

**LAPORAN TUGAS BESAR
MIKROPROSESOR DAN ANTARMUKA**

SISTEM KENDALI SUHU RUANG DENGAN STM32F411



Kelompok: 7

Firdaus Arif Ramadhani (101032300131) - [Bagian Tugas]

Azmi Aziz Syahputra (101032300216) - [Bagian Tugas]

Muhammad Haiqal (101032300125) - [Bagian Tugas]

Muhammad Arman Baihaqi (101032300190) - [Bagian Tugas]

Dosen: Faisal Candrasyah Hasibuan, S.T., M.T.

**UNIVERSITAS TELKOM
FAKULTAS TEKNIK ELEKTRO
PROGRAM STUDI TEKNIK KOMPUTER
SEMESTER 5 - TAHUN AKADEMIK 2025/2026**

ABSTRAK

Laporan ini menjelaskan implementasi sistem kendali kipas berbasis suhu menggunakan mikrokontroler STM32F411 untuk memenuhi CLO3. Sistem dirancang untuk memonitor suhu ruangan dan mengatur kecepatan kipas DC secara otomatis.

Implementasi meliputi: (1) komunikasi serial UART untuk logging data suhu ke komputer, (2) timer interrupt untuk sampling sensor secara periodik, (3) external interrupt untuk input tombol pengatur setpoint suhu.

Sistem terdiri dari empat subsistem utama yaitu subsistem input berupa sensor LM35 dan tombol EXTI, subsistem timing menggunakan timer dengan interrupt periodik, subsistem output berupa PWM dan driver transistor, serta subsistem komunikasi melalui UART. Hasil pengujian menunjukkan *[hasil key findings]*. Implementasi berbasis register menunjukkan pemahaman arsitektur STM32 pada level peripheral.

Kata Kunci: STM32F411, UART, Timer Interrupt, External Interrupt, PWM, Register Programming, Sensor Suhu

ABSTRACT

This report describes the implementation of a temperature-based fan control system using an STM32F411 microcontroller to meet CLO3. The system is designed to monitor room temperature and automatically adjust the speed of a DC fan.

The implementation includes: (1) UART serial communication for logging temperature data to a computer, (2) timer interrupts for periodic sensor sampling, (3) external interrupts for temperature setpoint control button inputs.

The system consists of four main subsystems, namely the input subsystem in the form of an LM35 sensor and EXTI button, the timing subsystem using a timer with periodic interrupts, the output subsystem in the form of PWM and transistor drivers, and the communication subsystem via UART. The test results show *[key findings]*. The register-based implementation demonstrates an understanding of the STM32 architecture at the peripheral level.

Keywords: STM32F411, UART, Timer Interrupt, External Interrupt, PWM, Register Programming, Temperature Sensor

BAB I: PENDAHULUAN

1.1 Latar Belakang

Mikrokontroler merupakan sebuah komponen sistem tertanam yang di dalamnya terdapat mikroprosesor, memori, dan periferan I/O. Komponen ini mampu melakukan pemrosesan data secara digital sesuai dengan perintah bahasa program yang diberikan. Dengan memanfaatkan mikrokontroler, sebuah sistem dapat dikendalikan karena adanya mikroprosesor sebagai otak utama yang mengontrol fungsi sistem tertanam. Mikrokontroler dapat membaca data sensor, memprosesnya, dan mengendalikan aktuator secara *real-time* yang menjadikannya perangkat ideal untuk sebuah sistem kendali otomatis. Dengan adanya kemampuan otomatisasi, sebuah aktuator pada sistem kendali dapat bekerja secara terkontrol menyesuaikan masukan dari data sensor dan umpan balik secara *real-time* tanpa membutuhkan tindakan eksternal dalam hal mekanis aktuator.

Komunikasi serial merupakan metode pertukaran data yang biasa digunakan dalam sistem tertanam. UART (*Universal Asynchronous Receiver-Transmitter*) adalah salah satu protokol komunikasi serial antar perangkat yang digunakan oleh papan mikrokontroler untuk berkomunikasi dengan komputer^[1]. Protokol ini berfungsi sebagai komunikasi serial asinkron di mana format data dan kecepatan transmisi dapat dikonfigurasi. Pada sistem kendali kipas berbasis suhu, UART berguna dalam hal proses *debugging* dan *monitoring*. Pengembang dapat mengirimkan data dari mikrokontroler ke komputer untuk mencatat perubahan suhu, memantau status sistem, dan mengamati respons aktuator secara *real-time*.

Timer adalah periferan yang menghitung siklus clock pada hardware dan juga memicu interrupt atau sebuah aksi pada interval yang telah ditentukan, yang dimana bertujuan untuk eksekusi tugas periodik seperti sampling sensor tanpa memblokir program utama. External interrupt juga dapat digunakan sebagai penanganan input dari

pengguna. Tombol fisik dikonfigurasi sebagai sumber external interrupt untuk mikrokontroler agar dapat langsung merespons saat tombol ditekan. Interrupt akan membuat sistem lebih responsif dan efisien karena prosesor tidak perlu terus-menerus memeriksa kondisi secara manual (*polling*). Dalam sistem kendali kipas berbasis suhu, timer interrupt inilah yang mengatur sampling sensor secara periodik, sementara external interrupt menangani input tombol pengatur setpoint.

Dalam pemrograman mikrokontroler, ada dua metode dalam memprogramnya, yaitu menggunakan HAL (*Hardware Abstraction Layer*) *library* atau mengakses register secara langsung yang difasilitasi oleh CMSIS (*Cortex Microcontroller Software Interface Standard*) *library*. Pemrograman berbasis register memberikan akses langsung ke periferal mikrokontroler melalui alamat memori, dan dalam basis ARM Cortex-M, pendekatan ini difasilitasi oleh CMSIS yang menyediakan definisi register dan struktur data standar untuk mengakses periferal secara langsung tanpa lapisan abstraksi tambahan. CMSIS memetakan register-register hardware periferal seperti GPIO, ADC, UART, dll., yang digunakan untuk mengontrol perangkat keras. Penggunaan pemrograman level register tentunya membutuhkan pemahaman bagaimana periferal bekerja pada tingkat paling dasar.

Berdasarkan latar belakang di atas, maka dalam tugas besar mata kuliah Mikroprosesor dan Antarmuka ini mengimplementasikan "Sistem Kendali Suhu Ruang". Proyek ini memiliki relevansi dengan aplikasi *thermal chamber*, yaitu ruangan tertutup dengan suhu yang terkontrol. Beberapa implementasi nyata telah diterapkan pada berbagai bidang diantaranya yakni ruang pengering hasil pertanian membutuhkan suhu stabil pada rentang 50-60°C^[2] juga sebuah inkubator penetasan telur yang memerlukan suhu terkontrol antara 36 sampai 38°C agar embrio berkembang dengan optimal^[3].

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, rumusan masalah pada proyek ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan komunikasi serial UART berbasis register untuk *logging data* suhu ke komputer?
2. Bagaimana mengkonfigurasi timer untuk menghasilkan sinyal PWM dan menjalankan *sampling* sensor secara periodik?
3. Bagaimana mengimplementasikan sistem *interrupt* yang mencakup *timer interrupt* untuk *sampling* dan *external interrupt* untuk input tombol?
4. Bagaimana mengintegrasikan periferal ADC, Timer, PWM, UART, dan *External Interrupt* dalam satu sistem kendali kipas berbasis suhu?

1.3 Tujuan

Umum:

Merancang dan mengimplementasikan sistem kendali suhu ruang berbasis mikrokontroler STM32F411 yang mampu mengatur kecepatan kipas secara otomatis berdasarkan pembacaan sensor suhu.

Khusus:

1. Mengimplementasikan komunikasi UART berbasis register untuk mengirimkan log data suhu dan status sistem ke komputer secara real-time.
2. Mengkonfigurasi timer untuk menghasilkan sinyal PWM pengendali kecepatan kipas dan mengatur *sampling* sensor secara periodik menggunakan timer *interrupt*. Mengimplementasikan *external interrupt* untuk menangani input tombol pengatur setpoint.
3. Mengimplementasikan seluruh konfigurasi periferal menggunakan akses register langsung melalui library CMSIS tanpa menggunakan HAL.
4. Mengintegrasikan periferal ADC, Timer, PWM, UART, dan *External Interrupt* menjadi satu sistem kendali suhu.

1.4 Batasan Masalah

Proyek ini memiliki beberapa batasan sebagai berikut:

1. Hardware:
 - a. Mikrokontroler yang digunakan adalah STM32F411CEU6 (*Black Pill*).
 - b. Sensor suhu menggunakan LM35 dengan rentang pengukuran 0-100°C.
 - c. Aktuator berupa kipas DC 12V yang dikendalikan melalui transistor TIP120.
 - d. Catu daya menggunakan adaptor 12V 3A dan modul MB102 untuk tegangan 5V dan 3.3V.
2. Software:
 - a. Pemrograman menggunakan bahasa C dengan library CMSIS.
 - b. Development environment menggunakan STM32CubeIDE.
 - c. Tidak menggunakan library HAL (*Hardware Abstraction Layer*).
3. Lingkup:
 - a. Sistem hanya mengendalikan satu kipas sebagai aktuator pendingin.
 - b. Tidak menggunakan elemen pemanas, hanya pendinginan melalui kipas.
 - c. Sistem tidak terhubung ke jaringan internet (*non-IoT*).

1.5 Manfaat

Proyek ini memberikan beberapa manfaat sebagai berikut:

1. Pembelajaran:
 - a. Meningkatkan pemahaman tentang arsitektur mikrokontroler ARM Cortex-M4 pada STM32F411.
 - b. Memenuhi capaian pembelajaran yang mencakup komunikasi serial, timer interrupt, external interrupt, dan pemrograman berbasis register.
2. Teknis:
 - a. Mengembangkan kemampuan pemrograman *embedded system* pada level register.
 - b. Melatih keterampilan debugging dan integrasi *multiple peripheral* dalam satu sistem.

- c. Memahami cara kerja ADC, Timer, PWM, UART, dan External Interrupt secara mendalam.
 - 3. Aplikatif:
 - a. Menghasilkan prototipe sistem kendali suhu yang dapat dikembangkan untuk aplikasi *thermal chamber* sederhana.
 - b. Menjadi dasar pengembangan sistem kontrol suhu yang lebih kompleks di masa yang akan datang.
-

BAB II: LANDASAN TEORI

2.1 Mikrokontroler STM32F411

STM32F411CEU6 merupakan seri mikrokontroler STM32 yang terpasang pada *Black Pill development board*. Mempunyai kinerja processor 32-bit RISC (*Reduced Instruction Set Computer*) yang beroperasi pada frekuensi maksimum 100MHz^[4]. Seri ini termasuk ke dalam tipe ARM Cortex-M4 *High-Performance*^[4]. STMicroelectronics mengembangkan mikrokontroler ini sebagai bagian dari lini *Dynamic Efficiency* yang mengutamakan efisiensi daya dan performa tinggi.

STM32F411CEU6 adalah anggota dari seri STM32F411xE yang memiliki memori Flash 512 KB untuk penyimpanan program dan data. Memori SRAM (*Static Random Access Memory*) yang berkapasitas 128 KB tersedia untuk operasi runtime^[4], yang dimana kapasitas ini cukup besar untuk aplikasi kendali suhu dengan logging data. Mikrokontroler ini dilengkapi dengan *ART Accelerator* (*Adaptive Real-Time Accelerator*) guna mengoptimalkan eksekusi program dari memori *Flash*. Dengan *ART Accelerator* aktif, CPU dapat mengakses Flash tanpa *wait state* pada frekuensi hingga 100 MHz^[4]. Ini setara dengan performa eksekusi dari RAM.

CPU dilengkapi dengan *Floating Point Unit* (FPU) presisi tunggal untuk mempercepat operasi matematika *floating-point* secara signifikan yang berguna untuk kalkulasi kontrol PID pada sistem kendali suhu. Selain itu, Cortex-M4 juga mendukung instruksi

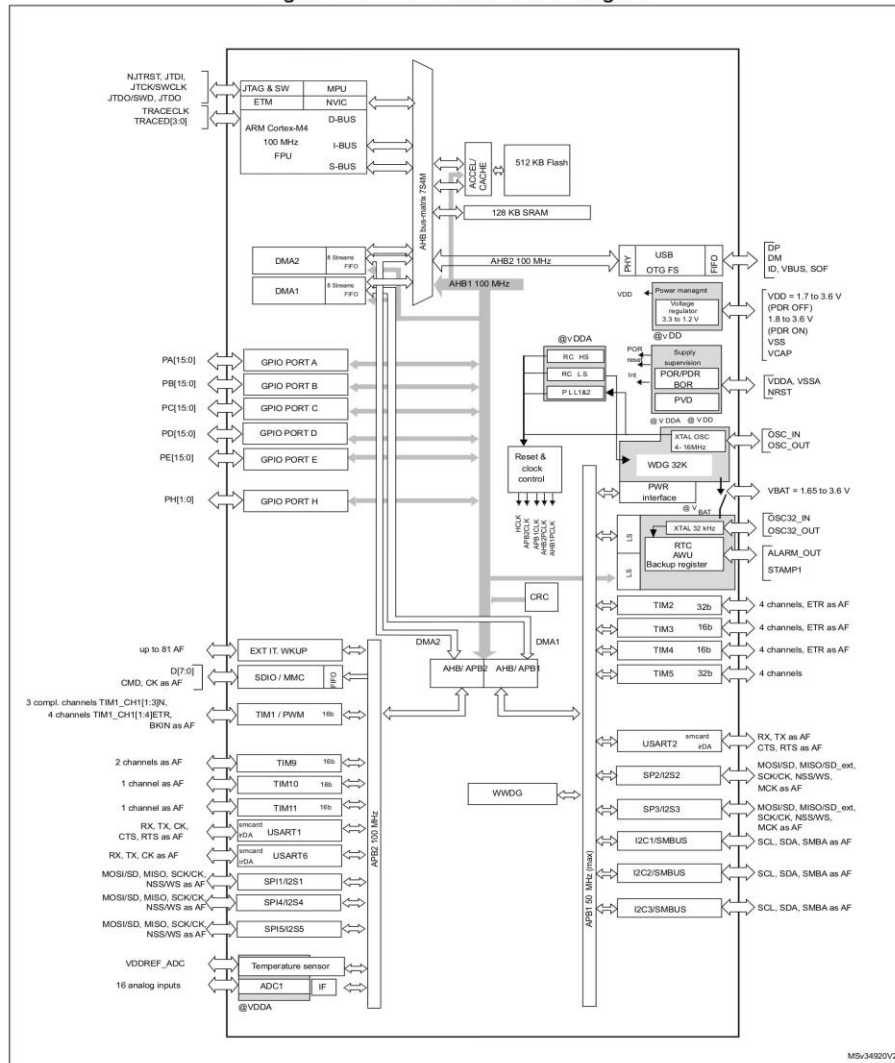
DSP (*Digital Signal Processing*). Fungsi dari DSP sendiri adalah untuk mengoptimalkan kinerja aplikasi yang membutuhkan manipulasi sinyal digital secara *real-time*, seperti suhu dan kontrol motor^[5].

Peripherals		STM32F411xC			STM32F411xE		
Flash memory in Kbytes		256			512		
SRAM in Kbytes	System	128					
Timers	General-purpose	7					
	Advanced-control	1					
Communication interfaces	SPI/ I ² S	5/5 (2 full duplex)					
	I ² C	3					
	USART	3					
	SDIO	1					
	USB OTG FS	1					
GPIOs		36	50	81	36	50	81
12-bit ADC		1					
Number of channels		10	16		10	16	
Maximum CPU frequency		100 MHz					
Operating voltage		1.7 to 3.6 V					
Operating temperatures		Ambient temperatures: - 40 to +85 °C / - 40 to + 105 °C/ - 40 to + 125 °C					
		Junction temperature: – 40 to + 130 °C					
Package		WLCSP49 UFQFPN48	LQFP64	UFBGA100 LQFP100	WLCSP49 UFQFPN48	LQFP64	UFBGA100 LQFP100

Tabel 1. Fitur STM32F411xC/xE
(Datasheet STM32F411xC/xE, (2017). Halaman 12)

Untuk bagian Bus Matrix dan Peripheral pada STM32F411xC/xE, Multi-AHB bus matrix menghubungkan semua master (CPU, DMA) dengan slave (Flash, RAM, peripheral). Arsitektur ini memastikan operasi simultan beberapa peripheral tanpa konflik. DMA1 dan DMA2 masing-masing memiliki 8 stream dengan FIFO dedicated.

Figure 3. STM32F411xC/xE block diagram

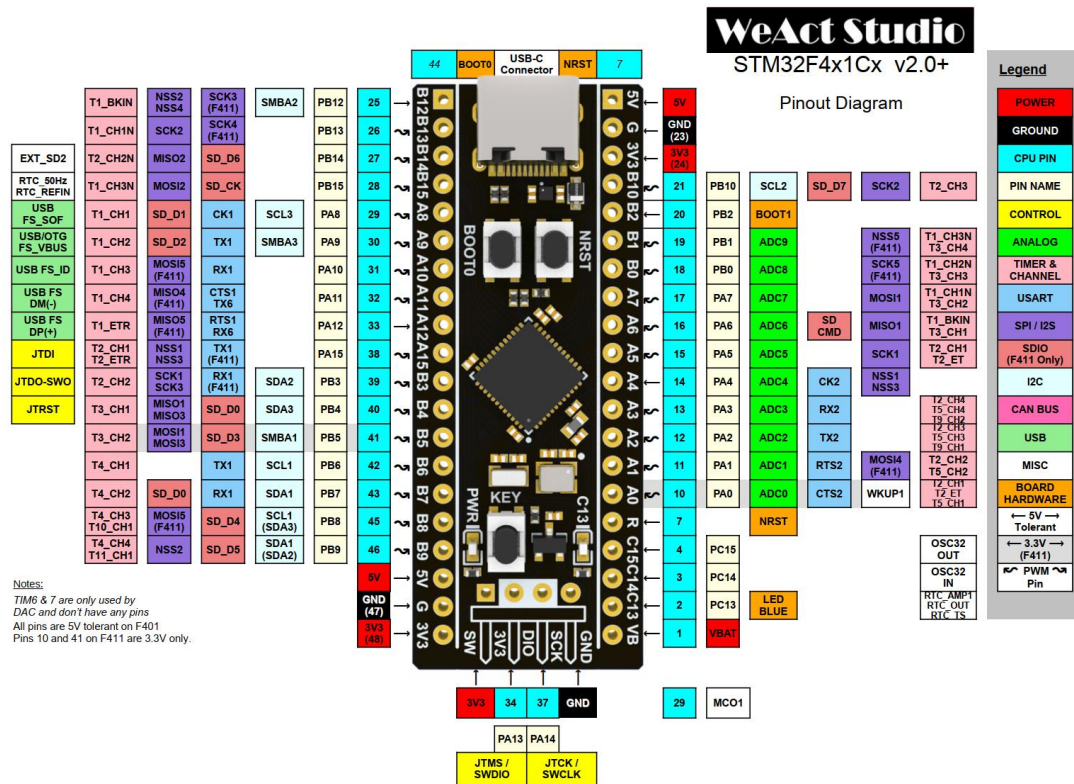


1. The timers connected to APB2 are clocked from TIMxCLK up to 100 MHz, while the timers connected to APB1 are clocked from TIMxCLK up to 100 MHz.

Gambar 2. Diagram blok STM32F411xC/xE
(Datasheet STM32F411xC/xE, (2017). Halaman 15)

Diagram blok di atas menunjukkan arsitektur internal STM32F411. Terlihat bagaimana CPU Cortex-M4 terhubung dengan memori Flash dan SRAM melalui bus matrix. Peripheral terdistribusi pada bus APB1 dan APB2 dengan bridge ke AHB.

Development board Black Pill menggunakan STM32F411CEU6 dalam paket UFQFPN48^[4]. Board versi V2.0+ dilengkapi tiga tombol: reset, BOOT0, dan user key^[5]. Kristal eksternal 25 MHz terpasang untuk HSE dan 32.768 kHz untuk LSE, keduanya menggunakan kristal metal shell untuk stabilitas yang lebih baik^[5]. Board menyediakan pin 5V dan 3.3V untuk catu daya^[5].



Gambar 2. Pinout Diagram Board Black Pill STM32F4x1Cx

(Sumber: WeAct Studio, 2020)

2.2 Serial Communication

2.2.1 Protokol UART

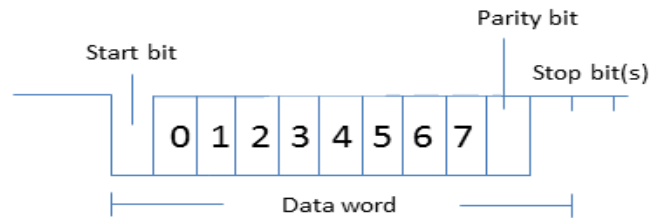
UART (*Universal Asynchronous Receiver/Transmitter*) adalah protokol komunikasi serial asinkron yang umum dipakai pada sistem tertanam untuk pertukaran data antar perangkat menggunakan minimal dua jalur, yaitu TX (*transmit*) dan RX (*receive*), tanpa sinyal clock terpisah. Karena bersifat asinkron, kedua perangkat yang berkomunikasi harus menyepakati parameter yang sama seperti baud rate, panjang data, paritas, dan stop bit agar data dapat diterima dengan benar. [deepbluembedded.com]

Pada keluarga STM32, periferal USART dapat beroperasi pada mode asinkron (UART) dan menyediakan parameter konfigurasi yang fleksibel seperti data length, parity, jumlah stop bit, serta pilihan oversampling 8 atau 16 untuk kompromi antara kecepatan dan ketahanan noise/ketidakakuratan clock. [st.com]

Komunikasi UART mengirimkan data dalam bentuk frame. Satu frame terdiri dari:

1. Start bit (1 bit): garis (*line*) ditarik LOW selama 1 periode bit untuk menandai awal frame sekaligus sinkronisasi penerima. [st.com]
2. Data bits (umumnya 7/8/9 bit): bit-bit data dikirim setelah start bit. Pada implementasi UART umumnya, urutan pengiriman data adalah LSB terlebih dahulu. [st.com], [fastbitlab.com]
3. Parity bit (opsional, 1 bit): digunakan untuk deteksi kesalahan sederhana (*odd/even*). Pada STM32, jika parity diaktifkan, bit parity berada setelah data (menjadi bit terakhir sebelum stop). [st.com]
4. Stop bit (1 / 1.5 / 2 bit): garis (*line*) ditarik HIGH untuk menandai akhir frame.

Contoh konfigurasi frame yang paling umum dipakai untuk logging ke PC adalah 8N1, karena sederhana dan kompatibel dengan mayoritas terminal serial. [st.com]



Gambar 3. Pengiriman data serial tak sinkron
(Sumber: docs.tibbo.com, 2022)

Gambar 3 memperlihatkan pola sinyal komunikasi serial port menggunakan format 8N1, yang berarti terdiri dari 8 bit data, tanpa bit paritas, dan 1 bit stop. Saat tidak ada pengiriman data (kondisi idle), jalur serial port berada pada kondisi *Mark* atau logika 1. Ketika proses transmisi dimulai, diawali dengan bit start yang memiliki nilai logika 0. Selanjutnya, setiap bit dalam karakter data dikirimkan secara berurutan satu demi satu. Pengiriman dilakukan dengan urutan bit dari LSB (Least Significant Bit) atau bit terendah lebih dulu, kemudian dilanjutkan dengan MSB (Most Significant Bit) atau bit tertinggi. Sebagai penanda akhir transmisi, sebuah bit stop dengan logika 1 ditambahkan pada bagian akhir paket data.

Berikut adalah register-register penting USART:

1. USART_CR1 (*Control Register 1*)

Register kontrol utama untuk mengatur mode operasi USART. Bit-bit diantaranya meliputi:

- a. UE (bit 13): *USART Enable* untuk mengaktifkan peripheral USART
- b. M (bit 12): *Word Length* untuk menentukan panjang data (0 = 8 bit, 1 = 9 bit)
- c. PCE (bit 10): *Parity Control Enable* untuk mengaktifkan kontrol parity
- d. PS (bit 9): *Parity Selection* (0 = even parity, 1 = odd parity)
- e. TE (bit 3): *Transmitter Enable* untuk mengaktifkan transmitter
- f. RE (bit 2): *Receiver Enable* untuk mengaktifkan receiver

- g. OVER8 (bit 15): *Oversampling mode* (0 = oversampling by 16, 1 = oversampling by 8)

2. USART_SR (*Status Register*)

Register status yang menyimpan flag kondisi USART:

- a. TXE (bit 7): *Transmit Data Register Empty* yang menandakan buffer transmit kosong dan siap menerima data baru
- b. TC (bit 6): *Transmission Complete* yang menandakan transmisi frame terakhir telah selesai
- c. RXNE (bit 5): *Read Data Register Not Empty* yang menandakan ada data baru yang diterima
- d. IDLE (bit 4): *Idle Line Detected* mendeteksi jalur idle
- e. ORE (bit 3): *Overrun Error* terjadi ketika data baru diterima sebelum data sebelumnya dibaca
- f. NF (bit 2): *Noise Flag* yang mendeteksi noise pada frame yang diterima
- g. FE (bit 1): *Framing Error* untuk kesalahan deteksi stop bit
- h. PE (bit 0): *Parity Error* kesalahan parity

3. USART_DR (*Data Register*)

Register 9-bit yang berfungsi ganda (*dual-function*):

- a. TDR (*Transmit Data Register*): Buffer untuk data yang akan ditransmisikan
- b. RDR (*Receive Data Register*): Buffer untuk data yang diterima

Operasi write ke USART_DR akan menulis ke TDR, sedangkan operasi read akan membaca dari RDR.

4. USART_BRR (*Baud Rate Register*)

Register untuk mengkonfigurasi baud rate dengan struktur:

- a. DIV_Mantissa[11:0] (bits 15:4): Bagian mantissa dari divisor baud rate
- b. DIV_Fraction[3:0] (bits 3:0): Bagian fraction dari divisor baud rate

Baud rate adalah kecepatan transmisi data yang dinyatakan dalam satuan bits per second (bps). STM32F411 menggunakan fractional baud rate generator untuk menghasilkan baud rate yang akurat. Untuk mode asinkron standar (USART), formula baud rate berbeda tergantung pada konfigurasi oversampling:

1. Oversampling by 16 (OVER8 = 0):

$$\text{Baud Rate} = \frac{f_{PCLK}}{16 \times \text{USARTDIV}}$$

2. Oversampling by 8 (OVER8 = 1):

$$\text{Baud Rate} = \frac{f_{PCLK}}{8 \times \text{USARTDIV}}$$

Dari formula di atas, nilai USARTDIV dapat dihitung dengan membalik persamaan:

- Untuk oversampling by 16:

$$\text{USARTDIV} = \frac{f_{PCLK}}{16 \times \text{Baud Rate}}$$

- Untuk oversampling by 8:

$$\text{USARTDIV} = \frac{f_{PCLK}}{8 \times \text{Baud Rate}}$$

Dalam sistem kendali suhu ruang, UART dikonfigurasi dengan spesifikasi berikut:

- Baud rate: 9600 bps

- Data bits: 8 bit
- Parity: None (no parity)
- Stop bits: 1 bit
- Flow control: None

Konfigurasi ini dipilih karena: (1) baud rate 9600 bps memberikan error yang sangat kecil dan kompatibilitas tinggi dengan berbagai terminal, (2) format 8N1 merupakan standar yang paling umum digunakan, dan (3) tidak memerlukan *flow control* karena rate transmisi data suhu relatif rendah. Nantinya data yang dikirim melalui UART berupa string ASCII yang berisi informasi:

- Timestamp atau counter
- Nilai suhu terbaca dari sensor LM35
- Status sistem (PWM *duty cycle* kipas, *setpoint*, dll)
- Format: "Temp: 25.5C, PWM: 50%, Setpoint: 30Cm"

2.3 Timer dan Counter

Timer pada mikrokontroler STM32F411 adalah periferal yang menghitung pulsa clock secara hardware untuk menghasilkan time base yang presisi. STM32F411 memiliki 7 timer general-purpose (TIM2-TIM5, TIM9-TIM11) dan 1 *advanced-control timer* (TIM1). Setiap timer memiliki counter register yang bertambah atau berkurang setiap pulsa clock sesuai konfigurasi. Timer dapat menghasilkan interrupt, mengontrol PWM, atau melakukan *input capture* dari CPU, sehingga tidak membebani program utama.

Timer pada STM32 bekerja dengan prinsip *counting* berbasis *clock source* yang telah dikonfigurasi. *Clock source timer* berasal dari bus APB (*Advanced Peripheral Bus*), dimana TIM2-TIM5 terhubung ke APB1 sedangkan TIM1, TIM9-TIM11 terhubung ke APB2. Jika prescaler APB bernilai 1, maka timer clock sama dengan bus clock. Namun jika prescaler APB lebih dari 1, maka timer clock otomatis dikalikan 2 oleh hardware untuk mempertahankan resolusi timing.

STM32 timer dapat beroperasi dalam berbagai mode diantaranya:

1. Counter Mode

Timer berfungsi sebagai pencacah murni yang dapat menghitung naik (*up-counting*), turun (*down-counting*), atau center-aligned. Mode ini digunakan untuk time base generation atau delay presisi.

- a. Up-counting: Counter bertambah dari 0 hingga ARR, kemudian reset ke 0
- b. Down-counting: Counter berkurang dari ARR hingga 0, kemudian reload ARR
- c. Center-aligned: Counter naik ke ARR lalu turun ke 0 secara bergantian

Pada sistem kendali suhu ini, digunakan *up-counting mode* untuk sampling sensor secara periodik setiap 500 ms.

2. PWM (*Pulse Width Modulation*) Mode

Timer dapat menghasilkan sinyal PWM dengan *duty cycle* yang dapat diatur melalui *Compare Register* (CCR). PWM dihasilkan dengan membandingkan nilai counter dengan CCR:

- a. Ketika $CNT < CCR \rightarrow \text{Output HIGH}$
- b. Ketika $CNT \geq CCR \rightarrow \text{Output LOW}$

Duty cycle dapat dihitung:

$$\text{Duty Cycle (\%)} = \frac{CCR}{ARR + 1} \times 100\%$$

Frekuensi PWM:

$$f_{PWM} = \frac{f_{counter}}{ARR + 1}$$

3. *Input Capture Mode*

Mode ini menangkap nilai counter saat terjadi edge (*rising/falling*) pada input

pin. Berguna untuk mengukur frekuensi, periode, atau pulse width dari sinyal eksternal.

2.4 Interrupt System

- NVIC (Nested Vectored Interrupt Controller)
- External Interrupt (EXTI)
- Interrupt priority
- Register: EXTI_IMR, EXTI_RTSTR, EXTI_FTSR, NVIC_ISER

2.5 GPIO Configuration

- Input modes: Floating, Pull-up/down, Analog
- Output modes: Push-pull, Open-drain
- Register: GPIOx_CRL, GPIOx_CRH, GPIOx_IDR, GPIOx_ODR

2.6 ADC

- Conversion modes
- Sampling time
- Register: ADC_CR1, ADC_CR2, ADC_SQR, ADC_DR

BAB III: REQUIREMENT DAN SPESIFIKASI SISTEM

3.1 System Requirements

Sistem Kendali Suhu Ruang berbasis STM32F411 bertujuan untuk mengatur suhu ruangan secara otomatis dengan mengontrol kecepatan kipas DC berdasarkan pembacaan sensor suhu real-time. Sistem dirancang sebagai prototipe thermal chamber sederhana yang dapat mempertahankan suhu pada setpoint tertentu. Implementasi ini memenuhi capaian pembelajaran CLO3 melalui penerapan komunikasi serial UART,

timer interrupt untuk periodic sampling, dan external interrupt untuk input pengguna, dengan konfigurasi berbasis register menggunakan library CMSIS.

Persyaratan utama sistem:

1. Input Handling

Sistem menerima input dari sensor suhu LM35 yang dibaca melalui ADC 12-bit dengan sampling periodik 500 ms menggunakan timer interrupt (TIM2). Response time pembacaan sensor maksimal 10 ms. Tombol *push button* dikonfigurasi dengan *external interrupt* (EXTI) untuk mengatur *setpoint* suhu dengan *response time* maksimal 50 ms. Tombol memiliki *debouncing software* untuk mencegah *false trigger* dan setiap penekanan meningkatkan *setpoint* 2°C dalam rentang 24-36°C (*cycling*).

2. Output/Display

Kipas DC 12V dikontrol melalui PWM 1 kHz (TIM3 Channel 1) dengan *duty cycle* 0-100% yang disesuaikan berdasarkan error suhu. Algoritma kontrol menggunakan *proportional control* sederhana dengan *dead-band* $\pm 1^{\circ}\text{C}$ dari *setpoint*. LED indikator menunjukkan status sistem: hijau (stabil), kuning (*cooling*), merah (*overheat*). *Update rate* untuk PWM dan LED adalah setiap 500 ms sesuai dengan *periodic sampling cycle*.

3. Communication

Komunikasi serial menggunakan UART (USART2) dengan konfigurasi 115200 baud, 8N1 pada pin PA2 (TX) dan PA3 (RX). Sistem mengirimkan *data logging* setiap 500 ms dalam format CSV berisi timestamp, suhu aktual, *setpoint*, *duty cycle* PWM, dan status sistem. Data ditransmisikan ke komputer untuk monitoring *real-time* dan debugging. Implementasi menggunakan *interrupt-driven* untuk mencegah *blocking* pada main loop.

3.2 Functional Requirements

ID	Requirement	Priority
FR-01	Sistem harus mampu membaca suhu dari sensor LM35 melalui ADC dengan resolusi 12-bit dan akurasi $\pm 1^{\circ}\text{C}$ pada rentang 0-100 $^{\circ}\text{C}$	High
FR-02	Sistem harus melakukan sampling sensor suhu secara periodik setiap 500 ms menggunakan timer interrupt (TIM2) tanpa blocking program utama	High
FR-03	Sistem harus mengkonversi nilai ADC menjadi suhu dalam satuan Celsius	High
FR-04	Sistem harus dapat menerima input tombol untuk mengatur setpoint suhu menggunakan external interrupt (EXTI)	High
FR-05	Duty cycle PWM harus dapat diatur dari 0% (kipas OFF) hingga 100% (kipas maksimal) berdasarkan selisih suhu aktual dengan setpoint	High
FR-06	Sistem harus mengirimkan data logging melalui UART (115200 baud, 8N1) setiap 500 ms dalam format CSV: "Time(ms), Temp($^{\circ}\text{C}$), Setpoint($^{\circ}\text{C}$), PWM(%), Status"	High
FR-07	Sistem harus menggunakan USART2 dengan pin PA2 untuk TX dan PA3 untuk RX sebagai interface komunikasi serial	High
FR-08	Sistem harus menampilkan status operasi melalui LED indikator: LED hijau (suhu stabil dalam $\pm 1^{\circ}\text{C}$ setpoint), LED kuning (kipas aktif cooling), LED merah (overheat $> 5^{\circ}\text{C}$ dari setpoint)	Medium
FR-09	Sistem harus mengupdate status LED sesuai dengan kondisi suhu aktual	Medium
FR-10	Sistem harus mengimplementasikan safety override: jika suhu $> 50^{\circ}\text{C}$ maka kipas otomatis set ke 100% duty cycle	Medium
FR-11	Semua konfigurasi peripheral (GPIO, ADC, Timer, UART, EXTI) harus dilakukan menggunakan direct register access melalui CMSIS library tanpa HAL	High

ID	Requirement	Priority
FR-12	Interrupt priority harus dikonfigurasi: EXTI (highest), TIM2 (medium), USART (lowest) untuk mencegah konflik interrupt	High
FR-13	Sistem harus menggunakan setpoint default 26°C setelah power-on atau reset	Low

3.3 System Specifications

3.3.1 Hardware Specifications

Komponen Utama:

Komponen	Fungsi	Keterangan
STM32F411CEU6	Mikrokontroler	ARM Cortex-M4, USB Type=C
LM35 Sensor Module	Sensor suhu	Analog, Langsung ADC
Kipas DC 12V 8cm	Aktuator	Fan PC
TIP120 / 2N2222	Driver Kipas	Transistor Switching
Resistor 1K	Base Resistor	Pembatas arus ke transistor
ST-LINK V2	Programmer/Debugger	Flash Firmware ke STM32
MB102 Power Supply	Regulator 5V/3.3V	Power untuk breadboard
Adaptor 12V 3A	Power Supply utama	Input ke MB102 + kipas
Breadboard	Prototyping	400P + 830P
Kabel Jumper	Koneksi	Dupont + tunggal

Pin Mapping:

Pin STM32	Connect To	Fungsi	Peripheral
PB6	LCD SCL	I2C Clock	I2C1_SCL
PB7	LCD SDA	I2C Data	I2C1_SDA
PA0	Button 1	Input	EXTI0
[tambahkan	[device]	[fungsi]	[peripheral]

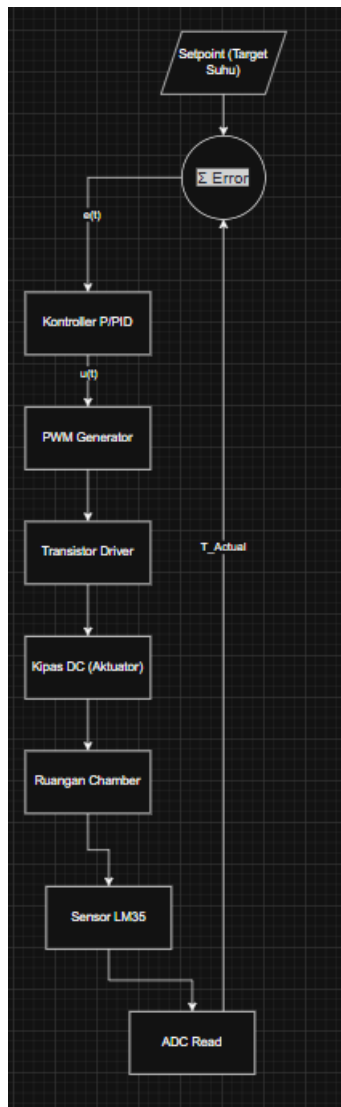
3.3.2 Software Specifications

- Development Tool: VSCode + PlatformIO + CMISIS + Arduino
- Programming: C language, register-based
- Clock: 16 Mhz (HSI) atau 100MHz (HSE + PLL)
- Communication: I2C (100kHz) / UART (9600 baud)
- Timer: 1MHz (1 μ s resolution)

BAB IV: PERANCANGAN SISTEM

4.1 Arsitektur Sistem

Block Diagram:



\\

4.2 Desain Hardware4.2.1 Schematic Diagram

Minimal berisi:

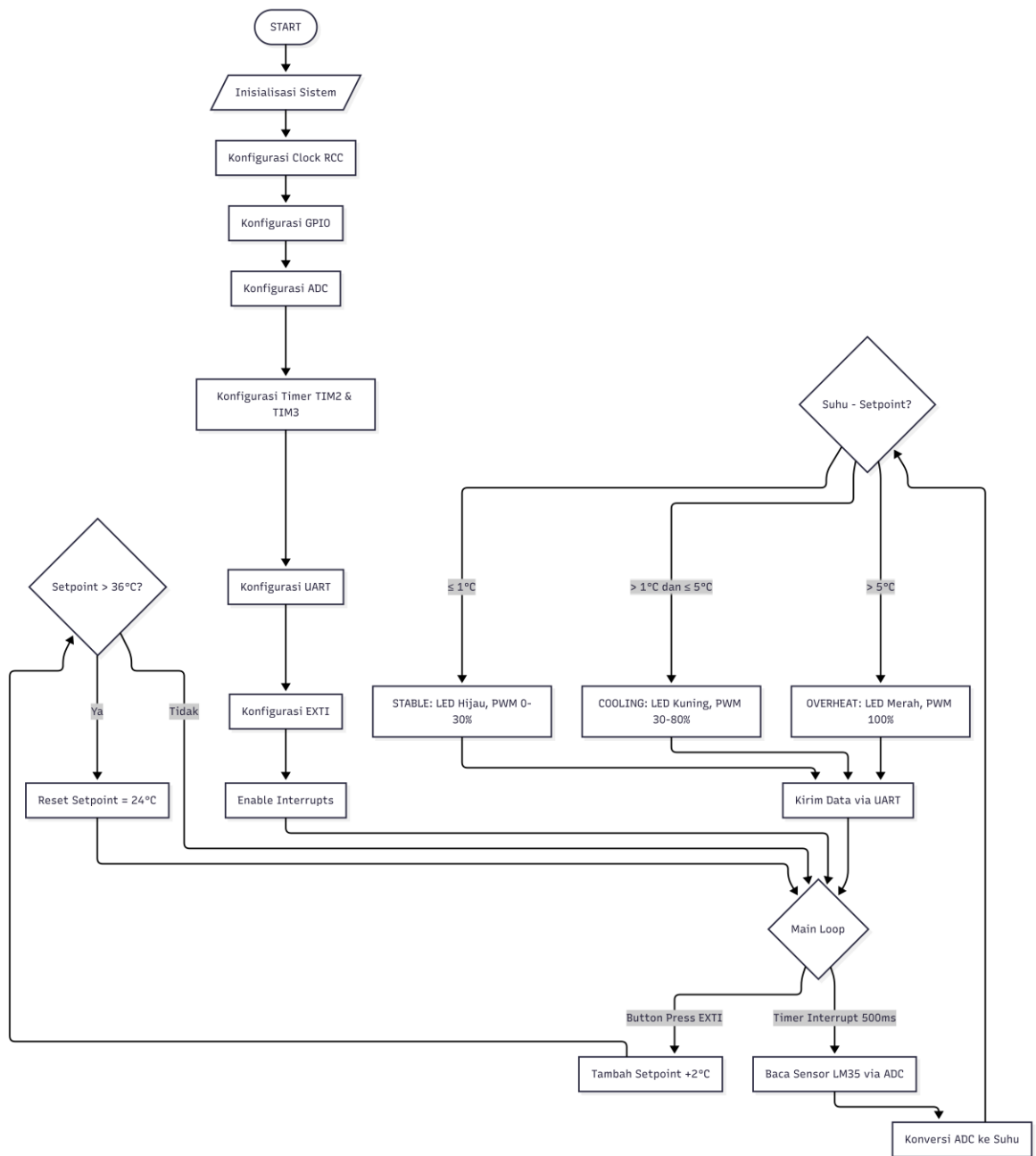
- STM32 pin connections
- Pull-up/down resistors
- Current limiting resistors
- Decoupling capacitors

4.2.2 Bill of Materials (BOM)

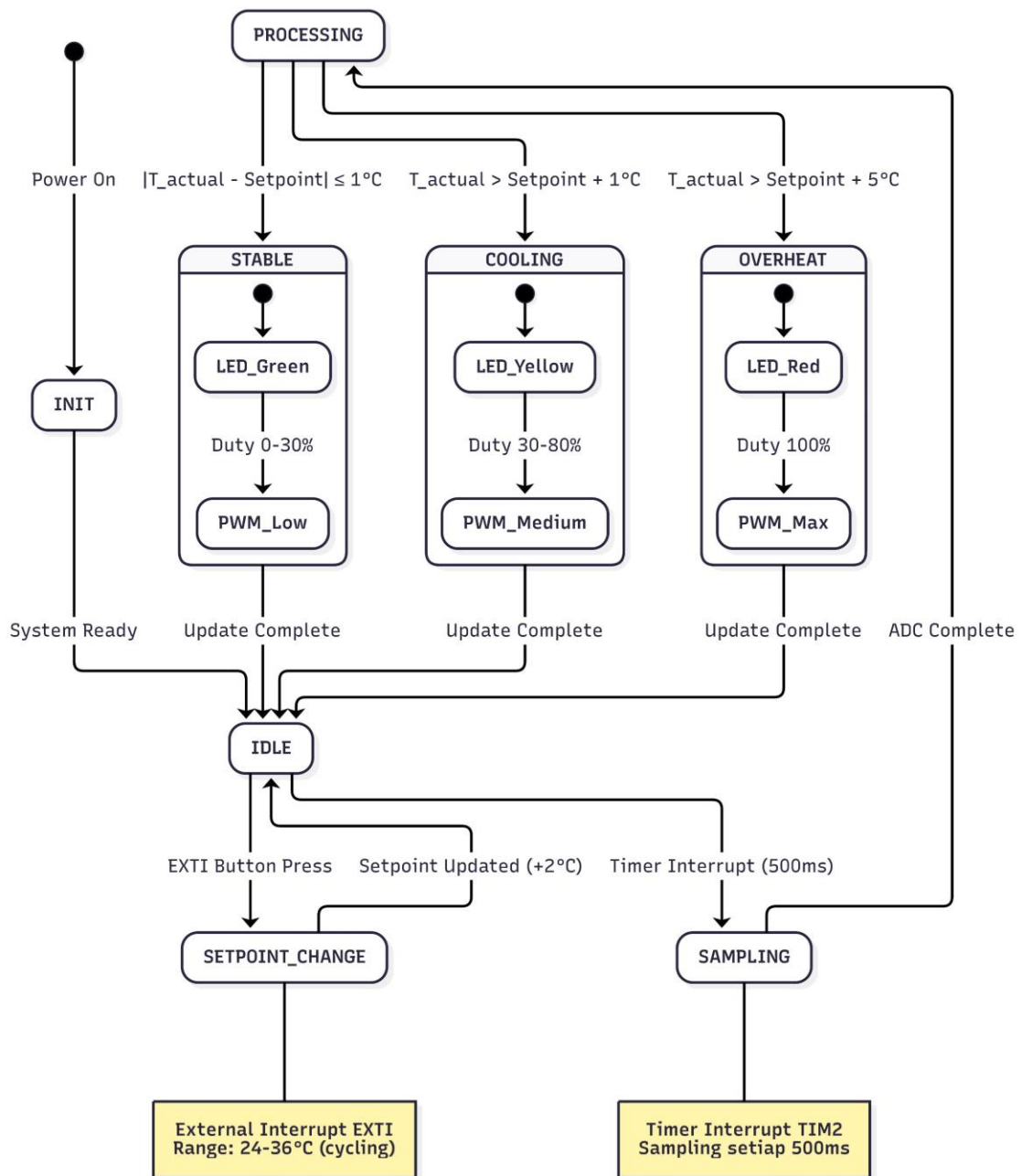
No	Item	Qty	Harga
1	STM32F411CEU6	1	Rp75.000
2	ST-LINK V2 Programmer	1	Rp21.500
3	LM35 Sensor Module	1	Rp15.520
4	Kipas 8cm 12V	1	Rp19.400
5	TIP120 Transistor	2	Rp 5.820
6	2N2222 Transistor (10pcs)	1	Rp 2.000
7	Resistor 1K 1/4W	10	Rp 1.940
8	MB102 Power Supply 3.3V/5V	1	Rp 8.500
9	Adaptor 12V 3A Samsung	1	Rp 36.500
10	Breadboard 400P	1	Rp 6.500
11	Breadboard 830P	1	Rp 8.900
12	Paket Kabel Jumper Dupont 15CM	1	Rp 36.300
13	Kabel Jumper Tunggal 350pcs	1	Rp 30.000
	Total		Rp 267.880

4.3 Desain Software

4.3.1 Flowchart Sistem



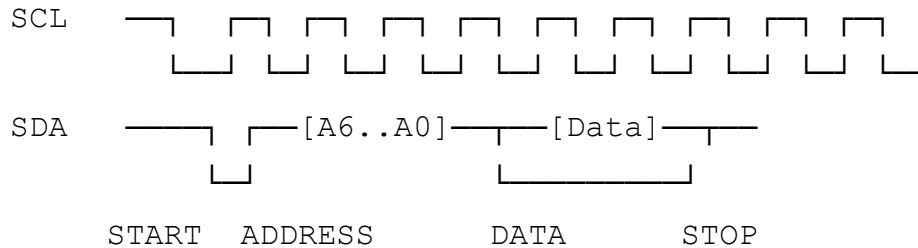
4.3.2 State Machine Diagram



4.3.3 Timing Diagram

[Gambar timing diagram untuk I2C/UART/interrupt]

Contoh I2C Timing:



4.4 Desain Subsystem

4.4.1 Subsystem Input (EXTI & GPIO)

Design Debouncing:

Waktu stabil button setelah bounce:

State Machine:

States: [Released, Pressed, Debouncing]

Transitions:

- Released → Debouncing: Jika interrupt terjadi
- Debouncing → Pressed: Jika state stabil selama t_{stable}
- Pressed → Released: Jika button dilepas

4.4.2 Subsystem Timing (Timer Dynamics)

Konfigurasi Timer Counter:

- Clock source: $APB1 \times 1 = 16\text{MHz}$
- Prescaler: $PSC = 16 - 1 = 15$
- $f_{timer} = 16\text{MHz} / 16 = 1\text{MHz}$

- Resolution: 1 μ s per count

Auto-Reload Register (ARR):

Untuk interrupt setiap 1ms:

$$ARR = (1ms \times 1MHz) - 1 = 999$$

4.4.3 Subsystem Display (I2C Communication)

I2C Clock Calculation:

I2C Clock Calculation:

$$CCR = f_APB1 / (2 \times f_I2C)$$

frekuensi i2c =

State Machine:

States: [Idle, Start, Address, Write_Data, Stop]

Transitions:

- Idle \rightarrow Start: Ketika ada data untuk dikirim
- Start \rightarrow Address: Setelah START condition
- Address \rightarrow Write_Data: Jika ACK received
- Write_Data \rightarrow Stop: Setelah semua data terkirim

4.4.4 Subsystem ADC (Optional)

Konversi ADC:

$$V_digital = (ADC_value / 4095) \times V_ref$$

Sampling Time:

$$t_sample = (SMP + 12.5) / f_ADC$$

Dengan:

- SMP = 55.5 cycles (konfigurasi)

- $f_{\text{ADC}} = 12\text{MHz}$ (maksimum)
- $t_{\text{sample}} = 5.625\mu\text{s}$

BAB V: IMPLEMENTASI

5.1 Implementasi Hardware

[Foto hasil rakitan hardware dari berbagai angle]
 [Foto close-up koneksi penting]
 [Foto testing dengan multimeter/oscilloscope/logic analyzer] optional

5.2 Implementasi Software (Register-Based)

5.2.1 RCC Configuration

Kode:

```
void RCC_init(void) {

    // [Lengkapi dengan penjelasan setiap register]
}
```

Penjelasan Register:

Register	Bit	Value	Fungsi
RCC->x		1	
RCC->x		1 (read)	
[dst]	[bit]	[value]	[fungsi]

5.2.2 GPIO Configuration

Kode:

```
void GPIO_Config(void) {  
    // [Kode konfigurasi GPIO]  
}
```

Penjelasan Register: [Tabel penjelasan]

5.2.3 I2C Configuration

Kode:

```
void I2C_Config(void) {  
    // [Kode konfigurasi I2C]  
}
```

Penjelasan Register: [Tabel penjelasan]

5.2.4 Timer Configuration

Kode:

```
void Timer_Config(void) {  
    // [Kode konfigurasi Timer]  
}
```

Penjelasan Register: [Tabel penjelasan]

5.2.5 External Interrupt Configuration

Kode:

```
void EXTI_Config(void) {  
    // [Kode konfigurasi EXTI]  
}
```

Penjelasan Register: [Tabel penjelasan]

5.2.6 Interrupt Service Routines

Kode:

```
void TIM2_IRQHandler(void) {  
    // [Kode ISR]  
}  
  
void EXTI0_IRQHandler(void) {  
    // [Kode ISR]  
}
```

Analisis ISR:

- Execution time: [waktu]
- Priority: [level priority]
- Critical section: [bagian critical]

5.3 Pembagian Tugas Kelompok

Anggota	Tugas	Status
Nama 1	Hardware design & wiring	✓
Nama 2	I2C implementation	✓
Nama 3	Timer & interrupt	✓
Nama 4	Integration & testing	✓
Nama 5	Documentation	✓

BAB VI: PENGUJIAN DAN ANALISIS

6.1 Metodologi Pengujian

- Unit Testing: [jelaskan]
- Integration Testing: [jelaskan]
- System Testing: [jelaskan]
- Tools: Logic analyzer, oscilloscope, multimeter

6.2 Pengujian Sistem Keseluruhan

Skenario Pengujian:

1. [Skenario 1]
2. [Skenario 2]
3. [dst]

Hasil: [Tabel hasil]

Dokumentasi:

[Foto/video demonstrasi sistem berfungsi]

6.6 Troubleshooting

Problem	Cause	Solution
I2C not working	[penyebab]	[solusi]
[problem]	[cause]	[solution]

BAB VII: KESIMPULAN DAN SARAN

7.1 Kesimpulan

1. [Kesimpulan terkait CLO3: Serial Communication]
2. [Kesimpulan terkait CLO3: Timer & Interrupt]
3. [Kesimpulan terkait register-level programming]
4. [Kesimpulan hasil pengujian]

7.2 Saran

1. [Saran pengembangan sistem]
2. [Saran pembelajaran]
3. [Saran penelitian lanjutan]

AI Editing Statement

Saya menyatakan bahwa saya menggunakan alat kecerdasan buatan (AI) hanya untuk keperluan editing, termasuk perbaikan tata bahasa, penyusunan kalimat, konsistensi istilah, dan peningkatan keterbacaan dokumen ini.

Seluruh ide, analisis, perhitungan, desain, dan konten substantif merupakan hasil pekerjaan saya sendiri. AI tidak digunakan untuk menghasilkan konten inti, menyelesaikan tugas teknis, atau membuat analisis secara otomatis.

Sebagai bentuk transparansi, berikut adalah daftar penggunaan AI beserta nama alat AI yang digunakan:

No.	Bagian yang Diedit	Jenis Bantuan AI	Nama AI yang Digunakan
1	Latar Belakang	Revisi konsistensi tata bahasa secara teknis	Claude
2	(isi sesuai penggunaan)	(jenis bantuan)	(nama AI)
3	(isi sesuai penggunaan)	(jenis bantuan)	(nama AI)

Nama Mahasiswa: Firdaus Arif Ramadhani

NIM: 101032300131

Tanggal: 14 Desember 2025

DAFTAR PUSTAKA

- [1] Sigit Prakosa Adhi Nugraha, None Lilo Sunuharjo, & None Muhammad 'Atiq. (2024). *Komunikasi Arduino I2C, SPI dan UART*. 2(4), 80–85.
<https://doi.org/10.62951/switch.v2i4.187>
- [2] Suryadi, Sukmawaty, Mahardhian, G., & Putra, D. (2017). SCALE UP DAN UJI TEKNIS ALAT PENGERING TIPE FLUIDIZED BED Scale Up and Technical Test of Fluidized Bed Dryer. *Jurnal Ilmiah Rekayasa Pertanian Dan Biosistem*, 5(2).
<https://media.neliti.com/media/publications/255847-scale-up-dan-uji-teknis-alat-pengering-t-8ed95989.pdf>
- [3] Karsid Karsid. (2024). Sistem Kontrol Suhu dan Kelembaban Udara pada Alat Penetas Telur Menggunakan Arduino Uno. *Elektriese*, 14(01), 25–32.
<https://doi.org/10.47709/elektriese.v14i01.3734>
- [4] STMicroelectronics. (2017). Datasheet STM32F411xC STM32F411xE.
<https://www.st.com/resource/en/datasheet/stm32f411ce.pdf>
- [5] Roza Dastres, & Mohsen Soori. (2021). A Review in Advanced Digital Signal Processing Systems. *OPAL (Open@LaTrobe) (La Trobe University)*.
<https://doi.org/10.6084/m9.figshare.14330879.v1>
- [6] WeAct Studio. (2020). MiniSTM32F4x1 Product Literature. GitHub.
<https://github.com/WeActStudio/WeActStudio.MiniSTM32F4x1>
- [x] [Tambahkan referensi buku/paper/tutorial yang digunakan]
-

LAMPIRAN

LAMPIRAN A: Full Source Code

// [Lampirkan full source code dengan comments jangan
screenshot]

LAMPIRAN B: Datasheet Excerpts

[Screenshot bagian penting dari datasheet]

LAMPIRAN C: Testing Results

[Data lengkap hasil pengujian]

LAMPIRAN D: Video Demonstration

Link: [URL YouTube/Drive]

QR Code: [QR code ke video]

CHECKLIST KELENGKAPAN LAPORAN

Content: 10%

- ☐ Abstrak
- ☐ BAB I-VII lengkap
- ☐ Minimum 20 halaman (tidak termasuk lampiran)
- ☐ Semua gambar, table diberi caption dan nomor

Technical: 40%

- ☐ Register configuration explained dengan tabel
- ☐ Schematic diagram, jelas dan lengkap
- ☐ Flowchart dan state machine included

- ☐ Photo dokumentasi hardware
- ☐ Video demonstration

CLO3 Requirements: 40 %

- ☐ Penerapan Serial communication (I2C/UART/SPI)
- ☐ Penerapan Timer configuration explained
- ☐ Interrupt system explained
- ☐ Register-level code documented
- ☐ Integration testing performed

Formatting: 10%

- ☒ Font: Times New Roman 12pt
- ☒ Line spacing: 1.5
- ☒ Margin: 4-3-3-3 cm
- ☐ Page numbers