Embedded System

Rahima Tartila

Tartila Press
ISBN 978-634-05-2943-2 · 2026

Cuplikan akademik untuk indeks ilmiah. Memuat abstrak, kata pengantar, daftar isi, Bab 1, daftar pustaka, dan biodata penulis. Buku lengkap: www.tartilapress.com/buku/embedded-system

Abstrak

Teknologi modern telah menghadirkan berbagai perangkat pintar yang mampu bekerja secara otomatis, mulai dari mesin cuci, kendaraan, alat kesehatan, sistem industri, hingga perangkat Internet of Things (IoT). Di balik seluruh teknologi tersebut terdapat sebuah komponen penting yang menjadi "otak" pengendali sistem, yaitu embedded system atau sistem tertanam. Buku Embedded System: Memahami, Merancang, dan Membangun Sistem Tertanam untuk Dunia Nyata disusun sebagai panduan komprehensif bagi mahasiswa, praktisi, maupun pembelajar mandiri yang ingin memahami konsep, arsitektur, serta implementasi embedded system secara sistematis dan aplikatif.

Materi disajikan secara bertahap mulai dari pengenalan dasar sistem tertanam, arsitektur perangkat keras, bahasa pemrograman C dan Assembly, manajemen interrupt, komunikasi serial, antarmuka sensor dan aktuator, hingga teknologi komunikasi nirkabel. Pembahasan kemudian dilanjutkan pada topik yang lebih lanjut seperti Real-Time Operating System (RTOS), sinkronisasi antar-task, manajemen memori dan waktu, pengelolaan daya, teknik pengujian dan debugging, keamanan sistem tertanam, Embedded Linux, serta implementasi Internet of Things (IoT). Untuk memperkuat pemahaman, buku ini juga dilengkapi berbagai studi kasus dan proyek akhir yang merepresentasikan kebutuhan dunia industri modern.

Dengan bahasa yang mudah dipahami, didukung tabel, ilustrasi, diagram, serta contoh kode program yang relevan, buku ini tidak hanya menjelaskan teori, tetapi juga memberikan gambaran nyata mengenai proses pengembangan embedded system dari tahap perancangan hingga implementasi. Setelah mempelajari buku ini, pembaca diharapkan mampu memahami prinsip-prinsip dasar sistem tertanam, memilih teknologi yang sesuai dengan kebutuhan aplikasi, serta mengembangkan solusi berbasis embedded system secara lebih terstruktur dan profesional. Buku ini menjadi referensi yang tepat bagi mahasiswa Teknik Komputer, Teknik Elektro, Sistem Komputer, Informatika, maupun siapa saja yang ingin memasuki dunia pengembangan perangkat cerdas yang saat ini menjadi fondasi berbagai inovasi teknologi di era digital.

Kata kunci

sistem tertanam; embedded system; RTOS; Embedded Linux; Internet of Things

Cara mengutip

Tartila, R. (2026). Embedded System. Tartila Press. ISBN 978-634-05-2943-2.

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya buku **Embedded System** ini dapat diselesaikan dengan baik.

Buku ini disusun dengan tujuan untuk belajar sekaligus berbagi ilmu kepada siapa saja yang ingin mengenal dan mendalami dunia *embedded system*. Sasaran utama buku ini adalah mahasiswa, khususnya dari bidang Teknik Komputer, Teknik Elektro, Sistem Komputer, Informatika, dan bidang terkait lainnya. Namun demikian, buku ini juga ditujukan bagi masyarakat umum yang memiliki minat untuk belajar secara mandiri dan ingin memahami teknologi yang menjadi fondasi berbagai perangkat elektronik modern.

Di era yang semakin maju seperti saat ini, terdapat banyak cara untuk memperoleh pengetahuan. Pembelajaran dapat dilakukan melalui jalur formal seperti perkuliahan, pelatihan, kursus, maupun *bootcamp*. Selain itu, akses informasi juga semakin terbuka melalui berbagai media digital seperti mesin pencari, video pembelajaran, forum diskusi, hingga teknologi kecerdasan buatan (*Artificial Intelligence*). Beragam sumber tersebut memberikan kemudahan bagi siapa saja untuk belajar kapan pun dan di mana pun.

Meskipun demikian, penulis meyakini bahwa buku masih memiliki tempat yang istimewa dalam proses belajar. Masih banyak pembaca yang menikmati pengalaman membaca secara lebih tenang, menandai halaman penting, mencatat gagasan, serta menyentuh langsung lembar demi lembar yang berisi ilmu pengetahuan. Buku memberikan ruang untuk belajar secara lebih terstruktur, mendalam, dan berkesinambungan dibandingkan informasi yang sering kali tersebar di berbagai sumber.

Atas dasar itulah buku ini disusun dengan alur yang sistematis, dimulai dari konsep dasar embedded system, arsitektur perangkat keras, pemrograman, komunikasi, sistem operasi real-time, keamanan, hingga implementasi pada Internet of Things (IoT) dan berbagai studi kasus. Penulis berupaya menyajikan materi secara bertahap dan runut agar mudah dipahami oleh pembaca yang baru memulai maupun yang ingin memperdalam pemahaman di bidang ini.

Penulis menyadari bahwa ilmu pengetahuan dan teknologi terus berkembang dari waktu ke waktu. Oleh karena itu, buku ini tidak dimaksudkan sebagai sumber yang sempurna, melainkan sebagai salah satu rujukan yang dapat membantu pembaca memahami prinsip-prinsip dasar yang akan tetap relevan dalam beberapa tahun ke depan. Penulis berharap buku ini dapat menjadi teman belajar, sumber inspirasi, dan pijakan awal bagi siapa saja yang ingin mengembangkan kemampuan di bidang embedded system.

Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dalam proses penyusunan buku ini. Kritik, saran, dan

masukan yang membangun sangat diharapkan demi perbaikan pada edisi-edisi berikutnya.

Selamat belajar, terus berkembang, dan jangan pernah berhenti untuk mengeksplorasi teknologi.

Penulis

Batam, Juli 2026

Penulis

DAFTAR ISI

DAFTAR ISI iii
DAFTAR GAMBAR viii
DAFTAR TABEL x
DAFTAR LAMPIRAN xv
BAB 1: PENGENALAN EMBEDDED SYSTEM 1
1.1 Definisi dan Karakteristik 1
1.2 Perbedaan dengan PC 5
1.3 Aplikasi Nyata 7
1.4 Tren Industri 9
1.5 Ringkasan 11
Bab 2: Arsitektur Hardware Dasar 13
2.1 Prosesor dan Mikrokontroler 13
2.2 Memori (ROM, RAM, Flash) 16
2.3 I/O dan Peripheral 18
2.4 Bus dan Interkoneksi 20
2.5 Ringkasan 21
Bab 3: Siklus Pengembangan Sistem 23
3.1 Proses Desain 23
3.2 Toolchain dan IDE 25
3.3 Simulasi dan Emulasi 28
3.4 Debugging Dasar 30
3.5 Ringkasan 32
BAB 4: BAHASA C UNTUK EMBEDDED 35
4.1 C Standar vs Embedded C 35
4.2 Tipe Data dan Register 37
4.4 Manipulasi Bit (Bit Manipulation) 43
4.5 Optimasi Kode 47
4.6 Struktur Data untuk Embedded 51

4.7 Praktik Terbaik dan Pola Umum 54
4.8 Ringkasan 59
BAB 5: PEMROGRAMAN ASSEMBLY 61
5.1 Mengapa Assembly Masih Relevan 61
5.2 Arsitektur Register ARM Cortex-M 67
5.3 Instruksi Dasar 70
5,4 Mode Pengalamatan (Addressing Modes) 75
5.5 Manipulasi Bit dalam Assembly 79
5.6 Integrasi C dan Assembly 81
5.7 Ringkasan 88
BAB 6: MANAJEMEN INTERRUPT 91
6.1 Konsep Dasar Interrupt 91
6.2 Interrupt Service Routine (ISR) 97
6.3 Prioritas dan Nesting Interrupt 100
6.4 Interrupt Latency 105
6.5 Debouncing 106
6.6 Ringkasan 110
BAB 7: PROTOKOL KOMUNIKASI SERIAL 111
7.1 UART / USART 111
7.2 SPI (Serial Peripheral Interface) 116
7.3 I²C (Inter-Integrated Circuit) 121
7.4 CAN Bus (Controller Area Network) 126
7.5 USB Dasar 130
7.6 Ringkasan 131
..... 134
BAB 8: ANTARMUKA SENSOR & AKTUATOR 135
8.1 ADC (Analog-to-Digital Converter) 135
8.2 DAC (Digital-to-Analog Converter) 139
8.3 PWM (Pulse Width Modulation) 140
8.4 Sensor Digital dan Analog 145

8.5 Aktuator: Motor dan Relay 147
8.6 Ringkasan 149
BAB 9: KOMUNIKASI NIRKABEL 151
9.1 Wi-Fi dan Bluetooth 151
9.2 Zigbee dan LoRa 153
9.3 RFID dan NFC 156
9.4 Protokol IoT: MQTT 159
9.5 Pertimbangan Pemilihan Teknologi Nirkabel 163
9.6 Ringkasan 165
BAB 10: KONSEP RTOS 167
10.1 Real-Time vs General Purpose OS 167
10.2 Kernel dan Scheduler 169
10.3 Task dan Thread 170
10.4 FreeRTOS: Overview dan Implementasi 173
10.5 Ringkasan 175
BAB 11: SINKRONISASI & KOMUNIKASI ANTAR-TASK 177
11.1 Semaphore 177
11.2 Mutex 178
11.3 Queue 180
11.4 Event Group 182
11.5 Deadlock dan Race Condition 183
11.6 Ringkasan 184
BAB 12: MANAJEMEN MEMORI & WAKTU 186
12.1 Manajemen Heap dan Stack 186
12.2 Memory Pool 189
12.3 Software Timer 190
12.4 Watchdog Timer 191
12.5 Profiling Timing 192
12.6 Ringkasan 195
BAB 13: MANAJEMEN DAYA 197

13.1 Mode Sleep dan Hibernate 197
13.2 Strategi Low-Power Design 200
13.3 Energy Harvesting 204
13.4 Estimasi dan Pengukuran Konsumsi Daya 205
13.5 Ringkasan 207
..... 208
BAB 14: PENGUJIAN & DEBUGGING 209
14.1 Unit Testing Embedded 209
14.2 JTAG dan Debugging Hardware 213
14.3 Logic Analyzer dan Oscilloscope 215
14.4 Fault Injection 217
14.5 Ringkasan 218
BAB 15: KEAMANAN SISTEM TERTANAM 219
15.1 Threat Model 219
15.2 Secure Boot 220
15.3 Enkripsi dan Komunikasi Aman 224
15.4 Firmware Update Aman (OTA) 227
15.5 Ringkasan 230
BAB 16: EMBEDDED LINUX 232
16.1 Konfigurasi Kernel Linux 232
16.2 Buildroot dan Yocto : Build System untuk Embedded Linux 234
16.3 Device Tree 236
16.5 Ringkasan 241
BAB 17: INTERNET OF THINGS (IoT) 243
17.1 Arsitektur IoT 243
17.2 Edge Computing 244
17.3 Cloud Connectivity 247
17.4 Dashboard dan Monitoring 250
17.5 Ringkasan 250
BAB 18: PROYEK AKHIR & STUDI KASUS 252

18.1 Studi Kasus 1: Smart Home Node 252
18.2 Studi Kasus 2: Sistem Monitoring Industri 255
18.3 Studi Kasus 3: Wearable Device 257
18.4 Studi Kasus 4: Smart Agriculture Node 260
18.5 Panduan Membangun Portofolio Proyek Embedded System 261
18.6 Penutup 263
DAFTAR PUSTAKA 265
LAMPIRAN 269
BIODATA PENULIS 290

BAB 1: PENGENALAN EMBEDDED SYSTEM

Embedded system hadir di hampir setiap aspek kehidupan modern dari peralatan rumah tangga, kendaraan, perangkat medis, hingga infrastruktur industri. Bab ini memperkenalkan konsep dasar embedded system, karakteristik yang membedakannya dari komputer umum, serta gambaran aplikasi dan tren industri terkini.

1.1 Definisi dan Karakteristik

Sistem tertanam (embedded system) merupakan sistem komputasi khusus yang dirancang untuk menjalankan satu atau beberapa fungsi tertentu dalam suatu perangkat atau sistem yang lebih besar. Berbeda dengan komputer serbaguna (general-purpose computer) seperti PC atau laptop yang dapat digunakan untuk berbagai keperluan, sistem tertanam dikembangkan untuk melaksanakan tugas yang spesifik dan telah ditentukan sejak tahap perancangan.

Secara umum, sistem tertanam dapat didefinisikan sebagai sistem komputasi berbasis mikroprosesor atau mikrokontroler yang terintegrasi ke dalam perangkat mekanis maupun elektronik untuk mengendalikan fungsi tertentu. Sistem ini biasanya bekerja dengan berbagai keterbatasan sumber daya, seperti kapasitas memori, daya listrik, ruang fisik, biaya produksi, serta kebutuhan respons waktu nyata (real-time). Oleh karena itu, desain sistem tertanam lebih berfokus pada efisiensi, keandalan, dan ketepatan fungsi dibandingkan fleksibilitas penggunaan.

Istilah embedded atau “tertanam” mengacu pada kenyataan bahwa sistem komputer tersebut menjadi bagian integral dari perangkat yang dikendalikannya. Dalam banyak kasus, keberadaan sistem tertanam tidak terlihat secara langsung oleh pengguna. Pengguna hanya berinteraksi dengan perangkat utamanya, sementara proses komputasi berlangsung secara otomatis di balik layar. Dengan kata lain, sistem tertanam bekerja secara transparan sehingga sering kali pengguna tidak menyadari bahwa perangkat yang digunakan sebenarnya mengandung sebuah komputer khusus di dalamnya.

Sistem tertanam merupakan kombinasi antara perangkat keras (hardware), perangkat lunak (firmware atau software), serta berbagai komponen pendukung seperti sensor, aktuator, dan antarmuka komunikasi. Seluruh komponen tersebut dirancang untuk bekerja secara terpadu guna mencapai tujuan fungsional tertentu. Karena sifatnya yang khusus, fungsi sistem tertanam umumnya tidak diubah selama masa pakai perangkat, berbeda dengan komputer serbaguna yang dapat dipasang berbagai aplikasi sesuai kebutuhan pengguna.

Dalam kehidupan sehari-hari, sistem tertanam dapat ditemukan pada berbagai perangkat yang sering digunakan. Sebagai contoh, mesin cuci modern menggunakan mikrokontroler untuk mengatur siklus pencucian, mengontrol

putaran motor, serta memantau sensor ketinggian air. Pengguna memandang perangkat tersebut sebagai mesin cuci biasa, padahal di dalamnya terdapat sistem komputer kecil yang mengelola seluruh proses operasional. Contoh lainnya meliputi kalkulator, televisi pintar (smart TV), kamera digital, router internet, sistem navigasi kendaraan, drone, jam tangan pintar (smartwatch), hingga kendaraan modern yang memiliki puluhan unit pengendali elektronik (Electronic Control Unit atau ECU).

Perkembangan teknologi digital, Internet of Things (IoT), kecerdasan buatan (Artificial Intelligence), dan otomasi industri semakin memperluas peran sistem tertanam dalam kehidupan manusia. Saat ini, sistem tertanam menjadi fondasi bagi berbagai perangkat pintar yang memungkinkan proses pemantauan, pengendalian, dan pengambilan keputusan dilakukan secara otomatis. Oleh karena itu, dapat dikatakan bahwa sebagian besar teknologi modern yang digunakan sehari-hari sebenarnya bergantung pada keberadaan sistem tertanam yang bekerja secara senyap namun sangat penting di balik operasional perangkat tersebut.

Karakteristik	**Penjelasan Singkat**	**Contoh**
Dedicated Function	Dirancang untuk menjalankan tugas tertentu yang telah ditentukan sehingga hardware dan software dapat dioptimalkan untuk fungsi tersebut.	Thermostat pintar yang hanya memantau dan mengendalikan suhu ruangan.
Resource-Constrained	Memiliki keterbatasan memori, kecepatan prosesor, dan konsumsi daya sehingga penggunaan sumber daya harus efisien.	Mikrokontroler dengan RAM beberapa kilobyte dan daya rendah.
Real-Time Constraints	Harus merespons kejadian dalam batas waktu tertentu agar sistem tetap berfungsi dengan benar.	Sistem airbag kendaraan yang harus aktif dalam hitungan milidetik.
Reliability	Dirancang untuk beroperasi dalam waktu lama dengan tingkat kegagalan yang sangat rendah.	Pacemaker yang bekerja terus-menerus di dalam tubuh pasien.
Cost-Sensitive	Pemilihan komponen mempertimbangkan biaya agar produk tetap ekonomis pada produksi massal.	Pemilihan mikrokontroler dengan spesifikasi

		yang sesuai kebutuhan aplikasi.

Tabel 1. 1 Karakteristik Utama Embedded System

Selain karakteristik utama tersebut, beberapa embedded system juga dirancang untuk beroperasi tanpa pengawasan (*unattended operation*), mampu bekerja pada lingkungan yang ekstrem, serta mendukung pembaruan firmware (*firmware update*) tanpa perlu mengganti perangkat keras.

Komponen Penyusun Embedded System

Secara umum, embedded system terdiri atas dua komponen utama, yaitu hardware dan software. Kedua komponen tersebut bekerja secara terpadu untuk menjalankan fungsi yang telah ditentukan. Hardware menyediakan kemampuan fisik untuk berinteraksi dengan lingkungan, sedangkan software mengendalikan dan mengatur bagaimana kemampuan tersebut digunakan.

1. Komponen Hardware

Perangkat keras merupakan bagian fisik dari embedded system yang bertugas melakukan pemrosesan, menerima masukan, menghasilkan keluaran, serta mengelola komunikasi dan kebutuhan daya.

Komponen	**Fungsi Utama**	**Contoh**
Mikrokontroler / Prosesor	Menjalankan program dan mengendalikan seluruh operasi sistem	STM32, ESP32, AVR, Raspberry Pi
Memori	Menyimpan program dan data	Flash, EEPROM, SRAM, DDR
Sensor	Mengubah besaran fisik menjadi sinyal yang dapat diproses	Sensor suhu, cahaya, kelembapan, tekanan
Aktuator	Mengubah sinyal listrik menjadi aksi fisik	Motor DC, Servo, Relay, Buzzer
Antarmuka Komunikasi	Memungkinkan pertukaran data dengan perangkat lain	UART, SPI, I^2C, CAN, Wi-Fi, Bluetooth
Catu Daya	Menyediakan energi untuk seluruh sistem	Adaptor, baterai, panel surya

Tabel 1. 2 Komponen Hardware pada Embedded System

2. Komponen Software

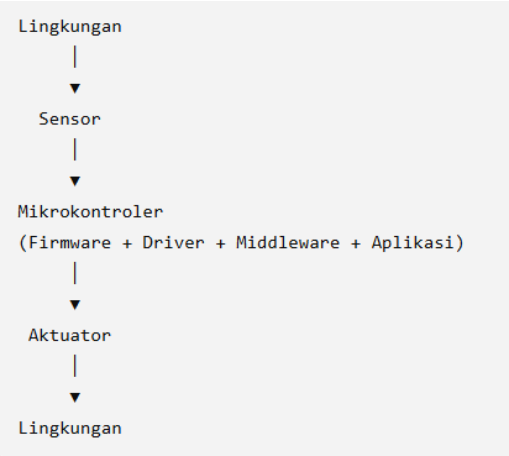
Perangkat lunak mengendalikan hardware dan mengimplementasikan fungsi yang diinginkan pada sistem. Secara umum, software embedded disusun dalam beberapa lapisan agar lebih mudah dikembangkan dan dipelihara.

Komponen	**Fungsi Utama**	**Contoh**
Firmware	Mengendalikan hardware secara langsung	Program utama pada mikrokontroler
Driver	Menyediakan antarmuka ke periferal	Driver UART, SPI, sensor, LCD
Middleware	Menyediakan layanan umum bagi aplikasi	FreeRTOS, TCP/IP Stack, FATFS, BLE Stack
Aplikasi	Mengimplementasikan fungsi utama produk	Kontrol suhu, pemantauan kesehatan, sistem keamanan

Tabel 1. 3 Komponen Software pada Embedded System

3. Hubungan Hardware dan Software

Hardware dan software bekerja secara saling melengkapi. Sensor memperoleh data dari lingkungan, mikrokontroler memproses data tersebut melalui firmware dan aplikasi, kemudian aktuator menghasilkan respons yang sesuai. Tanpa software, hardware tidak dapat menjalankan fungsi yang diinginkan. Sebaliknya, software tidak dapat bekerja tanpa dukungan hardware yang tepat.



Gambar 1. 1 Hubungan Hardware dan Software pada Embedded System

1.2 Perbedaan dengan PC

Meskipun sama-sama merupakan sistem komputer yang terdiri atas perangkat keras dan perangkat lunak, *embedded system* dan *personal computer* (PC) dirancang dengan tujuan yang berbeda. Personal computer dikembangkan sebagai perangkat komputasi serbaguna yang dapat menjalankan berbagai aplikasi sesuai kebutuhan pengguna. Sebaliknya, *embedded system* dirancang untuk menjalankan fungsi tertentu secara khusus dan menjadi bagian dari perangkat yang lebih besar.

Perbedaan tersebut memengaruhi hampir seluruh aspek desain, mulai dari perangkat keras, perangkat lunak, kebutuhan sumber daya, hingga cara pengguna berinteraksi dengan sistem. Tabel berikut memberikan gambaran umum mengenai perbedaan antara keduanya.

Aspek	**Embedded System**	**Personal Computer (PC)**
Tujuan	Dirancang untuk menjalankan fungsi spesifik (*dedicated function*)	Dirancang untuk berbagai kebutuhan komputasi (*general-purpose computing*)
Sistem Operasi	Dapat berjalan tanpa sistem operasi, menggunakan RTOS, atau distribusi Linux ringan	Menggunakan sistem operasi lengkap seperti Windows, macOS, atau Linux desktop
Sumber Daya (Resource)	Terbatas, biasanya memiliki memori dan kapasitas penyimpanan yang relatif kecil	Memiliki sumber daya besar dengan RAM dan penyimpanan berkapasitas tinggi
Antarmuka Pengguna	Minimal atau bahkan tanpa antarmuka pengguna (*headless*)	Menggunakan antarmuka grafis lengkap dengan layar, keyboard, dan mouse
Siklus Hidup Produk	Umumnya panjang, dapat digunakan hingga puluhan tahun pada aplikasi industri	Relatif lebih pendek karena perkembangan teknologi yang cepat
Pembaruan Perangkat Lunak	Terbatas dan biasanya dilakukan melalui pembaruan firmware atau OTA (*Over-the-Air*)	Mudah diperbarui melalui installer, pembaruan sistem operasi, atau toko aplikasi

Konsumsi Daya	Dioptimalkan untuk konsumsi daya rendah	Membutuhkan daya yang lebih besar untuk mendukung berbagai aplikasi
Biaya Produksi	Dirancang agar ekonomis sesuai fungsi yang dibutuhkan	Biaya lebih tinggi karena harus mendukung berbagai jenis aplikasi
Keandalan	Diutamakan untuk bekerja stabil dan terus-menerus dalam jangka panjang	Lebih berfokus pada fleksibilitas dan kenyamanan pengguna
Contoh Perangkat	Mesin cuci, smart TV, kalkulator, drone, ECU mobil, smartwatch	Desktop PC, laptop, workstation

Tabel 1. 4 Perbedaan Embedded System dan Personal Komputer

Secara konseptual, sebuah PC dapat diibaratkan sebagai "pisau serbaguna" yang mampu melakukan berbagai pekerjaan, mulai dari mengetik dokumen, mengedit video, bermain gim, hingga menjalankan aplikasi bisnis. Sebaliknya, *embedded system* lebih mirip "alat khusus" yang dirancang untuk satu tujuan tertentu dan melakukannya dengan sangat efisien.

Sebagai contoh, sebuah laptop memungkinkan pengguna memasang dan menjalankan ribuan aplikasi yang berbeda sesuai kebutuhan. Sebaliknya, sistem komputer yang terdapat pada mesin cuci hanya bertugas mengatur siklus pencucian, membaca sensor air, serta mengendalikan motor penggerak. Meskipun kemampuan komputasinya jauh lebih terbatas dibandingkan laptop, sistem tersebut dirancang agar dapat bekerja secara andal selama bertahun-tahun dengan konsumsi daya yang rendah.

Menariknya, perkembangan teknologi modern membuat batas antara *embedded system* dan komputer umum semakin kabur. Perangkat seperti smartphone, smart TV, dan sistem infotainment kendaraan kini memiliki kemampuan komputasi yang mendekati komputer pribadi. Namun, karena perangkat tersebut tetap dirancang untuk fungsi-fungsi tertentu dalam suatu produk, perangkat tersebut masih dapat dikategorikan sebagai *embedded system*.

Dengan demikian, perbedaan utama antara *embedded system* dan PC bukan terletak pada ukuran atau kemampuan komputasinya, melainkan pada tujuan perancangan, tingkat fleksibilitas, serta peran sistem tersebut dalam perangkat yang digunakannya.

1.3 Aplikasi Nyata

Perkembangan teknologi digital telah menjadikan *embedded system* sebagai bagian yang tidak terpisahkan dari kehidupan modern. Saat ini, sistem tertanam digunakan dalam berbagai sektor industri untuk mengendalikan, memantau, dan mengotomatisasi beragam proses. Keunggulan utamanya terletak pada kemampuannya menjalankan fungsi khusus secara efisien, andal, dan sering kali dalam waktu nyata (*real-time*).

Meskipun sering tidak terlihat oleh pengguna, *embedded system* bekerja di balik berbagai perangkat yang digunakan setiap hari, mulai dari peralatan rumah tangga hingga sistem kritis pada pesawat terbang. Setiap sektor memiliki karakteristik dan kebutuhan yang berbeda, sehingga desain sistem tertanam pun disesuaikan dengan lingkungan operasinya.

Sektor	**Contoh Aplikasi**
Konsumer	Smart TV, *smart speaker*, *smartwatch*, mesin cuci otomatis, rice cooker digital, kamera digital
Otomotif	*Engine Control Unit* (ECU), sistem ABS, airbag, sistem infotainment, *Advanced Driver Assistance System* (ADAS)
Medis	Alat pacu jantung, pompa infus, monitor pasien, alat USG portabel, alat pengukur kadar gula darah
Industri	*Programmable Logic Controller* (PLC), robot industri, sistem SCADA, sensor pada jalur produksi (*conveyor belt*)
Telekomunikasi	Router, modem, *access point*, *base station* jaringan seluler, perangkat jaringan optik
Dirgantara dan Pertahanan	Sistem navigasi pesawat, autopilot, drone, radar, sistem kendali rudal dan satelit

Tabel 1. 5 Contoh Aplikasi Embedded System Berdasarkan Sektor

Embedded System pada Perangkat Konsumen

Sektor konsumen merupakan salah satu pengguna terbesar teknologi *embedded system*. Hampir seluruh perangkat elektronik modern yang digunakan di rumah mengandung mikrokontroler atau prosesor yang mengendalikan fungsi-fungsi perangkat tersebut.

Sebagai contoh, sebuah mesin cuci modern menggunakan sistem tertanam untuk mengatur durasi pencucian, mengontrol putaran motor, membaca sensor ketinggian air, serta menentukan proses pembilasan dan pengeringan. Demikian pula dengan *smart TV*, yang tidak hanya menampilkan siaran televisi tetapi juga menjalankan aplikasi streaming, mengelola koneksi internet, dan memproses perintah suara.

Embedded System pada Industri Otomotif

Kendaraan modern dapat dianggap sebagai kumpulan puluhan bahkan ratusan sistem tertanam yang bekerja secara bersamaan. Sebuah mobil masa kini tidak lagi sekadar perangkat mekanis, melainkan sebuah sistem komputasi bergerak yang kompleks.

Salah satu contoh paling penting adalah *Engine Control Unit* (ECU), yaitu komputer yang mengatur proses pembakaran bahan bakar, pengapian, dan emisi kendaraan. Selain itu terdapat sistem ABS yang mencegah roda terkunci saat pengereman mendadak, airbag yang harus bereaksi dalam hitungan milidetik ketika terjadi tabrakan, serta sistem ADAS yang membantu pengemudi melalui fitur seperti pengereman otomatis dan deteksi jalur.

Embedded System pada Bidang Medis

Dalam dunia kesehatan, keandalan dan keselamatan menjadi faktor utama dalam pengembangan sistem tertanam. Kesalahan kecil dapat berdampak langsung terhadap keselamatan pasien sehingga perangkat medis harus memenuhi standar kualitas yang sangat ketat.

Contohnya adalah alat pacu jantung (*pacemaker*) yang secara terus-menerus memantau dan mengatur ritme detak jantung pasien. Selain itu, pompa infus elektronik mengontrol laju pemberian obat secara presisi, sementara monitor pasien mengumpulkan dan menampilkan data vital seperti tekanan darah, denyut jantung, dan kadar oksigen secara real-time.

Embedded System pada Industri dan Otomasi

Sektor industri memanfaatkan *embedded system* untuk meningkatkan efisiensi, produktivitas, dan keamanan proses produksi. Salah satu perangkat yang paling banyak digunakan adalah *Programmable Logic Controller* (PLC), yaitu komputer industri yang dirancang untuk mengendalikan mesin dan proses otomatis.

Pada pabrik modern, sistem tertanam digunakan untuk mengontrol robot industri, membaca sensor pada jalur produksi, memantau kondisi mesin, serta mengintegrasikan data ke dalam sistem pengawasan industri seperti SCADA (*Supervisory Control and Data Acquisition*). Keberadaan sistem ini memungkinkan proses produksi berjalan secara otomatis dengan tingkat akurasi yang tinggi.

Embedded System pada Telekomunikasi

Jaringan komunikasi modern tidak dapat beroperasi tanpa dukungan sistem tertanam. Router, modem, *access point*, dan perangkat jaringan lainnya menggunakan prosesor khusus untuk mengelola lalu lintas data yang sangat besar.

Ketika pengguna mengakses internet melalui Wi-Fi atau jaringan seluler, berbagai sistem tertanam pada perangkat jaringan bekerja secara otomatis untuk mengatur pengiriman data, menjaga keamanan komunikasi, serta memastikan kualitas layanan tetap optimal.

Embedded System pada Dirgantara dan Pertahanan

Bidang dirgantara dan pertahanan merupakan salah satu sektor yang memiliki persyaratan paling ketat terhadap *embedded system*. Sistem harus mampu beroperasi dengan tingkat keandalan yang sangat tinggi karena kegagalan dapat menyebabkan kerugian besar atau bahkan mengancam keselamatan manusia.

Contohnya adalah sistem navigasi pesawat yang membantu menentukan posisi dan arah penerbangan secara akurat. Drone menggunakan berbagai sensor dan algoritma kontrol untuk menjaga stabilitas selama terbang, sedangkan sistem radar dan kendali pertahanan harus mampu memproses data dalam waktu sangat singkat untuk mendukung pengambilan keputusan secara real-time.

Dari berbagai contoh tersebut terlihat bahwa *embedded system* telah menjadi fondasi bagi sebagian besar teknologi modern. Mulai dari perangkat sederhana seperti rice cooker hingga sistem kompleks pada kendaraan, rumah sakit, pabrik, dan pesawat terbang, semuanya bergantung pada kemampuan sistem tertanam untuk menjalankan fungsi-fungsi khusus secara efisien, andal, dan otomatis. Karena itu, dapat dikatakan bahwa kehidupan modern saat ini tidak dapat dipisahkan dari keberadaan *embedded system* yang bekerja di balik berbagai teknologi di sekitar kita.

1.4 Tren Industri

Perkembangan teknologi digital telah mendorong evolusi *embedded system* dari perangkat yang hanya menjalankan fungsi sederhana menjadi sistem yang semakin cerdas, terhubung, dan adaptif. Kebutuhan akan konektivitas, efisiensi energi, keamanan, serta kemampuan pemrosesan data secara real-time menjadi faktor utama yang membentuk arah perkembangan industri ini.

Saat ini, *embedded system* tidak lagi hanya berfungsi sebagai pengendali perangkat elektronik, tetapi juga menjadi fondasi bagi berbagai inovasi seperti Internet of Things (IoT), kendaraan otonom, perangkat medis pintar, hingga sistem industri berbasis kecerdasan buatan. Berikut beberapa tren utama yang sedang dan akan terus memengaruhi perkembangan industri *embedded system*.

- Internet of Things (IoT)

Salah satu tren paling signifikan dalam dunia *embedded system* adalah berkembangnya konsep *Internet of Things* (IoT), yaitu jaringan perangkat yang saling terhubung melalui internet untuk bertukar data dan berinteraksi secara otomatis. Pada ekosistem IoT, sistem tertanam menjadi komponen utama yang mengumpulkan data dari lingkungan melalui sensor, memproses informasi tersebut, kemudian mengirimkannya ke server atau layanan *cloud*. Teknologi ini memungkinkan berbagai perangkat, seperti lampu pintar, kamera keamanan, sensor pertanian, hingga perangkat industri, bekerja secara terintegrasi dan dapat dipantau dari jarak jauh.

Pertumbuhan IoT diperkirakan akan terus meningkat karena semakin banyak organisasi dan individu yang membutuhkan sistem pemantauan dan otomasi yang efisien.

- Edge AI dan TinyML

Kemajuan kecerdasan buatan (*Artificial Intelligence*) tidak lagi terbatas pada komputer berperforma tinggi atau pusat data. Saat ini, model AI dapat dijalankan langsung pada perangkat tertanam melalui pendekatan *Edge AI* dan *Tiny Machine Learning* (TinyML). *Edge AI* memungkinkan proses analisis dan pengambilan keputusan dilakukan langsung pada perangkat tanpa harus mengirim data ke *cloud*. Pendekatan ini memberikan berbagai keuntungan, seperti respons yang lebih cepat, konsumsi bandwidth yang lebih rendah, serta peningkatan privasi data.

Sementara itu, TinyML memungkinkan model pembelajaran mesin berukuran kecil dijalankan pada mikrokontroler dengan sumber daya yang sangat terbatas. Teknologi ini banyak digunakan pada perangkat *wearable*, sensor pintar, sistem deteksi suara, dan berbagai aplikasi IoT modern.

- Arsitektur Prosesor RISC-V

Dalam beberapa tahun terakhir, arsitektur prosesor *RISC-V* mendapatkan perhatian besar sebagai alternatif terhadap arsitektur komersial yang telah lama mendominasi industri. RISC-V merupakan arsitektur instruksi (*Instruction Set Architecture* atau ISA) yang bersifat terbuka (*open-source*), sehingga perusahaan maupun institusi pendidikan dapat mengembangkan prosesor tanpa harus membayar lisensi. Fleksibilitas tersebut mendorong inovasi yang lebih cepat dan biaya pengembangan yang lebih rendah.

Dengan semakin luasnya dukungan perangkat lunak dan ekosistem pengembangan, RISC-V diprediksi akan menjadi salah satu teknologi penting dalam generasi berikutnya dari sistem tertanam.

- Security by Design

Meningkatnya jumlah perangkat yang terhubung ke internet juga meningkatkan risiko serangan siber. Oleh karena itu, keamanan tidak lagi dianggap sebagai fitur tambahan, melainkan menjadi bagian yang harus dipertimbangkan sejak tahap

awal perancangan sistem. Konsep *Security by Design* menekankan bahwa aspek keamanan harus diintegrasikan ke dalam seluruh siklus pengembangan produk, mulai dari desain perangkat keras, pengembangan firmware, hingga mekanisme pembaruan perangkat lunak. Beberapa implementasi yang umum digunakan meliputi *secure boot*, enkripsi data, autentikasi perangkat, manajemen kunci kriptografi, serta mekanisme pembaruan firmware yang aman. Pendekatan ini sangat penting untuk melindungi perangkat dari akses tidak sah dan berbagai ancaman keamanan siber.

- Low-Power Design dan Efisiensi Energi

Efisiensi energi menjadi salah satu fokus utama dalam pengembangan *embedded system*, terutama untuk perangkat yang menggunakan baterai atau beroperasi di lokasi yang sulit dijangkau. Desain berdaya rendah (*low-power design*) bertujuan untuk memperpanjang masa pakai baterai sekaligus mengurangi konsumsi energi secara keseluruhan. Berbagai teknik digunakan untuk mencapai tujuan tersebut, seperti mode tidur (*sleep mode*), pengelolaan daya dinamis (*dynamic power management*), pemrosesan berbasis peristiwa (*event-driven processing*), serta penggunaan komponen elektronik dengan konsumsi daya yang sangat rendah. Selain mendukung perangkat portabel, tren ini juga sejalan dengan meningkatnya perhatian terhadap isu keberlanjutan (*sustainability*) dan efisiensi energi global.

- Software-Defined Hardware

Tren berikutnya adalah meningkatnya penggunaan perangkat keras yang dapat dikonfigurasi ulang (*reconfigurable hardware*), seperti *Field Programmable Gate Array* (FPGA). Konsep *Software-Defined Hardware* memungkinkan sebagian fungsi perangkat keras diubah melalui perangkat lunak tanpa harus mengganti komponen fisik. Hal ini memberikan fleksibilitas yang tinggi dalam pengembangan produk, mempercepat proses inovasi, serta memungkinkan pembaruan fitur setelah perangkat diproduksi. Teknologi ini banyak digunakan pada bidang telekomunikasi, kecerdasan buatan, pemrosesan sinyal digital, dan sistem industri yang membutuhkan kemampuan adaptasi terhadap perubahan kebutuhan.

1.5 Ringkasan

Pada bab ini telah dibahas konsep dasar *embedded system* sebagai fondasi untuk memahami teknologi sistem tertanam secara lebih mendalam. Embedded system merupakan sistem komputer yang dirancang untuk menjalankan fungsi tertentu secara khusus (*dedicated function*) dan menjadi bagian integral dari perangkat yang dikendalikannya. Berbeda dengan komputer serbaguna, sistem tertanam dikembangkan dengan mempertimbangkan berbagai keterbatasan, seperti kapasitas memori, daya komputasi, konsumsi energi, biaya, serta kebutuhan respons waktu nyata (*real-time*).

Pembahasan juga menunjukkan bahwa embedded system terdiri atas kombinasi perangkat keras dan perangkat lunak yang bekerja secara terpadu. Perangkat keras mencakup komponen seperti mikrokontroler, memori, sensor, aktuator, antarmuka komunikasi, dan catu daya, sedangkan perangkat lunak terdiri atas firmware, driver, middleware, serta aplikasi yang mengimplementasikan fungsi utama sistem.

Selain itu, telah dijelaskan perbedaan mendasar antara embedded system dan *personal computer* (PC). Meskipun keduanya merupakan sistem komputasi, embedded system dirancang untuk tugas yang spesifik dengan sumber daya terbatas dan tingkat keandalan yang tinggi, sedangkan PC dikembangkan sebagai platform komputasi serbaguna yang fleksibel dan dapat menjalankan berbagai aplikasi sesuai kebutuhan pengguna.

Keberadaan embedded system juga dapat ditemukan pada hampir seluruh sektor kehidupan modern. Berbagai perangkat konsumen seperti *smart TV*, *smartwatch*, dan mesin cuci digital memanfaatkan sistem tertanam untuk menjalankan fungsinya. Di sektor otomotif, embedded system digunakan pada ECU, sistem ABS, dan fitur bantuan pengemudi. Pada bidang medis, teknologi ini mendukung perangkat seperti alat pacu jantung dan monitor pasien. Sementara itu, sektor industri, telekomunikasi, serta dirgantara dan pertahanan juga sangat bergantung pada sistem tertanam untuk mendukung proses otomatisasi, komunikasi, dan pengendalian sistem yang kompleks.

Perkembangan teknologi turut mendorong lahirnya berbagai tren baru dalam industri embedded system. Integrasi dengan *Internet of Things* (IoT) memungkinkan perangkat saling terhubung dan bertukar data melalui jaringan. Kemajuan *Edge AI* dan TinyML memungkinkan kecerdasan buatan dijalankan langsung pada perangkat dengan sumber daya terbatas. Di sisi lain, meningkatnya ancaman siber menjadikan konsep *Security by Design* sebagai kebutuhan utama dalam pengembangan produk. Efisiensi energi melalui *low-power design* serta penggunaan perangkat keras yang dapat dikonfigurasi ulang seperti FPGA juga menjadi arah penting dalam perkembangan teknologi sistem tertanam modern. Dengan memahami konsep, karakteristik, komponen, aplikasi, dan tren perkembangan embedded system, pembaca diharapkan memperoleh gambaran menyeluruh mengenai peran sistem tertanam dalam berbagai teknologi yang digunakan saat ini. Pengetahuan dasar ini akan menjadi landasan untuk mempelajari aspek-aspek teknis yang lebih mendalam pada bab-bab berikutnya.

DAFTAR PUSTAKA

[1] ARM Limited, *ARM Cortex-M4 Devices Generic User Guide*, ARM Holdings, 2010. [Online]. Available: https://developer.arm.com/documentation/dui0553

[2] M. Barr, *Programming Embedded Systems in C and C++*. Sebastopol, CA: O'Reilly Media, 1999.

[3] M. Barr and A. Massa, *Programming Embedded Systems: With C and GNU Development Tools*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2006.

[4] A. Bejo, *C dan AVR: Rahasia Kemudahan Bahasa C dalam Mikrokontroler ATMega8535*. Yogyakarta: Graha Ilmu, 2008.

[5] Bluetooth SIG, *Bluetooth Core Specification Version 5.4*. Bluetooth SIG, Inc., 2023. [Online]. Available: https://www.bluetooth.com/specifications/specs/core54

[6] Linux Foundation, *Buildroot: Making Embedded Linux Easy*, 2024. [Online]. Available: https://buildroot.org/docs.html

[7] G. C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, 3rd ed. New York, NY: Springer, 2011.

[8] C.-Y. Chen, M. Hasan, and S. Mohan, "Securing real-time Internet-of-Things," *Sensors*, vol. 18, no. 12, p. 4356, Dec. 2018. doi: 10.3390/s18124356.

[9] I. Craig, *Real-Time Systems Scheduling, Analysis and Verification*. Chichester, UK: John Wiley & Sons, 2002.

[10] Espressif Systems, *ESP32 Technical Reference Manual*. Espressif Systems, 2024. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

[11] FreeRTOS, *FreeRTOS Reference Manual*. Real Time Engineers Ltd., 2023. [Online]. Available: https://www.freertos.org/Documentation/RTOS_book.html

[12] J. Ganssle, *The Art of Designing Embedded Systems*, 2nd ed. Burlington, MA: Newnes/Elsevier, 2008.

[13] S. Heath, *Embedded Systems Design*, 2nd ed. Oxford, UK: Newnes/Elsevier, 2003.

[14] M. A. Heryanto dan W. A. P., *Pemrograman Bahasa C untuk Mikrokontroler ATMega8535*. Yogyakarta: Penerbit Andi, 2008, ISBN: 978-979-29-0209-9.

[15] C. A. R. Hoare, "Communicating sequential processes," *Communications of the ACM*, vol. 21, no. 8, pp. 666–677, Aug. 1978. doi: 10.1145/359576.359585.

[16] IEC, *IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems*. International Electrotechnical Commission, 2010.

[17] ISO, *ISO 26262: Road Vehicles — Functional Safety*. International Organization for Standardization, 2018.

[18] A. Kadir, *Panduan Praktis Mempelajari Aplikasi Mikrokontroler dan Pemrogramannya Menggunakan Arduino*. Yogyakarta: Penerbit Andi, 2013, ISBN: 978-979-29-4017-6.

[19] R. Kamal, *Embedded Systems: Architecture, Programming and Design*, 2nd ed. New Delhi, India: Tata McGraw-Hill, 2008.

[20] B. W. Kernighan and D. M. Ritchie, *The C Programming Language*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1988.

[21] P. Koopman, *Better Embedded System Software*. Pittsburgh, PA: Drumnadrochit Education, 2010.

[22] D. Lacamera, *Embedded Systems Architecture: Design and Write Software for Embedded Devices to Build Safe and Connected Systems*, 2nd ed. Birmingham, UK: Packt Publishing, 2023, ISBN: 978-1-80323-954-5.

[23] J. J. Labrosse, *MicroC/OS-II: The Real-Time Kernel*, 2nd ed. Lawrence, KS: CMP Books, 2002.

[24] J. J. Labrosse, *µC/OS-III: The Real-Time Kernel*. Weston, FL: Micrium Press, 2009.

[25] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Communications of the ACM*, vol. 21, no. 7, pp. 558–565, Jul. 1978. doi: 10.1145/359545.359563.

[26] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, Jan. 1973. doi: 10.1145/321738.321743.

[27] J. W. S. Liu, *Real-Time Systems*. Upper Saddle River, NJ: Prentice Hall, 2000.

[28] P. Marwedel, *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things*, 4th ed. Cham, Switzerland: Springer, 2021. doi: 10.1007/978-3-030-60910-8.

[29] M. A. Mazidi, R. D. McKinlay, and D. Causey, *PIC Microcontroller and Embedded Systems: Using Assembly and C for PIC18*. Upper Saddle River, NJ: Pearson Education, 2008.

[30] M. Messier, *Embedded Systems Security: Practical Methods for Safe and Secure Software and Systems Development*. Burlington, MA: Newnes/Elsevier, 2015.

[31] MISRA Consortium, *MISRA C:2012 — Guidelines for the Use of the C Language in Critical Systems*, 3rd ed. Nuneaton, UK: MIRA Ltd., 2019.

[32] NXP Semiconductors, *UM10204: I2C-Bus Specification and User Manual*, Rev. 7.0. NXP Semiconductors N.V., 2021. [Online]. Available: https://www.nxp.com/docs/en/user-guide/UM10204.pdf

[33] OASIS, *MQTT Version 5.0*. OASIS Standard, 2019. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

[34] M. J. Pont, *Embedded C*. Harlow, UK: Addison-Wesley, 2002.

[35] RISC-V International, *The RISC-V Instruction Set Manual, Volume I: Unprivileged ISA*, Document Version 20191213. RISC-V Foundation, 2019. [Online]. Available: https://riscv.org/technical/specifications/

[36] A. Shostack, *Threat Modeling: Designing for Security*. Indianapolis, IN: Wiley, 2014.

[37] D. E. Simon, *An Embedded Software Primer*. Reading, MA: Addison-Wesley Professional, 1999, ISBN: 978-0-201-61569-2.

[38] C. Simmonds, *Mastering Embedded Linux Programming*. Birmingham, UK: Packt Publishing, 2015.

[39] A. N. Sloss, D. Symes, and C. Wright, *ARM System Developer's Guide: Designing and Optimizing System Software*. San Francisco, CA: Morgan Kaufmann, 2004.

[40] STMicroelectronics, *RM0090: STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 Reference Manual*. STMicroelectronics, 2023. [Online]. Available: https://www.st.com/resource/en/reference_manual/rm0090-stm32f405415-stm32f407417-stm32f427437-and-stm32f429439-advanced-armbased-32bit-mcus-stmicroelectronics.pdf

[41] D. Uckelmann, M. Harrison, and F. Michahelles, Eds., *Architecting the Internet of Things*. Berlin, Germany: Springer, 2011.

[42] F. Vahid and T. D. Givargis, *Embedded System Design: A Unified Hardware/Software Introduction*. Hoboken, NJ: John Wiley & Sons, 2002, ISBN: 978-0-471-38678-0.

[43] J. W. Valvano, *Embedded Systems: Introduction to ARM® Cortex™-M Microcontrollers*, 5th ed. CreateSpace Independent Publishing Platform, 2018, ISBN: 978-1-477-50899-2.

[44] J. W. Valvano, *Embedded Systems: Real-Time Interfacing to ARM® Cortex™-M Microcontrollers*, 2nd ed. CreateSpace Independent Publishing Platform, 2012.

[45] J. W. Valvano, *Embedded Systems: Real-Time Operating Systems for ARM® Cortex™-M Microcontrollers*, 3rd ed. CreateSpace Independent Publishing Platform, 2017.

[46] P. Warden and D. Situnayake, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. Sebastopol, CA: O'Reilly Media, 2019.

[47] R. B. Widodo, *Embedded System Menggunakan Mikrokontroler dan Pemrograman C*. Yogyakarta: Penerbit Andi, 2009, ISBN: 978-979-115-325-6.

[48] Wi-Fi Alliance, *Wi-Fi 6 (802.11ax) Technology Introduction*. Wi-Fi Alliance, 2019. [Online]. Available: https://www.wi-fi.org/discover-wi-fi/wi-fi-certified-6

[49] E. White, *Making Embedded Systems: Design Patterns for Great Software*. Sebastopol, CA: O'Reilly Media, 2011.

[50] A. Winoto, *Mikrokontroler AVR ATMega8/16/32/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR*. Bandung: Informatika, 2008.

[51] J. Yiu, *The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors*, 3rd ed. Oxford, UK: Newnes/Elsevier, 2013.

[52] Yocto Project, *Yocto Project Documentation*. Linux Foundation, 2024. [Online]. Available: https://docs.yoctoproject.org

[53] Zephyr Project, *Zephyr RTOS Documentation*. Linux Foundation, 2024. [Online]. Available: https://docs.zephyrproject.org

BIODATA PENULIS

Rahima Tartila, S.T

Lahir pada tahun 2000 dan merupakan lulusan Teknik Komputer Universitas Andalas. Ketertarikannya pada bidang embedded system tumbuh sejak mengenal sensor, mikrokontroler, dan berbagai teknologi yang memungkinkan perangkat elektronik berinteraksi dengan lingkungan sekitarnya. Baginya, embedded system merupakan bidang yang menarik karena menjadi fondasi dari banyak perangkat dan sistem yang digunakan dalam kehidupan sehari-hari.

Melalui kegiatan membaca, menulis, dan melakukan riset secara mandiri, Rahima terus memperdalam pemahamannya mengenai perkembangan teknologi, khususnya di bidang sistem tertanam dan komputasi. Ketertarikannya pada pembelajaran berkelanjutan serta berbagi pengetahuan mendorongnya untuk aktif mengembangkan berbagai karya tulis yang dapat menjadi sumber belajar bagi mahasiswa, praktisi, maupun masyarakat umum.

Cita-cita tidak harus ditemukan, dapat pula diciptakan.