

**RANCANG BANGUN APLIKASI *CONTROL* PANEL GUI BERBASIS
PYTHON UNTUK PEMROSESAN DATA PARTIKEL UDARA
DARI INSTRUMEN LCSAQ**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Memperoleh Gelar Sarjana (S1)

Pada Departemen Teknik Elektronika Program Studi Informatika

Universitas Negeri Padang



Oleh:

Vonny Amanda

NIM/TM. 20343028/2020

PROGRAM STUDI TEKNIK INFORMATIKA

DEPARTEMEN TEKNIK ELEKTRONIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI PADANG

2025

HALAMAN PERSETUJUAN TUGAS AKHIR

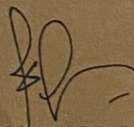
**RANCANG BANGUN APLIKASI CONTROL PANEL GUI BERBASIS PYTHON
UNTUK PEMANTAUAN REAL TIME DATA PARTIKEL UDARA DARI
INSTRUMEN LCSAQ**

Nama : Vonny Amanda
NIM / TM : 20343028 / 2020
Program Studi : Informatika
Departemen : Teknik Elektronika
Fakultas : Teknik

Padang, 27 November 2025

Disetujui Oleh,

Pembimbing



Dr. Svafrijon, S.Pd., M.Kom.
NIP. 19730701200112 1 002

Kepala Departemen Teknik Elektronika FT-UNP



Dr. Dedy Irfan, S.Pd., M.Kom.
NIP. 19760408 200501 1 002

PENGESAHAN TIM PENGUJI

Dinyatakan lulus setelah dipertahankan di depan Tim Penguji Tugas Akhir

Program Studi Informatika

Departemen Teknik Elektronika, Fakultas Teknik

Universitas Negeri Padang

Judul : RANCANG BANGUN APLIKASI CONTROL PANEL
GUI BERBASIS PYTHON UNTUK PEMANTAUAN
REAL-TIME DATA PARTIKEL UDARA DARI
INSTRUMEN LCSAQ

Nama : Vonny Amanda

NIM / TM : 20343028/ 2020

Program Studi : Informatika

Departemen : Teknik Elektronika

Fakultas : Teknik

Padang, 27 November 2025

Tim Penguji

Nama	Tanda Tangan
1. Ketua : Dr. Yeka Hendriyani, S.Kom., M.Kom.	1. 
2. Anggota : Dr. Syafrijon, S.Pd., M.Kom.	2. 
3. Anggota : Dr. Phil. Dony Novaliendry, M.Kom.	3. 

SURAT PERNYATAAN TIDAK PLAGIAT

Saya yang bertanda tangan di bawah ini:

Nama : Vonny Amanda
Nim/TM : 20343028/2020
Program Studi : Informatika
Departemen : Teknik Elektronika
Fakultas : Teknik

Dengan ini menyatakan, bahwa tugas akhir saya yang berjudul **“Rancang Bangun Aplikasi *Control Panel* GUI Berbasis Python Untuk Pemrosesan Data Partikel Udara Dari Instrumen LCSAQ”** adalah benar hasil karya saya sendiri, bukan hasil plagiat dari karya orang lain kecuali sebagai acuan atau kutipan dengan mengikuti tata cara penulisan karya ilmiah yang lazim. Apabila suatu saat terbukti saya melakukan plagiat, maka saya bersedia diproses dan menerima sanksi akademis maupun hukum sesuai dengan ketentuan yang berlaku, baik di institusi UNP maupun masyarakat dan Negara.

Demikian, surat pernyataan ini saya buat dengan penuh kesadaran dan rasa tanggung jawab sebagai anggota masyarakat ilmiah.

Padang, 26 November 2025

Saya yang menyatakan,



Vonny Amanda
NIM.20343028

ABSTRAK

**Vonny Amanda Rancang Bangun Aplikasi *Control* Panel GUI Berbasis
(20343028) Python Untuk Pemrosesan Data Partikel Udara Dari
Instrumen LCSAQ**

LCSAQ adalah sensor yang dimiliki Stasiun GAW Bukit Kototabang untuk mengukur partikel udara PM_{2.5} and PM₁₀, CO₂, *Temperature*, and *Relative Humidity*. Data pemantauan dikirimkan menuju MQTT Broker dan selanjutnya menuju *Raspberry* Pi menggunakan *lcsaq.py* sebagai program kontrol dan transfer untuk menyimpan data tersebut menuju *database* lokal. Terdapat keterbatasan dalam alur transfer data ini yaitu pemrosesan, fungsi kontrol, dan *output* yang masih berbentuk CLI dengan *file* yang terpisah-pisah, tidak ada pendeteksi masalah yang mungkin terjadi pada data atau perangkat, dan data belum berbentuk informasi yang dapat dimengerti oleh masyarakat. Penelitian ini bertujuan untuk membuat aplikasi LCSAQ *Control* Panel GUI yang terhubung dengan seluruh perangkat untuk memenuhi keterbatasan-keterbatasan sebelumnya. Aplikasi dibangun menggunakan Python sebagai *control* panel berbasis GUI dengan memakai metode *Agile* dalam pengembangannya yang memiliki lima halaman aplikasi dan satu halaman web *localhost* berisi pemrosesan data, fungsi kontrol, fungsi filter, pendeteksi *error*, serta hasil pengolahan data menjadi informasi peringatan masalah kesehatan dini bagi masyarakat. Hasil penelitian menunjukkan aplikasi LCSAQ *Control* Panel GUI yang berhasil melewati tahap *Testing* dan berhasil memproses serta mengolah data dari LCSAQ secara *real-time*.

Kata kunci: LCSAQ, GUI, python, *control panel*, metode *agile*

KATA PENGANTAR

Puji syukur ke hadirat Allah Swt. yang selalu melimpahkan rahmat-Nya sehingga penulis dapat menyelesaikan Tugas Akhir berjudul **“RANCANG BANGUN APLIKASI CONTROL PANEL GUI BERBASIS PYTHON UNTUK PEMROSESAN DATA PARTIKEL UDARA DARI INSTRUMEN LCSAQ”**. Tugas Akhir ini disusun sebagai salah satu syarat kelulusan gelar Sarjana pada Program Studi Informatika, Departemen Teknik Elektronika, Fakultas Teknik, Universitas Negeri Padang.

Penulisan Tugas Akhir ini tidak lepas dari berbagai pihak yang telah memberikan bantuan, bimbingan, serta dukungan kepada penulis. Penulis mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak Dr. Ir. Krismadinata, S.T, M.T, Ph.D selaku Rektor Universitas Negeri Padang.
2. Bapak Dr. Muhammad Anwar, S.Pd., M.T. selaku Dekan Fakultas Teknik Universitas Negeri Padang.
3. Bapak Dr. Dedy Irfan, S.Pd., M.Kom. selaku Kepala Departemen Teknik Elektronika Universitas Negeri Padang.
4. Ibu Dr. Yeka Hendriyani, S.Kom., M.Kom. selaku Koordinator Program Studi Teknik Informatika Universitas Negeri Padang dan Dosen Penguji Tugas Akhir.
5. Bapak Dr. Phil. Dony Novaliendry, M.Kom. selaku Dosen Pembimbing Akademik serta Dosen Penguji Tugas Akhir.
6. Bapak Dr. Syafrijon, S.Pd., M.Kom. selaku Dosen Pembimbing yang telah banyak memberikan bimbingan, bantuan, dan dukungan kepada penulis selama

masa penyelesaian Tugas Akhir.

7. Seluruh keanggotaan Stasiun Pemantau Atmosfer Global Bukit Kototabang yang telah memberikan kesempatan dan bimbingan bagi penulis dalam melakukan penelitian di Stasiun GAW.
8. Ibunda tercinta yang tidak pernah lelah memberikan dukungan moral dan materil kepada penulis sehingga dapat menyelesaikan pendidikan di Universitas Negeri Padang sebagai seorang Sarjana.
9. Adik-adik dan sepupu tercinta, serta seluruh keluarga besar penulis yang telah memberikan banyak dukungan selama penulis menempuh Pendidikan di Universitas Negeri Padang.
10. Teman-teman seperjuangan, mahasiswa Program Studi Informatika Angkatan 2020 yang telah memberikan dukungan dan motivasi kepada penulis.
11. Semua pihak yang telah memberikan bantuan dalam penyelesaian Tugas Akhir yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kata sempurna. Penulis memohon maaf atas kekurangan yang ditemui serta mengharapkan kritik dan saran untuk perbaikan ke depannya. Terima kasih.

Padang, 3 November 2025



Vonny Amanda

DAFTAR ISI

ABSTRAK	i
KATA PENGANTAR.....	ii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	vii
DAFTAR TABEL	xvi
BAB I PENDAHULUAN.....	1
A. Latar Belakang.....	1
B. Identifikasi Masalah	7
C. Batasan Masalah	8
D. Rumusan Masalah.....	9
E. Tujuan	9
F. Manfaat	10
BAB II LANDASAN TEORI	11
A. LCSAQ (<i>Low Cost Sensor Air Quality</i>).....	11
B. ISPU (Indeks Standar Pencemar Udara).....	16
C. <i>Control Panel</i>	22
D. GUI (<i>Graphical User Interface</i>)	22
E. Python	23
F. OOP (<i>Object-Oriented Programming</i>).....	24
G. Tkinter.....	28
H. <i>CustomTkinter</i>	29
I. Debezium.....	30
J. MQTT (<i>Message Queuing Telemetry Transport</i>)	32
K. <i>Raspberry Pi</i>	34

L. <i>Heat Index</i>	37
M. <i>CO₂ Index</i>	42
N. <i>Flask-SocketIO</i>	42
O. Metode Penelitian.....	43
P. Pemodelan Sistem	45
Q. Penelitian Terkait.....	48
BAB III METODOLOGI PENELITIAN	51
A. Metode Pengembangan Sistem	51
B. <i>Requirements</i>	51
C. <i>Design</i>	55
D. <i>Development</i>	82
E. <i>Testing</i>	82
F. <i>Deployment</i>	83
BAB IV HASIL DAN PEMBAHASAN	84
A. Mengkonfigurasi Debezium dan Kafka	84
B. Membangun Program <i>Main</i>	91
C. Membangun Halaman <i>Dashboard</i>	95
D. Membangun Halaman Status	105
E. Membangun Halaman <i>Raw Data</i>	111
F. Membangun Halaman <i>Event Log</i>	119
G. Membangun Halaman <i>Configuration</i>	124
H. Membangun Program <i>Setting</i>	129
I. Membangun Halaman Web	131
G. <i>Testing</i>	152
H. <i>Deployment</i>	161

BAB V PENUTUP	169
DAFTAR PUSTAKA	171
LAMPIRAN.....	177

DAFTAR GAMBAR

Gambar 1. Instrumen LCSAQ.....	12
Gambar 2. NAB Untuk Konsentrasi PM _{2.5} Dengan Satuan µg/m ³	15
Gambar 3. NAB Untuk Konsentrasi PM ₁₀ Dengan Satuan µg/m ³	15
Gambar 4. Contoh <i>Class</i> dari Implementasi OOP Python.....	25
Gambar 5. Contoh <i>Object</i> Pada <i>Class</i> dari Implementasi OOP Python	25
Gambar 6. <i>Parent</i> dan <i>Child Class</i>	27
Gambar 7. Arsitektur MQTT	33
Gambar 8. <i>Raspberry</i> Pi 4 Model B.....	35
Gambar 9. <i>Raspberry</i> Pi di Dalam <i>Case</i>	36
Gambar 10. <i>Heat Index Chart</i>	39
Gambar 11. Klasifikasi <i>Heat Index</i>	40
Gambar 12. Kategori CO ₂	42
Gambar 13. Tahapan Dalam Model <i>Agile</i>	44
Gambar 14. <i>Use Case</i> Diagram Aplikasi LCSAQ <i>Control Panel</i> GUI	56
Gambar 15. <i>Use Case</i> Diagram Halaman Web	57
Gambar 16. <i>Activity</i> Diagram Halaman <i>Dashboard</i>	59
Gambar 17. <i>Activity</i> Diagram <i>Switch lcsaq.service</i>	60
Gambar 18. <i>Activity</i> Diagram <i>Restart</i> dan <i>Test lcsaq.service</i>	61
Gambar 19. <i>Activity</i> Diagram Koneksi <i>Real-time MQTT Broker</i>	62
Gambar 20. <i>Activity</i> Diagram <i>Ping IP MQTT Broker</i>	63
Gambar 21. <i>Activity</i> Diagram <i>Real-time</i> Data 60 Menit Terakhir	64
Gambar 22. <i>Activity</i> Diagram Filter Tabel Data	65

Gambar 23. <i>Activity Diagram</i> Reset Tabel Data	67
Gambar 24. <i>Activity Diagram</i> Save Tabel Data	68
Gambar 25. <i>Activity Diagram</i> <i>Real-time</i> Log 30 Menit Terakhir	69
Gambar 26. <i>Activity Diagram</i> Filter Tabel Log	70
Gambar 27. <i>Activity Diagram</i> Reset Tabel Log	72
Gambar 28. <i>Activity Diagram</i> Pengaturan Koneksi <i>Database</i> Lokal.....	73
Gambar 29. <i>Activity Diagram</i> Pengaturan Koneksi <i>MQTT Broker</i>	74
Gambar 30. <i>Activity Diagram</i> Direktori <i>File .xlsx</i> dan <i>File Log</i>	75
Gambar 31. <i>Activity Diagram</i> Halaman Web <i>Localhost</i>	77
Gambar 32. Halaman <i>Dashboard</i> Aplikasi <i>LCSAQ Control Panel GUI</i>	78
Gambar 33. Halaman Status Aplikasi <i>LCSAQ Control Panel GUI</i>	79
Gambar 34. Halaman <i>Database Table</i> Aplikasi <i>LCSAQ Control Panel GUI</i>	79
Gambar 35. Halaman <i>Log Table</i> Aplikasi <i>LCSAQ Control Panel GUI</i>	80
Gambar 36. Halaman <i>Configuration</i> Aplikasi <i>LCSAQ Control Panel GUI</i>	81
Gambar 37. Halaman Web <i>Localhost</i>	81
Gambar 38. Tampilan Pesan <i>Error</i> pada Aplikasi	82
Gambar 39. Direktori Zookeeper	86
Gambar 40. Direktori Server Kafka	86
Gambar 41. Direktori Debezium.....	87
Gambar 42. <i>File</i> Konektor Debezium.....	87
Gambar 43. Status Aktif Konektor Debezium	89
Gambar 44. <i>Database binlog</i>	90
Gambar 45. <i>Offsets</i> Baru dari <i>Database</i>	90

Gambar 46. <i>Library Program Main</i>	91
Gambar 47. <i>Class Parent_GUI()</i>	92
Gambar 48. <i>Icon dan Navigation Bar</i>	93
Gambar 49. <i>Button Menu</i>	93
Gambar 50. <i>Desain Switch Theme</i>	93
Gambar 51. <i>Setup Pages</i>	94
Gambar 52. <i>Switch Theme Light dan Dark</i>	94
Gambar 53. <i>Mainloop (root)</i>	95
Gambar 54. <i>Library Program realtime_data_page</i>	95
Gambar 55. <i>Class RealtimeData_Page()</i>	96
Gambar 56. <i>Content, Header, dan Title</i>	97
Gambar 57. <i>Desain Frame PM_{2.5}</i>	97
Gambar 58. <i>Desain Frame CO₂</i>	98
Gambar 59. <i>Variable dan Function Program realtime_data_page</i>	98
Gambar 60. <i>Memulai Zookeeper, Kafka, dan Debezium</i>	99
Gambar 61. <i>Threading Services Memulai KafkaConsumer</i>	99
Gambar 62. <i>Function on_close</i>	100
Gambar 63. <i>Cek Status Database</i>	100
Gambar 64. <i>Parameter untuk KafkaConsumer</i>	101
Gambar 65. <i>Proses Consume Program realtime_data_page</i>	101
Gambar 66. <i>Message Konsentrasi Partikel Udara realtime_data_page</i>	102
Gambar 67. <i>Threading Function Consume</i>	102
Gambar 68. <i>Function Hapus Offsets</i>	103

Gambar 69. <i>Update</i> Label Nilai Partikel Udara.....	103
Gambar 70. <i>Update Timestamp</i> Partikel Udara	103
Gambar 71. <i>Update</i> Status CO ₂ , Temp, dan RH.....	104
Gambar 72. <i>Update</i> Status PM _{2.5}	104
Gambar 73. <i>Update</i> GUI <i>Dashboard</i>	104
Gambar 74. Menampilkan <i>Error</i> dengan <i>MessageBox</i>	105
Gambar 75. Menampilkan <i>Warning</i> dengan <i>MessageBox</i>	105
Gambar 76. <i>Library</i> Program <i>status_page</i>	106
Gambar 77. <i>Class</i> <i>Status_Page()</i>	106
Gambar 78. Desain <i>Switch</i> <i>lcsaq.service</i>	107
Gambar 79. <i>Button</i> <i>Restart</i> dan <i>Test</i> <i>lcsaq.service</i>	107
Gambar 80. <i>Textbox</i> <i>Stream</i> <i>MQTT Broker</i>	107
Gambar 81. <i>Button</i> serta <i>Textbox</i> <i>Ping</i> dan <i>NTP</i>	108
Gambar 82. <i>List</i> dan <i>Function</i> Program <i>status_page</i>	108
Gambar 83. Parameter <i>MQTT Client</i>	108
Gambar 84. Mulai Koneksi ke <i>MQTT Broker</i>	108
Gambar 85. <i>Threading</i> <i>Stream</i> Data <i>MQTT</i>	109
Gambar 86. <i>Filter</i> <i>Stream</i> Data <i>MQTT</i>	109
Gambar 87. <i>Update</i> <i>Textbox</i> <i>MQTT</i>	109
Gambar 88. Sistem <i>Switch</i> <i>lcsaq.service</i>	110
Gambar 89. <i>Restart</i> dan <i>Test</i> <i>lcsaq.service</i>	110
Gambar 90. <i>Ping</i> IP <i>MQTT Broker</i>	111
Gambar 91. <i>Library</i> Program <i>rawdata_page</i>	111

Gambar 92. <i>Class RawData_Page()</i>	112
Gambar 93. Desain Kalender Program <i>rawdata_page</i>	112
Gambar 94. Desain <i>Combobox rawdata_page</i>	113
Gambar 95. <i>Button</i> Filter, Reset, Stop, dan <i>Save</i>	113
Gambar 96. Desain Tabel <i>Raw Data</i>	114
Gambar 97. <i>Consumer</i> Program <i>rawdata_page</i>	115
Gambar 98. <i>Update</i> Tabel <i>Raw Data</i>	115
Gambar 99. <i>Update</i> Isi Kolom Tabel <i>Raw Data</i>	116
Gambar 100. Menghapus Data dari Tabel <i>Raw Data</i>	116
Gambar 101. Koneksi Program <i>rawdata_page</i> Menuju <i>Database</i>	116
Gambar 102. Proses Filter Data dari <i>Database</i>	117
Gambar 103. <i>Update</i> Data Hasil Filter Menuju Tabel <i>Raw Data</i>	117
Gambar 104. Reset dan Stop Tabel <i>Raw Data</i>	118
Gambar 105. Menyimpan Data Tabel ke <i>File .xlsx</i>	118
Gambar 106. <i>Path</i> Penyimpanan <i>File .xlsx</i>	119
Gambar 107. <i>Library</i> Program <i>eventlog_page</i>	119
Gambar 108. <i>Class EventLog_Page()</i>	120
Gambar 109. Desain <i>Timepicker eventlog_page</i>	120
Gambar 110. Desain Tabel <i>Event Log</i>	121
Gambar 111. <i>List</i> dan <i>Function</i> Program <i>eventlog_page</i>	121
Gambar 112. <i>Load File Log</i>	121
Gambar 113. Proses <i>Fetch</i> dan Penyimpanan Log	122
Gambar 114. <i>Update</i> Tabel <i>Event Log</i>	122

Gambar 115. Proses Filter Data Log.....	123
Gambar 116. <i>Variables</i> Tanggal dan Waktu.....	123
Gambar 117. Reset Tabel <i>Event</i> Log	123
Gambar 118. <i>Library</i> Program <i>configuration_page</i>	124
Gambar 119. <i>Class Configuration_Page()</i>	125
Gambar 120. Desain <i>Entrybox</i> Untuk Koneksi <i>Database</i>	125
Gambar 121. Desain <i>Entrybox</i> Untuk Koneksi MQTT	125
Gambar 122. Desain <i>Entrybox</i> Untuk <i>Path</i>	126
Gambar 123. <i>Update</i> dan <i>Save</i> Parameter <i>Database</i>	126
Gambar 124. <i>Update File</i> Konektor Debezium	127
Gambar 125. <i>Update</i> dan <i>Save</i> Parameter MQTT	127
Gambar 126. <i>Update Path</i> Penyimpanan .xlsx	127
Gambar 127. <i>Update Path</i> Menuju <i>File</i> Log	128
Gambar 128. <i>Save</i> Seluruh Parameter ke <i>File config.txt</i>	128
Gambar 129. <i>Load</i> Seluruh Parameter dari <i>File config.txt</i>	129
Gambar 130. <i>Storage Key</i> Untuk Setiap Parameter	130
Gambar 131. <i>Instance set_setting</i> dan <i>get_setting</i>	130
Gambar 132. <i>Load File config.txt</i> Pada Program <i>Setting</i>	131
Gambar 133. <i>Library</i> Web pada Program <i>main</i>	131
Gambar 134. <i>Instance Flask</i> dan <i>socketio</i>	132
Gambar 135. <i>Function</i> Global Untuk Memulai Koneksi Web	132
Gambar 136. <i>Pass Instance</i> Web ke <i>realtimedata_page</i>	133
Gambar 137. <i>Start Flask</i> Pada <i>Mainloop</i>	133

Gambar 138. <i>Instance</i> Web Pada Program <i>realtimedata_page</i>	133
Gambar 139. <i>Function</i> <i>send_data()</i>	133
Gambar 140. <i>Library</i> Program <i>webpage</i>	134
Gambar 141. <i>Class</i> <i>Webpage_Data()</i> dan <i>variables</i>	134
Gambar 142. Menghitung Persentase CO ₂	135
Gambar 143. Kategori Status CO ₂	135
Gambar 144. Rumus Perhitungan Nilai <i>Heat Index</i>	136
Gambar 145. Kategori Status <i>Heat Index</i>	136
Gambar 146. Rumus Perhitungan Nilai ISPU	137
Gambar 147. Kategori Status ISPU	137
Gambar 148. <i>Time Group</i> Data ISPU	138
Gambar 149. <i>Update</i> dan <i>Delete Value Time Group</i>	138
Gambar 150. <i>Generate Chart</i> ISPU	139
Gambar 151. <i>Function</i> <i>emit_values()</i> menuju HTML	140
Gambar 152. Memanggil <i>Time Group</i> dari <i>emit_values()</i>	140
Gambar 153. <i>Import</i> dan <i>Link</i> Untuk Halaman Web	141
Gambar 154. Membangun Koneksi ke IP <i>Localhost</i>	141
Gambar 155. Konfigurasi <i>LocalStorage</i>	142
Gambar 156. Seluruh Data dan ID-nya Pada <i>update_data</i>	142
Gambar 157. Warna Kategori PM, CO ₂ dan <i>Heat Index</i>	143
Gambar 158. Warna Kategori ISPU.....	143
Gambar 159. Menampilkan <i>Chart</i> Pada Halaman Web.....	144
Gambar 160. Fungsi <i>Read File .csv</i> Data Lampau	144

Gambar 161. <i>Library</i> Program Percobaan Prediksi	145
Gambar 162. <i>Training</i> dan Proses Prediksi <i>RandomForestRegressor</i>	146
Gambar 163. Chart Perbandingan Data Prediksi dan Data Aktual	146
Gambar 164. <i>Chart</i> Perbandingan Hasil Prediksi dan Data Aktual.....	147
Gambar 165. <i>File Dataset</i> dan Gambar <i>Chart</i>	148
Gambar 166. <i>Update</i> dan <i>Load File Dataset</i>	148
Gambar 167. <i>Lags</i> dan <i>Train Dataset</i>	149
Gambar 168. Proses <i>Forecast</i> Data ISPU	149
Gambar 169. Menampilkan <i>Chart Forecast</i> Data ISPU	150
Gambar 170. Fungsi Utama Halaman Web Simulasi	151
Gambar 171. Memanggil Fungsi <i>Training</i> dan <i>Forecast</i>	151
Gambar 172. <i>Flowchart Function self.consume_realtime_data()</i>	153
Gambar 173. <i>Flowgraph Function self.consume_realtime_data()</i>	154
Gambar 174. <i>Test</i> Menghapus Tabel	158
Gambar 175. <i>Pop-Up</i> Pesan <i>Error</i> dari Tabel	158
Gambar 176. <i>Test</i> Menghapus Data dari <i>Database</i>	159
Gambar 177. Tampilan Data <i>None</i> Pada <i>Dashboard</i>	159
Gambar 178. <i>Insert</i> Data -999.99 ke <i>Database</i>	160
Gambar 179. Tampilan Data -999.99 Pada <i>Dashboard</i>	160
Gambar 180. Desain Akhir Halaman <i>Dashboard</i>	161
Gambar 181. Desain Akhir Keterangan Pada Halaman <i>Dashboard</i>	162
Gambar 182. Desain Akhir Halaman <i>Dashboard</i> Mode <i>Dark</i>	162
Gambar 183. Desain Akhir Halaman Status	163

Gambar 184. Desain Akhir Halaman <i>Raw Data</i>	164
Gambar 185. Desain Akhir Hasil Filter Halaman <i>Raw Data</i>	164
Gambar 186. Desain Akhir Halaman <i>Event Log</i>	165
Gambar 187. Desain Akhir Halaman <i>Setting</i>	165
Gambar 188. Desain Akhir Halaman Web	166
Gambar 189. Desain Akhir Halaman Web (Simulasi).....	167
Gambar 190. Penjelasan <i>Range</i> Nilai Partikel Udara	168

DAFTAR TABEL

Tabel 1. Kategori ISPU	17
Tabel 2. Penjelasan Kategori ISPU	17
Tabel 3. Tabel Konversi Konsentrasi Parameter ISPU	20
Tabel 4. Komponen <i>Use Case</i> Diagram.....	46
Tabel 5. Komponen <i>Activity</i> Diagram.....	47
Tabel 6. Penelitian Terkait	48
Tabel 7. Tingkat Resiko Program	155
Tabel 8. <i>Paths</i> dan Hasilnya.....	156

BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi sudah banyak membantu manusia dalam mempermudah pekerjaan mereka. Salah satu teknologi tersebut adalah perkembangan alat sensor yang sudah umum dijumpai saat ini. Alat sensor digunakan untuk mengukur satuan besaran dan mengubahnya menjadi satuan analog sehingga dapat dibaca dan diproses oleh rangkaian elektronik (Rimbawati et al., 2019). Hasil pemantauan oleh alat sensor akan menjadi *output* yang dibutuhkan oleh manusia untuk berbagai kebutuhan.

Menurut laporan dari *e-Library* milik WMO (*World Meteorological Organization*) yang berjudul “WMO *Global Atmosphere Watch* (GAW) *Implementation Plan: 2016-2023*”, Stasiun Pemantau Atmosfer Global Bukit Kototabang atau Stasiun GAW merupakan sebuah stasiun yang melaksanakan program pengamatan atmosfer secara global di daerah ekuatorial yang termasuk ke dalam sistem monitoring dan riset oleh WMO. Stasiun GAW berfokus kepada komposisi atmosfer yaitu iklim, cuaca, dan polusi udara, sehingga mengharuskan mereka memiliki banyak instrumen berupa alat sensor yang sesuai dengan program pengamatan atmosfer.

Salah satu dari instrumen tersebut adalah LCSAQ (*Low Cost Sensor Air Quality*), yaitu sensor pengukur konsentrasi partikel udara yang mempengaruhi kualitas udara untuk memantau dan menangkap data partikel udara seperti PM_{2.5} dan PM₁₀ (*Particulate Matter*), CO₂ (*Carbon Dioxide*), Temp

(*Temperature*), dan RH (*Relative Humidity*). Instrumen LCSAQ sudah terpasang pada beberapa titik di Indonesia, dengan salah satu penempatannya berada di kawasan Stasiun GAW Kototabang.

Setelah dilakukan observasi di Stasiun GAW terhadap instrumen-instrumen yang mereka miliki, dapat dilihat bahwa instrumen LCSAQ akan menangkap konsentrasi partikel-partikel udara ke dalam bentuk data mentah dan dikirimkan menuju MQTT (*Message Queuing Telemetry Transport*), sebuah protokol transfer data antar perangkat. Stasiun GAW memiliki MQTT *broker* (server) miliknya sendiri yang akan melewatkan kumpulan data mentah secara *real-time* dari seluruh instrumen LCSAQ yang tersebar di Indonesia ke dalam bentuk *message* untuk dapat ditangkap oleh MQTT *client* yang membutuhkan data tersebut. Untuk instrumen LCSAQ yang berada di Stasiun GAW, MQTT *client* yang akan meminta data adalah *Raspberry Pi*, sebuah *mini-computer* dengan sistem operasi *Raspberry Pi OS*.

Berperan sebagai server untuk menyimpan data, di dalam *Raspberry Pi* telah dibangun sebuah *package* bernama *lcsaq* dengan *daemon* *lcsaq.service* atau program berbasis Linux yang akan bekerja sebagai *background service* dalam menghubungkan antara instrumen LCSAQ dengan *Raspberry Pi* dan mengontrol instrumen LCSAQ menggunakan *command-line* pada terminal *Raspberry Pi*. Beberapa fungsi kontrol yang dapat langsung dipanggil menggunakan *lcsaq.service* adalah *command* *start*, *stop*, dan *restart*. Kemudian, *Raspberry Pi* memiliki sebuah *database* lokal berbasis MariaDB yang dapat diakses dengan *login* sebagai *root* atau *user* dari phpMyAdmin. *Database* lokal

digunakan sebagai penyimpanan data hasil pemantauan dari instrumen LCSAQ yang terus di-*update* secara *real-time* selama masih ada data baru yang masuk.

Selain itu, terdapat juga suatu program berbasis Python bernama `lcsaq.py` yang dibangun oleh Stasiun GAW untuk mengatur alur transfer data partikel udara yang telah ditangkap oleh instrumen LCSAQ. Program `lcsaq.py` akan dimulai secara otomatis setiap *Raspberry Pi* dihidupkan oleh *daemon lcsaq.service* yang akan berjalan sebagai *background service*.

Program `lcsaq.py` mencakup beberapa proses yang akan terjadi setiap kali terdapat *update* data partikel udara dari instrumen LCSAQ, yaitu: 1) Membangun koneksi menuju MQTT *broker* sebagai *client* dan menuju *database* lokal sebagai *user*; 2) Menangkap data mentah yang baru saja melewati MQTT *broker* secara *real-time*; 3) Menuliskan kumpulan data mentah yang didapatkan ke dalam *file* berformat `.csv` yang disimpan di *Raspberry Pi*, di mana setiap data baru akan disimpan menjadi sebuah *file* baru; 4) Menuliskan kumpulan data mentah yang didapatkan menjadi sebuah baris berisi konsentrasi partikel udara dan mengirimkannya menuju *database* lokal untuk dimasukkan ke dalam tabel data; 5) Menuliskan log dari aktivitas koneksi dan transfer data *real-time* yang terjadi antara instrumen LCSAQ, MQTT, dan *database* lokal ke dalam satu *file .txt* yang disimpan dalam direktori lokal *Raspberry Pi*; 6) Menyimpan seluruh parameter yang diperlukan untuk koneksi MQTT *broker* dan *database* ke dalam *file* berformat `.txt` dan dipanggil ketika program pertama kali dijalankan.

Melihat dari penjabaran alur transfer data partikel udara dari instrumen LCSAQ menuju perangkat dan *file* lainnya, penulis masih menemukan banyak keterbatasan di dalamnya. Beberapa keterbatasan tersebut adalah: 1) Pembacaan hasil pemantauan tidak efisien karena kumpulan data hanya dibentuk ke dalam satu baris berisi kumpulan angka konsentrasi partikel udara baik di dalam *database* lokal maupun pada *file* log yang sulit untuk dibaca; 2) Terlalu banyak *file* terpisah dalam menampilkan data hasil pemantauan yang secara keseluruhan terbagi menjadi log dalam format .txt, data yang melewati MQTT broker dalam format .csv, dan tampilan web tabel data di *database* lokal pada phpMyAdmin; 3) Fungsi kontrol instrumen LCSAQ dengan *lcsaq.service* hanya dapat dilakukan menggunakan *command-line* pada terminal *Raspberry Pi*; 4) Jika data partikel udara yang telah ditangkap secara mendadak tidak muncul atau *None*, harus dilakukan pengecekan secara satu-persatu terhadap konfigurasi dan koneksi antara instrumen LCSAQ, program *lcsaq.py*, MQTT, *database* lokal, *Raspberry Pi*, dan *file* log sampai menemukan permasalahan yang terjadi; 5) Seluruh alur pemrosesan data hanya berupa CLI (*Command Line Interface*) sehingga tidak memiliki antarmuka menarik dan tidak *user-friendly*; 6) Data hasil pemantauan partikel udara terbatas hanya bisa diakses dan dibaca oleh pengamat di stasiun GAW saja; 7) Data belum diolah sama sekali menjadi informasi yang dapat dimengerti dan bermanfaat bagi masyarakat umum.

Maka dari itu, penulis memutuskan untuk merancang dan membangun sebuah aplikasi bernama “LCSAQ *Control Panel GUI*” yang akan berjalan

secara *real-time* untuk mengolah data hasil pemantauan partikel udara dari instrumen LCSAQ menjadi informasi yang dapat digunakan dan disampaikan kepada masyarakat. Aplikasi ini akan menjadi sebuah *control* panel dengan antarmuka berbasis GUI (*Graphical User Interface*) yang dibangun menggunakan Python yang terhubung dengan perangkat (MQTT *broker*, *Raspberry Pi*, program *lcsaq.py*, dan *database* lokal) serta dengan halaman web berisi informasi yang didapatkan dari pengolahan data hasil pemantauan sebagai pendeteksi dan peringatan dini bagi masyarakat untuk masalah kesehatan dan bencana alam yang berkaitan dengan tingkat kualitas udara.

Konsentrasi partikel udara sebagai data hasil pemantauan instrumen LCSAQ akan diolah berdasarkan ISPU (Indeks Standar Pencemar Udara) yang telah diatur pada Peraturan Menteri Lingkungan Hidup dan Kehutanan Republik Indonesia Nomor 14 Tahun 2020. Partikel udara $PM_{2.5}$ dan PM_{10} yang merupakan polutan penyebab masalah kesehatan dan bencana alam akan dihitung menggunakan persamaan untuk mendapatkan nilai ISPU. Nilai ini akan menjadi informasi yang dibutuhkan oleh Stasiun GAW agar dapat melakukan langkah lebih lanjut dalam mengamati kemungkinan terjadinya bencana alam di area yang termasuk dalam jangkauan instrumen LCSAQ.

Python dipilih sebagai bahasa pemrograman untuk aplikasi disebabkan kepopulerannya baik oleh pengembang pemula maupun profesional karena dapat diterapkan di aspek mana saja terkait dengan pembangunan dan pengembangan perangkat lunak (Thangarajah et al., 2019). Untuk tampilan GUI dari aplikasi ini, akan digunakan *library* bernama *CustomTkinter* yang

merupakan sebuah *project* milik Tom Schimansky sebagai pengembangan dari *library* Tkinter untuk Python, dengan tampilan yang modern dan mudah untuk dikustomisasi. *CustomTkinter* dapat diakses secara publik pada *repository* Github milik Schimansky: <https://github.com/TomSchimansky/CustomTkinter>. Karena aplikasi ini akan menggabungkan banyak fungsi dalam pemrosesan data partikel udara, maka dibutuhkan OOP (*Object-Oriented Programming*) agar dapat memudahkan dalam *import class* dan *variable* sebagai parameter-parameter yang dibutuhkan dalam koneksi antar perangkat dan *file*, menjalankan fungsi kontrol, mengolah dan mengirimkan data, maupun dalam mendesain tampilan aplikasi.

Aplikasi akan dibangun di dalam komputer *Raspberry Pi* dengan mengakses *lcsaq.service* untuk fungsi kontrol instrumen LCSAQ dan mengakses direktori lokal untuk membaca maupun menyimpan *file* yang dibutuhkan. Lalu aplikasi akan membangun koneksi menuju MQTT *broker* sebagai MQTT *client* untuk menangkap data yang masuk dengan melakukan *subscribe* pada *topic* yang menyalurkan data partikel udara dan menuliskannya untuk kebutuhan *maintenance*. Kemudian Debezium akan digunakan sebagai platform untuk menangkap perubahan data *real-time* yang terjadi pada *database* lokal dan mengirimkannya melalui Kafka sebagai *message broker* yang dikelola oleh Zookeeper *service* (Sayar et al., 2023).

Data partikel udara yang telah didapatkan akan diproses dan ditampilkan menuju aplikasi ke dalam bentuk halaman-halaman *control panel* yang dapat diakses dari *Raspberry Pi*. Sedangkan untuk halaman web, *library Flask*-

SocketIO digunakan untuk menghubungkan Python menuju *localhost*, sehingga hasil pemantauan partikel udara yang telah diolah oleh aplikasi *control* panel GUI menjadi informasi terkait kualitas udara dapat dikirimkan dan ditampilkan secara *real-time* dan untuk ke depannya dapat diakses dengan lebih luas oleh masyarakat pada berbagai perangkat.

B. Identifikasi Masalah

Permasalahan yang ditemukan pada latar belakang dapat diidentifikasi sebagai berikut:

1. Hasil pemantauan partikel udara hanya berupa kumpulan baris data berisi angka konsentrasi partikel udara yang ditangkap oleh instrumen LCSAQ dan *daemon lcsaq.service* hanya dapat diakses menggunakan *command-line* pada terminal *Raspberry Pi*. Sehingga berkemungkinan terjadinya *human error* baik untuk pembacaan *output* maupun penulisan fungsi kontrol.
2. Terlalu banyak *file* terpisah untuk hasil pemantauan yaitu *file* berformat *.txt* untuk log aktivitas koneksi dan transfer data, *file* berformat *.csv* untuk pencatatan kumpulan data yang melewati MQTT, dan *phpMyAdmin* sebagai antarmuka tabel data *database* lokal, sehingga membutuhkan waktu bagi pengamat dalam menemukan data tertentu.
3. Program *lcsaq.py* tidak memiliki pendeteksi dan pemberitahuan untuk kendala seperti data hasil pemantauan yang tidak muncul atau kesalahan koneksi menuju *database* lokal, MQTT *broker*, *Raspberry Pi*, dan instrumen LCSAQ sehingga menghabiskan lebih banyak waktu hanya untuk mencari apa dan dimana letak kendala tersebut.

4. Seluruh alur pemrosesan data partikel udara memiliki format CLI (*Command Line Interface*) yang hanya terdiri dari *background services* dan *command-line* untuk fungsi kontrol dengan hasil pemantauan berupa kumpulan angka sangat jauh dari kata *user-friendly*.
5. Belum ada satupun bentuk pengolahan dari data hasil pemantauan partikel udara oleh instrumen LCSAQ yang dapat dijadikan sebuah informasi bagi pengamat Stasiun GAW untuk tindak lanjut dalam mendeteksi masalah kesehatan atau bencana serta bagi masyarakat untuk dapat diakses dengan mudah.

C. Batasan Masalah

Berdasarkan latar belakang dan identifikasi masalah yang didapatkan, maka dapat ditentukan batasan masalah sebagai berikut:

1. Aplikasi akan dibangun berbasis Python memanfaatkan *library* serta implementasi OOP (*Object-Oriented Programming*) untuk fungsi-fungsi pemrosesan data menjadi sebuah aplikasi *control* panel berbentuk lima halaman yaitu *Dashboard*, *Status*, *Database Table*, *Log Table*, dan *Configuration* dengan tampilan GUI menarik serta mudah diakses dapat mendeteksi dan menampilkan kendala yang terjadi pada perangkat dan *file* yang terhubung secara *real-time* dalam bentuk notifikasi *error* dan saran penanganannya.
2. Aplikasi hanya dapat dijalankan pada komputer *Raspberry Pi* dan akan terhubung dengan platform *Debezium* dan *Kafka* yang akan diinstal dan dikonfigurasi pada komputer untuk koneksi *real-time database* lokal.

3. Aplikasi akan mengolah data hasil pemantauan partikel udara menjadi informasi kualitas udara secara *real-time* dan mengirimkannya menuju *localhost* untuk dapat ditampilkan dalam bentuk halaman web yang ke depannya dapat diakses dengan lebih luas kepada masyarakat.

D. Rumusan Masalah

Berdasarkan batasan masalah yang telah ditentukan, maka rumusan masalah yang didapatkan adalah:

1. Bagaimana merancang dan membangun kelima halaman sebuah aplikasi *control panel* mengimplementasikan konsep OOP dan *library* pada Python yang mencakup fungsi pemrosesan data dan pendeteksi *error* secara *real-time* dengan tampilan GUI yang *user-friendly*?
2. Bagaimana menginstal dan mengkonfigurasi platform Debezium dan Kafka pada *Raspberry Pi* untuk koneksi dari *database* lokal menuju aplikasi *control panel* GUI secara *real-time*?
3. Bagaimana mengolah data hasil pemantauan partikel udara menjadi informasi kualitas udara secara *real-time* lalu mengirimkannya menuju *localhost* untuk ditampilkan dalam bentuk halaman web yang ke depannya dapat diakses oleh masyarakat?

E. Tujuan

Tujuan yang ingin dicapai dari tugas akhir ini adalah:

1. Merancang desain GUI dan membangun koneksi menuju *database* lokal, MQTT *broker*, *Raspberry Pi*, dan *file* lainnya, mendeteksi dan menampilkan notifikasi *error* terkait kendala yang terjadi, dan menampilkan data hasil

pemantauan partikel udara secara *real-time* menjadi satu kesatuan aplikasi *control* panel dengan memanfaatkan OOP dan *library* pada Python.

2. Melakukan instalasi Debezium dan Kafka pada *Raspberry* Pi dan mengatur konfigurasi kedua platform untuk dihubungkan dengan aplikasi *control* panel sehingga dapat digunakan dalam membangun koneksi menuju *database* lokal.
3. Melakukan pengolahan terhadap data hasil pemantauan partikel udara menjadi informasi kualitas udara untuk pengamat dan masyarakat serta merancang halaman web sederhana untuk menampilkan hasil pengolahan tersebut secara *real-time* pada *localhost*.

F. Manfaat

Manfaat dari tugas akhir ini adalah:

1. Manfaat Teoritis

Penelitian ini diharapkan dapat dijadikan sebagai referensi oleh peneliti lainnya dalam pengembangan penelitian serupa maupun lanjutan yang berkaitan.

2. Bagi Praktisi

- a. Bagi penulis, sebagai penerapan pengetahuan di bidang keilmuan Informatika yang sudah diperoleh selama perkuliahan dan sebagai tugas akhir untuk salah satu syarat kelulusan Strata 1.
- b. Bagi Stasiun Pemantau Atmosfer Bukit Kototabang, sebagai titik awal dalam mempelajari penerapan pemograman Python lebih lanjut untuk pengolahan data hasil pemantauan dari berbagai instrumen di sana.

BAB V

PENUTUP

A. Kesimpulan

Kesimpulan yang dapat diambil berdasarkan pembahasan mengenai rancang bangun Aplikasi *Control Panel* GUI LCSAQ adalah sebagai berikut:

1. Penelitian ini berhasil menciptakan sebuah aplikasi yang telah mengimplementasikan GUI dan OOP berbasis Python dalam membangun koneksi menuju *database* lokal, MQTT *broker*, *Raspberry Pi*, dan *file* lainnya, mendeteksi *error* terkait kendala yang terjadi, serta menampilkan data hasil pemantauan partikel udara secara *real-time* dalam satu kesatuan aplikasi *control panel*.
2. Instalasi dan konfigurasi dari Debezium dan Kafka di dalam sistem *Raspberry Pi* berhasil dilakukan dan dihubungkan dengan aplikasi *control panel* dalam membangun koneksi menuju *database* lokal.
3. Aplikasi ini berhasil mengolah data hasil pemantauan partikel udara yang telah didapatkan menjadi informasi kualitas udara dan mengirimkannya ke sebuah halaman web dalam menampilkan hasil pengolahan tersebut secara *real-time* pada *localhost*.

B. Saran

Dari pembuatan Aplikasi *Control Panel* GUI LCSAQ, berikut saran yang dapat digunakan untuk pengembangan penelitian lebih lanjut ke depannya:

1. Aplikasi dapat dikembangkan untuk sistem operasi lainnya terutama yang lebih umum digunakan seperti Windows.

2. Halaman web dapat di-*hosting* dan memiliki domain sendiri sehingga informasi kualitas udara dapat dilihat secara luas dari berbagai perangkat oleh masyarakat.
3. Pengolahan data hasil pemantauan partikel udara dapat dikembangkan lebih jauh, seperti tersedianya prediksi dan arsip data lama dari data partikel udara.

DAFTAR PUSTAKA

- Andre Rajendradika, D., Hermawan Mitrakusuma, W., Teknik Refrigerasi dan Tata Udara, J., & Negeri Bandung, P. (2022). Rancang Bangun Web Server Sistem Akuisisi Data Kondisi Udara Ruangan Berbasis Internet Menggunakan Raspberry Pi. *Prosiding The 13th Industrial Research Workshop and National Seminar*.
- Beniz, D. B., & Espindola, A. M. (2016). Using Tkinter of python to create graphical user interface (GUI) for scripts in LNLs. *Proceedings of the 11th International Workshop on Personal Computers and Particle Accelerator Controls, PCaPAC 2016*.
- Budianto, J. E., & Tony. (2024). Penerapan Fitur “Ambil Tanpa Ribet” Pada Aplikasi Mobile Reine Laundry Untuk Pengalaman Laundry Yang Praktis. *Journal of Information Technology and Computer Science (INTECOMS)*, 7(1).
- Cuevas-Gonzalez, D., Reyna, M. A., Sanchez, M. A., Garcia-Vazquez, J. P., Altamira-Colado, E., & Avitia, R. L. (2024). Test Chamber for Particulate Matter Sensors Measurements in Controlled Environment Through a Python GUI. *Pan American Health Care Exchanges, PAHCE*. <https://doi.org/10.1109/GMEPE/PAHCE62423.2024.10534691>
- Halfacree, G. (2020). *THE OFFICIAL Raspberry Pi Beginner’s Guide* (P. King, Ed.; 4th ed.). Raspberry Pi Trading Ltd.
- Hardiyan, I. A., & Zulistyawan, K. A. (2023). IDENTIFIKASI KONSENTRASI CO, CO₂, NO₂, SO₂, DAN PM₁₀ YANG TERUKUR DI STASIUN GAW

- BUKIT KOTOTABANG SELAMA MUDI K LEBARAN TAHUN 2019-2023. *Megasains*, 14(2), 39–47.
- Hwang, H. C., Park, J. S., & Shon, J. G. (2016). Design and Implementation of a Reliable Message Transmission System Based on MQTT Protocol in IoT. *Wireless Personal Communications*, 91(4). <https://doi.org/10.1007/s11277-016-3398-2>
- Khamaeni, M. G. AL. (2023). IMPLEMENTASI WHITE BOX TESTING BERBASIS PATH PADA APLIKASI BERBASIS WEB. *Jurnal Siliwangi*, 9(1), 8–13.
- Kurniawan, A. (2018). PENGUKURAN PARAMETER KUALITAS UDARA (CO, NO₂, SO₂, O₃ DAN PM₁₀) DI BUKIT KOTOTABANG BERBASIS ISPU. *Jurnal Teknosains*, 7(1). <https://doi.org/10.22146/teknosains.34658>
- Light, R. A. (2017). Mosquitto: server and client implementation of the MQTT protocol. *The Journal of Open Source Software*, 2(13). <https://doi.org/10.21105/joss.00265>
- Londjo, M. F. (2021). Implementasi White Box Testing Dengan Teknik Basis Path Pada Pengujian Form Login. *Jurnal Siliwaangi*, 7(2).
- Lu, Y. C., & Romps, D. M. (2022). Extending the Heat Index. *Journal of Applied Meteorology and Climatology*, 61(10). <https://doi.org/10.1175/JAMC-D-22-0021.1>
- Mehata, S., Linus, L., & Vinayakvitthal, L. (2019). Real Time Data Plotting Tool using Open Source Platform like Raspberry Pi and Python. *2019 Global*

Conference for Advancement in Technology, GCAT 2019.

<https://doi.org/10.1109/GCAT47503.2019.8978280>

Michalski, R. (2018). Information presentation compatibility in a simple digital control panel design: eye-tracking study. *International Journal of Occupational Safety and Ergonomics*, 24(3).

<https://doi.org/10.1080/10803548.2017.1317469>

Okaem, T. T. (2022). PENGEMBANGAN INSTRUMEN PENGUKURAN KUALITAS UDARA MENGGUNAKAN SENSOR PMS7003. *Megasains*, 13(01), 31–38. <https://doi.org/10.46824/megasains.v13i01.103>

Oulasvirta, A., Dayama, N. R., Shiripour, M., John, M., & Karrenbauer, A. (2020). Combinatorial Optimization of Graphical User Interface Designs. *Proceedings of the IEEE*, 108(3), 434–464. <https://doi.org/10.1109/JPROC.2020.2969687>

P.E., N.-K., F.U., O., A.U., D., J.S., I., & N.H., O. (2023). Detailed Study of the Object-Oriented Programming (OOP) Features in Python. *British Journal of Computer, Networking and Information Technology*, 6(1), 83–93. <https://doi.org/10.52589/bjcnit-facsojao>

Praniffa, A. C., Syahri, A., Sandes, F., Fariha, U., Giansyah, Q. A., & Hamzah, M. L. (2023). Pengujian black box dan white box system informasi parkir berbasis web. *Jurnal Testing Dan Implementasi Sistem Informasi*, 1(1).

Rahmaliani, I., Abdussalaam, F., Gunawan, E., & Soelistijaningrum, M. (2023). Tata Kelola Rekam Medis Berbasis Elektronik Dalam Pelaporan Mortalitas Pasien Rawat Inap Menggunakan Metode Agile Software Development.

- INOVTEK Polbeng - Seri Informatika*, 8(2).
<https://doi.org/10.35314/isi.v8i2.3532>
- Rais, M. (2019). Penerapan Konsep Object Oriented Programming Untuk Aplikasi Pembuat Surat. *Jurnal PROtek*, 06(2), 96–101.
- Ramli, M., Sultan, P., & Abidin, M. Z. (2021). *Python GUI Programming with Tkinter*. Politeknik Muadzam Shah.
<https://www.researchgate.net/publication/371730724>
- Raptis, T. P., & Passarella, A. (2023). A Survey on Networked Data Streaming with Apache Kafka. *IEEE Access*, 11, 85333–85350.
<https://doi.org/10.1109/ACCESS.2023.3303810>
- Reuben, E. C., & Meng, L. Y. (2022). GUI Composer: A Lightweight GUI Software Development Framework for WSN, I.o.T and Control System Application. *2022 5th International Conference on Mechatronics, Robotics and Automation, ICMRA 2022*.
<https://doi.org/10.1109/ICMRA56206.2022.10145621>
- Rimbawati, Setiadi, H., Ananda, R., & Ardiansyah, M. (2019). Perancangan Alat Pendeteksi Kebocoran Tabung Gas LPG Dengan Menggunakan Sensor MQ-6 Untuk Mengatasi Bahaya Kebakaran. *Journal of Electrical Technology*, 4(2), 53–58.
- Romano, Fabrizio. (2018). *Learn Python Programming : a Beginner's Guide to Learning the Fundamentals of Python Language to Write Efficient, High-Quality Code, 2nd Edition*. Packt Publishing Ltd.

- Sayar, A., Arslan, S., Cakar, T., Ertugrul, S., & Akcay, A. (2023). High-Performance Real-Time Data Processing: Managing Data Using Debezium, Postgres, Kafka, and Redis. *2023 Innovations in Intelligent Systems and Applications Conference, ASYU 2023*.
<https://doi.org/10.1109/ASYU58738.2023.10296737>
- Setiawan, F. B., Kusuma, H. W., Riyadi, S., & Leonardus Heru PratomO. (2022). Penerapan PI Cam Menggunakan Program Berbasis Raspberry PI 4. *CYCLOTRON: Jurnal Teknik Elektro*, 5(2), 51–56.
- Sie, J. B. L., Izmy Alwiah Musdar, & Syamsul Bahri. (2022). Pengujian White Box Testing Terhadap Website Room Menggunakan Teknik Basis Path. *KHARISMA Tech*, 17(2), 45–57.
<https://doi.org/10.55645/kharismatech.v17i2.235>
- Sindy Nova, Nurul Khotimah, & Maria Y Aryati Wahyuningrum. (2024). PEMANFAATAN CHATBOT MENGGUNAKAN NATURAL LANGUAGE PROCESSING UNTUK PEMBELAJARAN DASAR-DASAR GUI TKINTER PADA BAHASA PEMROGRAMAN PYTHON. *Jurnal Ilmiah Teknik*, 3(1), 58–65. <https://doi.org/10.56127/juit.v3i1.1162>
- Siswanto, E. A. (2022). *Rancang Bangun Dynamic REST API untuk Memenuhi Kebutuhan Pemberian Data PT XYZ kepada Anak Perusahaan* [Bachelor Thesis]. Universitas Multimedia Nusantara.
- Soni, D., & Makwana, A. (2017). A Survey on MQTT: a Protocol of Internet of Things (IOT). *International Conference on Telecommunication, Power Analysis and Computing Techniques (ICTPACT-2017)*, 20(April).

- Sumirat, L. P., Cahyono, D., Kristyawan, Y., & Kacung, S. (2023). *DASAR-DASAR Rekayasa Perangkat Lunak* (1st ed.). Madza Media. www.madzamedia.co.id
- Thangarajah, V., MMM, F., & A.L, S. S. (2019). PYTHON CURRENT TREND APPLICATIONS- AN OVERVIEW. *International Journal of Advance Engineering and Research Development*, 6(10), 6–12.
- Wiora, A., Wiora, J., & Kasprzyk, J. (2024). Indication Variability of the Particulate Matter Sensors Dependent on Their Location. *Sensors*, 24(5). <https://doi.org/10.3390/s24051683>
- Zahtamal, Restila, R., & Nazriati, E. (2023). Korelasi Sebaran Titik Panas dengan Kualitas Udara di Kota Pekanbaru. *Prepotif: Jurnal Kesehatan Masyarakat*, 7(1).