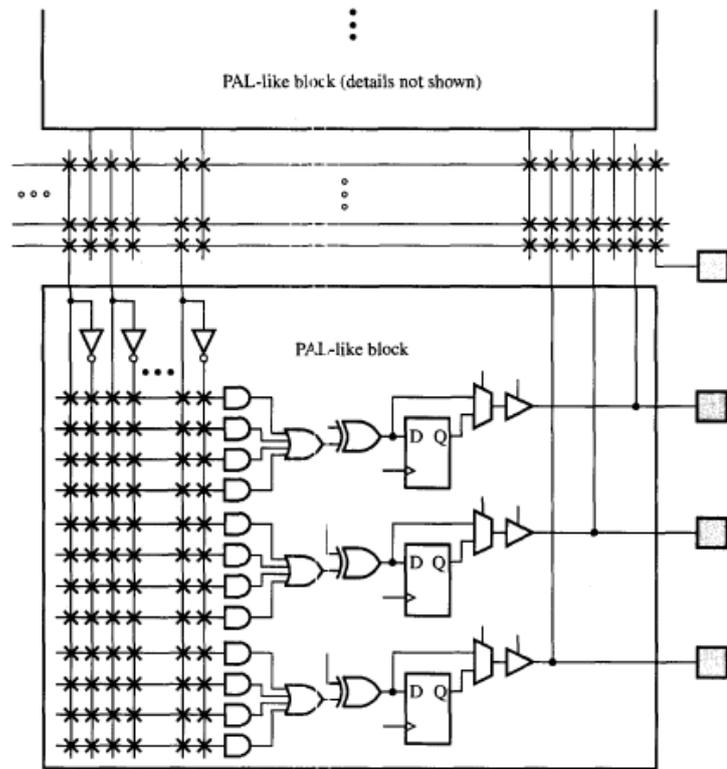
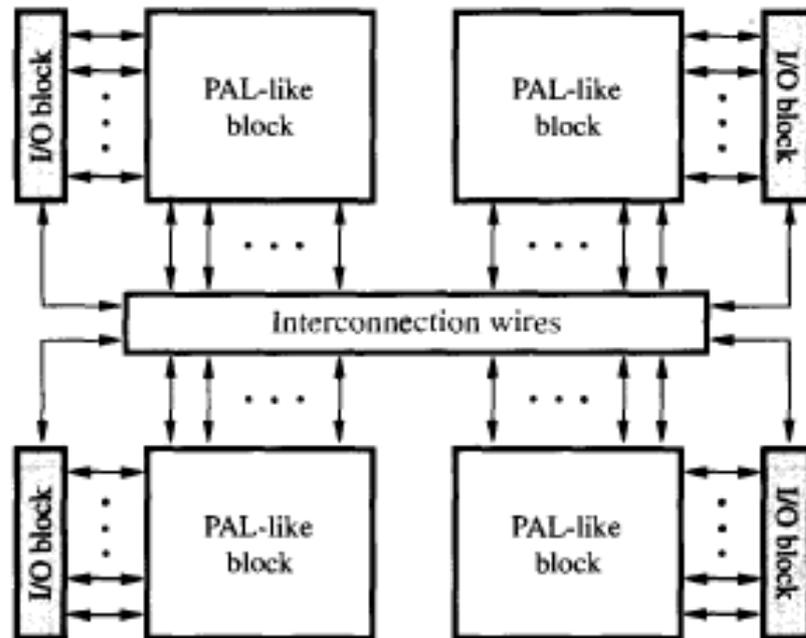
Pada dunia digital meliputi perangkat keras yang bermula dari transistor, dan transistor yang semakin berkembang hingga diciptakannya chip dari ukuran transistor yang dapat semakin kecil. Dalam perancangan produk hardware digital ini dapat dilakukan mulai dari desain dan membangun rangkaian logika dari dasar.

Untuk ini rangkaian tersebut dapat dibagi menjadi standard chips, programmable logic devices, dan custom chips [16].



**Gambar 2.6** Lini waktu perkembangan teknologi digital [17]

Programmable logic device sendiri dapat terdiri dari berbagai jenis seperti terlihat pada gambar 2.6, termasuk di dalamnya ialah CPLD. CPLD merupakan peningkatan dari PLA serta PAL yang pada perangkatnya membutuhkan lebih banyak. Pada CPLD strukturnya terdiri dari beberapa block circuit pada sebuah chip seperti struktur yang dapat dilihat pada gambar 2.7.

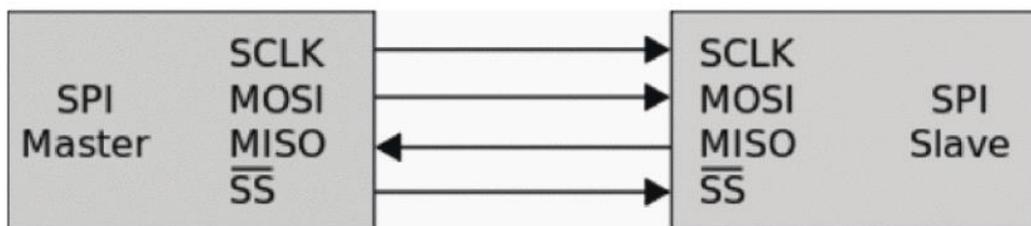


**Gambar 2.7** Struktur dari complex programmable logic device (CPLD) [16]

Perancangan rangkaian logika yang melibatkan implementasi pada hardware dapat dilakukan menggunakan *Hardware Descriptive Language* dan salah satunya merupakan Verilog HDL. Dengan melakukan perancangan dengan Bahasa ini, kita dapat melakukan sintesis rangkaian logika tersebut hingga diimplementasikan pada CPLD [16].

## 2.4 Komunikasi SPI

Dalam komunikasi data antar perangkat terdapat berbagai jenis yang dapat menghubungkan kedua perangkat dan salah satunya adalah Serial Peripheral Interface. Komunikasi SPI merupakan jenis protocol yang tersinkron yang menghubungkan perangkat master dan slave [18].



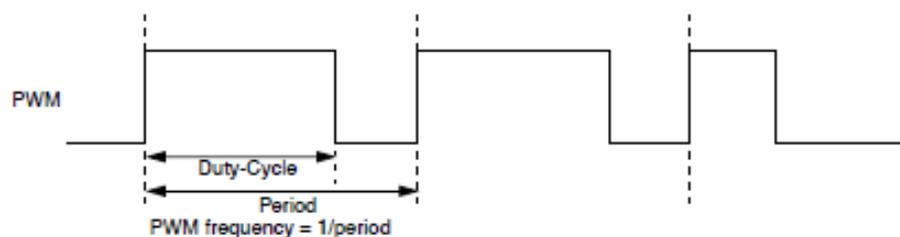
**Gambar 2.8** Block diagram SPI

Gambar 2.8 menunjukkan blok diagram dari komunikasi SPI. SPI dapat melakukan komunikasi secara sinkron dengan perangkat master menyediakan clock. Perubahan clock pada SPI dapat terjadi tanpa mengganggu proses pengiriman data dan perubahan ini hanya mempengaruhi data rate. SPI merupakan protocol master-slave seperti pada gambar 2.8 [18].

## 2.5 Pulse Width Modulation

Metode-metode dalam pengendalian Motor memiliki banyak variasi, yaitu Direct Torque Control (DTC), Field Oriented Control (FOC), Space Vector Pulse Width Modulation (SVPWM), Space Vector Modulation based Direct Torque Control (SVM-DTC), dan PWM Generator. Metode yang paling umum untuk pengendalian kecepatan secara umum untuk motor adalah PWM. Metode ini dilakukan dengan mencocokkan sebuah sinyal frekuensi tinggi dengan duty cycle tertentu yang dikalikan jumlah sinyal switching dari inverter [19].

Metode kendali ini hanya menggunakan sinyal PWM yang dihasilkan melalui Microcontroller atau FPGA untuk mengendalikan Motor berupa sinyal pulsa yang disusun sedemikian rupa agar motor dapat dikendalikan. Pulse width modulation (PWM) secara umum adalah sebuah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam satu periode, untuk mendapatkan tegangan rata-rata yang berbeda. Modulasi PWM ini dilakukan dengan cara mengubah perbandingan lebar pulsa positif terhadap lebar pulsa negative ataupun sebaliknya dalam frekuensi sinyal yang tetap. Artinya, total 1 periode (T) pulsa dalam PWM tetap.



**Gambar 2.9** Sinyal pulse width modulation

## BAB 3

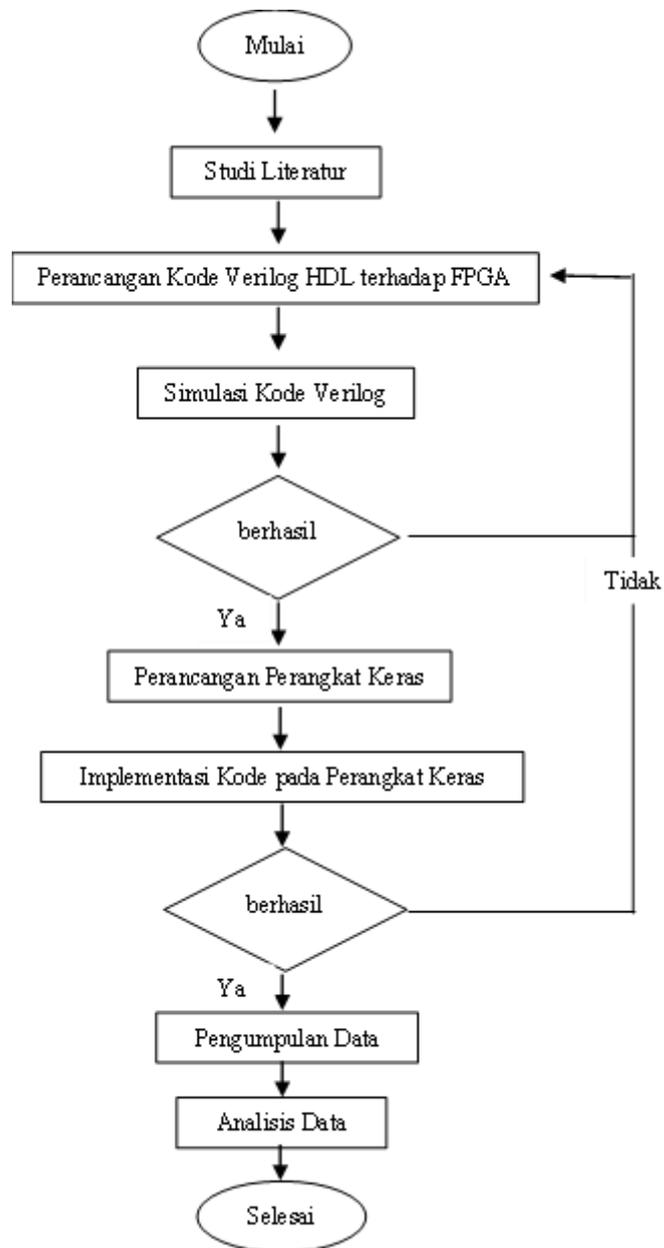
### METODOLOGI PENELITIAN

#### 3.1 Gambaran Umum

Secara garis besar, maka proses penelitian ini dibagi menjadi beberapa tahap yaitu sebagai berikut:

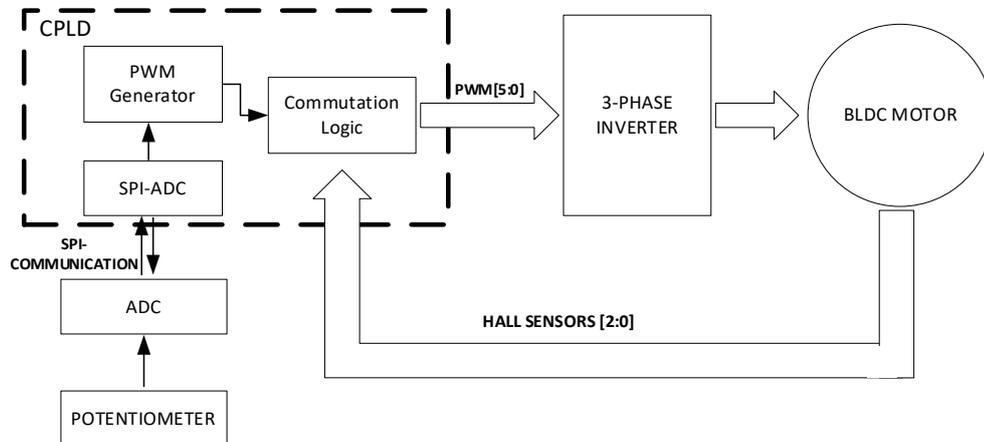
- a. Studi pustaka, mengumpulkan referensi yang berhubungan dengan penelitian dan mempelajari teori-teori dasar yang menunjang dalam penelitian.
- b. Perancangan sistem, melakukan perencanaan dan implementasi mengenai sistem yang akan diuji. Perancangan sistem meliputi perancangan perangkat lunak berupa kode Verilog HDL dan perancangan perangkat keras.
- c. Pengujian perangkat, untuk memastikan perangkat telah mampu berjalan dan bekerja sesuai dengan fungsi yang diberikan untuk memastikan kelancaran proses pengambilan data. Apabila terdapat *error*, maka dilakukan *troubleshooting*.
- d. Pengambilan data, melakukan pengujian dan pengukuran kinerja jaringan terhadap serangkaian skenario pengujian yang ditentukan. Apabila telah berjalan sebagaimana mestinya, maka dilanjutkan dengan menganalisa data dan menarik kesimpulan akhir.

Berikut adalah keseluruhan tahapan metodologi dalam bentuk flowchart yang dapat dilihat pada gambar 3.1.



**Gambar 3.1** Diagram alir tahapan penelitian

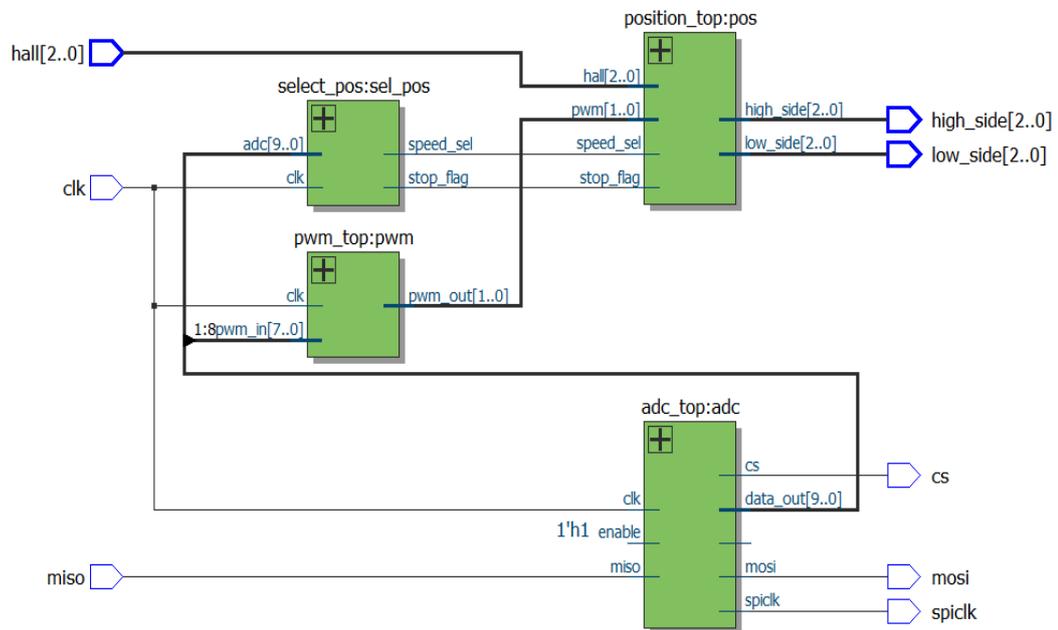
### 3.2 Perancangan Sistem



**Gambar 3.2** Diagram blok sistem kontrol motor BLDC

Pada pengendalian motor BLDC, komutasi motor diatur secara elektronik. Secara garis besar, perancangan sistem yang akan dilakukan dapat dilihat pada gambar 3.2. Dalam sistem yang dirancang terdapat dua input yang pertama dari potentiometer melalui ADC, kemudian hall sensor yang berada pada motor digunakan untuk mengetahui posisi rotor. Proses utama dari sistem diolah oleh CPLD, berdasarkan kedua inputan tersebut maka CPLD akan membangkitkan sinyal PWM yang akan digunakan untuk mengaktifkan gate pada inverter tiga fasa. Kecepatan motor BLDC setara dengan masukan tegangan pada motor, untuk itu perlu pengendalian tegangan pada motor BLDC[19]. Untuk pengendalian tersebut dilakukan dengan mengubah duty cycle dari sinyal PWM.

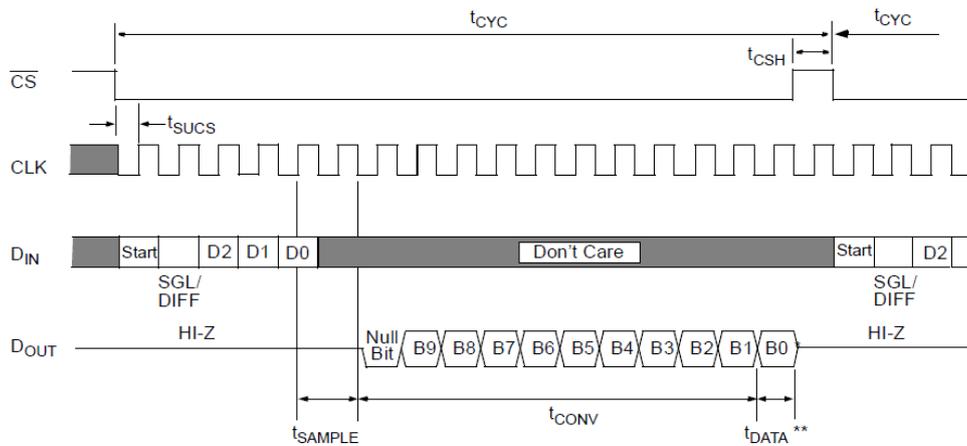
### 3.2.1 Perancangan Kode Verilog



**Gambar 3.3** RTL Viewer dari desain Verilog HDL

Pada perancangan kode Verilog ini dibagi menjadi beberapa blok logika, antara lain untuk komunikasi SPI ADC, PWM generator, dan blok komutasi. Masing-masing blok bekerja secara Bersama-sama untuk menghasilkan keluaran sinyal PWM yang akan digunakan pada motor BLDC.

Blok komunikasi SPI-ADC digunakan karena pada CPLD tidak terdapat ADC internal, sehingga digunakan ADC eksternal dengan tipe MCP3008 yang merupakan chip ADC 10-bit dengan protocol SPI.



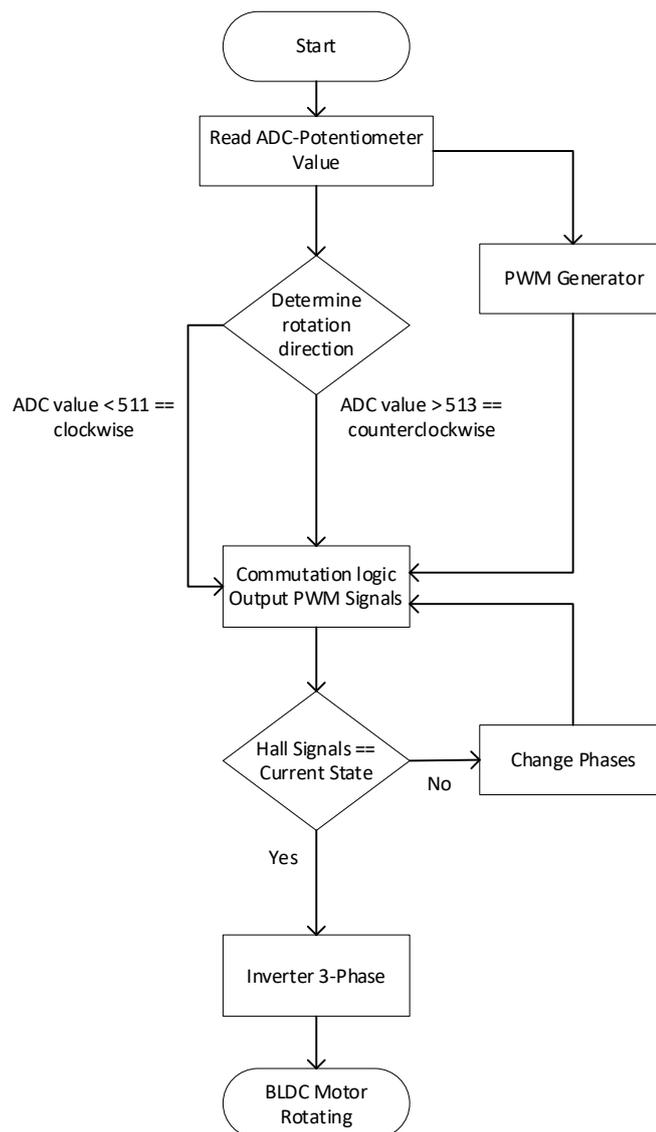
**Gambar 3.4** Komunikasi SPI MCP3008

Untuk menggunakan MCP3008 dilakukan komunikasi antara CPLD dengan ADC, dengan CPLD sebagai master dan ADC sebagai slave. Untuk berkomunikasi dengan protocol SPI diperlukan clock yang sama antar CPLD dan ADC. Dalam menginisiasi komunikasi diperlukan sinyal low pada CS untuk menandakan mulai terjadinya pertukaran data seperti tertera pada gambar 3.4 [20]. Seluruh sistem komunikasi ini dirancang berdasarkan FSM, dengan data dikirim melalui MOSI dan MISO. Berikut ini nilai keluaran digital dari ADC didapat dari persamaan di bawah:

$$Digital\ output\ code = \frac{1024 \times V_{in}}{V_{ref}} \quad (1)$$

Untuk mengendalikan motor BLDC dilakukan dengan mengatur tegangan yang diberikan kepada motor BLDC dikarenakan tegangan yang masuk proporsional dengan kecepatan motor. Salah satu cara mengaturnya ialah menggunakan *pulse width commutation* (PWM).

Dalam membangkitkan sinyal PWM, CPLD akan menerima input berupa nilai digital dari ADC serta nilai hall sensor. Sinyal PWM yang akan dibangkitkan yaitu memiliki frekuensi 20 KHz. Gambar 3.5 menunjukkan proses yang terjadi dan logika yang diimplementasikan pada kode Verilog HDL.

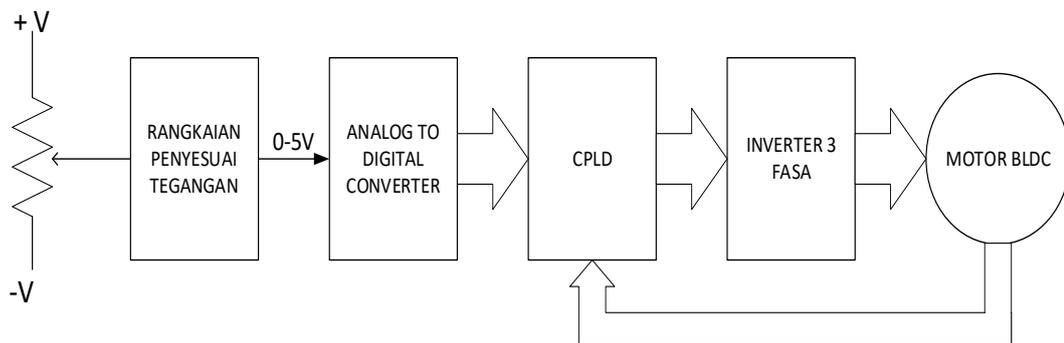


**Gambar 3.5** Diagram alir kontrol BLDC

Untuk blok komutasi dilakukan proses sesuai dengan six-step commutation. Hubungan antar posisi rotor dan fasa yang aktif dapat mengikuti datasheet ataupun

ditentukan secara manual [21]. Pada blok ini diimplementasikan logika sesuai pada tabel 3.1 dan tabel 3.2.

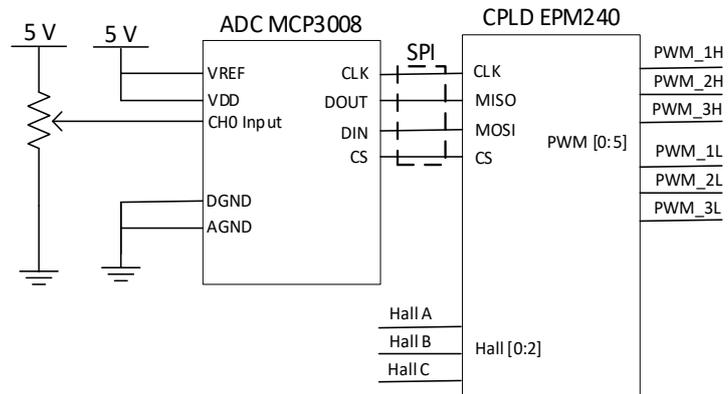
### 3.2.2 Perancangan Perangkat Keras



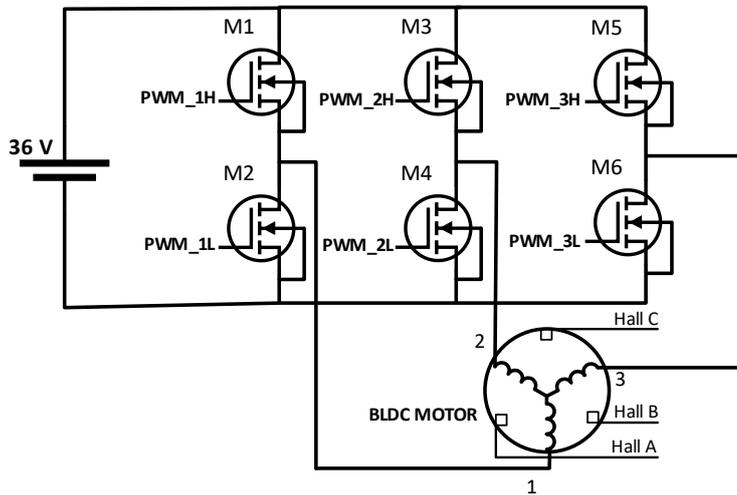
**Gambar 3.6** Diagram blok perancangan perangkat keras

Secara garis besar perancangan pada perangkat keras terdapat pada gambar 3.6. Perangkat keras terdiri dari rangkaian penyesuai tegangan yang berfungsi menyesuaikan tegangan masukan ke cakupan tegangan yang bisa digunakan pada ADC. Selanjutnya masuk ke CPLD dan menuju inverter tiga fasa yang kemudian masuk ke motor BLDC.

Pada perancangan perangkat keras meliputi perancangan rangkaian inverter tiga fasa beserta driver gate mosfet. Komponen dalam perancangan inverter tiga fasa yaitu menggunakan MOSFET bertipe IRF3808, dan untuk komponen driver gate menggunakan TLP250 serta IR2110.



(a) Input and output connection



(b) 3-phase Inverter

### Gambar 3.7 Implementasi Perangkat Keras

Gambar 3.7 menunjukkan hubungan koneksi pada perangkat keras. Dapat dilihat tegangan potentiometer masuk ke dalam ADC dengan tegangan maksimum 5V serta tegangan ref ADC juga 5V. kemudian masuk ke CPLD dengan komunikasi SPI dan CPLD menerima juga masukan dari hall sensor. Hubungan fasa yang aktif dan posisi rotor dapat dilihat di bawah:

**Tabel 3.1** Clockwise Six-step Commutation

Hall Sensors			Motor Phases			MOSFET					
<i>A</i>	<i>B</i>	<i>C</i>	<i>Phase 1</i>	<i>Phase 2</i>	<i>Phase 3</i>	<i>1L</i>	<i>1H</i>	<i>2L</i>	<i>2H</i>	<i>3L</i>	<i>3H</i>
0	0	1	Off	-	+	0	0	1	0	0	1
0	1	1	-	Off	+	1	0	0	0	0	1
0	1	0	-	+	Off	1	0	0	1	0	0
1	1	0	Off	+	-	0	0	0	1	1	0
1	0	0	+	Off	-	0	1	0	0	1	0
1	0	1	+	-	Off	0	1	1	0	0	0

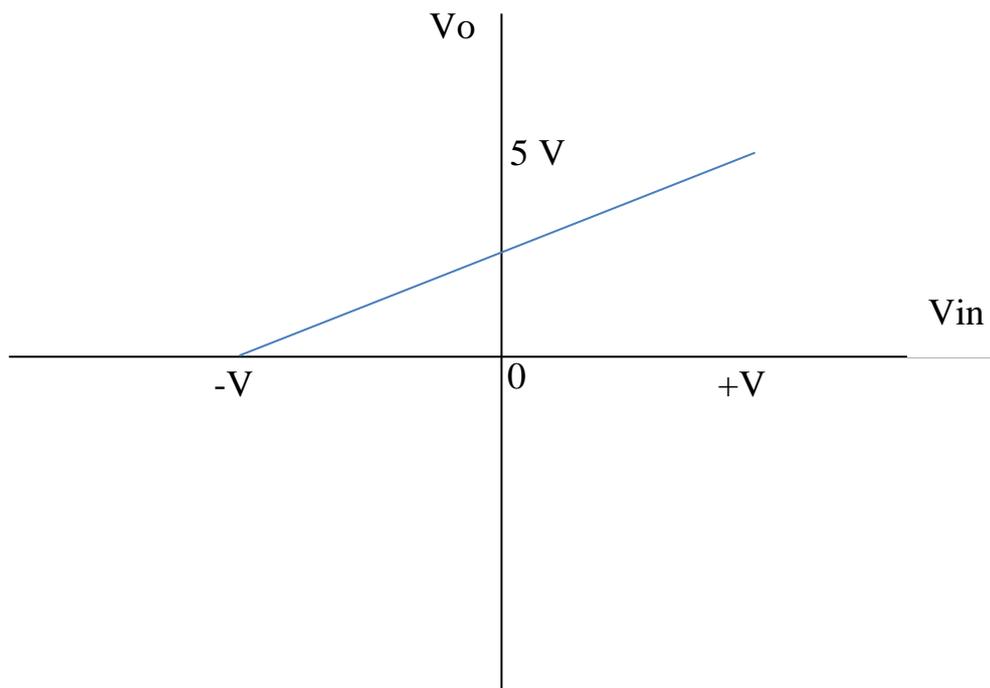
**Tabel 3.2** CounterClockwise Six-step Commutation

Hall Sensors			Motor Phases			MOSFET					
<i>A</i>	<i>B</i>	<i>C</i>	<i>Phase A</i>	<i>Phase B</i>	<i>Phase C</i>	<i>1L</i>	<i>1H</i>	<i>2L</i>	<i>2H</i>	<i>3L</i>	<i>3H</i>
0	0	1	Off	+	-	0	0	0	1	1	0
0	1	1	+	Off	-	0	1	0	0	1	0
0	1	0	+	-	Off	0	1	1	0	0	0
1	1	0	Off	-	+	0	0	1	0	0	1
1	0	0	-	Off	+	1	0	0	0	0	1
1	0	1	-	+	Off	1	0	0	1	0	0

### 3.2.3 Perancangan Rangkaian Penyesuai Tegangan

Dalam keperluan pengendalian, BLDC sebaiknya dapat menerima masukan tegangan 2 polaritas positif serta negatif. Hal ini dikarenakan, terdapat beberapa sensor yang memiliki keluaran dengan dua polaritas seperti sensor posisi sehingga akan membantu dalam penggunaan pada daur tertutup.

Rancangan rangkaian untuk dapat menerima dua polaritas harus memiliki karakteristik seperti pada gambar 3.8 dan persamaan 3.1. Karakteristik menunjukkan persamaan linear dengan tegangan keluaran 0 – 5 Volt dengan nilai  $b = 2,5$  dan  $a = 2,5/V$  pada persamaan 3.1.



**Gambar 3.8** Karakteristik Rangkaian

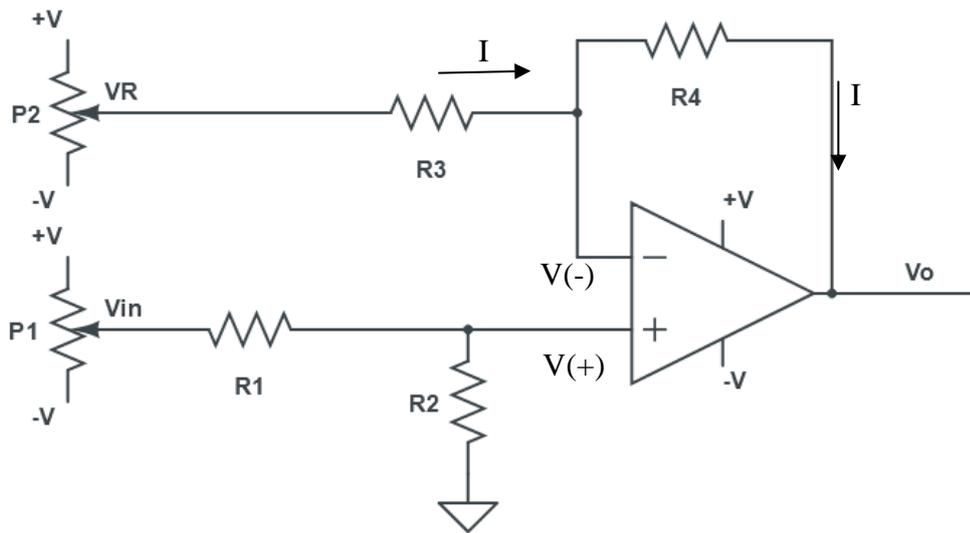
$$V_o = aV_i + b \quad (3.1)$$

$$V_i = +V, \text{ maka } V_o = 5 \text{ V}$$

$$V_i = -V, \text{ maka } V_o = 0 \text{ V}$$

Untuk menghasilkan rancangan dengan karakteristik tadi, salah satu caranya yaitu menggunakan rangkaian dengan 1-OpAmp, seperti pada gambar 3.9. Menggunakan karakteristik dari Op-Amp yaitu tegangan pada masukan non-

inverting dan masukan inverting sama besar, didapatkan persamaan umum seperti pada persamaan 3.4.



**Gambar 3.9** Rangkaian Penyesuai Tegangan

Pada masukan non-inverting serta masukan inverting dari op-amp, dapat ditentukan dengan persamaan 3.2 dan 3.3.

$$V(+)=\frac{R2}{R1+R2}\times Vin \quad (3.2)$$

$$\frac{VR-V(-)}{R3}=\frac{V(-)-Vo}{R4}=I \quad (3.3)$$

$$Vo=\left\{R4\times V(-)\times\left(\frac{1}{R4}+\frac{1}{R3}\right)\right\}-\left(\frac{R4}{R3}\times VR\right)$$

$$Vo=\left(1+\frac{R4}{R3}\right)\times V(-)-\left(\frac{R4}{R3}\times VR\right)$$

Dengan  $V(+)=V(-)$ , maka

$$Vo=\left(1+\frac{R4}{R3}\right)\times\left(\frac{R2}{R1+R2}\right)\times Vin-\frac{R4}{R3}\times VR \quad (3.4)$$

$$= aV_{in} + b$$

Sehingga untuk mendapatkan fungsi yang kita inginkan sesuai karakteristik nilai R1, R2, R3, R4 dan VR dapat disesuaikan sesuai masukan tegangan yang digunakan.

### **3.3 Skenario Pengujian**

Pengujian sistem kendali motor BLDC akan dilakukan dalam bentuk simulasi serta implementasi secara langsung. Pengujian ini bertujuan untuk melakukan validasi dan verifikasi terhadap hasil yang diinginkan.

Proses pengujian yang dilakukan terbagi atas 3 skenario, yaitu:

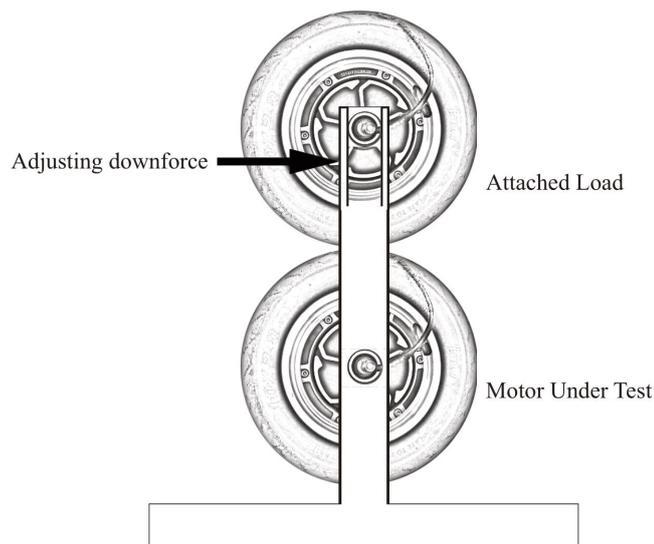
1. Pengujian sistem kendali motor BLDC.
2. Pengujian karakteristik kecepatan motor BLDC terhadap beban.
3. Pengujian rangkaian penerima tegangan dua polaritas secara simulasi.

#### **3.3.1 Skenario Pengujian Sistem Kendali motor BLDC**

Dalam skenario ini, pengujian dilakukan menggunakan simulasi modelsim-altera serta implementasi secara langsung pada perangkat keras. Pada pengujian ini akan dilihat logika yang berjalan pada CPLD dan diverifikasi kesesuaian dengan metode six-step commutation

### 3.3.2 Skenario Pengujian Karakteristik Kecepatan Motor BLDC Terhadap Beban

Dalam skenario ini, pengujian dilakukan dengan mengubah-ubah tegangan pada potentiometer untuk menghasilkan perubahan duty cycle. Pengujian dilakukan dengan 3 keadaan, tanpa beban, beban variasi1, dan beban variasi 2. Untuk pembebanan motor disusun dengan motor lain, dan motor di atas diberi tekanan ke bawah untuk memberi variasi beban pada motor yang diuji pada bagian bawah seperti terlihat pada gambar



**Gambar 3.10** Skenario pengujian dengan beban

### 3.3.3 Skenario Rangkaian Rangkaian Penyesuai Tegangan

Dalam skenario ini, pengujian dilakukan menggunakan SPICE berdasarkan rangkaian yang telah dibuat sebelumnya. Pada bagian ini akan dilihat korelasi tegangan yang diberikan terhadap tegangan keluaran, serta melihat hubungan secara lengkap dengan masukan dan kecepatan motor.

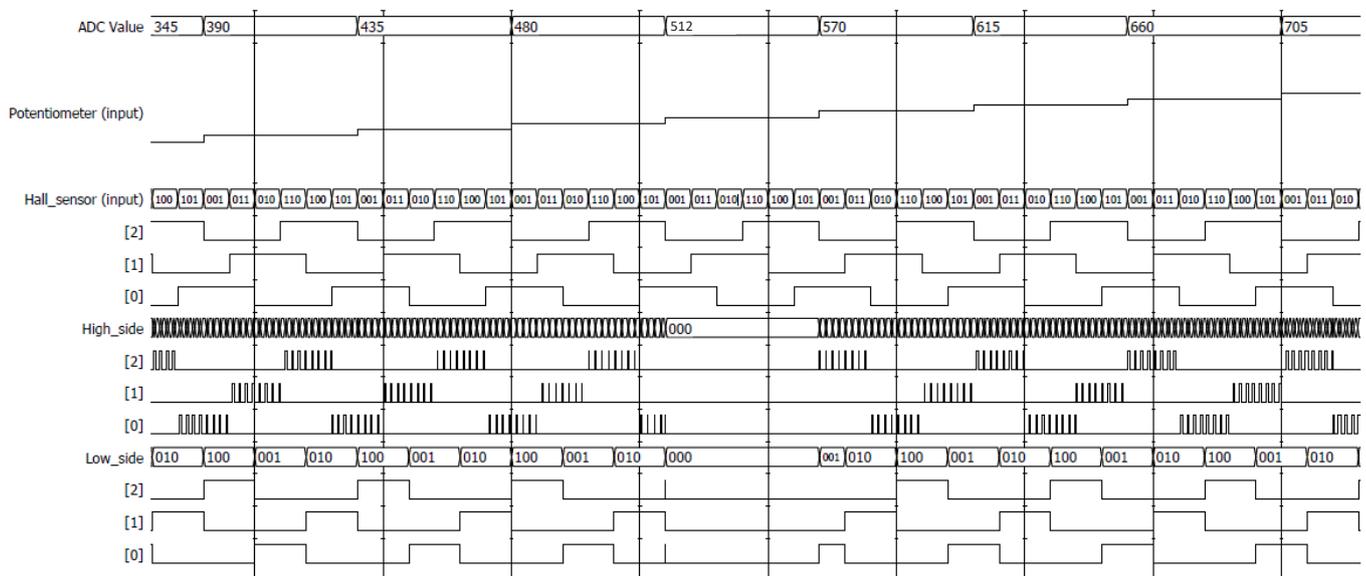
## BAB 4

### HASIL DAN PEMBAHASAN

#### 4.1 Pengujian Sistem Kendali motor BLDC

##### 4.1.1 Hasil Simulasi Verilog HDL

Berdasarkan algoritma yang telah dibuat dan dijelaskan pada bagian sebelumnya, dilakukan simulasi untuk memverifikasi kesesuaian logika yang telah dibuat. Simulasi dilakukan menggunakan software Modelsim-Altera. Untuk melakukan simulasi, terlebih dahulu dilakukan pembuatan testbench yang akan berguna sebagai kotak hitam yang menguji rangkaian yang dibuat pada gambar 3.3 dan menjadikan module control\_bldc sebagai *unit under test*.



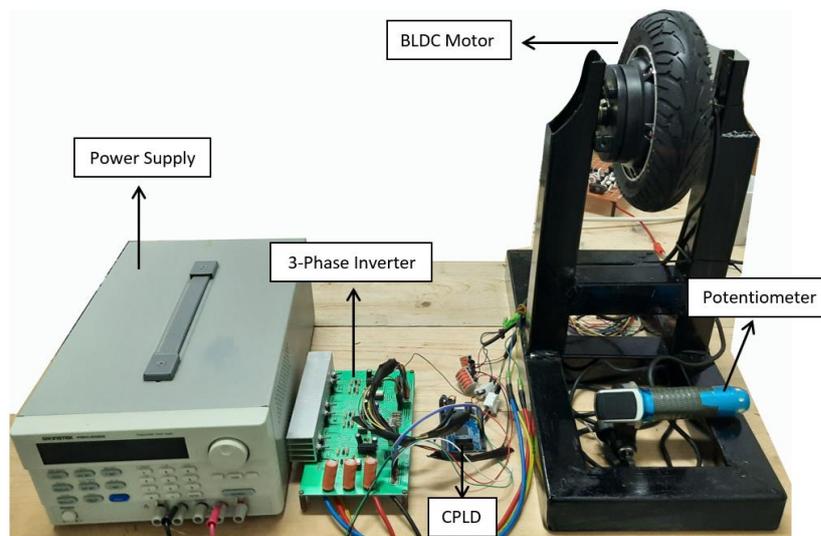
**Gambar 4.1** Hasil Simulasi pada Modelsim-Altera

Sebelum memulai simulasi terlebih dahulu diperlukan signal hall sensor serta potentiometer untuk dapat melakukan simulasi. Pada simulasi ini dilakukan perubahan nilai potentiometer serta hall sensor untuk melakukan pengujian terhadap keluaran yang dihasilkan. Selain dua sinyal tersebut juga dilakukan simulasi terhadap clock frekuensi yaitu sebesar 50 MHz sesuai dengan target perangkat yang akan digunakan.

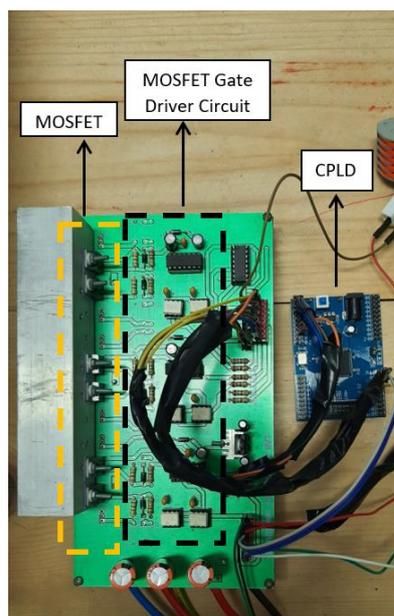
Gambar 4.1 menunjukkan hasil simulasi yang dilakukan. Dapat dilihat dari hasil simulasi tersebut bahwa perubahan pada hall sensor mengubah pembangkitan pwm yang aktif. Sementara itu nilai potentiometer menentukan perubahan dari duty cycle yang dibuat serta menentukan arah putaran motor. Simulasi menunjukkan urutan yang benar sesuai dengan tabel 3.1 dan tabel 3.2.

#### **4.1.2 Hasil Implementasi**

Pada hasil simulasi sebelumnya telah menunjukkan bahwa kode Verilog yang dibuat telah sesuai, sehingga selanjutnya dilakukan implementasi pada perangkat keras. Implementasi dilakukan pada CPLD Max II: EPM240T100C5 menggunakan JTAG melalui software Quartus. Hasil implementasi dari rangkaian logika pada CPLD tersebut menggunakan 133 dari 240 *Logic Elements*, sekitar 55% dari kemampuan EPM240, serta menggunakan 14 pin dari 80 pin input dan output perangkat.



(a) Experimental Setup

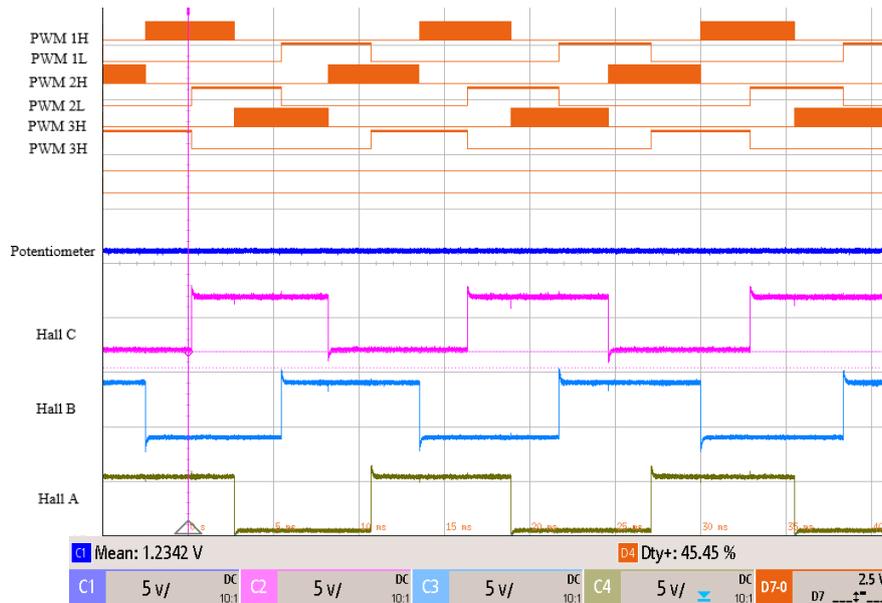


(b) 3-phase Inverter

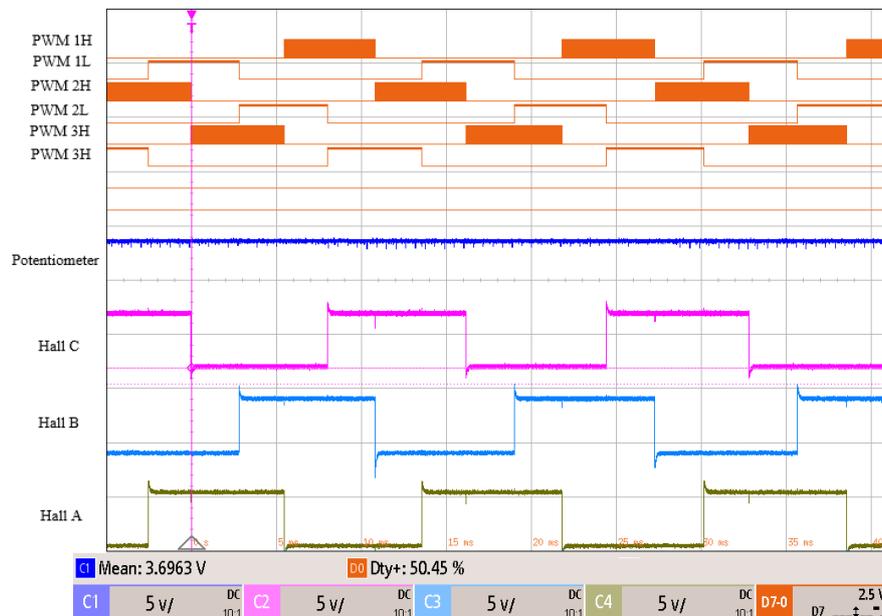
**Gambar 4.2** Perangkat Uji Coba

Pada gambar 4.2 menunjukkan perangkat secara nyata berdasarkan rancangan yang dibuat pada gambar 3.6. Perangkat terdiri dari inverter tiga fasa,

CPLD, potentiometer, serta power supply. Untuk motor BLDC yang diuji coba menggunakan motor BLDC dengan *rate* tegangan 36 V dan daya 350W. Hall sensor pada motor terhubung langsung ke CPLD setelah diberikan pull-up resistor.



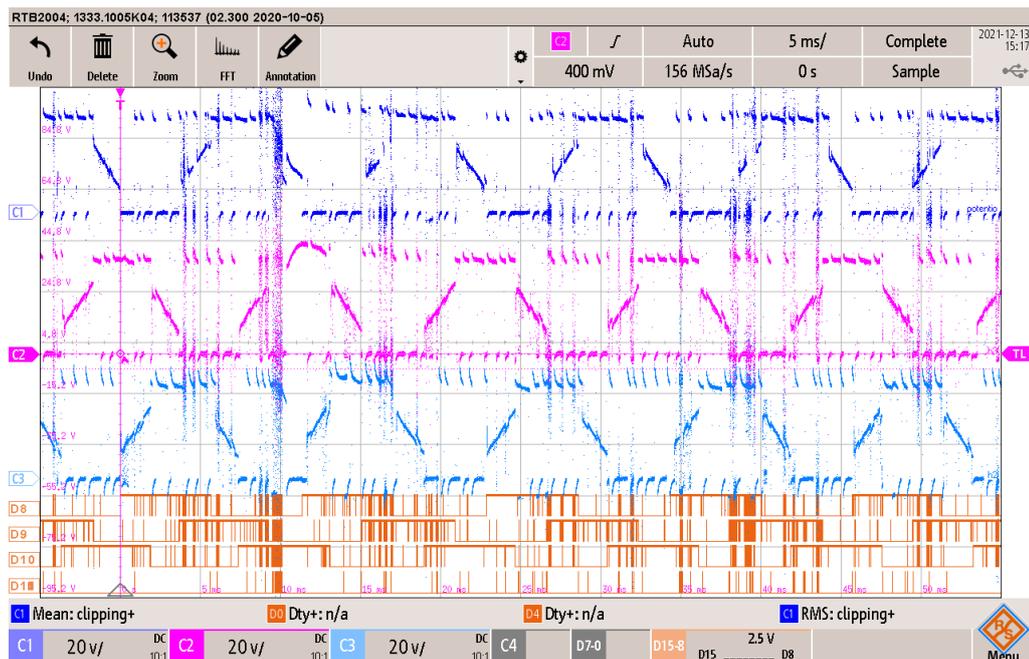
(a) Clockwise Direction



(b) Counterclockwise Direction

**Gambar 4.3** Sinyal kendali motor BLDC pada duty cycle 50%

Hasil secara eksperimental ditunjukkan pada gambar 4.3. Kondisi pengujian Ketika duty cycle 50% untuk kedua arah motor yang diuji. Dapat dilihat perubahan pola sesuai dengan simulasi dan tabel komutasi dari six-step commutation. Untuk gambar 4.4 dapat dilihat tegangan pada fasa Ketika duty cycle 100%.

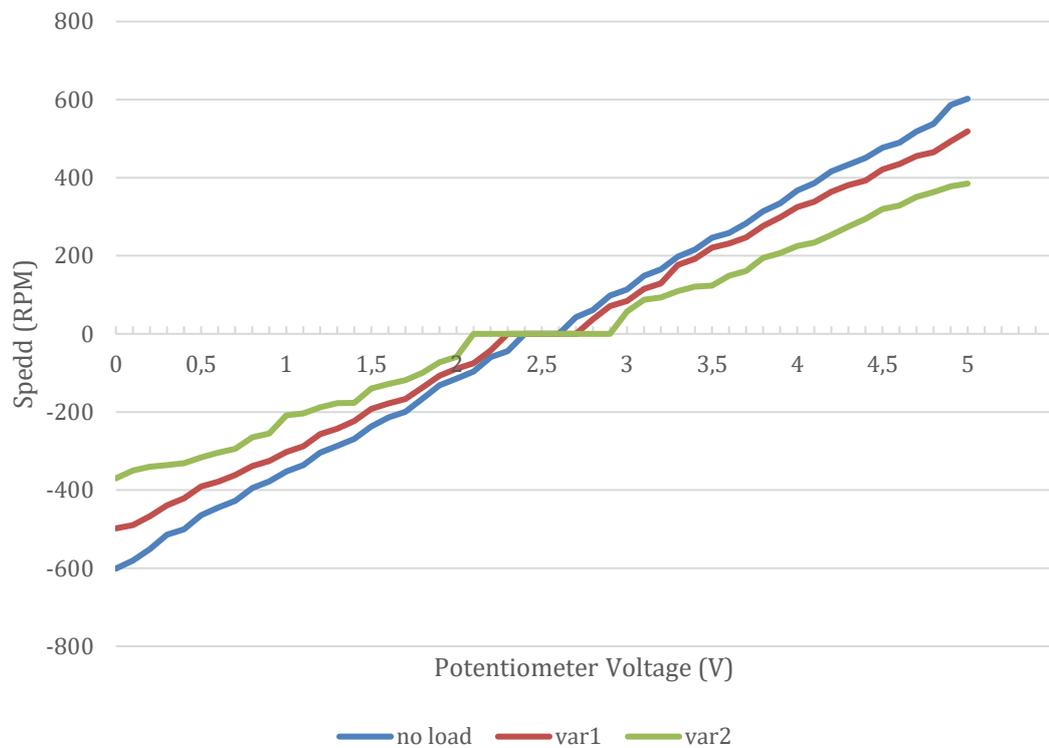


**Gambar 4.4** Sinyal pada Fasa Motor BLDC

## 4.2 Hasil Pengujian Karakteristik kecepatan Motor BLDC terhadap Perubahan Beban

Berdasarkan hasil yang telah didapat pada pengujian sebelumnya, dilakukan pengujian untuk melihat karakteristik motor BLDC terhadap variasi beban. Pengambilan data dilakukan menggunakan tachometer untuk melihat kecepatan motor serta mengukur tegangan masukan input ADC sebagai sumber tegangan referensi dalam menentukan arah serta duty cycle motor BLDC.

Seperti digambarkan pada gambar 3.7, pengukuran akan dilakukan dengan motor yang ditempelkan pada motor yang kita uji coba dengan mengatur tekanan pada motor untuk memberikan variasi beban. Dalam pengujian diambil 3 variasi data yaitu tanpa beban, dengan beban variasi 1, dan dengan beban variasi 2. Pengambilan data dilakukan dengan mengambil beberapa sample data berdasarkan tegangan potentiometer.



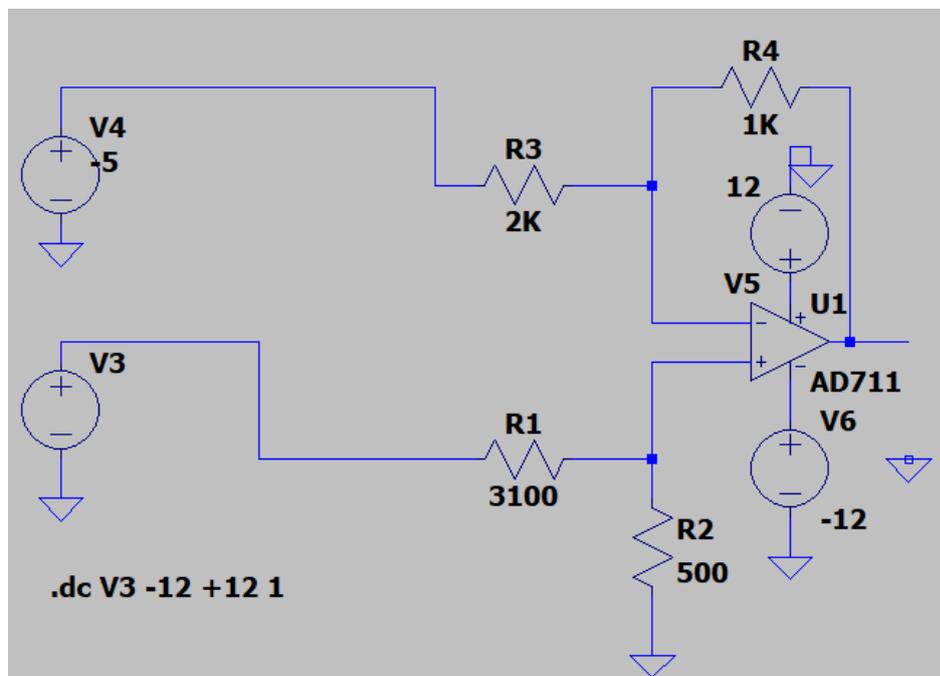
**Gambar 4.5** Grafik hubungan tegangan potentiometer terhadap kecepatan motor BLDC

Hasil pengujian yang dilakukan dapat dilihat pada gambar 4.5 di atas. Perubahan tegangan pada potentiometer menyebabkan perubahan duty cycle serta membuat tegangan pada motor berubah. Pada kondisi tanpa beban dapat dilihat

motor memiliki kecepatan yang lebih tinggi dengan kecepatan maksimum berada pada 600 RPM, sementara itu penambahan beban menyebabkan penurunan kecepatan dari motor BLDC, secara berturut-turut untuk beban 1 memiliki maksimum kecepatan sekitar 500 RPM dan untuk beban 2 memiliki kecepatan maksimum 370 RPM. Untuk arus power supply menunjukkan nilai secara berurut untuk tanpa beban, beban 1, dan beban 2 adalah 0,76A, 1,34A, dan 2,27A.

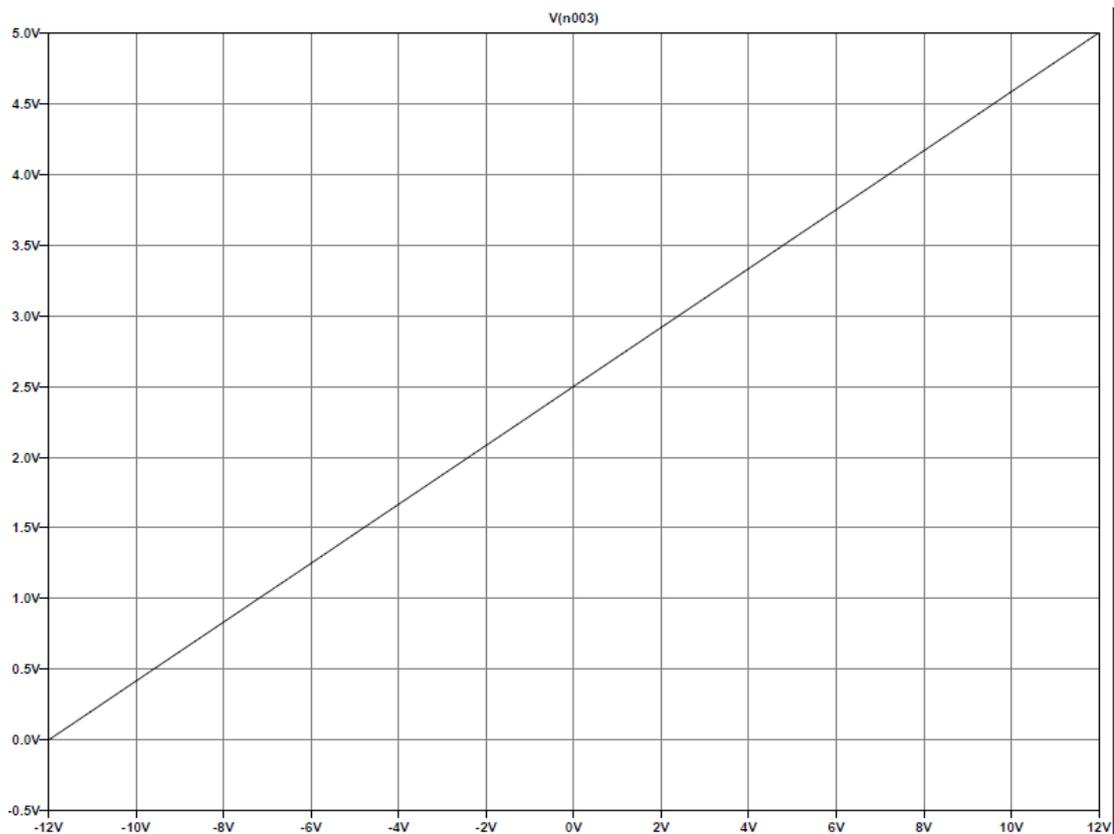
### 4.3 Hasil Pengujian Rangkaian Penyesuai Tegangan Secara Simulasi

Pada pengujian ini dilakukan pengujian menggunakan SPICE dengan rangkaian seperti pada gambar 4.6. Pengujian dilakukan dengan masukan tegangan dari -12 V hingga +12 V, sehingga dengan mengikuti persamaan 3.3 dan terlebih dahulu menentukan salah satu nilai resistor kita dapat mengetahui nilai komponen lainnya.



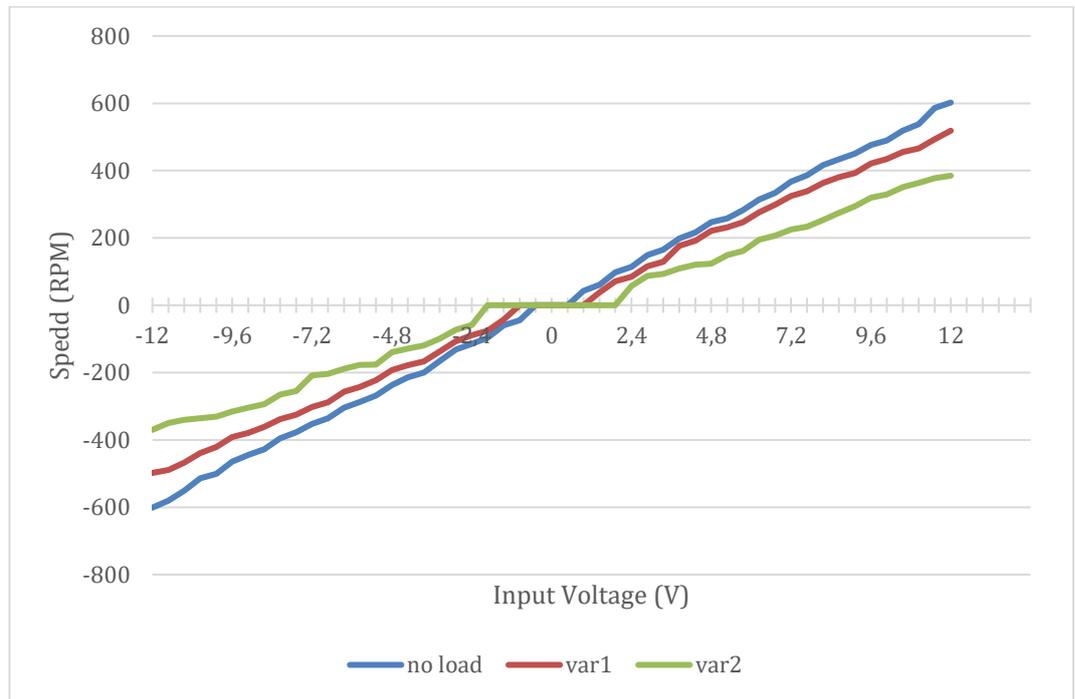
Gambar 4.6 Sinyal pada Fasa Motor BLDC

Simulasi dilakukan dengan melakukan DC sweep pada V3 sebagai tegangan masukan dan melakukan perubahan dari -12 V hingga +12V. Hubungan masukan dan keluaran dari simulasi dapat dilihat pada gambar 4.7. Simulasi menunjukkan hasil sesuai dengan karakteristik yang diinginkan dengan keluaran 0 – 5 Volt.



**Gambar 4.7** Sinyal keluaran rangkaian penyesuai tegangan

Menggunakan hasil ini kita dapat melihat hubungan tegangan masukan dengan dua polaritas terhadap kecepatan motor pada gambar 4.8. Sehingga terlihat karakteristik motor BLDC yang digunakan dengan masukan yang memiliki tegangan negatif dan tegangan positif untuk menentukan arah terhadap kecepatan motor BLDC.



**Gambar 4.8** Grafik hubungan tegangan dua polaritas terhadap kecepatan motor BLDC

## **BAB 5**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan seluruh proses implemmentasi dan pengujian yang telah dilakukan, maka dapat disimpulkan sebagai berikut:

1. Sistem kontrol motor BLDC menghasilkan sinyal PWM dengan frekuensi 20 KHz dan bekerja sesuai dengan pola komutasi six-step menggunakan hall sensor.
2. Sistem kontrol motor BLDC sebagai pengatur kecepatan diperoleh dengan mengubah duty cycles dari sinyal PWM.
3. Algoritma yang diimplementasikan pada CPLD Max II: EPM240T100C5 menggunakan 133 dari 240 Logic Elements (LE) dan 14 pin untuk masukan serta keluaran dari keseluruhan sistem.
4. Karakteristik motor BLDC yang diuji memiliki kecepatan maksimal masing-masing pada pengujian tanpa beban, dengan beban pertama, dan beban kedua yaitu 600 RPM, 500 RPM, dan 370 RPM untuk putaran searah jarum jam dan berlawanan jarum jam.

#### **5.2 Saran**

Beberapa saran yang diharapkan dapat menunjang pengembangan dan penyempurnaan riset terkait adalah sebagai berikut:

1. Menganalisis pengaruh *noise* pada kinerja sistem control motor

2. Menganalisis lebih dalam pengaruh beban yang diberikan terhadap motor BLDC

## DAFTAR PUSTAKA

- [1] J. de Santiago *et al.*, “Electrical Motor Drivelines in Commercial All-Electric Vehicles: A Review,” *IEEE Trans. Veh. Technol.*, vol. 61, no. 2, pp. 475–484, Feb. 2012, doi: 10.1109/TVT.2011.2177873.
- [2] A. Hughes, *Electric Motor and Drives*. Elsevier Ltd., 2006.
- [3] G. Tao, Z. Ma, L. Zhou, and L. Li, “A novel driving and control system for direct-wheel-driven electric vehicle,” *2004 12th Symp. Electromagn. Launch Technol.*, vol. 41, no. 1, pp. 514–517, 2004, doi: 10.1109/elt.2004.1398134.
- [4] M. Jafarboland and M. H. R. Silabi, “New sensorless commutation method for BLDC motors based on the line-to-line flux linkage theory,” *IET Electr. Power Appl.*, vol. 13, no. 6, pp. 703–711, 2019, doi: 10.1049/iet-epa.2018.5356.
- [5] M. Baszynski and S. Pirog, “A novel speed measurement method for a high-speed BLDC motor based on the signals from the rotor position sensor,” *IEEE Trans. Ind. Informatics*, vol. 10, no. 1, pp. 84–91, 2014, doi: 10.1109/TII.2013.2243740.
- [6] Hong-xing Wu, Shu-kang Cheng, and Shu-mei Cui, “A controller of brushless DC motor for electric vehicle,” *IEEE Trans. Magn.*, vol. 41, no. 1, pp. 509–513, Jan. 2005, doi: 10.1109/TMAG.2004.839304.
- [7] C.-L. Xia, *Permanent Magnet Brushless DC Motor Drives and Controls*. John Wiley & Sons Singapore Pte. Ltd., 2012.

- [8] S. J. Chapman, *Electric Machinery Fundamentals*, Fourth Edi. McGraw-Hill, 2005.
- [9] Tae-Hyung Kim and M. Ehsani, "Sensorless control of the BLDC motors from near-zero to high speeds," *IEEE Trans. Power Electron.*, vol. 19, no. 6, pp. 1635–1645, Nov. 2004, doi: 10.1109/TPEL.2004.836625.
- [10] K. Kolano, "Improved sensor control method for BLDC motors," *IEEE Access*, vol. 7, pp. 186158–186166, 2019, doi: 10.1109/ACCESS.2019.2960580.
- [11] B. P. Reddy and A. Murali, "SoC FPGA-based field oriented control of BLDC motor using low resolution Hall sensor," *IECON Proc. (Industrial Electron. Conf.)*, pp. 2941–2945, 2016, doi: 10.1109/IECON.2016.7793092.
- [12] R. Krishnan, *Electric Motor Drives - Modeling Analysis and Control*. Prentice Hall, Inc, 2001.
- [13] M. Jafarboland and M. H. R. Silabi, "New sensorless commutation method for BLDC motors based on the line-to-line flux linkage theory," *IET Electr. Power Appl.*, vol. 13, no. 6, pp. 757–765, 2019, doi: 10.1049/iet-epa.2018.5356.
- [14] N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics : Converters, Applications, and Design*, Second Edi. John Wiley & Sons, Inc, 1995.
- [15] M. H. Rashid, *The power electronics handbook*, Third edit. Elsevier Inc, 2011.

- [16] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic with Verilog Design*. McGraw-Hill, 2003.
- [17] C. Maxfield, *The Design Warrior's Guide to FPGAs*. Elsevier Inc, 2004.
- [18] M. Poorani and R. Kurunjimalar, "Design implementation of UART and SPI in single FGPA," in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, Jan. 2016, pp. 1–5, doi: 10.1109/ISCO.2016.7726983.
- [19] R. M. Pindoriya, A. K. Mishra, B. S. Rajpurohit, and R. Kumar, "FPGA Based Digital Control Technique for BLDC Motor Drive," *IEEE Power Energy Soc. Gen. Meet.*, vol. 2018-Augus, pp. 1–5, 2018, doi: 10.1109/PESGM.2018.8586472.
- [20] M. T. Inc., "Mcp3004/3008." pp. 1–35, 2002.
- [21] G. Mihalache and A. Ioan, "FPGA Implementation of BLDC Motor Driver with Hall Sensor Feedback," in *2018 International Conference and Exposition on Electrical And Power Engineering (EPE)*, Oct. 2018, pp. 624–629, doi: 10.1109/ICEPE.2018.8559886.

# LAMPIRAN

## Lampiran 1 Paper Seminar Hasil

# BLDC Motor Control using a Complex Programmable Logic Device with Hall-Sensors

Muhammad Fajri Sachruddin  
Department of Electrical Engineering  
Universitas Hasanuddin  
Makassar, Indonesia  
sachruddinmf16d@student.unhas.ac.id

Faizal Arya Samman  
Department of Electrical Engineering  
Universitas Hasanuddin  
Makassar, Indonesia  
faizalas@unhas.ac.id

Rhiza S. Sadjad  
Department of Electrical Engineering  
Universitas Hasanuddin  
Makassar, Indonesia  
rhiza@unhas.ac.id

**Abstract**—This paper presents the design and implementation of speed control for a brushless direct current (BLDC) motor using a complex programmable logic device (CPLD). Implementation of speed control is using a PWM technique by varying duty cycles applied to a three-phase inverter. Rotor position determines by hall sensors which are used as references to synchronize the PWM control signals. The control model is written using Verilog Hardware Description Language (HDL) and verified by simulation using ModelSim-Altera. An experimental setup is built to test the performance of the BLDC Motor under the PWM control. The control algorithm is implemented using Max II EPM240T100C5 devices on a 350W 36V rated BLDC motor. The number of the used logic elements (LEs) of the CPLD is about 133 of 240 LEs. PWM with controllable duty cycle generated in this system for having working frequency about 20KHz and commutate sequentially according to six-step commutation up to 600 RPM on no-load condition.

**Keywords**—CPLD, PWM, BLDC, Hall Sensor, ADC

## I. INTRODUCTION

The development and usage of electric vehicles (EVs) are becoming more popular and replacing fossil energy [1]. There are several obstacles to implement EV, one of it is a complicated control system [2], and therefore many publications and implementation encounter this problem. An electric motor is one important part of the electric vehicle, and many various motor types are currently on the market.

Brushless DC motor is widely used on many applications like computers, military, automotive, and many more [3]. In EV, this motor has become a good option compared to other motors. Several advantages of BLDC motor are high efficiency, convenient maintenance, and high torque density [2][4][5][6]. All these basic characteristics are required for traction application, such as in EV [7]. However, an electronic commutator is required for the BLDC motor instead of a mechanical commutator like in a DC motor.

In order to commutate BLDC motors, it is required to identify rotor position [4]. There are two ways to get the rotor position, i.e., using sensor and sensorless methods. Sensor method typically uses hall sensor to detect electromagnetic flux from a permanent magnet. The sensorless method detects it using zero-crossing back-EMF. Both methods have advantages and disadvantages. Hall sensors are much easier to use, but production costs can be higher and have low resolution. However, there are several methods to overcome this problem [8][9]. The sensorless method can be universally used on various BLDC motors, but it is rather complex to implement the algorithm [4][10].

The control system will be implemented in a complex programmable logic device (CPLD). Compared to microcontrollers, it offers several advantages due to flexibility and low power consumption [9] [11]. A FPGA and CPLD has a higher processing speed and can perform complex computations and parallel processes [11] [12]. Furthermore, CPLD and FPGA are programmed using a hardware description language and potentially implemented in system-on-chip devices for mass production. FPGA and CPLD also can easily simulate the design before being implemented on hardware to reduce error during implementation.

This paper presents a BLDC control system based on CPLD as the main controller to generate PWM signals. The commutation sequence is determined by using built-in hall-effect sensor to detect permanent magnet rotor position. Compared to other works [9] [12] [13][13], the implementation of CPLD offers more benefits than microcontroller or FPGA. CPLD process speed is better than microcontrollers for such parallel processing. With the minimal algorithm in this design, the use case of FPGA will overrate resources. Furthermore, since CPLD is a non-volatile device, there is no need for programming and configuring when initial power-up of the system frequently occurs.

The remaining chapter of this paper is organized as follows. Chapter II describes in detail about control algorithm using for this BLDC motor drive system. Chapter III explains simulation results, and Chapter IV is implementation and verification of BLDC control system through experimental testing to BLDC motor.

## II. BLDC MOTOR CONTROL SYSTEM

The construction of a brushless DC motor is similar to an induction motor and a DC motor without brushes. It can have various configurations on total phase or winding arrangement, either star or delta pattern. In this paper, we will focus on control the three-phase BLDC motor.

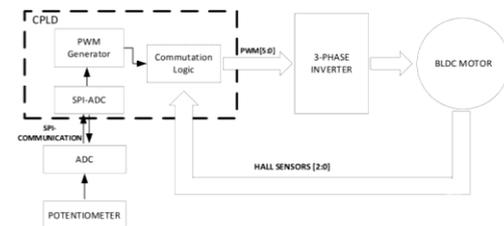


Fig. 1. BLDC motor control system

Commutation of BLDC motor is controlled electronically, based on rotor-stator position. The speed of BLDC motor is proportional to the applied voltage, and therefore we are required to control voltage on BLDC motor [14]. Pulse width modulation technique will be used to control voltage by varying the duty cycle of PWM signals.

The overall system of this motor control system is presented in Figure 1. There are two inputs in our system, the signal from hall sensors and potentiometer. The built-in hall sensors are used to determine rotor position of the motor and potentiometer to adjust speed of BLDC motor. Logic part will be controlled using complex programmable logic device (CPLD) as the brain of our system.

#### A. Phase Sequence Based on Hall Sensor

Figure 2 shows a three-phase inverter using six MOSFETs as switch with PWM signal to gate of MOSFET. This will be VSI to drive BLDC motor and using two-phase conduction method [15].

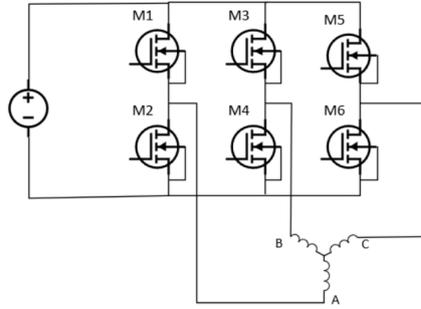


Fig. 2. 3-Phase Inverter Schematic

The commutation of BLDC motor is determined by hall sensor pattern and called six-step commutation. Relation between active phase and rotor position can be seen on manufacturer datasheet or manually determined [13]. Table I and Table II shows sequence for six-step commutation in both direction clockwise and counterclockwise direction. Every MOSFET is commutate every 60 electrical degrees.

TABLE I. CLOCKWISE SIX-STEP COMMUTATION

Hall Sensors			Motor Phases		
A	B	C	Phase A	Phase B	Phase C
0	0	1	Off	-	+
0	1	1	-	Off	+
0	1	0	-	+	Off
1	1	0	Off	+	-
1	0	0	+	Off	-
1	0	1	+	-	Off

TABLE II. COUNTERCLOCKWISE SIX-STEP COMMUTATION

Hall Sensors			Motor Phases		
A	B	C	Phase A	Phase B	Phase C
0	0	1	Off	+	-
0	1	1	+	Off	-
0	1	0	+	-	Off
1	1	0	Off	-	+
1	0	0	-	Off	+
1	0	1	-	+	Off

#### B. Interfacing ADC for Potentiometer

The output of potentiometer is analog voltage signal and cannot directly use on CPLD. To process the signal digitally, it must be converted to digital form using ADC chip. MCP3008 ADC chip is used in this case. It is 10-bit ADC with 8 channels and using SPI protocol-based to communicate to control unit.

SPI-protocol is communication for short-distance using serial interface. To initiating communication with ADC and CPLD as master device by send low signal to CS line. Figure 3 shows complete communication through MCP3008 ADC[16].

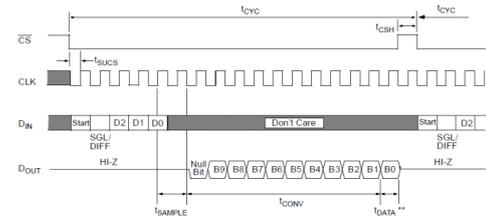


Fig. 3. Communication of MCP3008

Module to use this ADC is written in Verilog HDL and performed like FSM based. The data is sending through MOSI and MISO lines. Theoretical digital output of ADC is determined using equation as shown below.

$$\text{Digital output code} = \frac{1024 \times V_{in}}{V_{ref}} \quad (1)$$

#### C. The Generated PWM signals

Figure 4 shows a flowchart for speed control of BLDC. Digital value of potentiometer is used to determine direction of rotation and duty cycle on our PWM signal on high side of the MOSFET. Using counter and compare it to value of ADC input as a reference.

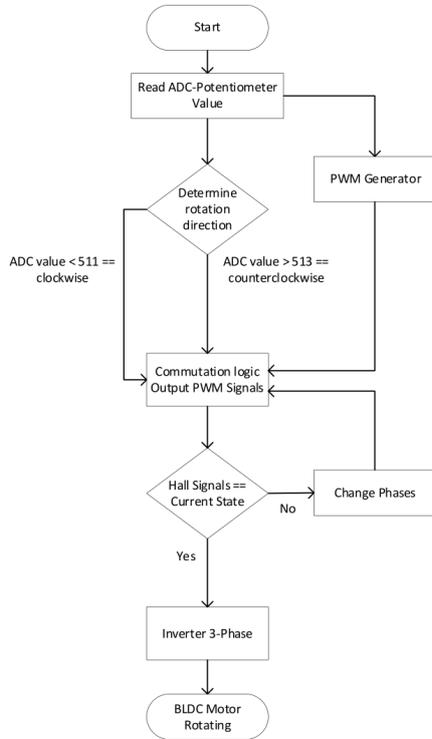


Fig. 4. Flowchart of the BLDC Control System.

### III. SIMULATION RESULTS

The control algorithm of BLDC motor drive is simulate using Modelsim-Altera. Testbench was developed using Verilog HDL and our control\_bldc module as a unit under test. This simulation aims to validate PWM signals generated in our design before being implemented to hardware.

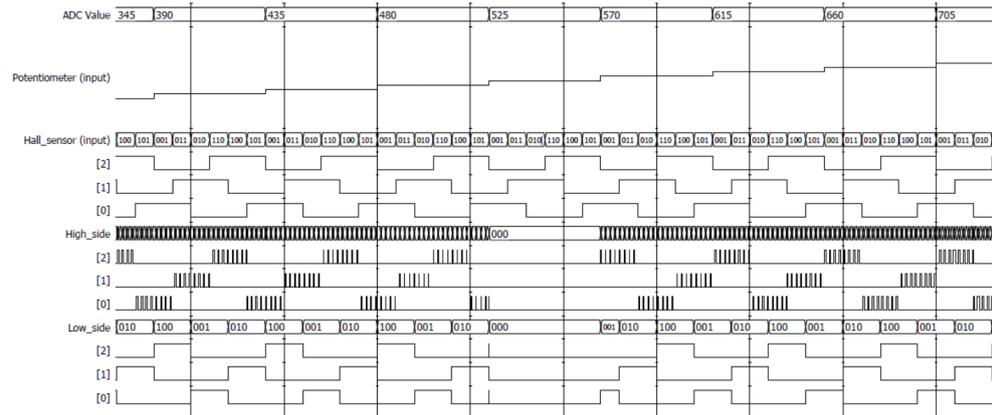


Fig. 5. Modelsim-Altera Simulation.

The signals input from hall sensors and potentiometer are required to simulate the testbench. It simulates the changing potentiometer value reference and hall effects to examine output of the control motor. The clock frequency for this system is 50 MHz based on targeted device that will be implemented.

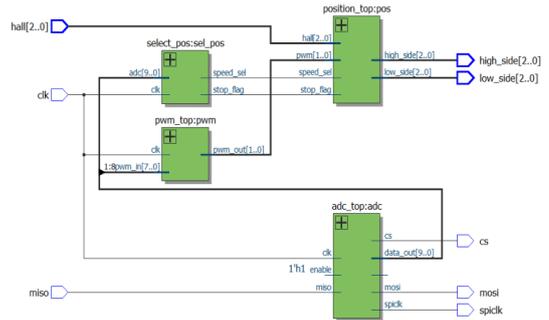
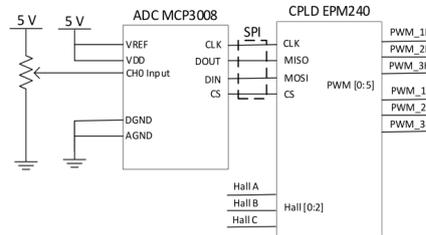


Fig. 6. RTL Viewer of Verilog HDL Design.

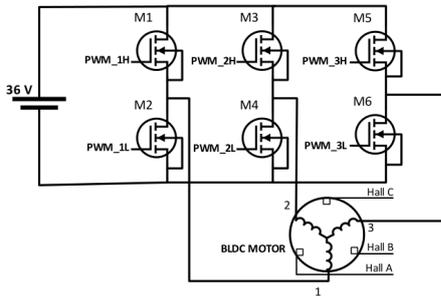
The simulation result for control algorithm is shown in Figure 5. PWM in this simulation is outputting duty cycle according to increasing value of ADC to the high side signals. By simulate 3-bit signals to hall input and potentiometer value, we can see correct commutation sequence for each phase according to Table I and Table II.

### IV. IMPLEMENTATION AND EXPERIMENTAL RESULT

The simulation results verified the control algorithm according to the proposed system. Logic control was then implemented to CPLD Max II: EPM240T100C5 using JTAG via Quartus software. Figure 6 shows the logic schematic implemented to the CPLD. Total logic elements used in this design are 133 of 240 logic elements (LEs), utilize around 55% of EPM240 capability, and take 14 pins of 80 for input and output.



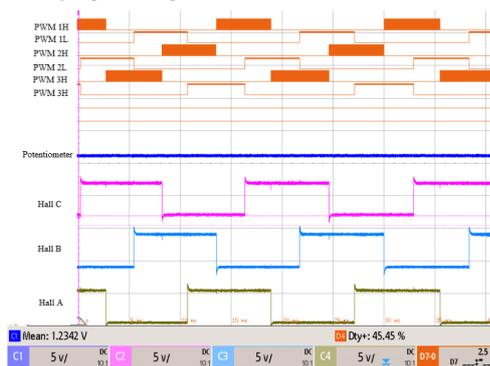
(a) Input and output connection



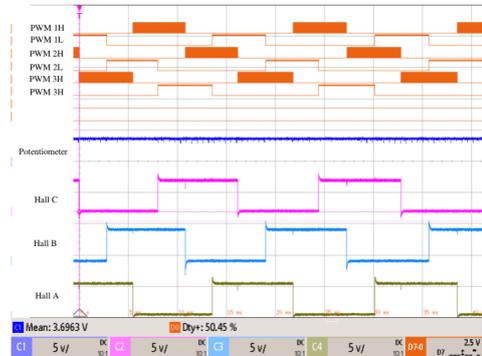
(b) 3-phase Inverter

Fig. 7. Hardware Implementation Setup.

The experimental setup is shown in Figure 7. BLDC motor used is rated for 36V and 350W, and six MOSFET is IRF3808. Hall built-in sensor on BLDC motor directly connected to CPLD through a pull-up resistor. Input signal from hall sensor is 5 volts for high logic signal and 0 volts for low logic signal. CPLD EPM240 runs at 3.3 V logic, and therefore it is needed a MOSFET driver to amplify the signal to the gate of MOSFETs. The result in Figure 8 shows the correlation between hall sensor, potentiometer value, and phase sequence. The correlation shows both commutation clockwise and counterclockwise direction at 50% duty cycle. Each high side output variable duty cycle according to potentiometer, and for low side, it changes whenever hall sensor jumps to next pattern.



(a) Clockwise Direction



(b) Counterclockwise Direction

Fig. 8. BLDC control motor signals on 50% duty cycle (experimental result).

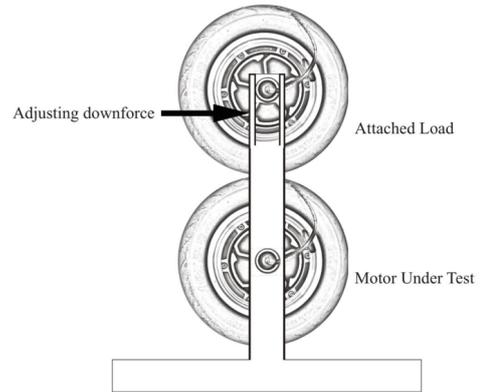


Fig. 9. Load testing experimental setup.

The testing of BLDC motor established in a condition without load and two variations with different loads. The load testing using other motor attach on top of the tested motor. Figure 10 shows speed measured of the BLDC motor during three loads variation on both directions. The changing speed of BLDC motor based on changes value of potentiometer voltage as PWM set reference. Maximum speed achieved on no-load condition is 600 RPM, variation 1 load is 500RPM, and variation 2 load is 370 RPM. The current for no-load, variation 1 and variation 2 are around 0,76A, 1,34A, and 2,27A.

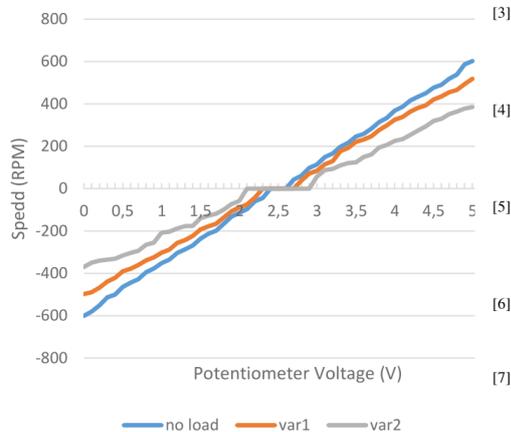


Fig. 10. Speed results on clockwise and counterclockwise direction.

#### V. CONCLUSION

In this paper, the control system of the BLDC motor using a complex programmable logic device (CPLD) is presented. The control system generates PWM signals according to six-step commutation using hall sensors. BLDC motor speed control is achieved by varying duty cycles of PWM with a frequency 20KHz. Duty cycle controlled with comparing ADC value from a potentiometer and compare with an internal counter. The control system was created with Verilog HDL and simulated using Modelsim-Altera. The algorithm implemented to CPLD Max II: EPM240T100C5 with total logic elements (LEs) is about 133 out of 240, around 55% of EPM240 capability, and takes 14 pins for input and output for the overall system.

#### ACKNOWLEDGMENT

The authors gratefully acknowledge Ministry of Research and Technology of the Republic of Indonesia, Board of National Research and Innovation for supporting the research under the scheme of Applied Research Grant with Grant Contract Number 7/E1/KP.PTNBH/2021.

#### REFERENCES

[1] J. De Santiago *et al.*, "Electrical motor drivelines in commercial all-electric vehicles: A review," *IEEE Trans. Veh. Technol.*, vol. 61, no. 2, pp. 475–484, 2012, doi: 10.1109/TVT.2011.2177873.

[2] G. Tao, Z. Ma, L. Zhou, and L. Li, "A novel driving and control system for direct-wheel-driven electric vehicle," *2004 12th Symp. Electromagn. Launch Technol.*, vol. 41, no. 1, pp. 514–517, 2004, doi: 10.1109/elt.2004.1398134.

[3] Tae-Hyung Kim and M. Ehsani, "Sensorless control of the BLDC motors from near-zero to high speeds," *IEEE Trans. Power Electron.*, vol. 19, no. 6, pp. 1635–1645, Nov. 2004, doi: 10.1109/TPEL.2004.836625.

[4] M. Jafarboland and M. H. R. Silabi, "New sensorless commutation method for BLDC motors based on the line-to-line flux linkage theory," *IET Electr. Power Appl.*, vol. 13, no. 6, pp. 757–765, 2019, doi: 10.1049/iet-epa.2018.5356.

[5] M. Baszynski and S. Pirog, "A novel speed measurement method for a high-speed BLDC motor based on the signals from the rotor position sensor," *IEEE Trans. Ind. Informatics*, vol. 10, no. 1, pp. 84–91, 2014, doi: 10.1109/TII.2013.2243740.

[6] H. X. Wu, S. K. Cheng, and S. M. Cui, "A controller of brushless DC Motor for electric vehicle," *2004 12th Symp. Electromagn. Launch Technol.*, vol. 41, no. 1, pp. 528–533, 2004.

[7] Z. Q. Zhu and D. Howe, "Electrical machines and drives for electric, hybrid, and fuel cell vehicles," *Proc. IEEE*, vol. 95, no. 4, pp. 746–765, 2007, doi: 10.1109/JPROC.2006.892482.

[8] K. Kolano, "Improved Sensor Control Method for BLDC Motors," *IEEE Access*, vol. 7, pp. 186158–186166, 2019, doi: 10.1109/ACCESS.2019.2960580.

[9] B. P. Reddy and A. Murali, "SoC FPGA-based field oriented control of BLDC motor using low resolution Hall sensor," *IECON Proc. (Industrial Electron. Conf.)*, pp. 2941–2945, 2016, doi: 10.1109/IECON.2016.7793092.

[10] T. H. Kim and M. Ehsani, "Sensorless control of the BLDC motors from near-zero to high speeds," *IEEE Trans. Power Electron.*, vol. 19, no. 6, pp. 1635–1645, 2004, doi: 10.1109/TPEL.2004.836625.

[11] J. Cervantes, E. Cordova, A. I. S. Marrufo, I. U. P. Monarrez, and M. Nandayapa, "BLDC motor commutation based on DSP builder for FPGA," *Int. Power Electron. Congr. - CIEP*, vol. 2016-Augus, pp. 166–171, 2016, doi: 10.1109/CIEP.2016.7530750.

[12] A. Tashakori, M. Hassanudeen, and M. Ektesabi, "FPGA based controller drive of BLDC motor using digital PWM technique," in *2015 IEEE 11th International Conference on Power Electronics and Drive Systems*, Jun. 2015, pp. 658–662, doi: 10.1109/PEDS.2015.7203584.

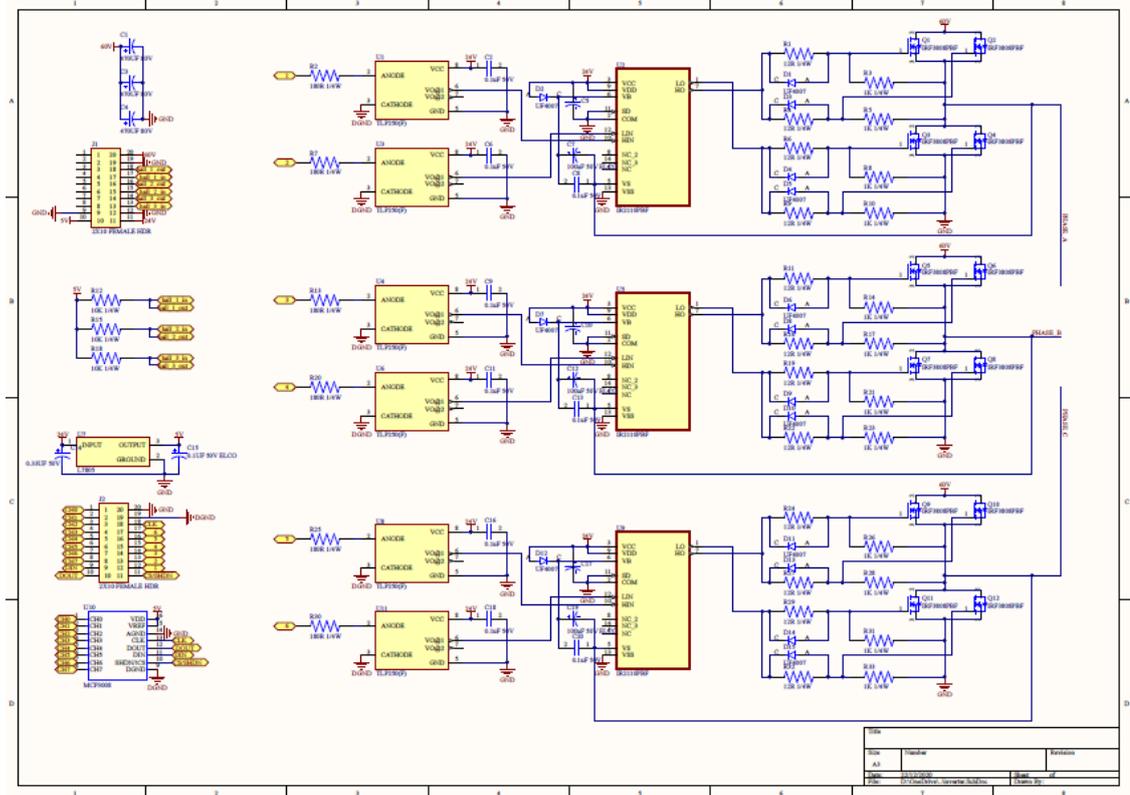
[13] G. Mihalache and A. D. Ioan, "FPGA Implementation of BLDC Motor Driver with Hall Sensor Feedback," *EPE 2018 - Proc. 2018 10th Int. Conf. Expo. Electr. Power Eng.*, pp. 624–629, 2018, doi: 10.1109/ICEPE.2018.8559886.

[14] R. M. Pindoriya, A. K. Mishra, B. S. Rajpurohit, and R. Kumar, "FPGA Based Digital Control Technique for BLDC Motor Drive," *IEEE Power Energy Soc. Gen. Meet.*, vol. 2018-Augus, pp. 1–5, 2018, doi: 10.1109/PESGM.2018.8586472.

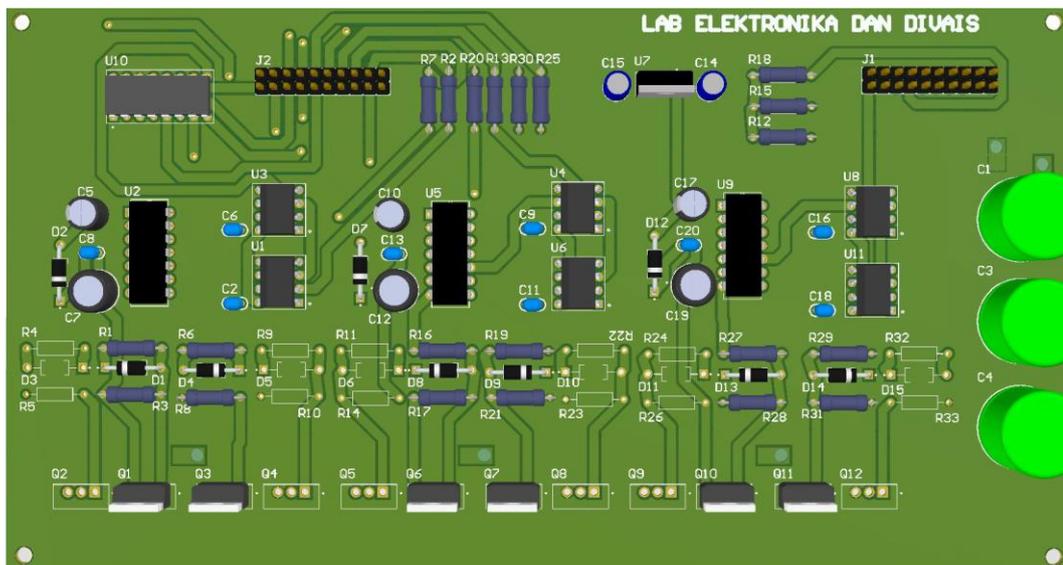
[15] M. H. Rashid, *The power electronics handbook*, Third edit. Elsevier Inc, 2011.

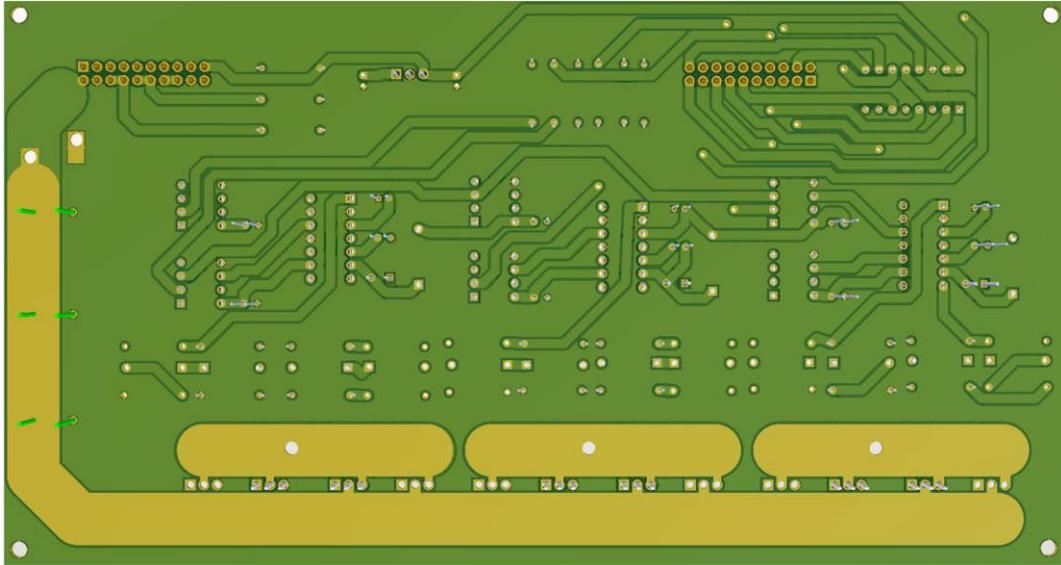
[16] M. T. Inc., "Mcp3004/3008." pp. 1–35, 2002.

## Lampiran 2 Skematik Rangkaian Inverter 3 Fasa



## Lampiran 3 Layout PCB Inverter Tiga Fasa





Lampiran 4 Netlist LTSpice Rangkaian Penyesuai Tegangan

```
R1 N006 N005 3100 tol=1 pwr=0.1
R2 N006 0 500 tol=1 pwr=0.1
R3 N002 N001 2K tol=1 pwr=0.1
R4 N003 N002 1K tol=1 pwr=0.1
V3 N005 0
V5 N004 0 12
V6 N007 0 -12
V4 N001 0 -5
XU1 N006 N002 N004 N007 N003 AD712
.dc V3 -12 +12 1
.lib ADI1.lib
.backanno
.end
```

## Lampiran 5 Kode Verilog HDL SPI-MCP3008 ADC

```
`timescale 1ns / 1ps

module adc_top(clk,enable,mosi,miso,spiclk,cs,done,data_out);

input clk,enable,miso;
output cs, done, mosi,spiclk;
output [9:0] data_out;

wire [9:0] data;

adc_spi spi(spiclk, enable, 4'b1000, cs, done, miso, mosi, data);
clk_div a1(clk, 1'b0, spiclk);
fdcl0 a0(data,spiclk,done,1'b0,data_out);

endmodule
```

```

`timescale 1ns / 1ps

module adc_spi(clk, enable, control_bit, cs, done, miso, mosi, data);

    input clk, enable, miso;
    input [3:0] control_bit;
    output reg cs, done, mosi;
    output reg [9:0] data;

    parameter N = 4;
    |
    reg [2:0] state = 0;
    reg [4:0] count = 0;
    reg [2:0] bit_posout = 3;
    reg [4:0] bit_pos = 0;

    always @(negedge clk) begin
        if (enable == 1) begin
            case (state)
                0: begin
                    cs <= 0;
                    done <= 0;
                    mosi = 1;
                    state <= state + 3'b001;
                    count <= count + 5'b00001;
                    data <= 0;
                end
                1: begin
                    if (count < 4 ) begin
                        cs <= 0;
                        done <= 0;
                        mosi = control_bit[bit_posout];
                        bit_posout = bit_posout - 3'b001;
                        count <= count + 5'b00001; //4
                        state <= 1;
                    end
                end
            endcase
        end
    end
end

```

```

else begin
    cs <= 0;
    done <= 0;
    mosi = control_bit[bit_posout];
    bit_posout = bit_posout - 3'b001;
    count <= count + 5'b00001; //5
    state <= state + 3'b001;
end
end
2: begin
    if (count < 8) begin
        cs <= 0;
        done <= 0;
        mosi = 0;
        state <= 2;
        count <= count + 5'b00001; end //6
    else begin
        cs <= 0;
        done <= 0;
        mosi = 0;
        state <= 3;
        bit_pos = 5'b10001 - count;
        data[bit_pos] <= miso;
        count <= count + 5'b00001;
    end
end
3: begin
    if (count < 17) begin
        cs <= 0;
        done <= 0;
        mosi = 0;
        state <= 3;
        bit_pos = 5'b10001 - count;
        data[bit_pos] <= miso;
        count <= count + 5'b00001;end
    else begin

```

```

        cs    <= 0;
        done  <= 1;
        mosi  = 0;
        state <= state + 3'b001;
        bit_pos = 5'b10001 - count;
        data[bit_pos] <= miso;
        count <= count + 5'b00001;
    end
end
4:    begin
    if (count == 18) begin
        cs    <= 1;
        done  <= 0;
        mosi  = 0;
        state <= 0;
        count <= 0;
        bit_posout = 3;
    end
end

    default: begin
        cs <= 0;
        state <= 0;
        count <= 0;
        done <= 0;
    end
endcase
end
else begin
//reset
    cs <= 0;
    state <= 0;
    count <= 0;
    done <= 0;
    data <= 0;
end
end
endmodule

```

---

```
module clk_div
#(
parameter WIDTH = 8, // Width of the register required
parameter N = 250// 6 We will divide by 12 for example in this case
)
(clk,reset, clk_out);

input clk;
input reset;
output clk_out;

reg [WIDTH-1:0] r_reg = 1'b0;
wire [WIDTH-1:0] r_nxt;
reg clk_track = 1'b0;

always @(negedge clk)

begin
if (reset)
begin
r_reg <= 0;
clk_track <= 1'b0;
end

else if (r_nxt == N)
begin
r_reg <= 0;
clk_track <= ~clk_track;
end

else
r_reg <= r_nxt;
end

assign r_nxt = r_reg+1'b1;
assign clk_out = clk_track;
endmodule
```

```
`timescale 1ns / 1ps

module fdc10(a,clk,en,reset,y);
    input [9:0] a;
    input clk,en,reset;
    output [9:0] y;
    fdc1 d1(y[0],clk,en,reset,a[0]);
    fdc1 d2(y[1],clk,en,reset,a[1]);
    fdc1 d3(y[2],clk,en,reset,a[2]);
    fdc1 d4(y[3],clk,en,reset,a[3]);
    fdc1 d5(y[4],clk,en,reset,a[4]);
    fdc1 d6(y[5],clk,en,reset,a[5]);
    fdc1 d7(y[6],clk,en,reset,a[6]);
    fdc1 d8(y[7],clk,en,reset,a[7]);
    fdc1 d9(y[8],clk,en,reset,a[8]);
    fdc1 d10(y[9],clk,en,reset,a[9]);

endmodule
```

---

```
`timescale 1ns / 1ps

module fdc1(q,clk,ce,reset,d);
    input d,clk,ce,reset;
    output reg q;
    initial begin q=0; end
    always @ (negedge (clk)) begin
    if (reset)
        q <= 1'b0;
    else if (ce)
        q <= d;
    else
        q<= q ;
    end
endmodule
```

## Lampiran 6 Kode Verilog HDL PWM Generator

```
`timescale 1ns / 1ps

module pwm_top(clk, pwm_in, pwm_out);

input clk;
input [7:0] pwm_in;
output [1:0] pwm_out;

wire clk_out;
reg[7:0] counter_PWM=5'd0;

clk_div clk0(clk,1'b0, clk_out);
    defparam clk0.N = 5;

always @(posedge clk_out)
begin
    begin
        counter_PWM <= counter_PWM + 8'b00000001;
        if(counter_PWM>=255)
            counter_PWM <= 0;
        end
    end
end

assign pwm_out[0] = counter_PWM <= pwm_in ? 1'b1:1'b0;
assign pwm_out[1] = counter_PWM >= pwm_in ?1'b1:1'b0;

/*
pwm_high pwm0(clk_div, data, pwm_tmp0);
pwm_low pwm1(clk_div, data, pwm_tmp1);

fd al(clk_div, 1'b0, 1'b1, pwm_in, data);
clock_div prescaler(clk, clk_div);

assign pwm_out = (speed_sel == 1'b1) ? pwm_tmp0:pwm_tmp1;
*/

endmodule
```

---

```

module clk_div
#(
parameter WIDTH = 8, // Width of the register required
parameter N = 250// 6 We will divide by 12 for example in this case
)
(clk,reset, clk_out);

input clk;
input reset;
output clk_out;

reg [WIDTH-1:0] r_reg = 1'b0;
wire [WIDTH-1:0] r_nxt;
reg clk_track = 1'b0;

always @(negedge clk)

begin
if (reset)
begin
r_reg <= 0;
clk_track <= 1'b0;
end

else if (r_nxt == N)
begin
r_reg <= 0;
clk_track <= ~clk_track;
end

else
r_reg <= r_nxt;
end

assign r_nxt = r_reg+1'b1;
assign clk_out = clk_track;
endmodule

```

## Lampiran 7 Kode Verilog HDL Logika Komutasi

```
`timescale 1ns / 1ps

module position_top(speed_sel, stop_flag, pwm, hall, high_side,low_side);

input speed_sel, stop_flag;
input [1:0]pwm;
input [2:0] hall;
output reg [2:0] high_side,low_side;

//wire [2:0] hall_dff;
//wire speed_sel_dff;

//fdcl0 dff0(hall,clk,1'b1,1'b0,hall_dff);
//fdcel dff1(speed_sel_dff,clk,1'b1,1'b0,speed_sel);

always@(speed_sel or hall or pwm or stop_flag)
  if (stop_flag)
    begin
      high_side <= 3'b000;
      low_side <= 3'b000;
    end
  else
    begin
      case({speed_sel,hall})//CBA, speed_sel 0 = cw
        4'b0_001 : begin //0 = A
          high_side[0] <= pwm[1];
          high_side[1] <= 1'b0;
          high_side[2] <= 1'b0;
          low_side <= 3'b100; /* complimentary
          low_side[0] <= 1'b0;
          low_side[1] <= 1'b0;
          low_side[2] <= pwm_out; */
        end
        4'b0_011 : begin
          high_side[0] <= 1'b0;
          high_side[1] <= pwm[1];
          high_side[2] <= 1'b0;
          low_side <= 3'b100; /*
          low_side[0] <= 1'b0;

```

---

```
        low_side[1] <= 1'b0;
        low_side[2] <= pwm_out; */
    end
4'b0_010 : begin
    high_side[0] <= 1'b0;
    high_side[1] <= pwm[1];
    high_side[2] <= 1'b0;
    low_side <= 3'b001; /*
    low_side[0] <= pwm_out;
    low_side[1] <= 1'b0;
    low_side[2] <= 1'b0; */
    end
4'b0_110 : begin
    high_side[0] <= 1'b0;
    high_side[1] <= 1'b0;
    high_side[2] <= pwm[1];
    low_side <= 3'b001; /*
    low_side[0] <= pwm_out;
    low_side[1] <= 1'b0;
    low_side[2] <= 1'b0; */
    end
4'b0_100 : begin
    high_side[0] <= 1'b0;
    high_side[1] <= 1'b0;
    high_side[2] <= pwm[1];
    low_side <= 3'b010; /*
    low_side[0] <= 1'b0;
    low_side[1] <= pwm_out;
    low_side[2] <= 1'b0; */
    end
4'b0_101 : begin
    high_side[0] <= pwm[1];
    high_side[1] <= 1'b0;
    high_side[2] <= 1'b0;
    low_side <= 3'b010; /*
    low_side[0] <= 1'b0;
    low_side[1] <= pwm_out;
    low_side[2] <= 1'b0; */
    end
end
```

---

```

4'b1_001 : begin //0 = A
    high_side[0] <= 1'b0;
    high_side[1] <= 1'b0;
    high_side[2] <= pwm[0];
    low_side <= 3'b001;
end
4'b1_011 : begin
    high_side[0] <= 1'b0;
    high_side[1] <= 1'b0;
    high_side[2] <= pwm[0];
    low_side <= 3'b010;
end
4'b1_010 : begin
    high_side[0] <= pwm[0];
    high_side[1] <= 1'b0;
    high_side[2] <= 1'b0;
    low_side <= 3'b010;
end
4'b1_110 : begin
    high_side[0] <= pwm[0];
    high_side[1] <= 1'b0;
    high_side[2] <= 1'b0;
    low_side <= 3'b100;
end
4'b1_100 : begin
    high_side[0] <= 1'b0;
    high_side[1] <= pwm[0];
    high_side[2] <= 1'b0;
    low_side <= 3'b100;
end
4'b1_101 : begin
    high_side[0] <= 1'b0;
    high_side[1] <= pwm[0];
    high_side[2] <= 1'b0;
    low_side <= 3'b001;
end

default: begin
    high side <= 3'b000;

    low_side <= 3'b000;
end
endcase
end

endmodule

```

## Lampiran 8 Kode Verilog HDL Logika Arah Putar

```
module select_pos(clk, adc, stop_flag, speed_sel);

input clk;
input [9:0] adc;
output reg stop_flag, speed_sel;

always@(posedge clk)
begin
    if(adc > 10'b01_1111_1110 && adc < 10'b10_0000_0001)
    begin
        stop_flag <= 1'b1;
    end
    else
        stop_flag <= 1'b0;
end

always@(adc)
begin
    case({adc[9]})
        1'b1 : begin //0 = A
            speed_sel <= 1'b1;
        end
        1'b0 : begin
            speed_sel <= 1'b0;
        end
    endcase
end

endmodule
```

## Lampiran 9 Kode Verilog HDL Top-level Kontrol BLDC

```
`timescale 1ns / 1ps

module control_top(clk, hall, mosi,miso,spiclk,cs, high_side,low_side);

input clk;
input miso;
input [2:0] hall;
output mosi,spiclk,cs;
output [2:0] high_side,low_side;

wire [9:0] adc_value;
wire done;
wire speed_sel;
wire stop_flag;
wire [1:0] pwm_out;

adc_top adc(clk,1'b1,mosi,miso,spiclk,cs,done,adc_value);
pwm_top pwm(clk, adc_value[8:1], pwm_out);
position_top pos(speed_sel, stop_flag, pwm_out, hall, high_side,low_side);
select_pos sel_pos(clk, adc_value, stop_flag,speed_sel);

endmodule
```

## Lampiran 10 Kode Verilog HDL Testbench

```
`timescale 1 ns / 10 ps

module control_bldc_tb();

reg clk;
reg [2:0] out_hall;
reg [9:0] out_adc = 10'd210;
wire [2:0] high_side, low_side;

reg [2:0] hall [0:5];

integer i = 0, j = 0;
integer period_hall;

control_top DUT(clk, out_hall, out_adc, high_side, low_side);

always
begin
    clk = 1'b1;
    #500;
    clk = 1'b0;
    #500;
end

//array hall position
initial begin
    hall[0] = 3'b001;
    hall[1] = 3'b011;
    hall[2] = 3'b010;
    hall[3] = 3'b110;
    hall[4] = 3'b100;
    hall[5] = 3'b101;
end

initial begin
    period_hall = 20_000_000;

    for(i=0;i<24; i = i + 1) begin
```

```
    out_adc <= out_adc + 45;
    for(j=0; j< 6 ; j= j+1) begin
        out_hall <= hall[j];
        #10000000;
        if(period_hall>2000000) begin
            period_hall = period_hall - 170000;
        end else begin
            period_hall = 1930000;
        end
    end
end
end

if (period_hall == 1930000) begin
    out_adc <= 10'd920;
    for(i=24;i<50; i = i + 1) begin
        for(j=0; j< 6 ; j= j+1) begin
            out_hall <= hall[j];
            #10000000;
        end
    end
end
end
end

endmodule
```