

BAB I PENDAHULUAN

1.1 Latar Belakang

Pandemi COVID-19 sangat mempengaruhi kehidupan masyarakat dalam beraktivitas, karena pemerintah melakukan PPKM (Pembatasan Kegiatan Masyarakat) dengan adanya peraturan tersebut masyarakat harus berada di dalam rumah pada jangka waktu yang cukup lama. Karena diberlakukan aturan baru yang membatasi kegiatan masyarakat untuk bersosialisasi. Permasalahan pada pandemi ini membuat warga kesulitan untuk berkumpul dan melakukan aktivitas sehari-hari. Dalam kasus ini maka dilakukan proses transformasi digital dalam segala bidang, seperti bidang Pendidikan, Ekonomi, dan sebagainya. Hal tersebut dilakukan untuk mengurangi kontak fisik.

Salah satu dampak dari pandemi di lingkungan RT (Rukun Tetangga) yaitu terkendalanya pembayaran uang iuran jimpitan dan uang sampah yang selain itu kurang efisien dalam membagikan informasi oleh pengurus RT (Rukun Tetangga). Dikarenakan pembagian informasi masih melalui grup *whatsapp* dan biasanya tidak semua orang membuka grup, sehingga pesan yang disampaikan tidak terbac.

Berdasarkan permasalahan tersebut maka dibuat sebuah aplikasi berbasis android untuk menangani masalah yang ada. Aplikasi yang digunakan untuk melakukan pembayaran uang sampah dan jimpitan pada era *new normal*. Dalam aplikasi tersebut juga terdapat fitur agenda dan pemberitahuan agar warga bisa mengetahui berita terkini di lingkungan sekitar.

Adapun batasan masalah dalam pembuatan aplikasi *Jimpit.in* agar aplikasi dapat digunakan, yaitu :

1. Bahasa pemrograman yang digunakan untuk aplikasi klien Android adalah Java.
2. Bahasa pemrograman yang digunakan untuk aplikasi server adalah PHP dengan menggunakan framework CI 3 dan framework Bootstrap.
3. Admin panel menggunakan template Admin LTE.
4. Sistem manajemen basis data yang digunakan adalah MySQL.

5. Aplikasi yang digunakan untuk membuat aplikasi klien adalah Android Studio Arctic Fox (2020.3.1).
6. Aplikasi yang digunakan untuk membuat aplikasi server adalah Visual Studio Code, XAMPP, dan Google Chrome.
7. Figma digunakan sebagai aplikasi untuk melakukan UI/UX Design.
8. Versi minimum Android yang didukung adalah Android 4.1 (API Level 16).
9. Layanan dan alat yang digunakan untuk pengujian aplikasi adalah AWS Device Farm dan Postman.
10. Aplikasi ini digunakan pada lingkungan RT (Rukun Tetangga).

Tujuan pengembangan produk ini untuk membuat prototype aplikasi android dengan memperhatikan kebutuhan fungsional dan non-fungsional agar memudahkan warga melakukan pembayaran uang jimpitan dan uang sampah serta mengakses informasi sekitar lingkungan RT (Rukun Tetangga). Sehingga pencatatan keuangan dalam lingkungan RT (Rukun Tetangga) lebih terstruktur dan menghindari salah pencatatan oleh pengurus. Pada fitur agenda juga dapat digunakan sebagai sarana oleh pengurus RT (Rukun Tetangga) untuk membagikan informasi seputar berita terkini di lingkungan sekitar.

Manfaat dari aplikasi Jimpit.in bagi masyarakat di lingkungan RT (Rukun Tetangga) yaitu memudahkan warga untuk melakukan transaksi pembayaran uang sampah dan jimpitan, fitur agenda memudahkan warga mengetahui jadwal kegiatan lingkungan RT (Rukun Tetangga), dan mengurangi penggunaan uang tunai dan kontak fisik. Sedangkan bagi pengurus manfaat penggunaan aplikasi jimpitin adalah untuk mengurangi kontak fisik dan resiko penularan virus, dan adanya fitur agenda kegiatan di lingkungan RT (Rukun Tetangga) lebih terjadwal oleh pengurus.

1.2 Profil

Indoneris IT Competition merupakan sebuah kompetisi Nasional dalam bidang IT yang diselenggarakan oleh Universitas Amikom Purwokerto yang bekerja sama dengan NERIS, CORIS dan INDOCEISS. Pada tahun 2021 INDONERIS

mengusung tema “Prepare The Post-Pandemic Society by Interconnected Technology”.

1.2.1. Profil Penyelenggara Lomba

1. Universitas Amikom Purwokerto

Universitas Amikom Purwokerto merupakan perguruan tinggi swasta berbasis teknologi dan bisnis yang berlokasi di Purwokerto, Jalan Letjend Poli. Soemarto, Kabupaten Banyumas, Jawa Tengah. Tepat di depan Akademi Kepolisian Negara Purwokerto.

Halaman web: <https://amikompurwokerto.ac.id/>

2. NERIS

Neris (Network Research Inter University) merupakan sebuah organisasi yang berfokus

3. CORIS

CORIS didirikan pada tahun 2012 oleh Dr. Djoko Soetarno, DEA, dosen Bina Nusantara yang bersemangat memajukan perguruan tinggi di daerah. Dalam diskusinya dengan pimpinan universitas STIKOM Bali dan Potensi Utama Medan, mereka sepakat untuk membentuk paguyuban dengan Dr. Djoko Soetarno mendirikan DEA sebagai direktur. Pada 2013, beberapa universitas diundang. Dalam pertemuan di Universitas AMIKOM Yogyakarta, beberapa perguruan tinggi sepakat untuk bergabung dalam organisasi ini. Perguruan tinggi tersebut adalah STIKOM ITB Bali, Universitas Utama Potensial Medan, Universitas AMIKOM Yogyakarta, STMIK Diponegara Makassar, STMIK Pontianak dan STMIK Tasikmalaya. Saat itu semua universitas tersebut masih dalam bentuk STMIK. Setelah berbagai kegiatan, banyak perguruan tinggi lain yang bergabung dan pada awal tahun 2021 CORIS telah memiliki 21 perguruan tinggi anggota di 16 provinsi dari Papua hingga Sumatera Utara.

Halaman web : <http://coris-group.org/>

4. INDOCEISS

Organisasi IndoCEISS berbentuk kesatuan dimana kedaulatan tertinggi di tangan anggota melalui Musyawarah Nasional. IndoCEISS merupakan organisasi yang dibentuk atas dasar kesamaan profesi. Organisasi ini didirikan di Yogyakarta pada tanggal 19 Januari 2010 sebagai hasil pemikiran adanya kebutuhan organisasi profesi bidang komputer, elektronika dan instrumentasi di Indonesia.

1.2.2. Kategori Lomba

Karya ini mengikuti kategori Mobile

1.2.3. Syarat Lomba

1. Peserta adalah mahasiswa aktif yang tercatat pada PDDIKTI.
2. Biaya pendaftaran Rp. 150.000 (Seratus lima puluh ribu rupiah).
3. Tema animasi competition "Indonesia Bangkit Melawan Covid-19 di masa Pandemi".
4. Konten merupakan karya orisinal serta tidak boleh menggunakan materi yang melanggar hak cipta.
5. Konten didalamnya tidak boleh mengandung unsur SARA dan pornografi.
6. Peserta bertanggungjawab atas segala permasalahan hak cipta maupun hal yang terkait dengan pembuatan dan kepemilikan karya yang didaftarkan.
7. Jumlah peserta dalam 1 tim terdiri dari 3 sampai 5 orang peserta.
8. Pemenang akan diumumkan di website Indoneris.
9. Panitia berhak mendiskualifikasi produk sebelum, selama, dan setelah penjurian berlangsung apabila dinilai melanggar peraturan dan syarat-syarat.
10. Dengan mengikuti lomba ini, peserta memberikan hak kepada panitia untuk menggunakan Mobile Competition untuk keperluan publikasi lomba, dokumentasi yang akan dimanfaatkan oleh acara INDONERIS.

1.2.4. Level Lomba

Lomba Indoneris *IT Competition* merupakan sebuah kompetisi Nasional.

1.3 Landasan Teori

1.3.1 SDLC

Siklus Hidup Pengembangan Sistem (SDLC) merupakan pendekatan sistematis yang digunakan dalam pengembangan perangkat lunak dan sistem informasi. SDLC menggambarkan serangkaian tahapan atau fase yang harus dilalui dalam proses pengembangan, mulai dari perencanaan awal hingga implementasi dan pemeliharaan sistem. Tujuan dari SDLC adalah untuk memastikan bahwa pengembangan sistem dilakukan secara terstruktur, efisien, dan sesuai dengan kebutuhan pengguna[1].

1.3.2 Metode Waterfall

Pengembangan perangkat lunak tertua yang bersifat sistematis dan berurutan. Nama model ini sebenarnya adalah "*Linear Sequential Model*". Model ini sering disebut juga dengan "*classic life cycle*" atau metode waterfall. Model ini dapat digunakan pada saat kebutuhan untuk sebuah masalah telah dipahami dengan baik, dan pekerjaan dapat mengalir secara linear dari proses komunikasi hingga penyebaran (*deployment*)[2].



Gambar 1 Model Waterfall[2]

Dalam metode waterfall terdapat beberapa tahapan yaitu *feasibility study*, *requirement analysis and specification*, *design*, *coding and unit testing*, *integration*

system testing, dan *maintenance*. Berikut penjelasan dari tahapan metode *waterfall*

:

a. Feasibility Study

Kegiatan yang dilakukan pada feasibility study adalah menentukan apakah layak secara finansial dan teknis dalam pengembangan perangkat lunak yang akan dikembangkan. Studi kelayakan ini melibatkan pemahaman terhadap permasalahan dan kemudian merancang berbagai system[2].

b. Requirement System and Specification

Kegiatan yang dilakukan pada tahap ini adalah analisis dan spesifikasi kebutuhan untuk memahami kebutuhan atau permintaan yang tepat dari pelanggan dan didokumentasikan dengan benar.

1. Pengumpulan dan analisa kebutuhan; Semua persyaratan terkait pengembangan perangkat lunak dikumpulkan dari pelanggan dan kemudian persyaratan yang dikumpulkan untuk dilakukan analisis. Tujuan dari bagian analisis adalah menghilangkan ketidaklengkapan dan ketidakkonstisenan kebutuhan dalam pengembangan perangkat lunak.
2. Spesifikasi kebutuhan: Persyaratan yang tidak dilakukan analisis ini didokumentasikan dalam dokumen spesifikasi kebutuhan pengembangan perangkat lunak (*Software Requirement Specification*). Dokumen ini berfungsi sebagai kontrak antara tim pengembangan dan pelanggan[2].

c. Design

Pada tahapan ini, kegiatan yang dilakukan adalah untuk mengubah persyaratan dalam dokumen SRS (*Software Requirement Specification*) menjadi struktur yang sesuai untuk diimplementasikan dalam beberapa bahasa pemrograman. Pada tahap ini masalah dijabarkan lebih lanjut dengan menggambar sketsa dan diagram[2].

d. Coding and Unit Testing

Tahap pengkodean (*Coding*) adalah kegiatan yang dilakukan untuk

menerjemahkan desain perangkat lunak ke dalam bahasa pemrograman. Tahapan ini juga disebut dengan tahapan implementasi. Setiap komponen desain diimplementasikan sebagai modul program. Produk akhir dari tahap ini adalah sekumpulan modul program yang telah diuji secara individu. Setelah melakukan pengkodean, selanjutnya setiap modul diuji untuk menentukan apakah kinerja setiap modul berjalan dengan lancar[2].

e. Integration and System Testing

Kegiatan yang dilakukan pada tahap integrasi dan pengujian testing adalah integrasi berbagai modul dilakukan setelah proses penerjemahan ke dalam bahasa pemrograman dan setiap unit modul diuji secara individu. Integrasi modul dilakukan secara bertahap melalui sejumlah Langkah. Setelah itu dilakukan pengujian sistem[2].

f. Maintenance

Kegiatan yang dilakukan pada tahap maintenance atau perawatan memperbaiki kerusakan pada perangkat lunak atau perubahan atau peningkatan perangkat lunak diminta oleh pelanggan. Pada tahapan ini terpenting dari siklus hidup perangkat lunak. Upaya dihabiskan untuk pemeliharaan 60% dari total Upaya yang dihabiskan dalam pengembangan perangkat lunak lengkap[2].

1.3.3 Android

Android adalah sebuah sistem operasi untuk perangkat mobile yang didasarkan pada kernel Linux. Sistem operasi ini terdiri dari komponen-komponen seperti sistem operasi inti, middleware, dan aplikasi. Keunikan dari Android adalah platformnya yang terbuka, memberikan kesempatan bagi para pengembang untuk membuat dan mengembangkan aplikasi mereka sendiri. Dengan adanya platform terbuka ini, pengembang memiliki kebebasan untuk mengeksplorasi dan menciptakan berbagai jenis aplikasi yang beragam sesuai dengan kebutuhan dan preferensi pengguna[3].

1.3.4 Android Studio

Android Studio adalah sebuah editor kode yang cerdas. Salah satu keunggulannya adalah fitur auto completion yang memberikan saran kode saat kita mengetik. Selain itu, IDE ini juga memiliki kemampuan analisis kode yang handal serta mampu melakukan refactoring. Dengan fitur-fitur ini, waktu pembuatan program dapat dipercepat sehingga para pengembang menjadi lebih produktif. Selain itu, Android Studio juga menyediakan sistem build yang handal dan fleksibel. Fitur ini memudahkan para pengembang dalam melakukan build aplikasi Android. Setelah melakukan kompilasi dan menjalankan aplikasi, file APK akan secara otomatis terbentuk sehingga para pengembang tidak perlu repot untuk melakukan build aplikasi lagi[4].

1.3.5 Project Manager

Kepala Proyek (Kapro) adalah salah satu pihak yang bertanggung jawab dan mengendalikan kegiatan harian proyek untuk mencapai kesuksesannya. Istilah ini sesuai dengan definisi PMBOK (2013) yang menyebutkan bahwa kepala proyek adalah seseorang yang ditugaskan oleh pelaksana untuk memimpin tim yang bertanggung jawab mencapai tujuan proyek. Kepala proyek memiliki tugas memastikan proyek berjalan sesuai dengan rencana yang telah dibuat. Selain itu, kepala proyek juga harus memiliki kompetensi dalam hal pengetahuan, kinerja, dan keterampilan personal yang memadai. Semua aspek kompetensi ini akan mendukung tercapainya tujuan proyek dan kesuksesannya. Dalam konteks lapangan di Indonesia, istilah kepala proyek lebih cocok digunakan untuk menggambarkan peran yang sama dengan project manager[5].

1.3.6 Use Case Diagram

Sebuah *Use Case* diagram menyatakan visualisasi interaksi yang terjadi antara pengguna (aktor) dengan sistem. Diagram ini bisa menjadi gambaran yang bagus untuk menjelaskan konteks dari sebuah sistem sehingga terlihat jelas batasan dari sistem. Ada 2 elemen penting yang harus digambarkan, yaitu aktor dan *Use Case*[6].

1.3.7 Use Case Deskripsi

Use Case deskripsi merupakan penjelasan secara tekstual dari sekumpulan skenario interaksi. Setiap skenario mendeskripsikan aksi atau langkah yang

dilakukan aktor ketika berinteraksi dengan sistem baik yang berhasil maupun gagal[6].

1.3.8 Whitebox Testing

White-box testing, atau pengujian white-box, dilakukan untuk menguji dan menganalisis kode program dengan memperhatikan struktur internal dan logika program yang ada. Metode ini memeriksa bagaimana program berperilaku dan merespon input yang berbeda dengan memeriksa kode program secara rinci[7].

1.3.9 Activity Diagram

Activity diagram merupakan diagram yang mampu menjelaskan secara prosedural alur proses dari sebuah sistem. Dalam diagram ini dimungkinkan untuk mengevaluasi kemungkinan adanya lebih dari satu jalur yang terbentuk dan berjalan secara bersamaan. Penggambaran dari activity diagram dimulai dari initial node hingga berakhir di end node. Yang dapat menjadi catatan bahwa initial node dalam sebuah activity diagram diperbolehkan lebih dari satu. Hal ini dilakukan untuk mengakomodasi jika sistem yang dimodelkan memiliki lebih dari satu input[8].

1.3.10 Blackbox Testing

Blackbox testing merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak tanpa memperhatikan detail internal kode program. Dalam *blackbox testing*, pengujian dilakukan dengan melihat sistem perangkat lunak sebagai sebuah kotak hitam di mana input diberikan dan output yang dihasilkan diamati, tanpa pengetahuan tentang bagaimana proses internalnya bekerja[9].