

**PROTOTYPE SISTEM *SMART PARKING*
DENGAN FITUR RESERVASI BERBASIS TELEGRAM
DAN *RADIO FREQUENCY IDENTIFICATION*
DI POLITEKNIK PENERBANGAN SURABAYA**

PROYEK AKHIR



**PROGRAM STUDI D3 TEKNIK NAVIGASI UDARA
POLITEKNIK PENERBANGAN SURABAYA**

2026

**PROTOTYPE SISTEM *SMART PARKING*
DENGAN FITUR RESERVASI BERBASIS TELEGRAM
DAN *RADIO FREQUENCY IDENTIFICATION*
DI POLITEKNIK PENERBANGAN SURABAYA**

PROYEK AKHIR

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



Oleh:

DESWINDA ROSALIA PUTRI

NIT. 30223002

PROGRAM STUDI D3 TEKNIK NAVIGASI UDARA

POLITEKNIK PENERBANGAN SURABAYA

2026

HALAMAN PERSETUJUAN

PROTOTYPE SISTEM *SMART PARKING* DENGAN FITUR RESERVASI BERBASIS TELEGRAM DAN *RADIO FREQUENCY IDENTIFICATION* DI POLITEKNIK PENERBANGAN SURABAYA

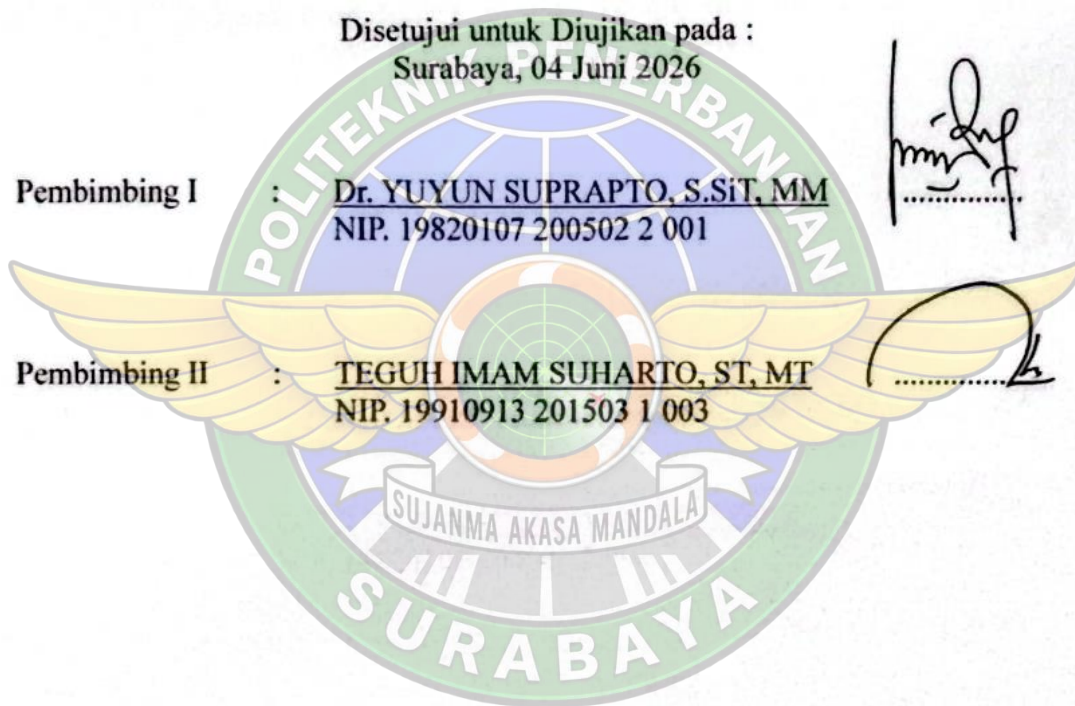
Oleh :
Deswinda Rosalia Putri
NIT. 30223002

Disetujui untuk Diujikan pada :
Surabaya, 04 Juni 2026

Pembimbing I : Dr. YUYUN SUPRAPTO, S.SiT, MM
NIP. 19820107 200502 2 001



Pembimbing II : TEGUH IMAM SUHARTO, ST, MT
NIP. 19910913 201503 1 003



LEMBAR PENGESAHAN

PROTOTYPE SISTEM *SMART PARKING*
DENGAN FITUR RESERVASI BERBASIS TELEGRAM
DAN *RADIO FREQUENCY IDENTIFICATION*
DI POLITEKNIK PENERBANGAN SURABAYA

Oleh :
Deswinda Rosalia Putri
NIT. 30223002

Telah dipertahankan dan dinyatakan lulus pada Ujian Tugas Akhir
Program Pendidikan Diploma 3 Teknik Navigasi Udara
Politeknik Penerbangan Surabaya
Pada tanggal : 04 Juni 2026

Panitia Penguji :

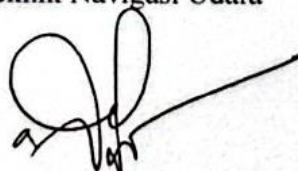
Ketua : Dr. BAMBANG BAGUS H., S.SiT, MM, MT
NIP. 19910915 200502 1 001

Sekretaris : NYARIS PAMBUDIYATNO, S.SiT, M.M.Tr
NIP. 19820525 200502 1 001

Anggota : Dr. YUYUN SUPRAPTO, S.SiT, MM
NIP. 19820107 200502 2 001



Mengetahui,
Ketua Program Studi
D3 Teknik Navigasi Udara



Dr. ADE IRFANSYAH, ST, MT
NIP. 19801125 200212 1 002

ABSTRAK

PROTOTYPE SISTEM *SMART PARKING* DENGAN FITUR RESERVASI BERBASIS TELEGRAM DAN *RADIO FREQUENCY IDENTIFICATION* DI POLITEKNIK PENERBANGAN SURABAYA

Oleh:

Deswinda Rosalia Putri

NIT.30223002

Dengan jumlah kendaraan yang semakin meningkat di lingkungan Kampus Politeknik Penerbangan Surabaya, kebutuhan akan sistem parkir yang terintegrasi juga semakin meningkat. Sistem parkir yang masih beroperasi secara konvensional mengharuskan pengguna melakukan pencarian tempat parkir secara manual dan kesulitan menemukan informasi tentang ketersediaan tempat parkir. Mereka juga tidak memiliki sistem reservasi dan identifikasi kendaraan yang terintegrasi. Tujuan dari penelitian ini adalah untuk merancang serta membuat prototype sistem *smart parking* yang menggunakan fitur reservasi Telegram dan *Radio Frequency Identification* (RFID). Analisis kebutuhan, perancangan sistem, pembuatan prototype, pengujian dan evaluasi, dan perbaikan sistem adalah bagian dari metode prototype yang digunakan. Hasil penelitian menunjukkan bahwa prototype yang dikembangkan dapat mengintegrasikan fungsi reservasi melalui bot Telegram, mengidentifikasi pengguna menggunakan RFID, mendeteksi ketersediaan tempat parkir menggunakan sensor infrared, dapat menggerakkan motor servo untuk mengendalikan *barrier gate* secara otomatis, dan memonitoring data melalui Firebase. Sensor infrared mendeteksi kondisi tempat parkir dengan akurasi 100%, sedangkan RFID, motor servo, Firebase, dan bot Telegram memiliki tingkat akurasi yang sama. Namun validasi RFID memerlukan waktu sekitar 4–6 detik, dan notifikasi Bot Telegram mengalami keterlambatan sekitar 3–9 detik karena pengaruh kualitas koneksi internet terhadap komunikasi data. Diharapkan prototype ini akan berfungsi sebagai alternatif untuk membantu pengelolaan parkir yang lebih terorganisir dan mendorong penerapan teknologi *Internet of Things* (IoT) di kampus Politeknik Penerbangan Surabaya.

Kata Kunci : *Smart parking*, *Internet of Things* (IoT), ESP32, Telegram, RFID, Firebase, sensor infrared.

ABSTRACT

PROTOTYPE SMART PARKING SYSTEM WITH TELEGRAM-BASED RESERVATION AND RADIO FREQUENCY IDENTIFICATION AT CIVIL AVIATION POLYTECHNIC OF SURABAYA

By:

Deswinda Rosalia Putri

NIT.30223002

With the increasing number of vehicles in the Civil Aviation Polytechnic of Surabaya Campus, the need for an integrated parking system is also increasing. Conventional parking systems require users to manually search for parking spaces and have difficulty finding information about parking availability. They also lack an integrated vehicle reservation and identification system. The purpose of this study is to design and build a prototype of a smart parking system that uses Telegram's reservation feature and Radio Frequency Identification (RFID). Requirements analysis, system design, prototype creation, testing and evaluation, and system improvement are part of the prototype method used. The results show that the developed prototype can integrate reservation functions through a Telegram bot, identify users using RFID, detect parking space availability using infrared sensors, can drive servo motors to automatically control barrier gates, and monitor data through Firebase. The infrared sensor detects parking space conditions with 100% accuracy, while RFID, servo motors, Firebase, and the Telegram bot have the same level of accuracy. However, RFID validation takes about 4–6 seconds, and Telegram Bot notifications experience a delay of about 3–9 seconds due to the impact of internet connection quality on data communication. It is hoped that this prototype will serve as an alternative to help manage parking more organized and encourage the implementation of Internet of Things (IoT) technology on the Civil Aviation Polytechnic of Surabaya Campus.

Keywords: *Smart parking, Internet of Things (IoT), ESP32, Telegram, RFID, Firebase, infrared sensor.*

KATA PENGANTAR

Puji syukur kami panjatkan kepada Allah SWT atas segala rahmat dan hidayah-Nya, yang telah memungkinkan saya untuk menyelesaikan penulisan Proyek Akhir dengan judul *"Prototype Sistem Smart Parking Dengan Fitur Reservasi Berbasis Telegram dan Radio Frequency Identification di Politeknik Penerbangan Surabaya"*. Dengan berkat-Nya, proyek akhir ini dapat diselesaikan tepat waktu dan sesuai dengan ketentuan yang berlaku.

Penyusunan Proyek Akhir ini merupakan salah satu syarat dalam menyelesaikan pendidikan di Politeknik Penerbangan Surabaya untuk memperoleh gelar Ahli Madya (A.Md). Saya ingin mengucapkan terima kasih yang tulus kepada semua pihak yang telah memberikan dukungan dan bantuan selama proses penyusunan Proyek Akhir ini, khususnya kepada :

1. Allah SWT, yang telah memberikan kesempatan kepada penulis untuk menjalani kehidupan di dunia dan mengizinkan penulis mendapatkan segala bentuk cinta, rahmat, ridha, dan kasih sayang-Nya.
2. Kedua Orang Tua yang telah memberikan doa serta dukungan dalam menyelesaikan Proyek Akhir.
3. Bapak Ahmad Bahrawi, SE., MT. selaku Direktur Politeknik Penerbangan Surabaya.
4. Bapak Ade Irfansyah, ST., MT. selaku Ketua Program Studi Teknik Navigasi Udara.
5. Ibu Dr. Yuyun Suprpto, S.SiT., MM. selaku Dosen Pembimbing I yang senantiasa membantu penulis dalam menyelesaikan Proyek Akhir
6. Bapak Teguh Imam Suharto, ST. MT selaku Dosen Pembimbing II yang senantiasa membantu penulis dalam menyelesaikan Proyek Akhir
7. Para dosen serta instruktur pada program studi Teknik Navigasi Udara Angkatan XVI yang telah memberikan ilmu pengetahuan.
8. Rekan Bantuan atas segala dukungan yang telah diberikan, rekan TNU XVI, serta para senior yang telah memberikan motivasi.

Penulis menyadari bahwa Proyek Akhir ini masih memiliki banyak kekurangan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang konstruktif untuk membantu memperbaiki kekurangan-kekurangan tersebut.

Surabaya, 04 Juni 2026



Deswinda Rosalia Putri
NIT. 30223002

DAFTAR ISI

Halaman

HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
ABSTRAK	v
<i>ABSTRACT</i>	vi
PERNYATAAN KEASLIAN DAN HAK CIPTA	vii
KATA PENGANTAR.....	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.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan.....	4
BAB 2 LANDASAN TEORI.....	6
2.1 Landasan Teori	6
2.1.1 Teori Metode Prototype	6
2.1.2 <i>Smart Parking System</i>	7
2.1.2 <i>Internet of Things (IoT)</i>	8
2.1.3 Mikrokontroler ESP32	9
2.1.4 Sensor Infrared	10
2.1.10 PCB	11
2.1.6 Bot Telegram.....	12
2.1.7 RFID (<i>Radio Frequency Identification</i>).....	13
2.1.8 Motor Servo	14
2.1.9 Power Supply.....	15
2.1.11 Pin Header	16
2.1.12 Kabel Jumper.....	17
2.1.13 Kabel Power AC	18
2.1.14 <i>Firebase</i>	19
2.1.15 Arduino IDE	21
2.2 Kajian Penelitian Terdahulu yang Relevan.....	22
BAB 3 METODE PENELITIAN.....	28
3.1 Metode Penelitian.....	28

3.1.1 Identifikasi Kebutuhan (<i>Analysis</i>).....	29
3.1.2 Desain Awal (Perancangan).....	30
3.1.3 Pembuatan Prototype (<i>Development</i>)	37
3.1.4 Pengujian dan Evaluasi (<i>Testing & Evaluation</i>)	38
3.1.5 Perbaikan Sistem (<i>Refinement</i>).....	47
3.2 Komponen Alat.....	48
3.2.1 Perangkat Keras (<i>Hardware</i>)	48
3.2.2 Perangkat Lunak (<i>Software</i>).....	50
3.3 Teknik Analisis Data.....	50
3.4 Waktu dan Tempat Penelitian	52
BAB 4	54
HASIL DAN PEMBAHASAN	54
4.1 Hasil Penelitian.....	54
4.1.1 Analisis Kebutuhan (<i>Analysis</i>).....	54
4.1.2 Desain Awal (Perancangan)	56
4.1.3 Pembuatan Prototype (<i>Development</i>)	112
4.1.4 Pengujian dan Evaluasi (<i>Testing & Evaluation</i>)	114
4.1.5 Perbaikan Sistem (<i>Refinement</i>)	138
BAB 5	141
PENUTUP	141
5.1 Kesimpulan.....	141
5.2 Saran	141
DAFTAR PUSTAKA	143
LAMPIRAN A	146
STANDARD OPERATIONAL PROCEDURE (SOP)	146
PENGUNAAN ALAT SMART PARKING	146
LAMPIRAN B	162
CODING ARDUINO IDE	162
LAMPIRAN C	197
PAMFLET / BROSUS SMART PARKING	197
LAMPIRAN D.....	198
HASIL PENGUJIAN	198
DAFTAR RIWAYAT HIDUP	201

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Flowchart Metode Prototype	7
Gambar 2.2 Mikrokontroller ESP32	10
Gambar 2.3 Sensor Infrared	11
Gambar 2.4 PCB	12
Gambar 2.5 Bot Telegram	13
Gambar 2.6 RFID.....	14
Gambar 2.7 Motor Servo.....	15
Gambar 2.8 Power Supply	16
Gambar 2.9 Pin Header.....	16
Gambar 2.10 Kabel Jumper	18
Gambar 2.11 Kabel Power AC	19
Gambar 2.12 Firebase	20
Gambar 2.13 Arduino IDE	21
Gambar 3.1 Metode Penelitian Prototype	29
Gambar 3.2 Wiring Diagram.....	31
Gambar 3.3 Blog Diagram	34
Gambar 3.4 Flowchart Kerja Sistem <i>Smart Parking</i>	35
Gambar 4.1 Wiring Diagram.....	57
Gambar 4.2 Desain Prototype <i>Smart Parking</i>	58
Gambar 4.3 Tampilan Arduino IDE	66
Gambar 4.4 Instalasi Library In & Out	68
Gambar 4.5 Instalasi <i>Board Manager</i> ESP32	68
Gambar 4.6 Koding Sensor Infrared	105
Gambar 4.7 Koding Jaringan Internet.....	105
Gambar 4.8 Layar Utama Firebase Console	106
Gambar 4.9 Membuat Nama Project.....	107

Gambar 4.10 Mengaktifkan Realtime Database	107
Gambar 4.11 URL Database & API Key	108
Gambar 4.12 Menambahkan ID Telegram Users	109
Gambar 4.13 BotFather	110
Gambar 4.14 Membuat Bot pada BotFather	110
Gambar 4.15 Chat ID Telegram	111
Gambar 4.16 Bot <i>Smart Parking</i>	112



DAFTAR TABEL

	Halaman
Tabel 2. 1 Kajian Terdahulu yang Relevan	22
Tabel 3. 1 Tabel Konfigurasi Pin.....	32
Tabel 3. 2 Teknik Pengujian Perangkat Keras.....	22
Tabel 3. 3 Pengujian Fungsi Sistem	22
Tabel 3. 4 Pengujian Integrasi Perangkat Lunak & Perangkat Keras	22
Tabel 3. 5 Pengujian Kestabilan dan Keandalan.....	22
Tabel 3. 6 Pengujian Antarmuka Pengguna (User Interface)	22
Tabel 3. 7 Pengujian Error Handling	22
Tabel 3. 8 Rancangan Hasil Pengujian.....	51
Tabel 3. 9 Jadwal Penelitian.....	22
Tabel 4. 1 Spesifikasi Komponen Alat.....	22
Tabel 4. 2 Pengujian Perangkat Keras (<i>hardware</i>).....	22
Tabel 4. 3 Pengujian Fungsi Sistem	22
Tabel 4. 4 Pengujian Integrasi.....	120
Tabel 4. 5 Pengujian Kestabilan dan Keandalan.....	122
Tabel 4. 6 Pengujian Antarmuka Pengguna	122
Tabel 4. 7 Pengujian Error Handling	122
Tabel 4. 8 Tabel Analisis Data.....	122
Tabel 4. 9 Evaluasi Kinerja Komponen	122

DAFTAR LAMPIRAN

	Halaman
Lampiran A <i>Standard Operational Procedure</i> (SOP)	A-1
Lampiran B Coding Arduino IDE	B-1
Lampiran C Pamflet / Brosur <i>Smart Parking</i>	22
Lampiran D Hasil Pengujian	22



DAFTAR PUSTAKA

- Ade Putra Ode Amane, S. S. M. S., Rusina Widha Febriana, S. K. M. K., Ir. Marina Artiyasa, S. T. M. T. I. P. M., M. A. O. C. S. E. M., Dr. Husain, S. T. M. P. M. T., Nabilah, M., Fachruzzaki, S. S. M. T., Ns. Aulia Asman, S. K. M. B. A., Achmad Ridwan, S. T. M. S., & Dr. Ir. Aji Suraji, S. T. M. S. (2023). *PEMANFAATAN DAN PENERAPAN INTERNET OF THINGS (IOT) DI BERBAGAI BIDANG*. PT. Sonpedia Publishing Indonesia. <https://books.google.co.id/books?id=8zWqEAAAQBAJ>
- Arissaputra, R., Sentika, S., Nusri, A. Z., Wardhani, W. N. R., Kusumawati, M., Sukmaningrum, D., & Siska, D. (2025). *BISNIS DIGITAL DI ERA IoT: STRATEGI DAN IMPLEMENTASI*. CV. Intelektual Manifes Media. <https://books.google.co.id/books?id=3XxyEQAAQBAJ>
- Assoc. Prof. Dr. Rahmayati, M. E. I. (2025). *Sistem Informasi Manajemen Dan Keamanan Di Lembaga Keuangan Syariah*. umsu press. <https://books.google.co.id/books?id=nMVVEQAAQBAJ>
- Azam, A. (2022). *Cara Cepat belajar IoT: ESP32: Pengenalan dan Instalasi Arduino IDE*. Radnet Digital Indonesia. <https://books.google.co.id/books?id=y2JkEAAAQBAJ>
- Budi, A. S., Pradini, R. S., Basuki, A., & Setiawan, E. (2026). *Internet of Things: Konsep dan Teknis Perangkat Keras, Protokol, dan Implementasi Sistem*. Universitas Brawijaya Press. <https://books.google.co.id/books?id=WAPDEQAAQBAJ>
- Budiman, K., Hutahaeen, K. T., Aziz, A. A., Suryosaputro, H., Nurfarida, L. W., Belipati, M. E., Rizqi, M., Purwinarko, A., & Ramadhan, H. D. (2023). *Bunga Rampai Pemanfaatan Sistem Smart Campus untuk Optimalisasi Pendidikan Tinggi Mandiri*. CV Eureka Media Aksara. <https://books.google.co.id/books?id=jDKPEQAAQBAJ>
- Daeng Rahmatullah, S. T. M. T., As'ad Shidqy Aziz, S. T. M. T., Reza Rahmadian S. ST., M. E. S., Widi Aribowo S. T., M. T., Abdullah Avif Putra Buana, S. P. M. T., & Belly Yan Dewantara, S. T. M. T. (2025). *Langkah Mudah Pemrograman Siemens PLC*. CV. Penerbit Ilmu Literasi dan Riset (Pilar). <https://books.google.co.id/books?id=nyqYEQAAQBAJ>
- Dandy Nugraha, Tohir, T., & Muhammad Ilman, S. (2025). Desain dan implementasi miniatur smart parking dengan sistem reservasi parkir berbasis Internet of Things. *JITEL (Jurnal Ilmiah Telekomunikasi, Elektronika, Dan Listrik Tenaga)*, 5(3), 167–178. <https://doi.org/10.35313/jitel.v5.i3.2025.167-178>
- Darmawan, A. (2025). *PENGANTAR IoT PRAKTIS*. Penerbit Andi. <https://books.google.co.id/books?id=ZvpZEQAAQBAJ>
- Husni, M., Figo, S., Fadhilah, A. F., & Tampubolon, A. P. Y. (2025). *Teknologi IoT*. Penerbit Andi. https://books.google.co.id/books?id=Q_xlEQAAQBAJ
- Inanc, Z. P. N. (2017). *Smart Parking Applications Using RFID Technology*. (October 2007), 2007–2010. <https://doi.org/10.1109/RFIDEURASIA.2007.4368108>
- Indra, A. &. (2025). *ALGORITMA SEMUT untuk Ekosistem Internet of Things*. PT Arr Rad Pratama. <https://books.google.co.id/books?id=BTlyEQAAQBAJ>

- Iswanto. (2016). *Belajar mikrokontroler at89s51 dengan bahasa basic*. Deepublish. <https://books.google.co.id/books?id=kDhmqAAQBAJ>
- Karim, A. (2025). Perancangan_Smart_Parking_Berbasis_IOT_Untuk_Pengelolaan_Parkir_24. *Jurnal Teknologi Informasi*, 11(1), 36–41. <https://doi.org/10.52643/jti.v11i1.6422>
- Kecheng, L., Quan, Y., Jian, W., Hua, C., Tohir, T., Xuehui, Z., Huajian, X., Yu, X., Yi, D., & Jing, G. (2026). *Aplikasi Motor Listrik dan Sistem Kendali Gerak*. LovRinz Publishing. <https://books.google.co.id/books?id=CCe4EQAAQBAJ>
- Kholisho, Y. N., & Amri, Z. (2025). *JITE (Journal of Informatics and Telecommunication Engineering) Village Administration*. 9(July), 281–291.
- Lubis, M. I. A., & Pahlefi, R. A. (2026). *RANCANG BANGUN ALAT PENYIRAM OTOMATIS PADA BIBIT TANAMAN KELENGKENG*. Uwais Inspirasi Indonesia. <https://books.google.co.id/books?id=tMi9EQAAQBAJ>
- Miftahul Huda, M. K. (2019). *Teknologi Komputer: Pemanfaatan teknologi komputer untuk mempermudah penyelesaian tugas dan pekerjaan yang dihadapi*. bisakimia. <https://books.google.co.id/books?id=2DmnDwAAQBAJ>
- Nugraha, J. A. P., Iwan, A. H., Gumuljo, E. F., Graciella, E., Mulyadinata, N., Budiman, K. O., & Nugroho, E. W. (2024). *Menguasai Arduino: Inspirasi Proyek-Proyek Arduino bagi Pemula*. SIEGA Publisher. <https://books.google.co.id/books?id=R5gFEQAAQBAJ>
- Purnomo, R. F., Purbo, O. W., & Aziz, R. Z. A. (2021). *Firestore: Membangun Aplikasi Berbasis Android*. Andi Offset. <https://books.google.co.id/books?id=iosaEAAAQBAJ>
- Putra, D. I., Aisuwarya, R., Ardopa, S., & Purnama, I. (2018). Sistem Cerdas Reservasi dan Pemantauan Parkir pada Lokasi Kampus Berbasis Konsep Internet of Things. *Jurnal Teknologi Dan Sistem Komputer*, 6(2), 57–63. <https://doi.org/10.14710/jtsiskom.6.2.2018.57-63>
- Ramsari, N. (2019). Smart Parking System Dengan Sistem Reservasi. *Journal of Informatics, Information System, Software Engineering and Applications (INISTA)*, 1(2), 9–19. <https://doi.org/10.20895/inista.v1i2.28>
- Rizal Mutaqin, S. K. M. S., Larian Ashor Yoziarde, S. T. M. E., & Dwi Shinta Dharmopadni, S. K. C. E. (2025). *Pemrograman Arduino*. PT KIMHSAFI ALUNG CIPTA. <https://books.google.co.id/books?id=eDNJEQAAQBAJ>
- Sukadana, I. W., Pande, I. M., & Yuda, D. (2021). *Prototyping PCB Menggunakan Computer-Aided Design*. 2(1), 37–43. <https://doi.org/10.38043/tiers.v2i2.3310>
- Thorat, P. S. S., Ashwini, M., Kelshikar, A., & Londhe, S. (2017). IoT Based Smart Parking System Using RFID. *International Journal of Computer Engineering In Research Trends*, 4(1), 9–12. <https://doi.org/10.22362/ijcert/2017/v4/i1/xxxx>
- Trigreisian, A. A., & Harani, N. H. (2023). *Telegram Bot Wawancara Kerja dengan Algoritma Long Short Term Memory*. Penerbit Buku Pedia. <https://books.google.co.id/books?id=IxuuEAAAQBAJ>
- Umam, A. S. R. F. (2022). *Project Sistem Kontrol Berbasis Arduino*. Media Nusa Creative (MNC Publishing). <https://books.google.co.id/books?id=ormeEAAAQBAJ>
- Widiasmadi, N. (2023). *Smart Agriculture 2*. Penerbit Adab. <https://books.google.co.id/books?id=5jL2EAAAQBAJ>

- Wijayanti, D., & Haryadi, E. (2020). *Sistem Informasi Penyewaan Rumah Kontrakan Berbasis Web Dengan Menggunakan Metode Prototype*. 15(April 2020), 18–25.
- Yudhanto, Y., & Azis, A. (2019). *Pengantar Teknologi Internet of Things (IoT)*. UNSPress. <https://books.google.co.id/books?id=lK33DwAAQBAJ>
- Yusniar Nur Syarif Sidiq, Rd. Nuraini Siti Fathonah, S.S., M.Hum. Noviana Riza, S.Si., M. T. (2020). *Metode Klasifikasi Menentukan Kenaikan Level UKM Bandung Timur Dengan Algoritma Naïve Bayes Pada Sistem JURAGAN Berbasis Komunitas*. CV. Kreatif Industri Nusantara. <https://books.google.co.id/books?id=BIv9DwAAQBAJ>



LAMPIRAN A
STANDARD OPERATIONAL PROCEDURE (SOP)
PENGUNAAN ALAT SMART PARKING

Tujuan :

Memberikan panduan langkah demi langkah dalam penggunaan sistem *smart parking* berbasis *Internet of Things* (IoT) guna memastikan proses parkir berjalan otomatis, akurat, serta memudahkan pengguna dalam melakukan reservasi, validasi RFID, dan pemantauan ketersediaan tempat parkir.

Prosedur Operasional :

1. Persiapan Alat

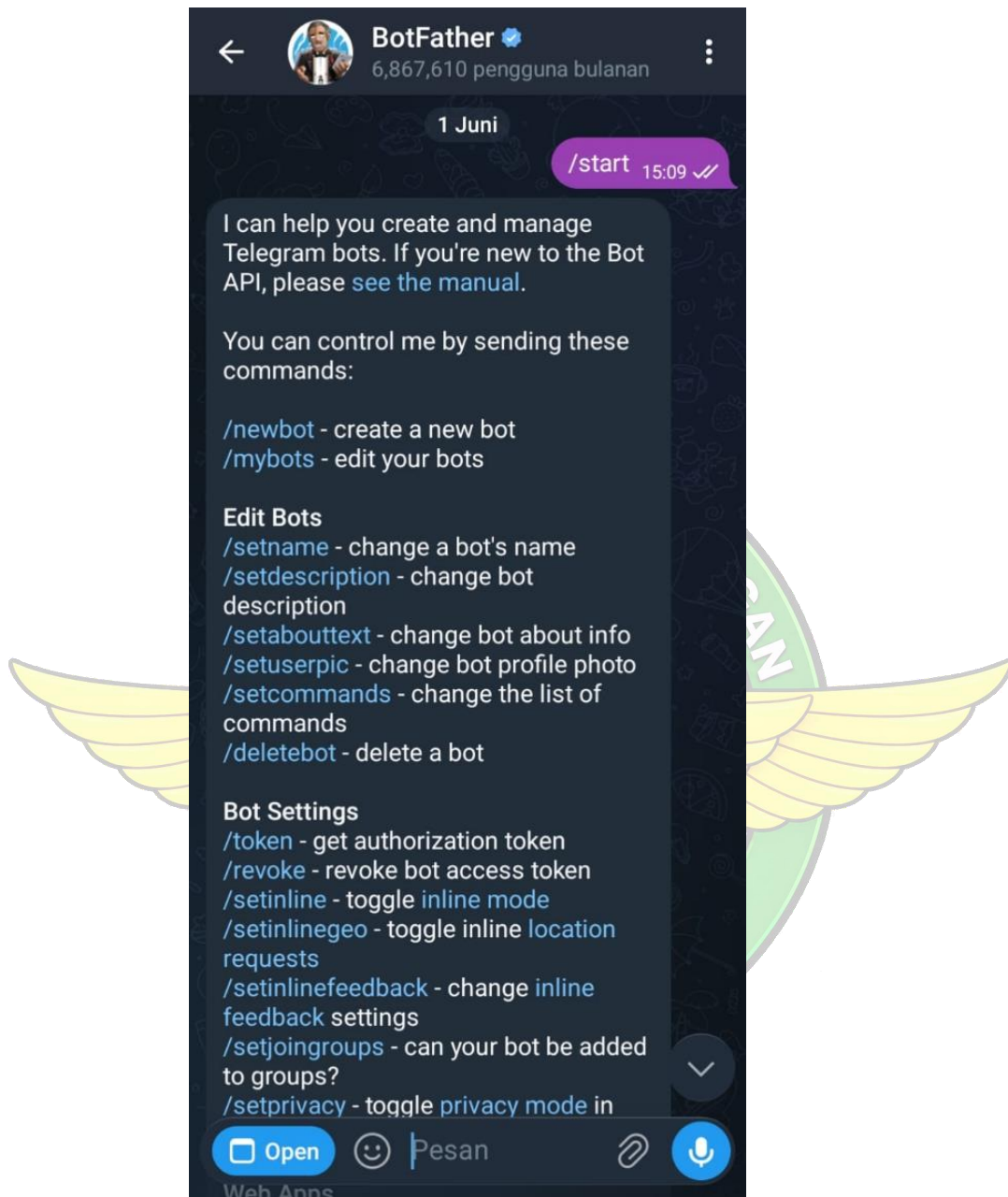
- a. Pastikan seluruh perangkat terhubung dengan sumber daya listrik.
- b. Pastikan koneksi internet/WiFi dalam kondisi stabil
- c. Siapkan Laptop atau PC untuk memprogram alat

➤ **TELEGRAM**

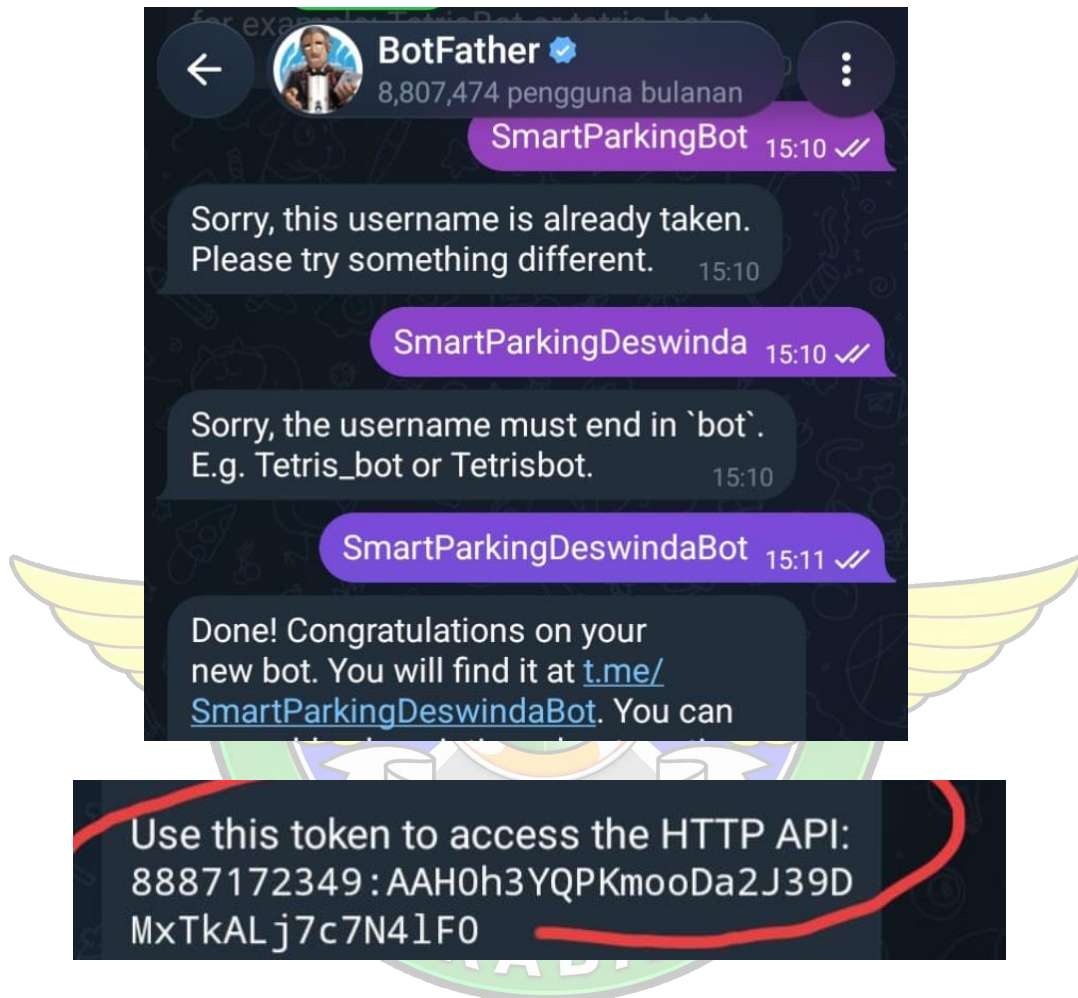
- 1) Instal Aplikasi Telegram
- 2) Login Telegram dengan nomer aktif
- 3) Klik tombol search dan ketik "BotFather"



4) Ketik "/start" pada bot

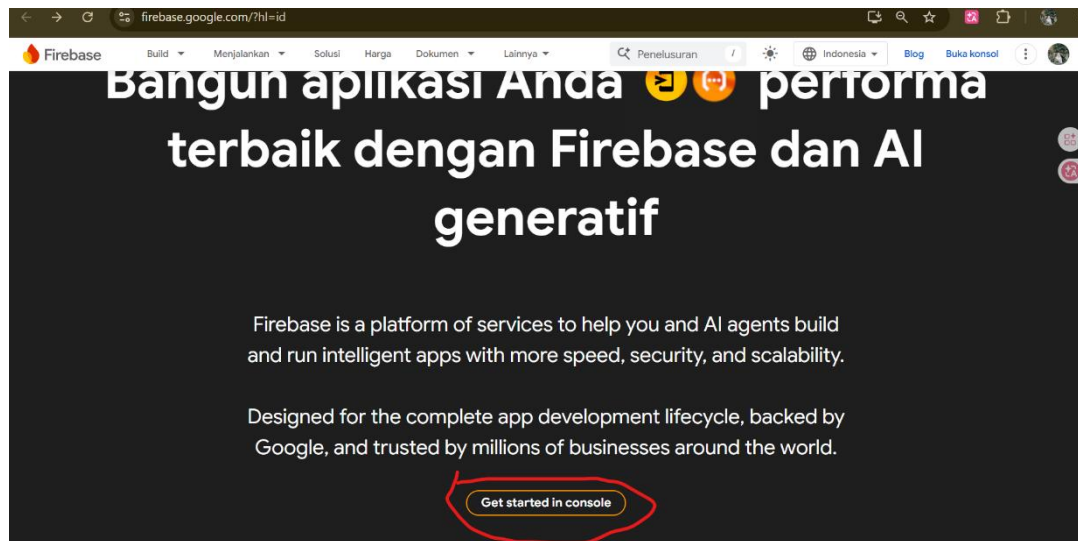


- 5) Klik pada kalimat /newbot dan buat nama bot yang diinginkan, jika sudah membuat nama bot, BotFather akan memberikan link bot telegram yang sudah dibuat dan salin token yang sudah dibuat oleh Botfather

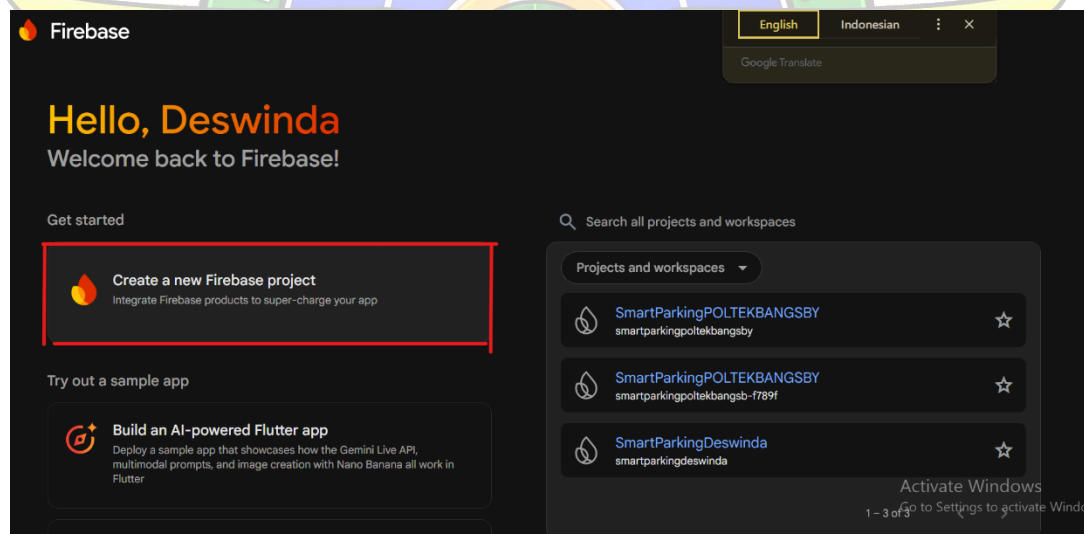


➤ Firebase

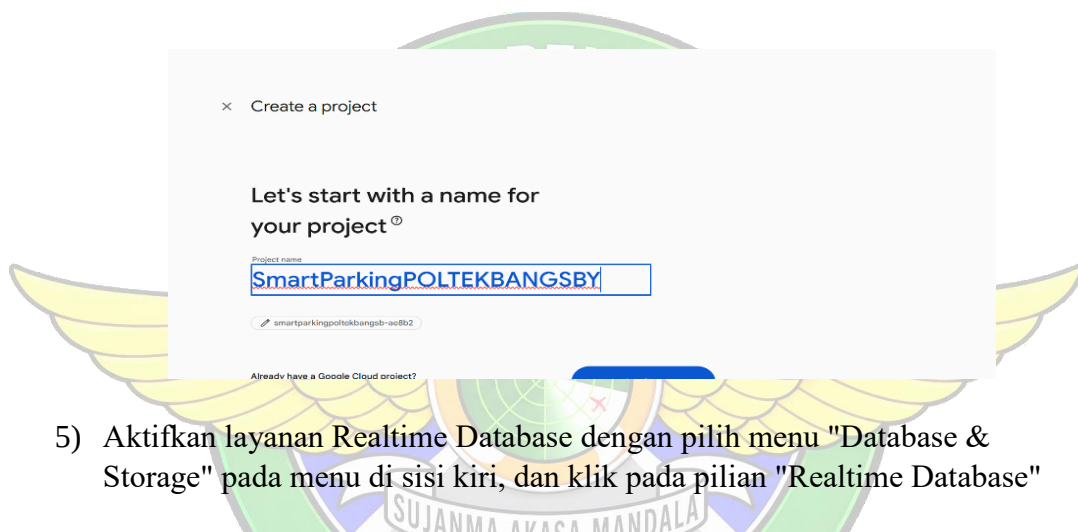
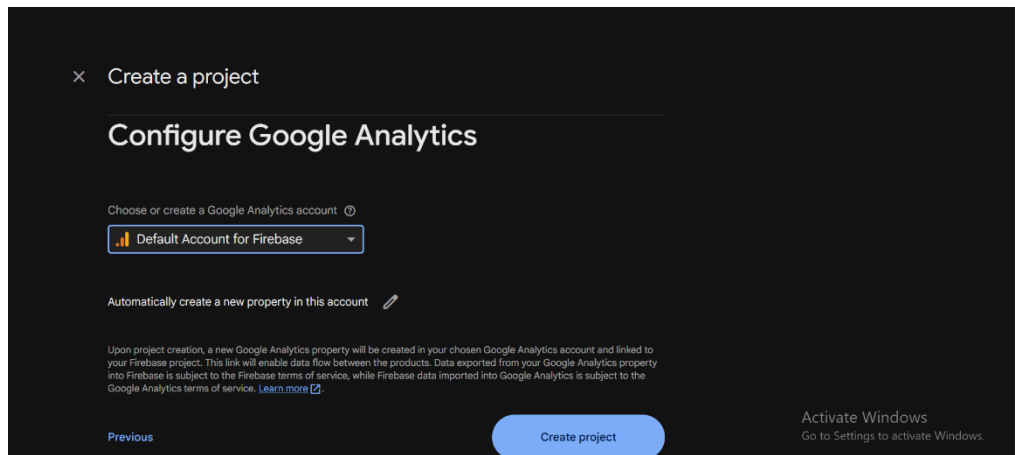
- 1) Akses situs resmi Firebase pada Google melalui alamat <https://firebase.google.com/?hl=id> dan login dengan akun google yang aktif
- 2) Klik "Get started in concole"



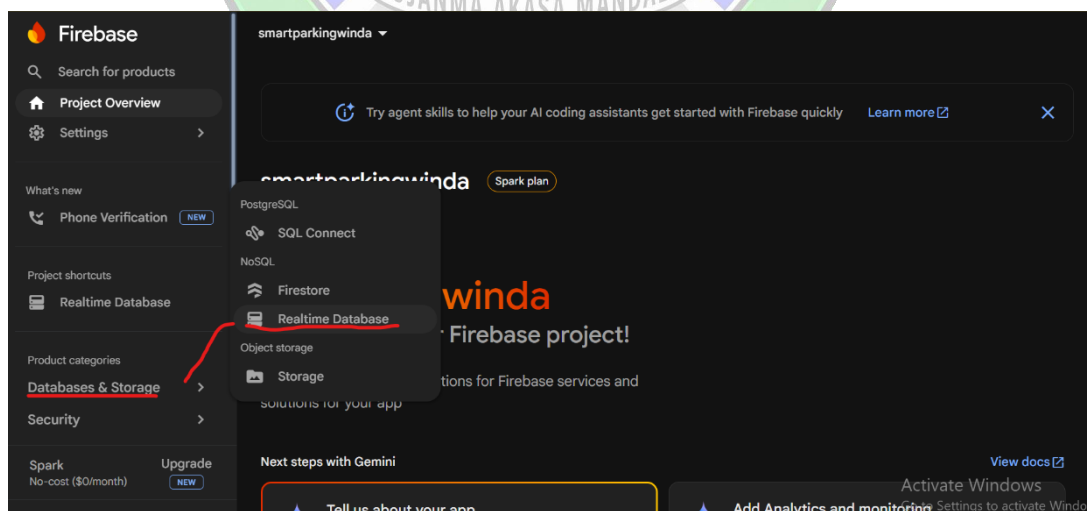
- 3) Buat proyek baru pada halaman utama Firebase Console dengan pilih menu "Create a new Firebase Project"



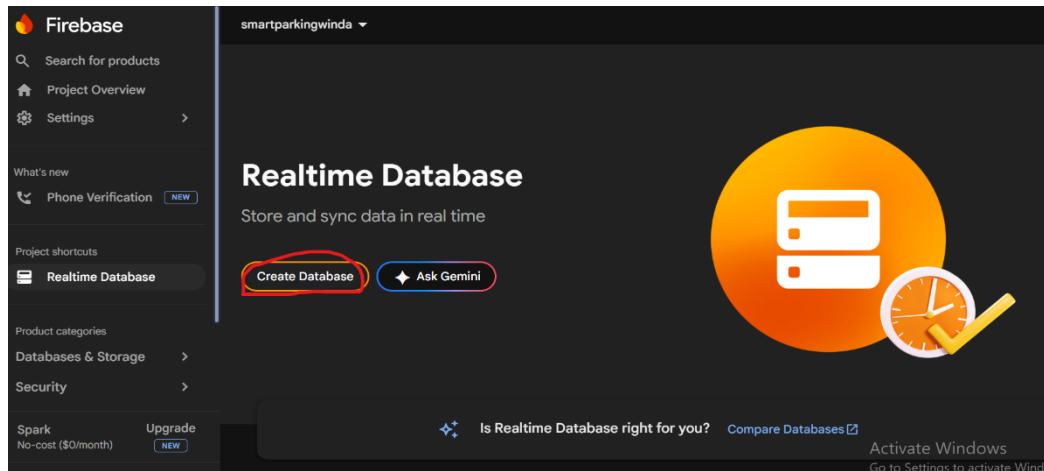
- 4) Jika sudah pilih menu tersebut, masukkan nama proyek *smart parking* dan selesaikan konfigurasi hingga proyek berhasil dibuat.



- 5) Aktifkan layanan Realtime Database dengan pilih menu "Database & Storage" pada menu di sisi kiri, dan klik pada pilihan "Realtime Database"

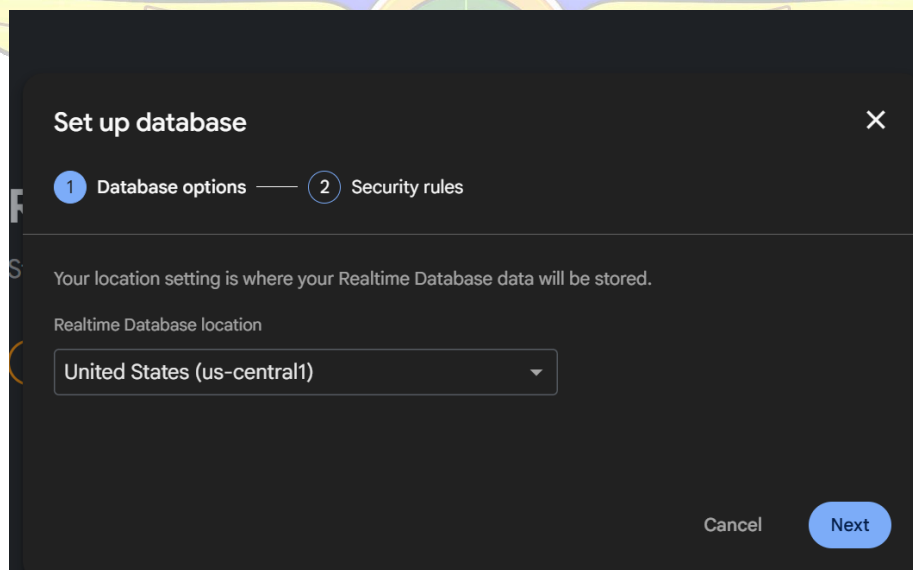


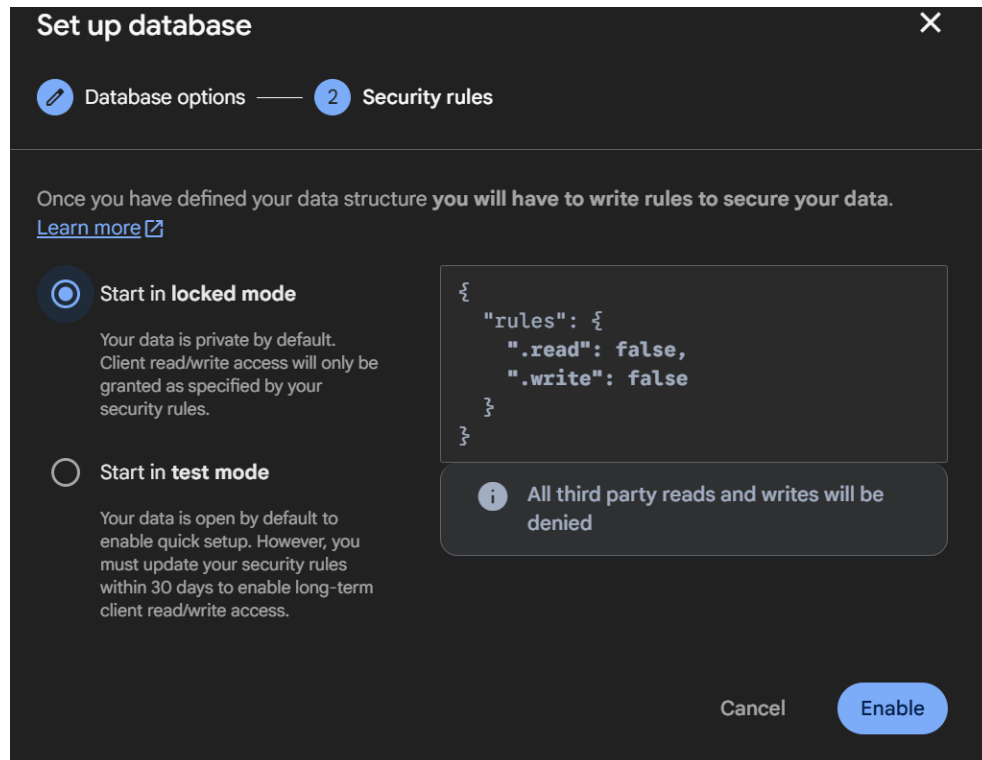
- 6) Jika sudah muncul tampilan layar pada menu Realtime Database lalu klik "Create Database"



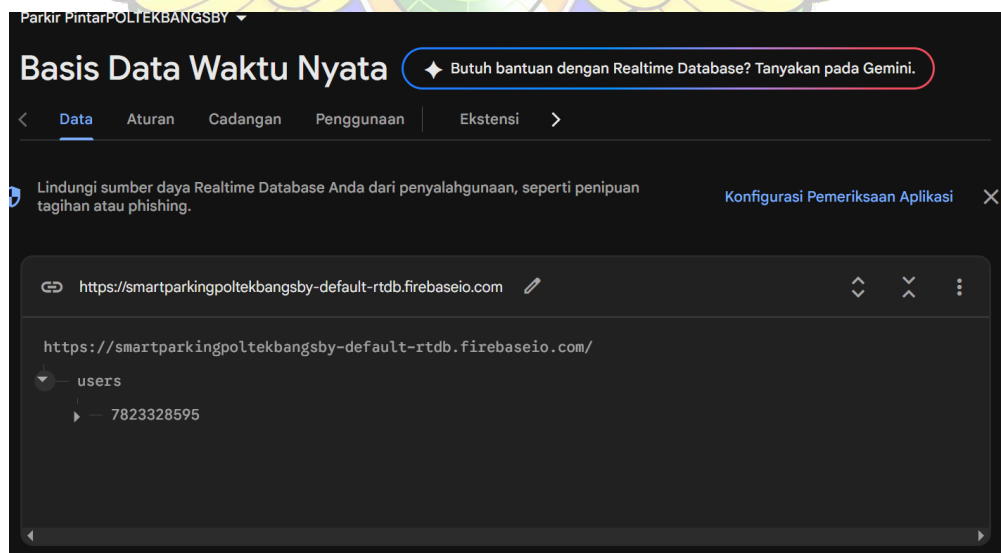
- 7) Atur Database dengan 2 langkah yaitu

- Database Options -> pilih location "United States" (us-central1)-> next
- Security rules -> klik "Start in locked mode"-> klik enable





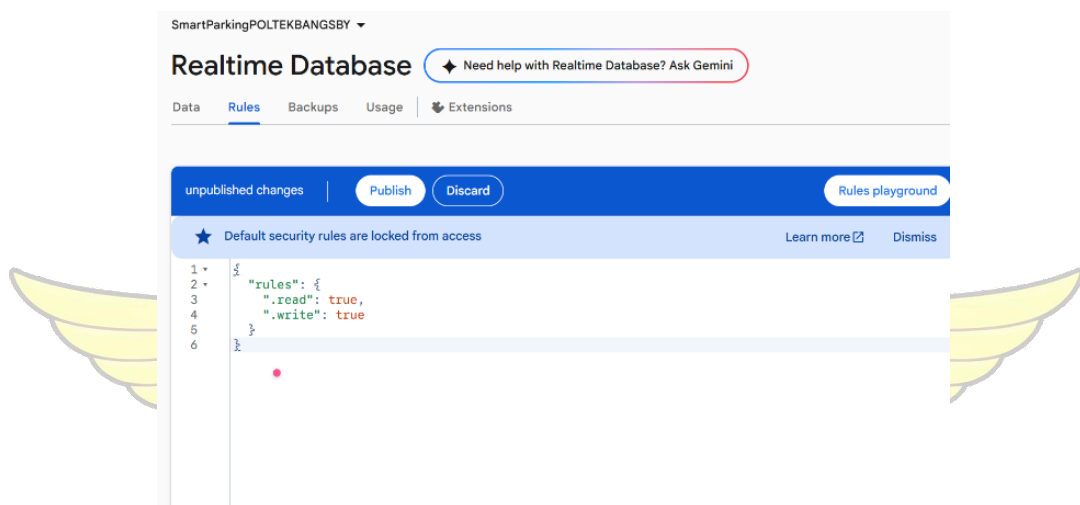
8) Jika sudah diatur, maka akan muncul tampilan layar pada Realtime Database seperti ini :



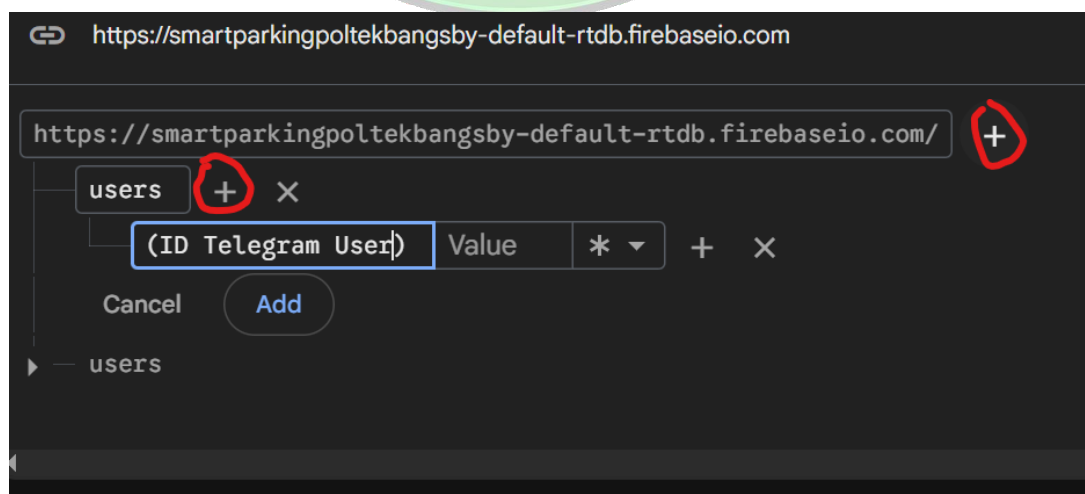
9) Pilih menu Aturan/Rules dan atur rules untuk pertukaran data antara ESP32 dan website monitoring

- Rules :

```
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

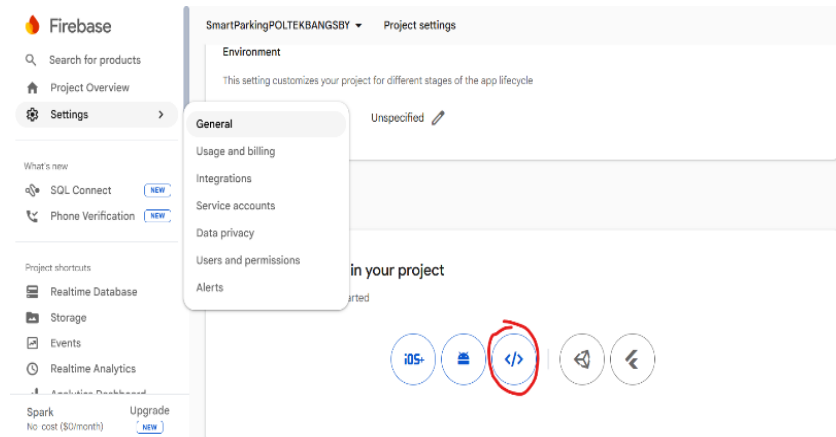


10) Kembangkan website monitoring yang terhubung dengan Firebase untuk menampilkan ID Telegram pengguna reservasi

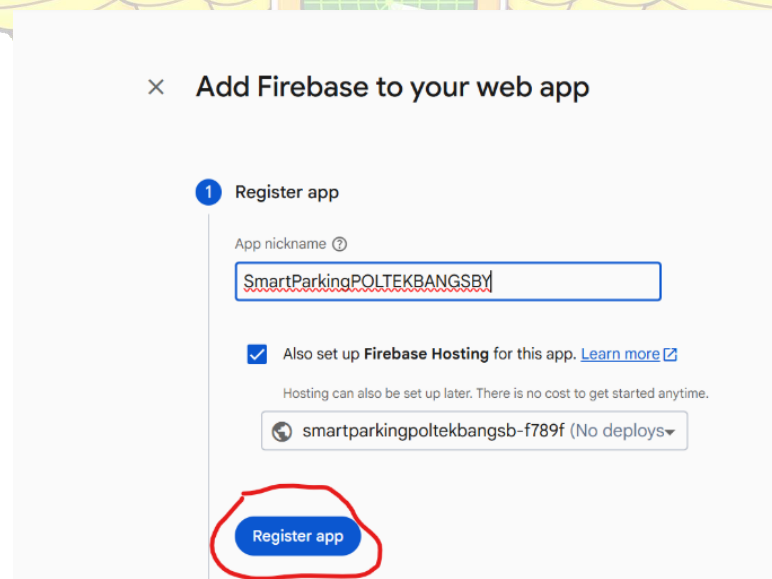


11) Ambil konfigurasi Firebase seperti Database URL, API Key, dan parameter lainnya untuk dimasukkan ke program Arduino IDE dengan cara :

- Pilih menu "Setting" pada pilihan menu di sisi kiri -> pilih "General"-> dan klik pada pilihan yang sudah dilingkari merah



- Masukkan nama project yang sudah dibuat pada langkah pertama -> klik Register App



- Jika sudah melakukan langkah sebelumnya, maka akan muncul tampilan konfigurasi untuk Database URL, API Key sebagai berikut

☒ Use npm ☐ Use a <script> tag

If you're already using [npm](#) and a module bundler such as [webpack](#) or [Rollup](#), you can run the following command to install the latest SDK ([Learn more](#)):

```
$ npm install firebase
```

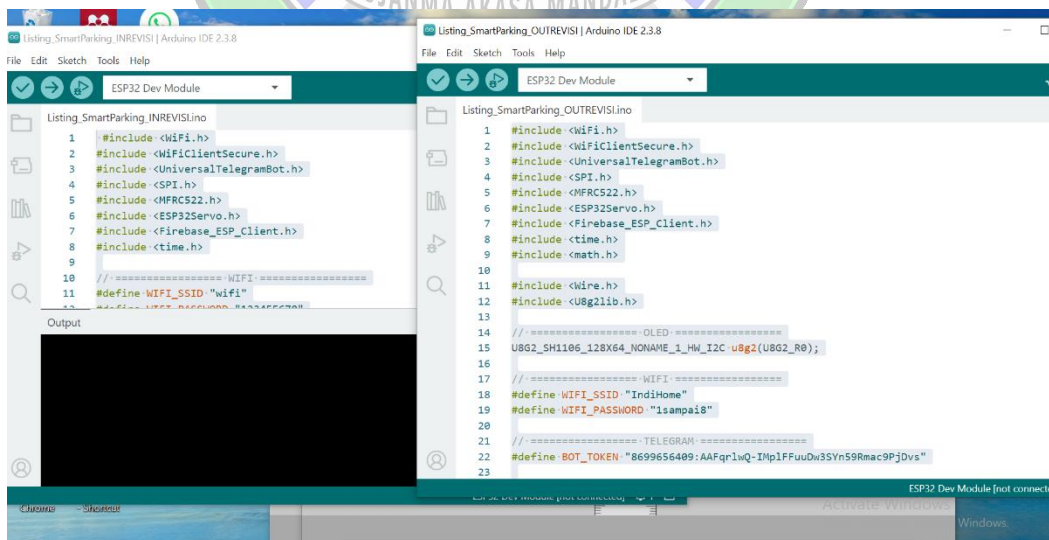
Then, initialize Firebase and begin using the SDKs for the products you'd like to use.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

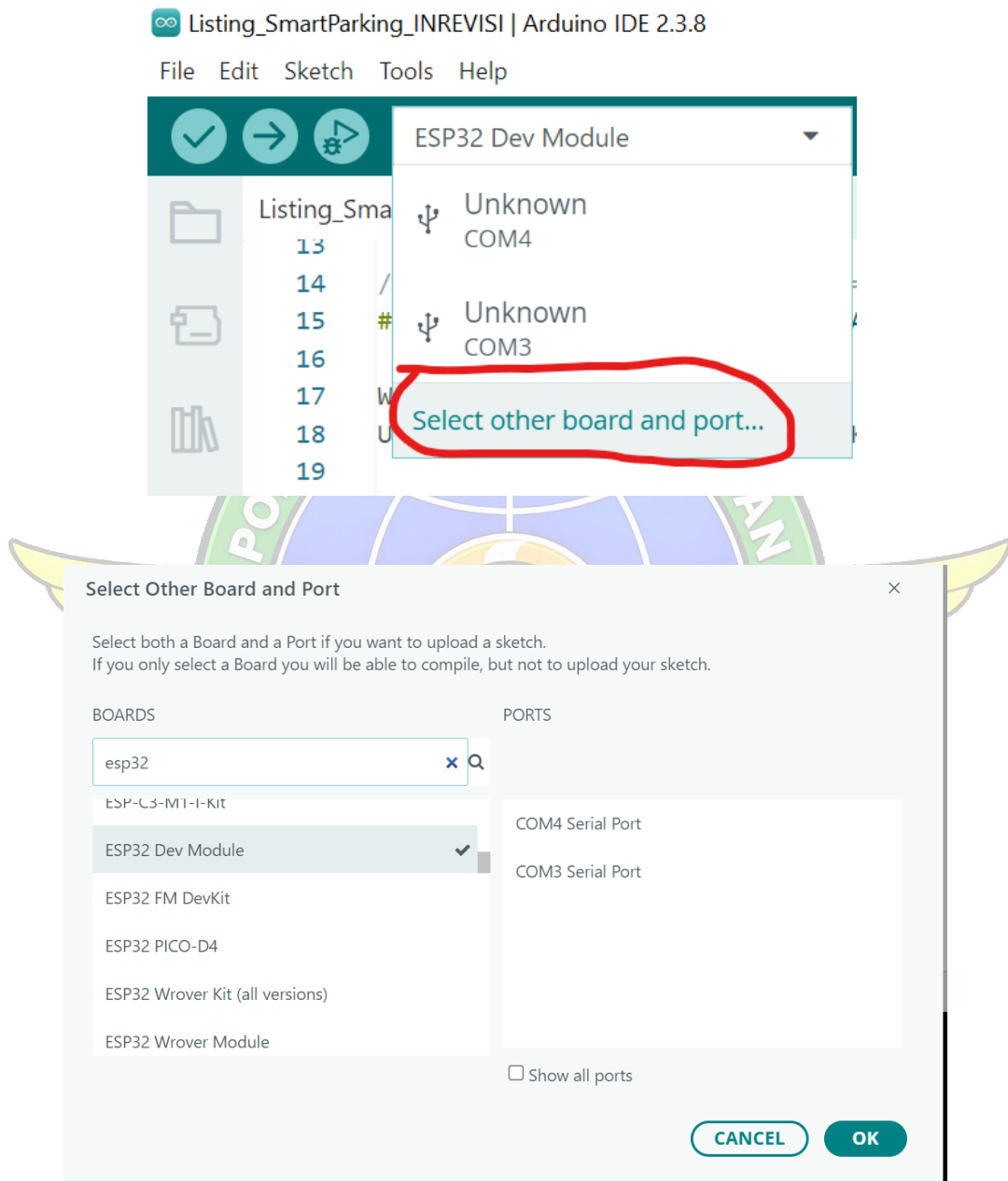
// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyB-Btq0L0n2Absrzggn0LRQwGAjnLRk9k",
  authDomain: "smartparkingpoltekbangs-b-f789f.firebaseio.com",
  databaseURL: "https://smartparkingpoltekbangs-b-f789f.firebaseio.com",
  projectId: "smartparkingpoltekbangs-b-f789f",
  storageBucket: "smartparkingpoltekbangs-b-f789f.firebaseio.com",
  messagingSenderId: "429350237175",
  appId: "1:429350237175:web:e8f7e7535a2d05210c74f2",
  measurementId: "G-6J2L9SEVT7"
};
```

➤ Arduino IDE

- 1) Instal aplikasi Arduino IDE di Google melalui alamat
- 2) Buka aplikasi Arduino IDE dengan 2 platform untuk program Arduino IDE (in & Out), karena ESP32 berjumlah 2 buah, maka untuk program Arduino IDE juga berjumlah 2



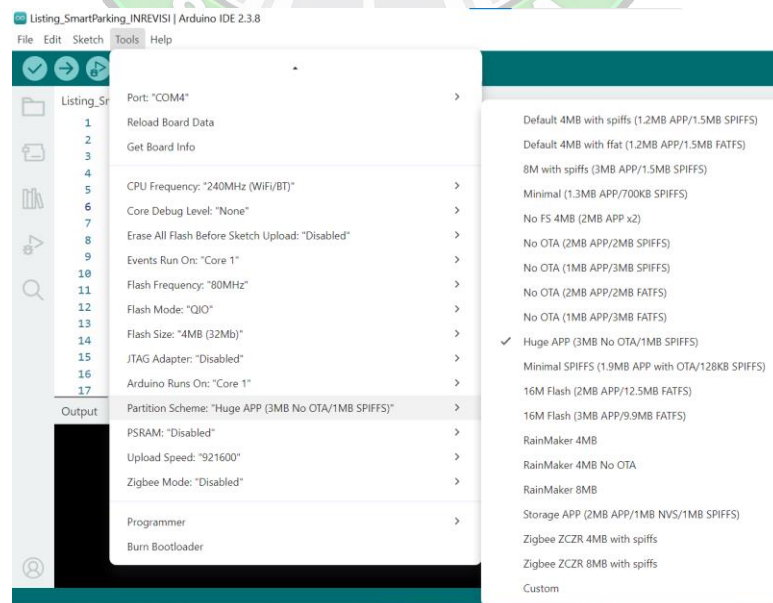
- 3) Masukkan program untuk In dan Out pada masing masing platform Arduino IDE, dan install Library untuk program In dan Out
- 4) Pilih Board ESP32 Dev Module



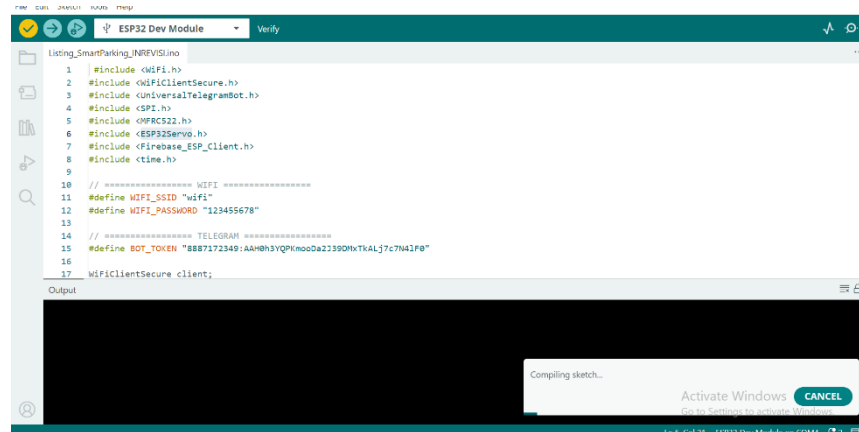
- 5) Pilih menu "Tools" dan klik pada menu port untuk mengirimkan data program ke mikrokontroler ESP32



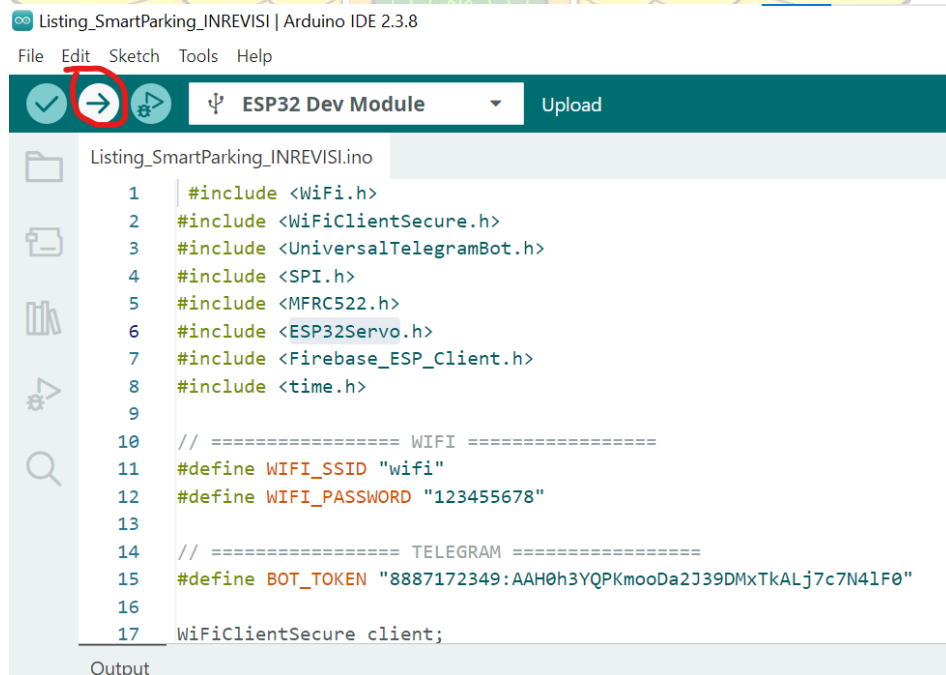
- 6) Atur *Partition Scheme* pada menu "Tools" menjadi “Huge APP (3MB No OTA/1MB SPIFFS)” pada Arduino IDE agar kapasitas memori program lebih besar sehingga proses upload program *smart parking* dapat berjalan lebih optimal dan tidak mengalami kekurangan memori



- 7) Sebelum program di upload ke mikrokontroler ESP32, *compiling* terlebih dahulu pada kedua program Arduino IDE untuk meminimalisir kesalahan atau error pada program



- 8) Jika tidak ada kesalahan atau error pada program, maka bisa langsung di upload ke mikrokontroler ESP32



- 9) Periksa koneksi alat agar saat disimulasikan alat *smart parking* bisa berjalan dengan lancar

2. Memulai Reservasi Tempat Parkir

- a. Buka Bot Telegram Smart Parking melalui hasil scan QR Code.



- b. Ketik perintah `"/cekparkir"` untuk melihat ketersediaan tempat parkir
- c. Bot Telegram akan menampilkan status seluruh tempat parkir yang tersedia maupun yang sudah terisi.
- d. Pilih tempat parkir yang masih tersedia.
- e. Ketik perintah reservasi sesuai nomor tempat parkir yang diinginkan, contoh: `"booking A1"`.
- f. Bot Telegram akan memproses permintaan reservasi dan mengirimkan notifikasi bahwa tempat parkir berhasil dipesan.
- g. Status tempat parkir yang dipilih akan berubah menjadi `"Terisi"` pada sistem monitoring.

3. Proses Parkir Kendaraan

- a. Pengguna menuju lokasi tempat parkir yang telah dipesan melalui Bot Telegram.
- b. Pengguna menempelkan kartu RFID pada RFID Reader yang tersedia di area parkir untuk melakukan verifikasi identitas dan reservasi.
- c. Sistem akan membaca data kartu RFID dan mencocokkannya dengan data

reservasi yang tersimpan pada database Firebase.

- d. Jika data RFID dan reservasi valid, *barrier gate* (motor servo) akan terbuka secara otomatis.
- e. Pengguna memarkirkan kendaraan pada tempat parkir yang telah dipesan.
- f. Sensor inframerah akan mendeteksi keberadaan kendaraan dan mengubah status tempat parkir menjadi "Terisi".
- g. Data kondisi tempat parkir akan dikirim secara otomatis ke Firebase sehingga status tempat parkir diperbarui.

4. Verifikasi Sistem

- a. Pastikan status tempat parkir pada Bot Telegram sesuai dengan kondisi sebenarnya di area parkir.
- b. Periksa apakah sensor inframerah dapat mendeteksi kendaraan dengan baik.
- c. Pastikan data reservasi tersimpan pada database Firebase.
- d. Jika terjadi ketidaksesuaian data, lakukan pemeriksaan koneksi internet.

5. Penyelesaian

- a. Setelah pengguna selesai menggunakan tempat parkir, kendaraan diarahkan menuju pintu keluar area parkir.
- b. Pengguna menempelkan kembali kartu RFID pada RFID Reader yang berada di pintu keluar sebagai proses verifikasi data parkir.
- c. Sistem akan membaca kartu RFID dan mencocokkannya dengan data kendaraan yang tersimpan pada database Firebase.
- d. Jika kartu RFID valid, sistem akan menghitung durasi parkir berdasarkan waktu masuk dan waktu keluar kendaraan.
- e. Hasil perhitungan durasi parkir akan ditampilkan pada layar OLED beserta informasi biaya parkir yang harus dibayarkan.
- f. Sistem juga akan menampilkan informasi saldo pengguna yang tersisa setelah biaya parkir dipotong secara otomatis.
- g. Setelah proses verifikasi RFID berhasil dan transaksi selesai dilakukan, motor servo (*barrier gate*) akan terbuka secara otomatis sehingga kendaraan dapat keluar dari area parkir.

- h. Sensor inframerah akan mendeteksi bahwa tempat parkir telah kosong dan mengubah status tempat parkir.
- i. Data ketersediaan tempat parkir akan diperbarui secara otomatis.

Catatan:

- Pengguna wajib melakukan reservasi tempat parkir melalui Bot Telegram sebelum memasuki area parkir.
- Perintah "/cekparkir" digunakan untuk memantau ketersediaan tempat parkir.
- Perintah "booking A1" digunakan untuk memesan tempat parkir sesuai nomor yang diinginkan (A1 dapat diganti dengan nomor tempat parkir lain yang tersedia).
- Sistem monitoring dan reservasi terhubung dengan Bot Telegram
- Setelah reservasi berhasil dilakukan, pengguna memiliki batas waktu 10 menit untuk tiba di lokasi parkir dan melakukan verifikasi menggunakan kartu RFID.
- Apabila pengguna tidak datang dalam waktu lebih dari 10 menit sejak reservasi dilakukan, maka status reservasi akan hangus secara otomatis dan tempat parkir akan kembali tersedia untuk pengguna lain.
- Pengguna wajib menggunakan kartu RFID yang telah terdaftar oleh operator untuk proses masuk dan keluar area parkir.

LAMPIRAN B

CODING ARDUINO IDE

1. Program In

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <SPI.h>
#include <MFRC522.h>
#include <ESP32Servo.h>
#include <Firebase_ESP_Client.h>
#include <time.h>

// ===== WIFI =====
#define WIFI_SSID "wifi"
#define WIFI_PASSWORD "123455678"

// ===== TELEGRAM =====
#define BOT_TOKEN "8887172349:AAH0h3YQPKmooDa2J39DMxTkALj7c7N4lF0"

WiFiClientSecure client;
UniversalTelegramBot bot(BOT_TOKEN, client);

// ===== FIREBASE =====
#define API_KEY "AIzaSyABapIo9C06o_5Y5WpPUfq8K8MvfaoGbEg"
#define DATABASE_URL "https://smartparkingpoltekbangsby-default-
rtbd.firebaseio.com"

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

// ===== RFID =====
#define SS_PIN 27
#define RST_PIN 14

MFRC522 rfid(SS_PIN, RST_PIN);

// ===== SERVO =====
Servo servoIn;

#define SERVO_PIN 25
```



```

// ===== IR SENSOR =====
#define IR_A1 32
#define IR_A2 33
#define IR_A3 34

// ===== LED =====
#define LED_HIJAU 13
#define LED_MERAH 26

unsigned long lastTimeBot = 0;
int botDelay = 3000;

// =====
// LED CONTROL
// =====
void updateLED()
{
    bool a1 = (digitalRead(IR_A1) == LOW);
    bool a2 = (digitalRead(IR_A2) == LOW);
    bool a3 = (digitalRead(IR_A3) == LOW);

    if (a1 && a2 && a3)
    {
        digitalWrite(LED_HIJAU, LOW);
        digitalWrite(LED_MERAH, HIGH);
    }
    else
    {
        digitalWrite(LED_HIJAU, HIGH);
        digitalWrite(LED_MERAH, LOW);
    }
}

// =====
// FORMAT WAKTU
// =====
String formatWaktu(time_t waktu)
{
    struct tm *tm_info = localtime(&waktu);

    char buffer[30];

    strftime(
        buffer,
        sizeof(buffer),

```

```

        "%d-%m-%Y %H:%M",
        tm_info
    );

    return String(buffer);
}

// =====
// BOOKING EXPIRED 10 MENIT
// =====
void cekBookingExpired()
{
    String slotList[3] =
    {
        "A1",
        "A2",
        "A3"
    };

    time_t now;
    time(&now);

    for (int i = 0; i < 3; i++)
    {
        String slot = slotList[i];

        if (Firebase.RTDB.getInt(
            &fbdo,
            "/booking/" + slot + "/waktu"))
        {
            long waktuBooking =
                fbdo.intData();

            if ((now - waktuBooking) > 600)
            {
                String chatID = "";

                if (Firebase.RTDB.getString(
                    &fbdo,
                    "/booking/" + slot + "/chatid"))
                {
                    chatID =
                        fbdo.stringData();
                }
            }
        }
    }
}

```

```

        if (chatID.length() > 0)
        {
            bot.sendMessage(
                chatID,
                "🕒 BOOKING HANGUS\n\n"
                "Slot : " + slot +
                "\nMelebihi 10 menit.",
                ""
            );
        }

        Firebase.RTDB.deleteNode(
            &fbdo,
            "/booking/" + slot
        );
    }
}
}

// =====
// SETUP
// =====
void setup()
{
    Serial.begin(115200);

    pinMode(IR_A1, INPUT_PULLUP);
    pinMode(IR_A2, INPUT_PULLUP);
    pinMode(IR_A3, INPUT_PULLUP);

    pinMode(LED_HIJAU, OUTPUT);
    pinMode(LED_MERAH, OUTPUT);

    digitalWrite(LED_HIJAU, HIGH);
    digitalWrite(LED_MERAH, LOW);

    // WIFI
    WiFi.begin(
        WIFI_SSID,
        WIFI_PASSWORD
    );

    while (WiFi.status() != WL_CONNECTED)
    {

```

```

        delay(300);
        Serial.print(".");
    }

    Serial.println();
    Serial.println("WIFI OK");

    client.setInsecure();

    // FIREBASE
    config.api_key = API_KEY;
    config.database_url =
        DATABASE_URL;

    Firebase.signUp(
        &config,
        &auth,
        "",
        ""
    );

    Firebase.begin(
        &config,
        &auth
    );

    Firebase.reconnectWiFi(true);

    fbdo.setBSSLBufferSize(
        2048,
        512
    );

    config.timeout.serverResponse =
        10000;

    Serial.println("FIREBASE OK");

    // NTP
    configTime(
        7 * 3600,
        0,
        "pool.ntp.org"
    );

```

```

// RFID
SPI.begin();
SPI.setFrequency(4000000);

rfid.PCD_Init();

Serial.println("RFID OK");

// SERVO
ESP32PWM::allocateTimer(0);

servoIn.attach(
    SERVO_PIN
);

servoIn.write(160);

Serial.println("SERVO OK");

bot.sendMessage(
    "701519670",
    "🚗 SMART PARKING ONLINE",
    ""
);
}
// =====
// TELEGRAM
// =====
void handleTelegram()
{
    int msgCount =
        bot.getUpdates(
            bot.last_message_received + 1
        );

    while (msgCount)
    {
        for (int i = 0; i < msgCount; i++)
        {
            String text =
                bot.messages[i].text;

            String userChatID =
                bot.messages[i].chat_id;

            text.toUpperCase();

```



```

// =====
// CEK PARKIR
// =====
if (text == "/CEKPARKIR")
{
    String msg =
        "🚗 STATUS PARKIR 🚗\n\n";

    String slot[3] =
    {
        "A1",
        "A2",
        "A3"
    };

    int pinIR[3] =
    {
        IR_A1,
        IR_A2,
        IR_A3
    };

    for (int j = 0; j < 3; j++)
    {
        String namaSlot =
            slot[j];

        // SLOT TERISI
        if (digitalRead(pinIR[j]) == LOW)
        {
            msg += namaSlot +
                " : □ TERISI\n\n";
        }
        else
        {
            bool adaBooking = false;

            if (Firebase.RTDB.getString(
                &fbdo,
                "/booking/" +
                namaSlot +
                "/uid"))
            {
                String uidBooking =
                    fbdo.stringData();
            }
        }
    }
}

```

```

        uidBooking.trim();

        if (uidBooking.length() > 0)
        {
            adaBooking = true;
        }
    }

    if (adaBooking)
    {
        msg += namaSlot +
            " : □ BOOKING\n\n";
    }
    else
    {
        msg += namaSlot +
            " : □ KOSONG\n\n";
    }
}

bot.sendMessage(
    userChatID,
    msg,
    ""
);
}

// =====
// BOOK SLOT
// =====
else if (text.startsWith("BOOK "))
{
    String slot =
        text.substring(5);

    slot.trim();
    slot.toUpperCase();

    if (
        slot != "A1" &&
        slot != "A2" &&
        slot != "A3"
    )
    {
        bot.sendMessage(

```

```

        userChatID,
        "✗ SLOT TIDAK VALID\n\n"
        "Gunakan:\n"
        "BOOK A1\n"
        "BOOK A2\n"
        "BOOK A3",
        ""
    );

    continue;
}

// =====
// AMBIL UID USER
// =====
String uidUser = "";

if (!Firebase.RTDB.getString(
    &fbdo,
    "/users/" +
    userChatID +
    "/uid"))
{
    bot.sendMessage(
        userChatID,
        "✗ UID RFID BELUM TERDAFTAR",
        ""
    );

    continue;
}

uidUser =
    fbdo.stringData();

uidUser.trim();
uidUser.toUpperCase();

// =====
// CEK SUDAH BOOKING ATAU BELUM
// =====
bool sudahBooking =
    false;

String daftarSlot[3] =
{

```

```

        "A1",
        "A2",
        "A3"
    };

    for (int k = 0; k < 3; k++)
    {
        if (Firebase.RTDB.getString(
            &fbdo,
            "/booking/" +
            daftarSlot[k] +
            "/chatid"))
        {
            String c =
                fbdo.stringData();

            if (c == userChatID)
            {
                sudahBooking = true;
                break;
            }
        }
    }

    if (sudahBooking)
    {
        bot.sendMessage(
            userChatID,
            "✗ ANDA MASIH MEMILIKI BOOKING AKTIF",
            ""
        );

        continue;
    }

    // =====
    // CEK SLOT TERISI SENSOR
    // =====
    bool slotTerisi = false;

    if (slot == "A1")
        slotTerisi =
            (digitalRead(IR_A1) == LOW);

    if (slot == "A2")
        slotTerisi =

```

```

        (digitalRead(IR_A2) == LOW);

    if (slot == "A3")
        slotTerisi =
            (digitalRead(IR_A3) == LOW);

    if (slotTerisi)
    {
        bot.sendMessage(
            userChatID,
            "✕ SLOT " +
            slot +
            " SEDANG TERISI",
            ""
        );

        continue;
    }

    // =====
    // CEK SUDAH DIBOOKING
    // =====
    String uidBooking = "";

    if (Firebase.RTDB.getString(
        &fbdo,
        "/booking/" +
        slot +
        "/uid"))
    {
        uidBooking =
            fbdo.stringData();
    }

    uidBooking.trim();

    if (uidBooking.length() > 0)
    {
        bot.sendMessage(
            userChatID,
            "✕ SLOT SUDAH DIBOOKING USER LAIN",
            ""
        );

        continue;
    }

```

```

        // =====
        // SIMPAN BOOKING
        // =====
        time_t now;
        time(&now);

bool ok1 = Firebase.RTDB.setString(
    &fbdo,
    "/booking/" + slot + "/uid",
    uidUser
);

bool ok2 = Firebase.RTDB.setString(
    &fbdo,
    "/booking/" + slot + "/chatid",
    userChatID
);

bool ok3 = Firebase.RTDB.setInt(
    &fbdo,
    "/booking/" + slot + "/waktu",
    now
);

Serial.println("===== BOOKING DEBUG =====");
Serial.println("SLOT      : " + slot);
Serial.println("UID       : " + uidUser);
Serial.println("CHATID    : " + userChatID);

Serial.println("SET UID      : " + String(ok1));
Serial.println("SET CHATID   : " + String(ok2));
Serial.println("SET WAKTU    : " + String(ok3));

if(!ok1)
    Serial.println("ERR UID : " + fbdo.errorReason());

if(!ok2)
    Serial.println("ERR CHATID : " + fbdo.errorReason());

if(!ok3)
    Serial.println("ERR WAKTU : " + fbdo.errorReason());

if(ok1 && ok2 && ok3)
{
    bot.sendMessage(
        userChatID,

```



```

        "✅ BOOKING BERHASIL\n\n"
        "Slot : " + slot +
        "\nUID : " + uidUser,
        ""
    );
}
else
{
    bot.sendMessage(
        userChatID,
        "❌ GAGAL MENYIMPAN BOOKING KE FIREBASE",
        ""
    );
}
    }
}

    msgCount =
        bot.getUpdates(
            bot.last_message_received + 1
        );
}
}
// =====
// LOOP
// =====
void loop()
{
    cekBookingExpired();

    updateLED();

    if (WiFi.status() != WL_CONNECTED)
    {
        WiFi.reconnect();
        delay(200);
    }

    if (millis() - lastTimeBot > botDelay)
    {
        handleTelegram();
        lastTimeBot = millis();
    }

    // =====
    // RFID

```

```

// =====
delay(50);

if (!rfid.PICC_IsNewCardPresent())
    return;

if (!rfid.PICC_ReadCardSerial())
    return;

String uid = "";

for (byte i = 0; i < rfid.uid.size; i++)
{
    if (rfid.uid.uidByte[i] < 0x10)
        uid += "0";

    uid += String(
        rfid.uid.uidByte[i],
        HEX
    );
}

uid.toUpperCase();

Serial.println(
    "UID : " + uid
);

String slotAktif = "";
String bookingChatID = "";

String slotList[3] =
{
    "A1",
    "A2",
    "A3"
};

// =====
// CARI BOOKING YANG SESUAI UID
// =====
for (int i = 0; i < 3; i++)
{
    String slot =
        slotList[i];

```

```

if (Firebase.RTDB.getString(
    &fbdo,
    "/booking/" +
    slot +
    "/uid"))
{
    String uidBooking =
        fbdo.stringData();

    uidBooking.trim();
    uidBooking.toUpperCase();

    bool slotKosong = false;

    if (slot == "A1")
        slotKosong =
            (digitalRead(IR_A1) == HIGH);

    if (slot == "A2")
        slotKosong =
            (digitalRead(IR_A2) == HIGH);

    if (slot == "A3")
        slotKosong =
            (digitalRead(IR_A3) == HIGH);

    if (
        uidBooking == uid &&
        slotKosong
    )
    {
        slotAktif = slot;

        if (Firebase.RTDB.getString(
            &fbdo,
            "/booking/" +
            slot +
            "/chatid"))
        {
            bookingChatID =
                fbdo.stringData();
        }

        break;
    }
}

```

```

}

// =====
// UID SESUAI BOOKING
// =====
if (slotAktif != "")
{
    time_t now;
    time(&now);

    Firebase.RTDB.setString(
        &fbdo,
        "/parkir/" +
        uid +
        "/slot",
        slotAktif
    );

    Firebase.RTDB.setString(
        &fbdo,
        "/parkir/" +
        uid +
        "/status",
        "MASUK"
    );

    Firebase.RTDB.setInt(
        &fbdo,
        "/parkir/" +
        uid +
        "/waktu",
        now
    );

    // =====
    // DATA SLOT TERISI
    // =====
    Firebase.RTDB.setString(
        &fbdo,
        "/slot/" +
        slotAktif +
        "/uid",
        uid
    );

    Firebase.RTDB.setInt(

```

```

        &fbdo,
        "/slot/" +
        slotAktif +
        "/waktu",
        now
    );

    // =====
    // BUKA PORTAL
    // =====
    servoIn.write(10);

    delay(10000);

    servoIn.write(160);

    delay(500);

    // =====
    // HAPUS BOOKING
    // =====
    Firebase.RTDB.deleteNode(
        &fbdo,
        "/booking/" +
        slotAktif
    );

    // =====
    // NOTIF PEMILIK BOOKING
    // =====
    if (bookingChatID.length() > 0)
    {
        bot.sendMessage(
            bookingChatID,
            "✅ KENDARAAN MASUK\n\n"
            "Slot : " +
            slotAktif +
            "\nUID : " +
            uid,
            ""
        );
    }

    Serial.println(
        "AKSES DITERIMA"
    );

```

```

    }
    else
    {
        Serial.println(
            "UID TIDAK SESUAI BOOKING"
        );
    }

    rfid.PICC_HaltA();

    rfid.PCD_StopCrypto1();

    delay(200);
}

```

2. Program Out

```

#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <SPI.h>
#include <MFRC522.h>
#include <ESP32Servo.h>
#include <Firebase_ESP_Client.h>
#include <time.h>
#include <math.h>

#include <Wire.h>
#include <U8g2lib.h>

// ===== OLED =====
U8G2_SH1106_128X64_NONAME_1_HW_I2C u8g2(U8G2_R0);

// ===== WIFI =====
#define WIFI_SSID "IndiHome"
#define WIFI_PASSWORD "1sampai8"

// ===== TELEGRAM =====
#define BOT_TOKEN "8887172349:AAH0h3YQPKmooDa2J39DMxTkALj7c7N41F0"

WiFiClientSecure client;
UniversalTelegramBot bot(BOT_TOKEN, client);

// ===== FIREBASE =====
#define API_KEY "AIzaSyABapIo9C06o_5Y5WpPUfq8K8MvfaoGbEg"

```



```

#define DATABASE_URL "https://smartparkingpoltekbangsby-default-
rtdb.firebaseio.com"

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

// ===== RFID =====
#define SS_PIN 27
#define RST_PIN 14

MFRC522 rfid(SS_PIN, RST_PIN);

// ===== SERVO =====
Servo servoOut;

#define SERVO_PIN 26

// =====
// OLED TAP KARTU
// =====
void tampilTapKartu()
{
    u8g2.firstPage();

    do
    {
        u8g2.setFont(
            u8g2_font_logisoso20_tf
        );

        u8g2.drawStr(
            18,
            28,
            "TAP"
        );

        u8g2.drawStr(
            2,
            58,
            "KARTU"
        );

    } while (u8g2.nextPage());
}

```

```

// =====
// OLED INFO PARKIR
// =====
void tampilInfo(
    String slot,
    int durasi,
    int biaya,
    int saldo)
{
    u8g2.firstPage();

    do
    {
        u8g2.setFont(
            u8g2_font_6x12_tf
        );

        u8g2.drawStr(
            22,
            10,
            "PARKIR OUT"
        );

        u8g2.drawStr(
            0,
            25,
            ("Slot : " + slot).c_str()
        );

        u8g2.drawStr(
            0,
            38,
            ("Jam : " +
             String(durasi) +
             " M").c_str()
        );

        u8g2.drawStr(
            0,
            51,
            ("Biaya : " +
             String(biaya)).c_str()
        );

        u8g2.drawStr(
            0,

```

```

        64,
        ("Saldo : " +
         String(saldo)).c_str()
    );

    } while (
        u8g2.nextPage()
    );
}

// =====
// OLED ERROR
// =====
void tampilError(
    String msg)
{
    u8g2.firstPage();

    do
    {
        u8g2.setFont(
            u8g2_font_7x14B_tf
        );

        u8g2.drawStr(
            5,
            35,
            msg.c_str()
        );

    } while (
        u8g2.nextPage()
    );
}

// =====
// CEK WAKTU VALID
// =====
bool waktuValid()
{
    time_t now;

    time(&now);

    return (
        now > 1700000000
    );
}

```

```

    );
}

// =====
// HITUNG BIAYA
// =====
int hitungBiaya(
    int menit)
{
    if (menit <= 120)
    {
        return 5000;
    }

    return
        5000 +
        ((menit - 120 + 59) / 60)
        * 5000;
}

// =====
// FORMAT TANGGAL
// =====
String formatWaktu(
    time_t waktu)
{
    struct tm *tm_info =
        localtime(&waktu);

    char buffer[30];

    strftime(
        buffer,
        sizeof(buffer),
        "%d-%m-%Y %H:%M",
        tm_info
    );

    return String(buffer);
}

// =====
// CARI CHAT ID BERDASARKAN UID
// =====
String cariChatID(
    String uidCari)

```

```

{
    if (
        Firebase.RTDB.getJSON(
            &fbdo,
            "/users"
        )
    )
    {
        FirebaseJson &json =
            fbdo.jsonObject();

        size_t len =
            json.iteratorBegin();

int type;
String key;
String value;

        for (
            size_t i = 0;
            i < len;
            i++
        )
        {
            json.iteratorGet(
                i,
                type,
                key,
                value
            );

            String pathUID =
                "/users/" +
                key +
                "/uid";

            if (
                Firebase.RTDB.getString(
                    &fbdo,
                    pathUID
                )
            )
            {
                String uidFirebase =
                    fbdo.stringData();
            }
        }
    }
}

```

```

        uidFirebase.trim();
        uidFirebase.toUpperCase();

        if (
            uidFirebase ==
            uidCari
        )
        {
            json.iteratorEnd();

            return key; // CHAT ID
        }
    }
}

    json.iteratorEnd();
}

return "";
}
// =====
// SETUP
// =====
void setup()
{
    Serial.begin(115200);

    // ===== OLED =====
    Wire.begin(21, 22);

    u8g2.begin();

    tampilTapKartu();

    Serial.println("OLED OK");

    // ===== WIFI =====
    WiFi.begin(
        WIFI_SSID,
        WIFI_PASSWORD
    );

    while (
        WiFi.status() != WL_CONNECTED
    )
    {

```

```

    delay(300);
    Serial.print(".");
}

Serial.println();
Serial.println("WIFI OK");

client.setInsecure();

// ===== FIREBASE =====
config.api_key =
    API_KEY;

config.database_url =
    DATABASE_URL;

Firebase.signUp(
    &config,
    &auth,
    "",
    ""
);

Firebase.begin(
    &config,
    &auth
);

Firebase.reconnectWiFi(
    true
);

fbdo.setBSSLBufferSize(
    2048,
    512
);

config.timeout.serverResponse =
    10000;

Serial.println(
    "FIREBASE OK"
);

// ===== NTP =====
configTime(

```



```

    7 * 3600,
    0,
    "pool.ntp.org"
);

Serial.print(
    "SYNC TIME"
);

struct tm timeinfo;

while (
    !getLocalTime(
        &timeinfo
    )
)
{
    Serial.print(".");
    delay(500);
}

Serial.println();
Serial.println(
    "TIME OK"
);

// ===== RFID =====
SPI.begin(
    18,
    19,
    23,
    SS_PIN
);

pinMode(
    RST_PIN,
    OUTPUT
);

digitalWrite(
    RST_PIN,
    LOW
);

delay(50);

```

```

digitalWrite(
    RST_PIN,
    HIGH
);

delay(50);

SPI.setFrequency(
    1000000
);

rfid.PCD_Init();

byte version =
    rfid.PCD_ReadRegister(
        MFRC522::VersionReg
    );

Serial.print(
    "RFID VERSION : 0x"
);

Serial.println(
    version,
    HEX
);

if (
    version == 0x00 ||
    version == 0xFF
)
{
    Serial.println(
        "RFID ERROR"
    );
}
else
{
    Serial.println(
        "RFID OK"
    );
}

// ===== SERV0 =====
ESP32PWM::allocateTimer(
    0

```

```

);

servoOut.attach(
    SERVO_PIN
);

// POSISI TERTUTUP
servoOut.write(
    150
);

delay(1000);

Serial.println(
    "SERVO OK"
);

// ===== NOTIF START =====
if (
    Firebase.RTDB.getJSON(
        &fbdo,
        "/users"
    )
)
{
    FirebaseJson &json =
        fbdo.jsonObject();

    size_t len =
        json.iteratorBegin();

    int type;
    String key;
    String value;

    for (
        size_t i = 0;
        i < len;
        i++
    )
    {
        json.iteratorGet(
            i,
            type,
            key,
            value

```

```

    );

    bot.sendMessage(
        key,
        "🚗 SMART PARKING OUT ONLINE",
        ""
    );
}

json.iteratorEnd();
}

tampilTapKartu();

Serial.println(
    "SYSTEM READY"
);
}
// =====
// LOOP
// =====
void loop()
{
    if (WiFi.status() != WL_CONNECTED)
    {
        WiFi.reconnect();
        delay(200);
    }

    delay(30);

    // =====
    // CEK RFID
    // =====
    if (!rfid.PICC_IsNewCardPresent())
        return;

    if (!rfid.PICC_ReadCardSerial())
        return;

    String uid = "";

    for (byte i = 0; i < rfid.uid.size; i++)
    {
        if (rfid.uid.uidByte[i] < 0x10)
            uid += "0";

```

```

        uid += String(
            rfid.uid.uidByte[i],
            HEX
        );
    }

    uid.toUpperCase();

    Serial.println("=====");
    Serial.println("UID : " + uid);

    // =====
    // CARI CHAT ID PEMILIK UID
    // =====
    String chatID =
        cariChatID(uid);

    if (chatID == "")
    {
        tampilError(
            "UID TDK TERDAFTAR"
        );

        delay(3000);

        tampilTapKartu();

        return;
    }

    // =====
    // CEK WAKTU VALID
    // =====
    if (!waktuValid())
    {
        tampilError(
            "TIME ERROR"
        );

        return;
    }

    time_t now;

    time(&now);

```

```

String base =
    "/parkir/" + uid;

String saldoPath =
    "/users/" +
    chatID +
    "/saldo";

// =====
// AMBIL SLOT
// =====
if (
    !Firebase.RTDB.getString(
        &fbdo,
        base + "/slot"
    )
)
{
    tampilError(
        "DATA TDK ADA"
    );

    delay(3000);

    tampilTapKartu();

    return;
}

String slot =
    fbdo.stringData();

Serial.println(
    "SLOT : " + slot
);

// =====
// AMBIL WAKTU MASUK
// =====
if (
    !Firebase.RTDB.getInt(
        &fbdo,
        base + "/waktu"
    )
)
{

```

```

    tampilError(
        "WAKTU ERROR"
    );

    delay(3000);

    tampilTapKartu();

    return;
}

time_t masuk =
    fbdo.intData();

int durasiMenit =
    max(
        0,
        (int)(now - masuk) / 60
    );

int biaya =
    hitungBiaya(
        durasiMenit
    );

// =====
// AMBIL SALDO USER
// =====
if (
    !Firebase.RTDB.getInt(
        &fbdo,
        saldoPath
    )
)
{
    tampilError(
        "SALDO ERROR"
    );

    delay(3000);

    tampilTapKartu();

    return;
}

```

```

int saldo =
    fbdo.intData();

// =====
// SALDO TIDAK CUKUP
// =====
if (saldo < biaya)
{
    tampilError(
        "SALDO KURANG"
    );

    bot.sendMessage(
        chatID,
        "❌ SALDO TIDAK CUKUP\n\n"
        "Biaya : Rp " +
        String(biaya) +
        "\nSaldo : Rp " +
        String(saldo),
        ""
    );

    delay(3000);

    tampilTapKartu();

    return;
}

int sisa =
    saldo - biaya;

// =====
// TAMPILKAN OLED
// =====
tampilInfo(
    slot,
    durasiMenit,
    biaya,
    sisa
);

// =====
// KIRIM TELEGRAM
// =====
String msg =

```



```

"🚗 PARKIR KELUAR\n\n"
"UID : " + uid + "\n" +
"Slot : " + slot + "\n\n" +
"Durasi : " +
String(durasiMenit) +
" menit\n\n" +
"Biaya : Rp " +
String(biaya) +
"\n\n" +
"Saldo Awal : Rp " +
String(saldo) +
"\n" +
"Saldo Akhir : Rp " +
String(sisa) +
"\n\n"
"Portal akan terbuka 5 detik lagi";

bot.sendMessage(
    chatID,
    msg,
    ""
);

// =====
// TUNGGU 5 DETIK
// =====
delay(5000);

// =====
// UPDATE SALDO
// =====
Firebase.RTDB.setInt(
    &fbdo,
    saldoPath,
    sisa
);

// =====
// HAPUS DATA PARKIR
// =====
Firebase.RTDB.deleteNode(
    &fbdo,
    base
);

```

```

// =====
// HAPUS DATA SLOT
// =====
Firebase.RTDB.deleteNode(
    &fbdo,
    "/slot/" + slot
);

// =====
// BUKA PORTAL
// =====
servoOut.write(10);

delay(5000);

// =====
// TUTUP PORTAL
// =====
servoOut.write(150);

delay(500);

tampilTapKartu();

rfid.PICC_HaltA();

rfid.PCD_StopCrypto1();

delay(200);
}

```



LAMPIRAN C
PAMFLET / BROSUS SMART PARKING

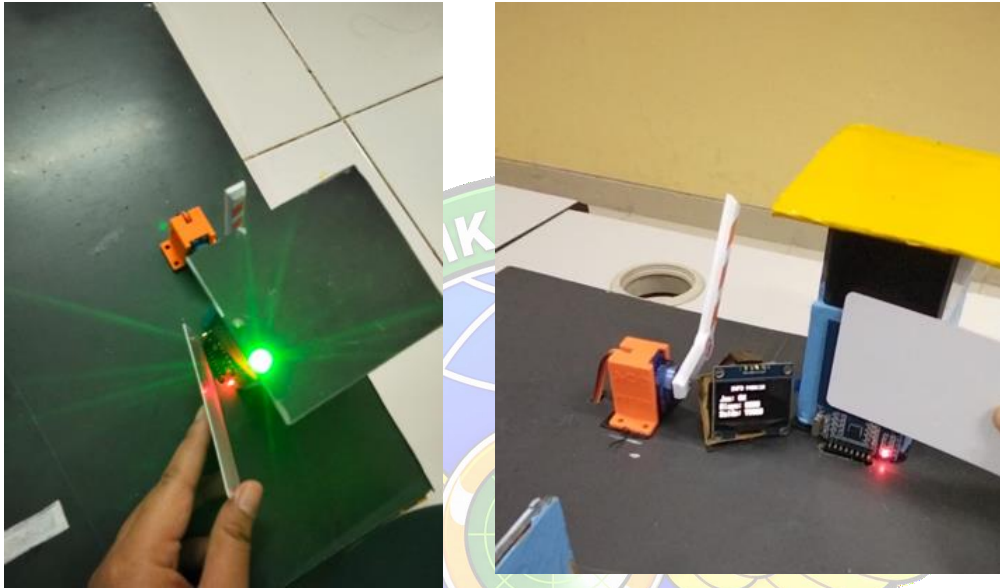


LAMPIRAN D

HASIL PENGUJIAN

A. Pengujian Perangkat Keras (*Hardware*)

1. Pengujian RFID, Motor Servo, dan Status Parking LEDs



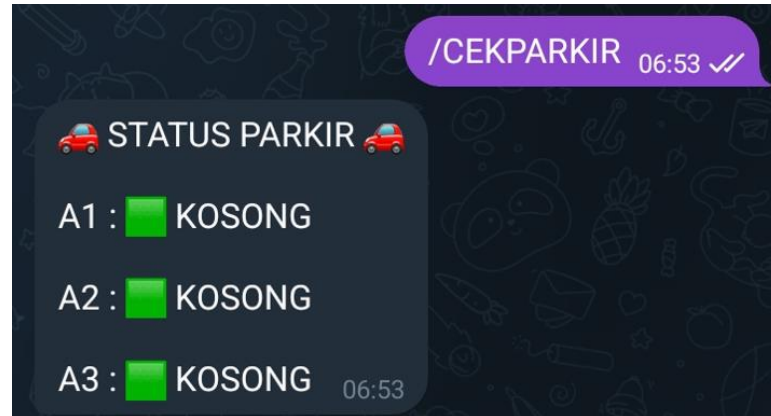
2. Pengujian Sensor Infrared



B. Pengujian Perangkat Lunak (*Software*)

1. Pengujian Tampilan Bot Telegram

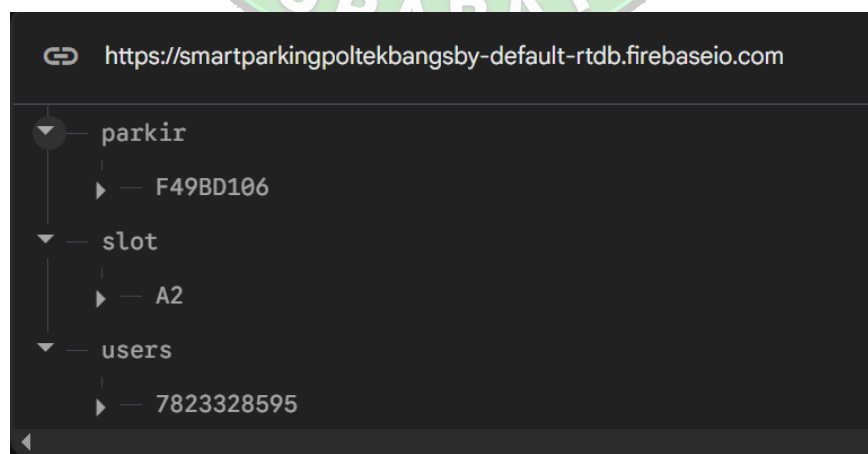
a. Informasi Ketersediaan Tempat Parkir



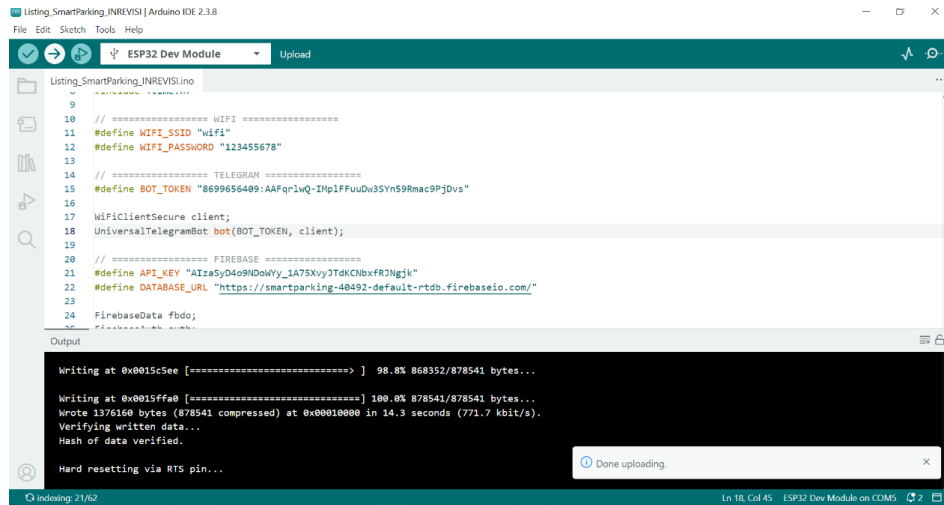
b. Reservasi



2. Pengujian Monitoring Firebase



3. Pengujian Arduino IDE



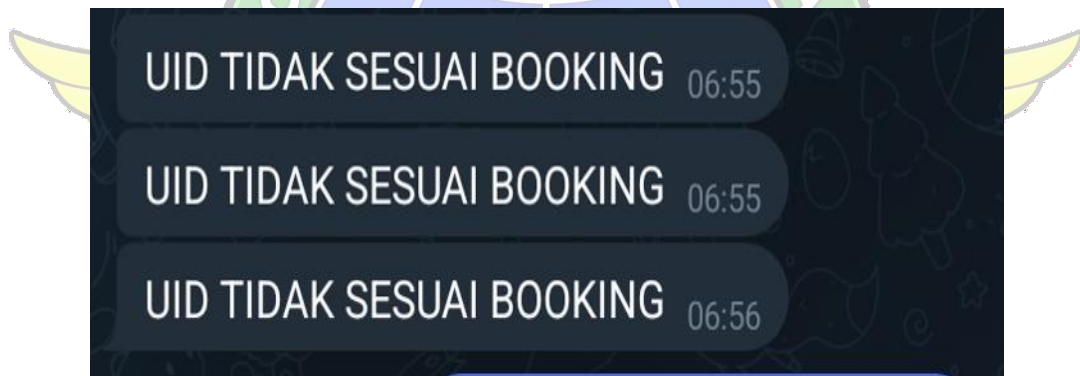
```
Listing_SmartParking_INREVISI.ino
9
10 // ===== WIFI =====
11 #define WIFI_SSID "wifi"
12 #define WIFI_PASSWORD "123455678"
13
14 // ===== TELEGRAM =====
15 #define BOT_TOKEN "8699656409:AAFqrlwQ-IMp1FFuuDu3SYn59Rmac9PjDvs"
16
17 WiFiClientSecure client;
18 UniversalTelegramBot bot(BOT_TOKEN, client);
19
20 // ===== FIREBASE =====
21 #define API_KEY "AIzaSyD4o9NDokWy_1A75XvyJTdKChbxFR3Ngjk"
22 #define DATABASE_URL "https://smartparking-40492-default-rtdb.firebaseio.com/"
23
24 FirebaseData fbd;
25 FirebaseData fdb;

Output
Writing at 0x0015c5ee [=====] 98.8% 868352/878541 bytes...
Writing at 0x0015ffa0 [=====] 100.0% 878541/878541 bytes...
Wrote 1376168 bytes (878541 compressed) at 0x00010000 in 14.3 seconds (771.7 kbit/s).
Verifying written data...
Hash of data verified.
Done uploading.
Hard resetting via RTS pin...

Q indexing: 21/62
Ln 18, Col 45 ESP32 Dev Module on COM5 2
```

C. Pengujian Error Handling

1. Kegagalan Pembacaan RFID yang Tidak Sesuai



2. Pengujian Kondisi Tempat Parkir Sudah Terisi



DAFTAR RIWAYAT HIDUP



Nama : Deswinda Rosalia Putri
Nama Panggilan : Deswinda
Tempat, Tanggal lahir : Demak, 16 Desember 2004
Agama : Islam
Orang Tua : Wiwid Widaryo dan Dwi Kharismawati
Saudara : Kharesta dan Bagas
Alamat : Jl. Perkutut no.5 RT: 09/
RW: 05 Batusari, Mranggen, Demak,
Jawa Tengah
Hobi : Memasak

Latar Belakang Pendidikan:

- 2011-2017 : SD Negeri Mranggen 1
- 2017-2020 : SMP Negeri 9 Semarang
- 2020-2023 : SMA Negeri 2 Mranggen
- 2023-2026 : Politeknik Penerbangan Surabaya

On the Job Training:

- (Oktober 2025 – Desember 2025) Perum LPPNPI Kantor Cabang Surabaya Bandar Udara Internasional Juanda
- (Januari 2026 – Maret 2026) UPT Pelayanan Jasa Kebandarudaraan Abdulrachman Saleh Malang

Lisensi dan Sertifikat Kompetensi:

- Pengoperasian Pengukuran
- *Information System (IFS)*
- *Navigation System*
- Rating N (VOR/DME)

DAFTAR RIWAYAT HIDUP



DESWINDA ROSALIA PUTRI, lahir di Demak pada 16 Desember 2004. Anak pertama dari pasangan Bapak Wiwid Widaryo dan Ibu Dwi Kharismawati. Bertempat tinggal di Jl. Perkutut no.5 RT. 09/RW. 05, Batusari, Mranggen, Demak, Jawa Tengah. Memulai pendidikan di SD Negeri Mranggen 1 pada tahun 2011-2017, kemudian melanjutkan pendidikan di SMP Negeri 9 Semarang pada tahun 2017-2020. Melanjutkan Sekolah Menengah Atas di SMA Negeri 2 Mranggen, lulus pada tahun 2023. Memulai pendidikan D-III Teknik Navigasi Udara pada tahun 2023 dan direncanakan selesai pada tahun 2026 di Politeknik Penerbangan Surabaya. Selama masa pendidikan, telah mengikuti *On the Job Training (OJT)* di Perum LPPNPI Kantor Cabang Surabaya dari Oktober 2025 hingga Desember 2025 dan di UPT Pelayanan Jasa Kebandarudaraan Abdulrachman Saleh Malang dari Januari 2026 hingga Maret 2026.

