PROTOTYPE SISTEM PENGHITUNG OTOMATIS JUMLAH JEMAAT BERBASIS *IOT* PADA GEREJA GKE EWANGGELION

TUGAS AKHIR

Disusun Untuk Memenuhi Syarat Kelulusan Program Strata I Pada
Sekolah Tinggi Manajemen Informatika Dan Komputer (STMIK)
Palangkaraya

OLEH

IMANUEL NATALIS TIMISELA
C2155201097
PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER (STMIK) PALANGKARAYA
2025

PROTOTYPE SISTEM PENGHITUNG OTOMATIS JUMLAH JEMAAT BERBASIS *IOT* PADA GEREJA GKE EWANGGELION

TUGAS AKHIR

Disusun Untuk Memenuhi Syarat Kelulusan Program Strata I Pada
Sekolah Tinggi Manajemen Informatika Dan Komputer (STMIK)
Palangkaraya

OLEH

IMANUEL NATALIS TIMISELA
C2155201097
PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER (STMIK) PALANGKARAYA
2025

PERSETUJUAN

PROTOTYPE SISTEM PENGHITUNG OTOMATIS JUMLAH JEMAAT BERBASIS *IOT* PADA GEREJA GKE EWANGGELION

Tugas Akhir ini Telah Disetujui dan Disahkan

Pembimbing I,	Pembimbing II,
Herkules, S.Kom., M.Cs.	Hotmian Sitohang, M.Kom.
NIK. 198510042010106	NIK. 198503282008002

Mengetahui

Ketua STMIK Palangkaraya,

Ir. Hj. Siti Maryamah, M.M.

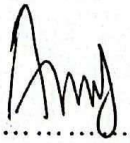
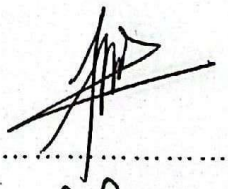
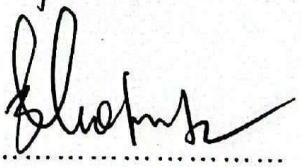
NIK. 196902021995004

PENGESAHAN

PROTOTYPE SISTEM PENGHITUNG OTOMATIS JUMLAH JEMAAT BERBASIS *IOT* PADA GEREJA GKE EWANGGELION

Tugas Akhir ini Telah Diujikan, Dinilai, dan Disahkan
Oleh Tim Penguji pada Tanggal 27 Mei 2025

Tim Penguji Tugas Akhir :

1. Elia Zakharia, M.T ..... 
   Ketua
2. Catharina Elmayantie, M.Pd ..... 
   Sekretaris
3. Amaya Andri Damaini, S.Kom., M.T. ..... 
   Anggota
4. Herkules, S.Kom., M.Cs. ..... 
   Anggota
5. Hotmian Sitohang, M.Kom. ..... 
   Anggota

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama Mahasiswa : **IMANUEL NATALIS TIMISELA**

NIM : **C2155201097**

menyatakan bahwa Tugas Akhir dengan judul :

PROTOTYPE SISTEM PENGHITUNG OTOMATIS JUMLAH JEMAAT BERBASIS *IOT* PADA GEREJA GKE EWANGGELION

adalah hasil karya saya dan bukan merupakan duplikasi sebagian atau seluruhnya dari karya orang lain, kecuali bagian yang sumber informasi cantumkan.

Pernyataan ini dibuat dengan sebenar-benarnya secara sadar dan bertanggungjawab dan saya bersedia menerima sanksi pembatalan Tugas Akhir apabila terbukti melakukan duplikasi terhadap Tugas Akhir atau karya ilmiah lain yang sudah ada.

Palangka Raya, 20 Mei 2025

Yang Membuat Pernyataan,



IMANUEL NATALIS TIMISELA

Dipindai dengan CamScanner

MOTTO DAN PERSEMBAHAN

“ Diberkatilah orang yang mengandalkan Tuhan, yang menaruh harapannya pada Tuhan! ”.(Yeremia 17:7)

Tugas Akhir ini kupersembahkan untuk :

Tuhan Yesus Kristus yang telah memberikan berkat, karunia, dan penyertaan-Nya dalam penyelesaian Tugas Akhir ini.

Orang tua tercinta, Daniel Octavianus Timisela & Resie yang telah memberikan semangat, dukungan, serta doa yang tiada henti nya untuk saya.

Dosen pembimbing, Bapak Herkules, S.Kom., M.Cs. dan Ibu Hotmian Sitohang, M.Kom. atas bimbingan dan arahan selama mengerjakan Tugas Akhir. Dan kepada Bapak dan Ibu dosen serta seluruh jajaran STMIK Palangka Raya atas bantuan dan didikan selama berkuliah.

INTISARI

Imanuel Natalis Timisela, C2155201097, 2025. *Prototype* Sistem Penghitung Otomatis Jumlah Jemaat Berbasis *IOT* Pada Gereja GKE Ewanggelion, Pembimbing I Herkules, S.Kom., M.Cs., Pembimbing II Hotmian Sitohang, M.Kom.

Pesatnya perkembangan teknologi pada berbagai aspek kehidupan mulai diterapkan. Salah satu nya pada tempat ibadah bagi umat kristiani seperti GKE Ewanggelion. Penghitungan jumlah jemaat yang beribadah di GKE Ewanggelion dilakukan secara manual oleh para petugas pada saat menyambut kedatangan jemaat pada pintu masuk. Dengan cara hitung manual tersebut tidak menutup kemungkinan terjadi kesalahan dari petugas dikarenakan terbaginya konsentrasi petugas antara mengurus persiapan dan kelengkapan untuk beribadah dan menghitung jumlah jemaat.

Penelitian ini bertujuan untuk memudahkan petugas gereja agar dapat berfokus pada kesiapan dan kelengkapan dalam ibadah yang akan berlangsung tanpa perlu membagi tugas lagi untuk menghitung jumlah jemaat.

Metode penelitian yang digunakan adalah *R&D (Research and Development)* dengan jenis model pengembangan *prototyping*. Data dikumpulkan melalui observasi, wawancara, dan studi pustaka. Penghitungan dilakukan menggunakan *Arduino Uno R3 ATMega328P* dengan sensor *PIR (Passive Infrared)* untuk mendeteksi pergerakan manusia berdasarkan perubahan panas dan sensor *Ultrasonic* untuk mendeteksi jarak antara sensor dan objek (Manusia).

Hasil Penghitungan Prototype Sistem Penghitung Otomatis Jumlah Jemaat Berbasis *IOT* Pada gereja GKE Ewanggelion menunjukkan akurasi sebesar 95,6% terhadap data manual. Persentase ini menunjukkan tingkat kesesuaian yang tinggi, sehingga sistem telah memenuhi kriteria keakuratan yang diharapkan dalam penelitian ini.

Kata kunci : *Prototype*, Sistem Penghitung Otomatis, *IOT*, Jemaat GKE.

ABSTRACT

Imanuel Natalis Timisela, C2155201097, 2025. Prototype of an Automatic Congregation Counting System Based on IoT at GKE Ewanggelion Church, Advisor I: Herkules, S.Kom., M.Cs., Advisor II: Hotmian Sitohang, M.Kom.

The rapid development of technology in various aspects of life has begun to be applied. One of them is in a place of worship for Christians such as GKE Ewanggelion. The calculation of the number of congregants who worship at GKE Ewanggelion is done manually by officers when welcoming the congregation at the entrance. With this manual calculation method, it is possible for errors to occur from officers considering the high concentration of officers between the service of readiness and completeness for worship and counting the number of congregants.

This study aims to make it easier for church officers to focus on the readiness and completeness of the worship that will take place without having to divide the task again to count the number of congregants.

The research method used is R&D (Research and Development) with a prototyping development model type. Data was collected through observation, interviews, and literature studies. The calculation was carried out using Arduino Uno R3 ATMega328P with a PIR (Passive Independent) sensor to detect human movement based on heat changes and an Ultrasonic sensor to detect the distance between the sensor and the object (Human).

The results of the Prototype Calculation of the Automatic Counting System for the Number of Congregation Based on IOT at the GKE Ewanggelion Church showed an accuracy of 95.6% against manual data. This percentage indicates a high level of conformity, so the system has met the accuracy criteria expected in this study.

Keywords : Prototype, Automatic Counter System, IOT, GKE Congregation.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa atas berkatNya yang telah mengasihi dan menguatkan penulis dalam menyelesaikan Tugas Akhir yang berjudul "***PROTOTYPE* SISTEM PENGHITUNG OTOMATIS JUMLAH JEMAAT BERBASIS *IOT* PADA GEREJA GKE EWANGGELION".** Tugas Akhir ini disusun untuk memenuhi syarat dalam memperoleh gelar Sarjana Komputer di Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya.

Pada kesempatan ini, penulis mengungkapkan penghargaan mendalam kepada semua yang telah memberikan kontribusi berharga dalam menyelesaikan penulisan Tugas Akhir ini. Oleh karena itu, penulis ingin menyampaikan terima kasih yang tulus kepada :

1. Herkules, S.Kom., M.Cs., selaku dosen pembimbing I yang memberikan Bimbingan dalam penulisan Tugas Akhir.
2. Hotmian Sitohang, M.Kom., selaku dosen pembimbing II yang memberikan banyak masukan dan saran selama penulisan Tugas Akhir ini.
3. Seluruh jemaat GKE Ewanggelion beserta petugas-petugas gereja baik Pendeta, Vikaris, Diakonia yang telah memberikan bimbingan dan arahan selama Penelitian,

Penulis mengharapkan masukan, saran dan kritik yang bersifat membangun untuk menyempurnakan Tugas Akhir ini, semoga Tugas Akhir ini dapat bermanfaat bagi penulis dan pembaca terutama untuk menambah ilmu pengetahuan.

Palangkaraya, 20 Mei 2025

Penulis

DAFTAR ISI

PERSETUJUAN.....i
PENGESAHAN.....ii
PERNYATAAN KEASLIAN.....iii
MOTTO DAN PERSEMBAHAN.....iv
INTISARI.....v
ABSTRACT.....vi
KATA PENGANTAR.....vii
DAFTAR ISI.....viii
DAFTAR TABEL.....ix
DAFTAR GAMBAR.....x
DAFTAR LAMPIRAN.....xi
BAB I PENDAHULUAN.....1
1.1 Latar Belakang Masalah.....1
1.2 Perumusan Masalah.....3
1.3 Batasan Masalah.....3
1.4 Tujuan dan Manfaat.....3
1.5 Sistematika Penulisan.....4
BAB II LANDASAN TEORI.....6
1.1 Kajian Teori.....6
1.2 Penelitian Yang Relevan.....16
BAB III METODE PENELITIAN.....21
3.1 Tinjauan Umum.....21
3.2 Jenis Penelitian.....21
3.3 Desain Penelitian.....22
3.4 Instrumen Penelitian.....23
3.5 Analisis Kebutuhan.....24
3.6 Desain.....30
BAB IV HASIL DAN PEMBAHASAN.....36
4.1 Hasil.....36
4.2 Pembahasan.....49
BAB V KESIMPULAN DAN SARAN.....54
5.1 Kesimpulan.....54
5.2 Saran.....55
DAFTAR PUSTAKA.....56

DAFTAR TABEL

Tabel 2. 2 Penelitian Yang Relevan 19
Tabel 3. 1 Kebutuhan Perangkat Lunak 25
Tabel 3. 2 Kebutuhan Perangkat Keras 26
Tabel 3. 3 Komponen Alat 26
Tabel 4. 1 Pengujian *Black Box* ……49
Tabel 4. 2 *Progress* Pengujian Minggu ke-1 sampai Minggu ke-4……49

DAFTAR GAMBAR

Gambar 2. 1 Cara kerja *IoT* 7
Gambar 2. 2 Metode *Prototype* 9
Gambar 2. 4 *Arduino Uno* 10
Gambar 2. 5 *Sensor PIR* 11
Gambar 2. 6 *LCD (Liquid Crystal Display)*..... 12
Gambar 2. 7 Alat Penghitung 12
Gambar 2. 8 *Wemos D1 Mini*.....13
Gambar 2. 9 *Sensor Ultrasonic* 14
Gambar 2. 10 *Breadboard*..... 14
Gambar 2. 11 Kabel *Jumper* 15
Gambar 2. 12 *Adaptor* 16
Gambar 3. 1 Diagram Blok Sistem Penghitung Berbasis *IoT* 28
Gambar 3. 2 Alur Flowchart Sistem Penghitung Berbasis *IoT* 31
Gambar 3. 3 Susunan *Hardware* 33
Gambar 3. 4 Desain Antarmuka *Website E-Counter*.....35
Gambar 3. 5 Rancangan Desain Tempat Perangkat *IoT* 36
Gambar 4. 1 Tampilan Depan *Prototype* 37
Gambar 4. 2 Tampilan Samping Kanan *Prototype*.....38
Gambar 4. 3 Tampilan Samping Kiri *Prototype*.....39
Gambar 4. 4 Tampilan Atas *Prototype* 39
Gambar 4. 5 Program Utama *Server Flask*…..…….. 41
Gambar 4. 6 Konfigurasi *Database* 42
Gambar 4. 7 *Routes Monitoring*.....43
Gambar 4. 8 *Routes Websocket*.....44
Gambar 4. 9 *User Interface Web* 45
Gambar 4. 10 Konfigurasi Sensor.....46
Gambar 4. 11 Konfigurasi *Wemos D1 Mini*.....47
Gambar 4. 12 Konfigurasi *Wemos D1 Mini* pada *Arduino IDE* 48
Gambar 4. 13 Penghitungan Minggu ke-1.....50
Gambar 4. 14 Penghitungan Minggu ke-2.....51
Gambar 4. 15 Penghitungan Minggu ke-3.....52
Gambar 4. 16 Penghitungan Minggu ke-4.....53

DAFTAR LAMPIRAN

Lampiran 1. Surat Tugas Pembimbing Tugas Akhir
Lampiran 2. Lembar Konsultasi Bimbingan Tugas Akhir
Lampiran 3. Surat Izin Penelitian
Lampiran 4. Surat Keterangan Telah Melakukan Penelitian
Lampiran 5. Lembar Wawancara
Lampiran 6. Lembar Dokumentasi Wawancara
Lampiran 7. Lembar Kuesioner
Lampiran 8. Surat Tugas Penguji Sidang Tugas Akhir
Lampiran 9. Berita Acara Penilaian Sidang Tugas Akhir
Lampiran 10. Berita Acara Uji Coba Blackbox
Lampiran 11. Lembar Dokumentasi Pengujian
Lampiran 12. Listing Program

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Seiring dengan perkembangan teknologi yang semakin pesat seperti saat ini, khususnya perkembangan teknologi dalam bidang keilmuan elektronika, maka banyak sekali sistem ataupun teknologi yang berhasil diciptakan dan dikembangkan untuk memberikan kemudahan terhadap berbagai pekerjaan manusia.

Sistem otomatisasi adalah suatu sistem yang memungkinkan suatu proses berjalan dengan intervensi manusia yang minimal. Sistem ini sangat efektif diterapkan dalam kegiatan yang bersifat berulang dan membutuhkan tingkat akurasi tinggi, seperti pencatatan kehadiran (Sutanta, 2011).

Gereja menjadi salah satu tempat dengan mobilitas yang tinggi mengingat gereja adalah tempat ibadah bagi umat kristen yang melaksanakan ibadah tiap minggu nya dan ibadah kekristenan lainnya pada waktu-waktu tertentu. Jumlah orang yang beribadah pada suatu gereja pasti nya sangat banyak karena dari berbagai elemen masyarakat baik yang tua maupun yang muda.

Berdasarkan pengamatan penulis, penghitungan jumlah jemaat yang beribadah di GKE Ewanggelion dilakukan secara manual oleh para petugas pada saat pengumpulan persembahan. Dengan cara hitung manual tersebut tidak menutup kemungkinan terjadi kesalahan dari petugas dikarenakan terbaginya konsentrasi petugas antara mengumpulkan persembahan dan menghitung jumlah jemaat. Pengumpulan data jumlah pengunjung atau jemaat tersebut sering digunakan sebagai bahan evaluasi bagi Majelis Gereja untuk memantau perkembangan Jemaat di GKE Ewanggelion agar mengetahui peningkatan jumlah jemaat yang beribadah di GKE Ewanggelion.

Kemajuan teknologi saat ini memungkinkan manusia untuk mengatasi berbagai permasalahan di berbagai bidang kehidupan dengan lebih efisien dan efektif. Salah satu cara yang digunakan adalah melalui penggabungan beberapa teknologi canggih, seperti teknologi sensor dan mikrokontroler.

Pemanfaatan sensor dalam sistem otomatis dapat meminimalkan kesalahan akibat faktor manusia, terutama dalam pengumpulan data yang bersifat kuantitatif seperti jumlah orang atau objek (Suryani, 2014).

Mikrokontroler ini memainkan peran sentral dalam mengolah data dari sensor dan mengontrol berbagai perangkat lain berdasarkan logika yang telah diprogramkan sebelumnya. Seluruh sistem ini dapat bekerja secara otomatis tanpa memerlukan intervensi manusia secara langsung, sehingga meningkatkan efisiensi, mengurangi kesalahan, dan mempercepat proses kerja.

Internet of Things (IoT) adalah paradigma baru dalam teknologi informasi yang bertujuan menghubungkan berbagai objek fisik melalui jaringan internet untuk saling bertukar data dan informasi, serta mampu beroperasi secara otomatis. IoT dinilai mampu meningkatkan efisiensi, akurasi, dan kecepatan dalam pengambilan keputusan berbasis data real-time (Atzori et al, 2010).

Dengan adanya keperluan untuk sistem penghitung yang terkoneksi dengan komputer, agar pemantauan jumlah jemaat menjadi lebih mudah dengan hanya melihat layar *monitor*. Konsep inilah yang mejadikan ide untuk menyusun tugas akhir dengan judul “ *Prototype* Sistem Penghitung Otomatis Jumlah Jemaat Berbasis *IoT* Pada Gereja GKE Ewanggelion”.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang sudah diuraikan di atas maka dapat dirumuskan suatu masalah yaitu "Bagaimana merancang dan membangun *prototype* sistem penghitung otomatis jumlah jemaat untuk mengetahui jumlah mobilitas jemaat dan menghitung jumlah jemaat yang beribadah menggunakan *Internet of Thing* Di Gereja GKE Ewanggelion ?"

1.3 Batasan Masalah

Agar permasalahan dalam pembuatan alat ini tidak melebar luas, maka diberikan batasan-batasan untuk pendekatan permasalahan agar lebih terinci dalam pelaksanaannya, yaitu :

a. Sistem yang dikembangkan merupakan *prototype*, untuk pengujian dilakukan pada Gereja GKE Ewanggelion.
b. Sistem yang dikembangkan menggunakan *Arduino versi "UNO"* yang merupakan *board* mikrokontroler berbasis *ATmega328* (*datasheet*) serta menggunakan *Sensor PIR* (*Passive Infrared Receiver*) dan *Sensor Ultrasonic*.
c. Sistem yang dikembangkan menggunakan *sensor* yang terletak di depan pintu masuk yang dapat mendeteksi orang yang masuk dan keluar.
d. Sistem menampilkan hasil hitung jumlah pengunjung pada layar *LCD* (*Liquid Crystal Display*) dan pada layar *monitor/desktop* kemudian data dikirimkan ke *server* untuk penyimpanan atau pemrosesan lebih lanjut.

1.4 Tujuan Dan Manfaat

a. Tujuan Penelitian

Tujuan penelitian ini adalah untuk merancang dan membangun *Prototype* Sistem Penghitung Otomatis Jumlah Jemaat Berbasis *IoT* pada Gereja GKE Ewanggelion.

b. Manfaat Penelitian

1) Bagi Penulis yaitu untuk menerapkan pengalaman dan pengetahuan serta mengembangkan *Prototype* Sistem Penghitung Otomatis Jumlah Jemaat Berbasis *IoT* pada Gereja GKE Ewanggelion.
2) Bagi Gereja GKE Ewanggelion yaitu memudahkan petugas ibadah dalam menghitung jumlah orang/jemaat yang beribadah tanpa mengganggu konsentrasi saat ibadah dan dapat lebih fokus untuk beribadah.
3) Bagi STMIK Palangkaraya yaitu menambah literatur pada perpustakaan STMIK Palangkaraya yang dapat digunakan sebagai perbandingan untuk pengembangan alat yang lebih baik.

1.5 Sistematika Penulisan

Berikut ini merupakan sistematika penulisan laporan tugas akhir yaitu :

BAB I PENDAHULUAN

Bab ini berisikan latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat, serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisikan tentang teori-teori dan memaparkan ringkasan hasil penelitian yang relevan dan mendukung judul serta mendasari pembahasan secara detail.

BAB III METODE PENELITIAN

Bab ini berisikan tentang tahapan yang dilakukan peneliti dalam mengumpulkan informasi atau data yang dibutuhkan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisikan tentang hasil dari penelitian dan pembahasan dari hasil yang didapat.

BAB V KESIMPULAN DAN SARAN

Bab ini berisikan tentang kesimpulan dan saran yang diberikan untuk pengembangan selanjut

BAB II

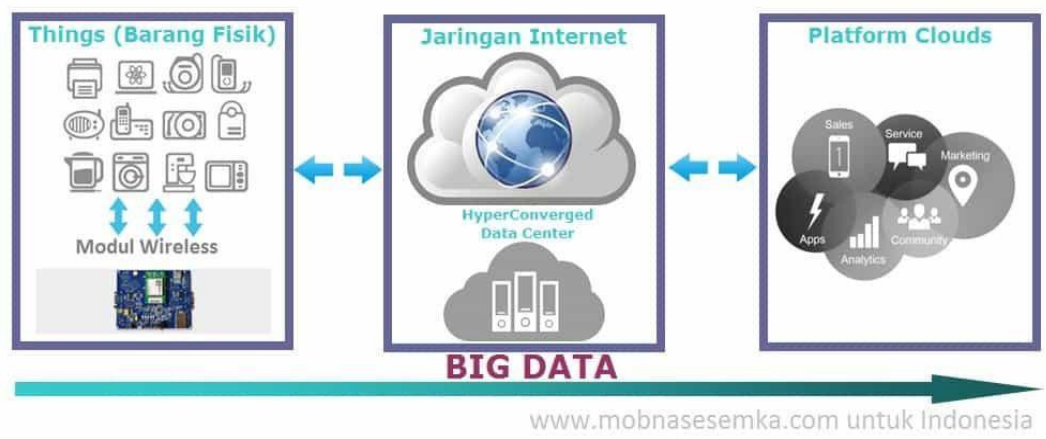
TINJAUAN PUSTAKA

2.1 Kajian Teori

2.1.1 *IoT* (*Internet of Things*)

Internet Of Things adalah sebuah teknologi canggih yang pada dasarnya merujuk pada banyaknya *device* dan suatu sistem di seluruh dunia yang saling terhubung satu sama lain dengan menggunakan *internet* dan bisa saling berbagi data, teknologi – teknologi ini memiliki seperti *sensor* dan *software* dengan tujuan untuk berkomunikasi, mengendalikan, menghubungkan, dan bertukar data melalui perangkat lain selama masih terhubung dengan *internet* dan mendukung kinerja tanpa menggunakan bantuan kabel dan berbasis *wireless IoT* memiliki hubungan yang erat dengan istilah *machine-to machine* atau *M2M* (Selay et al., 2022).

Seluruh alat yang memiliki kemampuan komunikasi *M2M* ini sering disebut dengan perangkat cerdas atau *smart devices*. Perangkat cerdas ini diharapkan dapat membantu kerja manusia dalam menyelesaikan berbagai urusan atau tugas yang ada. Penelitian ini bertujuan untuk mengetahui (1) Pengertian *IoT*; (2) Cara kerja *IoT*; (3) Manfaat *IoT* (4) Contoh *IoT* dalam kehidupan sehari hari; ; (5) Sejarah *IoT*. Metode dasar yang digunakan dalam penelitian jurnal ini, yaitu dari referensi yang ada, penelitian dari artikel - artikel. Hasil penelitian menunjukan bahwa, *Internet Of Things* adalah suatu Teknologi yang canggih menggunakan *device* atau perangkat yang sangat cerdas dan mampu membantu manusia dalam kehidupan sehari hari (Selay et al., 2022).



Gambar 2. 1 Cara kerja *IoT*

(Sumber : Selay et al., 2022)

2.1.2 Metode *Prototype*

Metode *prototype* merupakan suatu teknik untuk mengumpulkan informasi tertentu mengenai kebutuhan informasi pengguna secara cepat. Berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai, *prototype* tersebut akan dievaluasi oleh pelanggan/pemakai dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak (Dara Andariana, 2019). Ada 4 metodologi *prototype* yang paling utama yaitu :

1. *Illustrative,* menghasilkan contoh laporan dan tampilan layar.
2. *Simulated,* mensimulasikan beberapa alur kerja sistem tetapi tidak menggunakan data *real*.
3. *Functional,* mensimulasikan beberapa alur sistem yang sebenarnya dan menggunakan data *real*.
4. *Evolutionary*, menghasilkan model yang menjadi bagian dari operasional *system*.

Fase fase dalam *Prototyping* adalah sebagai berikut :

a. Analisa Kebutuhan

   Di tahap ini pengembang melakukan identifikasi *software* dan semua kebutuhan sistem yang akan dibuat.

b. Membangun *Prototyping*

   Membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penghitungan terhadap jemaat.

c. Evaluasi *Prototyping*

   Evaluasi ini dilakukan untuk mengetahui apakah *prototyping* sudah sesuai dengan harapan pengguna.

d. Mengkodekan Sistem

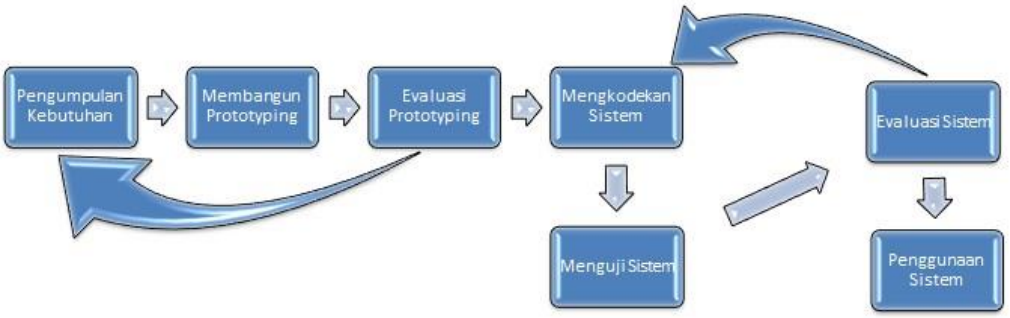
   Pada tahap ini *prototyping* yang sudah disetujui akan diubah ke dalam bahasa pemrograman.

e. Menguji Sistem

   Di tahap ini dilakukan untuk menguji sistem perangkat lunak yang sudah dibuat pengujian.

f. Evaluasi Sistem

   Perangkat lunak yang sudah siap jadi akan dievaluasi oleh pelanggan untuk mengetahui apakah sistem sesuai dengan yang diharapkan.



Gambar 2. 2 Metode *Prototype*
(Sumber : Dara Andariana, 2019)

2.1.3 *Arduino IDE*

Arduino dikatakan sebagai sebuah *platform* dari *physical computing* yang bersifat *open source*. Pertama-tama perlu dipahami bahwa kata "*platform*" di sini adalah sebuah pilihan kata yang tepat. *Arduino* tidak hanya sekedar sebuah alat pengembangan, tetapi ia adalah kombinasi dari *hardware*, bahasa pemrograman dan *Integrated Development Environment (IDE)* yang canggih.

IDE adalah sebuah *software* yang sangat berperan untuk menulis program, meng-*compile* menjadi kode *biner* dan meng-*upload* ke dalam *memory microcontroller*. Ada banyak projek dan alat-alat dikembangkan oleh akademisi dan profesional dengan menggunakan *Arduino*, selain itu juga ada banyak modul- modul pendukung (*sensor*, tampilan, penggerak dan sebagainya) yang dibuat oleh pihak lain untuk bisa disambungkan dengan *Arduino*. *Arduino* berevolusi menjadi sebuah *platform* karena ia menjadi pilihan dan acuan bagi banyak praktisi (Feri Djuandi, 2020).

2.1.4 *Arduino Uno*

Arduino Uno adalah papan sirkuit berbasis mikrokontroler *Atmega328p*. *IC (intergrated circuit)* ini memiliki 14 *input/output* digital (6 *output* untuk *PWM*), 6 *analog input*, resonator Kristal keramik *16MHz*, koneksi *USB*,soket *adaptor*, *pin header ICSP*, dan tombol *reset*. Hal inilah yang dibutuhkan untuk *mensupport* mikrokontroler secara mudah terhubung dengan kabel *power USB* atau kabel *power supply adaptor AC* ke *DC* atau juga *battery*.

Uno berbeda dari semua *board* mikrokontrol diawal-awal yang tidak menggunakan *chip* khusus *driver FTDI USB-to-serial*. Sebagai penggantinya penerapan *USB-to-serial* adalah *ATmega16U2* versi *R2* (versi sebelumnya *ATmega8U2*). Versi *Arduino Uno Rev.2* dilengkapi resistor ke *8U2* ke garis *ground* yang lebih mudah diberikan ke mode *DFU* (Eka & Susi, 2019).



Gambar 2. 4 *Arduino Uno*

(Sumber : Eka & Susi, 2019)

2.1.5 *Sensor PIR*

Sensor Passive Infrared Receiver (PIR), sensor ini merupakan *sensor* berbasis *infrared* namun tidak sama dengan *IR LED* dan *fototransistor*. *Sensor PIR* merespon energi dari pancaran *infrared* pasif yang dimiliki oleh setiap benda yang terdeteksi olehnya. Salah satu benda yang memiliki pancaran *infrared* pasif adalah tubuh manusia. Energi panas yang dipancarkan oleh benda dengan suhu diatas nol mutlak akan dapat ditangkap oleh sensor tersebut. Bagian-bagian dari *PIR* adalah *Fresnel Lens, IR Filter, Pyroelectric sensor, amplifier,* dan *comparator* (Desyantoro et al., 2021).



Gambar 2. 5 *Sensor PIR*

(Sumber : Desyantoro et al., 2021)

2.1.6 *LCD (Liquid Crystal Display)*

LCD (Liquid Crystal Display) adalah salah satu jenis *display* elektronik yang dibuat dengan teknologi *CMOS logic* yang bekerja dengan tidak menghasilkan cahaya, tetapi memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. *LCD (Liquid Crystal Display)* berfungsi sebagai penampil data, baik dalam bentuk karakter, huruf, angka ataupun grafik (Bawotong, 2019).



Gambar 2. 6 LCD (Liquid Crystal Display)
(Sumber : Bawotong, 2019)

2.1.7 Alat Penghitung

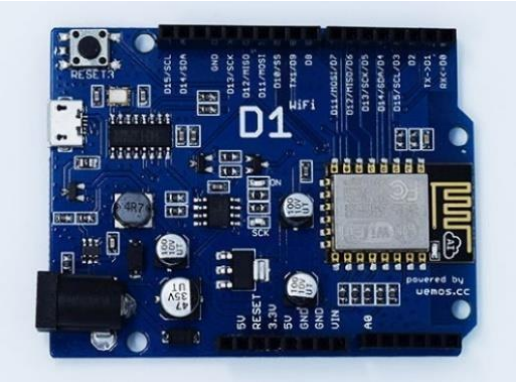
Alat bantu hitung atau biasa di sebut *tally counter* ini merupakan salah satu alat yang di butuhkan untuk menghitung objek. Misalnya saja yakni untuk menghitung kendaraan yang ada di jalan raya. Selain itu juga bisa di pakai untuk menghitung barang atau stok produk di dalam sebuah perindustrian. Alat bantu hitung ini sangat lah bermanfaat sekali dalam hal menghitung stok barang dalam perindustrian. Yang mana cara menggunakannya hanya di tekan berkali-kali untuk menandakan Anda sudah menghitungnya (Indriani et al., 2024).



Gambar 2. 7 Alat Penghitung
(Sumber : Indriani et al., 2024)

2.1.8 *Wemos D1*

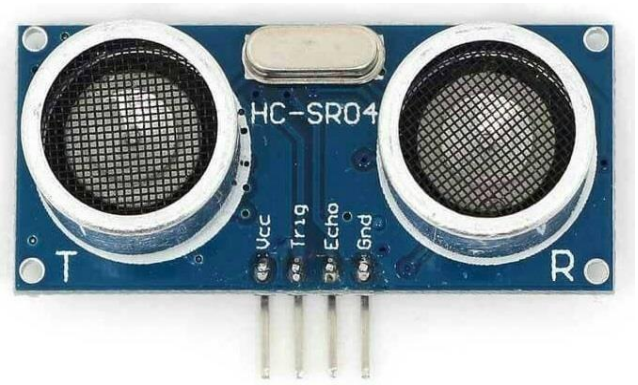
Mikrokontroler ini berbasis *ESP8266* yaitu sebuah modul *mikrokontroler* nirkabel (*Wifi*) *802.11* yang kompatibel dengan *Arduino IDE*. Tata letak mikrokontroler ini didasarkan pada desain *hardware Arduino* standar dengan proporsi yang sama dengan *Arduino Uno* dan *Leonardo*. Mikrokontroler ini juga sudah termasuk satu *set header Arduino* standar yang artinya kompatibel dengan beragam *Arduino shield*., sudah mencakup sebuah *CH340 USB to serial interface* seperti kabel *USB micro* yang umum digunakan (Widya & Syaputra, 2021).



Gambar 2. 8 Wemos D1
(Sumber : Widya & Syaputra, 2021).

2.1.9 *Sensor Ultrasonik*

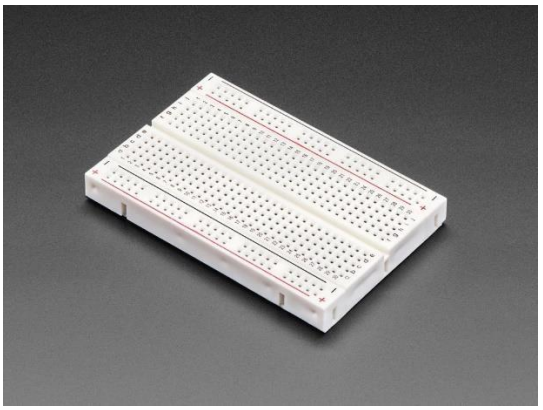
Sensor ultrasonik adalah *sensor* yang bekerja berdasarkan prinsip pantulan gelombang suara dan digunakan untuk mendeteksi keberadaan suatu objek atau benda tertentu didepan frekuensi kerja pada daerah diatas gelombang suara dari *20 kHz* hingga *2 MHz*. *Sensor ultrasonik* terdiri dari dari dua unit, yaitu unit pemancar dan unit penerima struktur unit pemancar dan penerima (Arsada, 2023).



Gambar 2. 9 Sensor Ultrasonik
(Sumber : Arsada, 2023)

2.1.10 *BreadBoard*

Breadboard adalah sebuah papan yang digunakan untuk membantu proses perangkaian *prototipe* elektronik tanpa harus menyolder komponen komponen tersebut (Tantowi & Yusuf, 2020).



Gambar 2. 10 *Breadboard*
(Sumber : Tantowi & Yusuf, 2020)

2.1.11 Kabel *Jumper*

Kabel *Jumper* merupakan kabel elektrik yang berfungsi untuk menghubungkan antar komponen yang ada di *breadboard* atau papan *arduino* tanpa harus menggunakan *solder*. Umumnya memang kabel *Jumper* sudah dilengkapi dengan *pin* yang terdapat pada setiap ujungnya (Tantowi & Yusuf, 2020).



Gambar 2. 11 Kabel *Jumper*
(Sumber : Tantowi & Yusuf, 2020)

2.1.12 *Adaptor*

Adaptor digunakan sebagai sumber tegangan untuk mengaktifkan *Arduino UNO*, tegangan yang dibutuhkan *Arduino UNO* sebesar *5-7V DC*. Pada pengujian ini *input adaptor* dihubungkan ke tegangan listrik *220V AC* dan *output* diukur menggunakan *multimeter*. Hasil pengukuran *adaptor* menggunakan *multimeter*, *output* tegangan *adaptor* sebesar *6V*. Output tegangan *adaptor* sesuai dengan kebutuhan *Arduino UNO*, dimana tegangan yang dibutuhkan sebesar *5-7V* (Amsar et al., 2020).



Gambar 2. 12 *Adaptor*
(Sumber : Amsar et al., 2020)

2.1.13 *WebSocket*

Websocket adalah standar baru untuk komunikasi *full-duplex* (dua arah secara bersamaan) sehingga komunikasi yang terjadi antara *client* dan *server* lebih *real-time* (Maulana & Rahmatulloh, 2019).

2.2 Penelitian yang Relevan

Dalam suatu penelitian diperlukan dukungan dari hasil beberapa penelitian yang relevan yang telah ada sebelumnya dan berkaitan dengan penelitian yang sudah diteliti. Penulis akan menguraikan dan menjelaskan secara singkat beberapa penelitian yang relevan yang memiliki topik serupa atau mendekati persamaan yang digunakan sebagai acuan penelitian yaitu :

a. Penelitian pertama yang dijadikan sebagai acuan yaitu penelitian yang membahas rancang bangun sistem penghitung jumlah orang atau pengunjung berbasis Arduino “uno” menggunakan sensor PIR (*Passive Infrared Receiver*) dan sensor ultrasonik. Perancangan alat ini yaitu untuk menghitung jumlah pengunjung yang masuk pada pintu. (Angga Aditia, 2020).
b. Penelitian kedua yaitu Rancang Bangun Prototype alat Penghitung Jumlah Orang Dalam Antrian Berbasis ESP8266. Cara kerja sistem ini adalah menghitung jumlah orang yang masuk dan keluar pada sebuah ruangan

menggunakan sensor infrared dengan mikrokontroler ESP8266 sebagai pengendalinya (Wahyuni et al., 2022).

c. Penelitian ketiga yaitu Sistem Penghitung Jumlah Orang Pada Pintu Masuk Otomatis. Tujuan dalam pembuatan penelitian ini adalah untuk membatasi jumlah orang pada suatu ruangan dengan menghitung jumlah orang yang masuk dan keluar dari ruangan tersebut (Putri Alit Widyastuti Santiary & I Ketut Swardika, 2023).

d. Penelitian ke empat yaitu Prototype Penghitung Pengunjung Otomatis Pada Klinik Idaman As'adiyah Berbasis Arduino Uno dan Sensor PIR. Penelitian ini bertujuan untuk merancang sebuah prototipe yang mampu mengatasi masalah, khususnya mengenai sistem penghitung pengunjung otomatis di Klinik Idaman As'adiyah. Prototipe ini diharapkan dapat menjadi solusi untuk mengatasi masalah penghitungan pengunjung, sehingga menjadi lebih efektif dan efisien untuk memonitor pengunjung klinik secara otomatis dengan menggunakan sensor Passive Infrared Receiver (PIR) dan Arduino Uno (Dayera, Musa Bundaris Palungan, 2024).

e. Penelitian ke lima yaitu Rancang Bangun Prototype Sistem Pencatatan Pengunjung Pada Stadion Sepak Bola Menggunakan Sensor PIR (Passive Infra Red). Pemanfaatan sensor PIR dalam sebuah pertandingan sepak bola dapat digunakan sebagai penghitung pengunjung yang telah memasukki stadion, sehingga panitia dapat mengetahui jumlah pengunjung tanpa harus menghitung jumlah tiket yang telah terjual (Anggara Ady Prasetya, 2019).

Melalui ringkasan di atas, penelitian dengan judul *prototype* sistem penghitung otomatis jumlah jemaat berbasis *IoT* pada Gereja GKE Ewanggelion memiliki perbedaan dengan penelitian yang sebelumnya. Berikut tabel penelitian yang relevan sebagai bahan perbandingan hasil penelitian dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Penelitian Yang Relevan

No	Peneliti/ Penulis	Topik Penelitian	Metode Penelitian	Pembahasan	Hasil Penelitian
1	Angga Aditia /2020	Merancang dan membangun Sistem Penghitung Jumlah Pengunjung Berbasis Arduino Uno menggunakan Sensor PIR (Passive Infrared Receiver) dan Sensor Ultrasonic.	Studi Literature	Prinsip kerja alat sensor yang dipasang pada pintu masuk akan mendeteksi apakah yang melalui sensor tersebut orang atau benda mati. Sensor PIR ini bekerja dengan menangkap energi panas yang dihasilkan dari pancaran sinar inframerah pasif yang dimiliki setiap benda dengan suhu benda.	Pembuatan Rancangan alat menggunakan model prototype dipilih karena dipandang cocok dan efektif dalam menggunakan rancangan pembuatan alat.
2	Rindu Dwi Wahyuni Dkk /2022	Membangun sistem penghitungan jumlah orang menggunakan teknologi IoT untuk menyelesaikan permasalahan, bagaimana agar proses monitoring penghitungan jumlah orang dalam antrian	Prototype Dengan pendekatan Unified Modelling Languange (UML)	Cara kerja dari alat penghitung jumlah orang yaitu dengan cara sensor-sensor yang telah terpasang menghitung setiap objek yang melewati sensor tersebut. Data dari sensor tersebut diterima oleh	Proses pengitungan jumlah orang dalam antrian dari manual menjadi dilakukan secara otomatis dan dapat dimonitor secara real time menjadi lebih efisien.

No	Peneliti/ Penulis	Topik Penelitian	Metode Penelitian	Pembahasan	Hasil Penelitian
		dapat dilakukan secara otomatis dan efisien dan bagaimana melakukan pemesanan nomer antrian dapat dilakukan secara online. membangun sistem penghitungan jumlah orang menggunakan teknologi IoT untuk menyelesaikan permasalahan, bagaimana agar proses monitoring penghitungan jumlah orang dalam antrian dapat dilakukan secara online.		mikrokontroler ESP8266 dan dikirimkan ke platform IoT Firebase dan platform Android untuk ditampilkan di user interface.	
3	Putri Alit Widyastuti Santiary Dkk /2023	Sistem Penghitung Jumlah Orang Pada Pintu Masuk Otomatis. .	Studi Literature	Pada penelitian ini dirancang sebuah sistem penghitung jumlah orang pada pintu masuk otomatis dengan perangkat kontrolnya menggunakan arduino uno.	Sistem penghitung jumlah orang pada pintu otomatis dapat bekerja dengan baik sesuai dengan rancangan yang telah dibuat.

No	Peneliti/ Penulis	Topik Penelitian	Metode Penelitian	Pembahasan	Hasil Penelitian
4	Fachtur Miftawan Cahya dkk /2024	Mengatasi masalah penghitungan pengunjung, sehingga menjadi lebih efektif dan efisien untuk memonitor pengunjung klinik secara otomatis dengan menggunakan sensor Passive Infrared Receiver (PIR) dan Arduino Uno.	Prototype Dengan pendekatan Working Model	Diharapkan dapat meningkatkan akurasi dalam mencatat jumlah pengunjung dan menghemat waktu petugas dengan secara otomatis mencatat data pengunjung dan membantu petugas untuk fokus pada tugas lainnya.	Alat ini dapat Bekerja dengan baik Ketika terhubung Dengan sumber Daya listrik dan Menampilkan hasil Penghitungan Otomatis melalui layer monitor yang dapat dilihat secara langsung.
5	Anggara Adi Prasetya Dkk /2019	Merancang sebuah Prototype sistem pencatatan Pengunjung pada stadion Sepak bola menggunakan Sensor PIR (Passive Infrared Receiver).	Studi Literature	Menghitung jumlah pengunjung stadion dan mengatur pintu stadion tertutup otomatis Ketika di stadion sudah memenuhi kapasitas stadion dengan menggunakan sensor PIR.	Telah dirancang dan dibuat prototype system pencatatan pengunjung pada stadion sepak bola menggunakan sensor PIR (Passive Infrared Receiver) dan data jumlah pengunjung yang didapatkan lebih akurat.

BAB III

METODOLOGI PENELITIAN

3.1 Tinjauan Umum

Gereja Kalimantan Evangelis (GKE) Ewanggelion di Palangka Raya, Kalimantan Tengah, mengalami tantangan dalam proses penghitungan jemaat yang menghadiri kebaktian. Selama ini, penghitungan dilakukan secara manual oleh petugas gereja, yang sering kali menimbulkan ketidakakuratan karena konsentrasi petugas terbagi antara menghitung jemaat dan mengumpulkan persembahan. Selain itu, hasil penghitungan ini sangat penting bagi majelis gereja untuk menganalisis perkembangan jumlah jemaat yang hadir.

3.2 Jenis Penelitian

Jenis penelitian yang digunakan adalah *R&D* (*Research and Development*) dengan jenis model pengembangan *prototyping*. Dengan tahapan aktivitas yang dilakukan yaitu analisis kebutuhan, desain, perancangan *prototype*, uji coba, serta tahap validasi sistem penghitung otomatis jumlah jemaat berbasis *IoT* pada GKE Ewanggelion. Hasil yang diharapkan pada penelitian ini berupa sistem perhitungan jumlah pengunjung yang tepat dan akurat sehingga tidak menyulitkan petugas gereja dalam melaksanakan tugas lainnya saat ibadah berlangsung.

3.3 Desain Penelitian

Pada penelitian ini, penulis menggunakan desain penelitian *Prototyping*. Sebuah *prototype* adalah versi awal dari sistem perangkat lunak yang digunakan untuk mendemonstrasikan konsep-konsep, percobaan rancangan, dan menemukan lebih banyak masalah dan solusi yang memungkinkan (Nugraha & Syarif, 2018).

Berikut tahapan yang peneliti lakukan :

a. Analisa Kebutuhan

   Tahapan awal dari pemodelan software yang digunakan guna mengidentifikasi permasalahan-permasalahan yang ada, serta informasi-informasi lain yang diperlukan untuk membangun sistem pada gereja GKE Ewanggelion.

b. Membangun *Prototyping*

   Tahapan ini dikerjakan dengan kegiatan penentuan sumberdaya, spesifikasi untuk pengembangan alat berdasarkan kebutuhan sistem,dan tujuan berdasarkan pada hasil komunikasi yang dilakukan agar pengembangan dapat sesuai dengan yang diharapkan.

c. Evaluasi *Prototyping*

   Tahapan selanjutnya ialah representasi atau menggambarkan model sistem yang akan dikembangkan seperti proses dengan perancangan menggunakan Diagram blok . Dalam tahap ini, *Prototype* yang dibangun dengan sistem rancangan sementara kemudian dievaluasi apakah sudah sesuai dengan yang diinginkan atau masih perlu untuk dievaluasi kembali.

d. Mengkodekan Sistem

Tahapan ini digunakan untuk membangun dan menjalankan aplikasi yang digunakan sebagai acuan berjalannya sistem.

e. Menguji Sistem

Tahapan ini menggunakan metode kuantitatif yang berfokus pada pengujian sistem yang dilakukan sebanyak 4 kali, terutama dalam mengukur efektifitas, akurasi, dan efisiensi dari sistem yang dikembangkan.

f. Evaluasi Sistem

Tahapan ini memakai kuesioner untuk mendapatkan feedback dari pengguna, sebagai hasil evaluasi dari tahapan sebelumnya dan implementasi dari sistem yang dikembangkan.

3.4 Instrumen Penelitian

Teknik pengumpulan data dalam kegiatan penelitian ini dilakukan dengan beberapa tahap sebagai berikut :

a. Observasi

Pengumpulan data dengan melakukan pengamatan langsung terhadap orang-orang yang beribadah di gereja GKE Ewanggelion JL. Raden Saleh III No.24, sehingga diperoleh data yang lengkap dan akurat untuk kebutuhan sistem yang akan dibangun.

b. Wawancara

Pengumpulan data dengan cara melakukan komunikasi dan wawancara secara langsung dengan pihak-pihak terkait yaitu , Pdt. Sri Winarti, M.Div.

gereja GKE Ewanggelion agar sistem yang dibangun sesuai dan tepat dengan kebutuhan gereja GKE Ewanggelion.

c. Studi Pustaka

Pengumpulan data dengan menggunakan atau mengumpulkan sumber-sumber tertulis, dengan cara membaca, mempelajari dan mencatat hal- hal penting yang berhubungan dengan masalah yang sedang dibahas guna memperoleh gambaran secara teoritis.

3.5 Analisis Kebutuhan

Pada penelitian ini penulis memerlukan beberapa perangkat lunak dalam pembuatan alat, Berikut kebutuhan perangkat lunak dapat dilihat pada Tabel 3.1

Tabel 3. 1 Kebutuhan Perangkat Lunak

NO	Perangkat Lunak	Spesifikasi
1	*Windows 11 Home Single Language*	Sebagai sistem operasi untuk membuat alat
2	*Arduino IDE 2.3.4*	Sebagai pemrograman alat dan sistem.
3	*Xampp V3.3.0*	Sebagai penampil sekaligus penyimpan data pada database.
4	*Fritzing 0.9.3*	Sebagai perancang desain alat

Pada penelitian ini, penulis juga menggunakan perangkat keras komputer dalam pembuatan alat. Berikut kebutuhan perangkat keras dapat dilihat pada Tabel 3.2

Tabel 3. 2 Kebutuhan Perangkat Keras

Perangkat Keras	Spesifikasi
Type	*ASUS X441M*
Processor	*Intel(R) Celeron(R) N4000 CPU @ 1.10GHz 1.10 GHz*
Memory	*4,00 GB*
HDD	*1 TB*

Berikut merupakan komponen alat yang digunakan dapat dilihat pada Tabel 3.3

Tabel 3. 3 Komponen Alat

Nama Komponen	Satuan	Harga Satuan	Jumlah Harga
Arduino Uno R3	1 Buah	Rp. 100.000	Rp. 100.000
Breadboard	1 Buah	Rp. 10.000	Rp. 10.000
Kabel Jumper	2 Buah	Rp. 10.000	Rp. 20.000
Wemos D1 Mini	1 Buah	Rp. 25.000	Rp. 25.000
Sensor PIR HC-SR501	2 Buah	Rp. 10.000	Rp. 20.000
Sensor Ultrasonik HC-SR04	2 Buah	Rp. 10.000	Rp. 20.000
Adaptor 5V 1A	1 Buah	Rp. 15.000	Rp. 15.000
LCD I2C Module 20x4	1 Buah	Rp. 55.000	Rp. 55.000
AMS1117-3.3 Linear Voltage Regulators	2 Buah	Rp. 10.000	Rp. 20.000
Total Harga			**Rp. 285.000**

3.5.1. Analisis Data

Analisis sistem merupakan penguraian dari suatu sistem yang utuh ke dalam bagian- bagian komponennya untuk mengidentifikasi dan mengevaluasi permasalahan. Pada bagian analisis ini terdiri dari analisis sistem lama dan analisis sistem baru.

a. Analisis Sistem Lama

   Untuk melakukan penghitungan jumlah jemaat masih dilakukan secara manual dengan menggunakan alat hitung manual atau *Hand Tally Counter* sembari pengumpulan persembahan.

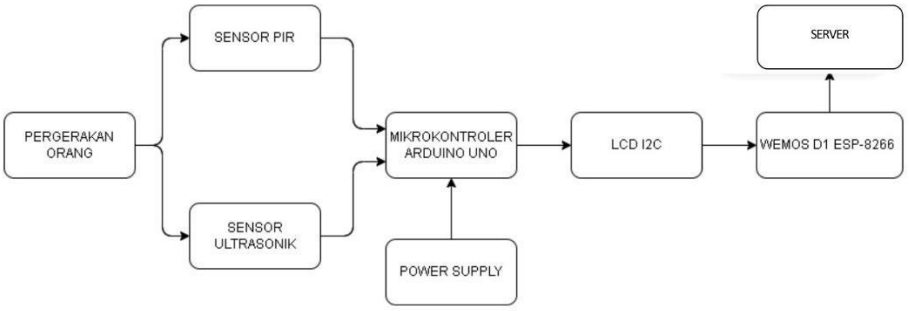
b. Analisis Sistem Baru

   Untuk melakukan penghitungan jumlah jemaat dilakukan dengan sistem penghitung otomatis yang menggunakan 2 *sensor* untuk mendeteksi pengunjung yang masuk maupun yang keluar.

3.5.2. Analisis Proses

Pada penelitian ini metode yang penulis terapkan adalah diagram blok. berikut merupakan tahapan diagram blok.

a. Tahapan atau proses memperoleh data-data yang digunakan untuk kebutuhan penelitian, untuk memudahkan penelitian dan pembaca memahami penelitian, maka dibuat proses bentuk blok diagram.
b. Berikut ini fungsi dari masing-masing blok perancangan (diagram) :



Gambar 3. 1 Diagram Blok Sistem Penghitung Berbasis *IoT*

Diagram blok ini sudah cukup sesuai untuk sistem *IoT* penghitung orang keluar-masuk. Berikut adalah tinjauan masing-masing komponen dan alur data :

1). Pergerakan Orang : Representasi dari objek (manusia) yang bergerak di dalam area yang dipantau oleh sistem.Pergerakan ini menjadi input utama yang memicu kerja sistem.

2). *Sensor PIR* (*Passive Infrared*) : Digunakan untuk mendeteksi adanya pergerakan manusia berdasarkan perubahan panas (inframerah) di lingkungan. Memberikan sinyal ke mikrokontroler *Arduino Uno* untuk diproses lebih lanjut.

3). *Sensor Ultrasonik* : Berfungsi untuk mendeteksi jarak antara *sensor* dengan objek yang berada di depannya. Berguna untuk memastikan apakah ada objek (manusia) yang mendekat atau menjauh.

1). Mikrokontroler *Arduino Uno* : Pusat pengolahan data dari *sensor PIR* dan *sensor ultrasonik*. Bertanggung jawab untuk mengintegrasikan data, menjalankan logika sistem, dan mengontrol perangkat output seperti *LCD*.
2). *Power Supply* : Menyediakan daya untuk semua komponen dalam sistem, termasuk mikrokontroler, *sensor*, dan modul lainnya. Sangat penting untuk menjaga stabilitas sistem agar tetap berfungsi dengan baik.
3). *LCD I2C* : Digunakan untuk menampilkan informasi langsung di lokasi, seperti jumlah orang yang masuk, keluar, atau informasi lain yang relevan.
4). *Wemos D1 ESP-8266* : Modul komunikasi yang digunakan untuk mengirimkan data dari sistem ke *server* melalui jaringan *Wi-Fi*. Menjembatani komunikasi antara mikrokontroler dan *server* untuk penyimpanan data secara *real-time*.
5). *Server* : Menyimpan data yang dikirim oleh *Wemos D1 ESP-8266* untuk analisis dan pemantauan lebih lanjut. *Server* dapat diakses untuk melihat laporan atau status sistem.

3.5.3. Analisis Kelemahan

Ada beberapa kelemahan yang terdapat dalam *prototype* sistem penghitung otomatis jumlah jemaat berbasis *IoT* pada Gereja GKE Ewanggelion yaitu:

a. Ketergantungan pada Koneksi Jaringan

   Jika koneksi antara perangkat *IoT* dan *server* (*WebSocket*) terputus, data tidak akan terkirim, dan sistem gagal mencatat aktivitas. Latensi atau

keterlambatan dalam pengiriman data dapat menyebabkan ketidaksesuaian data *real-time* di *User Interface (UI).*

b. Kelemahan Perangkat Keras

a) *Sensor PIR* :

Rentan terhadap *false positives* : *PIR* dapat mendeteksi pergerakan yang bukan manusia, seperti hewan kecil atau perubahan suhu yang tiba-tiba.

Keterbatasan sudut deteksi : *PIR* memiliki sudut deteksi tertentu (biasanya 90-120 derajat). Jika seseorang melewati area di luar sudut ini, deteksi mungkin gagal.

b) *Sensor Ultrasonik* :

Keterbatasan jarak deteksi : *Ultrasonik* biasanya efektif dalam jarak 2 cm hingga 4 meter. Jika objek berada di luar rentang ini, sensor tidak akan mendeteksi.

Kesalahan pengukuran pada permukaan tertentu : Permukaan yang menyerap atau menyebarkan gelombang *ultrasonik* (misalnya kain atau permukaan yang tidak rata) dapat menyebabkan data jarak yang tidak akurat.

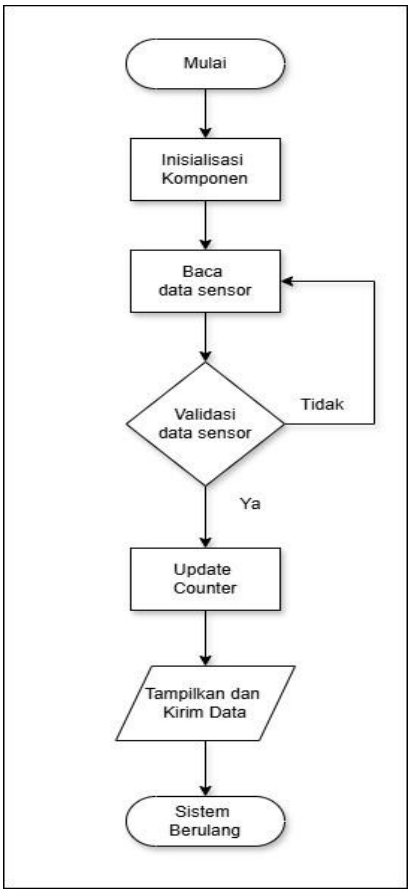
Pengaruh lingkungan : Suara keras atau getaran di sekitar dapat mengganggu pembacaan *ultrasonik*.

3.6 Desain

3.6.1. Desain Proses

a. Flowchart

Flowchart mengambarkan alur kerja dari desain proses cara kerja alat ini. Berikut merupakan *flowchart* yang di gunakan dalam penelitian ini:



Gambar 3. 2 Alur *Flowchart* Sistem Penghitung Berbasis *IoT*

Flowchart ini menggambarkan alur kerja sistem penghtung otomatis yang membaca dan menganalisis data dari *sensor*, kemudian mengirim data ke *server* dan menampilkannya. Langkah-langkah alur flowchart penelitian adalah sebagai berikut :

1) Mulai : Sistem dihidupkan, dan proses penghitungan jumlah orang dimulai.
2) Inisialisasi Komponen : Semua komponen sistem di inisialisasi, termasuk PIR Sensor HC-SR501 untuk mendeteksi gerakan,Ultrasonic HC-SR04 untuk memastikan jarak orang, LCD 20x4 I2C untuk menampilkan jumlah orang masuk dan keluar, dan Wemos D1 untuk mengirim data ke server.
3) Baca Data *Sensor* : Sistem membaca data dari *sensor PIR* dan memberikan sinyal jika ada gerakan. *Ultrasonic Sensor* memastikan jarak orang berada dalam rentang yang diharapkan.
4) Validasi Data *Sensor* : Data dari *sensor* divalidasi untuk menentukan apakah seseorang benar-benar terdeteksi. Jika terdeteksi, keterangan yang muncul yaitu Ya dan lanjut ke langkah berikutnya. Jika tidak terdeteksi, keterangan yang muncul yaitu Tidak. Kemudian sistem kembali membaca data *sensor*.
5) *Update Counter* : Jumlah orang yang masuk atau keluar diperbarui berdasarkan validasi data :

   Counter bertambah jika ada orang masuk.

   Counter berkurang jika ada orang keluar.

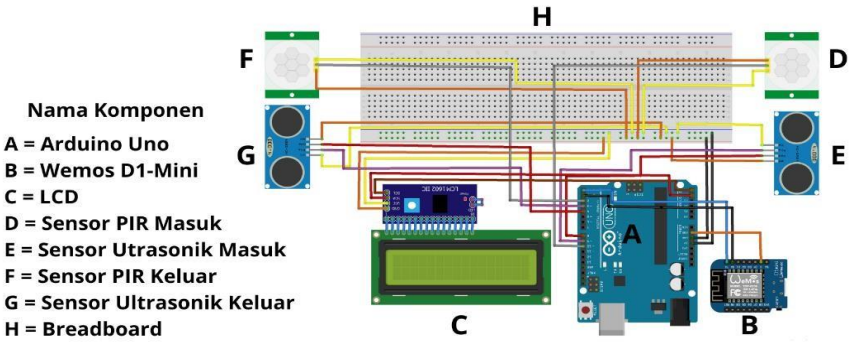
6) Tampilkan dan Kirim Data

LCD 20x4 I2C menampilkan jumlah orang masuk dan keluar.

Wemos D1 mengirim data ke *server* untuk disimpan atau ditampilkan dalam website.

7) Sistem Berulang : Sistem kembali ke langkah "Baca Data *Sensor*" untuk terus memproses data secara *real-time* hingga sistem dimatikan.

Pada tampilan selanjutnya, akan ditampilkan susunan hardware pada gambar 3.3.



Gambar 3. 3 Susunan *Hardware*

1) Komponen yang Terlibat:

PIR Sensor HC-SR501: Deteksi gerakan manusia.

Ultrasonic HC-SR04: Validasi jarak.

Arduino Uno: Kontrol utama sistem.

LCD 20x4 I2C: Menampilkan data jumlah orang.

Wemos D1: Mengirim data ke server melalui *Wi-Fi*.

b. Perancangan *Hardware*

1) *Breadboard* : berfungsi sebagai konduktor listrik tempat melekatkan kabel *jumper* atau *header pin male* supaya arus listrik dari komponen satu ke komponen lainnya bisa saling terhubung/terdistribusi.

2) Mikrokontroler *Arduino Uno* : Mikrokontroler ini menerima sinyal dari *sensor PIR* dan *ultrasonik*, memproses data, dan menghitung jumlah orang yang masuk atau keluar. Mikrokontroler juga berfungsi mengirim data ke perangkat berikutnya.

3) *Wemos D1 ESP-8266* : Perangkat ini menyediakan konektivitas *Wi-Fi*, memungkinkan sistem mengirim data ke *server*. Diagram menunjukkan *Wemos D1 ESP-8266*, yang mungkin berarti ada jalur komunikasi atau redundansi untuk koneksi *Wi-Fi*.

4) *Sensor PIR dan Sensor Ultrasonik* : Kedua sensor ini berfungsi untuk mendeteksi pergerakan orang. *Sensor PIR* (*Passive Infrared*) mendeteksi perubahan panas (biasanya untuk mendeteksi keberadaan orang), sedangkan *sensor ultrasonik* mendeteksi jarak, yang bisa digunakan untuk menentukan apakah seseorang melewati area tertentu.

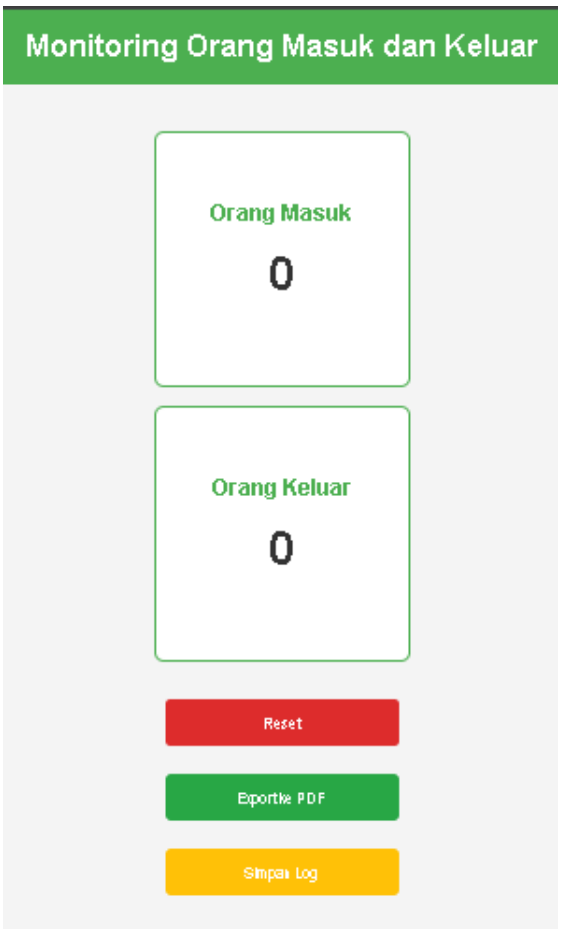
5) *LCD* : Setelah *Arduino Uno* memproses data, *LCD* digunakan untuk menampilkan informasi mengenai status deteksi. Misalnya, *LCD* bisa menunjukkan pesan seperti "Gerakan Terdeteksi" atau "Tidak Ada Gerakan".

6) *Power Supply* : Sumber daya untuk *Arduino Uno* agar sistem dapat beroperasi dengan baik.

7) Kabel *Jumper* : untuk menghubungkan satu komponen dengan komponen lainnya.

c. Rancangan Desain Antarmuka *Website E-Counter*

Pada perancangan ini penulis membuat *Website* yang terdapat beberapa fitur yang dapat digunakan sesuai dengan fungsi dan program yang akan di buat. Berikut ini tampilan antarmuka *Website E-Counter* dapat dilihat pada Gambar 3. 4



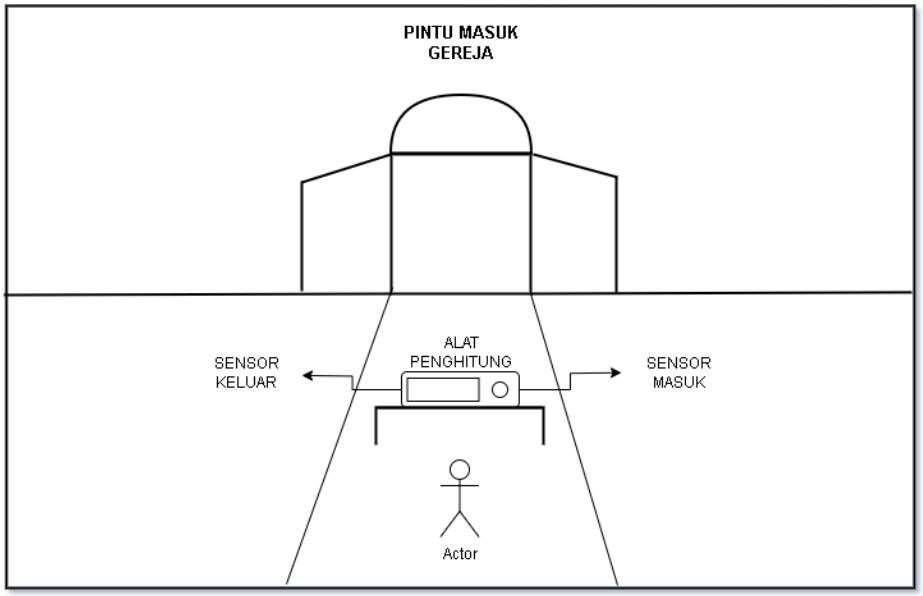
Gambar 3. 4 Desain Antarmuka *Website E-Counter*

a. Jumlah orang masuk, menampilkan jumlah orang yang masuk melewati pintu gereja

b. Jumlah orang keluar, menampilkan jumlah orang yang keluar melewati pintu gereja

d. Rancangan Desain Tempat Perangkat *IoT*

Bagian ini menerangkan penempatan alat pada gereja dan alur yang dilalui untuk menghitung jumlah jemaat/orang yang masuk ke dalam gereja agar rancangan alat yang dibangun dapat bekerja maksimal sesuai kebutuhan.



Gambar 3. 5 Rancangan Desain Tempat Perangkat *IoT*

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

4.1.1 Implementasi Perangkat Keras

Tahap implementasi ini mencakup perakitan dan konfigurasi perangkat keras dan perangkat lunak dari *prototype* sistem penghitung otomatis jumlah jemaat berbasis *IOT* pada gereja GKE Ewanggelion. Seluruh Komponen perangkat keras yang dijelaskan pada Bab III dirangkai menjadi satu kesatuan sistem yang terintegrasi. Berikut tampilan dari *prototype* :

1. Tampilan *Prototype*

   Pada tampilan depan *prototype* sistem penghitung otomatis jumlah jemaat berbasis *IOT* pada gereja GKE Ewanggelion terdapat 1 *LCD* yang berfungsi untuk menampilkan jumlah orang yang masuk dan keluar yang dapat dilihat pada gambar 4.1.



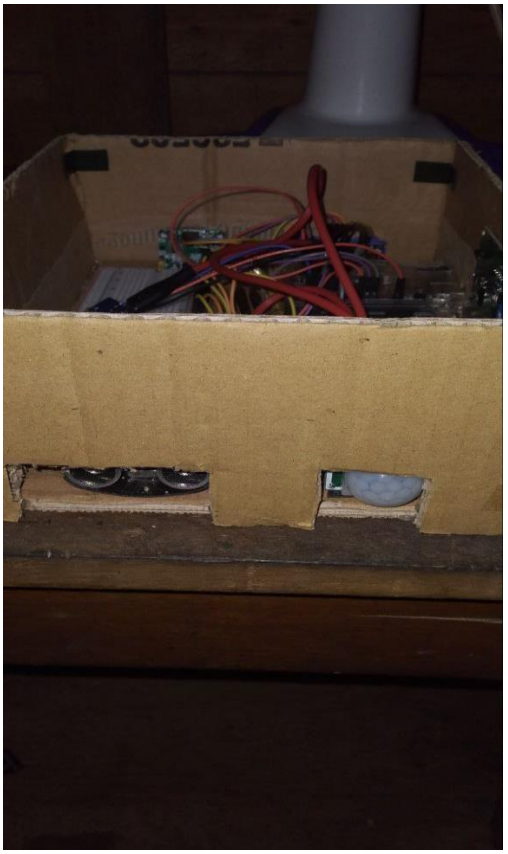
Gambar 4.1 Tampilan Depan *Prototype*

Pada bagian samping kanan *prototype* seperti yang ditunjukkan Gambar 4.2 terdapat 1 sensor ultrasonik dan 1 sensor *PIR* (*Passive Infrared*) yang berfungsi untuk menghitung jumlah orang yang masuk



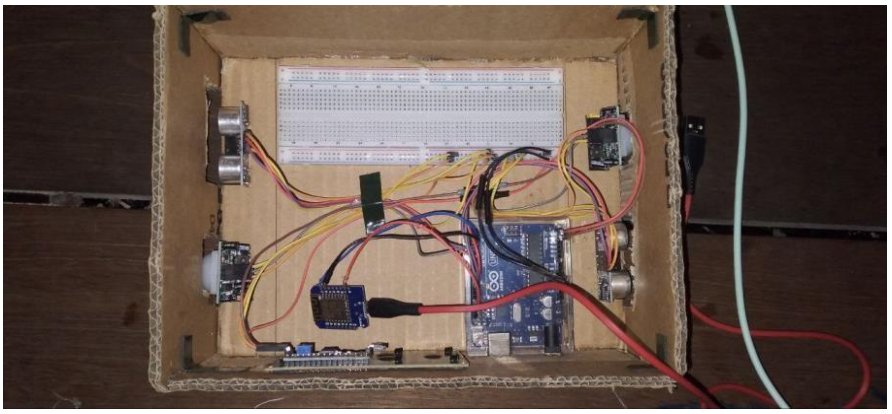
Gambar 4.2 Tampilan Samping Kanan *Prototype*

Pada bagian samping kiri *prototype* seperti yang ditunjukkan Gambar 4.3 terdapat 1 sensor ultrasonik dan 1 sensor *PIR* (*Passive Infrared*) yang berfungsi untuk menghitung jumlah orang yang keluar.



Gambar 4.3 Tampilan Samping Kiri *Prototype*

Pada bagian atas *prototype* terdapat komponen utama yang bisa kita lihat pada Gambar 4.4. komponen-komponen tersebut terdiri dari *BreadBoard*, *Arduino Uno, Wemos D1 Mini*, Sensor Ultrasonik, dan Sensor *PIR* (*Passive Infrared*).



Gambar 4.4 Tampilan Atas *Prototype*

Dalam pembuatan *prototype* sistem penghitung otomatis jumlah jemaat berbasis *IOT* pada gereja GKE Ewanggelion

1) *Breadboard* : berfungsi sebagai konduktor listrik tempat melekatkan kabel *jumper* atau *header pin male* supaya arus listrik dari komponen satu ke komponen lainnya bisa saling terhubung/terdistribusi.
2) Mikrokontroler *Arduino Uno* : Mikrokontroler ini menerima sinyal dari *sensor PIR* dan *ultrasonik*, memproses data, dan menghitung jumlah orang yang masuk atau keluar. Mikrokontroler juga berfungsi mengirim data ke perangkat berikutnya.
3) *Wemos D1 ESP-8266* : Perangkat ini menyediakan konektivitas *Wi-Fi*, memungkinkan sistem mengirim data ke *server*. Diagram menunjukkan *Wemos D1 ESP-8266*, yang mungkin berarti ada jalur komunikasi atau redundansi untuk koneksi *Wi-Fi*.
4) *Sensor PIR dan Sensor Ultrasonik* : Kedua sensor ini berfungsi untuk mendeteksi pergerakan orang. *Sensor PIR* (*Passive Infrared*) mendeteksi perubahan panas (biasanya untuk mendeteksi keberadaan orang), sedangkan *sensor ultrasonik* mendeteksi jarak, yang bisa digunakan untuk menentukan apakah seseorang melewati area tertentu.
5) *LCD* : Setelah *Arduino Uno* memproses data, *LCD* digunakan untuk menampilkan informasi mengenai jumlah orang yang masuk dan keluar.
6) *Power Supply* : Sumber daya untuk *Arduino Uno* agar sistem dapat beroperasi dengan baik.

7) Kabel *Jumper* : untuk menghubungkan satu komponen dengan komponen lainnya.

4.1.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak adalah tahapan yang berisi penjelasan terkait *source code* program pada *Arduino IDE* dan *Visual Studio Code* yang akan digunakan dalam membangun *prototype* sistem penghitung otomatis jumlah jemaat berbasis *IOT* pada gereja GKE Ewanggelion. Adapun proses program sebagai berikut :

Pada Gambar 4.5 merupakan program utama dari *server Flask* yang terintegrasi dengan *SocketIO* untuk komunikasi *real-time* dan *MySQL* sebagai *database*. Aplikasi dikonfigurasi dengan membaca file `*config.py*`, lalu menginisialisasi ekstensi `*mysql*` dan `*socketio*`. *Blueprint* `*monitoring_bp*` didaftarkan ke objek *Flask* untuk modularisasi *route*. Terakhir, aplikasi dijalankan dengan `*socketio.run*()` di *host* `0.0.0.0` pada *port* 5000 dengan mode *debug* aktif.

```
1  from flask import Flask
2  from routes.monitoring import monitoring_bp
3  from routes.websocket import socketio
4  from extensions import mysql, socketio  # ⬅️ Import dari file eksternal
5
6  app = Flask(__name__)
7
8  # Konfigurasi MySQL
9  app.config.from_pyfile('config.py')
10
11 mysql.init_app(app)  # ⬅️ Inisialisasi di sini
12 socketio.init_app(app)
13
14 # Register Blueprint
15 app.register_blueprint(monitoring_bp)
16
17 if __name__ == '__main__':
18     socketio.run(app, host='0.0.0.0', port=5000, debug=True)
19
```

Gambar 4.5 Program Utama *Server Flask*

Pada Gambar 4.6 Berisi Potongan kode konfigurasi untuk koneksi ke *database MySQL*. `*MYSQL_HOST*` menunjuk ke server *database* (*localhost*), `*MYSQL_USER*` dan `*MYSQL_PASSWORD*` digunakan untuk autentikasi (*user root* tanpa password), dan `*MYSQL_DB*` menunjukkan nama *database* yang digunakan yaitu ****Monitoring****. `*MYSQL_CURSORCLASS*` diatur ke `*'DictCursor*'` agar hasil *query* dikembalikan dalam bentuk *dictionary*. `*SECRET_KEY*` digunakan oleh *Flask* untuk keamanan sesi dan fitur lainnya.

```
1  MYSQL_HOST        = 'localhost'
2  MYSQL_USER        = 'root'
3  MYSQL_PASSWORD    = ''
4  MYSQL_DB          = 'Monitoring'
5  MYSQL_CURSORCLASS = 'DictCursor'
6  SECRET_KEY        = '101202'
7
```

Gambar 4.6 Konfigurasi *Database*

Pada Gambar 4.7 merupakan *blueprint Flask* bernama *monitoring_bp* yang memiliki dua *route* :

1. *Route* / : Menampilkan halaman monitoring.html.
2. *Route* /data : Mengambil data terbaru dari database MySQL (tabel counting) dengan query SELECT ... ORDER BY ID DESC LIMIT 1, lalu mengembalikan jumlah orang masuk dan keluar dalam format JSON. Jika data tidak ditemukan, akan dikembalikan nilai default 0. Bila terjadi error, pesan kesalahan dikembalikan dengan status HTTP 500.

```python
from flask import Blueprint, render_template, jsonify, request
from flask_socketio import emit
from extensions import mysql, socketio

monitoring_bp = Blueprint('monitoring', __name__)

@monitoring_bp.route('/')
def index():
    return render_template('monitoring.html')

@monitoring_bp.route('/data')
def get_data():
    """Mengambil data terbaru dari database."""
    try:
        cursor = mysql.connection.cursor()
        cursor.execute("SELECT orang_masuk, orang_keluar FROM counting ORDER BY ID DESC LIMIT 1")
        result = cursor.fetchone()
        cursor.close()

        if result:
            data = {'in': result['orang_masuk'], 'out': result['orang_keluar']}
        else:
            data = {'in': 0, 'out': 0}

        return jsonify(data)
    except Exception as e:
        print("Error:", e)
        return jsonify({'error': str(e)}), 500

```

Gambar 4.7 Routes Monitoring

Pada Gambar 4.8 Merupakan Kode untuk mengatur koneksi WebSocket menggunakan **Flask-SocketIO**. Objek `SocketIO` dibuat dengan mengizinkan semua origin (`cors_allowed_origins="*"`). Fungsi `handle_connect()` dijalankan saat klien terhubung, dan `handle_disconnect()` saat klien terputus. Keduanya hanya mencetak pesan ke konsol untuk mencatat status koneksi klien.

```python
from flask_socketio import SocketIO

socketio = SocketIO(cors_allowed_origins="*")

@socketio.on('connect')
def handle_connect():
    print("Client connected")

@socketio.on('disconnect')
def handle_disconnect():
    print("Client disconnected")

```

Gambar 4.8 Routes Websocket

Pada Gambar 4.9 merupakan kode untuk halaman antarmuka untuk monitoring jumlah orang yang masuk dan keluar. Halaman ini memuat dua elemen <h3> dengan ID peopleIn dan people Out untuk menampilkan data secara real-time. File CSS eksternal digunakan untuk styling, dan Socket.IO digunakan untuk komunikasi WebSocket. Saat halaman dimuat, koneksi socket dibuat ke server, siap menerima data untuk diperbarui secara dinamis di elemen yang tersedia.

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Monitoring Orang Masuk dan Keluar</title>
    <link href="{{ url_for('static', filename='css/style1.css') }}" rel="stylesheet" />
    <script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/4.0.0/socket.io.min.js"></script>
</head>
<body>
    <h1>Monitoring Orang Masuk dan Keluar</h1>
    <div class="container">
        <div class="box">
            <h2>Orang Masuk</h2>
            <h3 id="peopleIn" class="count">0</h3>
        </div>

        <div class="box">
            <h2>Orang Keluar</h2>
            <h3 id="peopleOut" class="count">0</h3>
        </div>

        <button id="resetButton" class="reset-btn">Reset</button>
    </div>

    <script>
        document.addEventListener("DOMContentLoaded", function() {
            const socket = io(location.origin, {
                transports: ["websocket", "polling"],
            });

```

Gambar 4.9 User Interface Web

Pada Gambar 4.10 merupakan kode program untuk menghitung jumlah orang yang masuk dan keluar ruangan menggunakan sensor PIR dan ultrasonik. Data ditampilkan di LCD I2C berukuran 20x4. Program mengatur pin sensor, parameter deteksi seperti jarak maksimal dan waktu debounce, serta variabel untuk menghitung dan mendeteksi kehadiran orang. Fungsi updateLCD() digunakan untuk memperbarui tampilan data di LCD.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Inisialisasi LCD dengan alamat I2C 0x27 dan ukuran 20x4
LiquidCrystal_I2C lcd(0x27, 20, 4);

// Pin untuk sensor PIR dan Ultrasonik
const int PIR_IN  = 10, TRIG_IN = 9, ECHO_IN  = 8; // Sensor masuk
const int PIR_OUT = 2, TRIG_OUT = 3, ECHO_OUT = 4; // Sensor keluar

// Parameter batasan sensor
const int MAX_DISTANCE  = 150;   // Maksimum jarak deteksi dalam cm
const int DEBOUNCE_TIME = 1000;  // Waktu debounce untuk menghindari pembacaan ganda (ms)
const int PRESENCE_TIME_LIMIT = 5000;  // Batas waktu kehadiran sebelum reset (ms)

// Variabel penghitung jumlah orang
int peopleCountIn     = 0, peopleCountOut      = 0;
int lastPeopleCountIn = -1, lastPeopleCountOut = -1;

// Variabel untuk deteksi & debounce
unsigned long lastDetectionTimeIn = 0, lastDetectionTimeOut = 0;
unsigned long presenceStartTimeIn = 0, presenceStartTimeOut = 0;
bool personPresentIn  = false, countedIn = false;
bool personPresentOut = false, countedOut = false;

// Deklarasi fungsi sebelum setup()
void updateLCD(bool forceUpdate = false);

```

Gambar 4.10 Konfigurasi Sensor

Pada Gambar 4.11 adalah kode program untuk menghubungkan modul ke jaringan WiFi dan mengirimkan data secara berkala ke server Flask melalui URL endpoint tertentu. Informasi seperti nama SSID, password, dan alamat server dikonfigurasi di awal, termasuk juga ID perangkat dan interval pengiriman data. Fungsi `connectWiFi()` bertugas menyambungkan ESP8266 ke jaringan WiFi, menampilkan status koneksi di Serial Monitor, dan menampilkan alamat IP lokal jika berhasil terhubung. Data nantinya akan dikirim dalam interval waktu tertentu (misalnya setiap 3 detik) menggunakan objek `WiFiClient`.

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>

// 🔧 Konfigurasi WiFi & Server
const char* SSID       = "Ell";       // Nama jaringan WiFi
const char* PASSWORD   = "24122002";   // Password WiFi
const char* SERVER_URL = "http://192.168.197.126:5000/ubah_data"; // URL endpoint server Flask
const int DEVICE_ID    = 1;  // ID unik untuk setiap perangkat (bisa diubah jika ada banyak perangkat)

// 🌐 Variabel untuk pengiriman data secara berkala
WiFiClient client;
unsigned long lastSendTime = 0;
const int SEND_INTERVAL    = 3000; // Interval pengiriman data dalam milidetik (3 detik)

// 📡 Fungsi untuk menghubungkan ke WiFi
void connectWiFi() {
    Serial.print("🌐 Menghubungkan ke WiFi ");
    WiFi.begin(SSID, PASSWORD); // Memulai koneksi ke jaringan WiFi

    // Tunggu hingga terhubung
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print("."); // Tampilkan indikator loading
        delay(500);
    }

    // Jika berhasil terhubung
    Serial.println("\n✅ WiFi Terhubung!");
    Serial.print("📡 IP Address ESP8266: ");
    Serial.println(WiFi.localIP()); // Tampilkan alamat IP ESP8266 di jaringan
}

```

Gambar 4.11 Konfigurasi Wemos D1 Mini

Pada Gambar 4.12 adalah kode program pada Arduino IDE untuk menghubungkan modul ke jaringan WiFi dan mengirimkan data secara berkala ke server Flask melalui URL endpoint tertentu. Informasi seperti nama SSID, password, dan alamat server dikonfigurasi di awal, termasuk juga ID perangkat dan interval pengiriman data. Fungsi `connectWiFi()` bertugas menyambungkan ESP8266 ke jaringan WiFi, menampilkan status koneksi di Serial Monitor, dan menampilkan alamat IP lokal jika berhasil terhubung. Data nantinya akan dikirim dalam interval waktu tertentu (misalnya setiap 3 detik) menggunakan objek `WiFiClient`.

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>

// 🔧 Konfigurasi WiFi & Server
const char* SSID       = "E11";       // Nama jaringan WiFi
const char* PASSWORD   = "24122002";   // Password WiFi
const char* SERVER_URL = "http://192.168.197.126:5000/ubah_data"; // URL endpoint server Flask
const int DEVICE_ID    = 1;  // ID unik untuk setiap perangkat (bisa diubah jika ada banyak perangkat)

// 🌐 Variabel untuk pengiriman data secara berkala
WiFiClient client;
unsigned long lastSendTime = 0;
const int SEND_INTERVAL    = 3000; // Interval pengiriman data dalam milidetik (3 detik)

// 🚀 Fungsi untuk menghubungkan ke WiFi
void connectWiFi() {
    Serial.print("🌐 Menghubungkan ke WiFi ");
    WiFi.begin(SSID, PASSWORD); // Memulai koneksi ke jaringan WiFi

    // Tunggu hingga terhubung
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print("."); // Tampilkan indikator loading
        delay(500);
    }

    // Jika berhasil terhubung
    Serial.println("\n✅ WiFi Terhubung!");
    Serial.print("📡 IP Address ESP8266: ");
    Serial.println(WiFi.localIP()); // Tampilkan alamat IP ESP8266 di jaringan
}
```

Gambar 4.12 Konfigurasi Wemos D1 Mini pada Arduino IDE

4.1.3 Pengujian Black box

Pengujian black box merupakan tahapan yang penting, dimana setiap komponen yang ada dalam *prototype* sistem penghitung otomatis jumlah jemaat berbasis *IOT* pada gereja GKE Ewanggelion dijalankan secara terpisah untuk mengetahui apakah ada kegagalan dalam sistem. Hasil pengujian berfokus pada komponen utama apakah bisa berjalan sesuai dengan yang diharapkan. Untuk pengujian *prototype* sistem penghitung otomatis jumlah jemaat berbasis *IOT* pada gereja GKE Ewanggelion diuji oleh **Bapak Ir. Abdul Hadi, S.T., M.Kom.** Untuk Hasil Pengujian dapat dilihat pada Tabel 4.1

Tabel 4.1 Pengujian *Black Box*

No	Pengujian	Hasil Yang Di Harapkan	Hasil Pengujian
1	Koneksi *Prototype* ke daya listrik	*Prototype* terhubung dan berfungsi	Berhasil
2	Arduino Uno	Arduino Uno terkoneksi dengan daya listrik dan berfungsi	Berhasil
3	Layar LCD 20x4	LCD menyala dan menampilkan tulisan "*Counting People* : Silahkan Lewat" serta menampilkan jumlah orang masuk dan jumlah orang keluar	Berhasil
4	Sensor *Ultrasonic*	Menghitung jumlah orang dengan mendeteksi jarak	Berhasil
5	Sensor PIR (*Passive Infrared*)	Menghitung jumlah orang dengan mendeteksi suhu tubuh	Berhasil
6	Wemos D1 *Mini*	Dapat terkoneksi dengan jaringan Wi-Fi dan mengirimkan data dari Arduino Uno ke server flask	Berhasil

4.2 Pembahasan

Penulis melakukan pengujian selama 4 kali tiap ibadah minggu pada Gereja GKE Ewanggelion. Untuk penghitungan pada minggu ke-1 dapat dilihat pada Gambar 4.13.



Gambar 4.13 Penghitungan Minggu ke-1

Pada Minggu ke-1, jumlah jemaat yang beribadah pada Gereja GKE Ewanggelion yang terdata pada Alat Hitung Otomatis berjumlah 183 orang secara total dari keterangan jumlah orang masuk 183 orang dan jumlah orang keluar 0 orang dan yang terdata pada Alat Hitung Manual berjumlah 186 orang.

Untuk Penghitungan pada minggu ke-2 dapat dilihat pada Gambar 4.14.



Gambar 4.14 Penghitungan Minggu ke-2

Pada Minggu ke-2, jumlah jemaat yang beribadah pada Gereja GKE Ewanggelion yang terdata pada Alat Hitung Otomatis berjumlah 212 orang secara total dari keterangan jumlah orang masuk 200 orang dan jumlah orang keluar 12 orang dan yang terdata pada Alat Hitung Manual berjumlah 196 orang.

Untuk Penghitungan pada minggu ke-3 dapat dilihat pada Gambar 4.15.

Minggu Ke-3



Alat Hitung Otomatis



Alat Hitung Manual

Gambar 4.15 Penghitungan Minggu ke-3

Pada Minggu ke-3, jumlah jemaat yang beribadah pada Gereja GKE Ewanggelion yang terdata pada Alat Hitung Otomatis berjumlah 203 orang secara total dari keterangan jumlah orang masuk 200 orang dan jumlah orang keluar 3 orang dan yang terdata pada Alat Hitung Manual berjumlah 208 orang.

Untuk Penghitungan pada minggu ke-4 dapat dilihat pada Gambar 4.16.



Gambar 4.16 Penghitungan Minggu ke-4

Pada Minggu ke-4, jumlah jemaat yang beribadah pada Gereja GKE Ewanggelion yang terdata pada Alat Hitung Otomatis berjumlah 193 orang secara total dari keterangan jumlah orang masuk 187 orang dan jumlah orang keluar 6 orang dan yang terdata pada Alat Hitung Manual berjumlah 191 orang.

Tabel 4.2 *Progress* Pengujian Minggu ke-1 sampai Minggu ke-4

No	Waktu	Permasalahan	Penyesuaian
1	23 Februari 2025	Jarak jangkauan sensor kurang luas	Jangkauan sensor diperluas
2	2 Maret 2025	Jarak jangkauan sensor terlalu luas	Jangkauan sensor dipersempit
3	9 Maret 2025	Penyesuaian peletakan alat	Letak alat disesuaikan
4	16 Maret 2025	Penyesuaian Akhir	Penyesuaian akhir alat

Berdasarkan hasil pengujian seluruh rangkaian *prototype* sistem penghitung otomatis jumlah jemaat berbasis *IOT* pada gereja GKE Ewanggelion berfungsi sesuai dengan yang diharapkan. Beberapa pengujian dilakukan seperti koneksi prototype ke daya listrik, Arduino Uno, Layar LCD 20x4, sensor ultrasonic, sensor PIR dan Wemos D1 Mini.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil penelitian pembuatan *prototype* sistem penghitung otomatis jumlah jemaat berbasis *IOT* pada gereja GKE Ewanggelion didapatkan kesimpulan sebagai berikut :

a. Prototype sistem penghitung otomatis jumlah jemaat berbasis IoT pada Gereja GKE Ewanggelion telah berhasil dirancang dan diimplementasikan menggunakan Arduino Uno, sensor PIR, sensor ultrasonik, dan Wemos D1 Mini.
b. Sistem mampu menghitung jumlah jemaat yang masuk dan keluar secara real-time serta menampilkannya melalui LCD dan antarmuka web berbasis Flask, yang terhubung ke database server.
c. Penggunaan sensor PIR dan sensor ultrasonik memungkinkan pendeteksian yang cukup akurat terhadap pergerakan manusia, meskipun tetap memiliki batasan dari sisi lingkungan dan karakteristik objek yang dideteksi.
d. Sistem ini dapat membantu petugas gereja untuk lebih fokus pada kegiatan ibadah tanpa harus menghitung jemaat secara manual, serta memberikan data yang dapat dimanfaatkan untuk analisis kehadiran jemaat.

5.2 Saran

Adapun saran bagi penelitian ini agar prototype ini dapat dikembangkan menjadi lebih baik, yaitu :

a. Peningkatan Akurasi

   Menggunakan kombinasi sensor yang lebih canggih atau teknologi pengenalan wajah untuk mengurangi kesalahan deteksi yang disebabkan oleh benda selain manusia atau kondisi lingkungan tertentu.

b. Redundansi Jaringan

   Menambahkan fitur cache lokal agar data tetap tersimpan sementara jika koneksi Wi-Fi terputus, dan data dapat dikirim ulang ke server saat koneksi tersambung kembali.

c. Integrasi dengan Sistem Informasi Gereja

   Sistem ini dapat dikembangkan lebih lanjut agar terintegrasi dengan sistem informasi jemaat atau laporan kehadiran otomatis.

d. Desain Fisik yang Lebih Stabil dan Permanen

   Pembuatan perangkat dalam bentuk casing atau rakitan permanen agar alat lebih aman dan mudah dipasang di pintu masuk gereja.

DAFTAR PUSTAKA

Amsar, A., Khairuman, K., & Marlina, M. (2020). Perancangan Alat Pendeteksi Co2 Menggunakan Sensor Mq-2 Berbasis Internet of Thing. *METHOMIKA Jurnal Manajemen Informatika Dan Komputerisasi Akuntansi*, *4*(1), 73–79. https://doi.org/10.46880/jmika.vol4no1.pp73-79

Anggara Ady Prasetya, A. A. P. Y. S. D. (2019). Rancang Bangun Prototype Sistem Pencatatan Pengunjung Pada Stadion Sepak Bola Menggunakan Sensor Pir (Passive Infra Red). *Jurnal Mahasiswa Fakultas Sains Dan Teknologi*, *0*(0).

Arsada, B. (2023). Aplikasi Sensor Ultrasonik Untuk Deteksi Posisi Jarak Pada Ruang Menggunakan Arduino Uno. *Jurnal Teknik Elektro*, *6*(2), 1–8.

Artha Lesmana, D. D., Arsa Suyadnya, I. M., & Shandyasa, I. W. (2023). Rancang Bangun Perangkat Keras Sistem Smart Lampu Penerangan Jalan Umum Berbasis Internet of Things Guna Mendukung Implementasi Smart City. *Jurnal SPEKTRUM*, *10*(3), 21. https://doi.org/10.24843/spektrum.2023.v10.i03.p3

Bawotong, V. T. (2019). Rancang Bangun Uninterruptible Power Supply Menggunakan Tampilan LCD Berbasis Mikrokontroler. *E-Journal Teknik Elektro Dan Komputer*, 1–7.

Dayera, Musa Bundaris Palungan, F. O. (2024). G-Tech : Jurnal Teknologi Terapan. *G-Tech : Jurnal Teknologi Terapan*, *8*(1), 186–195. https://ejournal.uniramalang.ac.id/index.php/g-tech/article/view/1823/1229

Desyantoro, E., Rochim, A. F., & Martono, K. T. (2021). Sistem Pengendali Peralatan Elektronik dalam Rumah secara Otomatis Menggunakan Sensor PIR, Sensor LM35, dan Sensor LDR. *Jurnal Teknologi Dan Sistem Komputer*, *3*(3), 405. https://doi.org/10.14710/jtsiskom.3.3.2015.405-411

Eka, P., & Susi, H. (2019). Rancang Bangun Sistem Monitoring Suhu Ruangan Bagian Pembukuan Berbasis WEB Menggunakan Mikrokontroler Arduino Uno R3. *Jurnal Teknologi Informasi Dan Komunikasi*, *11*(1), 18–33.

Feri Djuandi. (2020). Pengenalan Arduino. *E-Book. Www. Tobuku*, 1–24. http://www.tobuku.com/docs/Arduino-Pengenalan.pdf

Indriani, U., Syahputri, N., & Br Simanjuntak, A. C. (2024). Pelatihan Pembuatan Aplikasi Kalkulator Sederhana Menggunakan Vb.Net Pada Smk Tritech Medan. *INCIDENTAL : Journal Of Community Service and Empowerment*,
3(1 123–128. https://doi.org/10.62668/incidental.v3i1.739

Isrofi, A., Utama, S. N., & Putra, O. V. (2021). Rancang Bangun Robot Pemotong Rumput Otomatis Menggunakan Wireless Kontroler Modul ESP32-CAM Berbasis Internet of Things (IoT). *Jurnal Teknoinfo*, *15*(1), 45. https://doi.org/10.33365/jti.v15i1.675

Maulana, A. R., & Rahmatulloh, A. (2019). Websocket untuk Optimasi Kecepatan Data Transfer pada Real Time Chatting. *Innovation in Research of Informatics (INNOVATICS)*, *1*(1), 7–12. https://doi.org/10.37058/innovatics.v1i1.667

Nugraha, W., & Syarif, M. (2021). Penerapan Metode Prototype Dalam Perancangan Sistem Informasi Penghitungan Volume Dan Cost Penjualan Minuman Berbasis Website. *JUSIM (Jurnal Sistem Informasi Musirawas)*,
3(2), 94–101. https://doi.org/10.32767/jusim.v3i2.331

Putri Alit Widyastuti Santiary, & I Ketut Swardika. (2023). Sistem Penghitung Jumlah Orang Pada Pintu Masuk Otomatis. *Jurnal Teknologi Informasi Dan Komputer*, *9*(4), 467–471. https://doi.org/10.36002/jutik.v9i4.2570

Selay, A., Andgha, G. D., Alfarizi, M. A., Bintang, M. I., Falah, M. N., Khaira, M., & Encep, M. (2022). Karimah Tauhid, Volume 1 Nomor 6 (2022), e- ISSN 2963-590X. *Karimah Tauhid*, *1*(2963-590X), 861–862.

Tantowi, D., & Yusuf, K. (2020). Simulasi Sistem Keamanan Kendaraan Roda Dua Dengan Smartphone dan GPS Menggunakan Arduino. *Jurnal ALGOR*, *1*(2), 9–15. https://jurnal.buddhidharma.ac.id/index.php/algor/article/view/302/209

Wahyuni, R. D., Utomo, S., & ... (2022). Rancang Bangun Prototype Alat Penghitung Jumlah Orang Dalam Antrian Berbasis Esp8266. *... Seminar Nasional Ilmu ...*, *1*, 91–99. https://proceeding.unived.ac.id/index.php/snasikom/article/view/82%0Ahttps ://proceeding.unived.ac.id/index.php/snasikom/article/download/82/72

Widya, A. R., & Syaputra, H. A. (2021). Pengembangan Aplikasi Machine Monitoring System (MMS) Berbasis Teknologi IoT Wemos-D1 dan Raspberry Pi. *Seminar Nasional Informatika, Sistem Informasi Dan Keamanan Siber (SEINASI-KESI)*, *1*(1), 46–56. https://conference.upnvj.ac.id/index.php/seinasikesi/article/viewFile/33/43

LAMPIRAN

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangka Raya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

SURAT TUGAS

No.257/STMIK-3.C.2/KP/IX/2024

Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya menugaskan nama- nama tersebut di bawah ini :

1. Nama : Herkules, S.Kom., M.Cs.
   NIK : 198510042010106
   Sebagai Pembimbing I Dalam Pembuatan Program
2. Nama : Hotmian Sitohang, M.Kom.
   NIK : 198503282008002
   Sebagai Pembimbing II Dalam Penulisan Tugas Akhir

Untuk membimbing Tugas Akhir mahasiswa :

Nama	:	Imanuel Natalis Timisela
NIM	:	C2155201097
Program Studi	:	TEKNIK INFORMATIKA (55201)
Tanggal Daftar	:	30 Agustus 2024
Judul Tugas Akhir	:	Prototype Sistem Penghitung Otomatis Jumlah Jemaat Berbasis IoT Pada Gereja GKE Ewanggelion

Demikian surat ini dibuat agar dapat dipergunakan sebagaimana mestinya dan dilaksanakan dengan penuh tanggung jawab.

Palangka Raya, 10 September 2024
Ketua Program Studi Teknik Informatika,

Lili Rusdiana, M.Kom.
NIK. 198707282011007

Tembusan :
1. Pembimbing I dan II
2. Mahasiswa yang bersangkutan
3. Arsip

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangkaraya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

KARTU KEGIATAN KONSULTASI
TUGAS AKHIR

Nama Mahasiswa : IMANUEL NATALIS TIMISELA
NIM : C2155201092
Tanggal Persetujuan Judul : 6 SEPTEMBER 2024
Judul Tugas Akhir : PROTOTYPE SISTEM PENGHITUNG OTOMATIS JUMLAH JEMAAT BERBASIS IOT PADA GEREJA GKE EWANGGELION

No.	Tanggal Konsultasi		Uraian	Tanda Tangan
	Terima	Kembali		
		11/11 2024	Perbaiki latar Belakang Masalah lengkapi landasan teory Lanjut Bab III	
		14/12	tahapan penelitian di perbaiki	
		16/4	Acc Seminar Proposal	
		20/9/24	- Perbaikan judul *. BAB I sebaiknya Rumasalahan dan tempat penelitian *. Lanjut BAB II & III.	
		28/9/24	*. Masukkan penelitian relevan yg terkini dgn judul anda dan cari yg terbaru *. Alat uji coba menggunakan apa *. Lanjut seminar. Lampiran	
		08/11/24	*. Observasi ketempat penelitian *. Perancangan diberi dan di jelasin tiap tabel dan gambar * daftar Pustaka, isi di perbaiki	
		18/11/24	Acc seminar	

No.	Tanggal Konsultasi		Uraian	Tanda Tangan
	Terima	Kembali		
		05/05/25	PERBAIKAN PENULISAN BAB IV DAN V	
		21/04/25	Buat perbandingan manual dan sistem lot	
		07/05/25	PERBAIKAN BAB V	
		29/04/25	Rekomendasi Gambar Revisi, Hasil pada	
		30/05/25	PERBAIKAN BAB IV	
		02/05/25	Perbaikan BAB Program di perbaiki	
		14/05/25	PERBAIKAN BAB IV	
			DAFTAR GAMBAR, LATAR BELAKANG MASALAH.	
		09/05/25	Lengkapi lampiran	
		15/05/25	PERBAIKAN BAB III, BAB IV, BAB, BAB V	
			INSTRUMEN PENELITIAN, PERBAIKAN TABEL, DAN DAFTAR PUSTAKA	
			ACC Sidang	
			[redacted]	
		05/05/25	Acc Sidang	

Menyetujui :

Dosen Pembimbing I,

Hartika S. Evan. M. G

Dosen Pembimbing II,

Hotmian Sitinjak, M.Kom

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
STMIK PALANGKARAYA
Jl. G. Obos No. 114 ~ Telp. 0563-3224593 ~ Fax. 0536-3225515 Palangka Raya
Posel: humas@stmikplk.ac.id ~ Laman: www.stmikplk.ac.id

Nomor : 960/STMIK-G.2/AK/XI/2024
Lampiran : -
Perihal : Permohonan Izin Penelitian dan Pengumpulan Data untuk Tugas Akhir

Yth. Kepada
Ketua Majelis Pekerja Harian (MPH)
Jemaat GKE Ewanggelion
Jl. Raden Saleh III No.18, Kel. Menteng, Kec. Jekan Raya
Kota Palangka Raya, Kalimantan Tengah

Dengan hormat,

Sehubungan dengan penyusunan Tugas Akhir mahasiswa sebagai persyaratan kelulusan Program Studi Teknik Informatika (S1) pada STMIK Palangkaraya, maka dengan ini kami sampaikan permohonan izin penelitian dan pengumpulan data bagi mahasiswa kami berikut:

Nama	:	IMANUEL NATALIS TIMISELA
NIM	:	C2155201097
Prodi (Jenjang)	:	Teknik Informatika (S1)
Thn. Akad. (Semester)	:	2024/2025 (7)
Lama Penelitian	:	20 November 2024 s.d 20 Desember 2024
Tempat Penelitian	:	GKE EWANGGELION

Dengan judul Tugas Akhir:

PROTOTYPE SISTEM PENGHITUNG OTOMATIS JUMLAH JEMAAT BERBASIS IOT PADA GEREJA GKE EWANGGELION

Adapun ketentuan dan aturan pemberian informasi dan data yang diperlukan dalam penelitian tersebut menyesuaikan dengan ketentuan/peraturan pada instansi Bapak/Ibu.

Demikian permohonan ini disampaikan, atas perhatian dan kerja samanya diucapkan terima kasih.

Palangka Raya, 20 November 2024
Ketua,

Suparno, M.Kom.
NIK. 196901041995105

GEREJA KALIMANTAN EVANGELIS (ANGGOTA PGI)
KALIMANTAN EVANGELICAL CHURCH (MEMBER OF CCI)
Badan Hukum Kep. Gub.Jend.No.23 LN No. 217 tgl. 24-04-1937
Kep.Dirjen Bimas Kristen Protestan Depag RI No. 32 tgl. 03-02-1988;
NPWP. 01.222.297.2-731.000
MAJELIS JEMAAT EWANGGELION
Jl. Raden Saleh III Palangka Raya, Kalimantan Tengah, Indonesia Telp. (0536) 33229110

Palangkaraya, 28 November 2024

Nomor : 553/MPHJ-GKE-EWG/U.IV/11/2024
Lampiran : --
Perihal : Izin Penelitian dan Pengumpulan Data untuk Tugas Akhir

Kepada :
Yth. Ketua STMIK Palangkaraya

di –
Palangka Raya

Dengan Hormat,

Sehubungan dengan surat Ketua STIMIK Palangkaraya Nomor : 960/STMIK6.2/AK/XI/2024 tanggal 20 November 2024 Perihal Permohonan Izin Penelitian dan Pengumpulan Data untuk Tugas Akhir Mahasiswa STMIK atas nama **Imanuel Natalis Timisela** dengan NIM C2155201097, maka dengan ini kami sampaikan bahwa Majelis Jemaat GKE Ewanggelion memberikan izin seperti surat dimaksud.

Demikian Perihal ini disampaikan, atas perhatian dan Kerjasama yang baik diucapkan terima kasih.

MAJELIS PEKERJA HARIAN
JEMAAT GKE EWANGGELION

Pdt. EVA YULIANTY, M.Th
Ketua

Pnt. R. AGUSYONO PATIANOM
Sekretaris

Tembusan :

1. Sdr. Imanuel Natalis Timisela
2. Arsip

Transkrip Wawancara

Hasil Wawancara dengan Pendeta GKE Ewanggelion

Nama : Pdt. Sri Winarti M.Div.

Lokasi : Kantor MPH GKE Ewanggelion.

Tanggal : Jumat, 06 Desember 2024.

P : Bagaimana cara penghitungan jemaat dilakukan selama ini di gereja?

N : Dilakukan secara manual menggunakan alat hitung manual atau hand tally counter.

P : Apakah penghitungan dilakukan pada setiap ibadah atau pada ibadah tertentu saja?

N : Dilakukan pada saat ibadah minggu dan perayaan hari besar kekristenan lainnya.

P : Apa saja kendala yang dihadapi dalam penghitungan jemaat secara manual?

N : Penghitungan selama menggunakan alat hitung manual tidak ada kendala.

P : Bagaimana tingkat keakuratan penghitungan manual selama ini?

N : Cukup akurat karena tersedia 2 orang sebagai penerima tamu di depan pintu masuk yang melakukan penghitungan jemaat menggunakan alat hitung manual.

Dipindai dengan CamScanner

P : Bagaimana gereja memandang penerapan teknologi dalam konteks ibadah?

N : Sangat baik karena memudahkan petugas gereja dalam penghitungan jumlah jemaat dengan memanfaatkan teknologi.

P : Bagaimana cara mengetahui jumlah perkembangan jemaat yang beribadah?

N : Perkembangan jumlah jemaat diumumkan langsung saat ibadah tiap minggunya.

Palangka Raya, 06 Desember 2024

Narasumber

Pdt. Sri Winarti M.Div.

Dipindai dengan CamScanner





KUESIONER PENELITIAN

"PROTOTYPE SISTEM PENGHITUNG OTOMATIS JUMLAH JEMAAT BERBASIS IOT PADA GEREJA GKE EWANGGELION".

Shalom Bapak/Ibu dan Saudara/Saudari Jemaat beserta seluruh jajaran Petugas Gereja GKE Ewanggelion.

Mohon ijin, Saya Imanuel Natalis Timisela mahasiswa dari STMIK PALANGKARAYA sedang melakukan Penyelesaian Tugas Akhir untuk mengetahui tanggapan dari Bapak/Ibu dan Saudara/Saudari sekalian dari penelitian saya dengan judul " **PROTOTYPE SISTEM PENGHITUNG OTOMATIS JUMLAH JEMAAT BERBASIS IOT PADA GEREJA GKE EWANGGELION** ". Mohon kira nya dapat meluangkan waktunya sejenak untuk mengisi Kuesioner ini. Atas waktu dan perhatiannya, saya ucapkan banyak terima kasih, Tuhan Memberkati!

Petunjuk Pengisian:

1. Mohon isi identitas Anda dengan jujur.
2. Pilih jawaban anda pada salah satu dari pilihan yang tersedia.
3. Semua data yang dikumpulkan hanya digunakan untuk kepentingan penelitian dan dijaga kerahasiaannya.

Nama *

Teks jawaban singkat

Jenis Kelamin *

- Laki-Laki
- Perempuan

Umur *

- < 20 Tahun
- 20 - 30 Tahun
- 31 - 40 Tahun
- 41 - 50 Tahun
- > 50 Tahun

Status di Gereja *

- ○ Jemaat
- ○ Pengurus Gereja
- ○ Petugas Ibadah
- ○ Lainnya...

Apakah Anda merasa sistem ini mudah digunakan? *

- ○ Sangat Mudah
- ○ Mudah
- ○ Cukup Sulit
- ○ Sulit

Seberapa akurat sistem ini dalam menghitung jumlah jemaat? *

- ○ Sangat Akurat
- ○ Akurat
- ○ Kurang Akurat
- ○ Tidak Akurat

Apakah sistem ini membantu petugas gereja dalam menghitung jumlah jemaat? *

- ○ Sangat Membantu
- ○ Membantu
- ○ Kurang Membantu
- ○ Tidak Membantu

Seberapa puas anda dengan sistem penghitung otomatis ini? *

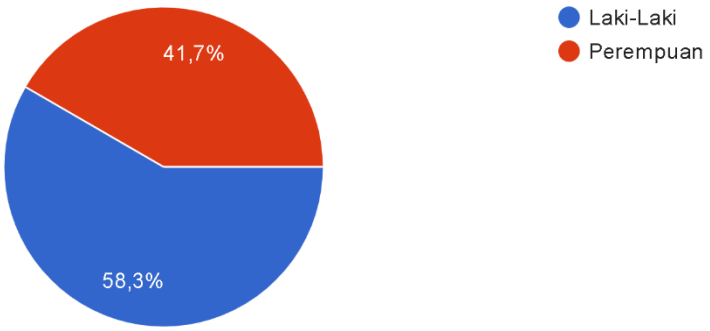
- Sangat Puas
- Puas
- Cukup Puas
- Tidak Puas

Apakah ada saran atau kritik untuk meningkatkan sistem ini? *

Teks jawaban panjang

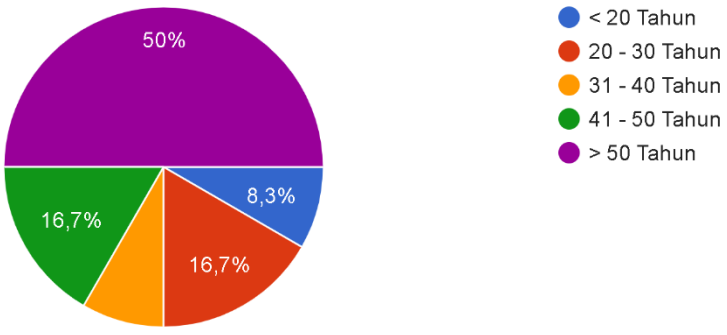
Jenis Kelamin

12 jawaban



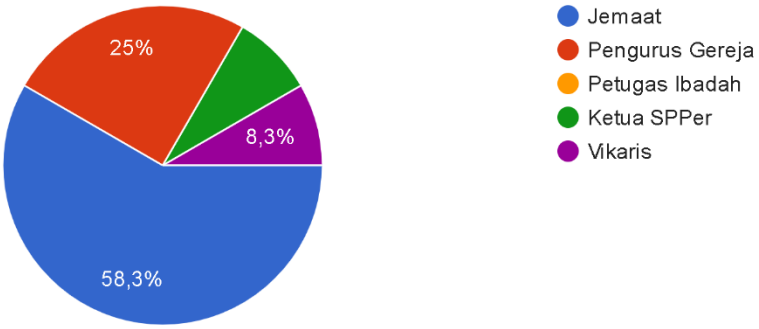
Umur

12 jawaban



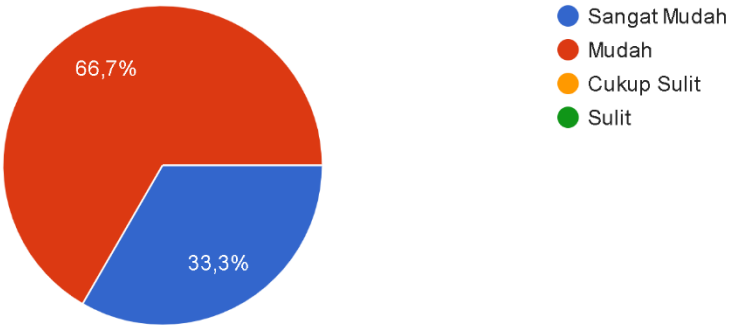
Status di Gereja

12 jawaban



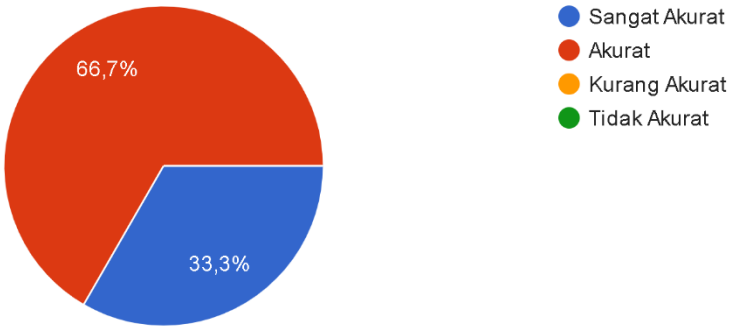
Apakah Anda merasa sistem ini mudah digunakan?

12 jawaban



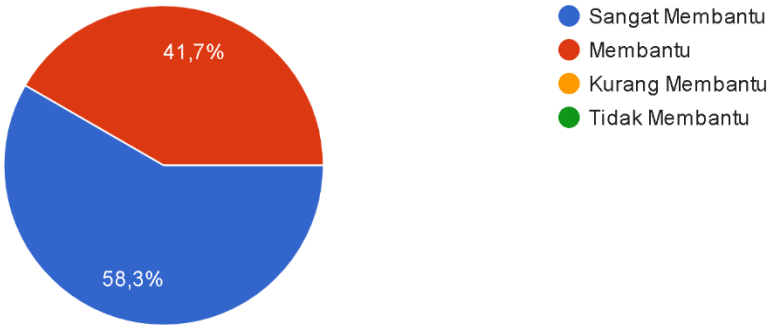
Seberapa akurat sistem ini dalam menghitung jumlah jemaat?

12 jawaban



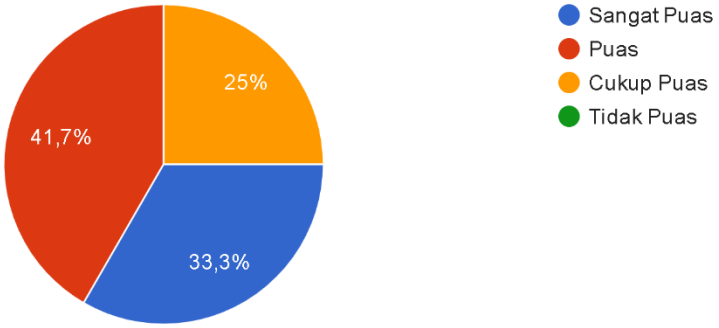
Apakah sistem ini membantu petugas gereja dalam menghitung jumlah jemaat?

12 jawaban



Seberapa puas anda dengan sistem penghitung otomatis ini?

12 jawaban



Apakah ada saran atau kritik untuk meningkatkan sistem ini?

12 jawaban

- tidak ada kritik dan saran karna saya rasa sistem ini sudah cukup bagus, sangat luar biasa bahwa SISTEM PENGHITUNG JUMLAH JEMAAT BERBASIS IOT yang telah dibuat sudah berjalan dengan baik!
- Smg sistem yg baru dan cabggih ini bisa terwujud
- belum tau aplikasinya bagaimana mengoperasikan tdk ada tutorial atau cara penggunaannya
- Perlu sosialisasi
- Kembangkan terus
- Saran agar dapat digunakan dan bermanfaat dlm ibadah jemaat.
- Sangat baik
- Kalu bisa system itu disediakan/dijual untuk digunakan sebagai alat penghitung otomatis.

-

Harus ada operator yg bisa mengoperasikannya

Cassing sebaiknya menggunakan Bahan yg lebih bagus, awet dg desain yg elegan.

Saran dr saya semoga sistem ini dapat di tingkat kan dengan tujuan agar manajemen pelayanan semakin meningkat dan bagus

Nama

12 jawaban

KEZIA MARIA GREATA SANDAI

Albertus Wichaksono

Sastori Aryanto

Yuliati Eka Asi

Novita Ariasi

Wartely Aprilini

Suriyatie

ILOK

Juniardo

Josia Jonlie

R. Agusyono Patianom

Boby Segah

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangka Raya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

SURAT TUGAS PENGUJI TUGAS AKHIR

No.114/STMIK-3.C.2/AU/V/2025

Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya menugaskan kepada nama- nama berikut :

1. Nama : Elia Zakharia, M.T.
   NIK : 199205262016104
   Sebagai Ketua
2. Nama : Catharina Elmayantie, M.Pd
   NIK : 197610252015003
   Sebagai Sekretaris
3. Nama : Amaya Andri Damaini, S.Kom., M.T
   NIK : 199309042018001
   Sebagai Anggota
4. Nama : Herkules, S.Kom., M.Cs
   NIK : 198510042010106
   Sebagai Anggota
5. Nama : Hotmian Sitohang, M.Kom
   NIK : 198503282008002
   Sebagai Anggota

Untuk menguji Tugas Akhir mahasiswa :

Nama Mahasiswa	:	IMANUEL NATALIS TIMISELA
N I M	:	C2155201097
Hari / Tanggal Ujian	:	Selasa, 27 Mei 2025
Waktu	:	11.30 WIB
Judul TA	:	PROTOTYPE SISTEM PENGHITUNG OTOMATIS JUMLAH JEMAAT BERBASIS IOT PADA GEREJA GKE EWANGGELION
Ruang	:	Sidang 1

Demikian surat ini dibuat agar dapat dipergunakan sebagaimana mestinya dan dilaksanakan dengan penuh tanggung jawab.

Palangka Raya, 26 Mei 2025

Ketua Program Studi Teknik Informatika,

26/05/2025

Ir. ABDUL HADI, S.T., M.Kom
NIK. 198505102021101

Tembusan :
1. Dosen Penguji
2. Mahasiswa yang Bersangkutan
3. Arsip

Dipindai dengan CamScanner

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangka Raya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

BERITA ACARA
UJIAN TUGAS AKHIR

Periode : Mei 2025

Pada hari ini Selasa, 27 Mei 2025 Pukul 11.30 WIB sampai dengan ..... WIB telah dilaksanakan Ujian Tugas Akhir mahasiswa atas nama :

Nama Mahasiswa	:	IMANUEL NATALIS TIMISELA
N I M	:	C2155201097
Program Studi	:	Teknik Informatika
Tahun Angkatan	:	2021
Judul TA	:	PROTOTYPE SISTEM PENGHITUNG OTOMATIS JUMLAH JEMAAT BERBASIS IOT PADA GEREJA GKE EWANGGELION

Dosen Penguji :

	Nama		Nilai	Tanda Tangan
1.	Elia Zakharia, M.T	=	.....	(.....)
2.	Catharina Elmayantie, M.Pd	=	.....	(.....)
3.	Amaya Andri Damaini, S.Kom., M.T	=	.....	(.....)
4.	Herkules, S.Kom., M.Cs	=	.....	(.....)
5.	Hotmian Sitohang, M.Kom	=	.....	(.....)

Hasil Ujian : **LULUS / ~~TIDAK LULUS~~** *) NILAI = 78,64

Revisi (Tidak Ada / ~~Minor / Mayor~~ *)

Catatan Penting :
1. Lama Perbaikan : 16 hari
2. Jika lebih dari 1 (satu) bulan dikenakan sanksi berupa denda sebesar Rp. 600.000,- (Enam ratus ribu rupiah) per bulan dari tanggal ujian.
3. Jika lebih dari 3 (tiga) bulan dari tanggal ujian maka hasil ujian dibatalkan dan wajib mengajukan judul dan pembimbing baru.

Demikian Berita Acara ini dibuat, agar dapat digunakan sebagaimana semestinya.

Palangka Raya, 27 Mei 2025

Ketua Penguji,

Elia Zakharia, M.T
NIK. 199205262016104

Mahasiswa,

IMANUEL NATALIS TIMISELA
NIM. C2155201097

Mengetahui :
Ketua Program Studi Teknik Informatika,

Ir. Abdul Hadi, S.T., M.Kom
NIK. 198505102021101

Tembusan:
3. Arsip Prodi Teknik Informatika
4. Mahasiswa yang bersangkutan

**) Coret yang tidak perlu*

Dipindai dengan CamScanner

LEMBAR BERITA ACARA PENGUJIAN BLACK BOX

Uji Coba **"*Prototype* Sistem Penghitung Otomatis Jumlah Jemaat Berbasis *IOT* Pada Gereja GKE Ewanggelion"**

Nama : IMANUEL NATALIS TIMISELA

NIM : C2155201097

Jurusan : Teknik Informatika

Jika hasil pengujian sesuai dengan yang diharapkan maka kolom hasil pengujian diisi dengan simbol "centang"(√), jika tidak maka isi dengan simbol "strip"(-).

No	Pengujian	Hasil Yang Di Harapkan	Hasil Pengujian
1	Koneksi *Prototype* ke daya listrik	*Prototype* terhubung dan berfungsi	✓
2	Arduino Uno	Arduino Uno terkoneksi dengan daya listrik dan berfungsi	✓
3	Layar LCD 20x4	LCD menyala dan menampilkan tulisan "*Counting People* : Silahkan Lewat" serta menampilkan jumlah orang masuk dan jumlah orang keluar	✓
4	Sensor *Ultrasonic*	Menghitung jumlah orang dengan mendeteksi jarak	✓
5	Sensor PIR (*Passive Infrared*)	Menghitung jumlah orang dengan mendeteksi suhu tubuh	✓
6	Wemos D1 *Mini*	Dapat terkoneksi dengan jaringan Wi-Fi dan mengirimkan data dari Arduino Uno ke server flask	✓

Palangkaraya, 9 April 2025

Dosen Penguji,

Abdul Hadi, S.T., M.Kom.

NIK. 198505102021 101

Dipindai dengan CamScanner





PROTOTYPE SISTEM PENGHITUNG OTOMATIS JUMLAH JEMAAT BERBASIS _IOT_ PADA GEREJA GKE EWANGGELION

- Program Utama Server Flask

```python
from flask import Flask
from routes.monitoring import monitoring_bp
from routes.websocket import socketio
from extensions import mysql, socketio  # ⬅️ Import dari file eksternal

app = Flask(__name__)

# Konfigurasi MySQL
app.config.from_pyfile('config.py')

mysql.init_app(app)  # ⬅️ Inisialisasi di sini
socketio.init_app(app)

# Register Blueprint
app.register_blueprint(monitoring_bp)

if __name__ == '__main__':
    socketio.run(app, host='0.0.0.0', port=5000, debug=True)

```

- Konfigurasi Database

```
MYSQL_HOST        = 'localhost'
MYSQL_USER        = 'root'
MYSQL_PASSWORD    = ''
MYSQL_DB          = 'Monitoring'
MYSQL_CURSORCLASS = 'DictCursor'
SECRET_KEY        = '101202'

```

- Routes Monitoring

```python
from flask import Blueprint, render_template, jsonify, request
from flask_socketio import emit
from datetime import datetime
from extensions import mysql, socketio

monitoring_bp = Blueprint('monitoring', __name__)

@monitoring_bp.route('/')
def index():
    return render_template('monitoring.html')

@monitoring_bp.route('/data')
def get_data():
    """Mengambil data terbaru dari database."""
    try:
        cursor = mysql.connection.cursor()
        cursor.execute("SELECT orang_masuk, orang_keluar FROM counting ORDER BY ID DESC LIMIT 1")
        result = cursor.fetchone()
        cursor.close()

        if result:
            data = {'in': result['orang_masuk'], 'out': result['orang_keluar']}
        else:
            data = {'in': 0, 'out': 0}

        return jsonify(data)
    except Exception as e:
        print("Error:", e)
        return jsonify({'error': str(e)}), 500

@monitoring_bp.route('/reset', methods=['POST'])
def reset_data():
    """Mereset data jumlah orang masuk dan keluar menjadi 0."""
    try:
        cursor = mysql.connection.cursor()
        cursor.execute("UPDATE counting SET orang_masuk = 0, orang_keluar = 0")
        mysql.connection.commit()
        cursor.close()

```

```python
        # Pastikan socketio sudah diinisialisasi sebelum emit
        if socketio:
            socketio.emit('reset_berhasil')
        else:
            print("⚠️  socketio belum diinisialisasi!")

        return jsonify({'message': 'Data berhasil direset'})
    except Exception as e:
        print("Error:", e)
        return jsonify({'error': str(e)}), 500

@monitoring_bp.route('/ubah_data', methods=['POST'])
def ubah_data():
    """Menerima data dari ESP8266 dan menyimpannya ke database."""
    try:
        if not request.is_json:
            return jsonify({"message": "Format request harus JSON!"}), 400

        data = request.get_json()
        print("Data Diterima:", data)

        id_value = data.get('ID')
        orang_masuk = data.get('orang_masuk')
        orang_keluar = data.get('orang_keluar')

        if id_value is None or orang_masuk is None or orang_keluar is None:
            return jsonify({"message": "Data tidak lengkap!"}), 400

        try:
            id_value     = int(id_value)
            orang_masuk  = int(orang_masuk)
            orang_keluar = int(orang_keluar)
        except ValueError:
            return jsonify({"message": "ID, orang_masuk, dan orang_keluar harus angka!"}), 400

        if orang_masuk < 0 or orang_keluar < 0:
            return jsonify({"message": "Nilai tidak boleh negatif!"}), 400

```

```
        cursor = mysql.connection.cursor()
        cursor.execute("SELECT * FROM counting WHERE ID = %s", (id_value,))
        existing_data = cursor.fetchone()

        if existing_data is None:
            cursor.execute("INSERT INTO counting (ID, orang_masuk, orang_keluar) VALUES (%s, %s, %s)",
                           (id_value, orang_masuk, orang_keluar))
        else:
            cursor.execute("UPDATE counting SET orang_masuk = %s, orang_keluar = %s WHERE ID = %s",
                           (orang_masuk, orang_keluar, id_value))

        mysql.connection.commit()
        cursor.close()

        socketio.emit('ubah_data', {'in': orang_masuk, 'out': orang_keluar})
        return jsonify({"message": "Data berhasil diperbarui!"}), 200

    except Exception as e:
        print("Error:", e)
        return jsonify({"message": "Terjadi kesalahan!", "error": str(e)}), 500

@monitoring_bp.route('/simpan_log', methods=['POST'])
def simpan_log():
    try:
        cur = mysql.connection.cursor()

        cur.execute("SELECT orang_masuk, orang_keluar FROM counting WHERE ID = 1")
        data = cur.fetchone()

        print("DEBUG - data dari counting:", data)  # Debug print

        if data is None:
            orang_masuk = 0
            orang_keluar = 0
        else:
            orang_masuk = data['orang_masuk']   # akses dengan key, bukan indeks
            orang_keluar = data['orang_keluar']


        now = datetime.now()
        tanggal = now.date()
        waktu = now.time()

        cur.execute("""
            INSERT INTO log_monitoring (tanggal, waktu, orang_masuk, orang_keluar)
            VALUES (%s, %s, %s, %s)
        """, (tanggal, waktu, orang_masuk, orang_keluar))

        mysql.connection.commit()
        cur.close()

        return jsonify({'status': 'success', 'message': 'Data berhasil disimpan'})
    except Exception as e:
        print("ERROR di simpan_log:", e)
        return jsonify({'status': 'error', 'message': str(e)})
```

- Routes Websocket

```python
from flask_socketio import SocketIO

socketio = SocketIO(cors_allowed_origins="*")

@socketio.on('connect')
def handle_connect():
    print("Client connected")

@socketio.on('disconnect')
def handle_disconnect():
    print("Client disconnected")

```

- User Interface Web

```html
<!DOCTYPE html>
<html lang="id">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Monitoring Orang Masuk dan Keluar</title>
    <link
      href="{{ url_for('static', filename='css/style1.css') }}"
      rel="stylesheet"
    />
    <script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/4.0.0/socket.io.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>

  </head>
  <body>
    <h1>Monitoring Orang Masuk dan Keluar</h1>
    <div class="container">
      <div class="box">
        <h2>Orang Masuk</h2>
        <h3 id="peopleIn" class="count">0</h3>
      </div>

      <div class="box">
        <h2>Orang Keluar</h2>
        <h3 id="peopleOut" class="count">0</h3>
      </div>

      <button id="resetButton" class="reset-btn">Reset</button>
      <button id="exportPDF" class="reset-btn">Export ke PDF</button>
      <button
        id="simpanLogBtn"
        class="reset-btn"
        onclick="fetch('/simpan_log', { method: 'POST' })"
      >
        Simpan Log
      </button>
    </div>

```

```
<script>
  document.addEventListener("DOMContentLoaded", function () {
    const socket = io(location.origin, {
      transports: ["websocket", "polling"],
    });

    let lastData = { in: 0, out: 0 }; // Data sebelumnya

    function updateUI(data) {
      // Cek perubahan data
      if (data.in !== lastData.in || data.out !== lastData.out) {
        console.log(`🔄 Data berubah!`);
        console.log(
          `📌 Sebelumnya: Masuk=${lastData.in}, Keluar=${lastData.out}`
        );
        console.log(`📌 Sekarang: Masuk=${data.in}, Keluar=${data.out}`);
      }

      document.getElementById("peopleIn").innerText = data.in;
      document.getElementById("peopleOut").innerText = data.out;

      lastData = data; // Simpan data terbaru
    }

    // Ambil data awal dari server setiap 1 detik
    setInterval(() => {
      fetch("/data")
        .then((response) => response.json())
        .then((data) => updateUI(data))
        .catch((error) => console.error("❌ Error fetching data:", error));
    }, 1000);

    document
      .getElementById("resetButton")
      .addEventListener("click", function () {
        fetch("/reset", {
          method: "POST",
```

```
77                headers: { "Content-Type": "application/json" },
78              })
79                .then((response) => response.json())
80                .catch((error) =>
81                  console.error("❌ Error resetting data:", error)
82                );
83            });
84
85          // Update UI saat data berubah melalui WebSocket
86          socket.on("ubah_data", function (data) {
87            updateUI(data);
88          });
89
90          // Saat reset berhasil
91          socket.on("reset_berhasil", function () {
92            console.log("✅ Data berhasil direset!");
93            updateUI({ in: 0, out: 0 });
94            alert("✅ Data berhasil direset!");
95          });
96        });
97
98        document
99          .getElementById("exportPDF")
100         .addEventListener("click", function () {
101           const { jsPDF } = window.jspdf;
102           const doc = new jsPDF();
103
104           const peopleIn = parseInt(
105             document.getElementById("peopleIn").innerText
106           );
107           const peopleOut = parseInt(
108             document.getElementById("peopleOut").innerText
109           );
110
```

```
111          const now = new Date();
112          const tanggal = now.toLocaleDateString("id-ID", {
113            weekday: "long",
114            day: "numeric",
115            month: "long",
116            year: "numeric",
117          });
118
119          const waktu = now.toLocaleTimeString("id-ID", {
120            hour: "2-digit",
121            minute: "2-digit",
122            second: "2-digit",
123            hour12: false,
124          });
125
126          const pageWidth = doc.internal.pageSize.getWidth();
127          const title = "LAPORAN MONITORING ORANG MASUK DAN KELUAR";
128
129          // Judul Tengah
130          doc.setFontSize(13);
131          doc.setFont("helvetica", "bold");
132          doc.text(title, pageWidth / 2, 20, { align: "center" });
133
134          // Informasi Tanggal dan Waktu
135          doc.setFontSize(12);
136          doc.setFont("helvetica", "normal");
137          doc.text(`Tanggal : ${tanggal}`, 10, 30);
138          doc.text(`Waktu   : ${waktu} WIB`, 10, 37);
139          doc.text(`Orang Masuk : ${peopleIn} Orang`, 10, 50);
140          doc.text(`Orang Keluar : ${peopleOut} Orang`, 10, 57);
141
```

```
142          // Judul Grafik
143          doc.setFont("helvetica", "bold");
144          doc.setFontSize(12);
145          doc.text("GRAFIK ORANG MASUK DAN KELUAR", pageWidth / 2, 70, {
146            align: "center",
147          });
148
149          // Gambar Grafik Batang
150          const maxBarWidth = 150;
151          const scale =
152            Math.max(peopleIn, peopleOut) > 0
153              ? maxBarWidth / Math.max(peopleIn, peopleOut)
154              : 0;
155
156          doc.setFont("helvetica", "normal");
157          doc.setFontSize(10);
158
159          // Orang Masuk
160          doc.text("Orang Masuk", 10, 78); // label
161          doc.setFillColor(0, 200, 0); // hijau
162          doc.rect(10, 82, peopleIn * scale, 10, "F"); // batang
163          doc.text(`${peopleIn} Orang`, 10 + peopleIn * scale + 5, 90); // jumlah di samping
164
165          // Orang Keluar
166          doc.text("Orang Keluar", 10, 98); // label
167          doc.setFillColor(220, 53, 69); // merah
168          doc.rect(10, 102, peopleOut * scale, 10, "F"); // batang
169          doc.text(`${peopleOut} Orang`, 10 + peopleOut * scale + 5, 110); // jumlah di samping
170
```

```
            const today = new Date();
            const yyyy = today.getFullYear();
            const mm = String(today.getMonth() + 1).padStart(2, "0");
            const dd = String(today.getDate()).padStart(2, "0");
            const formattedDate = `${yyyy}-${mm}-${dd}`;

            doc.save(`laporan_monitoring_${formattedDate}.pdf`);
        });
    </script>


</body>
</html>

```

- Konfigurasi Sensor

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Inisialisasi LCD dengan alamat I2C 0x27 dan ukuran 20x4
LiquidCrystal_I2C lcd(0x27, 20, 4);

// Pin untuk sensor PIR dan Ultrasonik
const int PIR_IN  = 10, TRIG_IN = 9, ECHO_IN  = 8; // Sensor masuk
const int PIR_OUT = 2, TRIG_OUT = 3, ECHO_OUT = 4; // Sensor keluar

// Parameter batasan sensor
const int MAX_DISTANCE  = 150;   // Maksimum jarak deteksi dalam cm
const int DEBOUNCE_TIME = 1000;  // Waktu debounce untuk menghindari pembacaan ganda (ms)
const int PRESENCE_TIME_LIMIT = 5000;  // Batas waktu kehadiran sebelum reset (ms)

// Variabel penghitung jumlah orang
int peopleCountIn     = 0, peopleCountOut      = 0;
int lastPeopleCountIn = -1, lastPeopleCountOut = -1;

// Variabel untuk deteksi & debounce
unsigned long lastDetectionTimeIn = 0, lastDetectionTimeOut = 0;
unsigned long presenceStartTimeIn = 0, presenceStartTimeOut = 0;
bool personPresentIn  = false, countedIn = false;
bool personPresentOut = false, countedOut = false;

// Deklarasi fungsi sebelum setup()
void updateLCD(bool forceUpdate = false);

void setup() {
    // Inisialisasi LCD
    lcd.init();
    lcd.backlight();

    // Konfigurasi mode pin untuk sensor masuk & keluar
    pinMode(PIR_IN, INPUT);
    pinMode(TRIG_IN, OUTPUT);
    pinMode(ECHO_IN, INPUT);

```

```
    pinMode(PIR_OUT, INPUT);
    pinMode(TRIG_OUT, OUTPUT);
    pinMode(ECHO_OUT, INPUT);

    Serial.begin(115200); // Mulai komunikasi serial

    // Tampilkan pesan awal pada LCD
    lcd.setCursor(2, 0);
    lcd.print("Counting People:");
    lcd.setCursor(3, 1);
    lcd.print("Silahkan Lewat");

    delay(2000); // Beri jeda

    updateLCD(true);  // Inisialisasi tampilan LCD dengan angka awal
}

// Fungsi membaca jarak dari sensor ultrasonik
long readDistance(int trigPin, int echoPin) {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    long duration = pulseIn(echoPin, HIGH, 30000); // Baca waktu pantulan ultrasonik
    long distance = duration * 0.034 / 2; // Konversi ke cm

    return (distance == 0 || distance > 200) ? -1 : distance; // Kembalikan -1 jika di luar jangkauan
}

// Fungsi untuk memperbarui tampilan LCD
void updateLCD(bool forceUpdate) {
    lcd.clear();  // Hapus layar sebelum menggambar ulang semua elemen

    Serial.print("{\"IN\":");
    Serial.print(peopleCountIn);
    Serial.print(",\"OUT\":");
    Serial.print(peopleCountOut);
    Serial.println("}");
    // Tampilkan label tetap
    lcd.setCursor(0, 0);
    lcd.print("Orang Masuk:");
    lcd.setCursor(0, 1);
    lcd.print("Jumlah: ");
    lcd.print(peopleCountIn);

    lcd.setCursor(0, 2);
    lcd.print("Orang Keluar:");
    lcd.setCursor(0, 3);
    lcd.print("Jumlah: ");
    lcd.print(peopleCountOut);

}

// Fungsi untuk mendeteksi orang masuk atau keluar
void detectPerson(int pirPin, int trigPin, int echoPin, unsigned long &lastDetectionTime,
                  unsigned long &presenceStartTime, bool &personPresent, bool &counted, bool isEntering) {
    int pirState  = digitalRead(pirPin); // Baca status sensor PIR
    long distance = readDistance(trigPin, echoPin); // Baca jarak dari sensor ultrasonik
    unsigned long currentMillis = millis(); // Ambil waktu saat ini

    // Jika PIR mendeteksi gerakan dan jarak valid
    if (pirState == HIGH && distance > 0 && distance <= MAX_DISTANCE) {
        if (!personPresent) {
            personPresent = true;
            presenceStartTime = currentMillis;
        }
        // Cek apakah orang dihitung dalam batas waktu yang ditentukan
        if (!counted && (currentMillis - presenceStartTime < PRESENCE_TIME_LIMIT)
            && (currentMillis - lastDetectionTime > DEBOUNCE_TIME)) {
```

```
            if (isEntering) {
                peopleCountIn++; // Tambah orang masuk
                Serial.println("Orang Masuk!");
            } else if (peopleCountIn > 0) {
                peopleCountOut++; // Tambah orang keluar
                peopleCountIn--;   // Kurangi jumlah orang di dalam
                Serial.println("Orang Keluar!");
            }
            lastDetectionTime = currentMillis;
            updateLCD(); // Perbarui tampilan LCD
            counted = true;
        }
    } else {
        counted = false; // Reset status jika tidak ada gerakan
        personPresent = false;
    }
}

void loop() {
    // Deteksi orang masuk dan keluar
    detectPerson(PIR_IN, TRIG_IN, ECHO_IN, lastDetectionTimeIn, presenceStartTimeIn, personPresentIn, countedIn, true);
    detectPerson(PIR_OUT, TRIG_OUT, ECHO_OUT, lastDetectionTimeOut, presenceStartTimeOut, personPresentOut, countedOut, false);
}
```

- Konfigurasi Wemos D1 Mini

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>

// 🔧 Konfigurasi WiFi & Server
const char* SSID       = "Ell";       // Nama jaringan WiFi
const char* PASSWORD   = "24122002";    // Password WiFi
const char* SERVER_URL = "http://192.168.197.126:5000/ubah_data"; // URL endpoint server Flask
const int DEVICE_ID    = 1;  // ID unik untuk setiap perangkat (bisa diubah jika ada banyak perangkat)

// 🔄 Variabel untuk pengiriman data secara berkala
WiFiClient client;
unsigned long lastSendTime = 0;
const int SEND_INTERVAL    = 3000; // Interval pengiriman data dalam milidetik (3 detik)

// 🚀 Fungsi untuk menghubungkan ke WiFi
void connectWiFi() {
    Serial.print("🔄 Menghubungkan ke WiFi ");
    WiFi.begin(SSID, PASSWORD); // Memulai koneksi ke jaringan WiFi

    // Tunggu hingga terhubung
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print("."); // Tampilkan indikator loading
        delay(500);
    }

    // Jika berhasil terhubung
    Serial.println("\n✅ WiFi Terhubung!");
    Serial.print("📡 IP Address ESP8266: ");
    Serial.println(WiFi.localIP()); // Tampilkan alamat IP ESP8266 di jaringan
}

// 🔄 Fungsi untuk reconnect WiFi jika terputus
void reconnectWiFi() {
    if (WiFi.status() != WL_CONNECTED) { // Periksa status WiFi
        Serial.println("⚠️ WiFi terputus! Mencoba menyambungkan ulang...");
        WiFi.disconnect(); // Putuskan koneksi saat ini
        WiFi.reconnect();  // Coba sambungkan kembali ke jaringan
        delay(1000);
```

```
    }
}

// 🔥 Fungsi untuk mengirim data ke server Flask
void sendData(int peopleIn, int peopleOut) {
    if (WiFi.status() != WL_CONNECTED) { // Pastikan koneksi WiFi tersedia sebelum mengirim data
        Serial.println("❌ WiFi tidak terhubung! Tidak bisa mengirim data.");
        return;
    }

    HTTPClient http; // Objek untuk melakukan HTTP request
    http.begin(client, SERVER_URL); // Tentukan URL tujuan
    http.addHeader("Content-Type", "application/json"); // Tentukan format request dalam JSON

    // Buat objek JSON untuk dikirim
    StaticJsonDocument<200> jsonOut;
    jsonOut["ID"] = DEVICE_ID;            // ID perangkat
    jsonOut["orang_masuk"] = peopleIn;    // Jumlah orang masuk
    jsonOut["orang_keluar"] = peopleOut;  // Jumlah orang keluar

    // Konversi objek JSON ke string
    String postData;
    serializeJson(jsonOut, postData);

    // ✅ Debug: Menampilkan data yang akan dikirim
    Serial.println("==================================");
    Serial.println("📤 Mengirim data ke server...");
    Serial.println("🔥 Data JSON: " + postData);
    Serial.println("==================================");

    // Kirim POST request ke server Flask
    int httpResponseCode = http.POST(postData);

    // Cek status respon dari server
    if (httpResponseCode > 0) {
        Serial.print("✅ Respons Server (Kode: ");
        Serial.print(httpResponseCode);
        Serial.println("):");
        Serial.println(http.getString()); // Tampilkan respon server
    } else {
        Serial.print("❌ Gagal mengirim! HTTP Code: ");
        Serial.println(httpResponseCode);
    }

    http.end(); // Akhiri koneksi HTTP
}

// 🧩 Fungsi untuk membaca dan parsing data JSON dari Serial (dari Arduino)
bool parseSerialData(int &peopleIn, int &peopleOut) {
    if (!Serial.available()) return false; // Cek apakah ada data dari Serial

    String data = Serial.readStringUntil('\n'); // Baca data dari Serial hingga karakter newline
    data.trim(); // Hapus spasi atau karakter newline tambahan

    // ✅ Debug: Menampilkan data asli yang diterima
    Serial.print("🔥 Data diterima dari Arduino (Raw): ");
    Serial.println(data);

    // ✅ Debug: Menampilkan data dalam format HEX (untuk cek karakter tersembunyi)
    Serial.print("📊 Data dalam HEX: ");
    for (int i = 0; i < data.length(); i++) {
        Serial.print("0x");
        Serial.print(data[i], HEX);
        Serial.print(" ");
    }
    Serial.println();

```

```
    // Parsing data JSON
    StaticJsonDocument<200> doc;
    DeserializationError error = deserializeJson(doc, data);

    if (error) { // Jika terjadi kesalahan parsing
        Serial.print("❌ Parsing JSON gagal! Error: ");
        Serial.println(error.c_str());
        return false;
    }

    // Ambil data dari JSON
    peopleIn  = doc["IN"];
    peopleOut = doc["OUT"];

    // ✅ Debug: Menampilkan hasil parsing
    Serial.print("📊 Parsed Data - Masuk: ");
    Serial.print(peopleIn);
    Serial.print(", Keluar: ");
    Serial.println(peopleOut);

    return true; // Berhasil membaca data JSON
}


void setup() {
    Serial.begin(115200); // Inisialisasi komunikasi serial dengan baud rate 115200
    connectWiFi(); // Hubungkan ke WiFi saat perangkat dinyalakan
}

void loop() {
    reconnectWiFi(); // Pastikan WiFi tetap terhubung

    int peopleIn, peopleOut;
    if (parseSerialData(peopleIn, peopleOut)) { // Jika ada data yang diterima dari Serial
        if (millis() - lastSendTime > SEND_INTERVAL) { // Kirim data sesuai interval waktu
            sendData(peopleIn, peopleOut);
            lastSendTime = millis(); // Update waktu terakhir pengiriman data
        }
    }
}
```