Training on the Design of the React-Based 'QueueGo' Digital Queue Management System for Multi-Counter Services at SMKN 24 East Jakarta

by Lanny Catrin Dale

Submission date: 25-Nov-2026 10:48AM (UTC+0700)
Submission ID: 3099567949
File name: 4104._Rano_Turnitin.pdf (743.08K)
Word count: 4452
Character count: 28854

Training on the Design of the React-Based 'QueueGo' Digital Queue Management System for Multi-Counter Services at SMKN 24 East Jakarta

Lanny Catrin Dale[1)], *Rano Agustino[2)], Dewi Andriyanti[3)], Lexi Majesty Pendong[4)]

[1)3)]Prodi Keamanan Sistem Informasi, Politeknik Jakarta Internasional, Indonesia
[2)]Prodi Sistem Informasi, Universitas Mohammad Husni Thamrin, Indonesia
[4)]Prodi Proteksi Tanaman, IPB University, Indonesia
***Correspondence author**: Rano Agustino, rano.agustino@gmail.com, Jakarta, Indonesia
DOI:

Abstract

This community service program (PKM) was carried out at SMKN 24 Jakarta Timur on 7–18 September 2026 to improve students' understanding and practical skills in designing and developing a digital queue management system for multi-counter services. The school partner previously relied on manual queue handling, which caused disorderly service sequences, limited visibility of queue positions for visitors, and difficulties for staff in monitoring multiple service counters simultaneously. To address these issues, the team conducted a structured training and workshop using QueueGo, a React-based queue management application, both as an interactive learning medium and as the prototype output. The implementation method comprised four stages: pretest, guided training and workshop, evaluation (posttest), and participant feedback. Meanwhile, the system prototype was developed using the Prototype model, establishing the system design and frontend interface as the activity checkpoint. The resulting outputs include a functional QueueGo prototype featuring separate customer and operator workflows (user registration and login, guest access, counter selection, digital queue tickets, real-time queue monitoring, operator control panel, automated voice announcements, and service history), comprehensive system design documentation (use case diagram and activity diagram, activity documentation, and a community service manuscript. Pretest and posttest evaluations demonstrated an increase in participants' average scores from 52.50 to 82.50 (N-Gain 0.63), while quantitative testing of performance, usability, reliability, and security is proposed as follow-up work.

Keywords: *Community Service, Digital Queue, Multi-Counter Service, Queuego, React*

Abstrak

Kegiatan pengabdian kepada masyarakat (PKM) ini dilaksanakan di SMKN 24 Jakarta Timur pada 7–18 September 2026 untuk meningkatkan pemahaman serta keterampilan praktis siswa dalam merancang dan mengembangkan sistem manajemen antrean digital pada pelayanan multi-loket. Mitra sekolah sebelumnya masih menerapkan pengelolaan antrean secara manual, sehingga urutan layanan kurang tertib, pengunjung tidak memperoleh informasi posisi antrean secara pasti, dan petugas mengalami kesulitan memantau beberapa loket secara bersamaan. Solusi yang diterapkan berupa pelatihan dan workshop terstruktur menggunakan QueueGo, aplikasi manajemen antrean berbasis React, sebagai media pembelajaran interaktif sekaligus prototipe luaran kegiatan. Metode pelaksanaan terdiri atas empat tahapan, yaitu pretest, pelatihan/workshop terbimbing, evaluasi (postest), serta pengumpulan umpan balik peserta dan mitra. Sementara itu, pengembangan perangkat lunak menerapkan model Prototype dengan menetapkan dokumen perancangan dan antarmuka frontend sebagai titik batas (checkpoint) kegiatan. Luaran yang dihasilkan meliputi prototipe aplikasi QueueGo dengan pemisahan fungsi pelanggan dan operator (registrasi dan login, akses tamu, pemilihan loket, tiket antrean digital, pemantauan status antrean, panel kendali operator, pengumuman suara otomatis, serta riwayat layanan), dokumen perancangan sistem (use case diagram, activity diagram), dokumentasi kegiatan, serta artikel ilmiah PKM. Hasil evaluasi menunjukkan peningkatan nilai rata-rata peserta dari 52,50 menjadi 82,50 (N-Gain 0,63), sedangkan pengujian kuantitatif terhadap performa, usability, reliabilitas, dan keamanan direncanakan sebagai tindak lanjut.

Kata Kunci: Antrean Digital, Pelayanan Multi-Loket, Pengabdian Kepada Masyarakat, QueueGo, React

PENDAHULUAN

Pelayanan yang menggunakan sistem antrean menghadapi kebutuhan untuk mengatur urutan pelanggan secara tertib, mengurangi kebingungan mengenai giliran, dan memberikan informasi status pelayanan secara cepat. Pengelolaan antrean secara manual dapat menyebabkan pelanggan harus menunggu tanpa informasi yang jelas, operator kesulitan memantau daftar tunggu, serta proses pemanggilan dan pencatatan riwayat menjadi kurang terstruktur. Kondisi tersebut menunjukkan kebutuhan terhadap sistem antrean digital yang dapat mengintegrasikan pengambilan nomor, pemantauan status, pemanggilan pelanggan, dan pencatatan riwayat dalam satu aplikasi (Bhat, 2015; Rema & Sikdar, 2021; Fauzi & Suhaimi, 2026).

Perkembangan teknologi web memungkinkan pengelolaan antrean dilakukan melalui antarmuka yang responsif dan mudah digunakan. QueueGo dikembangkan sebagai aplikasi manajemen antrean berbasis React yang menyediakan alur untuk pelanggan dan operator. Pelanggan dapat masuk sebagai pengguna terdaftar atau tamu, memilih loket, mengambil nomor antrean, dan melihat tiket antrean digital. Operator memperoleh panel operasional untuk memilih loket, memanggil antrean berikutnya, memanggil ulang, memulai pelayanan, menyelesaikan, atau melewati antrean (QueueGo, 2026).

Aspek pengendalian akses menjadi bagian penting dalam QueueGo karena fungsi pelanggan dan operator memiliki kebutuhan yang berbeda. Repository menyediakan model pengguna dengan role user, operator, dan admin, serta logika operator untuk membedakan akses operasional. Selain itu, aplikasi menggunakan Auth Context dan User Store untuk mengelola sesi pengguna, sementara konfigurasi API menyediakan endpoint register dan login. Pada sisi operasional, perubahan status antrean dilakukan melalui endpoint khusus sehingga alur waiting → calling → serving → completed atau skipped dapat dikelola secara terstruktur (National Institute of Standards and Technology [NIST], n.d.; Dale, 2025a; Dale, 2025b; QueueGo, 2026).

QueueGo menggunakan arsitektur frontend berbasis React 19 dengan Vite sebagai build tool. Antarmuka menggunakan Tailwind CSS 4 dan Lucide Icons, sedangkan React Router digunakan untuk navigasi. Service API menyediakan komunikasi JSON dengan backend yang dikonfigurasi pada VITE_API_BASE_URL, dengan default http://localhost:8095. Endpoint yang terdokumentasi meliputi autentikasi, loket, antrean hari ini, pembuatan antrean, pemanggilan, recall, serving, complete, dan skip. Struktur komponen dipisahkan ke dalam screen autentikasi, home, queue, context, model, service, dan store sehingga

mendukung pengembangan modular (Lucide, 2026; React, 2026; React Router, 2026; Tailwind CSS, 2026; Vite, 2026; QueueGo, 2026).

Analisis Situasi Mitra

Mitra kegiatan pengabdian kepada masyarakat (PKM) ini adalah SMKN 24 Jakarta Timur. Sekolah menyediakan laboratorium sebagai tempat kegiatan dan melibatkan siswa-siswinya sebagai peserta bootcamp perancangan dan pengembangan aplikasi. Pelayanan di lingkungan sekolah, seperti perpustakaan dan koperasi sekolah melibatkan beberapa loket sehingga pengaturan giliran pengunjung memerlukan sistem yang tertib. Penerapan sistem antrean digital melalui kegiatan pengabdian kepada masyarakat telah dilaporkan pada layanan klinik (Palupi et al., 2025), dan analisis kebutuhan pengguna merupakan tahap penting dalam pengembangan sistem antrean digital real-time (Azzikri et al., 2026).

Permasalahan Mitra

Hasil identifikasi bersama pihak sekolah menunjukkan permasalahan berikut, sehingga perlu dilakukan pelatihan/workshop:

- Pengelolaan antrean masih manual sehingga urutan giliran kurang tertib dan pelanggan tidak memperoleh informasi posisi antrean.
- Petugas kesulitan memantau daftar tunggu pada beberapa loket serta mencatat riwayat pelayanan secara terstruktur.
- Sekolah belum memiliki sistem antrean digital yang dapat dikembangkan dan dikelola secara mandiri.
- Pengetahuan dan keterampilan siswa dalam merancang serta mengembangkan aplikasi web modern (React dan REST API) masih perlu ditingkatkan melalui pelatihan terstruktur.

Tujuan PkM

Tujuan kegiatan PkM ini adalah (1) meningkatkan pemahaman siswa SMKN 24 Jakarta Timur tentang analisis kebutuhan dan perancangan sistem antrean digital; (2) melatih keterampilan siswa dalam mengimplementasikan antarmuka dan integrasi REST API menggunakan React; (3) menghasilkan prototipe QueueGo sebagai solusi pelayanan multi-loket bagi mitra; dan (4) mengukur peningkatan pemahaman peserta melalui pretest, postest, dan umpan balik.

Solusi dan Target Luaran

Solusi yang ditawarkan adalah pelatihan/workshop yang menggunakan QueueGo sebagai media praktik sekaligus produk yang dirancang bersama mitra. Target luaran kegiatan adalah sebagai berikut:

- Prototipe aplikasi QueueGo dengan fungsi pelanggan dan operator.
- Dokumen perancangan sistem berupa use case diagram, activity diagram, dan ERD.
- Peningkatan pemahaman peserta yang diukur melalui pretest dan postest.
- Dokumentasi kegiatan dan artikel PkM.

METODE PELAKSANAAN

Gambaran Lokasi, Waktu, dan Peserta

Kegiatan dilaksanakan di Laboratorium sekolah SMKN 24 Jakarta Timur pada tanggal 7 sampai 18 September 2026, pukul 09.00 sampai 16.00 WIB. Laboratorium dilengkapi 25 unit komputer dan jaringan internet sehingga peserta dapat berlatih secara langsung. Tim pelaksana terdiri atas dosen Prodi Keamanan Sistem Informasi Politeknik Jakarta Internasional, dosen Prodi Sistem Informasi Universitas Mohammad Husni Thamrin, dan dosen Prodi Proteksi Tanaman. Peserta berjumlah 20 siswa-siswi SMKN 24 Jakarta Timur dari jurusan Rekayasa Perangkat Lunak (RPL). Suasana kegiatan bersama siswa-siswi ditunjukkan pada Gambar 1.



Gambar 1: Dosen bersama siswa-siswi SMKN 24

Tahapan Pelaksanaan PkM

Pelaksanaan PkM mengikuti empat tahap, yaitu pretest, pelatihan/worksho, evaluasi (postest), dan feedback, sebagaimana ditunjukkan pada Gambar 2.



Gambar 2: Tahapan pelaksanaan PkM

Pada tahap pretest, peserta mengerjakan tes tertulis sebanyak 100 soal untuk mengukur pemahaman awal tentang sistem antrean digital, perancangan sistem, dan dasar React. Pada tahap pelatihan/workshop, materi disampaikan secara bertahap, yaitu konsep antrean digital dan analisis kebutuhan, perancangan sistem (UML, flowchart, dan ERD), dasar React dan komponen antarmuka, integrasi REST API serta pengelolaan status antrean, dan praktik terbimbing menggunakan QueueGo. Metode yang digunakan adalah ceramah, demonstrasi, praktik terbimbing, dan diskusi.

Pada tahap evaluasi, peserta mengerjakan postest dengan butir soal yang setara dengan pretest. Peningkatan pemahaman dianalisis dari selisih rata-rata nilai dan N-Gain, yaitu (skor postest − skor pretest) dibagi (skor maksimum − skor pretest). Pada tahap feedback, peserta dan pihak sekolah mengisi kuesioner skala Likert 1–5 serta berdiskusi untuk menyampaikan masukan terhadap materi, pelaksanaan, dan prototipe QueueGo.

Teknik Pengumpulan Data dan Karakteristik Pengguna

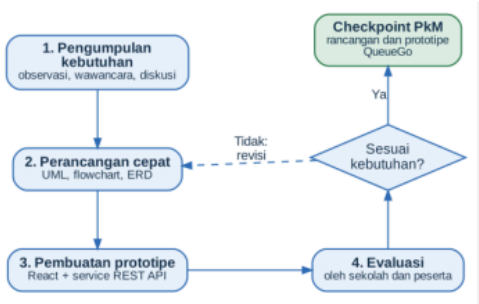
Data kebutuhan dikumpulkan melalui (1) observasi langsung pada lokasi layanan di sekolah; (2) wawancara dan diskusi dengan pihak mitra, yaitu guru, mengenai alur pelayanan dan kendala antrean; (3) diskusi dengan peserta untuk menggali harapan terhadap fitur; serta (4) pretest–postest, kuesioner umpan balik, dan dokumentasi untuk mengukur hasil pelatihan. Karakteristik pengguna yang dilibatkan dirangkum pada Tabel 1.

Tabel 1.
Karakteristik pengguna yang dilibatkan

Pengguna	Karakteristik	Kebutuhan utama
Pelanggan (siswa, guru, tamu)	Pengguna sesekali, mengakses dari ponsel atau komputer, tanpa pelatihan khusus.	Pengambilan nomor yang sederhana dan informasi posisi antrean.
Operator (petugas loket)	Pengguna berulang selama jam layanan dan mengelola satu loket.	Panel tunggal untuk memanggil, melayani, dan melewati antrean.
Admin	Pengelola sistem; peran tersedia pada model pengguna dan berfungsi di sisi backend.	Pengelolaan loket dan akun (tahap lanjutan).
Peserta pelatihan (siswa)	Kemampuan dasar pemrograman beragam Flutter, React dan REST API	Materi bertahap dan praktik terbimbing.

Metode Pengembangan Sistem

Pengembangan QueueGo dalam kegiatan ini menggunakan metode Prototype karena kebutuhan mitra belum terumus rinci di awal dan memerlukan umpan balik berulang dari sekolah dan peserta, sejalan dengan pendekatan iteratif pada pengembangan sistem informasi (Kumar & Gupta, 2023). Tahapannya meliputi (1) pengumpulan kebutuhan, (2) perancangan cepat berupa UML, flowchart, dan ERD, (3) pembuatan prototipe dengan React dan service REST API, (4) evaluasi oleh sekolah dan peserta terlihat pada gambar 3 dibawah ini.



Gambar 3: Siklus metode Prototype pada pengembangan QueueGo

Perancangan dan prototipe ini ditetapkan sebagai titik batas (checkpoint) kegiatan PkM. Bahan kegiatan berupa source code QueueGo (React 19, Vite 8, Tailwind CSS 4, React Router 7, dan Lucide Icons) yang mencakup modul autentikasi, beranda pengguna, panel operator, monitor antrean, riwayat, pengambilan antrean, tiket digital, service API, model data, dan state management (QueueGo, 2026; React, 2026; Vite, 2026). Implementasi backend penuh, penerapan pada layanan nyata, dan pengujian kuantitatif ditempatkan sebagai tindak lanjut.

Kelayakan Perguruan Tinggi

Kegiatan PKM ini didukung oleh Politeknik Jakarta Internasional dan Universitas Mohammad Husni Thamrin melalui penugasan dosen, penjadwalan kegiatan, dan fasilitasi kerja sama dengan SMKN 24 Jakarta Timur. Tim terdiri atas dosen bidang keamanan sistem informasi, sistem informasi (analisis dan perancangan sistem), dan proteksi tanaman, dengan pengalaman pengembangan aplikasi digital seperti aplikasi presensi digital (Dale, 2025a).

Kegiatan melibatkan 4 mahasiswa sebagai asisten pelatihan dan pendamping praktik. Kegiatan dilaksanakan bersama oleh dosen dari Politeknik Jakarta Internasional dan Universitas Mohammad Husni Thamrin.

HASIL DAN PEMBAHASAN

Dari segi analisis kebutuhan, QueueGo memiliki dua kelompok fungsi utama, yaitu pelanggan dan operator. Pelanggan membutuhkan proses pengambilan nomor yang sederhana, informasi loket yang tersedia, tiket digital, dan informasi jumlah antrean di depan. Operator membutuhkan panel untuk memilih loket tugas, melihat daftar tunggu, memanggil antrean, melakukan recall, memulai pelayanan, menyelesaikan atau melewati antrean. Pemisahan kebutuhan tersebut menjadi dasar pembentukan screen dan komponen pada aplikasi (Bhat, 2015; Rema & Sikdar, 2021; Fauzi & Suhaimi, 2026). Pemetaan antara permasalahan mitra, kebutuhan pengguna, dan fitur QueueGo ditunjukkan pada Tabel 2.

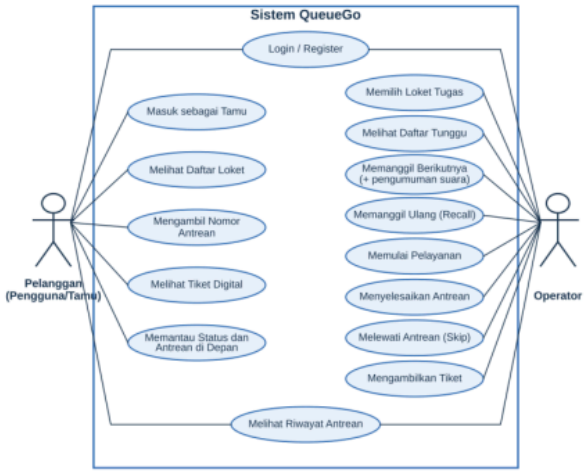
Tabel 2.
Pemetaan permasalahan mitra, kebutuhan pengguna, dan fitur QueueGo

No	Permasalahan mitra	Kebutuhan pengguna	Fitur QueueGo
1	Antrean manual dan giliran kurang tertib	Pengambilan nomor yang sederhana	Daftar loket, pengambilan nomor, tiket antrean digital
2	Pelanggan tidak mengetahui posisi antrean	Informasi status dan jumlah antrean di depan	Tiket digital, monitor status, refresh data berkala
3	Petugas sulit memantau daftar tunggu multi-loket	Panel operasional per loket	Panel operator: pilih loket, daftar tunggu, jumlah menunggu
4	Pemanggilan dan pelayanan belum terstruktur	Kendali pemanggilan dan pelayanan	Panggil berikutnya, recall, serving, complete, skip, pengumuman suara
5	Riwayat pelayanan belum tercatat	Rekap antrean selesai, dilewati, dibatalkan	Halaman riwayat
6	Kebutuhan akses berbeda antar pengguna	Pemisahan peran pengguna	Login/register, akses tamu, role user/operator/admin
7	Keterampilan peserta perlu ditingkatkan	Pelatihan bertahap dengan praktik	Workshop dengan QueueGo sebagai media praktik

Hasil Perancangan Sistem

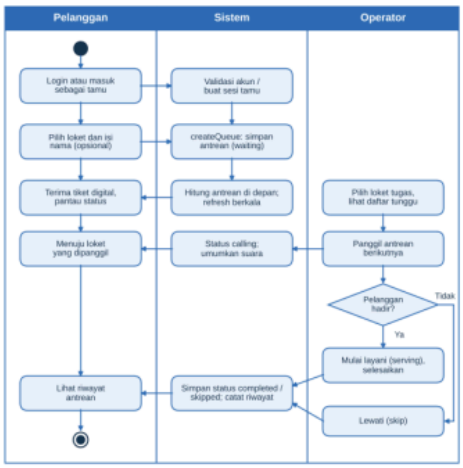
Kebutuhan pada Tabel 2 ditransformasikan menjadi rancangan sistem yang disajikan dalam bentuk use case diagram dan activity diagram.

Use case diagram pada Gambar 4 menunjukkan dua aktor utama. Pelanggan (pengguna terdaftar atau tamu) dapat melihat daftar loket, mengambil nomor, melihat tiket, dan memantau status antrean. Operator dapat memilih loket tugas, melihat daftar tunggu, memanggil berikutnya disertai pengumuman suara, memanggil ulang, memulai pelayanan, menyelesaikan, melewati, dan mengambilkan tiket. Login/register dan riwayat digunakan oleh kedua aktor.



Gambar 4: Use case diagram QueueGo

Activity diagram pada Gambar 5 menggambarkan interaksi pelanggan, sistem, dan operator, mulai dari login atau akses tamu, pengambilan nomor, pemanggilan oleh operator, hingga penyimpanan status completed atau skipped ke riwayat.



Gambar 5: Activity diagram QueueGo

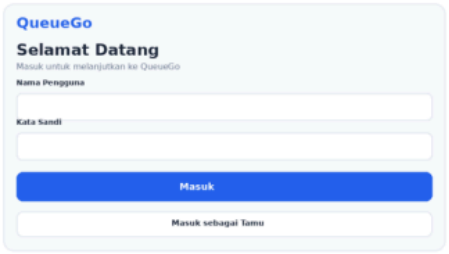
Dari segi arsitektur implementasi, QueueGo menggunakan pola client-server. React menjadi lapisan presentasi dan state management, API Service menjadi lapisan komunikasi, sedangkan REST API menjadi lapisan layanan data. Konfigurasi repository menunjukkan backend Go API dengan database MySQL atau MariaDB sebagai target integrasi, tetapi backend tersebut tidak terdapat pada ZIP React yang dianalisis sehingga implementasi server dan database tidak dinilai sebagai bagian dari hasil source code ini (QueueGo, 2026; React, 2026). Gambaran arsitektur ditampilkan pada Gambar 6.



Gambar 6: Struktur modul dan arsitektur QueueGo

Hasil Implementasi Prototipe

Dari segi arsitektur pengguna, QueueGo menyediakan autentikasi dan pembentukan sesi melalui AuthContext dan UserStore. Login mencoba menghubungi backend terlebih dahulu, sedangkan aplikasi juga menyediakan akun demo lokal dan akses sebagai tamu. Model pengguna menormalisasi role dari data API dan menentukan status operator untuk kebutuhan panel operasional (NIST, n.d.; QueueGo, 2026). Antarmuka login ditampilkan pada Gambar 7.



Gambar 7: Antarmuka Login QueueGo

Fungsi pelanggan meliputi masuk sebagai pengguna atau tamu, melihat status loket, memilih loket pelayanan, mengambil nomor antrean, dan membuka tiket antrean digital. Tiket menampilkan nomor antrean, nama pelanggan, waktu pengambilan, status, loket tujuan, serta jumlah antrean yang berada di depan. Peran dan fungsi utama pengguna dirangkum pada Tabel 3.

Tabel 3.
Peran dan fungsi utama pengguna QueueGo

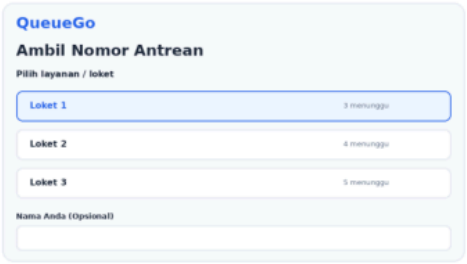
Peran	Platform	Fungsi Utama
Pelanggan	Web/React	Login/tamu, pilih loket, ambil nomor, lihat tiket dan posisi antrean.
Operator	Web/React	Pilih loket, panggil, recall, mulai pelayanan, selesai, skip, dan kelola daftar tunggu.
Admin	Web/React/API	Role ada pada model pengguna; detail backend tidak termasuk ZIP frontend.

Fungsi operator meliputi memilih loket tugas, melihat antrean aktif dan daftar tunggu, menguji suara, mengambilkan tiket, memanggil antrean berikutnya, memanggil ulang, memulai pelayanan, menyelesaikan, melewati, dan memanggil antrean tertentu. Fungsi-fungsi tersebut dikelompokkan pada OperatorTab sehingga aktivitas pelayanan dapat dilakukan dari satu panel. Pemisahan fungsi pelanggan dan operator ditunjukkan berdasarkan komponen antarmuka dan aksi yang tersedia pada repository (Tabel 4).

Tabel 4.
Matriks fungsi pelanggan dan operator QueueGo

Fitur	Pelanggan	Operator
Login/Register	Ya	Ya
Masuk sebagai Tamu	Ya	Tidak sebagai fungsi operasional
Ambil Antrean	Ya	Ya, melalui fungsi 'Ambilkan Tiket'
Lihat Daftar Tunggu	Ya	Ya
Panggil/Recall	Tidak	Ya
Serving/Complete/Skip	Tidak	Ya
Riwayat	Ya	Ya

Dari segi fitur utama, hasil pemeriksaan repository menunjukkan bahwa QueueGo memiliki beberapa lapisan fitur. Pada sisi pelanggan tersedia beranda, status loket, daftar antrean menunggu, pengambilan nomor, dan tiket antrean. Pada sisi operator tersedia panel operasional dan pengelolaan status antrean. Aplikasi juga memiliki halaman riwayat untuk menampilkan antrean yang selesai diproses, dilewati, atau dibatalkan (QueueGo, 2026). Halaman pengambilan nomor ditampilkan pada Gambar 8.



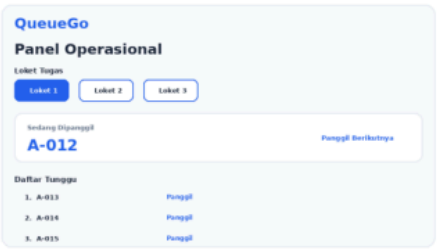
Gambar 8: Halaman Pengambilan Nomor Antrean QueueGo

Pada halaman pengambilan antrean, pengguna memilih salah satu loket yang tersedia dan secara opsional mengisi nama. Setelah tombol AMBIL ANTREAN dijalankan, aplikasi memanggil apiService.createQueue() dengan counterId dan customerName. Respons dari backend dinormalisasi melalui normalizeQueue sebelum disimpan ke state aplikasi. Proses tersebut menghasilkan tiket antrean yang dapat dipantau pengguna (QueueGo, 2026). Tiket yang dihasilkan ditampilkan pada Gambar 9.



Gambar 9: Tiket Antrean Digital QueueGo

Tiket antrean digital menampilkan nomor antrean, loket tujuan, nama pelanggan, waktu pengambilan, status antrean, dan jumlah antrean di depan. Informasi tersebut dirancang agar pelanggan dapat memperkirakan posisi dalam urutan pelayanan tanpa harus terus bertanya kepada petugas (Bhat, 2015; QueueGo, 2026). Panel operator ditampilkan pada Gambar 10.



Gambar 10: Panel Operasional Operator QueueGo

Dari segi alur antrean, proses dimulai ketika pelanggan login atau masuk sebagai tamu. Pengguna memilih loket dan membuat antrean. Sistem mengembalikan nomor antrean dan menyimpan data melalui API. Operator memilih loket tugas kemudian memanggil antrean berikutnya. Status antrean berubah dari waiting menjadi calling, kemudian dapat menjadi serving, completed, skipped, atau cancelled. HomeShell melakukan refresh data antrean secara berkala sehingga perubahan status dapat ditampilkan kembali pada antarmuka (Bhat, 2015; QueueGo, 2026).

Dari segi implementasi service API, API Service menyediakan fungsi Get Counters, Get Queues Today, Create Queue, Call Next Queue, Recall Queue, Serve Queue, Complete Queue, dan Skip Queue. Setiap respons API dinormalisasi agar data JSON memiliki bentuk yang konsisten pada sisi React. API Config mendefinisikan base URL melalui VITE_API_BASE_URL dan menyediakan endpoint REST untuk autentikasi, loket, dan antrean. Pola tersebut memisahkan logika komunikasi server dari komponen antarmuka (QueueGo, 2026).

Dari segi pengelolaan status, model Queue Status digunakan untuk menjaga konsistensi status antrean. Riwayat memfilter status completed, skipped, dan cancelled, sedangkan daftar tunggu mengambil status waiting. Home Shell juga menghitung antrean di depan berdasarkan loket dan urutan id sehingga pengguna dapat memperoleh informasi posisi antrean (QueueGo, 2026).

Dari segi implementasi frontend, QueueGo menggunakan React 19, Tailwind CSS 4, Lucide Icons, Vite 8, dan React Router 7. Struktur screen dipisahkan menjadi auth, home, dan queue. Context digunakan untuk autentikasi dan notifikasi, hooks digunakan untuk

text-to-speech, sedangkan model dan service memisahkan normalisasi data serta komunikasi API. Struktur tersebut mendukung pemeliharaan dan pengembangan fitur secara modular (Lucide, 2026; React, 2026; React Router, 2026; Tailwind CSS, 2026; Vite, 2026).

Hasil Pelatihan: Analisis Pretest dan Postest

Pengukuran pemahaman peserta dilakukan dengan pretest sebelum kegiatan dan postest setelah kegiatan. Hasilnya dirangkum pada Tabel 5.

Tabel 5.
Hasil pretest dan postest peserta (n = 20)

Aspek yang diukur	Rata-rata pretest	Rata-rata postest	Selisih	N-Gain
Konsep sistem antrean digital	60	85	25	0,63
Analisis kebutuhan dan perancangan (UML, ERD)	55	80	25	0,56
Dasar React dan komponen antarmuka	50	85	35	0,70
Integrasi REST API dan status antrean	45	80	35	0,64
Rata-rata keseluruhan	52,50	82,50	30	0,63

Berdasarkan data evaluasi yang tersedia, nilai rata-rata pretest peserta adalah 52,50, sedangkan nilai rata-rata posttest adalah 82,50. Dengan demikian, terjadi kenaikan sebesar 30 poin. Perhitungan N-Gain menghasilkan nilai 0,63, yaitu (82,50 – 52,50)/(100 – 52,50). Nilai N-Gain positif menunjukkan bahwa hasil posttest lebih tinggi daripada pretest sehingga dapat dikategorikan sebagai peningkatan dengan kategori sedang.

Analisis Ketercapaian Kegiatan

Ketercapaian kegiatan dinilai dengan membandingkan target luaran dengan capaian yang diperoleh, sebagaimana ditunjukkan pada Tabel 6.

Tabel 6.
Ketercapaian target luaran PkM

Target luaran	Indikator	Capaian
Prototipe QueueGo	Alur pelanggan dan operator berjalan pada antarmuka, dari login hingga riwayat	Tercapai (Gambar 7–10)
Dokumen perancangan sistem	Use case, activity diagram, flowchart, dan ERD tersusun	Tercapai (Gambar 4–5)
Peningkatan pemahaman peserta	Perbandingan nilai rata-rata pretest dan posttest peserta	Nilai rata-rata berubah dari 52,50 menjadi 82,50 (naik 30 poin; N-Gain 0,63)
Umpan balik peserta dan mitra	Kuesioner dan diskusi terlaksana	20 dari 20 siswa menyatakan puas (100%); masukan sekolah: fasilitasi perusahaan industri untuk menampung siswa yang lolos bootcamp sebagai peserta magang
Artikel PkM	Naskah tersusun	Tercapai (naskah ini)

Dokumentasi kegiatan	Foto dan dokumen kegiatan tersedia	Tercapai (Gambar 1)

Umpan Balik Peserta dan Mitra

Hasil kuesioner menunjukkan bahwa 20 dari 20 siswa (100%) menyatakan puas terhadap kegiatan. Pihak sekolah memberikan masukan agar tersedia perusahaan industri yang dapat menampung siswa-siswi SMKN 24 Jakarta yang lolos bootcamp untuk mengikuti program magang. Masukan tersebut menjadi bahan tindak lanjut kemitraan dengan dunia industri agar peserta yang telah menyelesaikan bootcamp memperoleh kesempatan menerapkan kompetensi yang dipelajari.

Evaluasi Prototipe dan Rencana Tindak Lanjut

Pada tahap ini, evaluasi prototipe bersifat fungsional berbasis rancangan dan struktur kode. package.json menyediakan perintah dev, build, lint, dan preview, serta source code menyediakan fallback akun lokal ketika backend tidak dapat diakses. Keberadaan fungsi tersebut tidak dapat diposisikan sebagai bukti kuantitatif terhadap performa, keamanan, atau usability (Fauzi & Suhaimi, 2026; OWASP Foundation, 2025). Pengujian kuantitatif terhadap performa, usability, reliabilitas, dan keamanan belum dilakukan dan direncanakan sebagai tindak lanjut sebagaimana pada Tabel 7.

Tabel 7.
Rencana pengujian lanjutan QueueGo

Aspek	Metode pengujian	Indikator
Fungsional	Black-box testing pada skenario alur status antrean	Seluruh transisi status berjalan sesuai rancangan
Performa	Pengukuran waktu respons dan uji beban pada API	Waktu respons pengambilan dan pemanggilan antrean
Usability	System Usability Scale (SUS) pada pelanggan dan operator	Skor SUS
Reliabilitas	Uji ketahanan koneksi dan operasi jangka Panjang	Tingkat kegagalan permintaan dan pemulihan
Keamanan	Pengujian berdasarkan panduan OWASP	Kerentanan autentikasi, kontrol akses, dan perlindungan data
Akurasi urutan	Uji urutan antrean per loket	Kesesuaian urutan pemanggilan

Dengan demikian, QueueGo telah memiliki alur end-to-end pada sisi frontend: autentikasi → pemilihan loket → pembuatan antrean → tiket digital → pemantauan → pengelolaan operator → perubahan status → riwayat. Integrasi API telah disiapkan melalui endpoint yang terpisah untuk setiap transisi status, sehingga menjadi dasar yang memadai untuk dikembangkan menjadi sistem pelayanan antrean multi-loket (QueueGo, 2026).

SIMPULAN

Kegiatan Pengabdian kepada Masyarakat (PKM) di SMKN 24 Jakarta Timur pada 7–18 September 2026 berhasil dilaksanakan melalui tahapan pretest, pelatihan/workshop, posttest, dan evaluasi umpan balik. Kegiatan ini menghasilkan dokumen perancangan sistem (*use case diagram* dan *activity diagram*) serta prototipe aplikasi antrean multi-loket QueueGo berbasis React dengan pemisahan fungsi pelanggan dan operator—mencakup autentikasi, akses tamu, pemilihan loket, tiket digital, pemantauan status antrean, panel kendali operator, pengumuman suara, dan riwayat layanan.

Evaluasi hasil pelatihan menunjukkan peningkatan pemahaman peserta dengan nilai rata-rata pretest sebesar 52,50 menjadi 82,50 pada posttest (kenaikan 30 poin dengan nilai N-Gain 0,63 pada kategori sedang). Selain itu, seluruh peserta (100%) menyatakan puas terhadap pelaksanaan kegiatan, disertai rekomendasi tindak lanjut berupa kerja sama magang industri bagi siswa yang lulus bootcamp serta pengujian kuantitatif pada lingkungan operasional nyata (aspek performa, *usability*, reliabilitas, dan keamanan API).

UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada dosen Politeknik Jakarta Internasional dan Universitas M.H. Thamrin, Kepala Sekolah, Wakil Kepala Sekolah, guru-guru, dan siswa/siswi SMKN 24 yang mengikuti bootcamp, beserta pihak-pihak yang mendukung pelaksanaan kegiatan pengabdian kepada masyarakat serta perancangan dan pengembangan QueueGo yang menjadi konteks kegiatan ini.

REFERENSI

1. Azzikri, F., Sabina, R. L., Yuniarti, Y., Azizah, A. W., Rizki, M. F., & Kurniawan, T. R. (2026). User needs analysis in HostCare application development as a real-time digital queue system in healthcare facilities. Jurnal Ilmiah Manajemen, Ekonomi, & Akuntansi (MEA), 10(2). https://doi.org/10.31955/mea.v10i2.7778
2. Cahyani, F. R., Syamsuddin, S., Hadi, S., & Adda, H. W. (2024). Queuing system in the health sector in Indonesia. Journal of Humanities, Community Service, and Empowerment, 1(4), 125–133. https://doi.org/10.58857/JHCSE.2024.v01.i04.p02

3. Dale, L. C. (2025a). Design and development of a digital attendance software application with data security for Jakarta International Polytechnic. Journal of Information Technology and Informatics Engineering, 1(2), 1–12. https://journal.jci.co.id/jitie/article/view/220
4. Dale, L. C. (2025b). Design and develop the deauthentication attack detection system on Wi-Fi networks based on Python and TCPDUMP. Multidisciplinary Journal of Systemic and Innovative Research, 2(4), 131–138. https://journalwbl.com/index.php/mjsis/article/view/478
5. Fauzi, N. A. M., & Suhaimi, N. H. S. (2026). Design and evaluation of QKlinik: A web-based queue management system for outpatient clinics. International Journal of Software Engineering and Computer Systems, 11(2), 147–159. https://doi.org/10.15282//ijsecs.11.2.2025.12.0144
6. Jo, E. H., Chi, O. H., & Kim, J. (2025). How waiting decisions are made: A comparison of restaurant waiting equity and behavioral intentions in physical and virtual queues. Journal of Hospitality and Tourism Management, 63, 140–152. https://doi.org/10.1016/j.jhtm.2025.03.009
7. Kumar, S., & Gupta, D. (2023). Applying agile and Scrum in healthcare information systems development. Healthcare Informatics Research, 29(4), 297–306. https://doi.org/10.4258/hir.2023.29.4.297
8. Kusuma, Y. A., & Avianto, D. (2024). Enhancing hospital efficiency with Healthyline: A digital solution for registration, appointment reminders, and queue management. Asian Journal of Research in Computer Science, 17(12), 252–271. https://doi.org/10.9734/ajrcos/2024/v17i12542
9. Matte, S., Ninawe, M., Madavi, A., Parate, L., & Nasare, P. (2025). Automated meeting request and queue management system for efficient administrative coordination. International Journal of Electrical, Electronics and Computer Systems, 14(1), 63–67. https://doi.org/10.65521/ijeecs.v14i1.346
10. Matte, S., Ninawe, M., Madavi, A., Parate, L., & Raut, A. (2025). Review on automated meeting request and queue management system for efficient administrative coordination. International Journal of Electrical, Electronics and Computer Systems, 14(1), 68–71. https://doi.org/10.65521/ijeecs.v14i1.347

11. Mustafa, M., Parad, M., Shaikh, A., Shaikh, Z., & Nalawade, A. (2025). PrintFlow: A cloud-based queue management system for automated printing services. International Journal of Computer Applications, 187(11), 30–35. https://doi.org/10.5120/ijca2025925072
12. Naumov, S., & Oliva, R. (2025). Structural feedback and behavioral decision making in queuing systems: A hybrid simulation framework. European Journal of Operational Research, 324(3), 855–870. https://doi.org/10.1016/j.ejor.2025.02.010
13. Okunade, O. A., Osunade, O., Omilabu, A. A., Akande, A. N., & Olanrewaju, B. S. (2024). Queue management application for healthcare providers. Covenant Journal of Informatics and Communication Technology, 12(1).
14. Palupi, I., Khairi, M. D., & Wahyudi, B. A. (2025). Efficient queue management system for Rumah Sehat Dokter Zoji Clinic in Cianjur. Jurnal Abdimas, 29(1). https://doi.org/10.15294/eawwtv27
15. Park, H., Lin, C., & Ahmad, M. (2024). Simulation-based optimization of outpatient flow using web-based queuing tools. PLOS ONE, 19(2), e0295112. https://doi.org/10.1371/journal.pone.0295112
16. Prabaharan, T., & Thishopana, B. (2024). Optimizing queue management in healthcare settings: Enhancing patient satisfaction through strategic approaches. Asian Journal of Economics, Business and Accounting, 24(5), 525–534. https://doi.org/10.9734/ajeba/2024/v24i51327
17. Vasenko, K., & Krasnoshlyk, N. (2023). Web-oriented electronic queue management system. Cherkasy University Bulletin: Applied Mathematics. Informatics. https://doi.org/10.31651/2076-5886-2023-1-10-26
18. Wang, T., Wu, D., & Yang, J. (2025). Efficiency with consent: Permutable queueing in on-demand services. Omega. https://doi.org/10.1016/j.omega.2025.103361
19. Yudertha, A., Abdillah, Z., Silky, K., Muttakin, Z., & Padila, A. Z. N. (2026). Design and development of a web-based customer queue information system on a local area network using PHP (case study: PT BPD Jambi Sharia Branch). Jurnal Teknologi dan Open Source, 9(1), 280–289. https://doi.org/10.36378/jtos.v9i1.5730
20. Yuniati, R., Fahrudda, A., Suswojo, H., & Sari, F. I. (2026). Waiting time and patient satisfaction in Mobile JKN cardiology services. Indonesian Journal of Innovation Studies, 27(2). https://doi.org/10.21070/ijins.v27i2.1996

Training on the Design of the React-Based 'QueueGo' Digital Queue Management System for Multi-Counter Services at SMKN 24 East Jakarta

ORIGINALITY REPORT

12%	12%	4%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	journals.mriindia.com Internet Source	2%
2	journal.stiemb.ac.id Internet Source	1%
3	aaltodoc.aalto.fi Internet Source	1%
4	sejurnal.com Internet Source	1%
5	ejournal.uniks.ac.id Internet Source	1%
6	Submitted to Washington University of Science and Technology Student Paper	1%
7	Submitted to University of Hertfordshire Student Paper	1%
8	Submitted to University of New South Wales Student Paper	1%
9	Submitted to University of Texas Health Science Center Student Paper	1%
10	researchhub.id Internet Source	1%
11	journalwbl.com Internet Source	1%
12	ami-ejournal.cdu.edu.ua Internet Source	1%
13	uby.ac.id Internet Source	1%
14	Submitted to University of Wales Institute, Cardiff Student Paper	1%

Exclude quotes	Off	Exclude matches	< 1%
Exclude bibliography	Off		