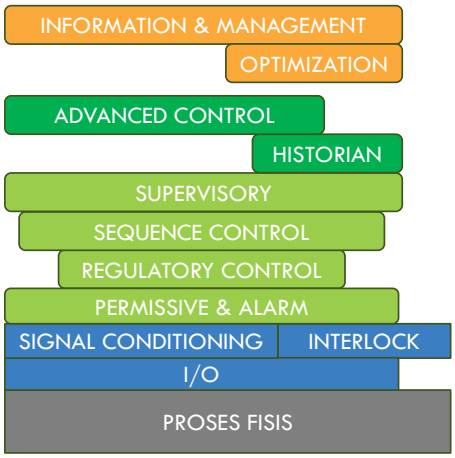
	Institut Teknologi Bandung Program Studi Teknik Fisika		
Praktikum	TF2207 Laboratorium TF II 2 SKS	Dosen	Dr. Ir. Eko Mursito Budi, M.T., IPM Dr.Eng. Muhammad Iqbal, S.T., M.T. Ashari Budi Nugraha, S.T., M.T.

5 KONTROL SEQUENSIAL

5.1 LATAR BELAKANG

Sistem kontrol adalah sistem yang dapat mengubah kondisi fisis secara otomatis. Secara keseluruhan, sistem kontrol sangat kompleks sehingga perlu ditata sebagai lapis-lapis seperti **Gambar 5.1**. Dengan mengacu pada arsitektur berlapis ini, keuntungan yang didapat adalah:

- Pembagian peran/fungsi komponen menjadi jelas.
- Modular, komponen antar lapis dapat saling bekerja selama interkoneksinya memenuhi standar.
- Komponen dapat dikembangkan secara independent.



Gambar 5.1. Arsitektur Berlapis Sistem Kontrol

Praktikum-praktikum sebelumnya telah dipelajari tentang berbagai I/O (digital, analog) dan kontrol regulatory. Praktikum kali ini mempelajari kontrol sekuensial, yaitu Kontrol yang memungkinkan sistem menjalani kondisi-kondisi secara berurut, dengan:

- Membaca data dari input
- Mengirim perintah langsung ke actuator
- Memberi set point ke Regulatory Control

Untuk merancang kontrol sequensial dapat digunakan metode:

- Finite state machine
- Petri-nets (untuk sistem yang melibatkan konkurensi)

5.2 KOMPETENSI

Pada praktikum ini, anda diharapkan untuk dapat menguasai beberapa kompetensi khusus yang berhubungan mekatronika dan sistem kontrol. Beberapa kompetensi khusus tersebut, adalah:

- Dapat menjelaskan cara kerja aktuator motor servo dan motor stepper serta mengoperasikannya dengan menggunakan mikrokontroller.
- Dapat merancang sistem kontrol sekuensial.
- Dapat membangun sistem dengan mengaplikasikan sistem sekuensial.

5.3 ALAT & BAHAN

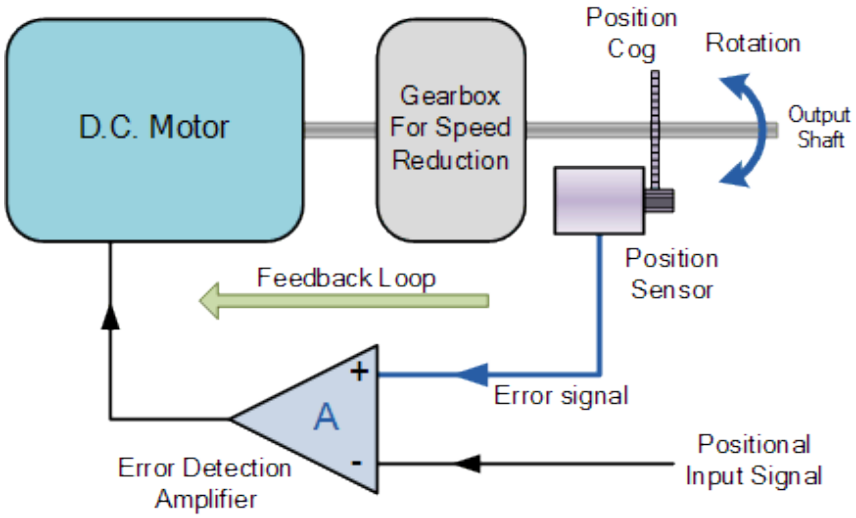
No	Item	Banyak	Keterangan
1	Breadboard	1	Dari Paket
2	EScope	1	
3	Kit motor stepper (dengan driver)	1	
4	Kit motor servo	1	
5	Kabel jumper	Secukupnya	
6	Kabel daya-USB (PENTING !)	1	
7			
8	Kabel micro-USB	1	Disiapkan peserta
9	Tang potong kabel	1	
10	Multimeter (jika perlu)	1	
11	Kepala charger HP 5V (minimal 1 A)	1	

5.4 PANDUAN TEKNIS

5.4.1 PERANGKAT KERAS

5.4.1.1 MOTOR SERVO

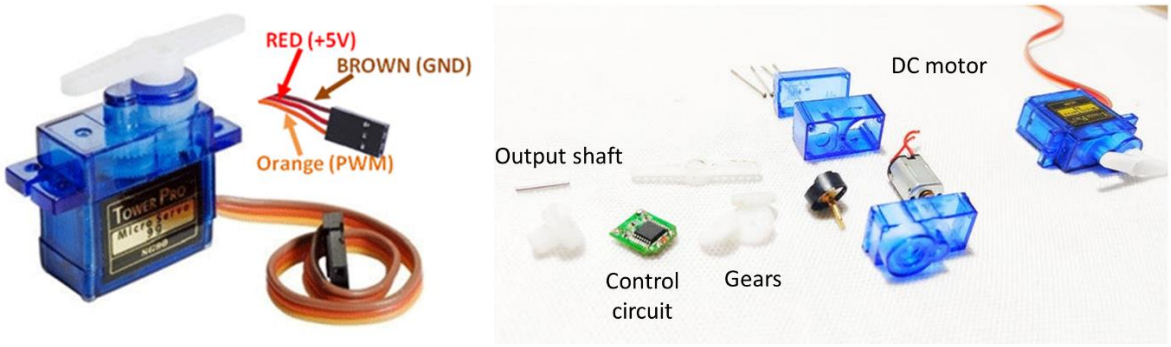
Motor servo pada dasarnya adalah motor dc yang dikontrol secara *servomechanism*. Diagram blok motor servo dc ditampilkan pada **Gambar 5.2**. Nampak bahwa terdapat sistem kontrol umpan balik di mana posisi sumbu motor output diumpankan ke rangkaian kontrol motor. Motor servo memiliki girboks *built-in* untuk mengurangi kecepatan rotasi motor dc dan mampu memberikan torsi yang tinggi secara langsung. Sumbu output motor servo tidak bisa berputar bebas seperti sumbu pada motor DC karena adanya girboks dan perangkat umpan balik. Motor servo dc pada umumnya tidak digunakan untuk konversi energi yang kontinu.



Gambar 5.2. Blok diagram motor servo dc. (Sumber: https://www.electronics-tutorials.ws/io/io_7.html).

Kecepatan atau posisi sumbu dikendalikan berdasarkan sinyal input posisi atau sinyal referensi yang diberikan ke perangkat motor servo. Amplifier pendeteksi error akan melihat sinyal masukan dan membandingkannya dengan sinyal umpan balik dari sumbu output motor dan menentukan apakah sumbu output motor berada dalam kondisi error. Jika kondisi error terdeteksi, pengontrol membuat koreksi yang sesuai, baik untuk mempercepat motor atau memperlambatnya. Respon terhadap perangkat umpan balik posisi ini mengindikasikan bahwa motor servo beroperasi dalam c*losed loop system*.

Pada praktikum Modul 5 ini, secara khusus kita akan menggunakan motor mikro servo dc SG90. Bentuk fisik dan komponen dalam dari motor micro servo dc SG90 ditunjukkan pada **Gambar 5.3**.



Gambar 5.3. Bentuk fisik motor mikro servo dc dan komponen dalamnya.

Motor mikro servo dc dikendalikan dengan sinyal PWM. Oleh karena itu kita perlu menghubungkan kabel sinyal dengan pin digital pada mikrokontroller yang memiliki kemampuan PWM. Motor mikro servo dc SG90 dapat berputar 180^o (90^o pada masing-masing arah) dan memiliki torsi 2,5 kg·cm. Rating tegangan operasinya adalah +5VDC. Kecepatan putar operasinya yaitu 0,1 detik/60^o. Untuk memberikan catu daya ke motor mikro servo dc dan mengendalikannya dengan mikrokontroller ESP32, kita perlu menghubungkan kabel sebagaimana ditunjukkan pada Tabel 5.1. Perhatikan bahwa motor servo mungkin akan memakai daya yang cukup tinggi, untuk itu kita memerlukan catu daya tambahan (bukan dari line +5V EScope). Pastikan menghubungkan ground dari catu daya eksternal bersama dengan ground EScope.

Tabel 5.1. Pengkabelan Motor Mikro Servo DC SG90 dan EScope.

Warna Kabel	Deskripsi
Brown	Kabel ground, dihubungkan ke line GND
Red	Kabel daya, dihubungkan ke line +5V
Orange	Sinyal PWM untuk mengendalikan motor, dihubungkan dengan pin PWM Escope

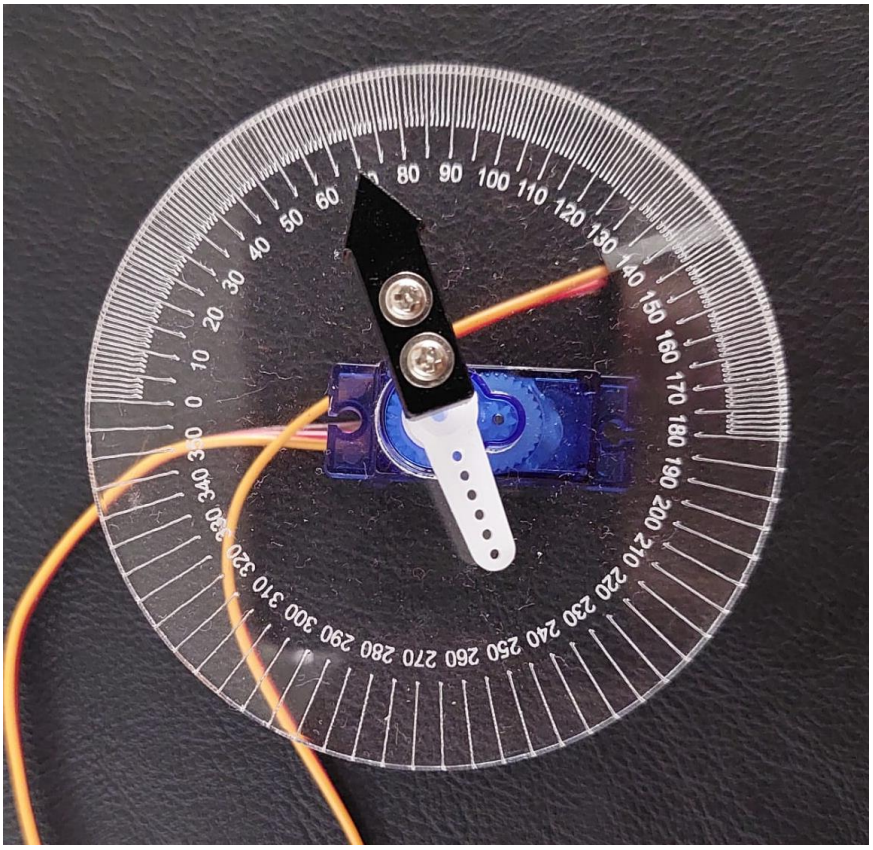
Untuk mengendallikan motor mikro servo SG90 dengan EScope, kita dapat memanfaatkan pustaka `Servo.h`. Dapatkan/install pustaka tersebut seperti biasa. Kode sumber utama untuk inisialisasi motor mikro servo dc SG90 menggunakan pustaka `Servo.h` ditampilkan pada **Kode 5.1**.

```
/* Program : Motor Servo
 * Test Motor Servo
 */

#include <Servo.h>

#define SERVO_PIN 13 // pin GPIO 13 alias DO0 pada EScope

Servo servoMotor; // membuat objek servoMotor pada class Servo

void setup() {
  servoMotor.attach(SERVO_PIN); // pin sinyal PWM motor dihubungkan ke DO0
}

void loop() {
  servoMotor.write(0); // jarum motor akan bergerak ke posisi 0 derajat
  delay(3000);

  servoMotor.write(45); // jarum motor akan bergerak ke posisi 45 derajat
  delay(3000);

  servoMotor.write(90); // jarum motor akan bergerak ke posisi 90 derajat
  delay(3000);

  servoMotor.write(180); // jarum motor akan bergerak ke posisi 180 derajat
  delay(3000);
}
```

Kode 5.1. Kode sumber mengendalikan motor mikro servo dc menggunakan pustaka Servo.h

Perhatikan bahwa besarnya sudut yang diberikan pada fungsi `servoMotor.write()` merupakan posisi sudut sekian derajat, bukan berupa perpindahan sejauh besarnya sudut tersebut.

Pada Modul 5 ini, motor servo dc dilengkapi dengan busur dan jarum penujuk derajat sudut yang terbuat dari akrilik (**Gambar 5.4**).



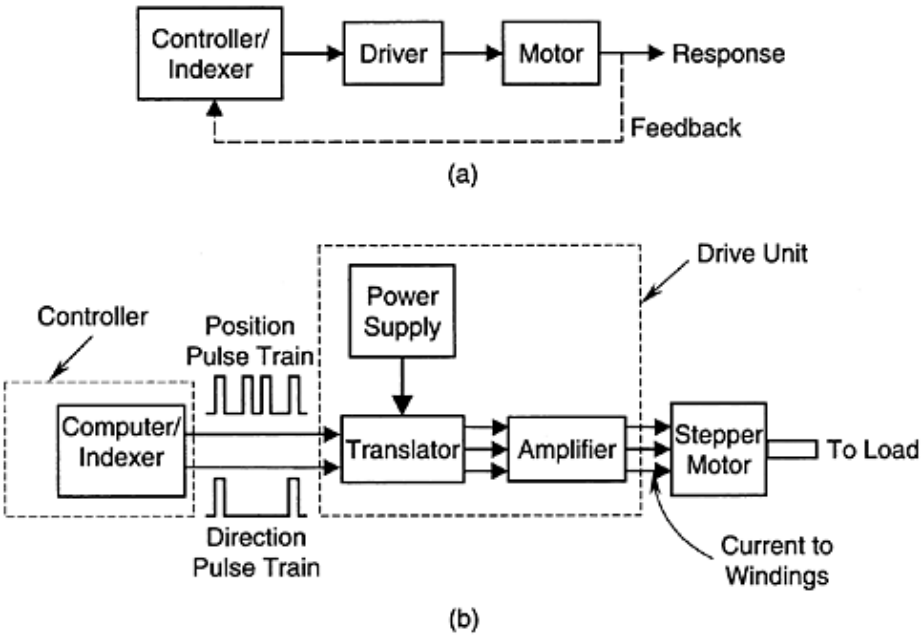
Gambar 5.4. Busur dan panah penunjuk derajat sudut motor servo yang terbuat dari akrilik.

5.4.1.2 MOTOR STEPPER

Motor stepper merupakan *increment-drive actuator* di mana motor jenis ini dikendalikan oleh sinyal step angular yang tetap. Motor ini terdiri dari roda bergerigi dan empat buah kumparan elektromagnet. Kumparan elektromagnet tersebut akan membuat roda bergerigi berputar ketika sinyal pulsa HIGH. Setiap pulsa HIGH yang dikirim, memberikan energi pada kumparan yang kemudian akan menarik gigi terdekat dari roda bergerigi dan menggerakkan motor satu step/langkah. Konstruksi tersebut menyebabkan motor stepper dapat diatur arah geraknya pada arah tertentu, misalnya searah jarum jam atau berlawanan dengan arah jarum jam. Cara menggerakkan kumparan elektromagnet pada motor ini sangat mempengaruhi perilaku motor, di mana:

- urutan pulsa menentukan arah putaran motor,
- frekuensi pulsa menentukan kecepatan motor, dan
- jumlah pulsa menentukan seberapa jauh motor akan berputar.

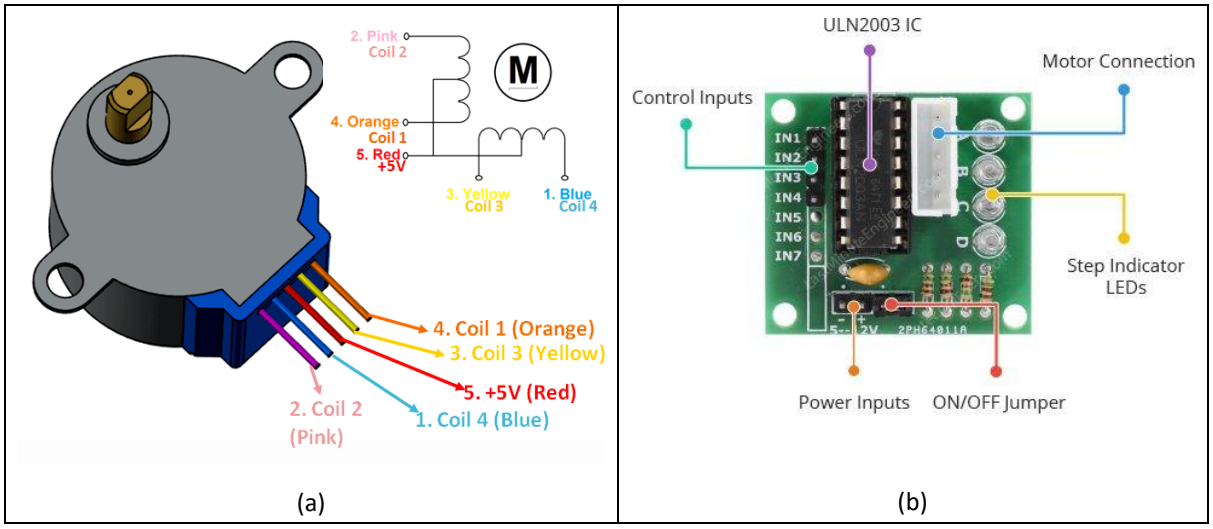
Motor stepper memerlukan suatu "*indexer*" untuk menghasilkan pulsa perintah dan "*driver*" untuk menafsirkan perintah dan menghasilkan arus yang tepat untuk fasa motor. Blok diagram secara umum dan aliran energi/sinyal pada pengendalilan motor stepper ditunjukkan pada **Gambar 5.5**.



Gambar 5.5. (a) Blok diagram umum pengendalian motor stepper dan (b) blok diagram dan aliran energi/sinyal dalam pengendalian motor stepper.

Pada praktikum Modul 5 ini, motor stepper yang digunakan adalah motor stepper 28BYJ-48 (**Gambar 5.6a**). Adapun yang berperan sebagai computer/indexer adalah mikrokontroller ESP32. Sedangkan yang berperan sebagai translator dan amplifier adalah modul driver motor berbasis IC ULN2003 (**Gambar 5.6b**).

Motor stepper 28BYJ-48 memiliki total empat kumparan. Salah satu ujung kumparan terhubung ke 5V, yang terhubung ke kabel motor berwarna merah. Ujung yang lain dari keempat kumparan terhubung dengan kabel berwarna biru, merah muda, kuning, dan oranye. Memberi energi pada kumparan dalam urutan logika membuat motor bergerak satu langkah ke suatu arah atau ke arah yang lainnya. Motor stepper 28BYJ-48 memiliki *stride angle* 5,625° / 64 dalam mode *half-step*. Artinya motor memiliki *step angle* 5,625 ° —sehingga dibutuhkan 360° / 5,625° = 64 langkah pada mode *half-step*. Dalam mode *full-step*, diperlukan 64 / 2 = 32 langkah untuk menyelesaikan satu putaran.



Gambar 5.6. (a) Skematik motor stepper 28BYJ-48 dan (b) driver motor stepper berbasis IC ULN2003.

Namun, sumbu poros output digerakkan dengan rasio roda gigi 64:1. Hal ini berarti sumbu poros yang terkspos di luar motor akan menyelesaikan putaran jika motor di dalam berputar 64 kali. Dengan demikian, motor harus bergerak 32 x 64 = 2048 langkah agar sumbu poros dapat menyelesaikan satu putaran penuh. Hal ini mengindikasikan bahwa motor stepper akan memiliki tingkat presisi 360º/2048 langkah = 0,176º/langkah. Jadi, motor stepper ini memiliki *steps per revolution* = 2048 langkah dan *step angle* = 0,176º/langkah.

Sebagaimana disebutkan di atas, untuk menghubungkan motor stepper dengan ESP32, kita akan menggunakan driver motor ULN2003. Modul driver motor dilengkapi dengan konektor yang memudahkan untuk menghubungkan motor ke modul. Modul driver memiliki empat pin input untuk mengontrol kumparan yang membuat motor stepper bergerak. Keempat LED menampilkan antarmuka visual dari keadaan kumparan. Untuk memudahkan dalam pemrograman, digunakan pusataka `Stepper.h` yang dapat diinstall pada library Arduino IDE. Pemrograman sederhana untuk mengendalikan gerak motor stepper sejauh sudut yang ditentukan menggunakan pustaka `Stepper.h` ditampilkan pada **Kode 5.2**.

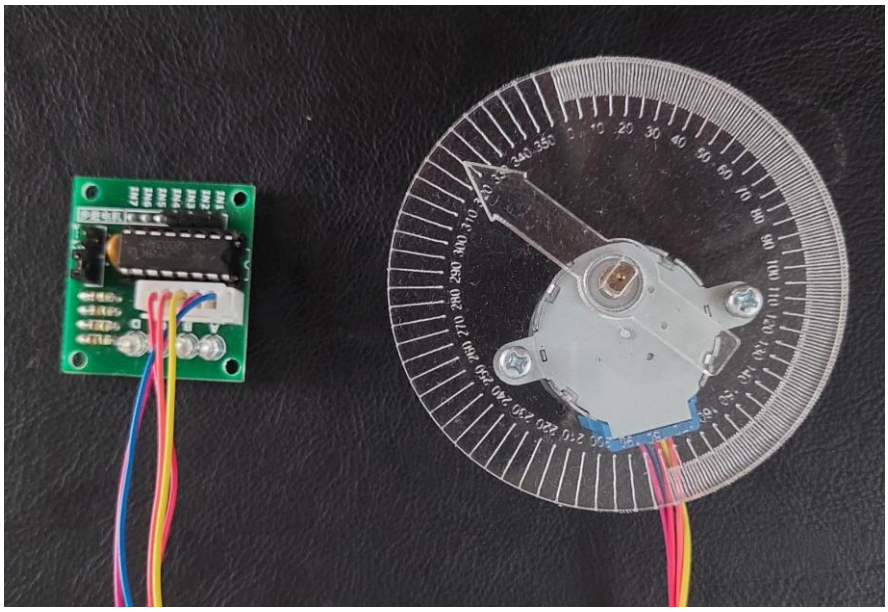
```
/* Program : Motor Stepper
 * Test Motor Stepper
 */

#include <Stepper.h>

// Pin drive motor ULN2003
#define IN1 13
#define IN2 12
#define IN3 14
#define IN4 27

float stepsPerRevolution = 2048;  // jumlah langkah per satu putaran penuh
float revolution = 360; // 1 revolusi penuh 360 derajat
float step_angle = revolution / stepsPerRevolution; // step angle

Stepper myStepper(stepsPerRevolution, IN1, IN3, IN2, IN4); // memesan objek
dengan class Stepper

// Konversi dari sudut menjadi step/langkah putaran motor
void step_Degree(int degree){
  myStepper.step(degree/step_angle);
}

void setup() {
  myStepper.setSpeed(15);
  Serial.begin(500000);
}

void loop() {
  step_Degree(45);
  delay(2000);
}
```

Kode 5.2. Kode sumber sederhana pengendalian motor stepper menggunakan pustaka Stepper.h.

Setiap kali fungsi `loop()` pada kode sumber di atas dijalankan, jarum pada motor stepper akan bergerak sejauh 45º. Perhatikan perbedaan pergerakan motor servo dengan motor stepper. Pada motor servo, nilai sudut akan

menggerakkan jarum ke posisi sudut tersebut. Sementara, nilai sudut pada motor stepper akan menggerakkan jarum sejauh besarnya sudut tersebut.

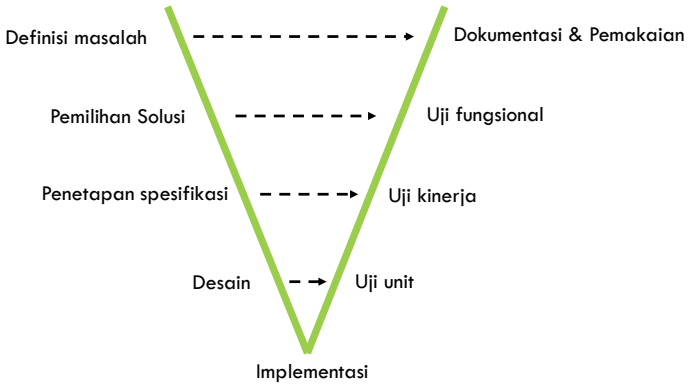
Pada Modul 5 ini, motor stepper juga dilengkapi dengan busur dan jarum penujuk derajat sudut yang terbuat dari akrilik (**Gambar 5.7**).



Gambar 5.7. Busur dan panah penunjuk derajat sudut yang terbuat dari akrilik.

5.4.2 KONTROL SEQUENSIAL

Salah satu metodologi (tahap-tahap umum) rekayasa sistem adalah metode V seperti gambar 5.8.



Gambar 5.8. Metode pengembangan V

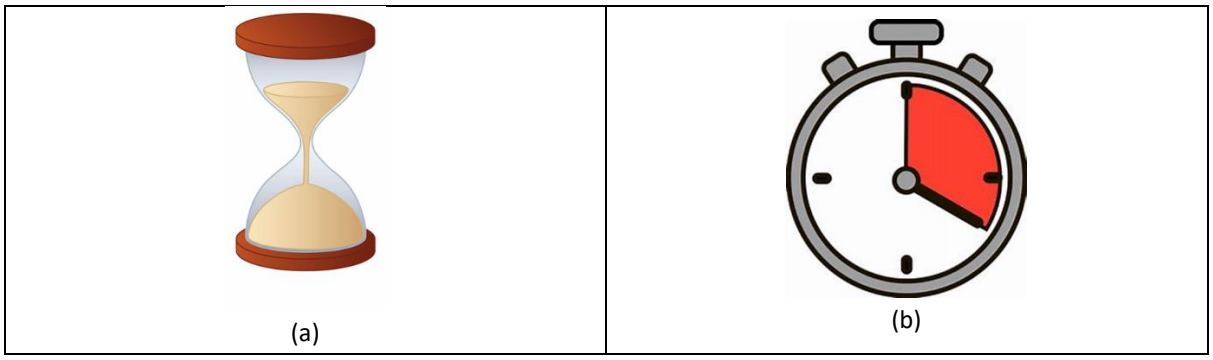
5.4.2.1 DEFINISI MASALAH

Pada tahap ini, dilakukan kajian atas masalah yang dihadapi pemakai, atau bisa juga disebut kebutuhan pemakai (*user requirement*). Sebagai contoh:

- Pemakai ingin mengukur selang waktu tertentu dengan ketelitian satu detik.

5.4.2.2 PEMILIHAN SOLUSI

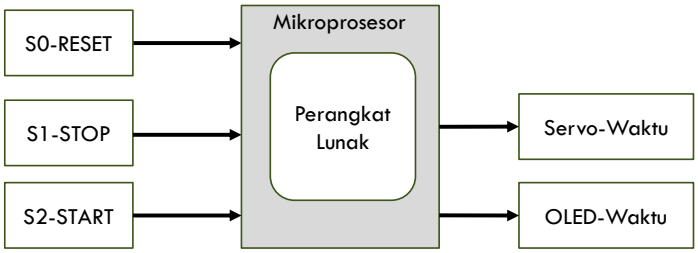
Pada tahap ini, insinyur mencari berbagai alternatif solusi dan memilih satu yang terbaik. Sebagai contoh, dari Solusi-solusi sebelumnya, untuk mengukur selang waktu bisa dipakai jarum pasir, atau stop watch. Dalam hal ini dipilih stop watch sebagai dasar ide (pemilihan solusi sebaiknya sistematis dan obyektif).



Gambar 5.9. Alternatif solusi

Untuk membuat stopwatch ini, solusi dapat digambarkan dengan diagram blok, dan selanjutnya didefinisikan cara menggunakannya, misalnya:

1. Saat dihidupkan, sistem dalam keadaan berhenti mencacah.
2. Sebagai awalan, pengguna harus menekan tombol RESET sehingga sistem akan mulai menghitung waktu dari 0 dan menampilkannya ke Servo-waktu (dan OLED).
3. Pengguna dapat menekan tombol START untuk mulai mencacah waktu dengan periode tertentu, dan sistem akan menampilkannya pada Servo-waktu (dan OLED).
4. Ketika pengguna menekan tombol STOP, maka sistem seketika menghentikan cacahan waktu, dan menampilkan waktu pada Servo-waktu (dan OLED).
5. Pengguna dapat meneruskan cacahan dengan menekan tombol START lagi.
6. Apabila cacahan sudah mencapai maksimum maka cacahan akan berhenti.
7. Pengguna dapat mengulangi cacahan waktu dari 0 dengan menekan tombol RESET.



Gambar 5.10. Diagram blok sistem stop-watch

5.4.2.3 PENETAPAN SPESIFIKASI

Setelah solusi yang diinginkan cukup jelas, dapat dilakukan penetapan spesifikasi sistem. Spesifikasi yang tinggi biasanya didorong oleh kebutuhan pengguna, namun dibatasi oleh berbagai hal seperti ketersediaan teknologi

maupun anggaran biaya. Untuk kasus stop watch ini, waktu akan ditunjukkan oleh motor stepper yang hanya dapat berputar 270^o. Dengan demikian spesifikasi adalah:

Parameter	Nilai	Keterangan
Rentang waktu	0 – 270 detik	Dibatasai oleh kemampuan motor servo
Resolusi waktu	1 detik	Memenuhi kebutuhan pengguna
Input	3 tombol tekan	Memenuhi solusi
Output / Tampilan	Piringan dengan jarum yang digerakkan motor servo	Memenuhi solusi

Pada contoh ini terjadi pembatasan oleh teknologi. Jika hal ini belum memenuhi kebutuhan pengguna, perlu dicari teknologi yang lain selama memang memungkinkan (misalnya waktu ditunjukkan oleh OLED).

5.4.2.4 DESAIN

Tahap desain adalah langkah metodologis untuk mewijudkan solusi dengan menciptakan sesuatu yang nyata (*tangible*) berupa komponen-komponen yang terstruktur dan memiliki fungsi tertentu. Pada rekayasa sistem embedded, desain dilakukan untuk perangkat keras dan perangkat lunak.

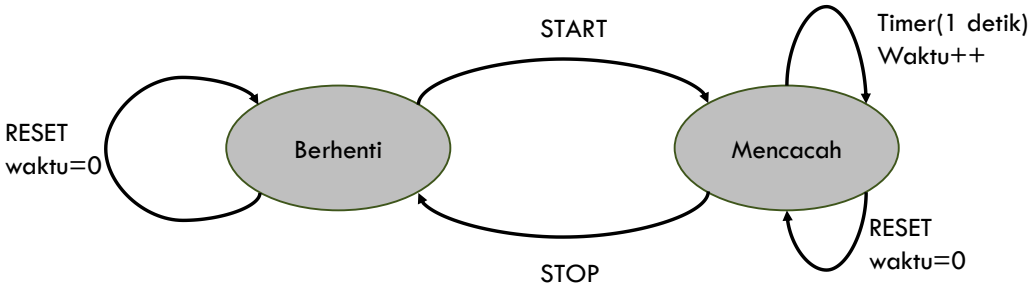
Pada contoh stopwatch ini, perangkat kerasnya adalah kit Escope yang dilengkapi kit motor servo. Dengan demikian :

- Komponen input adalah 3 buah tombol pada Escope (S0 – S1).
- Komponen output adalah kit motor servo, dengan jarum yang bisa menunjukkan sudut 0 – 270 derajat pada piringan. Sebagai tambahan, waktu juga bisa ditampilkan pada OLED.

Sementara itu perangkat lunak sistem sequensial, langkah metodologis yang paling umum adalah menggunakan metode finite state automation (FSA), yaitu dengan :

- Mendefinisikan masukan dan luaran
- Menentukan state (kondisi sistem)
- Menentukan transisi dan syarat terjadinya transisi

Untuk mendesainnya dapat digunakan tabel transisi keadaan, maupun diagram transisi keadaan. Sebagai contoh, desain FSA untuk sistem stop watch adalah sebagai berikut.



Gambar 5.11. Diagram Transisi Keadaan sistem stop-watch

5.4.2.5 IMPLEMENTASI

Implemetasi adalah tahap merealisasikan desain menjadi wujud nyatanya. Untuk sistem ini, perangkat keras dapat dirangkai dengan mengkoneksikan Escope dan kit motor servo. Sementara itu perangkat lunak yang sudah didesain menggunakan FSA dapat diprogram menggunakan algoritma *state driven* atau *event driven*.

5.4.2.6 PENGUJIAN

Sesuai dengan paradigma pengembangan V, maka ada tiga tahap pengujian yaitu:

- pengujian unit dilakukan untuk meyakinkan bahwa berbagai komponen / blok diagram sistem sudah berfungsi. Pengujian ini biasanya dilakukan secara whitebox (berdasar desain detail).
- Pengujian kinerja dilakukan untuk membuktikan bahwa sistem sudah memenuhi spesifikasinya. Dalam pengujian ini perlu dilakukan pengukuran menggunakan alat ukur yang valid. Misalnya saja untuk sistem stop watch ini, perlu dibandingkan dengan stop-watch yang asli dan lebih teliti.
- Pengujian fungsional membuktikan membuktikan bahwa sistem sudah memenuhi solusi yang diinginkan, khususnya dari segi cara kerjanya. Untuk melakukan uji fungsi maka perlu disusun skenario uji dan dapat dilakukan secara black-box.

5.4.2.7 DOKUMENTASI

Setelah sistem selesai dibuat, perlu dibuat dokumentasi, diantaranya:

- Dokumentasi pengguna: ditujukan untuk pengguna sistem (*users*).
- Dokumentasi teknis: ditujukan untuk teknisi.
- Dokumentasi internal: mencatat pengalaman untuk peningkatan di rekayasa berikutnya.

5.5 TUGAS AWAL

Untuk menambah kesiapan Anda mengikuti praktikum Modul 5, setiap mahasiswa harus mengerjakan tugas awal berikut:

a) Motor servo
   - Carilah datasheet/spesifikasi motor servo SG90. Jelaskan kemampuan / keterbatasan motor servo tersebut.
   - Jelaskan cara kerja motor servo sehingga motor tersebut mikrokontroller dapat menggerakkan motor servo dengan menggunakan output PWM.

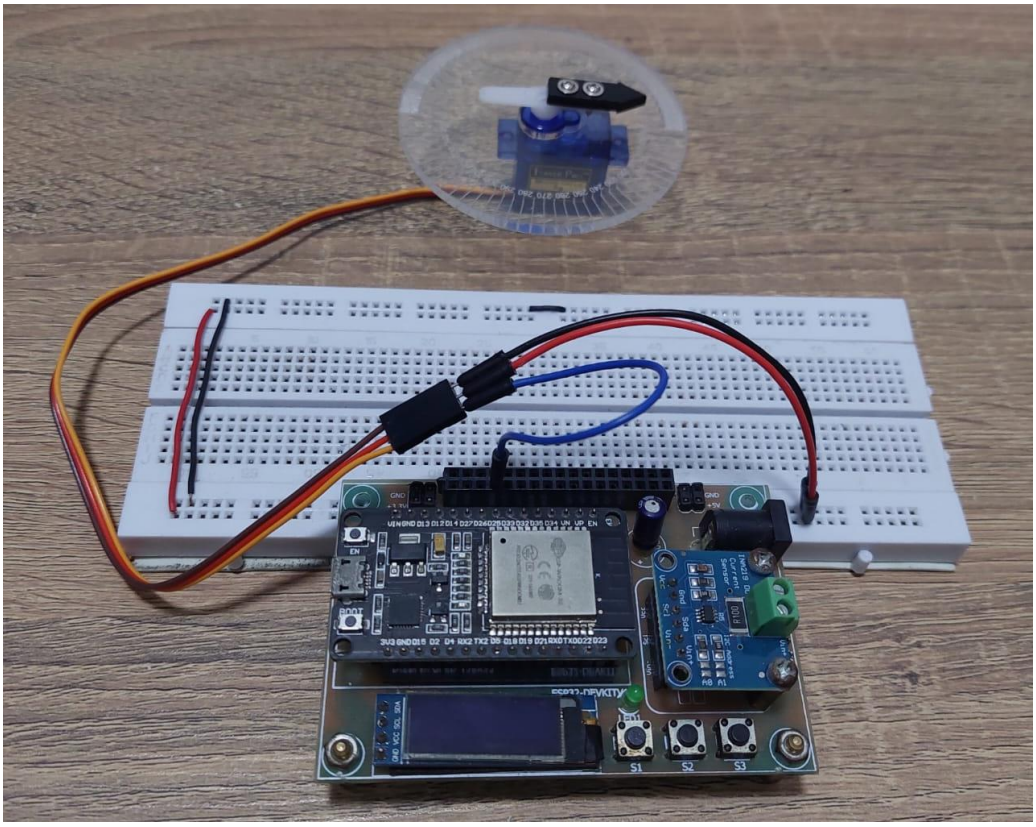
b) Motor stepper
   - Carilah datasheet/spesifikasi motor stepper 28BYJ-48. Jelaskan kemampuan / keterbatasan motor stepper tersebut.
   - Carilah referensi driver motor stepper berbasis ULN2003, dan gambarkan skematiknya.
   - Jelaskan cara kerja driver motor berbasis IC ULN2003 sehingga mikrokontroller dapat menggerakkan motor stepper tersebut melalui 4 digital output.

Selain mengerjakan tugas awal di atas, bagilah persiapan praktikum 5.5.1 hingga 5.5.3 di bawah ini di antara anggota regu.

5.5.1 MERANGKAI KIT MOTOR SERVO DAN ESCOPE

Sebagai bahan persiapan praktikum, rangkailah kit motor masing-masing sesuai petunjuk di bawah ini. Foto kesiapan rangkaian Anda dan masukkan ke dalam tugas awal.

A. Gambar Rangkaian



B. Pengkabelan (*Wiring*)

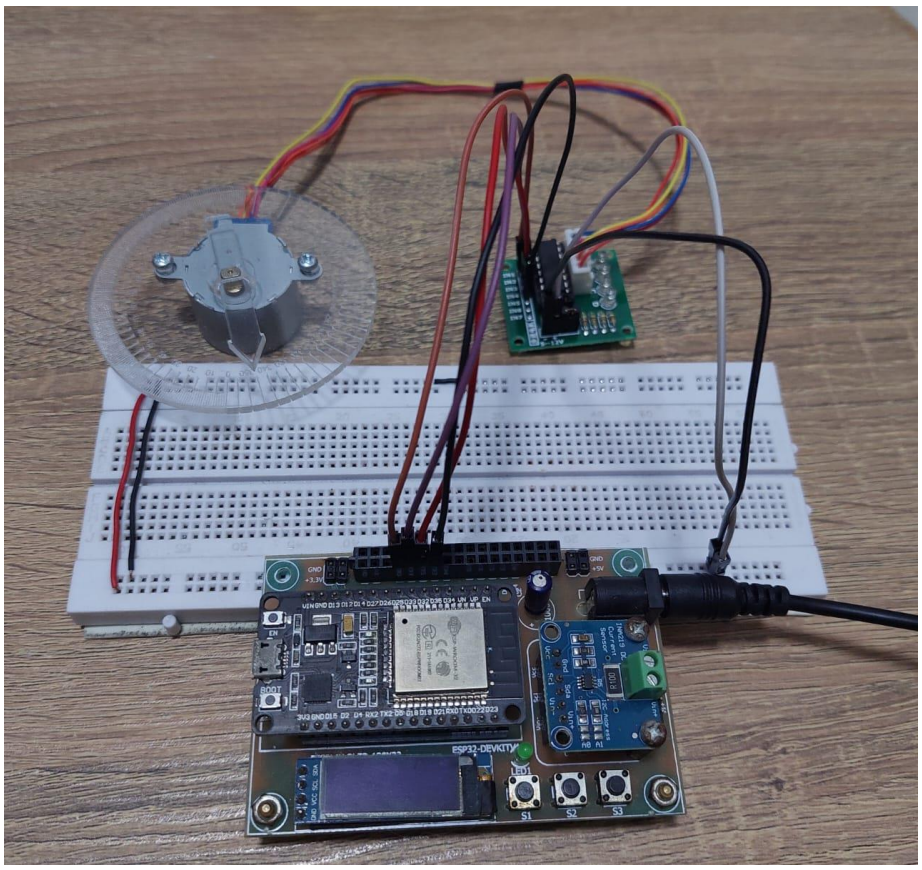
Motor Servo	EScope	Breadboard
Brown	⟷	Line GND (kanan)
Red	⟷	Line +5V
Orange	⟷ DO0	-

C. Pemrograman

Cobalah rangkaian ini dengan memrogramnya sesuai dengan **Kode 5.1**.

5.5.2 KIT MOTOR STEPPER

A. Gambar Rangkaian



B. Pengkabelan (*Wiring*)

Motor Stepper	Driver ULN2003	EScope	Breadboard
Soket kabel biru, merah muda, merah, kuning, dan oranye	↔ Soket kabel biru, merah muda, merah, kuning, dan oranye	-	-
IN1	←	→ DO0-	-
IN2	←	→ DO1	-
IN3	←	→ DO2	-
IN4	←	→ DO3	-
-	GND ←	→	Line GND (kanan)
-	VCC ←	→	Line +5V

C. Pemrograman

Cobalah rangkaian ini dengan memrogramnya sesuai dengan **Kode 5.2**.

PERHATIAN!

Untuk praktikum motor servo maupun motor stepper, harus ditambahkan catu daya 5V. Sambungkan kabel daya-USB pada jack power EScope ke kepala charger 5VDC (minimal 1A). Kemudian colokkan kepala chargernya ke colokan listrik PLN (220V) di tempat tinggal Anda. Tanpa catu daya tambahan, besar resiko mikroprosesor akan rusak karena arus berlebih.

5.5.3 SISTEM STOP-WATCH

Sesuai dengan desain stop-watch yang sudah dilakukan pada bagian teori, siapkan:

- Kode sumber untuk implementasi stop-watch, menggunakan algoritma state-driven atau event-driven.
- Skenario untuk uji fungsional.

5.6 PRAKTIKUM

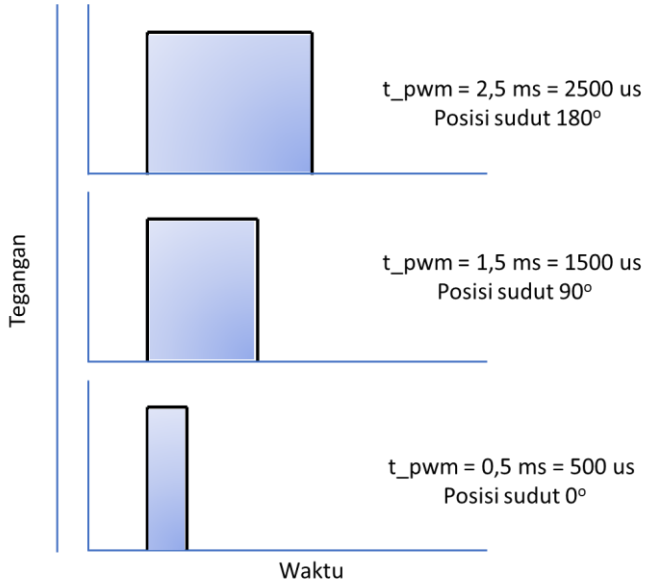
Berikut ini adalah panduan percobaan yang akan dilakukan untuk setiap jenis motor.

No.	Jenis Kit Motor	Percobaan
1.	Motor Servo	Kalibrasi sudut motor servo
		Kontrol motor servo hasil kalibrasi melalui serial monitor
2.	Motor Stepper	Kalibrasi sudut motor stepper
		Pengukuran kecepatan rotasi motor stepper

Kode sumber untuk setiap percobaan dapat diunduh di MS Teams.

5.6.1 KALIBRASI SUDUT MOTOR SERVO

Pada percobaan ini, akan dilakukan kalibrasi posisi sudut dengan periode PWM (`t_pwm`) yang diberikan ke motor servo sehingga nantinya kita dapat menggerakkan posisi motor servo tanpa menggunakan pustaka `Servo.h`. Sebagaimana telah dibahas, motor servo dikendalikan dengan sinyal PWM. Pada dasarnya, sinyal PWM merupakan sinyal pulsa yang periode/durasi ON-nya dapat diubah dengan amplitudo dan frekuensi yang tetap. Motor servo SG90 memiliki karakteristik hubungan antara periode PWM (t_pwm) dengan sudut sebagai berikut:



- Pada kode sumber M5-02-Kalibrasi_Motor_Servo, carilah fungsi `servoPulse()` yang mengonversi sudut menjadi peridoe PWM (t_pwm). Pin PWM diprogram untuk menghasilkan sinyal pulsa HIGH selama periode waktu tertentu yang diatur dengan `delayMicroseconds(t_pwm)`.
- Carilah hubungan antara sudut dan t_pwm sebagaimana karakteristik hubungan yang ditunjukkan pada Gambar di atas, kemudian lengkapi kode sumber tersebut.
- Jalankan kode sumber dan amati apakah posisi jarum motor servo sudah sesuai dengan penunjukkan derajat pada busur. Luncurkan juga Serial Monitor dan dapatkan tabel yang berisi kolom Sudut dan Periode PWM.
- Jika posisi jarum belum sesuai, koreksi kembali fungsi konversi sudut menjadi periode PWM tersebut.

5.6.2 KONTROL MOTOR SERVO MENGGUNAKAN SERIAL MONITOR

Pada percobaan ini, gerak motor servo pada posisi tertentu akan dilakukan melalui perintah yang diberikan melalui Serial Monitor.

1. Gunakan fungsi konversi sudut/posisi menjadi t_pwm motor servo pada percobaan sebelumnya.
2. Buka dan revisi kode program M5-03-Kontrol_Servo_Serial agar dapat dilakukan penunjukkan jarum sesuai dengan sudut/posisi yang diberikan melalui Serial Monitor.
3. Luncurkan Serial Monitor, masukkan posisi derajat dan tekan Enter agar jarum motor servo bergerak menuju posisi yang dimaksud.
4. Amati yang terjadi dengan gerakan jarum motor servo, apakah sudah sesuai?

5.6.3 KONTROL MOTOR STEPPER DAN KALIBRASI SUDUT

Pada percobaan ini, akan dilakukan kalibrasi sudut yang diperintahkan ke motor stepper dengan penunjukkan jarum motor.

1. Buka dan unggah kode sumber M5-04-Kalibrasi_Motor_Stepper.
2. BT0 menggerakkan jarum sebesar 15 derajat berlawanan arah jam. Luncurkan Serial Monitor.
3. Catat posisi awal jarum seperti yang dapat dilihat di busur derajat.
4. Tekan tombol reset terlebih dahulu di ESP32 (tombol EN) untuk mengembalikan `count` ke 0.
5. Tekan tombol BT0 berkali-kali hingga jarum berputar 360 derajat (periksa pada motor maupun pada Serial Monitor).
6. Jika gerak jarum dan besar sudut tidak sesuai, cari/hitung faktor koreksinya dan perbaiki pada fungsi `step_Degree()`.

5.6.4 PENGUKURAN KECEPATAN ROTASI MOTOR STEPPER

Pada percobaan ini, akan dilakukan pengukuran kecepatan rotasi motor stepper menggunakan `millis()`.

1. Buka dan unggah kode sumber M5-05-Kecepatan_Motor_Stepper.
2. Lunucurkan Serial Monitor.
3. Tekan tombol S1 (BT0) atau S3 (BT2) untuk menggerakkan jarum motor stepper sejauh 360^{o} masing-masing pada arah berlawanan jarum jam atau searah jarum jam dengan kecepatan tertentu.
4. Tombol S2 (BT1) untuk menghentikan gerakan motor.
5. Waktu tempuh satu revolusi dihitung berdasarkan persamaan `elapsed time = current_time – start_time`.
6. Waktu tempuh otomatis muncul pada Serial Monitor setiap kali program tombol S1 atau S3 ditekan.
7. Kecepatan putar per menit diperoleh dari waktu tempuh yang diukur dari hasil perhitungan. Bandingkan dengan nilai yang ada diisikan di `myStepper.setSpeed()`.

5.6.5 IMPLEMENTASI & PENGUJIAN STOP-WATCH

Sesuai dengan tugas awal yang sudah dilakukan, program sistem stop-watch dan lakukan pengujian sehingga memenuhi solusi dan spesifikasi yang diinginkan.

5.7 TUGAS TANTANGAN

Lakukan tugas tantangan berikut sesuai tahap-tahap rekayasa model V, dengan laporan yang lengkap.

Ingin dibuat suatu piring putar yang dapat bergerak ke tiga posisi oleh 3 tombol dengan cara kerja sebagai berikut:

- Posisi piring ditunjukkan oleh jarum yang digerakkan oleh motor stepper pada piringan sudut. Tentukan 3 posisi yang menjadi acuan (misalnya sudut P0=0^o, P1=120^o, P2=240^o derajat).
- Terdapat 3 tombol (S0 – S2) yang masing-masing akan memanggil jarum ke posisi yang sesuai (P0 – P3), dimana aturannya adalah :
    - Jarum harus bergerak dari posisi sebelumnya ke posisi pemanggilan dengan jarak sudut yang terdekat. Misalnya jika saat ini jarum pada posisi P0 lalu tombol S2 ditekan, maka jarum berputar langsung ke P2 tanpa melewati P1.
    - Jika jarum sedang bergerak kemudian ada tombol lain ditekan, maka gerakan sebelumya akan dibatalkan, dan jarum langsung bergerak ke posisi pemanggilan terakhir melalui jarak terdekat.

5.8 PUSTAKA

- Pulse Width Modulation (PWM) : https://www.electronicshub.org/esp32-pwm-tutorial/ ; https://randomnerdtutorials.com/esp32-pwm-arduino-ide/
- Interrupt: https://www.theengineeringprojects.com/2021/12/esp32-interrupts.html
- Motor Servo: https://randomnerdtutorials.com/esp32-servo-motor-web-server-arduino-ide/
- Motor Stepper: https://randomnerdtutorials.com/esp32-stepper-motor-28byj-48-uln2003/; https://www.instructables.com/BYJ48-Stepper-Motor/
- Motor DC: https://randomnerdtutorials.com/esp32-dc-motor-l298n-motor-driver-control-speed-direction/
- Sensor Opto-Coupler: https://www.instructables.com/How-to-Use-TCRT5000-IR-Sensor-Module-With-Arduino-/ ; https://diyi0t.com/tcrt5000-line-tracking-module-arduino-esp8266-esp32/