



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**Rancang Bangun Sistem Otomasi Pengomposan Berbasis
Mikrokontroler dengan Monitoring Multiparameter dan Deteksi
Kematangan Kompos**

TUGAS AKHIR

Nabila

2303321040

**POLITEKNIK
NEGERI
JAKARTA**

PROGRAM STUDI D3 ELEKTRONIKA INDUSTRI

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI JAKARTA

2026



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**Perancangan Sistem Monitoring Multiparameter dan Deteksi
Kematangan Kompos**

Nabila

2303321040

Diajukan sebagai salah satu syarat untuk memperoleh Diploma Tiga

**POLITEKNIK
NEGERI
JAKARTA**

PROGRAM STUDI D3 ELEKTRONIKA INDUSTRI

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI JAKARTA

2026



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Nabila
NIM : 2303321040
Tanda tangan : 
Tanggal : 6 Juli 2026

**POLITEKNIK
NEGERI
JAKARTA**



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

HALAMAN PENGESAHAN TUGAS AKHIR

Tugas akhir ini diajukan oleh:

Nama : Nabila
NIM : 2303321040
Program Studi : Elektronika Industri
Judul Tugas Akhir : Rancang Bangun Sistem Otomasi Pengomposan Berbasis Mikrokontroler dengan Monitoring Multiparameter dan Deteksi Kematangan Kompos
Sub Judul Tugas Akhir : Perancangan Sistem Monitoring Multiparameter dan Deteksi Kematangan Kompos

Telah diuji oleh tim penguji dalam sidang tugas akhir pada 6 Juli 2026 dan dinyatakan **LULUS**

Pembimbing : Britantyo Wicaksono, S.Si., M.Eng. ()
NIP. 198404242018031001

**POLITEKNIK
NEGERI
JAKARTA**

Depok, 10 Juli 2026

Disahkan oleh

Ketua Jurusan Teknik Elektro




Dr. Murie Dwiyanti, S.T., M.T.

NIP. 197803312003122002



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa, karena atas berkat dan kasih karunia-Nya penulis dapat menyelesaikan Tugas Akhir ini. Penulisan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Diploma Tiga. Tugas Akhir yang penulis buat berjudul "Perancangan Sistem Monitoring Multiparameter dan Deteksi Kematangan Kompos". Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai penyusunan tugas akhir ini, sangatlah sulit bagi penulis untuk menyelesaikan Tugas Akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Murie Dwiyanti, S.T., M.T. selaku Ketua Jurusan Teknik Elektro;
2. Ihsan Auditia Akhinov, S.T., M.T. selaku Kepala Program Studi Elektronika Industri;
3. Britantyo Wicaksono, S.Si., M.Eng. selaku Dosen Pembimbing Tugas Akhir;
4. Orang tua dan keluarga yang telah memberikan dukungan material, dan moral, serta doa dan dukungan yang diberikan;
5. Zulfan Sahli selaku rekan penulis dalam pembuatan tugas akhir ini;
6. Sahabat dan teman-teman ECA atas doa, dukungan, semangat, kebersamaan, serta bantuan yang diberikan sehingga penyusunan Tugas Akhir ini dapat diselesaikan dengan baik.

Akhir kata, segala kritik dan saran yang bersifat membangun sangat diharapkan demi penyempurnaan Tugas Akhir ini. Semoga karya ini dapat memberikan manfaat serta menjadi referensi bagi pengembangan ilmu pengetahuan, khususnya di bidang Teknik Elektronika.

Depok, 2026

Nabila



ABSTRAK

Sistem monitoring multiparameter dan deteksi kematangan kompos dirancang sebagai solusi untuk melakukan pemantauan kondisi kompos secara otomatis, real-time, dan terstruktur. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem monitoring berbasis Arduino Mega 2560 yang mampu membaca parameter suhu, kelembapan, pH, serta warna kompos dalam proses pengomposan. Sistem menggunakan sensor DS18B20 untuk pengukuran suhu, sensor kelembapan kapasitif untuk kadar air, sensor pH tanah untuk tingkat keasaman, serta sensor warna TCS3200 untuk identifikasi warna kompos. Data hasil pengukuran ditampilkan melalui HMI Nextion dan disimpan secara otomatis pada SD Card dengan pencatatan waktu menggunakan modul RTC DS1302. Penentuan status kematangan kompos dilakukan berdasarkan kombinasi empat parameter, yaitu suhu, kelembapan, pH, dan warna, dengan metode logika AND sehingga seluruh parameter harus memenuhi batas kondisi yang telah ditentukan agar kompos dinyatakan matang. Pengujian sistem dilakukan dengan pengambilan data sebanyak 30 kali percobaan untuk setiap sensor. Hasil pengujian menunjukkan bahwa sensor suhu memiliki rata-rata error sebesar 0,61°C, sensor kelembapan memiliki rata-rata error sebesar 0,93%, dan sensor pH memiliki selisih pembacaan sekitar 0,1–0,5 dibandingkan nilai referensi. Sistem juga mampu mendeteksi perubahan warna kompos dari hijau hingga coklat gelap sebagai indikator kematangan. Secara keseluruhan, sistem mampu bekerja dengan baik dalam melakukan monitoring, pencatatan data, serta klasifikasi kematangan kompos secara otomatis sesuai dengan parameter yang telah ditentukan.

Kata kunci: Arduino Mega, kematangan kompos, monitoring multiparameter, pH, warna kompos.

POLITEKNIK
NEGERI
JAKARTA

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Design of a Multiparameter Monitoring System and Compost Maturity Detection

ABSTRACT

The multiparameter monitoring system and compost maturity detection system were designed as an automated solution for real-time and structured monitoring of compost conditions. This study aims to design and implement a monitoring system based on Arduino Mega 2560 capable of measuring temperature, humidity, pH, and compost color during the composting process. The system utilizes a DS18B20 sensor for temperature measurement, a capacitive soil moisture sensor for humidity detection, a soil pH sensor for acidity level measurement, and a TCS3200 color sensor for compost color identification. The measurement data is displayed in real-time through a Nextion HMI and automatically stored on an SD Card with timestamp information provided by a DS1302 RTC module. Compost maturity determination is based on four parameters, namely temperature, humidity, pH, and color, using an AND logic method, where all parameters must meet the predetermined conditions for the compost to be classified as mature. System testing was conducted through 30 measurement trials for each sensor. The results show that the temperature sensor has an average error of 0.61°C, the humidity sensor has an average error of 0.93%, and the pH sensor shows a deviation of approximately 0.1–0.5 compared to reference values. The system is also capable of detecting compost color changes from green to dark brown as an indicator of maturity. Overall, the system performs well in real-time monitoring, data logging, and automatic compost maturity classification based on predefined parameters.

Keywords: Arduino Mega, compost maturity, compost color, multiparameter monitoring, pH.

**POLITEKNIK
NEGERI
JAKARTA**



DAFTAR ISI

HALAMAN SAMPUL/COVER	i
HALAMAN JUDUL	ii
HALAMAN PERNYATAAN ORISINALITAS.....	iii
HALAMAN PENGESAHAN TUGAS AKHIR.....	iv
KATA PENGANTAR.....	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	3
1.5 Luaran	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Pengomposan Sampah Organik	5
2.2 Kematangan Kompos	6
2.3 Sistem Monitoring Multiparameter.....	6
2.4 Parameter Kualitas dan Kematangan Kompos.....	6
2.5 Arduino Mega 2560	7
2.6 Sensor Suhu DS18B20	8
2.7 Sensor Kelembapan Tanah Kapasitif	8
2.8 Sensor pH tanah	9
2.9 Sensor Warna TCS3200	10
2.10 HMI Nextion	10
2.11 SD Card Module.....	11
2.12 RTC (Real Time Clock)	12
BAB III PERENCANAAN DAN REALISASI.....	13
3.1 Rancangan Alat	13
3.2 Realisasi Alat.....	23

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB IV PEMBAHASAN.....	39
4.1 Pengujian Sensor Suhu DS18B20.....	39
4.2 Pengujian Sensor Kelembapan Kapasitif.....	41
4.3 Pengujian Sensor pH.....	43
4.4 Pengujian Sensor Warna TCS3200	46
4.5 Pengujian HMI Nextion	48
4.6 Pengujian SD Card.....	51
4.7 Pengujian Sistem Monitoring Keseluruhan	53
BAB V PENUTUP.....	56
5.1 Simpulan	56
5.2 Saran.....	57
DAFTAR PUSTAKA.....	58
DAFTAR RIWAYAT HIDUP PENULIS	60
LAMPIRAN.....	xiii

**POLITEKNIK
NEGERI
JAKARTA**



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR GAMBAR

Gambar 2. 1 Proses Pengomposan	5
Gambar 2. 2 Arduino Mega 2560	7
Gambar 2. 3 Sensor Suhu DS18B20	8
Gambar 2. 4 Sensor Kapasitif	8
Gambar 2. 5 Sensor pH tanah	9
Gambar 2. 6 Sensor Warna TCS3200	10
Gambar 2. 7 HMI Nextion	10
Gambar 2. 8 SD Card Module	11
Gambar 2. 9 RTC 1302	12
Gambar 3. 1 Flowchart Sistem	16
Gambar 3. 2 Block Diagram Sistem	20
Gambar 3. 3 Diagram Blok Subsistem	21
Gambar 3. 4 Wiring Diagram	22
Gambar 3. 5 Realisasi Alat Sistem Otomasi Pengomposan	23
Gambar 3. 6 Flowchart Subsistem	28
Gambar 4. 1 Tampilan HMI Nextion	49
Gambar 4. 2 File LOG.CSV pada SD Card	52
Gambar 4. 3 Data Logger Hasil Monitoring pada Microsoft Excel	52

**POLITEKNIK
NEGERI
JAKARTA**



DAFTAR TABEL

Tabel 2. 1 Parameter Kualitas Kompos Berdasarkan SNI	6
Tabel 3. 1 Spesifikasi Alat.....	16
Tabel 3. 2 Spesifikasi Hardware pada Sistem Monitoring Multiparameter dan Deteksi Kematangan Kompos.....	18
Tabel 3. 3 Spesifikasi Software pada Sistem Monitoring Multiparameter dan Deteksi Kematangan Kompos.....	19
Tabel 3. 4 Durasi perpindahan halaman otomatis (auto page) pada Nextion.....	26
Tabel 3. 5 Program Akuisisi Data Sensor.....	29
Tabel 3. 6 Program Filter Median	30
Tabel 3. 7 Program Konversi Nilai pH.....	31
Tabel 3. 8 Program Pembacaan Nilai Pulse Sensor Warna TCS3200.....	32
Tabel 3. 9 Program Klasifikasi Warna Kompos	33
Tabel 3. 10 Program Deteksi Kematangan Kompos	34
Tabel 3. 11 Program Pencatatan Waktu (RTC)	35
Tabel 3. 12 Program Penyimpanan Data ke SD Card	36
Tabel 3. 13 Program Pengiriman Data ke HMI (Nextion).....	37
Tabel 4. 1 Pengujian Sensor Suhu.....	39
Tabel 4. 2 Pengujian Sensor Kelembapan Kapasitif.....	41
Tabel 4. 3 Pengujian Sensor pH	43
Tabel 4. 4 Pengujian Sensor Warna TCS3200	47
Tabel 4. 5 Pengujian Tampilan HMI Nextion	49
Tabel 4. 6 Pengujian Sistem Keseluruhan.....	54

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**POLITEKNIK
NEGERI
JAKARTA**



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR LAMPIRAN

Lampiran 1 Dokumentasi Alat	xiii
Lampiran 2 Kode Program.....	xiv
Lampiran 3 Desain HMI Nextion	lviii
Lampiran 4 Poster Pengujian Alat.....	lx
Lampiran 5 SOP Pengujian Alat	lxi



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB I PENDAHULUAN

1.1 Latar Belakang

Masalah sampah organik di Indonesia merupakan tantangan lingkungan yang signifikan, dengan volumenya meningkat secara konsisten setiap tahun. Jika tidak dikelola secara memadai, limbah ini tidak hanya menyebabkan polusi dan bau tidak sedap tetapi juga berfungsi sebagai habitat bagi mikroorganisme berbahaya, yang memerlukan solusi pengelolaan limbah berkelanjutan seperti pengomposan. (BPS, 2024) mencatat pengelolaan sampah dan limbah sebagai salah satu persoalan lingkungan yang belum tuntas. Sebagian besar sampah itu berupa sisa makanan, sayuran, daun, dan limbah dapur yang sebenarnya masih bisa diolah. (Widiatmanta & Santi, 2021) meneliti hal yang sama bahwa limbah rumah tangga didominasi bahan organik yang berpotensi diproses kembali.

Pengomposan adalah salah satu cara mengolah sampah organik tersebut. Prosesnya sederhana, yaitu mikroorganisme mengurai bahan organik hingga menghasilkan pupuk yang bisa memperbaiki kesuburan tanah. Dalam praktiknya, ada beberapa pendekatan yang umum dipakai, antara lain metode aerob dengan suplai oksigen, metode anaerob tanpa oksigen bebas, penguburan langsung di tanah, pengadukan mekanik, dan penambahan mikroorganisme untuk mempercepat dekomposisi. Pengadukan mekanik mulai banyak diterapkan agar dekomposisi berlangsung lebih merata. Komposter otomatis berbasis sensor juga mulai dikembangkan untuk memantau parameter pengomposan secara real-time (Handayani et al., 2023)

Pengomposan konvensional masih mengandalkan pengamatan manual, sehingga pemantauan parameter penting seperti suhu, kelembapan, dan pH belum dapat dilakukan secara konsisten dan real-time. Suhu, kelembapan, dan pH sangat memengaruhi kinerja mikroorganisme selama dekomposisi berlangsung. Ketika ketiga parameter ini tidak stabil, proses menjadi lebih lambat dan kualitas kompos menurun (Aprizal et al., 2025). Penentuan

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

kematangan kompos biasanya masih dilakukan secara visual, dan hasilnya tidak selalu konsisten. SNI 19-7030-2004 mendefinisikan kompos matang dari beberapa ciri: suhu mendekati suhu lingkungan, pH mendekati netral, warna coklat kehitaman, dan tidak berbau menyengat. Sensor dapat mengatasi masalah inkonsistensi itu. DS18B20, misalnya, memiliki akurasi pembacaan suhu yang baik dan terhubung ke mikrokontroler lewat komunikasi one-wire (Handayani et al., 2023). Perubahan warna kompos juga menjadi indikator kematangan yang dapat dideteksi oleh sensor warna.

Berdasarkan permasalahan tersebut, maka dirancang suatu sistem monitoring multiparameter dan deteksi kematangan kompos berbasis Arduino Mega 2560. Sistem ini menggunakan sensor suhu DS18B20, sensor kelembapan tanah kapasitif, sensor pH tanah, dan sensor warna TCS3200 untuk memantau kondisi kompos secara real-time. Selain itu, sistem dilengkapi dengan HMI Nextion sebagai media monitoring serta modul SD Card dan RTC sebagai media penyimpanan data pengomposan. Dengan adanya sistem ini diharapkan proses monitoring pengomposan dapat dilakukan secara lebih mudah, terukur, dan efisien sehingga membantu pengguna dalam mengetahui kondisi serta tingkat kematangan kompos secara lebih akurat.

1.2 Perumusan Masalah

Berdasarkan uraian latar belakang tersebut, perumusan masalah dalam Tugas Akhir ini disusun sebagai berikut.

1. Bagaimana merancang sistem monitoring multiparameter pada proses pengomposan berbasis Arduino Mega 2560?
2. Bagaimana mengintegrasikan sensor suhu DS18B20, sensor kelembapan tanah kapasitif, sensor pH tanah, dan sensor warna TCS3200 pada sistem monitoring pengomposan?
3. Bagaimana menampilkan data monitoring pengomposan secara real-time menggunakan HMI Nextion?
4. Bagaimana merealisasikan sistem penyimpanan data monitoring menggunakan modul SD Card dan RTC?



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

5. Bagaimana mendeteksi tingkat kematangan kompos berdasarkan parameter suhu, kelembapan, pH, dan warna?

1.3 Tujuan

Berdasarkan perumusan masalah yang telah ditetapkan, tujuan yang ingin dicapai dalam Tugas Akhir ini adalah sebagai berikut.

1. Merancang sistem monitoring multiparameter proses pengomposan berbasis Arduino Mega 2560.
2. Mengimplementasikan sensor suhu DS18B20, sensor kelembapan tanah kapasitif, sensor pH tanah, dan sensor warna TCS3200 pada sistem monitoring pengomposan.
3. Menampilkan data hasil monitoring parameter pengomposan secara real-time menggunakan HMI Nextion.
4. Merealisasikan sistem penyimpanan data monitoring menggunakan modul SD Card dan RTC.
5. Mendeteksi status kematangan kompos berdasarkan kombinasi parameter suhu, kelembapan, pH, dan warna menggunakan logika AND.

1.4 Batasan Masalah

Agar pembahasan Tugas Akhir lebih terarah dan tidak meluas, batasan masalah yang digunakan adalah sebagai berikut.

1. Pengujian menggunakan campuran bahan organik awal, sekam kering, dan kompos matang sebagai bahan pencampur. Tugas Akhir ini berfokus pada evaluasi fungsi monitoring, otomasi, dan deteksi kematangan dalam periode pengujian tertentu tanpa membahas keseluruhan proses dekomposisi hingga kompos matang.
2. Hasil monitoring ditampilkan melalui HMI Nextion dan disimpan ke dalam SD Card dengan informasi tanggal dan waktu dari modul RTC DS1302, tanpa menggunakan sistem monitoring berbasis Internet of Things (IoT).
3. Sistem dibatasi pada fungsi monitoring multiparameter dan deteksi status kematangan kompos berdasarkan hasil pembacaan sensor. Tugas Akhir ini



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

tidak membahas analisis kualitas kompos secara kimia maupun pengujian laboratorium.

1.5 Luaran

1. Prototype alat sistem monitoring dan deteksi kematangan kompos berbasis Arduino Mega 2560.
2. Laporan Tugas Akhir.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB V PENUTUP

5.1 Simpulan

1. Sistem monitoring multiparameter berbasis Arduino Mega 2560 berhasil dirancang dan direalisasikan untuk melakukan pemantauan kondisi kompos secara otomatis. Sistem mampu membaca parameter suhu, kelembapan, pH, dan warna kompos selama proses pengomposan secara terstruktur dan *real-time*.
2. Integrasi sensor DS18B20, sensor kelembapan tanah kapasitif, sensor pH tanah, dan sensor warna TCS3200 berhasil dilakukan pada sistem monitoring pengomposan. Masing-masing sensor mampu memberikan data pembacaan sesuai parameter yang diukur, meskipun sensor pH masih mengalami perubahan nilai pembacaan sehingga diperlukan proses kalibrasi yang lebih optimal untuk meningkatkan kestabilan dan akurasi pengukuran.
3. Sistem mampu menampilkan data hasil monitoring secara *real-time* melalui HMI Nextion. Data yang ditampilkan meliputi nilai suhu, kelembapan, pH, warna kompos, status aktuator, serta informasi proses pengomposan sehingga pengguna dapat melakukan pemantauan kondisi sistem dengan lebih mudah.
4. Sistem penyimpanan data monitoring berhasil direalisasikan menggunakan modul SD Card yang terintegrasi dengan RTC DS1302. Data hasil pembacaan sensor dapat disimpan dalam format CSV dengan informasi waktu pengambilan data, sehingga riwayat perubahan kondisi kompos dapat terdokumentasi selama proses pengomposan berlangsung.
5. Sistem mampu mendeteksi tingkat kematangan kompos berdasarkan empat parameter, yaitu suhu, kelembapan, pH, dan warna. Metode pendeteksian menggunakan logika *AND*, sehingga status kompos dinyatakan matang apabila seluruh parameter telah memenuhi batas kondisi yang telah ditentukan, yaitu suhu 30–40 °C, kelembapan 40–60%, pH 6,8–7,49, serta warna kompos mencapai kategori coklat gelap.



© Hak Cipta milik Politeknik Negeri Jakarta

5.2 Saran

1. Menambahkan fitur Internet of Things (IoT) sehingga data hasil monitoring dapat dipantau secara jarak jauh melalui website atau aplikasi tanpa harus melihat HMI secara langsung.
2. Menambahkan sensor lain yang mendukung proses pengomposan, seperti sensor gas MQ-135, sehingga kondisi kompos dapat dianalisis lebih komprehensif melalui parameter gas yang dihasilkan selama proses dekomposisi.
3. Melakukan pengujian sistem dalam jangka waktu yang lebih panjang, yaitu sekitar 14 hari hingga 1–2 bulan, agar perubahan parameter suhu, kelembapan, pH, dan warna dapat diamati secara lebih lengkap sesuai tahapan proses pengomposan serta meningkatkan validitas hasil penelitian.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR PUSTAKA

- Amal, R. I. (2020). *ANALISIS PEMBACAAN NILAI OUTPUT SENSOR D*.
- Aprizal, M., Wagya, A., Multimedia, S. B., Elektro, J. T., Jakarta, P. N., Siwabessy, J. P. G. A., & Depok, K. U. I. (2025). *Rancang Bangun Sistem Komposter Otomatis Berbasis IoT dengan Pemantauan Suhu , Kelembapan , dan pH secara Real Time*. 6, 345–351.
- BPS. (2024). Catalogue : 3305001. *Statistik Lingkungan Hidup Indonesia 2024*, 43(0216–6224).
- Chowdhury, S., Sen, S., & Janardhanan, S. (2022). *Comparative Analysis and Calibration of Low Cost Resistive and Capacitive Soil Moisture Sensor*. <http://arxiv.org/abs/2210.03019>
- Handayani, L., Pebralia, J., Amri, I., & Puji Lestari, A. (2023). Pengembangan Alat Ukur Kematangan Kompos Berbasis Arduino Atmega328. *Journal Online of Physics*, 8(2), 96–102. <https://doi.org/10.22437/jop.v8i2.24331>
- Isyanto, H., Jumail, J., Rahayu, R., & Firmansyah, N. (2021). Design of Monitoring Device for the Process of Organic Waste Decomposition into Compost Fertilizer and Plant Growth through Smartphones based on Internet of Things Smart Farming. *Journal of Electrical Technology UMY*, 5(2), 52–60. <https://doi.org/10.18196/jet.v5i2.12815>
- Lin, Y. H., Lin, Y. Z., & Lin, Y. H. (2022). Preliminary design for establishing compost maturity by using the spectral characteristics of five organic fertilizers. *Scientific Reports*, 12(1), 1–8. <https://doi.org/10.1038/s41598-022-19714-3>
- Muhammad Ferdiyansyah, Yoga Putra Pratama, & Emmidia Djonaedi. (2025). Analisis Pengaruh Jarak dan Kondisi Pencahayaan terhadap Alat Deteksi Warna dengan Sensor TCS3200 Berbasis Arduino. *Jurnal Riset Rumpun Ilmu Teknik*, 4(2), 88–96. <https://doi.org/10.55606/jurritek.v4i2.5556>
- Santoso, L., Sunarto, B., ... R. F.-I. J. I., & 2024, undefined. (2024). Perancangan Alat Pendeteksi Kualitas Udara Dan Sistem Filter Udara Dengan Antarmuka Visual HMI Nextion. *Ojs.Stttexmaco.Ac.Id LH Santoso, B Sunarto, R Fitri, I Permatasari INFOTEX: Jurnal Ilmiah Bidang Ilmu Teknik*, 2024•*ojs.Stttexmaco.Ac.Id*, 2(2), 2964–5352. <https://ojs.stttexmaco.ac.id/index.php/infotex/article/view/79>
- Susanta, M. H. (n.d.). *DESAIN SISTEM DATA LOGGER SUHU LINGKUNGAN BERBASIS*. 4(2).
- Tanah, K., & Suryana, T. (2021). *Capacitive Soil Moisture Sensor Untuk Mengukur*. 1–22.
- Veizaga Balta, W., Vega Torrejón, G., & Colque Ayaviri, J. C. (2025). Monitoreo y control de compostaje automatizado basadas en microcontroladores Raspberry pi, sensores de gases, temperatura, pH, humedad y placas de desarrollo. *Journal Boliviano de Ciencias*, 21(57), 75–93.



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

<https://doi.org/10.52428/20758944.v21i57.1326>

Widiatmanta, J., & Santi, I. H. (2021). Pemanfaatan Food Waste Disposer Berbasis Arduino Uno untuk Pembuatan Pupuk Kompos Limbah Rumah Tangga. *Briliant: Jurnal Riset Dan Konseptual*, 6(1), 163.
<https://doi.org/10.28926/briliant.v6i1.556>





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR RIWAYAT HIDUP PENULIS



Nabila

Anak kedua dari dua bersaudara. Lahir di Tangerang pada 30, Januari 2005. Lulus dari SDN Pasir Awi tahun 2017, SMPN 3 Cikupa tahun 2020, dan SMKN 1 Kab.Tangerang pada tahun 2023. Kuliah D3 Elektronika Industri, Jurusan Teknik Elektro Politeknik Negeri Jakarta.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LAMPIRAN

Lampiran 1

Dokumentasi Alat





Lampiran 2

Kode Program

```
#include <OneWire.h>
#include <DallasTemperature.h>
DeviceAddress alamatSensorSuhu;
#include <virtuabotixRTC.h>
#include <SD.h>
#include <SPI.h>

// ===== PIN DEFINITIONS =====
#define ONE_WIRE_BUS 22
#define PIN_KELEMBAPAN A3
#define PIN_PH A2
#define DMS_PIN 9
#define RELAY_POMPA 26
#define RELAY_BUZZER 27
#define PILOT_KUNING 25
#define PILOT_HIJAU 23
#define TCS_S0 45
#define TCS_S1 46
#define TCS_S2 47
#define TCS_S3 48
#define TCS_OUT 49
#define RTC_CLK 19
#define RTC_DAT 17
#define RTC_RST 18
#define SD_CS_PIN 53
#define MOTOR_RPWM 12
#define MOTOR_LPWM 13
#define MOTOR_R_EN 7
#define MOTOR_L_EN 8
#define BTN_START 3
#define BTN_RESET 11
#define BTN_STOP 10
#define NEXTION_SERIAL Serial3
#define NEXTION_BAUD 115200

// ===== PARAMETER PROSES=====
#define INTERVAL_SENSOR_DETIK 2
#define INTERVAL_LOG_DETIK 15

#define DURASI_ADUK_AWAL_MENIT 1
#define DURASI_ADUK_SETELAH_SIRAM_MENIT 1
#define DURASI_ADUK_TERJADWAL_MENIT 1

#define RH_MIN_MATANG 40.0
#define DURASI_POMPA_DETIK 26
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
#define DURASI_BUZZER_DETIK 10
// =====
// KALIBRASI pH DMS
// =====
#define PH_MIN_MATANG 6.8
#define PH_MAX_MATANG 7.49
#define PH_SLOPE -0.00702
#define PH_INTERCEPT 7.965
#define PH_TUNGGU_STABIL_MS 5000UL
#define PH_JUMLAH_SAMPEL 10
#define PH_SAMPLE_DELAY_MS 20
// KALIBRASI MOISTURE
#define ADC_KERING 515
#define ADC_BASAH 300
#define MOISTURE_JUMLAH_SAMPEL 10
#define MOISTURE_SAMPLE_DELAY_MS 10

#define MOTOR_PWM 255
#define DEBOUNCE_MS 50UL
// =====
// KALIBRASI SENSOR WARNA TCS3200
// =====
#define TCS_SAMPLE_COUNT 10
#define TCS_SAMPLE_DELAY_MS 5
#define TCS_PULSE_TIMEOUT_US 100000UL

#define PULSE_PUTIH_R 38UL
#define PULSE_PUTIH_G 36UL
#define PULSE_PUTIH_B 33UL
#define PULSE_HITAM_R 115UL
#define PULSE_HITAM_G 116UL
#define PULSE_HITAM_B 102UL

// [BARU] Parameter retry & health-check SD card
#define SD_INIT_MAX_RETRY 5
#define SD_INIT_RETRY_DELAY_MS 500UL
#define SD_HEALTHCHECK_MS 10000UL
#define SD_WRITE_MAX_RETRY 3

enum PolaritasTombol { TOMBOL_NO,
                      TOMBOL_NC };
#define BTN_START_POLARITAS TOMBOL_NO
#define BTN_RESET_POLARITAS TOMBOL_NO
#define BTN_STOP_POLARITAS TOMBOL_NC
// Jadwal pengadukan harian
const int JADWAL_ADUK_JAM[] = { 0, 4, 8, 12, 16, 20 };
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
const int JUMLAH_JADWAL_ADUK = sizeof(JADWAL_ADUK_JAM) /
sizeof(int);

// ===== OBJEK LIBRARY =====
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensorSuhu(&oneWire);
virtuabotixRTC myrtc(RTC_CLK, RTC_DAT, RTC_RST);
// ===== ENUM =====
enum SystemStatus { STANDBY,
                    RUNNING,
                    STOPPED,
                    FINISHED };
enum WarnaKompos { WARNA_HIJAU,
                  WARNA_HIJAU_KEKUNINGAN,
                  WARNA_KUNING,
                  WARNA_PUTIH,
                  WARNA_COKLAT_MUDA,
                  WARNA_COKLAT,
                  WARNA_COKLAT_GELAP };
enum MotorPurpose { MOTOR_NONE,
                   MOTOR_STIR_AWAL,
                   MOTOR_JADWAL,
                   MOTOR_SETELAH_SIRAM };

// Tahap Proses Awal
enum TahapAwal {
    TAHAP_STIR_AWAL,
    TAHAP_BACA_SENSOR_AWAL,
    TAHAP_CEK_KELEMBAPAN,
    TAHAP_POMPA_AKTIF,
    TAHAP_STIR_SETELAH_SIRAM,
    TAHAP_SELESAI_AWAL
};

// ===== VARIABEL GLOBAL =====
SystemStatus statusSistem = STANDBY;
WarnaKompos warnaKompos = WARNA_COKLAT_MUDA;
MotorPurpose motorPurpose = MOTOR_NONE;

bool prosesAktif = false;
bool komposMatang = false;
bool prosesAwalSelesai = false;

TahapAwal tahapAwal = TAHAP_STIR_AWAL;
bool aksiTahapDimulai = false;

int tahunMulai = 0, bulanMulai = 0, tanggalMulai = 0;
int jamMulai = 0, menitMulai = 0, detikMulai = 0;
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
int hariPengomposan = 0, jamSisaDurasi = 0, menitSisaDurasi = 0;

float suhu = -127.0;
float kelembapan = 0.0;
int adcKelembapanTerakhir = 0;
float pH = 0.0;
int adcPhTerakhir = 0;
bool suhuSensorError = false;
bool suhuSensorTerdeteksi = false;

unsigned long pulseR = 0, pulseG = 0, pulseB = 0;
int nilaiR = 0, nilaiG = 0, nilaiB = 0;
bool sensorWarnaError = false;

int hariIni = 0, bulanIni = 0, tahunIni = 0;
int jamIni = 0, menitIni = 0, detikIni = 0, hariMinggu = 0;

bool sdTersedia = false;
bool motorAktif = false;
bool pompaAktif = false;
bool buzzerAktif = false;

bool buzzerMatangSudahBunyi = false;
bool buzzerErrorSensorWarnaSudahBunyi = false;

unsigned long prevRtcMillis = 0;
unsigned long prevSensorMillis = 0;
unsigned long prevLogMillis = 0;
unsigned long motorStartMillis = 0, motorDurationMs = 0;
unsigned long pompaStartMillis = 0, pompaDurationMs = 0;
unsigned long buzzerStartMillis = 0, buzzerDurationMs = 0;
unsigned long prevNextionStoppedMillis = 0; // [BARU] throttle
update HMI saat STOPPED

unsigned long motorStopMillis = 0; // untuk cooldown noise
motor
unsigned long motorSisaMillis = 0;
unsigned long pompaSisaMillis = 0;

long tanggalAdukTerakhir = -1;
int jamAdukTerakhir = -1;

char namaFileLogAktif[20] = "";
bool fileLogSiap = false;

// [BARU] status & statistik SD card untuk diagnosa
unsigned long prevSdHealthCheckMillis = 0;
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
unsigned long jumlahGagalTulisSD = 0;
unsigned long jumlahSukesTulisSD = 0;
bool sdErrorSudahDilaporkan = false;

double akumulSuhu = 0.0, akumulKelembapan = 0.0, akumulPh = 0.0;
unsigned long jumlahSampelAkumulasi = 0;

bool startStable = false, resetStable = false;
bool startLastRaw = false, resetLastRaw = false;
unsigned long startDebounceMillis = 0, resetDebounceMillis = 0;

bool stopStable = false;
bool stopLastRaw = false;
unsigned long stopDebounceMillis = 0;
// ===== BUFFER RAM PAGE LOGDATA (7 baris)
=====
#define LOG_BUFFER_SIZE 7
struct LogEntry {
    char tanggal[9];
    char jam[6];
    float suhu;
    float kelembapan;
    float ph;
    char warna[20];
};
LogEntry logBuffer[LOG_BUFFER_SIZE];
int logBufferCount = 0;

// ===== WAVEFORM BUFFER =====
#define WAVEFORM_SUHU_ID 1
#define WAVEFORM_KELEMBAPAN_ID 12
#define WAVEFORM_PH_ID 13

// =====
// MEDIAN FILTER untuk sensor analog (meredam noise motor
BTS7960)
// =====
#define FILTER_SIZE 5

float bufferSuhu[FILTER_SIZE] = { 25, 25, 25, 25, 25 };
float bufferKelembapan[FILTER_SIZE] = { 50, 50, 50, 50, 50 };
float bufferPH[FILTER_SIZE] = { 7, 7, 7, 7, 7 };
int filterIdx = 0;

float medianFilter(float arr[], int size) {
    float temp[FILTER_SIZE];
    memcpy(temp, arr, size * sizeof(float));
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

for (int i = 0; i < size - 1; i++) {
    for (int j = 0; j < size - i - 1; j++) {
        if (temp[j] > temp[j + 1]) {
            float swap = temp[j];
            temp[j] = temp[j + 1];
            temp[j + 1] = swap;
        }
    }
}
return temp[size / 2];
}

// ===== PROTOTYPE =====

void initPins();
void initMotorPWM();
void initSensors();
void initSDCard();
bool cekKesehatanSD();
bool buatFileLogBaru();
void tulisHeaderCSV();
void bacaRTC();
void updateRTC();
void hitungDurasiPengomposan();
void bacaSensor();
float bacaKelembapanPersen();
float hitungPH();
int bacaAnalogRata(byte pin, byte jumlahSampel, unsigned int
jedaMs);
void bacaWarna();
unsigned long bacaPulseWarna(byte s2State, byte s3State);
int normalisasiPulse(unsigned long pulse, unsigned long
pulseMin, unsigned long pulseMax);
void klasifikasiWarna();
bool simpanDataSD();
void aktifkanPompa();
void pompaOFF();
void aktifkanMotor(unsigned long durasiMs, MotorPurpose
purpose);
void motorOFF();
void stopSementara(); // [BARU] ditambahkan ke daftar prototype
(implementasi sudah ada, sebelumnya belum dideklarasikan)
bool cekWaktuJadwalAduk();
bool belumDiadukJamIni();
void catatWaktuAdukTerakhir();
void cekKematanganKompos();
void masukFinished();
void aktifkanBuzzer(unsigned long durasiMs);
void bunyikanBuzzerMatang();

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
void bunyikanBuzzerErrorSensorWarna();
void buzzerOFF();
void updateAktuator();
void updateBuzzer();
void updatePilotLamp();
void handleTombol();
bool tombolPressed(byte pin, PolaritasTombol polaritas, bool
&stableState, bool &lastRawState, unsigned long
&lastDebounceMillis);
void tundaResponsif(unsigned long totalMs);
void mulaiProsesBaru();
void alurProsesAwal();
void alurMonitoringA();
void resetSistem();
void tampilkanSerial();
void tampilkanRingkasanAkhir();
void ubahStatusSistem(SystemStatus statusBaru); // [BARU] satu
pintu transisi state machine
const char *statusSistemText();
const char *warnaText();
const char *statusKomposText();
const char *statusOnOff(bool aktif);
const char *statusSDText();
void print2Digits(Print &out, int value);
void printTanggal(Print &out);
void printJam(Print &out);
bool isLeapYear(int year);
int jumlahHariBulan(int month, int year);
long tanggalKeNomorHari(int day, int month, int year);
long waktuKeDetikAbsolut(int day, int month, int year, int hour,
int minute, int second);
bool elapsed(unsigned long nowMs, unsigned long previousMs,
unsigned long intervalMs);

void nextionInit();
void nextionEndCmd();
void nextionSetText(const char *objName, const String &value);
void nextionUpdatePageMain();
void tambahKeBufferLog();
void nextionUpdatePageLogData();
void nextionUpdatePageSuhu();
void nextionUpdatePageKelembapan();
void nextionUpdatePagePH();
void nextionUpdatePageEnd();
void nextionCommand(const char *cmd);
void nextionGoPage(const char *pageName);
void nextionPaksaIconEnd();
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
byte suhuKeWaveform(float s); // [BARU]
ditambahkan ke daftar prototype (implementasi sudah ada,
sebelumnya belum dideklarasikan)
byte kelembapanKeWaveform(float rh); // [BARU]
byte phKeWaveform(float ph); // [BARU]
void nextionAddWaveformData(byte waveformId, byte channel, byte
value); // [BARU]
void resumeProses();
// SETUP -> kotak "INISIALISASI SISTEM ARDUINO SENSOR RTC SD
CARD HMI AKTUATOR"
void setup() {
    Serial.begin(115200);
    Serial.println(F("=== SISTEM OTOMASI PENGOMPOSAN - ARDUINO
MEGA 2560 ==="));
    Serial.println(F("=== (Alur sesuai diagram: MULAI -> masukkan
sampah -> inisialisasi -> TEKAN START) ==="));
    delay(2000);

    initPins();
    initMotorPWM();
    initSensors();
    delay(1000);

    bacaRTC();
    Serial.print(F("RTC Saat Startup : "));
    printTanggal(Serial);
    Serial.print(' ');
    printJam(Serial);
    Serial.println();

    initSDCard();
    nextionInit();

    bacaSensor();

    char tanggal[9];
    char jam[6];
    snprintf(tanggal, sizeof(tanggal), "%02d/%02d", hariIni,
bulanIni);
    snprintf(jam, sizeof(jam), "%02d:%02d", jamIni, menitIni);

    strcpy(logBuffer[0].tanggal, tanggal);
    strcpy(logBuffer[0].jam, jam);
    logBuffer[0].suhu = suhu;
    logBuffer[0].kelembapan = kelembapan;
    logBuffer[0].ph = pH;
    strcpy(logBuffer[0].warna, warnaText());
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
logBufferCount = 1;

ubahStatusSistem(STANDBY);
motorOFF();
pompaOFF();
buzzerOFF();

nextionUpdatePageMain();
nextionUpdatePageLogData();
nextionUpdatePageEnd();

Serial.println(F("Status awal: STANDBY. Masukkan sampah
organik lalu tekan START.));
}
// LOOP UTAMA
void loop() {
    unsigned long nowMs = millis();

    updateRTC();
    handleTombol();

    if (statusSistem == STOPPED) {
        // [DIPERBAIKI] Sebelumnya nextionUpdatePageMain() dipanggil di
        SETIAP
        // iterasi loop() tanpa jeda -> membanjiri Serial3 ke Nextion ->
        HMI
        // lag/macet, termasuk teks status "STOPPED" yang seharusnya
        tampil.
        // Sekarang dibatasi (throttle) sama seperti update HMI lainnya
        di kode ini.
        if (prevNextionStoppedMillis == 0 || elapsed(nowMs,
        prevNextionStoppedMillis, 250UL)) {
            prevNextionStoppedMillis = nowMs;
            nextionUpdatePageMain();
        }
        updatePilotLamp();
        return;
    }

    updateAktuator();
    updateBuzzer();
    updatePilotLamp();

    // [BARU] Health-check SD card berkala, TIDAK PEDULI status
    sistem,
    // supaya kartu yang lepas kontak / gagal init bisa terdeteksi
    dan
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
// dicoba disambung ulang secara otomatis, bukan hanya sekali
di setup().
    if (prevSdHealthCheckMillis == 0 || elapsed(nowMs,
prevSdHealthCheckMillis, SD_HEALTHCHECK_MS)) {
        prevSdHealthCheckMillis = nowMs;
        cekKesehatanSD();
    }

    if (prevSensorMillis == 0 || elapsed(nowMs, prevSensorMillis,
(unsigned long)INTERVAL_SENSOR_DETIK * 1000UL)) {
        prevSensorMillis = nowMs;
        bacaSensor();
        tampilkanSerial();
    }

    if (statusSistem == RUNNING) {
        static unsigned long prevNextionMain = 0;
        if (prevNextionMain == 0 || elapsed(nowMs,
prevNextionMain, 250UL)) {
            prevNextionMain = nowMs;
            nextionUpdatePageMain();
            nextionUpdatePageSuhu();
            nextionUpdatePageKelembapan();
            nextionUpdatePagePH();
        }
        static unsigned long prevNextionEnd = 0;
        if (prevNextionEnd == 0 || elapsed(nowMs, prevNextionEnd,
1000UL)) {
            prevNextionEnd = nowMs;
            nextionUpdatePageEnd();
        }
        if (statusSistem == RUNNING) {
            if (prevLogMillis == 0 || elapsed(nowMs, prevLogMillis,
(unsigned long)INTERVAL_LOG_DETIK * 1000UL)) {
                prevLogMillis = nowMs;
                simpanDataSD();
                tambahKeBufferLog();
                nextionUpdatePageLogData();
                if (prosesAwalSelesai) {
                    cekKematanganKompos();
                }
            }

            if (!prosesAwalSelesai) {
                alurProsesAwal();
            } else {
                alurMonitoringA();
            }
        }
    }
}
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
    }  
  }  
}  
  
// =====  
// PILOT LAMP  
// =====  
void updatePilotLamp() {  
  switch (statusSistem) {  
  
    case STANDBY:  
      digitalWrite(PILOT_KUNING, HIGH);  
      digitalWrite(PILOT_HIJAU, LOW);  
      break;  
  
    case STOPPED: {  
      // [BARU] kedip tiap 300ms supaya beda dari STANDBY (nyala tetap)  
      static unsigned long prevBlinkMillis = 0;  
      static bool ledState = false;  
      unsigned long nowMs = millis();  
      if (prevBlinkMillis == 0 || elapsed(nowMs, prevBlinkMillis, 300UL)) {  
        prevBlinkMillis = nowMs;  
        ledState = !ledState;  
        digitalWrite(PILOT_KUNING, ledState ? HIGH : LOW);  
      }  
      digitalWrite(PILOT_HIJAU, LOW);  
      break;  
    }  
  
    case RUNNING:  
    case FINISHED:  
      digitalWrite(PILOT_KUNING, LOW);  
      digitalWrite(PILOT_HIJAU, HIGH);  
      break;  
  }  
}  
  
//  
=====  
// INISIALISASI PIN  
//  
=====  
void initPins() {  
  pinMode(BTN_START, INPUT_PULLUP);
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
pinMode(BTN_RESET, INPUT_PULLUP);
pinMode(BTN_STOP, INPUT_PULLUP);

pinMode(RELAY_POMPA, OUTPUT);
digitalWrite(RELAY_POMPA, HIGH);
pinMode(RELAY_BUZZER, OUTPUT);
digitalWrite(RELAY_BUZZER, HIGH);

pinMode(PILOT_KUNING, OUTPUT);
digitalWrite(PILOT_KUNING, HIGH);
pinMode(PILOT_HIJAU, OUTPUT);
digitalWrite(PILOT_HIJAU, LOW);

pinMode(MOTOR_RPWM, OUTPUT);
pinMode(MOTOR_LPWM, OUTPUT);
pinMode(MOTOR_R_EN, OUTPUT);
pinMode(MOTOR_L_EN, OUTPUT);

pinMode(TCS_S0, OUTPUT);
digitalWrite(TCS_S0, HIGH);
pinMode(TCS_S1, OUTPUT);
digitalWrite(TCS_S1, LOW);
pinMode(TCS_S2, OUTPUT);
pinMode(TCS_S3, OUTPUT);
pinMode(TCS_OUT, INPUT);

pinMode(DMS_PIN, OUTPUT);
digitalWrite(DMS_PIN, HIGH);
}

//
=====
// PWM TIMER1 -> ~62.5 kHz supaya noise BTS7960 lebih mudah
diredam
//
=====

void initMotorPWM() {
  TCCR1A = _BV(COM1A1) | _BV(COM1B1) | _BV(WGM10);
  TCCR1B = _BV(WGM12) | _BV(CS10);
  Serial.println(F("[MOTOR] Timer1 PWM diset ke ~62.5 kHz
(preScaler=1, Fast PWM 8-bit)."));
}

void initSensors() {
  sensorSuhu.begin();
  Serial.println(F("Sensor suhu DS18B20 siap."));
}
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
// ===== KONSTANTA FILE
LOG=====

const char LOG_FILE_PREFIX[] = "LOG"; // 3 karakter, aman
untuk format 8.3
const char LOG_FILE_SUFFIX[] = ".CSV";
#define LOG_FILE_MAX_INDEX 999
void initSDCard() {
    pinMode(SD_CS_PIN, OUTPUT);
    digitalWrite(SD_CS_PIN, HIGH);
    SPI.begin();

    sdTersedia = false;
    for (int i = 1; i <= SD_INIT_MAX_RETRY; i++) {
        Serial.print(F("Mencoba inisialisasi SD Card...
percobaan "));
        Serial.print(i);
        Serial.print('/');
        Serial.println(SD_INIT_MAX_RETRY);

        if (SD.begin(SD_CS_PIN)) {
            sdTersedia = true;
            Serial.println(F("SD Card OK."));
            return;
        }
        delay(SD_INIT_RETRY_DELAY_MS);
    }

    sdTersedia = false;
    Serial.println(F("PERINGATAN: SD Card TIDAK terdeteksi
setelah beberapa percobaan."));
}

bool cekKesehatanSD() {
    if (sdTersedia) return true;

    Serial.println(F("[SD] Kartu tidak tersedia, mencoba
reconnect..."));
    if (SD.begin(SD_CS_PIN)) {
        sdTersedia = true;
        sdErrorSudahDilaporkan = false;
        Serial.println(F("[SD] Reconnect BERHASIL."));

        if (statusSistem == RUNNING && !fileLogSiap) {
            buatFileLogBaru();
        }
        return true;
    }
}
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
        if (!sdErrorSudahDilaporkan) {
            Serial.println(F("[SD] Reconnect GAGAL. Data sensor
tetap jalan, tidak tersimpan ke SD.));
            sdErrorSudahDilaporkan = true;
        }
        return false;
    }

    // [DIPERBAIKI] sekarang benar-benar memakai LOG_FILE_PREFIX
    & LOG_FILE_SUFFIX,
    // bukan hardcoded "LOG%03d.CSV" seperti sebelumnya ->
    konstanta jadi berguna,
    // bukan sekadar deklarasi mati.
    bool buatFileLogBaru() {
        fileLogSiap = false;

        if (!sdTersedia) {
            if (!cekKesehatanSD()) {
                Serial.println(F("[SD] Gagal membuat file log: SD Card
tidak tersedia.));
                return false;
            }
        }

        for (int idx = 1; idx <= LOG_FILE_MAX_INDEX; idx++) {
            char namaFile[13];
            snprintf(namaFile, sizeof(namaFile), "%s%03d%s",
LOG_FILE_PREFIX, idx, LOG_FILE_SUFFIX);

            if (!SD.exists(namaFile)) {
                strncpy(namaFileLogAktif, namaFile,
sizeof(namaFileLogAktif));

                File f = SD.open(namaFileLogAktif, FILE_WRITE);
                if (!f) {
                    Serial.print(F("[SD] Gagal membuat file: "));
                    Serial.println(namaFileLogAktif);
                    sdTersedia = false;
                    return false;
                }
                f.close();
                fileLogSiap = true;
                Serial.print(F("[SD] File dibuat: "));
                Serial.println(namaFileLogAktif);
                tulisHeaderCSV();
                return true;
            }
        }
    }
}
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
    }  
}  
  
    Serial.println(F("[SD] Nomor file penuh (index 1-999 sudah  
terpakai semua)."));  
    return false;  
}  
  
void tulisHeaderCSV() {  
    if (!sdTersedia || !fileLogSiap) return;  
    File dataFile = SD.open(namaFileLogAktif, FILE_WRITE);  
    if (!dataFile) {  
        Serial.println(F("[SD] GAGAL: tidak bisa membuka file  
CSV untuk header."));  
        sdTersedia = false;  
        return;  
    }  
    if (dataFile.size() == 0) {  
        dataFile.println(F("Hari,Tanggal,Jam,StatusSistem,Suhu,K  
elembapan,pH,"  
                            "PulseR,PulseG,PulseB,R,G,B,Warna,"  
                            "StatusMotor,StatusPompa,StatusBuzzer,  
StatusKompos,"  
                            "ADC_Kelembapan,ADC_pH"));  
    }  
    dataFile.close();  
}  
  
bool simpanDataSD() {  
    if (!sdTersedia) {  
        if (!cekKesehatanSD()) {  
            jumlahGagalTulisSD++;  
            Serial.println(F("[SD] GAGAL simpan data: SD Card  
tidak tersedia."));  
            return false;  
        }  
    }  
    if (!fileLogSiap) {  
        Serial.println(F("[SD] File log belum siap, mencoba  
membuat file baru..."));  
        if (!buatFileLogBaru()) {  
            jumlahGagalTulisSD++;  
            Serial.println(F("[SD] GAGAL simpan data: file log  
tidak bisa dibuat."));  
            return false;  
        }  
    }  
}
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
bool berhasil = false;
for (int percobaan = 1; percobaan <= SD_WRITE_MAX_RETRY &&
!berhasil; percobaan++) {
    File dataFile = SD.open(namaFileLogAktif, FILE_WRITE);
    if (!dataFile) {
        Serial.print(F("[SD] GAGAL buka file CSV (percobaan
"));

        Serial.print(percobaan);
        Serial.println(F(")."));
        sdTersedia = false;
        if (cekKesehatanSD()) buatFileLogBaru();
        continue;
    }

    dataFile.print(hariPengomposan);
    dataFile.print(',');
    printTanggal(dataFile);
    dataFile.print(',');
    printJam(dataFile);
    dataFile.print(',');
    dataFile.print(statusSistemText());
    dataFile.print(',');
    dataFile.print(suhu, 2);
    dataFile.print(',');
    dataFile.print(kelembapan, 2);
    dataFile.print(',');
    dataFile.print(pH, 2);
    dataFile.print(',');
    dataFile.print(pulseR);
    dataFile.print(',');
    dataFile.print(pulseG);
    dataFile.print(',');
    dataFile.print(pulseB);
    dataFile.print(',');
    dataFile.print(nilaiR);
    dataFile.print(',');
    dataFile.print(nilaiG);
    dataFile.print(',');
    dataFile.print(nilaiB);
    dataFile.print(',');
    dataFile.print(warnaText());
    dataFile.print(',');
    dataFile.print(statusOnOff(motorAktif));
    dataFile.print(',');
    dataFile.print(statusOnOff(pompaAktif));
    dataFile.print(',');
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

dataFile.print(statusOnOff(buzzerAktif));
dataFile.print(',');
dataFile.print(statusKomposText());
dataFile.print(',');
dataFile.print(adcKelembapanTerakhir);
dataFile.print(',');
dataFile.println(adcPhTerakhir);

dataFile.close();
berhasil = true;
}

if (berhasil) {
    jumlahSukesTulisSD++;
    Serial.print(F("[SD] Data tersimpan ke "));
    Serial.println(namaFileLogAktif);
} else {
    jumlahGagalTulisSD++;
    Serial.println(F("[SD] GAGAL total menyimpan data
setelah beberapa percobaan."));
}
return berhasil;
}
//
=====
// RTC DAN PERHITUNGAN DURASI
//
=====

void bacaRTC() {
    myrtc.updateTime();
    detikIni = myrtc.seconds;
    menitIni = myrtc.minutes;
    jamIni = myrtc.hours;
    hariMinggu = myrtc.dayofweek;
    hariIni = myrtc.dayofmonth;
    bulanIni = myrtc.month;
    tahunIni = myrtc.year;
}

void updateRTC() {
    unsigned long nowMs = millis();
    if (prevRtcMillis == 0 || elapsed(nowMs, prevRtcMillis,
200UL)) {
        prevRtcMillis = nowMs;
        bacaRTC();
        hitungDurasiPengomposan();
    }
}

```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
}

long waktuKeDetikAbsolut(int day, int month, int year, int
hour, int minute, int second) {
    long nomorHari = tanggalKeNomorHari(day, month, year);
    if (nomorHari <= 0) return -1;
    return (nomorHari * 86400L) + ((long)hour * 3600L) +
((long)minute * 60L) + (long)second;
}

void hitungDurasiPengomposan() {
    if (!prosesAktif) {
        hariPengomposan = jamSisaDurasi = menitSisaDurasi = 0;
        return;
    }

    long detikMulaiAbs = waktuKeDetikAbsolut(tanggalMulai,
bulanMulai, tahunMulai, jamMulai, menitMulai, detikMulai);
    long detikSekarangAbs = waktuKeDetikAbsolut(hariIni,
bulanIni, tahunIni, jamIni, menitIni, detikIni);

    if (detikMulaiAbs < 0 || detikSekarangAbs < 0 ||
detikSekarangAbs < detikMulaiAbs) {
        hariPengomposan = jamSisaDurasi = menitSisaDurasi = 0;
        return;
    }

    long selisihDetik = detikSekarangAbs - detikMulaiAbs;
    hariPengomposan = (int)(selisihDetik / 86400L);
    long sisaSetelahHari = selisihDetik % 86400L;
    jamSisaDurasi = (int)(sisaSetelahHari / 3600L);
    menitSisaDurasi = (int)((sisaSetelahHari % 3600L) / 60L);
}

bool isLeapYear(int year) {
    if (year % 400 == 0) return true;
    if (year % 100 == 0) return false;
    return (year % 4 == 0);
}

int jumlahHariBulan(int month, int year) {
    static const int hariPerBulan[] = { 31, 28, 31, 30, 31,
30, 31, 31, 30, 31, 30, 31 };
    if (month < 1 || month > 12) return 0;
    if (month == 2) return isLeapYear(year) ? 29 : 28;
    return hariPerBulan[month - 1];
}
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
long tanggalKeNomorHari(int day, int month, int year) {
    if (year < 2000 || month < 1 || month > 12 || day < 1)
return 0;
    if (day > jumlahHariBulan(month, year)) return 0;
    long totalHari = 0;
    for (int y = 2000; y < year; y++) totalHari +=
isLeapYear(y) ? 366 : 365;
    for (int m = 1; m < month; m++) totalHari +=
jumlahHariBulan(m, year);
    totalHari += day;
    return totalHari;
}

void bacaSensor() {
    sensorSuhu.requestTemperatures();
    float suhuBaca = sensorSuhu.getTempCByIndex(0);

    // [DIPERBAIKI] retry sampai 3x kalau gagal/di luar
rentang wajar DS18B20,
    // karena kegagalan baca kadang cuma sesaat (noise motor,
jalur kabel panjang)
    byte percobaanSuhu = 0;
    while ((suhuBaca == DEVICE_DISCONNECTED_C || suhuBaca < -
55.0 || suhuBaca > 125.0)
        && percobaanSuhu < 3) {
        percobaanSuhu++;
        delay(50);
        sensorSuhu.requestTemperatures();
        suhuBaca = sensorSuhu.getTempCByIndex(0);
    }

    if (suhuBaca == DEVICE_DISCONNECTED_C || suhuBaca < -55.0
|| suhuBaca > 125.0) {
        suhuSensorError = true;
        Serial.println(F("PERINGATAN: DS18B20 tidak terbaca
setelah 3x percobaan."));
        // suhu TIDAK diupdate -> tetap pakai nilai valid
terakhir, bukan ikut rusak
    } else {
        suhuSensorError = false;
        bufferSuhu[filterIdx] = suhuBaca;
        suhu = medianFilter(bufferSuhu, FILTER_SIZE);
    }

    float rhBaca = bacaKelembapanPersen();
    bufferKelembapan[filterIdx] = rhBaca;
    kelembapan = medianFilter(bufferKelembapan, FILTER_SIZE);
}
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
float phBaca = hitungPH();
bufferPH[filterIdx] = phBaca;
pH = medianFilter(bufferPH, FILTER_SIZE);

filterIdx = (filterIdx + 1) % FILTER_SIZE;

bacaWarna();
if (statusSistem == RUNNING && !komposMatang) {
    if (suhu > -100.0) {
        akumulSuhu += suhu;
        akumulKelembapan += kelembapan;
        akumulPh += pH;
        jumlahSampelAkumulasi++;
    }
}

float bacaKelembapanPersen() {
    int adc = bacaAnalogRata(PIN_KELEMBAPAN,
MOISTURE_JUMLAH_SAMPEL, MOISTURE_SAMPLE_DELAY_MS);
    adcKelembapanTerakhir = adc;
    float persen = ((float)(ADC_KERING - adc) * 85.0) /
(float)(ADC_KERING - ADC_BASAH);
    return constrain(persen, 15.0, 100.0);
}

float hitungPH() {
    digitalWrite(DMS_PIN, LOW);
    tundaResponsif(PH_TUNGGU_STABIL_MS);
    int adc = bacaAnalogRata(PIN_PH, PH_JUMLAH_SAMPEL,
PH_SAMPLE_DELAY_MS);
    digitalWrite(DMS_PIN, HIGH);
    adcPhTerakhir = adc;
    static int adcTerakhirValid = 110;
    if (adc > 5) {
        adcTerakhirValid = adc;
    } else {
        adc = adcTerakhirValid;
        adcPhTerakhir = adc;
    }

    float phValue = (PH_SLOPE * adc) + PH_INTERCEPT;

    return constrain(phValue, 0.0, 14.0);
}
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
int bacaAnalogRata(byte pin,
                    byte jumlahSampel,
                    unsigned int jedaMs)
{
    long total = 0;

    for(byte i=0;i<jumlahSampel;i++)
    {
        total += analogRead(pin);

        unsigned long t = millis();

        while(millis()-t < jedaMs)
        {
            handleTombol();

            if(statusSistem==STOPPED)
            {
                return total/(i+1);
            }

            updateAktuator();
            updateBuzzer();

            delay(1);
        }
    }

    return total/jumlahSampel;
}

//
=====
// SENSOR WARNA TCS3200
//
=====

void bacaWarna() {
    sensorWarnaError = false;

    pulseR = bacaPulseWarna(LOW, LOW);
    pulseG = bacaPulseWarna(HIGH, HIGH);
    pulseB = bacaPulseWarna(LOW, HIGH);

    if (pulseR == 0 || pulseG == 0 || pulseB == 0) {
        sensorWarnaError = true;
        Serial.println(F("ERROR SENSOR WARNA (TCS3200 tidak
        terbaca)."));
        if (statusSistem == RUNNING && !motorAktif) {
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
        bunyikanBuzzerErrorSensorWarna();
    }
    return;
}

buzzerErrorSensorWarnaSudahBunyi = false;
nilaiR = (int)pulseR;
nilaiG = (int)pulseG;
nilaiB = (int)pulseB;

klasifikasiWarna();
}

unsigned long bacaPulseWarna(byte s2State, byte s3State) {
    unsigned long total = 0;
    byte validSamples = 0;
    digitalWrite(TCS_S2, s2State);
    digitalWrite(TCS_S3, s3State);
    tundaResponsif(5);

    for (byte i = 0; i < TCS_SAMPLE_COUNT; i++) {
        unsigned long pulse = pulseIn(TCS_OUT, LOW,
TCS_PULSE_TIMEOUT_US);
        if (pulse > 0) {
            total += pulse;
            validSamples++;
        }
        tundaResponsif(TCS_SAMPLE_DELAY_MS);
    }

    return validSamples == 0 ? 0 : total / validSamples;
}

int normalisasiPulse(unsigned long pulse, unsigned long
pulseMin, unsigned long pulseMax) {
    if (pulseMin >= pulseMax) return 0;
    if (pulse <= pulseMin) return 255;
    if (pulse >= pulseMax) return 0;
    return (int)map((long)pulse, (long)pulseMin,
(long)pulseMax, 255, 0);
}

struct WarnaRef {
    WarnaKompos kategori;
    unsigned long r;
    unsigned long g;
    unsigned long b;
}
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
};

WarnaRef referensiWarna[] = {
    { WARNA_PUTIH, 53, 51, 47 },
    { WARNA_HIJAU, 54, 45, 55 },
    { WARNA_HIJAU_KEKUNINGAN, 54, 45, 55 }, // sementara
    { WARNA_KUNING, 29, 34, 46 },
    { WARNA_COKLAT_MUDA, 56, 62, 59 },
    { WARNA_COKLAT, 60, 66, 62 },
    { WARNA_COKLAT_GELAP, 73, 70, 63 }
};

const int JUMLAH_REFERENSI_WARNA = sizeof(referensiWarna) /
sizeof(WarnaRef);

void klasifikasiWarna() {
    float jarakTerkecil = 1e12;
    WarnaKompos kategoriTerdekat = WARNA_COKLAT_MUDA;

    for (int i = 0; i < JUMLAH_REFERENSI_WARNA; i++) {
        float dR = (float)pulseR - (float)referensiWarna[i].r;
        float dG = (float)pulseG - (float)referensiWarna[i].g;
        float dB = (float)pulseB - (float)referensiWarna[i].b;

        float jarakKuadrat = (dR * dR) + (dG * dG) + (dB * dB);

        if (jarakKuadrat < jarakTerkecil) {
            jarakTerkecil = jarakKuadrat;
            kategoriTerdekat = referensiWarna[i].kategori;
        }
    }

    warnaKompos = kategoriTerdekat;
}

//
// =====
// POMPA
//
// =====

void aktifkanPompa() {
    digitalWrite(RELAY_POMPA, LOW);
    pompaAktif = true;
    pompaStartMillis = millis();
    pompaDurationMs = (unsigned long)DURASI_POMPA_DETIK *
1000UL;
    Serial.print(F("[DIAGRAM] POMPA AKTIF selama "));
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
Serial.print(DURASI_POMPA_DETIK);
Serial.println(F(" detik."));
}

void pompaOFF() {
    digitalWrite(RELAY_POMPA, HIGH);
    pompaAktif = false;
}
//

=====

// MOTOR
//

=====

void aktifkanMotor(unsigned long durasiMs, MotorPurpose
purpose) {
    digitalWrite(MOTOR_R_EN, HIGH);
    digitalWrite(MOTOR_L_EN, HIGH);
    analogWrite(MOTOR_LPWM, MOTOR_PWM);
    analogWrite(MOTOR_RPWM, 0);
    motorAktif = true;
    motorStartMillis = millis();
    motorDurationMs = durasiMs;
    motorPurpose = purpose;
    motorStopMillis = 0;

    Serial.print(F("[MOTOR] ON - durasi: "));
    Serial.print(durasiMs / 1000UL);
    Serial.println(F(" detik."));
}

void motorOFF() {
    analogWrite(MOTOR_RPWM, 0);
    analogWrite(MOTOR_LPWM, 0);
    digitalWrite(MOTOR_R_EN, LOW);
    digitalWrite(MOTOR_L_EN, LOW);
    motorAktif = false;
    motorStopMillis = millis();

    Serial.println(F("[MOTOR] OFF - cooldown sensor
dimulai."));
}

void stopSementara()
{
    prosesAktif = false;
    Serial.println(F("=== SISTEM DI-PAUSE ==="));
}
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
    if (motorAktif) motorSisaMillis = motorDurationMs - (millis() - motorStartMillis);
    if (pompaAktif) pompaSisaMillis = pompaDurationMs - (millis() - pompaStartMillis);

    motorOFF();
    pompaOFF();
    buzzerOFF();

    ubahStatusSistem(STOPPED);

    nextionGoPage("Main");
    nextionUpdatePageMain();
    nextionUpdatePageLogData();
    nextionUpdatePageEnd();

    prevNextionStoppedMillis = millis(); // [BARU] sinkronkan throttle setelah update manual di atas
}

void resumeProses()
{
    prosesAktif = true;
    Serial.println(F("=== RESUME ==="));

    if(motorSisaMillis > 0)
    {
        aktifkanMotor(motorSisaMillis,motorPurpose);
        motorSisaMillis = 0;
    }

    if(pompaSisaMillis > 0)
    {
        digitalWrite(RELAY_POMPA, LOW);
        pompaAktif = true;
        pompaStartMillis = millis();
        pompaDurationMs = pompaSisaMillis;
        pompaSisaMillis = 0;
    }
    ubahStatusSistem(RUNNING);
    nextionUpdatePageMain();
}

//
=====
// UPDATE AKTUATOR (non-blocking timer utk motor & pompa)
//
=====
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
void updateAktuator() {
    unsigned long nowMs = millis();

    if (pompaAktif && elapsed(nowMs, pompaStartMillis,
pompaDurationMs)) {
        pompaOFF();
        Serial.println(F("Pompa OFF."));
    }

    if (motorAktif && elapsed(nowMs, motorStartMillis,
motorDurationMs)) {
        motorOFF();
        Serial.println(F("Motor pengaduk selesai."));
    }
}

bool cekWaktuJadwalAduk() {
    for (int i = 0; i < JUMLAH_JADWAL_ADUK; i++) {
        if (jamIni == JADWAL_ADUK_JAM[i]) return true;
    }
    return false;
}

bool belumDiadukJamIni() {
    long tanggalSekarang = tanggalKeNomorHari(hariIni,
bulanIni, tahunIni);
    if (tanggalSekarang <= 0) return false;
    return !(tanggalAdukTerakhir == tanggalSekarang &&
jamAdukTerakhir == jamIni);
}

void catatWaktuAdukTerakhir() {
    tanggalAdukTerakhir = tanggalKeNomorHari(hariIni,
bulanIni, tahunIni);
    jamAdukTerakhir = jamIni;
}

// KEMATANGAN KOMPOS (label B)
void cekKematanganKompos() {
    bool suhuOK =
        (suhu >= 30 && suhu <= 40);
    bool kelembapanOK =
        (kelembapan >= 40 && kelembapan <= 60);
    bool phOK =
        (pH >= PH_MIN_MATANG && pH <= PH_MAX_MATANG);
    bool warnaOK =
        (warnaKompos == WARNA_COKLAT_GELAP);
    if (
        suhuOK && kelembapanOK && phOK && warnaOK) {
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
        komposMatang = true;
        masukFinished();
        return;
    }
    komposMatang = false;
}
void masukFinished() {
    ubahStatusSistem(FINISHED);
    prosesAktif = false;

    // Hentikan aktuator yang mungkin masih berjalan (mis.
motor jadwal RTC)
    motorOFF();
    pompaOFF();

    // [DIAGRAM] AKTIFKAN BUZZER 10 DETIK (saat matang) -
hanya sekali
    bunyikanBuzzerMatang();

    // Simpan data final & ringkasan sebelum sistem menunggu
RESET
    simpanDataSD();
    tambahKeBufferLog();
    tampilkanRingkasanAkhir();

    // Update HMI supaya halaman End menampilkan status MATANG
& ikon checklist
    nextionUpdatePageMain();
    nextionUpdatePageLogData();
    nextionUpdatePageEnd();

    Serial.println(F("[DIAGRAM] KOMPOS MATANG -> SELESAI
(FINISHED). Tekan RESET untuk memulai ulang.));
}
//
=====
// BUZZER
//
=====

void aktifkanBuzzer(unsigned long durasiMs) {
    if (durasiMs == 0) return;

    digitalWrite(RELAY_BUZZER, LOW);
    buzzerAktif = true;
    buzzerStartMillis = millis();
    buzzerDurationMs = durasiMs;
}
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
void bunyikanBuzzerMatang() {
    if (buzzerMatangSudahBunyi) return;

    buzzerMatangSudahBunyi = true;
    aktifkanBuzzer((unsigned long)DURASI_BUZZER_DETIK *
1000UL);
    Serial.println(F("Buzzer MATANG aktif sekali (10
detik)."));
}

void bunyikanBuzzerErrorSensorWarna() {
    if (buzzerErrorSensorWarnaSudahBunyi) return;

    buzzerErrorSensorWarnaSudahBunyi = true;
    aktifkanBuzzer(1000UL);
    Serial.println(F("Buzzer ERROR sensor warna aktif
sekali."));
}

void buzzerOFF() {
    digitalWrite(RELAY_BUZZER, HIGH);
    buzzerAktif = false;
}

void updateBuzzer() {
    if (buzzerAktif && elapsed(millis(), buzzerStartMillis,
buzzerDurationMs)) {
        buzzerOFF();
        Serial.println(F("Buzzer OFF."));
    }
}

void ubahStatusSistem(SystemStatus statusBaru) {
    if (statusSistem == statusBaru) return;
    const char *dariText = statusSistemText();
    statusSistem = statusBaru;
    Serial.print(F("[STATE] "));
    Serial.print(dariText);
    Serial.print(F(" -> "));
    Serial.println(statusSistemText());
}

void handleTombol() {

    // [DEBUG - sementara] cek raw pin & status tombol STOP
    static unsigned long prevDebugMillis = 0;
    if (prevDebugMillis == 0 || millis() - prevDebugMillis >
500) {
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
prevDebugMillis = millis();
Serial.print(F("[DEBUG] Raw BTN_STOP pin = "));
Serial.print(digitalRead(BTN_STOP));
Serial.print(F(" | stopStable = "));
Serial.print(stopStable);
Serial.print(F(" | statusSistem = "));
Serial.println(statusSistemText());
}

bool startPressed = tombolPressed(
    BTN_START,
    BTN_START_POLARITAS,
    startStable,
    startLastRaw,
    startDebounceMillis);

bool stopPressed = tombolPressed(
    BTN_STOP,
    BTN_STOP_POLARITAS,
    stopStable,
    stopLastRaw,
    stopDebounceMillis);

bool resetPressed = tombolPressed(
    BTN_RESET,
    BTN_RESET_POLARITAS,
    resetStable,
    resetLastRaw,
    resetDebounceMillis);

// STOP bisa kapan saja selama RUNNING
if(statusSistem == RUNNING && stopPressed){
    stopSementara();
    return;
}

switch(statusSistem){

    case STANDBY:

        if(startPressed){
            mulaiProsesBaru();
        }
        break;
    case RUNNING:
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
// kalau ingin reset saat proses berjalan
if(resetPressed){
    resetSistem();
}
break;
case STOPPED:
if(startPressed)
{
    resumeProses();
}
if(resetPressed)
{
    resetSistem();
}
break;
case FINISHED:
    if(resetPressed){
        resetSistem();
    }
    break;
}
}

bool tombolPressed(byte pin,
    PolaritasTombol polaritas,
    bool &stableState,
    bool &lastRawState,
    unsigned long &lastDebounceMillis)
{
    bool rawState = digitalRead(pin);

    bool pressed = (polaritas == TOMBOL_NC) ?
        (rawState == HIGH) :
        (rawState == LOW);

    if (pressed != lastRawState) {
        lastDebounceMillis = millis();
        lastRawState = pressed;
    }

    if ((millis() - lastDebounceMillis) >= DEBOUNCE_MS) {

        if (stableState != pressed) {

            stableState = pressed;

            if (stableState) {
                return true;
            }
        }
    }
}
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
    }  
  }  
}  
  
return false;  
}  
  
void tundaResponsif(unsigned long totalMs)  
{  
  unsigned long mulai = millis();  
  
  while ((millis() - mulai) < totalMs)  
  {  
    handleTombol();  
  
    if(statusSistem == STOPPED)  
      return;  
  
    updateAktuator();  
    updateBuzzer();  
    updatePilotLamp();  
  
    delay(20);  
  }  
}  
  
void mulaiProsesBaru() {  
  bacaRTC();  
  
  if (!buatFileLogBaru()) {  
    Serial.println(F("[SD] PERINGATAN: Proses tetap dimulai  
walau file log SD " "GAGAL dibuat. Sistem akan terus mencoba  
membuat file " "secara otomatis selama RUNNING."));  
  }  
  prosesAktif = true;  
  motorPurpose = MOTOR_NONE;  
  motorStartMillis = 0;  
  motorDurationMs = 0;  
  pompaStartMillis = 0;  
  pompaDurationMs = 0;  
  buzzerStartMillis = 0;  
  buzzerDurationMs = 0;  
  komposMatang = false;  
  prosesAwalSelesai = false;  
  buzzerMatangSudahBunyi = false;  
  buzzerErrorSensorWarnaSudahBunyi = false;  
  tanggalMulai = hariIni;
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

bulanMulai = bulanIni;
tahunMulai = tahunIni;
jamMulai = jamIni;
menitMulai = menitIni;
detikMulai = detikIni;
tanggalAdukTerakhir = -1;
jamAdukTerakhir = -1;
akumSuhu = akumKelembapan = akumPh = 0.0;
jumlahSampelAkumulasi = 0;
jumlahGagalTulisSD = 0;
jumlahSukesTulisSD = 0;
motorStopMillis = 0;
filterIdx = 0;
for (int i = 0; i < FILTER_SIZE; i++) {
    bufferSuhu[i] = 25.0;
    bufferKelembapan[i] = 50.0;
    bufferPH[i] = 7.0;
}
logBufferCount = 0;
hitungDurasiPengomposan();
ubahStatusSistem(RUNNING);
motorOFF();
pompaOFF();
buzzerOFF();
tahapAwal = TAHAP_STIR_AWAL;
aksiTahapDimulai = false;
Serial.println(F("START ditekan dari STANDBY."));
Serial.print(F("Tanggal/Jam mulai: "));
printTanggal(Serial);
Serial.print(' ');
printJam(Serial);
Serial.println();
Serial.println(F("Status: RUNNING. Mengikuti alur diagram
dari tahap 'MOTOR PENGADUK 1 MENIT'."));
prevSensorMillis = 0;
prevLogMillis = 0;
prevSdHealthCheckMillis = 0;
bacaSensor();
simpanDataSD();
tambahKeBufferLog();
nextionUpdatePageMain();
nextionUpdatePageSuhu();
nextionUpdatePageKelembapan();
nextionUpdatePagePH();
nextionUpdatePageEnd();
nextionUpdatePageLogData();
}

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
void alurProsesAwal() {
    switch (tahapAwal) {
        case TAHAP_STIR_AWAL:
            if (!aksiTahapDimulai) {
                Serial.println(F("[ALUR] MOTOR PENGADUK 1 MENIT
(awal)."));
                aktifkanMotor((unsigned long)DURASI_ADUK_AWAL_MENIT
* 60000UL, MOTOR_STIR_AWAL);
                aksiTahapDimulai = true;
            }
            if (!motorAktif) {
                tahapAwal = TAHAP_BACA_SENSOR_AWAL;
                aksiTahapDimulai = false;
            }
            break;

        case TAHAP_BACA_SENSOR_AWAL:
            Serial.println(F("[ALUR] BACA SENSOR (menggunakan
nilai terkini dari pembacaan latar belakang)."));
            tahapAwal = TAHAP_CEK_KELEMBAPAN;
            break;

        case TAHAP_CEK_KELEMBAPAN:
            Serial.println(F("[ALUR] CEK KELEMBAPAN -> RH <
40%?"));
            if (kelembapan < RH_MIN_MATANG) {
                Serial.println(F(" -> YA. Menuju POMPA AKTIF 15
DETIK."));
                tahapAwal = TAHAP_POMPA_AKTIF;
            } else {
                Serial.println(F(" -> TIDAK. Lanjut ke loop
monitoring (A)."));
                tahapAwal = TAHAP_SELESAI_AWAL;
            }
            break;

        case TAHAP_POMPA_AKTIF:
            if (!aksiTahapDimulai) {
                aktifkanPompa();
                aksiTahapDimulai = true;
            }
            if (!pompaAktif) {
                tahapAwal = TAHAP_STIR_SETELAH_SIRAM;
                aksiTahapDimulai = false;
            }
            break;

        case TAHAP_STIR_SETELAH_SIRAM:
            // [DIAGRAM] MOTOR AKTIF 1 MENIT
            if (!aksiTahapDimulai) {
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
Serial.println(F("[ALUR] MOTOR AKTIF 1 MENIT
(setelah penyiraman)."));
    aktifkanMotor((unsigned
long)DURASI_ADUK_SETELAH_SIRAM_MENIT * 60000UL,
MOTOR_SETELAH_SIRAM);
    aksiTahapDimulai = true;
}
if (!motorAktif) {
    tahapAwal = TAHAP_SELESAI_AWAL;
    aksiTahapDimulai = false;
}
break;

case TAHAP_SELESAI_AWAL:
    // -> lompat ke label A pada diagram
    prosesAwalSelesai = true;
    prevSensorMillis = millis();
    prevLogMillis = millis();
    Serial.println(F("[ALUR] Proses awal selesai -> masuk
ke loop monitoring (A)."));
    break;
}
}

void alurMonitoringA() {
    unsigned long nowMs = millis();
    bool waktuTercapai = cekWaktuJadwalAduk();
    if (waktuTercapai && !motorAktif && !pompaAktif &&
belumDiadukJamIni()) {
        catatWaktuAdukTerakhir();
        Serial.print(F("[ALUR-A] Jadwal jam "));
        Serial.print(jamIni);
        Serial.println(F(":00 tercapai -> MOTOR 1 MENIT
(RTC)."));
        aktifkanMotor((unsigned long)DURASI_ADUK_TERJADWAL_MENIT
* 60000UL, MOTOR_JADWAL);
    }
}

void resetSistem() {
    motorOFF();
    pompaOFF();
    buzzerOFF();

    motorSisaMillis = 0;
    pompaSisaMillis = 0;
    motorPurpose = MOTOR_NONE;

    filterIdx = 0;
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
for (int i = 0; i < FILTER_SIZE; i++) {
    bufferSuhu[i] = 25.0;
    bufferKelembapan[i] = 50.0;
    bufferPH[i] = 7.0;
}
prosesAktif = false;
komposMatang = false;
prosesAwalSelesai = false;
tahapAwal = TAHAP_STIR_AWAL;
aksiTahapDimulai = false;
buzzerMatangSudahBunyi = false;
buzzerErrorSensorWarnaSudahBunyi = false;
tanggalMulai = bulanMulai = tahunMulai = 0;
jamMulai = menitMulai = detikMulai = 0;
hariPengomposan = jamSisaDurasi = menitSisaDurasi = 0;
tanggalAdukTerakhir = -1;
jamAdukTerakhir = -1;
akumSuhu = akumKelembapan = akumPh = 0.0;
jumlahSampelAkumulasi = 0;
fileLogSiap = false;
namaFileLogAktif[0] = '\0';
prevSensorMillis = 0;
prevLogMillis = 0;
prevSdHealthCheckMillis = 0;
jumlahGagalTulisSD = 0;
jumlahSukesTulisSD = 0;
sdErrorSudahDilaporkan = false;
logBufferCount = 0;
ubahStatusSistem(STANDBY);
startStable = false;
startLastRaw = false;
resetStable = false;
resetLastRaw = false;
stopStable = false;
stopLastRaw = false;
startDebounceMillis = 0;
resetDebounceMillis = 0;
stopDebounceMillis = 0;
Serial.println(F("RESET ditekan. Sistem kembali
STANDBY."));
nextionUpdatePageMain();
nextionUpdatePageLogData();
nextionUpdatePageEnd();
}
//
=====
// SERIAL MONITOR
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
//  
=====
```

```
void tampilkanSerial() {  
    Serial.println();  
    Serial.println(F("=== DATA MONITORING KOMPOS ==="));  
    Serial.print(F("Tanggal          : "));  
    printTanggal(Serial);  
    Serial.println();  
    Serial.print(F("Jam              : "));  
    printJam(Serial);  
    Serial.println();  
    Serial.print(F("Hari Pengomposan : "));  
    Serial.println(hariPengomposan);  
  
    Serial.print(F("Suhu              : "));  
    if (suhu <= -100.0) Serial.println(F("ERROR"));  
    else {  
        Serial.print(suhu, 2);  
        Serial.println(F(" C"));  
    }  
  
    Serial.print(F("ADC Moisture      : "));  
    Serial.println(adckKelembapanTerakhir);  
    Serial.print(F("Kelembapan        : "));  
    Serial.print(kelembapan, 2);  
    Serial.println(F(" %"));  
    Serial.print(F("ADC pH            : "));  
    Serial.println(adcpHTerakhir);  
    Serial.print(F("pH                : "));  
    Serial.println(pH, 2);  
    Serial.print(F("Pulse R (us)      : "));  
    Serial.println(pulseR);  
    Serial.print(F("Pulse G (us)      : "));  
    Serial.println(pulseG);  
    Serial.print(F("Pulse B (us)      : "));  
    Serial.println(pulseB);  
    Serial.print(F("Pulse Klasifikasi: R="));  
    Serial.print(nilaiR);  
    Serial.print(F(" G="));  
    Serial.print(nilaiG);  
    Serial.print(F(" B="));  
    Serial.println(nilaiB);  
    Serial.print(F("Warna              : "));  
    Serial.println(warnaText());  
    if (sensorWarnaError) Serial.println(F("ERROR: SENSOR  
TCS3200"));
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
Serial.print(F("Syarat Matang(B) : "));
Serial.print(F("pH="));
Serial.print((pH >= PH_MIN_MATANG && pH <= PH_MAX_MATANG)
? F("OK") : F("--"));
Serial.print(F(" Warna="));
Serial.println(warnaKompos == WARNA_COKLAT_GELAP ? F("OK")
: F("--"));

Serial.print(F("Status Motor      : "));
Serial.println(statusOnOff(motorAktif));
Serial.print(F("Status Pompa      : "));
Serial.println(statusOnOff(pompaAktif));
Serial.print(F("Status Sistem      : "));
Serial.println(statusSistemText());
Serial.print(F("Status Kompos      : "));
Serial.println(statusKomposText());
Serial.print(F("Status SD Card      : "));
Serial.println(statusSDText());
Serial.print(F("SD Sukses/Gagal      : "));
Serial.print(jumlahSukesTulisSD);
Serial.print('/');
Serial.println(jumlahGagalTulisSD);

Serial.println(F("====="));
}

void tampilkanRingkasanAkhir() {
    float rataSuhu = (jumlahSampelAkumulasi > 0) ? (akumSuhu /
jumlahSampelAkumulasi) : 0.0;
    float rataKelembapan = (jumlahSampelAkumulasi > 0) ?
(akumKelembapan / jumlahSampelAkumulasi) : 0.0;
    float rataPh = (jumlahSampelAkumulasi > 0) ? (akumPh /
jumlahSampelAkumulasi) : 0.0;

    Serial.println(F("=== RINGKASAN AKHIR PENGOMPOSAN
(SELESAI) ==="));
    Serial.print(F("Status Kompos      : "));
    Serial.println(statusKomposText());
    Serial.print(F("Rata-rata Suhu      : "));
    Serial.print(rataSuhu, 2);
    Serial.println(F(" C"));
    Serial.print(F("Rata-rata RH      : "));
    Serial.print(rataKelembapan, 2);
    Serial.println(F(" %"));
    Serial.print(F("Rata-rata pH      : "));
    Serial.println(rataPh, 2);
    Serial.print(F("Durasi Total      : "));
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
Serial.print(hariPengomposan);
Serial.print(F(" Hari "));
Serial.print(jamSisaDurasi);
Serial.print(F(" Jam "));
Serial.print(menitSisaDurasi);
Serial.println(F(" Menit"));
Serial.println(F("====="));
}

const char *statusSistemText() {
    switch (statusSistem) {
        case STANDBY: return "STANDBY";
        case RUNNING: return "RUNNING";
        case FINISHED: return "FINISHED";
        case STOPPED: return "STOPPED";
        default: return "UNKNOWN";
    }
}

const char *warnaText() {
    switch (warnaKompos) {
        case WARNA_HIJAU: return "HIJAU";
        case WARNA_HIJAU_KEKUNINGAN: return "HIJAU KEKUNINGAN";
        case WARNA_KUNING: return "KUNING";
        case WARNA_PUTIH: return "PUTIH";
        case WARNA_COKLAT_MUDA: return "COKLAT MUDA";
        case WARNA_COKLAT: return "COKLAT";
        case WARNA_COKLAT_GELAP: return "COKLAT GELAP";
        default: return "COKLAT MUDA";
    }
}

const char *statusKomposText() {
    return komposMatang ? "MATANG" : "BELUM MATANG";
}

const char *statusOnOff(bool aktif) {
    return aktif ? "ON" : "OFF";
}

const char *statusSDText() {
    return sdTersedia ? (fileLogSiap ? "OK" : "SIAP-TANPA-FILE") : "TIDAK TERDETEKSI";
}

void print2Digits(Print & out, int value) {
    if (value < 10) out.print('0');
    out.print(value);
}
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
void printTanggal(Print & out) {
    print2Digits(out, hariIni);
    out.print('/');
    print2Digits(out, bulanIni);
    out.print('/');
    out.print(tahunIni);
}

void printJam(Print & out) {
    print2Digits(out, jamIni);
    out.print(':');
    print2Digits(out, menitIni);
    out.print(':');
    print2Digits(out, detikIni);
}

bool elapsed(unsigned long nowMs, unsigned long previousMs,
unsigned long intervalMs) {
    return (unsigned long)(nowMs - previousMs) >= intervalMs;
}

//
=====
// HMI NEXTION
//
=====

void nextionInit() {
    NEXTION_SERIAL.begin(NEXTION_BAUD);
    Serial.print(F("HMI Nextion siap di Serial3, "));
    Serial.print(NEXTION_BAUD);
    Serial.println(F(" baud."));
}

void nextionEndCmd() {
    NEXTION_SERIAL.write(0xFF);
    NEXTION_SERIAL.write(0xFF);
    NEXTION_SERIAL.write(0xFF);
}

void nextionSetText(const char *objName, const String
&value) {
    NEXTION_SERIAL.print(objName);
    NEXTION_SERIAL.print(F(".txt=\""));
    NEXTION_SERIAL.print(value);
    NEXTION_SERIAL.print('"');
    nextionEndCmd();
}
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
}

void nextionCommand(const char *cmd) {
    NEXTION_SERIAL.print(cmd);
    nextionEndCmd();
}

void nextionGoPage(const char *pageName) {
    NEXTION_SERIAL.print("page ");
    NEXTION_SERIAL.print(pageName);
    nextionEndCmd();
}

void nextionPaksaIconEnd() {
    if (komposMatang || statusSistem == FINISHED) {
        nextionCommand("End.tm0.en=0");
        delay(100);
        nextionCommand("End.p0.pic=2");
        delay(100);
        nextionCommand("End.p0.pic=2");
    } else {
        nextionCommand("End.p0.pic=3");
        delay(100);
        nextionCommand("End.tm0.en=1");
    }
}

void nextionUpdatePageMain() {
    nextionSetText("Main.t0", String(suhu, 2) + " C");
    nextionSetText("Main.t1", String(kelembapan, 2) + " %");
    nextionSetText("Main.t2", String(pH, 2));
    nextionSetText("Main.g0", String(warnaText()));
    nextionSetText("Main.t4", String(statusSistemText()));
    nextionSetText("Main.t5",
String(statusOnOff(motorAktif)));
    nextionSetText("Main.t6",
String(statusOnOff(pompaAktif)));
    nextionSetText("Main.t7",
String(statusOnOff(buzzerAktif)));
    nextionSetText("Main.t8", String(hariPengomposan) + "
Hari");

    char durasi[24];
    snprintf(durasi, sizeof(durasi), "%d Jam %d Menit",
jamSisaDurasi, menitSisaDurasi);
    nextionSetText("Main.g1", String(durasi));
}
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
char rtcText[12];
sprintf(rtcText, sizeof(rtcText), "%02d:%02d:%02d",
jamIni, menitIni, detikIni);
nextionSetText("Main.t10", String(rtcText));

// [BARU] Tampilkan status SD Card di HMI (perlu komponen
teks
// "t11" pada halaman Main; sesuaikan nama objek jika
berbeda di
// project Nextion Anda).
nextionSetText("Main.t11", String("SD:") +
statusSDText());
}

void tambahKeBufferLog() {
if (logBufferCount >= LOG_BUFFER_SIZE) {
for (int i = 0; i < LOG_BUFFER_SIZE - 1; i++) {
logBuffer[i] = logBuffer[i + 1];
}
logBufferCount = LOG_BUFFER_SIZE - 1;
}

LogEntry entry;
sprintf(entry.tanggal, sizeof(entry.tanggal),
"%02d/%02d", hariIni, bulanIni);
sprintf(entry.jam, sizeof(entry.jam), "%02d:%02d",
jamIni, menitIni);
entry.suhu = suhu;
entry.kelembapan = kelembapan;
entry.ph = pH;
strncpy(entry.warna, warnaText(), sizeof(entry.warna));

logBuffer[logBufferCount] = entry;
logBufferCount++;
}

void nextionUpdatePageLogData() {
for (int i = 0; i < LOG_BUFFER_SIZE; i++) {
char objNo[16], objTanggal[16], objWaktu[16],
objSuhu[16],
objRH[16], objPh[16], objWarna[16];

sprintf(objNo, sizeof(objNo), "LogData.t%d", 35 + i);
sprintf(objTanggal, sizeof(objTanggal), "LogData.t%d",
28 + i);
sprintf(objWaktu, sizeof(objWaktu), "LogData.t%d", 21 +
i);
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

    snprintf(objSuhu, sizeof(objSuhu), "LogData.t%d", 0 +
i);

    snprintf(objRH, sizeof(objRH), "LogData.t%d", 7 + i);
    snprintf(objPh, sizeof(objPh), "LogData.t%d", 14 + i);
    snprintf(objWarna, sizeof(objWarna), "LogData.g%d", i);

    if (i < logBufferCount) {
        LogEntry &e = logBuffer[i];
        nextionSetText(objNo, String(i + 1));
        nextionSetText(objTanggal, String(e.tanggal));
        nextionSetText(objWaktu, String(e.jam));
        nextionSetText(objSuhu, String(e.suhu, 1));
        nextionSetText(objRH, String(e.kelembapan, 0));
        nextionSetText(objPh, String(e.ph, 1));
        nextionSetText(objWarna, String(e.warna));
    } else {
        nextionSetText(objNo, String(""));
        nextionSetText(objTanggal, String(""));
        nextionSetText(objWaktu, String(""));
        nextionSetText(objSuhu, String(""));
        nextionSetText(objRH, String(""));
        nextionSetText(objPh, String(""));
        nextionSetText(objWarna, String(""));
    }
}
//
=====
// KONVERSI NILAI SENSOR -> BYTE WAVEFORM
//
=====

byte suhuKeWaveform(float s) {
    if (s <= -100.0) return 0;
    float clamp = constrain(s, 10.0, 61.0);
    return (byte)((clamp - 10.0) / (61.0 - 10.0) * 255.0);
}

byte kelembapanKeWaveform(float rh) {
    float clamp = constrain(rh, 0.0, 100.0);
    return (byte)(clamp / 100.0 * 255.0);
}

byte phKeWaveform(float ph) {
    float clamp = constrain(ph, 4.0, 14.0);
    return (byte)((clamp - 4.0) / (14.0 - 4.0) * 255.0);
}

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
void nextionAddWaveformData(byte waveformId, byte channel,
byte value) {
    String kirimData = "add ";
    kirimData += waveformId;
    kirimData += ",";
    kirimData += channel;
    kirimData += ",";
    kirimData += value;
    NEXTION_SERIAL.print(kirimData);
    NEXTION_SERIAL.write(0xFF);
    NEXTION_SERIAL.write(0xFF);
    NEXTION_SERIAL.write(0xFF);
}

void nextionUpdatePageSuhu() {
    char buf[16];
    if (suhu <= -100.0) {
        nextionSetText("Suhu.t7", String("ERROR"));
    } else {
        snprintf(buf, sizeof(buf), "%.2f C", suhu);
        nextionSetText("Suhu.t7", String(buf));
    }
}

void nextionUpdatePageKelembapan() {
    char buf[16];
    snprintf(buf, sizeof(buf), "%.2f %%", kelembapan);
    nextionSetText("Kelembapan.t7", String(buf));
}

void nextionUpdatePagePH() {
    char buf[12];
    snprintf(buf, sizeof(buf), "%.2f", pH);
    nextionSetText("pH.t7", String(buf));
}

void nextionUpdatePageEnd() {
    float tampilSuhu =
        (jumlahSampelAkumulasi > 0) ? (akumSuhu /
jumlahSampelAkumulasi) : suhu;

    float tampilKelembapan =
        (jumlahSampelAkumulasi > 0) ? (akumKelembapan /
jumlahSampelAkumulasi) : kelembapan;

    float tampilPh =
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
(jumlahSampelAkumulasi > 0) ? (akumPh /
jumlahSampelAkumulasi) : pH;

// =====
// DATA PAGE END
// =====
nextionSetText("End.t0", String(statusKomposText()));

if (tampilSuhu <= -100.0) {
    nextionSetText("End.t1", String("ERROR"));
} else {
    nextionSetText("End.t1", String(tampilSuhu, 1) + " C");
}

nextionSetText("End.t2", String(tampilKelembapan, 1) + "
%");
nextionSetText("End.t3", String(tampilPh, 2));

char durasi[40];
sprintf(durasi, "%d Hari %d Jam %d Menit",
        hariPengomposan, jamSisaDurasi, menitSisaDurasi);
nextionSetText("End.t4", String(durasi));
if (komposMatang || statusSistem == FINISHED) {
    NEXTION_SERIAL.print("End.tm0.en=0");
    nextionEndCmd();
    delay(30);

    NEXTION_SERIAL.print("End.p0.pic=2");
    nextionEndCmd();
} else {
    NEXTION_SERIAL.print("End.p0.pic=3");
    nextionEndCmd();
    delay(30);

    NEXTION_SERIAL.print("End.tm0.en=1");
    nextionEndCmd();
}
}
```

Lampiran 3

Desain HMI Nextion

- Page Welcome



- Page Main





1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- | LOG DATA | | | | | | |
|----------|---------|-------|------|-----|-----|-------|
| NO | TANGGAL | WAKTU | SUHU | RH | PH | WARNA |
| t35 | t28 | t21 | t0 | t7 | t14 | g0 |
| t36 | t29 | t22 | t1 | t8 | t15 | g1 |
| t37 | t30 | t23 | t2 | t9 | t16 | g2 |
| t38 | t31 | t24 | t3 | t10 | t17 | g3 |
| t39 | t32 | t25 | t4 | t11 | t18 | g4 |
| t40 | t33 | t26 | t5 | t12 | t19 | g5 |
| t41 | t34 | t27 | t6 | t13 | t20 | g6 |

-
- STATUS KOMPOS**
- t0
- RATA-RATA SUHU
- t1
- RATA-RATA KELEMBAPAN
- t2
- RATA-RATA PH
- t3
- DURASI TOTAL
- t4
- LOG DATA
- BERANDA

Lampiran 4

Poster Pengujian Alat

- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
 2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Rancang Bangun Sistem Otomasi Pengomposan Berbasis Mikrokontroler dengan Monitoring Multiparameter dan Deteksi Kematangan Kompos



Tujuan Penelitian

POSTER PENGUJIAN ALAT

1. Merancang sistem pengadukan otomatis berdasarkan parameter suhu, kelembapan, ph, dan warna
2. Menguji performa sistem kontrol otomatis pada proses pengomposan sampah organik
3. Merancang sistem monitoring berbasis HMI Nextion
4. Menganalisis tingkat kematangan kompos berdasarkan parameter hasil pengukuran

Latar Belakang

Peningkatan sampah organik yang dipicu oleh aktivitas rumah tangga, pasar tradisional, dan bidang pertanian menjadi permasalahan lingkungan yang perlu ditanggapi. Salah satu metode pengolahan sampah organik yang sering dilakukan adalah metode pengomposan. Pelaksanaan proses pengomposan yang masih mengandalkan penyiraman dan pengadukan secara manual. Pada penelitian ini dirancang sistem pengomposan otomatis berbasis mikrokontroler

Cara Kerja Alat

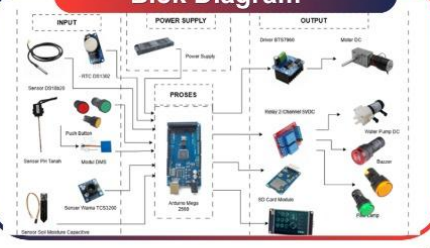
Sistem bekerja dimulai saat tombol "Start" ditekan, lalu Arduino Mega 2560 menginisialisasi seluruh sensor, modul RTC, HMI Nextion, dan SD Card, kemudian menjalankan pengadukan awal selama 1 menit. Setelah itu, sistem memantau kondisi kompos secara berkala. Apabila kelembapan berada di bawah 40%, pompa air akan aktif selama 20 detik untuk menyiram media kompos, diikuti pengadukan ulang selama 1 menit agar air terdistribusi merata, sementara pengadukan terjadwal tetap berjalan enam kali sehari. Proses ini terus berulang hingga parameter memenuhi kriteria kompos matang,



Dibuat Oleh

1. Nabila NIM 2303321040
2. Zulfan Sahli NIM 2303321079

Blok Diagram



Lampiran 5

SOP Pengujian Alat

