

PENGEMBANGAN APLIKASI DEWAN KELUARGA MASJID (DKM) REGISTRATION SYSTEM MENGGUNAKAN METODE WATERFALL

Muhamad Adillah Fatih¹⁾, Reza Arif Maulana²⁾, Rizki Reza Pratama³⁾, Muhammad Darwis⁴⁾, Retno Hendrowati⁵⁾

^{1,2,3,4,5}Program Studi Teknik Informatika

^{1,2,3,4,5}Fakultas Ilmu Rekayasa

^{1,2,3,4,5}Universitas Paramadina

E-mail: muhamad.fatih@students.paramadina.ac.id¹⁾, reza.maulana@students.paramadina.ac.id²⁾,

rizki.pratama@students.paramadina.ac.id³⁾, muhammad.darwis@paramadina.ac.id⁴⁾,

retno.hendrowati@paramadina.ac.id⁵⁾

Abstract

The Mosque Family Council (DKM) Paramadina regularly holds various events or activities such as the Prophet Muhammad's Maulid event, Qurban, to the Religious Festival. One of the problems often faced by DKM Paramadina is that registration and attendance data are not synchronized, making it difficult to ensure that participants are entitled to event benefits. To overcome this problem, DKM Paramadina needs an effective and efficient registration and attendance system in ensuring data consistency. The purpose of this study is to improve and develop an event registration and attendance recording system using the Waterfall Method approach. The result of this study is a web-based DKM Registration System, which allows the use of recording and monitoring the attendance of event participants efficiently. This system utilizes Next.js Framework for Frontend, Express.js for Backend, and Node.js as runtime. To store registration and attendance data, this system is integrated with Google Spreadsheet by utilizing Google Application Programming Interface (API). In addition, Node Mailer is also used to send a QR Code to the registrant's email for attendance. The Blackbox method has been used to test the effectiveness of the DKM Registration System and the results show good performance. This research provides a solution to overcome conventional problems in data management, ensuring registration and attendance data remain synchronized and consistent.

Keywords - Web Based Application, Registration and Attendance System, Waterfall Method, Blackbox Method.

Intisari

Dewan Keluarga Masjid (DKM) Paramadina rutin mengadakan berbagai acara atau kegiatan seperti acara Maulid Nabi Muhammad, Qurban, hingga Festival Keagamaan. Salah satu permasalahan yang sering dihadapi oleh DKM Paramadina adalah data registrasi dan absensi tidak sinkron, sehingga sulit untuk memastikan peserta yang berhak mendapatkan benefit acara. Untuk mengatasi masalah tersebut, DKM Paramadina membutuhkan sistem registrasi dan absensi yang efektif dan efisien dalam memastikan konsistensi data. Tujuan dari studi ini adalah memperbaiki dan mengembangkan sistem pendaftaran dan pencatatan kehadiran acara dengan menggunakan pendekatan Metode Waterfall. Hasil penelitian ini adalah DKM Registration System berbasis web, yang memungkinkan penggunaan pencatatan dan pemantauan kehadiran peserta acara secara efisien. Sistem ini memanfaatkan Framework Next.js untuk Frontend, Express.js untuk Backend, dan Node.js sebagai runtime. Untuk menyimpan data registrasi dan absensi, sistem ini terintegrasi dengan Google Spreadsheet dengan memanfaatkan Google Application Programming Interface (API). Selain itu, Node Mailer juga digunakan untuk mengirimkan QR Code ke email pendaftar untuk absensi. Metode Blackbox telah digunakan untuk menguji keefektifan DKM Registration System dan hasilnya menunjukkan kinerja yang baik. Penelitian ini memberikan solusi mengatasi permasalahan konvensional dalam pengelolaan data, memastikan data registrasi dan absensi tetap sinkron dan konsisten.

Kata Kunci—Aplikasi Berbasis Web, Sistem Registrasi dan Absensi, Metode Waterfall, Metode Blackbox.

1. PENDAHULUAN

Dewan Keluarga Masjid Universitas Paramadina (DKM Paramadina) merupakan salah satu organisasi atau unit kegiatan mahasiswa yang aktif dalam aspek sosial dan

keagamaan. Seperti organisasi pada umumnya, DKM Paramadina memiliki program kerja yang dilaksanakan rutin setiap tahun. Beberapa program kerja rutin yang dilaksanakan setiap tahun tersebut diantaranya Acara Maulid Nabi Muhammad, Acara Qurban, Festival Keagamaan,

dan sebagainya. Beberapa program kerja tersebut merupakan bagian dari program kerja besar yang bentuknya acara dan sifatnya terbuka untuk umum[1]. Untuk program kerja semacam itu, DKM Paramadina biasanya menggunakan sistem registrasi dan absensi untuk manajemen data peserta acara. Data registrasi dan absensi peserta juga penting kaitannya dengan *benefit* acara yang akan diberikan kepada peserta. Biasanya *benefit* acara berupa sertifikat apresiasi yang diberikan kepada peserta atas kehadirannya di acara tersebut.

Namun, sistem registrasi dan absensi acara yang saat ini digunakan oleh DKM Paramadina masih konvensional, dimana kertas menjadi bagian penting dalam proses tersebut. Sistem registrasi dan absensi yang masih konvensional memunculkan beberapa masalah kasual. Masalah seperti data registrasi dan absensi tidak sinkron, dimana peserta dapat melakukan absensi tanpa melakukan registrasi. Selain itu, absensi biasanya dilakukan dengan memberikan tautan *Google Form* baru yang berbeda dengan tautan registrasi. Ini juga memunculkan masalah, dimana peserta yang tidak menghadiri acara tetap dapat melakukan absensi dan mendapatkan *benefit* acara. Sistem registrasi dan absensi konvensional tidak efisien sehingga menyebabkan hilangnya waktu dan penumpukan orang selama proses registrasi serta memerlukan penggunaan media mahal seperti kertas[2], [3]. Sistem registrasi dan absensi memiliki kekurangan juga dalam hal pengelolaan data seperti kesalahan input data, inkonsistensi data registrasi dengan absensi, yang mempengaruhi kualitas informasi dan pelayanan[4]. Ini merupakan masalah yang merepotkan dalam proses distribusi *benefit* acara oleh DKM Paramadina kepada peserta karena harus memberikan waktu dan usaha lebih untuk melakukan proses sinkronisasi data registrasi dan absensi.

Nopriandi[5] telah melakukan penelitian sebelumnya yang berfokus pada pengembangan sistem informasi registrasi mahasiswa baru dengan memanfaatkan Program *Visual Basic*. Tujuan dari sistem ini adalah mengatasi tantangan dalam pengolahan sistem informasi registrasi mahasiswa di Sekolah Tinggi, terutama pada

aspek administrasi, dengan memberikan solusi berupa penyediaan informasi yang cepat dan akurat dalam pengelolaan data mahasiswa. Sistem ini dapat membantu mengurangi penggunaan sumber daya manusia tanpa mengorbankan kualitas layanan, dan juga membantu mengatasi kendala-kendala terkait dengan aspek biaya dan waktu yang digunakan.

Maka dari itu, perlu ada pembaharuan pada proses registrasi dan absensi acara. Penulis mengembangkan aplikasi registrasi dan absensi terpadu menggunakan *Google Spreadsheet* sebagai basis data. Sistem ini mengirimkan *QR Code* unik ke email pendaftar untuk absensi, yang kemudian dipindai oleh sistem pemindai. Pemindaian *QR Code* memastikan konsistensi data dengan mengamankan informasi dalam *QR Code*. Sistem ini dirancang untuk mengatasi masalah ketidaksesuaian data dan memastikan bahwa hanya peserta yang telah mendaftar yang dapat melakukan absensi sehingga berhak mendapatkan *benefit* dari acara yang diselenggarakan.

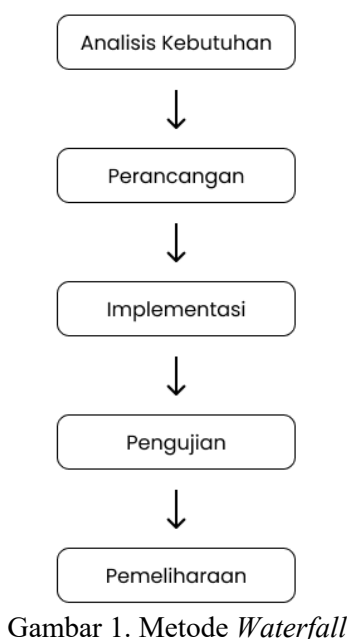
Hasil penelitian ini adalah aplikasi *DKM Registration System* berbasis web yang memungkinkan DKM Paramadina mencatat dan mengabsensi peserta dalam acara mereka. Bedanya dengan sistem registrasi sebelumnya oleh Nopriandi adalah fokusnya pada pengelolaan data registrasi dan absensi peserta acara dilakukan dengan pendekatan yang berbeda. *DKM Registration System* menggunakan *QR Code* yang akan dipindai oleh kamera pemindai untuk mencatat kehadiran peserta. Penggunaan *QR Code* dalam sistem absensi ini dapat mengurangi kemungkinan registrasi dan absensi palsu serta meningkatkan keamanan[6]. *DKM Registration System* dikembangkan dengan metode *Software Development Life Cycle* tipe *Waterfall* karena kecocokannya dengan karakteristik yang berurutan dan linier. Metode ini tidak hanya membantu pengembangan yang cepat dan efisien, tetapi juga mengurangi segmentasi berlebihan. Selain itu, metode *Waterfall* sangat sesuai untuk pengembangan aplikasi atau sistem dengan spesifikasi yang telah ditetapkan[7].

Sebagai gambaran mengenai alur artikel ini, setelah bagian ini, penulis menjelaskan secara rinci proses uji coba dan metode yang diterapkan dalam pengembangan *DKM Registration System*. Langkah-langkah praktis ini akan diikuti dengan

hasil penelitian yang diperoleh, diikuti dengan analisis dan diskusi terhadap temuan penulis. Terakhir, artikel ini akan ditutup dengan kesimpulan dan implikasi dari penelitian ini, serta saran untuk penelitian lanjutan dalam bidang ini.

2. METODOLOGI

Implementasi sistem ini menggunakan pendekatan SDLC (*Software Development Life Cycle*) model *Waterfall* karena karakteristiknya yang berurutan dan linier, sesuai untuk pengembangan aplikasi atau sistem dengan spesifikasi yang tetap. SDLC adalah proses yang terdefinisi dengan baik dan terorganisir yang digunakan untuk merencanakan, mengembangkan, menerapkan, dan memelihara sistem perangkat lunak berkualitas tinggi[8]. SDLC melibatkan berbagai tahap seperti perencanaan, analisis, desain, dan eksekusi. Banyak model SDLC, termasuk *Waterfall*, *Spiral*, dan *V-model*, yang telah dirancang. SDLC berisi rencana rinci tentang bagaimana perangkat lunak akan dikembangkan, dimodifikasi, dipelihara, dan diperbaharui. SDLC digunakan untuk menyelesaikan proses pengembangan perangkat lunak tepat waktu dan mengendalikan kualitas perangkat lunak. Model siklus hidup pengembangan perangkat lunak mendekati proses pengembangan perangkat lunak langkah demi langkah. Jika prosesnya kuat, produk akhir akan kuat, dan proyek akan sukses[9].



Gambar 1. Metode *Waterfall*

Model *Waterfall* adalah salah satu pendekatan dalam pengembangan perangkat lunak (SDLC). Metode ini mengikuti serangkaian langkah berurutan biasa disebut *Linear Sequential Model*, dimulai dari analisis kebutuhan, perancangan, implementasi, pengujian, hingga pemeliharaan. Dalam pendekatan ini, setiap fase harus selesai sebelum melangkah ke fase berikutnya[10], [11]. Tahapan metode *Waterfall* sebagai berikut:

1. Analisis kebutuhan: pada tahap ini, dilakukan pengumpulan informasi mengenai kebutuhan pengguna dan persyaratan sistem yang akan dibuat.
2. Perancangan: tahap ini melibatkan perancangan sistem secara keseluruhan, termasuk arsitektur sistem, arsitektur *database*, dan desain antarmuka pengguna.
3. Implementasi: tahap ini melibatkan pembuatan sistem berdasarkan desain yang telah dibuat pada tahap sebelumnya.
4. Pengujian: tahap ini melibatkan pengujian sistem untuk memastikan bahwa sistem berjalan dengan baik dan sesuai dengan kebutuhan pengguna.
5. Pemeliharaan: tahap ini melibatkan pemeliharaan sistem setelah sistem telah diimplementasikan dan diuji. Pemeliharaan meliputi perbaikan bug, peningkatan kinerja, serta penambahan fitur baru[10], [11], [12].

2.1 Analisis Kebutuhan

Pada tahap ini, penulis melakukan pengumpulan informasi tentang kebutuhan pengguna dan kebutuhan sistem yang akan dikembangkan. Proses ini melibatkan wawancara dengan pihak terkait untuk memahami secara mendalam kebutuhan peneliti. Wawancara dilakukan dengan anggota DKM Paramadina yang memiliki jabatan strategis seperti Ketua Umum dan Kepala-Kepala Divisi. Wawancara penting untuk analisis persyaratan dalam pengembangan sistem karena membantu mengidentifikasi fitur agar sesuai dengan kebutuhan[13].

Wawancara ini difokuskan pada pemahaman tentang kesulitan dan hambatan yang sering dihadapi dalam proses registrasi dan pencatatan kehadiran pada acara-acara sebelumnya. Wawancara dilakukan secara tatap muka atau

melalui komunikasi daring, tergantung pada ketersediaan dan preferensi responden. Durasi wawancara bervariasi, dengan rata-rata waktu sekitar 30-45 menit per sesi. Data dari wawancara dicatat dengan teliti dan dianalisis untuk mengidentifikasi pola-pola umum dalam kebutuhan pengguna dan sistem. Hasil analisis ini menjadi dasar untuk merumuskan spesifikasi sistem yang akurat dan sesuai dengan kebutuhan.

2.2 Perancangan

Dalam tahap perancangan, penulis merinci kebutuhan sistem secara menyeluruh. Ini mencakup arsitektur sistem, arsitektur database, dan desain antarmuka pengguna. Dalam tahap ini, penulis menerapkan *Unified Modeling Language* (UML). UML adalah bahasa standar untuk pemodelan visual yang digunakan dalam perancangan, dokumentasi, dan pemodelan sistem perangkat lunak. UML menawarkan kumpulan simbol visual yang digunakan untuk mengilustrasikan struktur, fungsi, serta rancangan perangkat lunak. Simbol-simbol ini mencakup representasi visual seperti diagram aktivitas, diagram *use case*, diagram kelas, dan berbagai jenis diagram lainnya[14], [15]. UML memiliki potensi sebagai cetak biru yang sangat komprehensif dan rinci. Dalam peran ini, UML dapat memberikan informasi yang sangat terperinci mengenai kode sumber untuk pemahaman program, *debugging*, pemeliharaan, bahkan memungkinkan untuk membaca kode program dan mengubahnya ke dalam bentuk diagram[16], [17].

2.3 Implementasi

Tahap implementasi melibatkan pembuatan aplikasi sesuai dengan desain sistem yang telah dirancang sebelumnya. Framework yang digunakan mencakup *Next.js* untuk *front-end*, *Express.js* untuk *back-end*, serta *Node.js* sebagai *runtime*. Layanan *Google Spreadsheet* digunakan untuk menyimpan data registrasi dan absensi peserta dengan memanfaatkan *Google Application Programming Interface (API)* agar dapat terkoneksi. *Node Mailer* dimanfaatkan untuk mengirim *QR Code* melalui *email* kepada peserta yang telah mendaftar untuk *event* dan akan digunakan sebagai absensi.

a. Next.js

Next.js adalah sebuah kerangka kerja (*framework*) *React* yang membantu dalam pengembangan aplikasi web dengan fitur-fitur seperti *routing* bawaan, dukungan CSS terintegrasi, komponen tata letak (*layout*), serta optimisasi gambar dan font. Dirancang untuk mengatasi masalah-masalah yang sering dihadapi dalam pengembangan aplikasi web, *Next.js* memungkinkan penggunaan teknik *prerendering* dan *server-side rendering* untuk mempercepat waktu muat aplikasi. Dengan kemampuan untuk mengoptimalkan kinerja melalui *code splitting*, *prefetching*, dan *lazy loading*, *Next.js* memberikan pengembangan yang efisien. Sumber daya seperti dokumentasi lengkap dan kursus interaktif juga tersedia untuk mempelajari dan menguasai penggunaan *Next.js* dengan baik[18].

b. Express.js

Express.js adalah sebuah *framework back-end* untuk *Node.js* yang sangat populer dan memudahkan pengembangan aplikasi web dan API. Dirancang untuk mempermudah proses pembuatan aplikasi dengan *Node.js*, *Express.js* menyediakan alat yang kuat dan sederhana untuk membangun berbagai jenis aplikasi web dan layanan API. Kerangka kerja ini terkenal karena kemampuannya dalam menangani permintaan HTTP, *routing*, *middleware*, pengelolaan permintaan, tanggapan, serta berbagai fitur yang mempercepat pengembangan aplikasi. *Express.js* juga memungkinkan integrasi dengan *middleware* dan berbagai modul *Node.js* lainnya, memperluas fungsionalitasnya secara signifikan[19], [20].

c. Node.js

Node.js adalah lingkungan *runtime open-source* dan *cross-platform* untuk menjalankan kode *JavaScript*. Dirilis pada 2009 oleh Ryan Dahl, *Node.js* digunakan secara luas untuk pemrograman *server-side* dan memungkinkan pengembang membuat aplikasi *front-end* dan *back-end* menggunakan *JavaScript*[21]. Ini berjalan dengan menggunakan mesin *JavaScript V8*, memungkinkan eksekusi kode *JavaScript* di luar lingkungan *browser*[22]. *Node.js* populer untuk berbagai jenis proyek dan memfasilitasi pengembangan aplikasi yang efisien dan skalabel[23], [24]. *Node.js* adalah dasar bagi *Express.js* dan *Next.js*. *Express.js* memanfaatkan *Node.js* sebagai *runtime environment* untuk

membangun aplikasi web dengan *JavaScript*. *Next.js*, pada tingkat yang lebih tinggi, menggunakan *Node.js* dan kadang-kadang *Express.js* sebagai fondasi untuk memberikan fitur-fitur yang lebih maju untuk pembangunan aplikasi web yang lebih kompleks dan kaya akan fitur[22], [25].

d. Google Application Programming Interface

Google Application Programming Interface (*Google API*) adalah kumpulan antarmuka pemrograman aplikasi yang dikembangkan oleh *Google*. Antarmuka ini memungkinkan komunikasi dengan layanan-layanan *Google* dan integrasi dengan platform *Google* seperti *Google Cloud*, *Google Documents*, *Google Maps*, dan banyak layanan lainnya. *Google API* menyediakan fungsionalitas seperti analitika, *machine learning* sebagai layanan (seperti *Prediction API*), akses ke data pengguna (dengan izin untuk membaca data), dan banyak lagi[26].

2.4 Pengujian

Pada tahap ini, sistem diuji menggunakan metode *Blackbox* untuk memastikan bahwa sistem beroperasi dengan baik dan memenuhi kebutuhan pengguna. Metode Pengujian *Blackbox* adalah cara untuk mengevaluasi fungsi suatu perangkat lunak dengan melihatnya dari luar, tanpa memeriksa rincian internal struktur atau logikanya. Pendekatan ini meninjau bagaimana aplikasi merespons *input*, *output*, dan situasi tertentu, tanpa memerlukan pengetahuan mendalam tentang kode sumber atau rancangan aplikasi. Menurut referensi dari berbagai jurnal ilmiah, pengujian ini berfokus pada kebutuhan pengguna dan memastikan kesesuaian aplikasi dengan spesifikasi yang telah ditetapkan, dilakukan tanpa akses ke detail internal sistem. Sebagai contoh, sebuah penelitian yang dilakukan oleh Dewi et al. dalam artikel "Pengujian Aplikasi Doctor to Doctor Menggunakan Metode *Blackbox Testing*" [27] menyoroti penerapan *Blackbox Testing* pada aplikasi web, khususnya dalam tahap *staging*. Referensi lainnya juga menunjukkan bahwa *Blackbox Testing* berguna untuk menemukan *bug* atau kesalahan tanpa memerlukan akses ke internal aplikasi, memastikan aplikasi layak

digunakan tanpa perlu memahami detail implementasinya[28], [29], [30].

2.5 Pemeliharaan

Walaupun tahap pemeliharaan merupakan tahap akhir pada metode *Waterfall*, penelitian ini berhenti di tahap pengujian karena setelah pengujian tujuan dari perancangan sistem ini sudah terpenuhi dan menjawab permasalahan.

3. HASIL DAN PEMBAHASAN

3.1 Analisis Kebutuhan

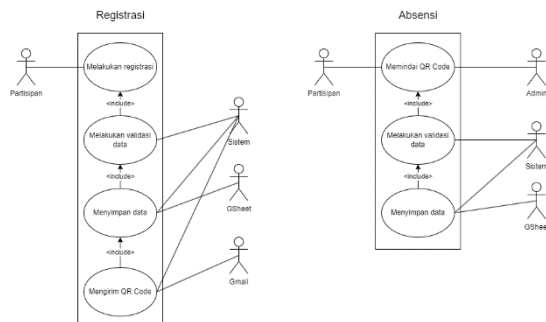
Penulis menetapkan serta membatasi kebutuhan sistem berdasarkan informasi yang diperoleh melalui wawancara dengan pihak terkait. Pembatasan kebutuhan terletak pada data apa saja yang dibutuhkan oleh DKM Paramadina terkait manajemen data peserta acara atau kegiatan. Data tersebut antara lain Email, Nama, Institusi, dan Role (*Lecturer*, *Students*, dan *Other*) peserta. Data ini merupakan data standar yang dibutuhkan oleh DKM Paramadina untuk manajemen data peserta acara atau kegiatan.

3.2. Perancangan

Dalam proses perancangan, penulis menggambarkan UML melalui beberapa diagram model sebagai desain dari sistem yang akan diimplementasikan.

a. Use Case Diagram

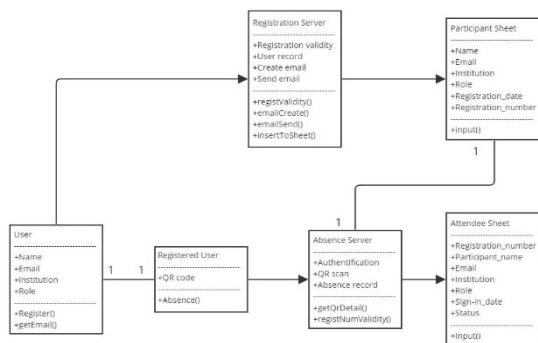
Pada *Use Case Diagram*, penulis mengidentifikasi kasus dan aktor untuk setiap kasus. Selanjutnya, merancang use case pada tiap kasusnya dan menentukan interaksi antara aktor dengan use case sistemnya. Terdapat dua kasus, yaitu Registrasi dan Absensi. Untuk kasus Registrasi, terdapat empat use case (registrasi, validasi data, menyimpan data, mengirim *QR Code*), dan empat aktor (Partisipan, Sistem, Gsheet, Gmail). Untuk kasus Absensi, terdapat tiga use case (memindai *QR Code*, validasi data, dan menyimpan data). Untuk rancangan diagram yang menjelaskan bagaimana interaksi antara aktor dengan use case, ditunjukkan pada Gambar 2.



Gambar 2. Use Case Diagram

b. Class Diagram

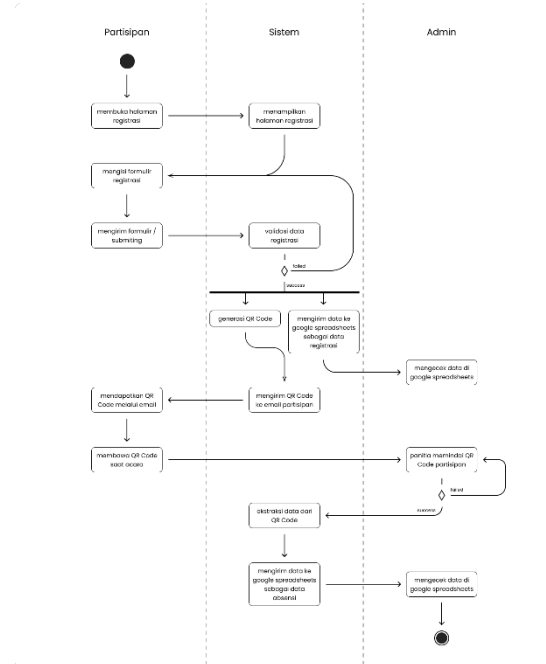
Selanjutnya penulis merancang *Class Diagram*. Diagram ini mendefinisikan seluruh class yang terdapat dalam sistem yang akan diimplementasikan. Dalam sistem ini, terdapat enam class (*User*, *Registered User*, *Registration Server*, *Absence Server*, *Participant Sheet*, dan *Attendee Sheet*) yang saling berhubungan. Hubungan antar class ditunjukkan pada Gambar 3.



Gambar 3. Class Diagram

c. Activity Diagram

Diagram terakhir dari penerapan UML pada sistem ini adalah *Activity Diagram*. *Activity Diagram* yang dibuat ini berfungsi untuk memetakan bagaimana aktivitas atau proses dari setiap aktor dan use case pada setiap kasusnya. Gambar 4 menunjukkan Activity Diagram yang dirancang untuk sistem ini.



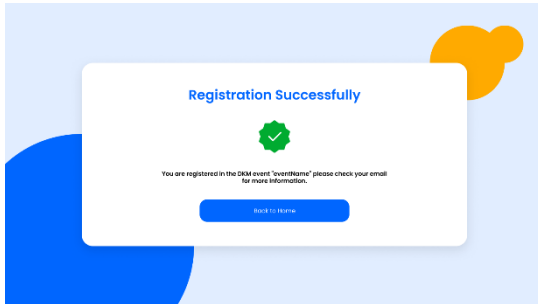
Gambar 4. Activity Diagram

3.3 Implementasi

Implementasi dari sistem ini direpresentasikan oleh tampilan antarmuka (UI) yang dibuat setelah melakukan perancangan sistem. Implementasi tentunya sesuai dengan apa yang sudah dirancang dan sesuai dengan *use case* yang sudah didefinisikan.

Gambar 5 merupakan tampilan dari halaman registrasi. Terdapat formulir dari registrasi yang terdiri dari email, nama lengkap, usia, nomor hp, jenis kelamin, dan asal instansi. Jika salah satu formulir tersebut tidak diisi maka akan muncul pesan kesalahan dan akan diminta untuk mengisi formulir dengan benar. Jika berhasil melakukan registrasi, maka akan diarahkan ke halaman yang menampilkan pesan sukses registrasi yang ditunjukkan oleh Gambar 6.

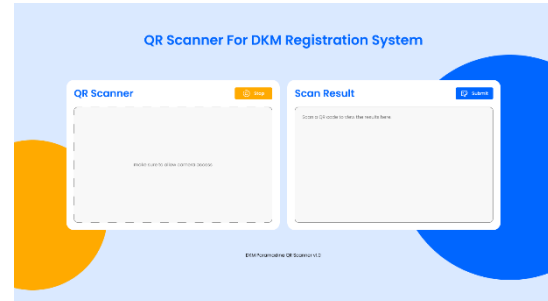
Gambar 5. Halaman Registrasi



Gambar 6. Halaman Sukses Registrasi

Setelah berhasil melakukan registrasi, maka email pendaftar akan dikirim QR Code yang berisikan data-data registrasi. QR Code yang diterima oleh peserta di-generate menggunakan *Google Chart API*. Isi dari QR Code yang sudah di-generate merupakan data yang dimasukkan oleh peserta ketika melakukan registrasi, dan tanda pengenal yang akan dikenali oleh pemindai QR (*QR Scanner*) yang ditunjukkan pada Gambar 7. Agar QR Code dapat di-generate, pendaftar diharuskan mengisi tiap kolom yang tersedia pada laman registrasi. QR Code tidak dapat di-generate jika pada saat pengisian data peserta masih terdapat kesalahan, seperti format *email*

yang keliru atau terdapat kolom yang belum diisi. QR Code tersebut berfungsi sebagai objek yang dipindai oleh Sistem *QR Scanner* untuk direkam datanya dan disimpan sebagai data absensi di *Google Spreadsheet*.



Gambar 7. Pemindai QR (QR Scanner)

3.4 Pengujian

Sampai tahap ini, penulis melakukan pengujian kepada sistem yang dibuat. Pengujian ditujukan agar aplikasi berfungsi sebagaimana mestinya sesuai dengan apa yang dirancang. Pengujian dilakukan dengan metode *Blackbox*. Hasil pengujian ditampilkan pada Tabel 1.

Tabel 1. Pengujian Blackbox

Kelas Uji	Kondisi	Langkah	Hasil Diharapkan	Hasil Sebenarnya	Keterangan
Registrasi	Cek apakah berhasil melakukan registrasi dengan data yang valid	<ol style="list-style-type: none"> 1. Membuka halaman registrasi 2. Mengisi setiap kolom formulir 3. Klik tombol 'Submit' 4. Redirect ke halaman sukses registrasi 	Registrasi Berhasil	Registrasi Berhasil	Sistem akan melakukan validasi data dan menyimpan ke <i>Google Spreadsheet</i> sebagai data registrasi
Registrasi	Cek apakah berhasil melakukan registrasi dengan data yang tidak valid	<ol style="list-style-type: none"> 1. Membuka halaman registrasi 2. Mengosongkan salah satu kolom formulir 3. Klik tombol 'Submit' 	Warning untuk mengisi kolom yang kosong	Warning untuk mengisi kolom yang kosong	Sistem akan melakukan validasi data dan mengirim notifikasi <i>warning</i>

Absensi	Cek apakah <i>QR Code</i> yang valid berhasil dipindai	<ol style="list-style-type: none"> 1. Membuka halaman pemindai QR 2. Klik tombol 'Scan' 3. Menunjukkan QR ke kamera pemindai 	Kolom 'Scan Result' akan menampilkan data yang dipindai	Kolom 'Scan Result' menampilkan data yang dipindai	Sistem akan melakukan validasi data dan menampilkan data yang tervalidasi
Absensi	Cek apakah <i>QR Code</i> yang tidak valid berhasil dipindai	<ol style="list-style-type: none"> 1. Membuka halaman pemindai QR 2. Klik tombol 'Scan' 3. Menunjukkan QR ke kamera pemindai 	<i>Warning</i> bahwa QR tidak valid	<i>Warning</i> bahwa QR tidak valid	Sistem akan melakukan validasi data dan mengirim <i>warning</i> QR tidak valid
Absensi	Cek apakah data yang valid berhasil disimpan di <i>Google Spreadsheet</i>	<ol style="list-style-type: none"> 1. Membuka halaman pemindai QR 2. Klik tombol 'Scan' 3. Menunjukkan QR ke kamera pemindai 4. Klik tombol 'Submit' 	Notifikasi data berhasil disimpan	Notifikasi data berhasil disimpan	Sistem akan mengirim data ke <i>Google Spreadsheet</i> sebagai data absensi

4. KESIMPULAN DAN SARAN

Kesimpulan penulis dalam penelitian ini adalah bahwa pengembangan sistem registrasi dan absensi acara terintegrasi menggunakan metode *Waterfall* memberikan solusi terhadap masalah konvensional dalam manajemen acara yang dihadapi oleh DKM Paramadina. Sistem baru ini memungkinkan pencatatan data registrasi dan absensi yang lebih terintegrasi dan terkontrol, serta meminimalisir inkonsistensi data dan tindak kecurangan pada saat absensi acara. Dengan adopsi metode *Waterfall*, tahapan pengembangan sistem dapat dilalui secara teratur dan sistematis, sehingga memastikan kualitas dan kehandalan sistem yang dihasilkan.

Selain memperbaiki permasalahan yang diidentifikasi pada pendahuluan, pengujian aplikasi menggunakan metode blackbox menunjukkan keseluruhan fungsi aplikasi telah berjalan dengan baik. Untuk penelitian lanjutan, sistem ini dapat dikembangkan lebih lanjut untuk menjadi sebuah sistem registrasi dan absensi terintegrasi yang dapat dikustomisasi oleh pengguna atau biasa disebut *Content Management System* (CMS).

DAFTAR PUSTAKA

- [1] DKM Paramadina, *Qanun Asasi Dan Qanun Dakhili DKM Paramadina*. Indonesia, 2024.

- [2] H. M. C. P, P. N. Holla K, S. A. Azeem, and M. G. Veena, "A Smart Attendance System based on Machine learning," in *2019 Global Conference for Advancement in Technology (GCAT)*, IEEE, Oct. 2019, pp. 1–7. doi: 10.1109/GCAT47503.2019.8978324.
- [3] R. Perez-Siguas, H. Matta-Solis, E. Matta-Solis, H. Matta-Perez, A. Cruzata-Martinez, and B. Meneses-Claudio, "Management of an Automatic System to Generate Reports on the Attendance Control of Teachers in a Educational Center," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 2, pp. 20–26, Mar. 2023, doi: 10.17762/ijritcc.v11i2.6106.
- [4] P. I. Listyorini, A. Dader, M. Z. Ulhaq, and A. S. Nugraini, "Literature Study: Factors Affecting the Implementation of Hospital Management Information Systems in Hospital," *Proceedings of the International Conference on Nursing and Health Sciences*, vol. 3, no. 1, pp. 167–178, May 2022, doi: 10.37287/picnhs.v3i1.1145.
- [5] H. Nopriandi, "PERANCANGAN SISTEM INFORMASI REGISTRASI MAHASISWA," *JURNAL TEKNOLOGI DAN OPEN SOURCE*, vol. 1, no. 1, pp. 73–79, Jun. 2018, doi: 10.36378/jtos.v1i1.1.
- [6] A. Patel, A. Joseph, S. Survase, and R. Nair, "Smart Student Attendance System Using QR Code," *SSRN Electronic Journal*, 2019, doi: 10.2139/ssrn.3370769.
- [7] S. M. Golodetz, C. Nicholls, I. D. Voiculescu, and S. A. Cameron, "Two tree-based methods for the waterfall," *Pattern Recognit*, vol. 47, no. 10, pp. 3276–3292, Oct. 2014, doi: 10.1016/j.patcog.2014.04.017.
- [8] S. R. Doddaguni, S. Asif S, M. MN, and R. R, "Understanding SDLC using CI/CD Pipeline," *International Journal of Soft Computing and Engineering*, vol. 9, no. 6, pp. 22–25, May 2020, doi: 10.35940/ijscce.F3405.059620.
- [9] R. S. Ghumatkar and A. Date, "Software Development Life Cycle (SDLC)," *Int J Res Appl Sci Eng Technol*, vol. 11, no. 11, pp. 1162–1165, Nov. 2023, doi: 10.22214/ijraset.2023.56554.
- [10] A. P. Wicaksono, D. A. S. Rukmana, and M. R. Pratama, "PEMBUATAN APLIKASI PADA PRAKTIK MANDIRI BIDAN MENGGUNAKAN METODE WATERFALL," *J-REMI : Jurnal Rekam Medik dan Informasi Kesehatan*, vol. 4, no. 2, pp. 97–107, Mar. 2023, doi: 10.25047/j-remi.v4i2.3863.
- [11] Y. Mulyanto, E. S. Susanto, and M. Ilyas, "APLIKASI PENDAFTARAN EKSTRAKURIKULER BERBASIS WEB DENGAN METODE WATERFALL," *Infotronik : Jurnal Teknologi Informasi dan Elektronika*, vol. 8, no. 1, p. 22, Jun. 2023, doi: 10.32897/infotronik.2023.8.1.2736.
- [12] A. M. I. Pratama and U. Chotijah, "Sistem Informasi Kasir Unit Pelayanan Jasa SMKN 1 Cerme Berbasis Website Dengan Metode Waterfall," *Jurnal Ilmiah ILKOMINFO - Ilmu Komputer & Informatika*, vol. 5, no. 2, pp. 60–67, Jul. 2022, doi: 10.47324/ilkominfo.v5i2.149.
- [13] A. Wahbeh, S. Sarnikar, and O. El-Gayar, "A socio-technical-based process for questionnaire development in requirements elicitation via interviews," *Requir Eng*, vol. 25, no. 3, pp. 295–315, Sep. 2020, doi: 10.1007/s00766-019-00324-x.
- [14] F.- Sonata, "Pemanfaatan UML (Unified Modeling Language) Dalam Perancangan Sistem Informasi E-Commerce Jenis Customer-To-Customer," *Jurnal Komunika : Jurnal Komunikasi, Media dan Informatika*, vol. 8, no. 1, p. 22, Jun. 2019, doi: 10.31504/komunika.v8i1.1832.
- [15] R. Pakaya, A. R. Tapate, and S. Suleman, "PERANCANGAN APLIKASI

- PENJUALAN HEWAN TERNAK UNTUK QURBAN DAN AQIQAH DENGAN METODE UNIFIED MODELING LANGUAGE (UML),” *Jurnal Technopreneur (JTech)*, vol. 8, no. 1, pp. 31–40, May 2020, doi: 10.30869/jtech.v8i1.531.
- [16] C. Alvin, B. Peterson, and S. Mukhopadhyay, “Static generation of UML sequence diagrams,” *International Journal on Software Tools for Technology Transfer*, vol. 23, no. 1, pp. 31–53, Feb. 2021, doi: 10.1007/s10009-019-00545-z.
- [17] A. Nanthaamornphong and A. Leatongkam, “Extended ForUML for Automatic Generation of UML Sequence Diagrams from Object-Oriented Fortran,” *Sci Program*, vol. 2019, pp. 1–22, Feb. 2019, doi: 10.1155/2019/2542686.
- [18] Next.js, “Next.js: Docs,” Next.js Documentation. Accessed: Jan. 10, 2024. [Online]. Available: <https://nextjs.org/docs>
- [19] Express, “Express - Node.js web application framework,” Express Website. Accessed: Jan. 10, 2024. [Online]. Available: <https://expressjs.com/>
- [20] S. Jain, “Express.js,” GeeksForGeeks Website. Accessed: Jan. 10, 2024. [Online]. Available: <https://www.geeksforgeeks.org/express-js/>
- [21] B. Semah, “What Exactly is Node.js? Explained for Beginners,” freeCodeCamp Website. Accessed: Jan. 10, 2024. [Online]. Available: <https://www.freecodecamp.org/news/what-is-node-js/>
- [22] Node.js, “Introduction to Node.js,” Node.js Website. Accessed: Jan. 10, 2024. [Online]. Available: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>
- [23] T. Sufiyan, “What is Node.js: A Comprehensive Guide,” Simplilearn Website. Accessed: Jan. 10, 2024. [Online]. Available: <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs#:~:text=js%3F-,Node.,was%20released%20in%20April%202021.>
- [24] R. Sheldon and J. Denman, “What is the Node.js (Node) runtime environment?—TechTarget Definition,” TechTarget Website. Accessed: Jan. 10, 2024. [Online]. Available: <https://www.techtarget.com/whatis/definition/Nodejs>
- [25] Dicoding, “Tutorial Node.js: Apa itu Node.js dan dasar-dasar Node.js,” Dicoding Website. Accessed: Jan. 10, 2024. [Online]. Available: <https://www.dicoding.com/blog/apa-itu-node-js/>
- [26] Google, “API Google Cloud,” Google . Accessed: Jan. 10, 2024. [Online]. Available: https://cloud.google.com/apis/docs/overview?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=tc&hl=id
- [27] S. Dewi, S. P. Adithama, and A. T. Suhardi, “Pengujian Aplikasi Doctor to Doctor Menggunakan Metode Black Box Testing,” 2023.
- [28] H. Raihan and A. Voutama, “Pengujian Black Box Pada Aplikasi Database Perguruan Tinggi dengan Teknik Equivalence Partition,” *Antivirus : Jurnal Ilmiah Teknik Informatika*, vol. 17, no. 1, pp. 1–18, May 2023, doi: 10.35457/antivirus.v17i1.2501.
- [29] A. Arifandi, R. Nafal Zuhdi Simamora, G. Azam Janitra, M. Ainul Yaqin, M. Maariful Huda, and I. Artikel Abstrak, “Survei Teknik-Teknik Pengujian Software Menggunakan Metode Systematic Literature Review,” *ILKOMNIKA: Journal of Computer Science and Applied Informatics E*, vol. 4, no. 3, pp. 297–315, 2022, doi: 10.28926/ilkomnika.v4i3.436.

- [30] S. L. Kekurangan *et al.*, “LITERATURE STUDY OF THE LACK AND EXCESS OF TESTING THE BLACK BOX,” *TEKNOMATIKA*, vol. 10, no. 02, pp. 1–5, 2020.