

**RANCANG BANGUN *ROBOT ASSISTANT* ARDUINO UNTUK
MEMBERI PERINTAH LAPTOP DAN PERANGKAT LISTRIK
DENGAN BAHASA PYTHON BERBASIS *ARTIFICIAL
INTELLIGENCE* (AI)**

PROYEK AKHIR



Oleh :

AHMAD AKMAL FAHMI
NIT. 30121003

**PROGRAM STUDI DIPLOMA 3 TEKNIK LISTRIK BANDARA
POLITEKNIK PENERBANGAN SURABAYA
2024**

**RANCANG BANGUN ROBOT ASSISTANT ARDUINO UNTUK
MEMBERI PERINTAH LAPTOP DAN PERANGKAT LISTRIK
DENGAN BAHASA PYTHON BERBASIS ARTIFICIAL
INTELLIGENCE (AI)**

PROYEK AKHIR

Diajukan sebagai Salah Satu Syarat untuk Mendapatkan Gelar Ahli Madya
(A.Md.) pada Program Studi Diploma 3 Teknik Listrik Bandara



Oleh :

AHMAD AKMAL FAHMI
NIT. 30121003

**PROGRAM STUDI DIPLOMA 3 TEKNIK LISTRIK BANDARA
POLITEKNIK PENERBANGAN SURABAYA
2024**

LEMBAR PERSETUJUAN

RANCANG BANGUN ROBOT ASSISTANT ARDUINO UNTUK MEMBERI
PERINTAH LAPTOP DAN PERANGKAT LISTRIK DENGAN BAHASA
PYTHON BERBASIS ARTIFICIAL INTELLIGENCE (AI)

Oleh :

Ahmad Akmal Fahmi

NIT. 30121003

Disetujui untuk diujikan pada :

Surabaya, 6 Agustus 2024

Pembimbing I

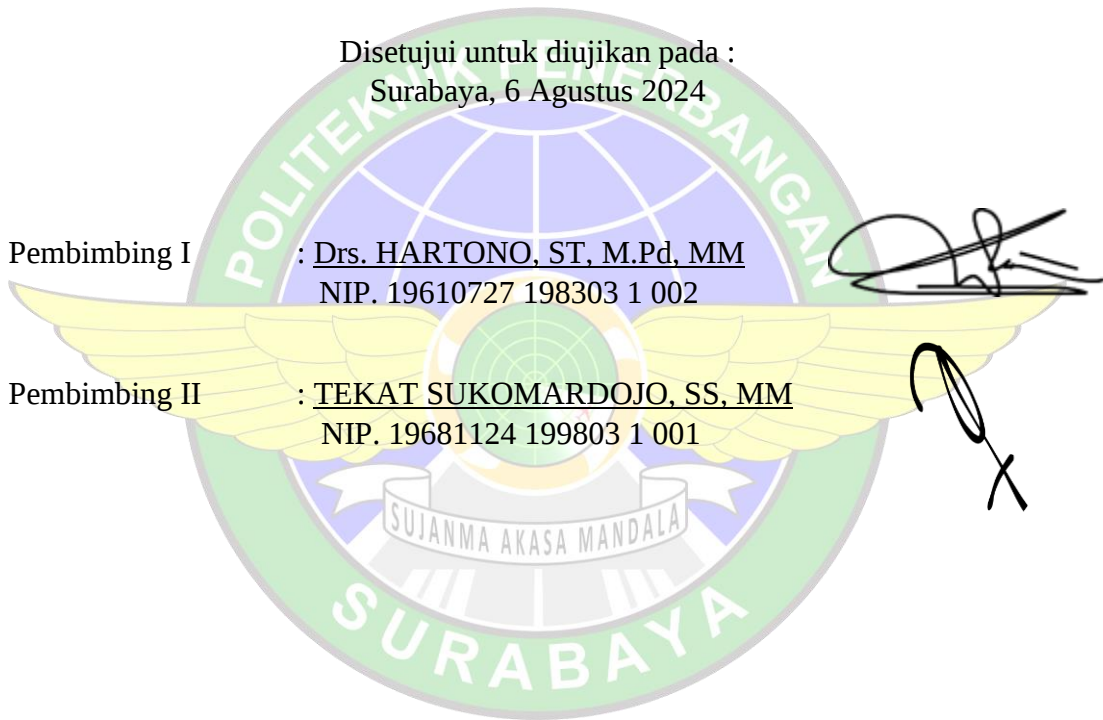
: Drs. HARTONO, ST, M.Pd, MM

NIP. 19610727 198303 1 002

Pembimbing II

: TEKAT SUKOMARDOJO, SS, MM

NIP. 19681124 199803 1 001



LEMBAR PENGESAHAN

RANCANG BANGUN *ROBOT ASSISTANT ARDUINO* UNTUK MEMBERI
PERINTAH LAPTOP DAN PERANGKAT LISTRIK DENGAN BAHASA
PYTHON BERBASIS *ARTIFICIAL INTELLIGENCE (AI)*

Oleh :
Ahmad Akmal Fahmi
NIT. 30121003

Telah dipertahankan dan dinyatakan lulus pada Ujian Proyek Akhir
Program Pendidikan Diploma 3 Teknik Listrik Bandar Udara
Politeknik Penerbangan Surabaya
pada tanggal : 6 Agustus 2024

Panitia Penguji :

1. Ketua : Dr. KUSTORI, ST, MM
NIP. 19590305 198503 1 002
2. Sekretaris : TEKAT SUKOMARDOJO, SS, MM
NIP. 19681124 199803 1 001
3. Anggota : Drs. HARTONO, ST, M.Pd, MM
NIP. 19610727 198303 1 002

Ketua Program Studi
D3 Teknik Listrik Bandara

Dr. GUNAWAN SAKTI, S.T., M.T.
NIP. 19881001 200912 1 003

ABSTRAK

RANCANG BANGUN *ROBOT ASSISTANT* ARDUINO UNTUK MEMBERI PERINTAH LAPTOP DAN PERANGKAT LISTRIK DENGAN BAHASA PYTHON BERBASIS *ARTIFICIAL INTELLIGENCE* (AI)

Oleh:

Ahmad Akmal Fahmi

NIT: 30121003

Penelitian ini mengembangkan sebuah robot asisten yang revolusioner berbasis Arduino yang mampu mengendalikan laptop dan perangkat listrik melalui bahasa pemrograman Python yang diintegrasikan dengan kecerdasan buatan (AI). Sistem ini dirancang untuk mengoptimalkan interaksi manusia dan mesin melalui perintah suara, menciptakan sebuah inovasi yang komunikatif. Robot ini memiliki kapabilitas luas untuk menjalankan berbagai perintah, seperti mengontrol perangkat listrik, menampilkan dan menganalisis data bebannya, serta menangani perintah khusus lainnya pada laptop.

Rancangan alat ini terdiri dari dua komponen utama yang saling terintegrasi yaitu sebuah mikrokontroler Arduino sebagai pengendali perangkat keras dan Python sebagai pengendali perangkat lunak. Sistem ini dilengkapi dengan antarmuka grafis yang diakses melalui metode login, memastikan kemudahan dan keamanan penggunaan. Perintah yang diterima diproses oleh kedua komponen ini untuk menghasilkan output utama pada laptop, lampu, perangkat listrik, serta modul LED display.

Pada hasil penelitian sistem dapat mengeksekusi perintah yang diprogram, menampilkan antarmuka interaktif, dan mengontrol perangkat listrik secara efektif. Penggunaan platform *open-source* Arduino, Python, dan berbagai *library open-source* memberikan fleksibilitas dan potensi pengembangan lebih lanjut yang besar. Fitur-fitur seperti pencarian informasi, kontrol aplikasi, analisa data kelistrikan, dan perintah lainnya memperlihatkan kemampuan sistem yang komprehensif dan optimal dalam mengeksekusi perintah.

Robot asisten berbasis Arduino dan Python yang dilengkapi dengan kecerdasan buatan ini merupakan solusi inovatif dan efektif untuk meningkatkan efisiensi dan kemudahan dalam pekerjaan manusia. Sistem ini menunjukkan potensi besar untuk pengembangan lebih lanjut dalam memanfaatkan teknologi kecerdasan buatan, menciptakan sebuah platform yang adaptif dan integratif untuk berbagai aplikasi di masa depan.

Kata Kunci : Arduino, Python, Robot Asisten, Kecerdasan Buatan

ABSTRACT

DESIGN BUILD ARDUINO ROBOT ASSISTANT TO COMMAND LAPTOP AND ELECTRICAL DEVICES WITH PYTHON LANGUAGE BASED ON ARTIFICIAL INTELLIGENCE (AI)

By:

Ahmad Akmal Fahmi

NIT: 30121003

This research develops a revolutionary Arduino-based robot assistant capable of controlling laptops and electrical devices through the Python programming language integrated with artificial intelligence (AI). The system is designed to optimize human-machine interaction through voice commands, creating a communicative innovation. The robot has extensive capabilities to execute various commands, such as controlling electrical devices, displaying and analyzing their load data, and handling other special commands on laptops.

The design of this tool consists of two main components that are integrated with each other, namely an Arduino microcontroller as a *hardware* controller and Python as a *software* controller. The system is equipped with a graphical *interface* that is accessed through a login method, ensuring ease and security of use. The received commands are processed by these two components to produce the main outputs on the laptop, lights, electrical devices, as well as the LED display module.

In the research results the system can execute programmed commands, display an interactive *interface*, and control electrical devices effectively. The use of the open-source platform Arduino, Python, and various open-source libraries provides great flexibility and potential for further development. Features such as information retrieval, application control, electrical data analysis, and other commands demonstrate the system's comprehensive and optimized ability to execute commands.

This Arduino and Python-based assistant robot equipped with artificial intelligence is an innovative and effective solution to improve efficiency and convenience in human work. The system shows great potential for further development in utilizing artificial intelligence technology, creating an adaptive and integrative platform for various applications in the future.

Keywords : Arduino, Python, Robot Assistant, Artificial Intelligence

PERNYATAAN KEASLIAN HAK CIPTA

Saya yang bertanda tangan di bawah ini :

Nama : Ahmad Akmal Fahmi
NIT : 30121003
Program Studi : D3 Teknik Listrik Bandara
Judul Proyek Akhir : Rancang Bangun *Robot Assistant* Arduino Untuk Memberi Perintah Laptop Dan Perangkat Listrik Dengan Bahasa Python Berbasis *Artificial Intelligence* (AI)

dengan ini menyatakan bahwa :

1. Proyek Akhir ini merupakan karya asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Politeknik Penerbangan Surabaya maupun di Perguruan Tinggi lain, serta dipublikasikan, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
2. Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan Hak Bebas Royalti Non Eksklusif (*Non-Exclusive Royalty-Free Right*) kepada Politeknik Penerbangan Surabaya beserta perangkat yang ada (jika diperlukan). Dengan hak ini, Politeknik Penerbangan Surabaya berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan Proyek Akhir saya dengan tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya. Apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi dan Akademi Penerbangan.

Surabaya, 6 Agustus 2024
Yang membuat pernyataan



Ahmad Akmal Fahmi
NIT. 30121003

KATA PENGANTAR

Dengan memanjatkan puja dan puji syukur atas kehadiran Allah SWT, karena berkah rahmat dan karuniaNya maka penulisan Proyek Akhir dapat selesai sesuai dengan waktu yang telah ditentukan. Penyusunan Proyek Akhir ini dimaksudkan sebagai syarat untuk menyelesaikan pendidikan program studi Diploma III Teknik Listrik Bandara Angkatan XVI

Pada kesempatan ini penulis menyampaikan terima kasih kepada :

1. Tuhan yang Maha Esa yang telah memberikan rahmat dan berkat-Nya sehingga dapat melaksanakan penyusunan Proyek Akhir dengan baik.
2. Bapak Ahmad Bahrawi, S.E., M.T. selaku Direktur Politeknik Penerbangan Surabaya.
3. Bapak Dr. Gunawan Sakti, S.T., M.T. selaku Ketua Program Studi Teknik Listrik Bandara Politeknik Penerbangan Surabaya.
4. Bapak Drs. Hartono, ST, M.Pd, MM selaku Dosen Pembimbing I yang telah memberi banyak ilmu dan bimbingan.
5. Bapak Tekat Sukomardojo, SS, MM selaku Dosen Pembimbing II yang telah memberikan saran dan masukan demi sempurnanya proyek akhir ini.
6. Bapak Rifdian Indrianto Sudjoko, S.T., MM., M.T. selaku Dosen Pembimbing yang berjasa dalam memberikan arahan.
7. Seluruh dosen dan civitas akademika Prodi Teknik Listrik Bandara Politeknik Penerbangan Surabaya, atas ilmu dan bimbingannya.
8. Orang Tua dan keluarga yang selalu memberikan semangat dan doa.
9. Rekan-rekan D3 Teknik Listrik Bandara Angkatan XVI, dan adik-adik D3 Teknik Listrik Bandara Angkatan XVII atas semangat dan kerjasamanya.
10. Orang-orang spesial yang selalu mendorong untuk terus hidup dan menggapai mimpi serta memberi pelajaran tentang arti hidup yang sebenarnya.

Penulis menyadari bahwa penulisan Proyek Akhir ini masih jauh dari sempurna, maka dari itu penulis mohon maaf atas segala kekurangan. Penulis berharap agar Proyek Akhir ini dapat memberi manfaat dan menjadi refrensi untuk kemajuan kehidupan yang akan datang.

Surabaya, 6 Agustus 2024

Penulis



AHMAD AKMAL FAHMI

NIT. 30121003

MOTTO

“I’LL BE MAIN CHARACTER IN EVERY UNIVERSE”



DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
ABSTRAK	iv
ABSTRACT	v
PERNYATAAN KEASLIAN HAK CIPTA	vi
KATA PENGANTAR.....	vii
MOTTO	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
 BAB 1 PENDAHULUAN	 1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	5
1.6 Sistematika Penulisan.....	6
 BAB 2 LANDASAN TEORI	 7
2.1 Teori Penunjang	7
2.1.1 Robot Arduino.....	7
2.1.2 Arduino Uno R3.....	8
2.1.3 Laptop	9
2.1.4 PCB (<i>Printed Circuit Board</i>)	10
2.1.5 Motor Servo S90	11
2.1.6 Modul LCD 16x2	12
2.1.7 Modul Relay.....	12
2.1.8 Sensor PZEM-004T	13
2.1.9 Lampu LED (<i>Light Emitting Diode</i>).....	14
2.1.10 Program Python.....	15
2.1.11 Arduino IDE.....	16
2.1.12 Microsoft Visual Studio	17
2.1.13 Gemini API	18
2.1.14 PyQt5	19
2.1.15 <i>Artificial Intelligence</i> (AI) atau Kecerdasan Buatan.....	20
2.2 Kajian Penelitian Terdahulu yang Relevan	21
 BAB 3 METODOLOGI PENELITIAN.....	 24
3.1 Desain Penelitian.....	24

3.2 Perancangan Alat.....	24
3.2.1 Desain Alat.....	24
3.2.3 Cara Kerja Alat	26
3.3 Komponen Alat	29
3.3.1 Perangkat Keras (<i>Hardware</i>)	29
3.3.2 Perangkat Lunak (<i>Software</i>).....	30
3.4 Teknik Pengujian.....	31
3.5 Teknik Analisis Data	33
3.6 Tempat dan Waktu Penelitian	34
BAB 4 HASIL DAN PEMBAHASAN	35
4.1 Hasil Penelitian	35
4.1.1 Pembuatan Komponen Perangkat Keras.....	35
4.1.2 Pembuatan Design PCB	36
4.1.3 Pencetakan Jalur Tembaga.....	37
4.1.4 Penyolderan Jalur PCB	37
4.1.5 Perakitan Komponen Perangkat Keras.....	38
4.1.6 Perakitan Body Robot	39
4.1.7 Pembuatan Sistem Perangkat Lunak Menggunakan Python.....	39
4.1.8 Pemrograman Pada Mikrokontroller.....	41
4.1.9 Integrasi Program Python dan Arduino.....	43
4.2 Pembahasan Hasil Penelitian	44
4.2.1 Pengujian Perangkat Keras	44
4.2.2 Pengujian Perangkat Lunak.....	51
4.2.3 Analisa Hasil Pengujian Sistem	76
4.3 Kelebihan dan Kekurangan Alat	79
4.1.1 Kelebihan Alat	79
4.3.2 Kekurangan Alat	79
BAB 5 PENUTUP.....	81
5.1 Kesimpulan.....	81
5.2 Saran.....	82
DAFTAR PUSTAKA	83
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1 Robot Arduino.....	8
Gambar 2.2 Arduino Uno R3.....	8
Gambar 2.3 Laptop.....	10
Gambar 2.4 PCB	11
Gambar 2.5 Motor Servo S90	12
Gambar 2.6 Modul LCD 16x2	12
Gambar 2.7 Modul Relay.....	13
Gambar 2.8 Sensor PZEM-004T.....	14
Gambar 2.10 Lampu LED.....	15
Gambar 2.11 Program Python.....	16
Gambar 2.12 Arduino IDE	17
Gambar 2.13 Microsoft Visual Studio	18
Gambar 2.14 Gemini API	19
Gambar 2.15 PyQt5.....	20
Gambar 2.16 Artificial Intelligence	21
Gambar 3.1 Desain Alat (<i>Software</i>).....	25
Gambar 3.2 Desain Alat (<i>Hardware</i>).....	25
Gambar 3.3 Flowchart Cara Kerja Alat	26
Gambar 4.1 Design PCB	36
Gambar 4.2 Rangkaian Schematic PCB.....	36
Gambar 4.3 Penyolderan Jalur PCB.....	38
Gambar 4.4 Perakitan Komponen Perangkat Keras.....	38
Gambar 4.5 Perakitan Body Robot	39
Gambar 4.6 Install <i>Library</i> Python	41
Gambar 4.7 Program Motor Servo S90.....	41
Gambar 4.8 Program Sensor PZEM-004T.....	42
Gambar 4.9 Program Arduino.....	43
Gambar 4.10 Integrasi Program Python dan Arduino.....	44
Gambar 4.11 Pengujian Arduino.....	44
Gambar 4.12 Skematik Arduino Uno R3.....	45
Gambar 4.13 Pengujian Motor Servo.....	45
Gambar 4.14 Pengujian Relay 1.....	46
Gambar 4.15 Pengujian Relay 2.....	46
Gambar 4.16 Skematik Relay	46
Gambar 4.17 Pengujian LCD 16x2	47
Gambar 4.18 Skematik LCD 16x2.....	47
Gambar 4.19 Pengujian LED 1	47
Gambar 4.20 Pengujian LED 2	48
Gambar 4.21 Pengujian Sensor PZEM-004T.....	49
Gambar 4.22 Skematik PZEM-004T	49

Gambar 4.23 Pengujian Arduino IDE	51
Gambar 4.24 Tampilan Login Bahasa Indonesia	52
Gambar 4.25 Tampilan Login Bahasa Inggris	52
Gambar 4.26 Tampilan Login Apabila Terdapat Kesalahan	52
Gambar 4.27 Tampilan Login <i>Show Password</i>	53
Gambar 4.28 Pengujian Respon Sapaan	55
Gambar 4.29 Pengujian Pertanyaan Umum	56
Gambar 4.30 Pengujian Tutup Jendela	57
Gambar 4.31 Pengujian Baca Tegangan	58
Gambar 4.32 Pengujian Baca Arus	58
Gambar 4.33 Pengujian Baca Daya	59
Gambar 4.34 Kontrol Aplikasi	60
Gambar 4.35 Implementasi Kontrol Aplikasi	61
Gambar 4.36 Pengujian Mode Ketik.....	62
Gambar 4.37 Implementasi Mode Ketik.....	62
Gambar 4.38 Pengujian Hitung Matematika.....	63
Gambar 4.39 Pengujian Terjemah Kalimat.....	64
Gambar 4.40 Pengujian Pencarian Video Youtube.....	65
Gambar 4.41 Pengujian Kontrol Video.....	66
Gambar 4.42 Pengujian Pencatatan Data Sensor	67
Gambar 4.43 Pengujian Deteksi Objek 1	68
Gambar 4.44 Pengujian Deteksi Objek 2	68
Gambar 4.45 Deteksi Gesture <i>Scroll Down</i>	70
Gambar 4.46 Deteksi Gesture <i>Scroll Up</i>	70
Gambar 4.47 Pengujian Penyimpanan Dokumen	72
Gambar 4.48 Analisa Data Sensor	73
Gambar 4.49 Pengujian Kontrol Kamera.....	74
Gambar 4.50 Pengujian Penjelasan Konsep.....	75
Gambar 4.51 Pengujian <i>Google Speech Recognition</i>	76
Gambar 4.52 Pengujian <i>Generative AI</i> dari Google.....	76
Gambar 4.53 Pengujian YOLOv5.....	77
Gambar 4.54 Pengujian MediaPipe pada Kamera	77
Gambar 4.55 Pengujian MediaPipe pada Deteksi Tangan.....	78
Gambar 4.56 Pengujian MediaPipe pada Deteksi Objek	78
Gambar 4.57 Pengujian Google Translate	79

DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino Uno R3	9
Tabel 2.2 Kajian Penelitian Terdahulu yang Relevan.....	22
Tabel 3.1 TeknikPengujian Integrasi Arduino dan Microsoft Visual Studio	31
Tabel 3.2 Teknik Pengujian Kontrol Lampu	32
Tabel 3.3 Teknik Pengujian Sensor PZEM-004T	32
Tabel 3.4 Teknik Pengujian Tampilan Modul LCD 16x2	32
Tabel 3.5 Teknik Pengujian Respon Sistem	33
Tabel 3.6 <i>Timeline</i> Waktu Penelitian	34
Tabel 4.1 Deteksi nyala lampu.....	48
Tabel 4.2 Pembacaan Data Sensor	50
Tabel 4.3 Pengujian Tampilan Login.....	53
Tabel 4.4 Pengujian Respon Sapaan	55
Tabel 4.5 Pengujian Pertanyaan Umum.....	56
Tabel 4.6 Pengujian Tutup Jendela	57
Tabel 4.7 Pengujian Baca Data Kelistrikan	59
Tabel 4.8 Pengujian Kontrol Aplikasi.....	61
Tabel 4.9 Pengujian Mode Ketik	63
Tabel 4.10 Pengujian Hitung Matematika	63
Tabel 4.11 Pengujian Terjemah Kalimat	64
Tabel 4.12 Pengujian Pencarian Video Youtube	65
Tabel 4.13 Pengujian Kontrol Video	66
Tabel 4.14 Pengujian Pencatatan Data Sensor	67
Tabel 4.15 Pengujian Deteksi Objek.....	68
Tabel 4.16 Tabel Pengujian Deteksi Gerakan Tangan.....	70
Tabel 4.17 Pengujian Penyimpan Dokumen.....	72
Tabel 4.18 Analisa Data Sensor	73
Tabel 4.19 Pengujian Kontrol Kamera.....	74
Tabel 4.20 Pengujian Penjelasan Konsep	75

DAFTAR LAMPIRAN

Lampiran A. STANDAR OPERATIONAL PROCEDURE (SOP).....	A-1
Lampiran B. PROGRAM ALAT	B-1
Lampiran C. Daftar Riwayat Hidup	C-1



DAFTAR PUSTAKA

- Abrar, A., Teknik Elektro, J., & Negeri Balikpapan, P. (2020). Rancang Bangun IoT Robotic Car Menggunakan Raspberry Pi dan Python. *Jurnal Sains Terapan*, 6(1), 33–38.
- Acharya, S., Jain, U., Kumar, R., Prajapat, S., Suthar, S., & Kumar Jain, R. (2021). JARVIS: A Virtual Assistant for Smart Communication. *International Journal of Advances in Engineering and Management (IJAEM)*, 3(6), 460. <https://doi.org/10.35629/5252-0306460465>
- Adha, U., & Singh, S. (2023). *Jarvis : Virtual Voice Command Desktop Assistant*. 8(11), 1675–1678.
- Ardiansyah, R. A. (2016). Design of an electronic narrator on assistant robot for blind people. *MATEC Web of Conferences*, 42, 1–5. <https://doi.org/10.1051/mateconf20164203013>
- Bhalke, S., Paygude, S., Lohar, S., & Patil, A. (2022). *Jarvis Voice-based assistant using machine learning*. May 2021, 8–11. <https://doi.org/10.17148/IJARCCCE.2021.105156>
- Brynjolfsson, E., & McAfee, A. (2017). The Business of Artificial Intelligence: what it can and cannot do for your organization. *Harvard Business Review Digital Articles*.
- Chromiński, K., Benko, L., Hernández-Figueroa, Z. J., González-Domínguez, J. D., & Rodríguez-del-Pino, J. C. (2021). Python Fundamentals. *Python Fundamentals*, c. <https://doi.org/10.17846/fpvai-2021-14>
- Das, S., Saha, A., Chakraborty, A., Thakur, S., & Chatterjee, S. (2022). *J.a.r.v.i.s - an ai technology based personal windows assistant system 1*. 9(2), 385–391.
- Deep, G., & Services, R. C. (2016). *Department of Computer Science Department of Information Technology* , March.
- Divya, K., Rahul, P., Raja, A., Divya, K., & Rahul, P. (2020). *EasyChair Preprint № 4818 Jarvis AI using PYTHON JARVIS AI USING PYTHON*.
- Fapal, A., Kanade, T., Janrao, B., Kamble, M., & Raule, M. (2021). Personal Virtual Assistant for Windows using Python. *International Research Journal of Modernization in Engineering*, 03(07), 485–491. www.irjmets.com

- Fowler, M. (2018). *Refactoring: Improving the Design of Existing Code*. 2nd ed. Addison-Wesley.
- Fryer, C. E., Luker, J. A., McDonnell, M. N., & Hillier, S. L. (2016). Self management programmes for quality of life in people with stroke. *Cochrane Database of Systematic Reviews*, 2016(8). <https://doi.org/10.1002/14651858.CD010442.pub2>
- Jain, R. K., Sharma, V., Kardam, R., & Rani, M. (2021). Artificial Intelligence Based A Communicative Virtual Voice Assistant Using Python & Visual Code Technology. *World Journal of Research and Review (WJRR)*, 13(5), 23–26.
- Jarvis, R., Dempsey, K., Gutierrez, G., Lewis, D., Rouleau, K., & Stone, B. (2017). Peer Coaching That Works: The Power of Reflection and Feedback in Teacher Triad Teams. *McREL International*, 1–13. https://www.mcrel.org/wp-content/uploads/2017/09/McREL_WhitePaper_InstructionalCoaching_Sept2017_web1.pdf
- Kakde, P., Shah, P., Meshram, A., Umap, Y., Misal, Y., & Anjekar, A. (2022). JARVIS – THE VIRTUAL ASSISTANT. 05, 1382–1385.
- Kaplan, A., & Haenlein, M. (2019). Siri, Siri, in the World: Artificial Intelligence in Business and What We Need to Do About It. *Business Horizons*, 62(1), 15–25. doi:10.1016/j.bushor.2018.08.002
- Kaswan, K. S., Dhatteval, J. S., & Balamurugan, B. (2023). Python for Beginners. In *Python for Beginners*. <https://doi.org/10.1201/9781003202035>
- Khobragade, S. (1970). *Jarvis, Digital Life Assistant*.
- Krishnan, J., & Reed, T. (2019). *SEVA: A Systems Engineer's Virtual Assistant*.
- Kuhlman, D. (2013). A Python Book: Beginning Python, Advanced Python, and Python Exercises. *A Python Book*, 1–227. <http://www.davekuhlman.org>
- Lebrun, T., & Audet, R. (2020). *Artificial Intelligence and the Book Industry White Paper*. 2020. <https://doi.org/10.5281/zenodo.4036258>
- Lee, S. (2023). "The Evolution of Printed Circuit Boards in Modern Electronics." Proceedings of the IEEE International Conference on Electronics Engineering, 78-86.
- Li, F.-F. (2024). Artificial Intelligence: Challenges and Opportunities. *Science*, 369(6504), 1204-1209. doi:10.1126/science.369.6504.1204

- Matarić, M. (2020). "The Robotics Revolution: Building a New Generation of Robots."
- Page, L. C., & Gehlbach, H. (2017). How an Artificially Intelligent Virtual Assistant Helps Students Navigate the Road to College. *AERA Open*, 3(4). <https://doi.org/10.1177/2332858417749220>
- Pandita, R., Bucuvalas, S., Bergier, H., Chakarov, A., & Richards, E. (2018). Towards J.A.R.V.I.S. for *Software Engineering*: Lessons Learned in Implementing a Natural Language Chat *Interface*. *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 779–782. www.aaai.org
- Pollack, M. E., Engberg, S., Matthews, J. T., Dunbar-jacob, J., McCarthy, C. E., & Thrun, S. (2002). Pearl: A mobile robotic assistant for the elderly. *Architecture*, 2002, 85–91. <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Pearl+:+A+Mobile+Robotic+Assistant+for+the+Elderly#0>
- Pratiwi, H. I., & Handoko, P. (2019). Aplikasi Arduino Uno Pada Modul Robot Manusia Sebagai Mediasi Pengajaran Desain, Narasi Dan Simulasi. *Widyakala Journal*, 6(2), 150. <https://doi.org/10.36262/widyakala.v6i2.196>
- Rabbi, F., Park, T., Fang, B., Zhang, M., & Lee, Y. (2018). When Virtual Reality Meets Internet of Things in the Gym. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(2), 1–21. <https://doi.org/10.1145/3214281>
- Rani, D. Y., Kaur, M. G., Rana, H., . S., & . N. (2023). JARVIS: A Virtual Assistant. *International Journal for Research in Applied Science and Engineering Technology*, 11(2), 751–753. <https://doi.org/10.22214/ijraset.2023.49111>
- Rissland, E. L. (1988). Artificial Intelligence and Legal Reasoning: A Discussion of the Field and Gardner's Book. *AI Magazine*, 9(3), 45–45. <https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/942>
- Stallings, W. (2010). *Computer Organization and Architecture*. 8th ed. Upper Saddle River, NJ: Prentice Hall.
- Sukri, H., Laksono, D. T., Laksono, D. T., & Ulum, M. (2021). Ultrasonic Signal Implementation in Arduino-Based Obstacle Robot Control System. *E3S Web of Conferences*, 328, 0–4. <https://doi.org/10.1051/e3sconf/202132802008>
- Susanto, D., Mujaahid, F., Syahputra, R., & Putra, K. T. (2019). Open Source System for Smart Home Devices Based on Smartphone Virtual Assistant. *Journal of Electrical Technology UMY*, 3(1), 1–7. <https://doi.org/10.18196/jet.3147>

Turban, E., Volonino, L., & Wood, G. R. (2021). *Information Technology for Management*.

V., G., C K, G., Kottamasu, M. S. V., & Kumar, N. P. (2021). The Voice Enabled Personal Assistant for Pc using Python. *International Journal of Engineering and Advanced Technology*, 10(4), 162–165.
<https://doi.org/10.35940/ijeat.d2425.0410421>

Vora, J., Yadav, D., Jain, R., & Gupta, J. (2021). JARVIS: A PC Voice Assistant. *International Journal of Advance Study and Research Work*, 4(7), 15–23.
<https://doi.org/10.5281/zenodo.5323068>

Wang, Z., Cai, S., Liu, A., Jin, Y., Hou, J., Zhang, B., Lin, H., He, Z., Zheng, Z., Yang, Y., Ma, X., & Liang, Y. (2023). JARVIS-1: Open-World Multi-task Agents with Memory-Augmented Multimodal Language Models. 1–28.
<http://arxiv.org/abs/2311.05997>

Widiasari, C. (2020). Sistem Monitoring Daya Listrik dan Pengontrolan Perangkat Elektronik Berbasis IoT. *Seminar Nasional Teknologi Informasi Komunikasi Dan Industri*, 342–349.



LAMPIRAN

Lampiran A. STANDAR OPERATIONAL PROCEDURE (SOP)

RANCANG BANGUN *ROBOT ASSISTANT ARDUINO* UNTUK MEMBERI PERINTAH LAPTOP DAN PERANGKAT LISTRIK DENGAN BAHASA *PYTHON* BERBASIS *ARTIFICIAL INTELLIGENCE (AI)*

1. Tujuan

Menetapkan prosedur yang standar untuk menyusun, mengoperasikan, dan menguji *robot assistant* yang berbasis Arduino dan Python agar alat berfungsi sesuai dengan desain dan tujuan yang telah ditentukan.

2. Prosedur

A. Persiapan Alat dan Komponen

1. Perangkat Keras (*Hardware*)

- Arduino Uno R3
Pastikan Arduino Uno R3 dalam kondisi baik dan terhubung dengan kabel USB.
- *Robot Assistant Body*
Pastikan body robot terpasang dengan baik dan servo motor terhubung dengan benar.
- Sensor PZEM-004T
Pastikan sensor PZEM-004T terhubung dengan benar ke Arduino.
- Modul LCD 16x2
Pastikan modul LCD terpasang dan terhubung dengan Arduino.
- Relay dan Lampu LED
Pastikan relay dan lampu LED terhubung dengan benar dan dapat dikontrol oleh Arduino.

2. Perangkat Lunak (*Software*)

- Python dan Library Pendukung
 - a. Pastikan Python versi Python 3.11.9 atau lebih baru telah terinstall.

- b. Pastikan library pendukung seperti PyQt5, SpeechRecognition, pygame, gTTS, pyautogui, googletrans, pyserial, dan google-generative-ai telah terinstall.

- Microsoft Visual Studio

Pastikan Microsoft Visual Studio terinstall dan berfungsi dengan baik.

- Arduino IDE

Pastikan Arduino IDE terinstall dan terhubung dengan Arduino Uno R3.

3. Pemasangan dan Pengaturan

- Instalasi Perangkat Keras

- a. Pasang Arduino Uno R3 pada board dan pastikan terhubung ke komputer melalui kabel USB.
- b. Pasang dan hubungkan sensor PZEM-004T, relay, modul LCD 16x2, dan lampu LED sesuai diagram blok yang telah dirancang.
- c. Pasang bodi robot dan hubungkan servo motor sesuai dengan posisi yang telah ditentukan.

- Instalasi Perangkat Lunak

- a. Buka Arduino IDE dan unggah program ke Arduino Uno R3.
- b. Buka Microsoft Visual Studio dan jalankan program Python untuk mengintegrasikan dengan Arduino.

- Pengoperasian Alat

- a. Hubungkan sumber 220 V ke *input power*
- b. Hubungkan Arduino ke komputer.
- c. Memulai Sistem dengan membuka program Microsoft Visual Studio dan jalankan program Python.
- d. Sistem akan menampilkan GUI (*Graphical User Interface*) pada layar.

- Mengoperasikan Sistem

- a. Login
- b. Masukkan *username* dan *password* pada tampilan login GUI.
- c. Perintah Suara
- d. Gunakan perintah suara untuk mengontrol *robot assistant*

Lampiran B. PROGRAM ALAT

A. Program Arduino :

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <PZEM004Tv30.h>
#include <SoftwareSerial.h>
#include <Servo.h>
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
#define PZEM_RX_PIN 9
#define PZEM_TX_PIN 8
#define LED_PIN_12 12
#define LED_PIN_13 13
#define SERVO_PIN1 2
#define SERVO_PIN2 3
```

```
int pos = 0;
```

```
SoftwareSerial pzemSWSerial(PZEM_RX_PIN, PZEM_TX_PIN);
PZEM004Tv30 pzem(pzemSWSerial);
Servo myServo1;
Servo myServo2;
```

```
float voltage;
float current;
float power;
float frequency;
float pf;
int prevTime = millis();
```

```
void setup() {
  Serial.begin(9600);
  pzemSWSerial.begin(9600);
  // Serial.print("1");
  lcd.begin();
```

```
  // Turn on the backlight and print a message.
  lcd.backlight();
```

```
  // Serial.print("2");
  pinMode(LED_PIN_12, OUTPUT);
  pinMode(LED_PIN_13, OUTPUT);
```

```
  digitalWrite(LED_PIN_12, HIGH);
```

```

digitalWrite(LED_PIN_13, HIGH);

myServo1.attach(SERVO_PIN1);
myServo2.attach(SERVO_PIN2);

myServo1.write(160);
myServo2.write(0);
}

void loop() {
  if (Serial.available() > 0) {
    String command = Serial.readStringUntil('\n');
    command.trim(); // Remove any leading/trailing whitespace

    // Process the command
    if (command == "baca") {
      Serial.print("~");
      Serial.print(voltage);
      Serial.print(":");
      Serial.print(current);
      Serial.print(":");
      Serial.print(power);
      Serial.print(":");
      Serial.print(frequency);
      Serial.print(":");
      Serial.print(pf);
      Serial.println("!");
    } else if (command == "on1") {
      digitalWrite(LED_PIN_12, LOW);
    } else if (command == "off1") {
      digitalWrite(LED_PIN_12, HIGH);
    } else if (command == "on2") {
      digitalWrite(LED_PIN_13, LOW);
    } else if (command == "off2") {
      digitalWrite(LED_PIN_13, HIGH);
    } else if (command == "gerak") {
      moveServo(2);
    }
  } else {
    readSensorData(10);
  }
}

```

B. Program Python:

```
import sys
import json
import os
import logging
from PyQt5.QtWidgets import (QApplication, QWidget, QVBoxLayout,
                              QHBoxLayout, QGridLayout, QLabel, QLineEdit, QPushButton,
                              QMessageBox, QCheckBox, QToolTip,
                              QGraphicsDropShadowEffect, QComboBox, QStatusBar)
from PyQt5.QtGui import QPalette, QColor, QMovie, QIcon, QFont
from PyQt5.QtCore import QSize, Qt, QTimer, QEasingCurve,
QPropertyAnimation
from subprocess import Popen

logging.basicConfig(level=logging.DEBUG, filename="debug.log",
filemode="w",
                    format="%(asctime)s - %(levelname)s - %(message)s")

def resource_path(relative_path):
    """ Get absolute path to resource, works for dev and for PyInstaller """
    try:
        base_path = sys._MEIPASS
    except Exception:
        base_path = os.path.abspath(".")
    return os.path.join(base_path, relative_path)

class LoginWindow(QWidget):
    def __init__(self):
        super().__init__()
        logging.debug("Initializing LoginWindow")
        self.initUI()

    def initUI(self):
        logging.debug("Setting up UI")
        self.setWindowTitle('Login')
        self.setWindowIcon(QIcon(resource_path("icon.png")))

        # Set 3/4 screen size
        screen = QApplication.primaryScreen()
        screen_size = screen.size()
        self.setFixedSize(screen_size.width() * 3 // 4, screen_size.height() * 3 // 4)
        self.move((screen_size.width() - self.width()) // 2, (screen_size.height() -
self.height()) // 2)

        # Create main layout
        main_layout = QGridLayout()
```

```

# Create label for the background GIF
self.background_label = QLabel(self)
self.background_label.setGeometry(0, 0, self.width(), self.height())
self.background_label.setScaledContents(True)
self.movie = QMovie(resource_path("ironman1.gif")) # Use resource_path to
access the gif
self.background_label.setMovie(self.movie)
self.movie.start()

# Create layout for form elements
form_layout = QVBoxLayout()
form_layout.setContentsMargins(50, 50, 50, 50)
form_layout.setSpacing(20)

# Create title label
self.title_label = QLabel("Welcome to Robot Assistant")
self.title_label.setAlignment(Qt.AlignCenter)
self.title_label.setStyleSheet("font-size: 32px; color: #FFD700; font-weight:
bold;")
form_layout.addWidget(self.title_label)

# Create language selector
self.language_selector = QComboBox()
self.language_selector.addItem("English")
self.language_selector.addItem("Indonesian")
self.language_selector.setStyleSheet("color: #FFD700; background-color:
#000000; font-size: 16px;")
self.language_selector.currentIndexChanged.connect(self.change_language)
form_layout.addWidget(self.language_selector)

# Create username label and text field
self.username_label = QLabel("Username:")
self.username_label.setStyleSheet("color: #FFD700; font-size: 20px;")
self.username_input = QLineEdit()
self.username_input.setPlaceholderText("Enter your username")
self.username_input.setToolTip("Your registered username")
self.username_input.setStyleSheet("background-color: #333333; color:
#FFD700; border: 1px solid #FFD700; padding: 10px; font-size: 20px;")
form_layout.addWidget(self.username_label)
form_layout.addWidget(self.username_input)

# Create password label and text field
self.password_label = QLabel("Password:")
self.password_label.setStyleSheet("color: #FFD700; font-size: 20px;")
self.password_input = QLineEdit()
self.password_input.setEchoMode(QLineEdit.Password)
self.password_input.setPlaceholderText("Enter your password")

```

```

self.password_input.setToolTip("Your account password")
self.password_input.setStyleSheet("background-color: #333333; color:
#FFD700; border: 1px solid #FFD700; padding: 10px; font-size: 20px;")
form_layout.addWidget(self.password_label)
form_layout.addWidget(self.password_input)

# Add password visibility toggle
self.show_password_button = QPushButton('Show')
self.show_password_button.setCheckable(True)
self.show_password_button.setStyleSheet("""
QPushButton {
    background-color: #333333;
    color: #FFD700;
    border: 1px solid #FFD700;
    padding: 5px;
    font-size: 16px;
}
QPushButton:checked {
    background-color: #FFD700;
    color: #333333;
}
""")
self.show_password_button.clicked.connect(self.toggle_password_visibility)
form_layout.addWidget(self.show_password_button)

# Create remember me checkbox
self.remember_me = QCheckBox('Remember Me')
self.remember_me.setStyleSheet("font-size: 18px; color: #FFD700;")
form_layout.addWidget(self.remember_me)

# Create login button
self.login_button = QPushButton('Login')
self.login_button.setStyleSheet("""
QPushButton {
    background-color: #FFD700;
    color: #333333;
    border: 1px solid #FFD700;
    padding: 10px;
    font-size: 20px;
    font-weight: bold;
}
QPushButton:hover {
    background-color: #DAA520;
}
QPushButton:pressed {
    background-color: #B8860B;
}
""")

```

```

    }
    """
    self.login_button.clicked.connect(self.check_login)
    form_layout.addWidget(self.login_button)

    # Add drop shadow effect to the login button
    shadow_effect = QGraphicsDropShadowEffect()
    shadow_effect.setBlurRadius(15)
    shadow_effect.setOffset(0, 0)
    shadow_effect.setColor(QColor(255, 215, 0, 150))
    self.login_button.setGraphicsEffect(shadow_effect)

    # Add status bar
    self.status_bar = QStatusBar()
    self.status_bar.setStyleSheet("color: #FFD700;")
    form_layout.addWidget(self.status_bar)

    # Add form layout to main layout
    main_layout.addLayout(form_layout, 1, 1)

    # Create labels for lightning GIFs
    self.lightning_label_left = QLabel(self)
    self.lightning_label_left.setGeometry(0, 0, 200, screen_size.height())
    self.lightning_label_left.setScaledContents(True)
    self.lightning_movie_left = QMovie(resource_path("lightning.gif")) # Use
resource_path to access the gif
    self.lightning_label_left.setMovie(self.lightning_movie_left)
    self.lightning_movie_left.start()

    self.lightning_label_right = QLabel(self)
    self.lightning_label_right.setGeometry(screen_size.width() - 200, 0, 200,
screen_size.height())
    self.lightning_label_right.setScaledContents(True)
    self.lightning_movie_right = QMovie(resource_path("lightning.gif")) # Use
resource_path to access the gif
    self.lightning_label_right.setMovie(self.lightning_movie_right)
    self.lightning_movie_right.start()

    # Add lightning labels to the main layout
    main_layout.addWidget(self.lightning_label_left, 0, 0, 3, 1)
    main_layout.addWidget(self.lightning_label_right, 0, 2, 3, 1)

    # Set main layout
    self.setLayout(main_layout)

    # Set dark theme

```

```

self.set_dark_theme()

# Adjust size of elements
self.adjust_sizes()

# Initialize tooltips
QToolTip.setFont(QFont('SansSerif', 12))
self.username_input.setToolTip('Enter your registered username.')
self.password_input.setToolTip('Enter your account password.')

def set_dark_theme(self):
    palette = QPalette()

    # Set the background color
    palette.setColor(QPalette.Window, QColor(0, 0, 0))
    palette.setColor(QPalette.WindowText, Qt.white)

    # Set the base color
    palette.setColor(QPalette.Base, QColor(33, 33, 33))
    palette.setColor(QPalette.AlternateBase, QColor(0, 0, 0))

    # Set the text color
    palette.setColor(QPalette.ToolTipBase, Qt.white)
    palette.setColor(QPalette.ToolTipText, Qt.white)
    palette.setColor(QPalette.Text, Qt.white)

    # Set the button color
    palette.setColor(QPalette.Button, QColor(33, 33, 33))
    palette.setColor(QPalette.ButtonText, Qt.white)
    palette.setColor(QPalette.BrightText, Qt.red)

    # Set the link color
    palette.setColor(QPalette.Link, QColor(255, 215, 0))

    # Set the highlight color
    palette.setColor(QPalette.Highlight, QColor(255, 215, 0))
    palette.setColor(QPalette.HighlightedText, Qt.black)

    QApplication.instance().setPalette(palette)
def adjust_sizes(self):
    self.username_input.setMinimumHeight(40)
    self.password_input.setMinimumHeight(40)
    self.login_button.setMinimumHeight(40)
    self.username_label.setMinimumHeight(30)
    self.password_label.setMinimumHeight(30)
    self.register_label.setMinimumHeight(30)

```

```

def check_login(self):
    username = self.username_input.text()
    password = self.password_input.text()

    try:
        with open('user.json', 'r') as f:
            users = json.load(f)
    except FileNotFoundError:
        self.show_error_message('User database not found.')
        return

    for user in users:
        if user['username'] == username and user['password'] == password:
            self.open_main_window()
            return

    self.show_error_message('Incorrect username or password.')

def adjust_sizes(self):
    self.username_input.setMinimumHeight(50)
    self.password_input.setMinimumHeight(50)
    self.login_button.setMinimumHeight(50)
    self.username_label.setMinimumHeight(40)
    self.password_label.setMinimumHeight(40)

def toggle_password_visibility(self):
    if self.show_password_button.isChecked():
        self.password_input.setEchoMode(QLineEdit.Normal)
        self.show_password_button.setText('Hide')
    else:
        self.password_input.setEchoMode(QLineEdit.Password)
        self.show_password_button.setText('Show')

def check_login(self):
    username = self.username_input.text()
    password = self.password_input.text()
    logging.debug(f"Attempting login with username: {username}")

    # Show loading spinner
    self.login_button.setText('Logging in...')
    self.login_button.setDisabled(True)

    # Show status message
    self.status_bar.showMessage('Logging in...')

```

```

# Add animation
self.animation = QPropertyAnimation(self.login_button, b"geometry")
self.animation.setDuration(500)
self.animation.setStartValue(self.login_button.geometry())
self.animation.setEndValue(self.login_button.geometry().adjusted(0, 10, 0, -
10))
self.animation.setEasingCurve(QEasingCurve.InOutQuad)
self.animation.start()

# Simulate login delay
QTimer.singleShot(2000, lambda: self.verify_login(username, password))

def verify_login(self, username, password):
    try:
        user_json_path = resource_path('user.json')
        logging.debug(f"Loading user data from: {user_json_path}")
        with open(user_json_path, 'r') as f:
            users = json.load(f)
    except FileNotFoundError:
        logging.error('User database not found.')
        self.show_error_message('User database not found.')
        self.login_button.setText('Login')
        self.login_button.setDisabled(False)
        self.status_bar.showMessage('User database not found.', 5000)
        return

    for user in users:
        if user['username'] == username and user['password'] == password:
            logging.debug(f"Login successful for user: {username}")
            self.open_main_window()
            return

    logging.warning('Incorrect username or password.')
    self.show_error_message('Incorrect username atau password.')
    self.login_button.setText('Login')
    self.login_button.setDisabled(False)
    self.status_bar.showMessage('Login failed.', 5000)

def show_error_message(self, message):
    logging.debug(f"Showing error message: {message}")
    msg_box = QMessageBox()
    msg_box.setWindowTitle('Error')
    msg_box.setText(message)
    msg_box.setIcon(QMessageBox.Warning)
    msg_box.setStyleSheet("""
        QMessageBox {

```



```

        background-color: #1b1b1b;
        color: white;
    }
    QMessageBox QLabel {
        color: white;
    }
    QMessageBox QPushButton {
        background-color: #2c2c2c;
        color: white;
        border: 1px solid #3a3a3a;
        padding: 5px;
    }
    """
msg_box.exec()

def open_main_window(self):
    process = Popen(['python', 'gui.py'])
    self.close()
    process.wait()

def change_language(self, index):
    logging.debug(f'Changing language to index: {index}')
    if index == 0: # English
        self.title_label.setText('Welcome to Robot Assistant')
        self.username_label.setText('Username:')
        self.password_label.setText('Password:')
        self.username_input.setPlaceholderText('Enter your username')
        self.password_input.setPlaceholderText('Enter your password')
        self.show_password_button.setText('Show')
        self.remember_me.setText('Remember Me')
        self.login_button.setText('Login')
        self.username_input.setToolTip('Enter your registered username.')
        self.password_input.setToolTip('Enter your account password.')
    elif index == 1: # Indonesian
        self.title_label.setText('Selamat Datang di Robot Assistant')
        self.username_label.setText('Nama Pengguna:')
        self.password_label.setText('Kata Sandi:')
        self.username_input.setPlaceholderText('Masukkan nama pengguna Anda')
        self.password_input.setPlaceholderText('Masukkan kata sandi Anda')
        self.show_password_button.setText('Tampilkan')
        self.remember_me.setText('Ingat Saya')
        self.login_button.setText('Masuk')
        self.username_input.setToolTip('Masukkan nama pengguna yang terdaftar.')
        self.password_input.setToolTip('Masukkan kata sandi akun Anda.')

```

```
if __name__ == '__main__':  
    app = QApplication(sys.argv)  
    login = LoginWindow()  
    login.show()  
    sys.exit(app.exec_())
```



DAFTAR RIWAYAT HIDUP



Nama : Ahmad Akmal Fahmi
Nama Panggilan : Akmal
Tempat, Tanggal Lahir : Wonosobo, 28 Agustus 2002
Agama : Islam
Orang Tua : Yazid Widodo dan Musnifah
Saudara : Putri Widiastuti dan Azwida
Hisni Savira
Alamat : Dusun Rejosari Rt 04, Rw 01
Desa Tambi, Kecamatan
Kejajar, Kabupaten Wonosobo,
Jawa Tengah
Hobi : Olahraga

Latar Belakang Pendidikan :

- 2009 – 2015 : SD Negeri Kejajar
- 2015 – 2018 : SMP Al-Madina Wonosobo
- 2018 – 2021 : SMK Telkom Purwokerto
- 2021 – 2024 : Politeknik Penerbangan Surabaya

On the Job Training :

- (Mei 2023 – September 2023) UPBU Lede Kalumbang, Tambolaka, NTT
- (Oktober 2023 – Februari 2024) Bandara Depati Amir, Pangkalpinang, Bangka Belitung

Sertifikat Kompetensi :

- ACS
- Transmisi Distribusi
- *Airfield Lighting System*