

Pengembangan Helpdesk Ticketing System berbasis Website dengan menggunakan metode SDLC, XP, dan Scrum

Mangapul Siahaan, Kelvin*

Fakultas Ilmu Komputer, Program Studi Sistem Informasi, Universitas Internasional Batam, Batam, Indonesia

Email: ¹mangapul.siahaan@uib.ac.id, ^{2,*}2031093.kelvin@uib.edu

Email Penulis Korespondensi: 2031093.kelvin@uib.edu

Abstrak—Kemajuan teknologi informasi saat ini yang cepat berkembang erat kaitannya dengan meningkatnya permintaan akan informasi. Web telah berkembang menjadi alat yang tidak hanya menyediakan informasi tetapi juga memprosesnya. Pemanfaatan teknologi web untuk pemrosesan informasi telah mengubah web menjadi media informasi yang dinamis. Ada banyak kemungkinan dengan aplikasi web, seperti pengembangan Sistem *Ticketing Helpdesk* untuk perusahaan atau lembaga untuk membentuk sistem pelaporan masalah yang lebih efektif dan efisien. Jumlah masalah yang dilaporkan secara internal yang semakin meningkat membuat pelacakan semua masalah yang dilaporkan menjadi lebih menantang. Hasil dari penelitian ini adalah prototipe sistem telah berhasil dibangun. Sistem yang dibangun dapat beroperasi sesuai yang diharapkan oleh pengguna sistem, seperti yang ditunjukkan oleh hasil pengujian menggunakan metode pengujian *black box testing*. Hasil dari penelitian ini bertujuan untuk mengidentifikasi model pengembangan aplikasi yang dapat mempercepat dan meningkatkan proses pengembangan menggunakan tiga metode, yaitu *Software Development Life Cycle (SDLC)*, *Extreme Programming (XP)*, dan *Scrum*. Pemanfaatan pendekatan metode XP oleh peneliti dalam membangun sistem tiket *helpdesk* terbukti sangat efektif dalam mempercepat pengembangan sistem.

Kata Kunci: SDLC; XP; Scrum; Ticketing Helpdesk; Black Box Testing

Abstract—The current rapid advancement of information technology is closely tied to the growing demand for information. The web has evolved into a tool that not only provides information but also processes it. Utilizing web technology for information processing has turned the web into a dynamic information medium. There are numerous possibilities with web applications, such as developing a Helpdesk Ticketing System for companies or institutions to establish a more effective and efficient issue reporting system. The increasing number of internally reported issues has made tracking all reported problems more challenging. The result of this research is the system prototype has been successfully built. The built system can operate as expected by system users, which is demonstrated by the results of testing using the black box testing method. The outcomes of this research is to identify the application development model that can expedite and enhance the development process using three methods, namely Software Development Life Cycle (SDLC), Extreme Programming (XP), and Scrum. The utilization of the XP method approach by the researcher in building the helpdesk ticketing system has proven to be highly effective in accelerating system development.

Keywords: SDLC; XP; Scrum; Ticketing Helpdesk; Black Box Testing

1. PENDAHULUAN

Perkembangan teknologi informasi saat ini begitu pesat seiring dengan kebutuhan akan informasi. Web telah berkembang menjadi alat yang tidak hanya mampu memberikan informasi tetapi juga untuk mengolah informasi. Pengolahan informasi dengan memanfaatkan teknologi web menjadikan web sebagai media informasi yang dinamis. Banyak hal yang bisa kita lakukan dengan membuat aplikasi web, seperti membangun *Helpdesk Ticketing System* bagi perusahaan atau institusi untuk membangun sistem pelaporan masalah yang lebih efektif dan efisien. Meningkatnya masalah yang dilaporkan secara internal, menyebabkan pelacakan semua masalah yang dilaporkan menjadi lebih sulit [1]. Di bidang teknologi informasi, saat kita memasuki era digital, *IT Support* bergerak di banyak perusahaan, institusi, dan organisasi lain untuk memberikan dukungan teknis, tidak hanya terbatas pada jaringan komputer, sistem operasi, koneksi internet, keamanan komputer, atau masalah perangkat lunak/perangkat keras lain yang mungkin timbul [1]. Sistem tiket memberikan solusi kepada semua klien yang terhubung dengan satu atau lebih organisasi jarak jauh atau di dalam infrastruktur. Sistem tiket juga digunakan sebagai perangkat lunak *helpdesk* yang dapat membantu klien mendapatkan layanannya secara efisien sambil mempertahankan permintaan layanan mereka [2].

Penelitian ini didasarkan dari beberapa penelitian yang telah ada sebelumnya, yaitu dari Phitsa Mauliana, Wildan Wiguna, dan Abrian Yudha Permana yang menjadi dasar utama dari penelitian ini. Penelitian tersebut melakukan penelitian berbasis terapan, dimana fokus pada penelitian ini adalah mengembangkan sebuah *helpdesk support system* berbasis web menggunakan metode *System Development Life Cycle (SDLC)* dengan model *Waterfall*. Hasil dari penelitian ini adalah sistem yang akan mendukung tugas *IT Support* dan menyederhanakan proses pengajuan keluhan dari karyawan internal di PT Akur Pratama [3].

Penelitian berikutnya dilaksanakan oleh Muhamad Saepuloh, yang bertujuan untuk merancang dan mengembangkan sebuah aplikasi sistem *helpdesk* berbasis web untuk CV Nagatech. Departemen *Helpdesk* memiliki tanggung jawab untuk memberikan bantuan dalam menangani permasalahan terkait perangkat lunak yang digunakan oleh pelanggan CV Nagatech. Dalam layanan *helpdesk*, instalasi perangkat lunak merupakan salah satu aspek yang dicakup, pemecahan masalah perangkat lunak, pemecahan masalah perangkat keras, dan pencadangan basis data. Saat ini, divisi *helpdesk* masih menggunakan aplikasi WhatsApp untuk memberikan pelayanan, yang seringkali tidak efisien karena sering terjadi penundaan atau masalah yang tidak terselesaikan tepat waktu. Oleh karena itu, diperlukan sebuah aplikasi yang dapat membantu dalam mengelola keluhan-keluhan tersebut, bahkan ketika permintaan layanan sedang tinggi. Penelitian ini menggunakan metode *Extreme Programming (XP)*. Harapannya, hasil dari penelitian ini akan berfungsi

dengan baik dan dapat memberikan dukungan yang lebih efektif bagi divisi *helpdesk* dalam menyelesaikan tugas-tugasnya [4].

Selanjutnya adalah penelitian yang dilakukan oleh Umami Azidah Rachmawati, Syarif Adam, dan Syarif Husain Alwi yang meneliti tentang implementasi *helpdesk ticketing system* berbasis web untuk mempermudah dalam penyelesaian masalah dan memenuhi kebutuhan akan penyajian informasi yang cepat, tepat, dan akurat di Universitas YARSI. Dalam penelitian ini, data dikumpulkan dengan menggunakan pendekatan kuantitatif, dan tingkat kepuasan pengguna diuji melalui metode *usability*. Sedangkan metode yang digunakan dalam pengembangan sistem tersebut adalah *System Development Life Cycle* dengan model *Waterfall*. Hasil dari penelitian ini adalah sistem yang dapat memberikan dampak yang lebih baik dan efisien dalam proses pelaporan masalah dibandingkan dengan monitoring secara manual [5].

Selanjutnya, penelitian yang dilakukan oleh Ryan Muhammad Bahrudin, Mohammad Ridwan, dan Hardjito S. Darmojo berfokus pada pengembangan sistem *helpdesk ticketing* berbasis web untuk menangani keluhan yang berkaitan dengan penggunaan sistem informasi di Universitas Islam Syekh-Yusuf. Ini termasuk keluhan terkait dengan ketidaksesuaian nilai, masalah registrasi, dan masalah keuangan, mahasiswa masih diharuskan untuk hadir secara langsung di kantor teknisi IT agar dapat melaporkan keluhan mereka. Mahasiswa juga merasa kebingungan saat harus melaporkan masalah ketika teknisi IT tidak berada di tempat, dan keluhan yang disampaikan tidak memiliki dokumentasi yang memadai. Untuk mengatasi masalah ini, diperlukan suatu aplikasi *helpdesk ticketing* sebagai sarana untuk melaporkan permasalahan dengan lebih terstruktur. Metode yang digunakan dalam penelitian ini adalah observasi, wawancara dan studi literatur untuk pengumpulan data, sedangkan di dalam pengembangan sistem metode yang digunakan adalah *System Development Life Cycle (SDLC)* dengan model *Prototype*. Hasil dari penelitian ini mencakup perancangan aplikasi *helpdesk ticketing* yang bertujuan untuk menyederhanakan proses pelaporan masalah oleh klien serta mempermudah teknisi IT dalam manajemen laporan tersebut [6].

Selanjutnya adalah penelitian dari Kurdiansyah Lesmana dan Yudi Ramdhani yang dimana mereka mengembangkan aplikasi web sebagai solusi untuk meningkatkan pelayanan kepada pengguna di PT. Ihsan Solusi Informatika, sebuah perusahaan teknologi yang fokus pada sistem perbankan dan koperasi simpan pinjam berbasis syariah. Perusahaan ini selalu berusaha memberikan pelayanan yang optimal kepada pengguna. Oleh karena itu, manajemen pelaporan, penanganan masalah, serta manajemen proyek dan pengembangan sistem menjadi hal yang penting untuk meningkatkan kualitas pelayanan. Penelitian ini memiliki tujuan untuk mengembangkan sebuah aplikasi berbasis web yang menggabungkan sistem *helpdesk ticketing* dan manajemen proyek dengan menggunakan metode Scrum. Hal ini akan membuktikan bahwa aplikasi yang dibangun akan memberikan manfaat signifikan bagi pengguna dan penyedia layanan. Selain itu, aplikasi ini juga akan memastikan bahwa pelaporan dilakukan secara terperinci dan dapat diakses oleh semua anggota penyedia layanan, sehingga dapat meningkatkan efisiensi dalam operasional internal perusahaan [7].

Kemudian penelitian dari Rehulina Tarigan, Imam Kusosi, dan Andi Usri melaksanakan penelitian terhadap pelayanan operasional teknologi informasi di PT. Indonesia Nippon Seiki. Pada saat ini, kualitas manajemen pengaduan masalah dapat dikategorikan sebagai belum optimal karena prosedur pengaduan kepada departemen *Management Information System (MIS)* belum tersusun secara sistematis. Laporan yang diberikan kepada atasan tidak lengkap akibat proses pendataan yang belum tertata dengan baik. Permasalahan ini muncul karena belum ada media atau sistem pencatatan data yang memadai. Pimpinan IT juga mengalami kesulitan dalam memantau kinerja tim dalam menangani setiap keluhan dari pengguna karena data tidak dapat diakses secara *online*. Metode yang digunakan dalam pengembangan sistem ini adalah *System Development Life Cycle* dan pemodelan proses bisnis dengan menggunakan UML (*Unified Modeling Language*) menghasilkan sebuah aplikasi komputer yang memungkinkan akses dalam waktu nyata, memungkinkan pengguna untuk secara langsung melihat status penyelesaian dari kasus yang mereka ajukan [8].

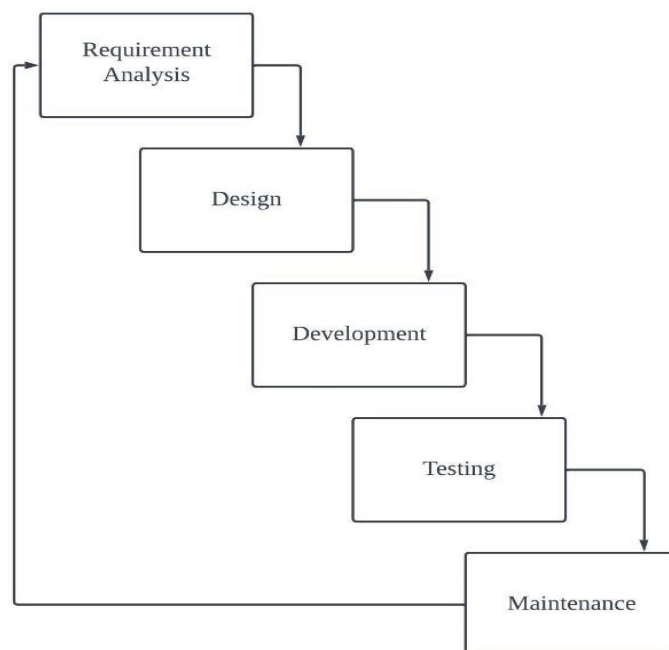
Di dunia bisnis saat ini, teknologi dan informasi telah menjadi kebutuhan krusial dalam menjalankan operasional perusahaan untuk mencapai tujuan yang telah ditetapkan. Pemanfaatan teknologi dan informasi memungkinkan perusahaan untuk bersaing dengan kompetitornya [9].

Penelitian terakhir adalah penelitian yang dilakukan oleh Stenly Ibrahim Adam, Jimmy H. Moedjahedy, dan Oktoverano Lengkong yang berfokus untuk menyelesaikan masalah yang ada pada proses pelaporan masalah di Universitas Klabat Seringkali, departemen yang melaporkan gangguan atau kerusakan tidak memiliki perkiraan jelas mengenai berapa lama masalah mereka akan ditangani karena proses antrian pekerjaan tidak terlihat dengan jelas, yang bisa memberikan kesan bahwa penanganan masalah agak lambat. Fokus dari penelitian ini adalah membuat sebuah sistem *IT helpdesk ticketing* agar pelaporan masalah dapat di monitor dengan baik dan penyelesaiannya menjadi lebih cepat. Metode yang digunakan dalam penelitian ini adalah SDLC (*System Development Life Cycle*) dengan model pengembangan *Waterfall*. Hasil dari penelitian ini adalah sistem yang dapat mendukung departemen IT dalam mengawasi tugas dan pekerjaan yang diterima [10].

Tujuan dari penelitian ini diantaranya adalah merancang dan membuat sistem informasi *helpdesk ticketing* dengan menggunakan bahasa pemrograman ReactJS, PHP, *framework* Laravel dan MySQL sebagai database; membentuk sistem pelaporan masalah yang lebih efektif dan efisien; membantu menyederhanakan proses pelaporan masalah; mengidentifikasi model pengembangan aplikasi yang dapat mempercepat dan meningkatkan proses pengembangan menggunakan tiga metode, yaitu Software Development Life Cycle (SDLC), Extreme Programming (XP), dan Scrum.

2. METODOLOGI PENELITIAN

Dalam penelitian ini, penulis akan melakukan peninjauan kembali terhadap referensi yang akan digunakan terkait dengan topik penulis yang sedang diteliti. Referensi yang telah direvisi tersebut akan dikumpulkan untuk menyelesaikan dan menanggapi permasalahan. Tahapan selanjutnya adalah pengembangan aplikasi, pada tahap ini penulis akan mulai pengembangan aplikasi berbasis web dengan menggunakan salah satu metode pengembangan yaitu *System Development Life Cycle (SDLC)* dengan menerapkan model *Waterfall* sehingga pengembangan dapat diselesaikan dan diuji oleh target calon pengguna. SDLC (*Software Development Life Cycle*) memiliki peran penting dalam memastikan bahwa pengembangan perangkat lunak dilaksanakan dengan cara yang terstruktur dan tertata, dengan akhir tujuan menghasilkan produk berkualitas tinggi yang memenuhi keperluan dari klien. Dalam industri perangkat lunak, *Software Development Life Cycle (SDLC)* adalah suatu metode yang digunakan untuk merencanakan, mengembangkan, dan menghasilkan produk perangkat lunak yang memiliki tingkat kualitas tinggi, kehandalan, efisiensi biaya, dan ketepatan waktu. SDLC juga dikenal sebagai kerangka kerja pengembangan perangkat lunak. Terdapat beberapa model proses SDLC yang dapat dipilih, seperti *Waterfall*, *Iterative*, *Spiral*, *V-Model*, *Big Bang*, *Agile*, *Rapid Application Development Model*, dan *Software Prototype* [11]. Dalam penelitian ini, digunakan model SDLC berupa *Waterfall*, dengan skema yang mirip seperti pada gambar 1.



Gambar 1. Metode SDLC [11]

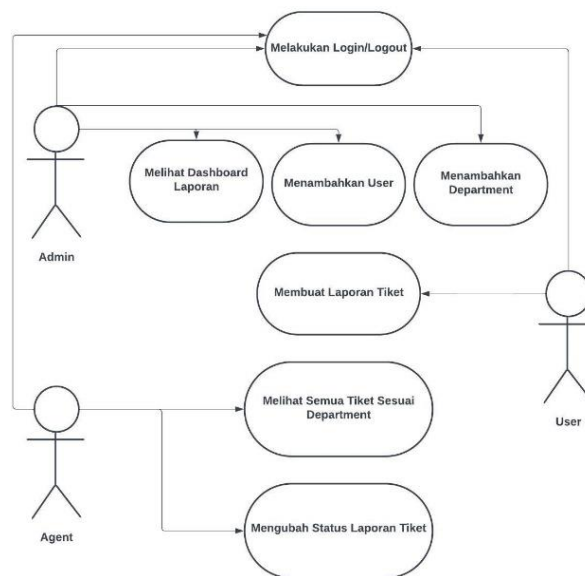
Model *Waterfall* adalah pendekatan pengembangan perangkat lunak yang paling mendasar, yang dimana bahwa seluruh tahap dalam *Software Development Life Cycle (SDLC)* dilaksanakan secara berurutan. Artinya, tahap berikutnya hanya dimulai setelah tahap sebelumnya telah selesai [11]. Pendekatan pengembangan sistem menggunakan model *Waterfall* melibatkan beberapa tahapan yang harus dilalui, yaitu:

2.1 Requirement Analysis

Tahap *Requirements Analysis* dimulai dengan memahami bagaimana kebutuhan pengguna terhadap sebuah perangkat lunak. Dari hasil wawancara akan menghasilkan masukan dan nilai prioritas ke *user stories* berdasarkan tingkat kebutuhan pengguna. *User stories* akan direpresentasikan dengan pemodelan diagram *Unified Modeling Language (UML)* yaitu *Use Case Diagram (UCD)* dan *Entity Relationship Diagram (ERD)*.

a. Use Case Diagram

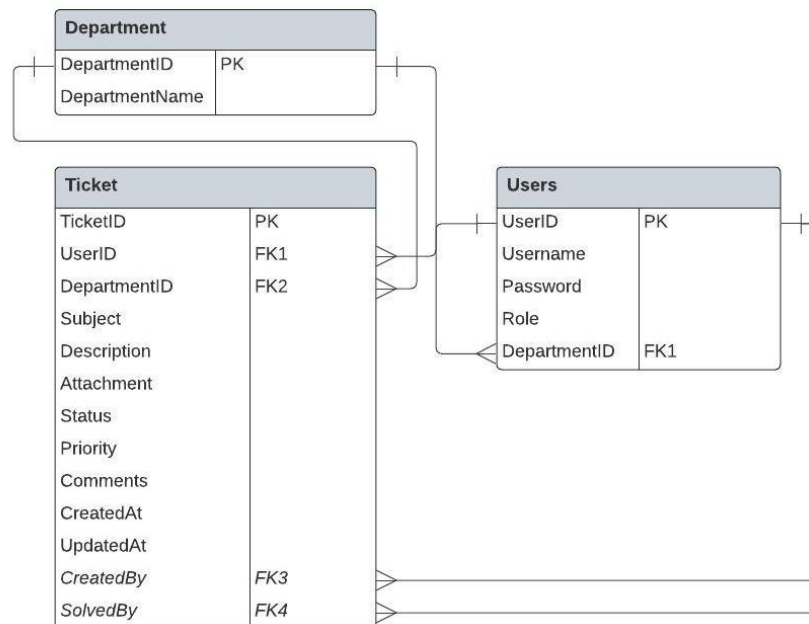
Dalam *Unified Modeling Language (UML)*, *use case* digunakan sebagai alat untuk menggambarkan interaksi antara sistem dan pengguna. *Use case* ini mengilustrasikan cara sistem beroperasi dari sudut pandang pengguna dengan menjelaskan tindakan atau skenario yang dapat dieksekusi oleh pengguna di dalam sistem tersebut [12]. *Use case* diagram penelitian ini dapat dilihat pada gambar 2.



Gambar 2. Use Case Diagram

b. Entity Relationship Diagram

Dalam *Unified Modeling Language (UML)*, *Entity Relationship Diagram (ERD)* adalah metode yang digunakan dalam langkah awal pembuatan database. ERD merupakan teknik desain database yang luas digunakan dan populer. ERD berdasarkan pada model hubungan-entitas. Data dari model hubungan-entitas yang dijelaskan secara visual dikenal sebagai ERD. ERD ini memvisualisasikan bagaimana entitas terkait satu sama lain dalam basis data. Pembuatan ERD dianggap sebagai alat yang membantu perancang dalam menganalisis basis data yang sedang dikembangkan [13]. Gambar 3 menampilkan *entity relationship diagram* dalam penelitian ini.



Gambar 3. Entity Relationship Diagram

2.2 Design

Tahap *Design* merupakan lanjutan dari tahap *Requirement Analysis* dalam metode *Waterfall*. Pada tahapan ini, penulis akan membuat desain tampilan sistem, desain diagram sistem, dan database berdasarkan *user stories* yang kemudian akan dijadikan sebagai acuan saat proses tahapan penulisan kode program.

2.3 Development

Development merupakan tahapan penulisan kode program. Pada tahapan ini, penulis akan menerjemahkan bentuk diagram sistem dengan menggunakan database MySQL serta desain tampilan sistem yang telah dibuat sebelumnya ke

dalam perangkat lunak penyunting kode Visual Studio Code dengan bahasa pemrograman ReactJS, PHP, dan *framework* Laravel.

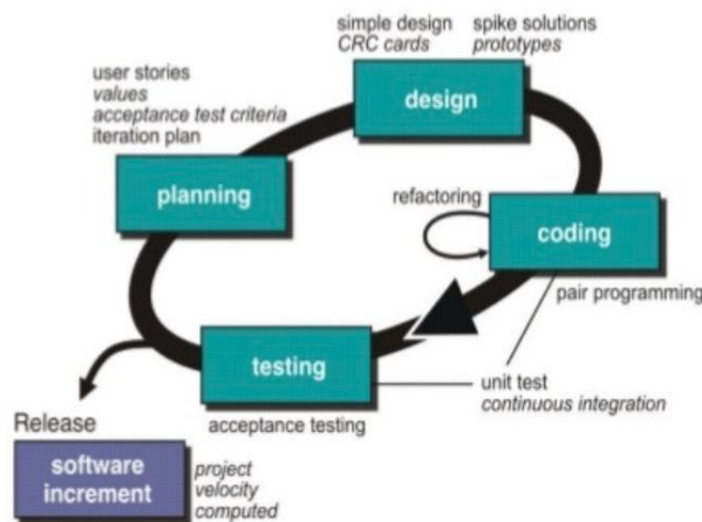
2.4 Testing

Testing merupakan tahapan pengujian dari metode *Waterfall*. Pada tahapan ini, sistem telah dibangun dan penulis akan berfokus menguji keseluruhan fitur sistem dan fungsionalitas yang berasal dari *user stories* yang telah didapatkan sebelumnya. Pengujian sistem ini menggunakan metode *blackbox testing* untuk mengamati hasil *input* dan *output* dari sistem tanpa menguji desain maupun kode program.

2.5 Maintenance

Maintenance merupakan tahapan terakhir dari metode *Waterfall*. Pada tahapan ini, penulis akan melakukan *deployment* pada aplikasi yang telah dikembangkan sistem telah dibangun. Selain itu, kesalahan yang tidak terdeteksi selama tahap pengembangan juga diperbaiki. Selain itu, dalam fase ini, sistem dikembangkan dengan mencakup penambahan fitur dan fungsi baru. Penelitian ini mengadopsi model *Waterfall* sesuai dengan persyaratan sistem.

Penelitian ini juga menggunakan model *Extreme Programming (XP)*. *Extreme Programming* merupakan salah satu metode dalam rangkaian *agile software development methodologies* yang menitikberatkan pada kegiatan pengkodean sebagai inti dari seluruh tahap dalam siklus pengembangan perangkat lunak. XP dikenal sebagai suatu pendekatan yang sangat adaptif terhadap perubahan dengan memungkinkan iterasi yang dapat dilakukan berulang kali sesuai dengan kebutuhan. Dalam XP, pendekatan ini menawarkan proses tahap-tahap pengembangan perangkat lunak yang singkat dan berulang untuk berbagai komponen sesuai dengan fokus yang ingin dicapai [14]. Metode pengembangan sistem dengan model *Extreme Programming* terdiri dari beberapa tahapan, seperti pada Gambar 4 berikut.



Gambar 4. Metode *Extreme Programming* [14]

2.6 Planning

Tahap *Planning* dalam *Extreme Programming (XP)* mencakup pemahaman kebutuhan pengguna, tujuan dari aplikasi yang akan dikembangkan, dan penyusunan rencana pengembangan [14].

2.7 Design

Dalam tahapan ini, penulis akan mengambil *user stories* dan merencanakan struktur kode agar dapat memenuhi kebutuhan klien. Tahap *Design* ini termasuk menentukan bahasa pemrograman, *environment*, serta *framework* yang akan diterapkan dalam pengembangan perangkat lunak.

2.8 Coding

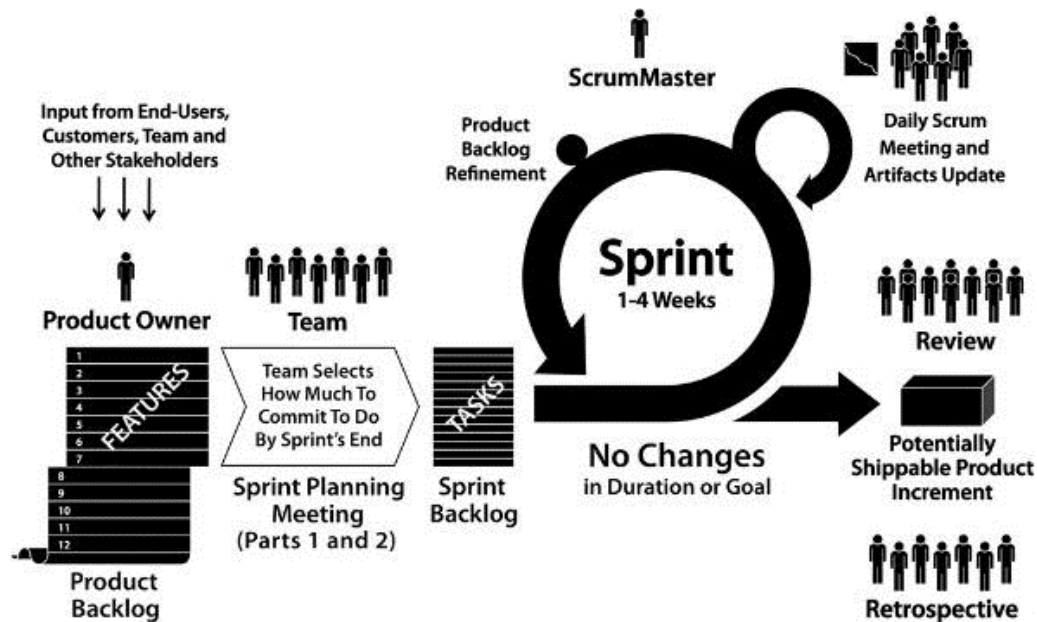
Tahap *Coding* melibatkan penulisan kode untuk mengimplementasikan desain yang telah dipersiapkan dengan menggunakan bahasa pemrograman yang bisa diinterpretasikan oleh komputer. Dalam tahap ini, penulis akan memanfaatkan MySQL sebagai database dan menggunakan bahasa pemrograman ReactJS, PHP, dan *framework* Laravel.

2.9 Testing

Testing merupakan tahapan pengujian dalam metode *Extreme Programming*. Dalam tahap ini, sistem yang telah selesai dikembangkan akan di uji oleh pengguna secara berkelanjutan dengan melakukan pengujian produk dengan berbagai *test cases* (pengujian penerimaan/fungsional) untuk memastikan kinerja yang optimal dari produk tersebut [15]. Pengujian

sistem ini mengadopsi metode *blackbox testing* yang memeriksa hasil *input* dan *output* dari sistem tanpa melibatkan penilaian desain atau kode program.

Model terakhir yang digunakan pada penelitian ini adalah *Scrum*. Pengembangan dengan model *Scrum* terbagi menjadi periode yang disebut *sprint*, yang biasanya memiliki durasi antara dua hingga empat minggu. Perencanaan *sprint* dilakukan di awal setiap periode *sprint* dan bertujuan untuk menentukan apa yang dapat dicapai dalam *sprint* berikutnya serta tindakan yang diperlukan untuk mencapainya [16]. Dalam metode *Scrum* terdapat beberapa tahapan seperti pada Gambar 5.



Gambar 5. Metode Scrum [7]

2.10 Product Backlog

Product Backlog adalah tahapan pertama dalam metode Scrum. Tahap ini melibatkan pembuatan daftar kebutuhan dan fitur yang disebut *Product Backlog*. *Product Backlog* ini memuat semua *item* yang ingin dikembangkan dalam proyek, umumnya dalam bentuk *user stories*.

2.11 Sprint Planning

Tahap *Sprint Planning* dalam *Scrum* adalah perencanaan *Sprint* yang melibatkan pemilihan *item-item* dari *Product Backlog* yang telah disusun sebelumnya, yang akan dikerjakan dalam periode *Sprint* tertentu.

2.12 Sprint

Tahap *Sprint* adalah tahap implementasi sebenarnya, di mana tim bekerja untuk mengembangkan fitur dan fungsionalitas yang telah dipilih dalam *Sprint Planning*.

2.13 Daily Scrum

Tahap *Daily Scrum* dalam *Scrum* adalah di mana setiap anggota tim berkumpul setiap hari dalam pertemuan singkat yang dikenal sebagai *stand-up*, di mana mereka berbagi informasi mengenai pekerjaan yang sudah diselesaikan, pekerjaan yang sedang dikerjakan, serta hambatan yang mungkin mereka hadapi.

2.14 Sprint Review

Pada tahapan ini setiap akhir periode *Sprint*, dilakukan pertemuan *Sprint Review* di mana melibatkan tim *Scrum* bersama dengan pihak-pihak terkait lainnya, termasuk penerimaan umpan balik dari tim *Scrum* serta proses pemeriksaan dan memperlihatkan hasil pekerjaannya kepada pihak-pihak yang berkepentingan, biasanya pemilik produk atau pengguna [17]. Selama pertemuan ini, tim dan pihak berkepentingan berdiskusi tentang pencapaian selama *Sprint* dan menerima masukan.

2.15 Sprint Retrospective

Sprint Retrospective adalah tahap selanjutnya dari *Sprint Review*. *Sprint retrospective* adalah pertemuan pemeriksaan di mana tim dapat merefleksikan kinerja keseluruhan tim, termasuk anggota, interaksi, prosedur, dan peralatan. Evaluasi yang teliti dan kritis terhadap hal-hal ini memungkinkan tim untuk membuat kesimpulan tentang bagaimana cara mereka bekerja selama *Sprint*, mencari cara untuk meningkatkan proses kerja dan mengidentifikasi hal yang perlu diperbaiki [18].

3. HASIL DAN PEMBAHASAN

Dari beragam metode yang diterapkan dalam penelitian ini, *Extreme Programming (XP)* menjadi kontributor penting dalam memperlancar proses pengembangan sistem *helpdesk ticketing* berbasis web. Tingkat efisiensi dan kecepatan dalam pengembangan *helpdesk ticketing system* dapat beragam bergantung pada sejumlah faktor, termasuk kebutuhan dan tujuan dari proyek tersebut, tingkat keahlian tim yang terlibat, serta tingkat keterlibatan pengguna. Dalam penelitian ini, ketiga metode yang disebutkan yaitu SDLC, *Extreme Programming*, dan *Scrum*, menampilkan kelebihan dan kelemahan yang dapat diidentifikasi dalam Tabel 1.

Tabel 1. Perbandingan Metode SDLC, XP, dan *Scrum*

No	Metode Pengembangan	Kelebihan	Kekurangan
1	SDLC	Struktur dan dokumentasi yang kuat, Cocok untuk proyek dengan kebutuhan stabil.	Kurang responsif terhadap perubahan kebutuhan, Proses panjang, bisa memakan waktu.
2	<i>Extreme Programming</i>	Responsif terhadap perubahan kebutuhan, Pengiriman produk yang cepat.	Tidak selalu cocok untuk proyek besar, Diperlukan tingkat disiplin yang tinggi.
3	<i>Scrum</i>	Adaptif terhadap perubahan kebutuhan, Kolaborasi yang kuat dengan pengguna.	Memerlukan peran pemilik produk yang kuat, Tidak selalu efektif untuk proyek kecil.

3.1 Planning

Pada tahap perencanaan, analisis kebutuhan pengguna dilakukan dengan wawancara dengan pengguna. Hasil dari wawancara tersebut memperoleh *user stories*. *User Stories* adalah kalimat singkat yang menggambarkan kebutuhan dari perspektif pengguna, dan dapat digunakan untuk menguraikan keinginan pengguna atau deskripsi produk [19]. *User stories* ini mencakup beberapa kebutuhan utama dalam sistem *helpdesk ticketing*, dari perspektif pengguna, admin *helpdesk*, dan agen. Dari *user stories* tersebut dapat memahami kebutuhan pengguna dalam proses pengembangan perangkat lunak seperti yang tercantum pada Tabel 2.

Tabel 2. *User Stories*

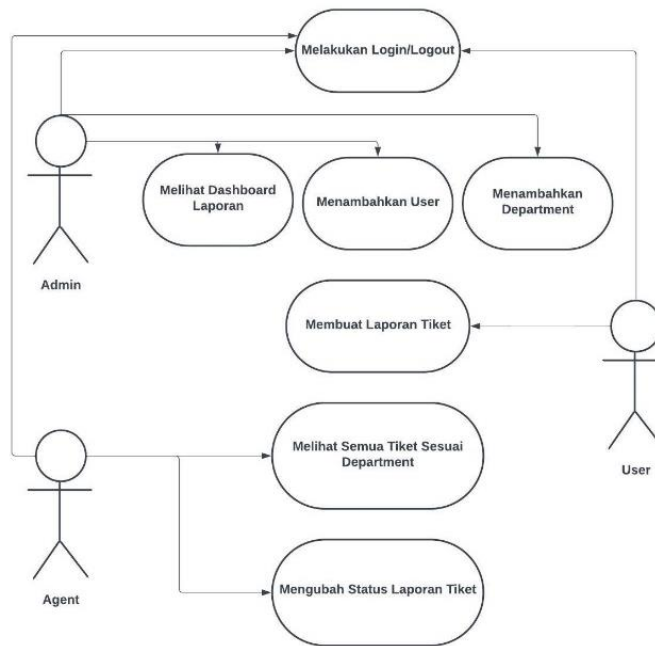
No	Deskripsi	Prioritas
1	Sebagai pengguna, dapat mengajukan tiket bantuan melalui sistem untuk melaporkan masalah teknis atau permintaan dukungan, sehingga dapat memperoleh bantuan dengan mudah.	Tinggi
2	Sebagai agen, dapat melihat daftar antrian tiket yang masuk sesuai departemen dalam agar dapat memproses laporan.	Tinggi
3	Sebagai pengguna, saya ingin mendapatkan pembaruan berkala tentang status tiket bantuan agar dapat mengetahui kemajuan penanganan masalah saya.	Sedang
4	Sebagai admin <i>helpdesk</i> , saya ingin memiliki laporan tentang kinerja tim agar kami dapat melakukan pemantauan proses pelaporan.	Tinggi
5	Sebagai pengguna, dapat mengirimkan lampiran seperti tangkapan layar atau file untuk menjelaskan detail masalah dengan rinci.	Sedang

3.2 Design

Tahapan desain adalah proses menganalisis dan merancang produk dengan berfokus pada kemudahan penggunaan sering disebut sebagai *User Experience (UX)*. Para desainer UX melakukan penelitian tentang perilaku dan kebutuhan pengguna, lalu mengembangkan perangkat lunak agar dapat digunakan dengan mudah dan memberikan pengalaman yang menyenangkan bagi pengguna [20]. Penulis akan membuat diagram sistem dalam bentuk *use case diagram* dan *entity relationship diagram* menggunakan aplikasi web yaitu *lucidchart.com*. Proses pembuatan desain akan mempresentasikan *user stories* dengan pemodelan diagram *Unified Modeling Language (UML)* yaitu *Use Case Diagram (UCD)* dan *Entity Relationship Diagram (ERD)*.

a. Use Case Diagram

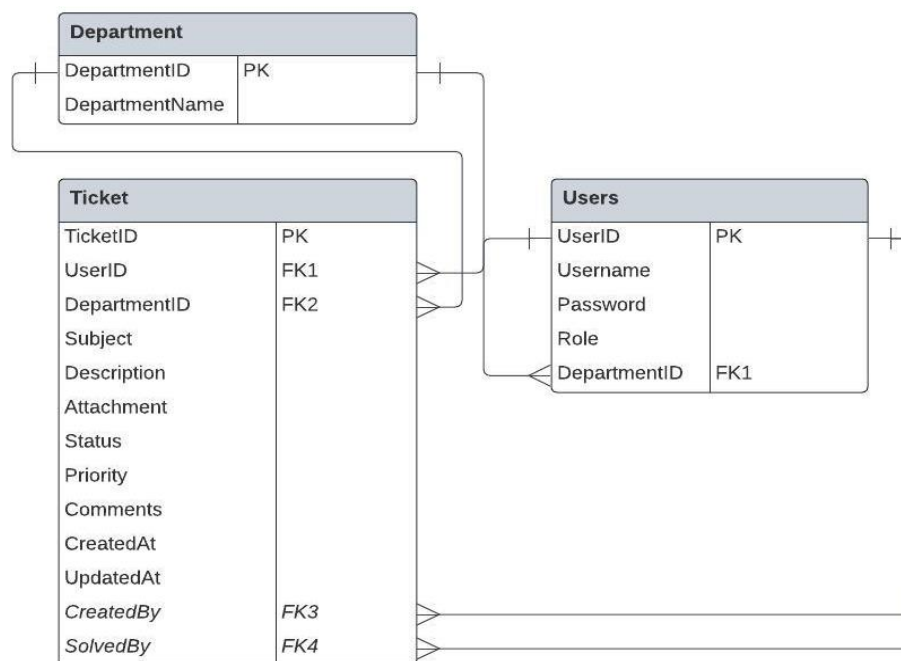
Gambar 6 menggambarkan diagram *use case* yang dibuat berdasarkan *user stories* sebelumnya. Diagram ini menggambarkan aktivitas-aktivitas yang dapat dilakukan oleh para aktor dalam sistem.



Gambar 6. Use Case Diagram

b. Entity Relationship Diagram

Entity Relationship Diagram (ERD) menggambarkan struktur sistem *helpdesk ticketing* dengan tabel-tabel yang terhubung seperti *Department*, *Users*, dan *Ticket*. Hubungan antara tabel-tabel ini ditentukan melalui *primary key* dan *foreign key*, yang menggambarkan interaksi antara pengguna, departemen, dan transaksi tiket seperti pada Gambar 7.



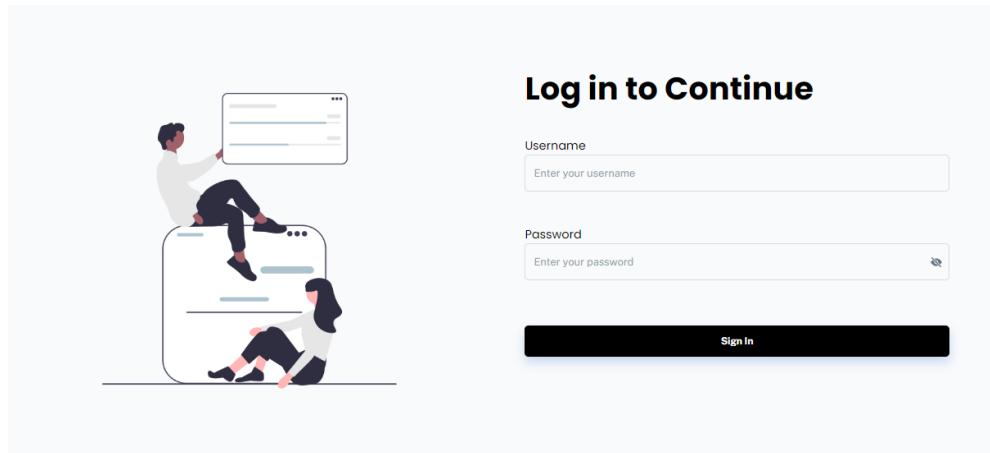
Gambar 7. Entity Relationship Diagram

3.3 Coding

Dalam metode *Extreme Programming*, pada tahap *coding* penulis menggunakan ReactJS, PHP, dan Laravel sebagai *framework* PHP. Kombinasi ini diharapkan dapat menghasilkan sistem *helpdesk ticketing* yang responsif dan cepat dengan bantuan *Virtual Document Object Model (DOM)* dari ReactJS untuk menghasilkan *rendering* tampilan yang cepat. Hasil implementasi dapat dilihat pada gambar dibawah ini.

a. Halaman *Login*:

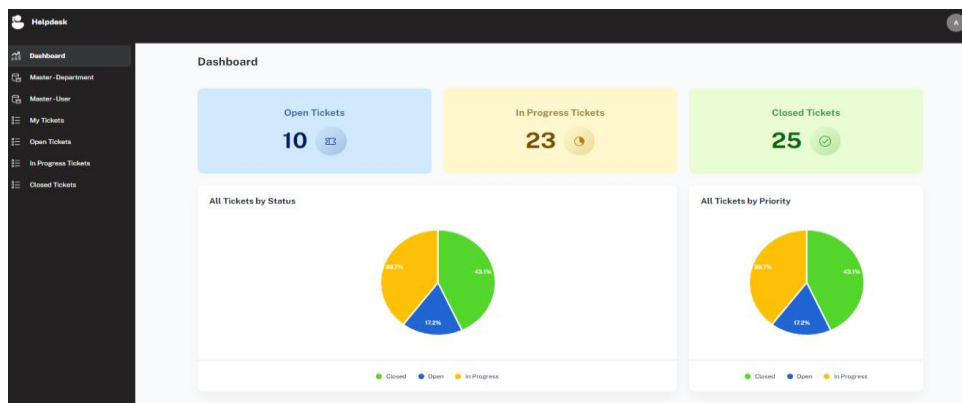
Pada Gambar 8 menggambarkan halaman *login* pengguna akan diminta untuk memasukkan *username* dan *password*. Setelah berhasil masuk tampilan akan menyesuaikan dengan hak akses pengguna yaitu *admin*, *agent*, dan *user*.



Gambar 8. Halaman Login

b. Halaman Dashboard:

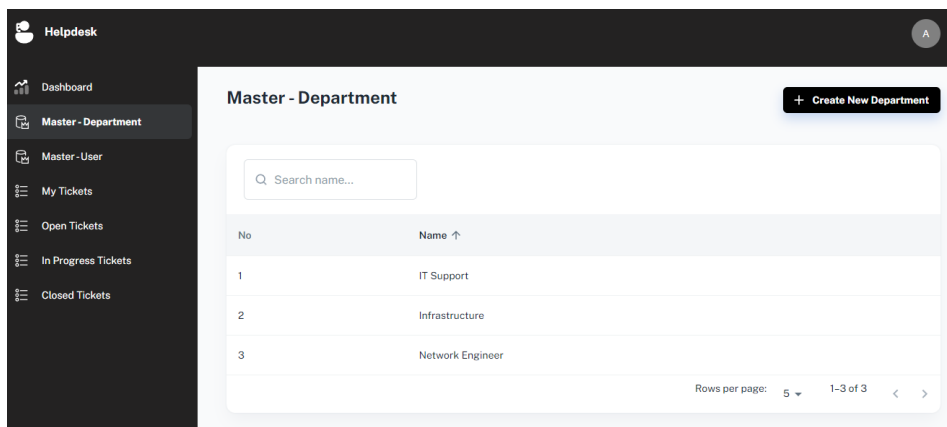
Pada Gambar 9 memperlihatkan tampilan halaman *Dashboard*. Halaman ini hanya dapat diakses oleh admin *helpdesk*. Halaman *dashboard* menampilkan beberapa informasi untuk memantau proses pelaporan tiket pada sistem yaitu jumlah tiket dengan status *open*, jumlah tiket dengan status *in progress*, dan jumlah tiket dengan status *closed*. Pada halaman ini juga terdapat grafik berbentuk *pie chart* untuk memvisualkan laporan.



Gambar 9. Halaman Dashboard

c. Halaman Department:

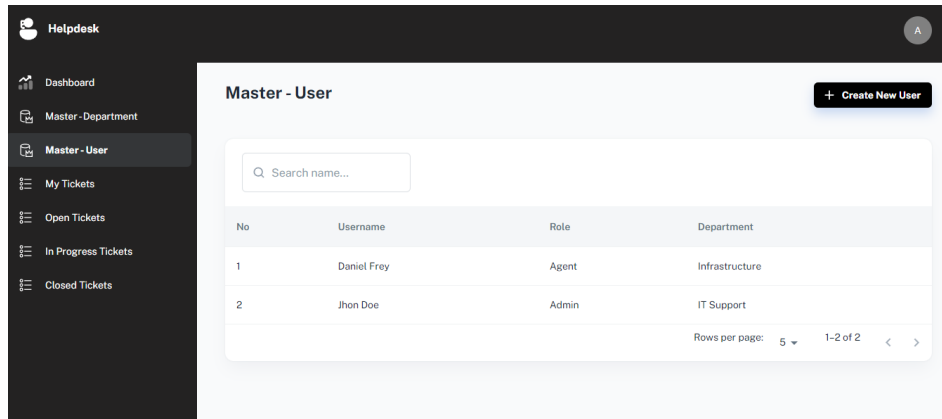
Pada Gambar 10 memperlihatkan tampilan antarmuka *Master Department*. Halaman ini adalah master data yang hanya dapat diakses oleh admin *helpdesk*. Pada halaman *department* ini, merupakan fitur penting dalam sistem aplikasi *helpdesk* yang berfungsi untuk membuat master data departemen yang akan digunakan dalam sistem *helpdesk*.



Gambar 10. Halaman Master Department

d. Halaman User:

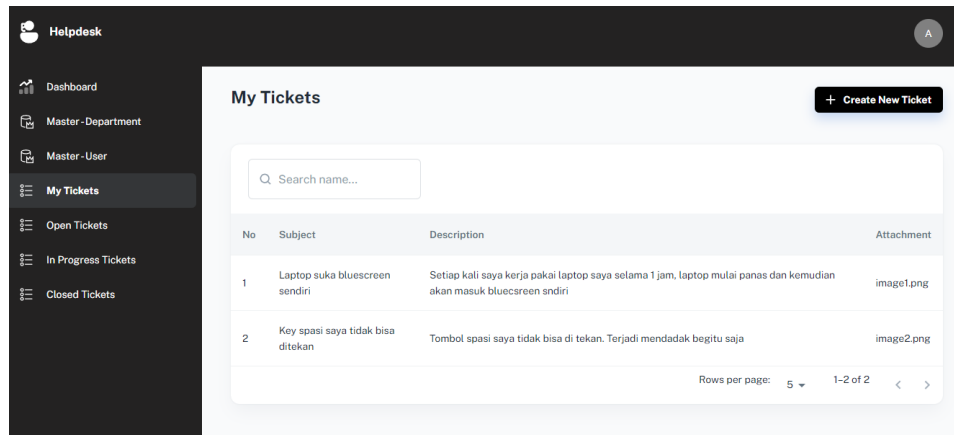
Pada Gambar 11 menampilkan antarmuka halaman *Master User*. Pada halaman master user ini hanya dapat diakses oleh admin *helpdesk*. Pada halaman *user* ini, merupakan fitur penting dalam sistem aplikasi *helpdesk* yang berfungsi untuk membuat master data pengguna yang akan menggunakan dalam sistem *helpdesk*.



Gambar 11. Halaman *Master User*

e. Halaman *My Tickets*:

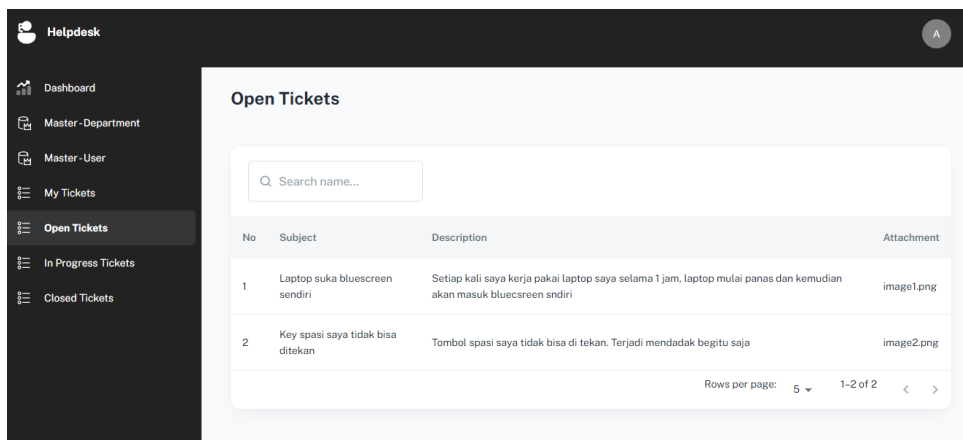
Pada Gambar 12 merupakan halaman *My Tickets* ini hanya dapat diakses oleh pengguna. Pada halaman ini pengguna dapat membuat tiket pelaporan baru, melihat pelaporan yang sudah pernah dibuat, dan memantau proses pelaporan yang sedang diproses.



Gambar 12. Halaman *My Tickets*

f. Halaman *Open Tickets*:

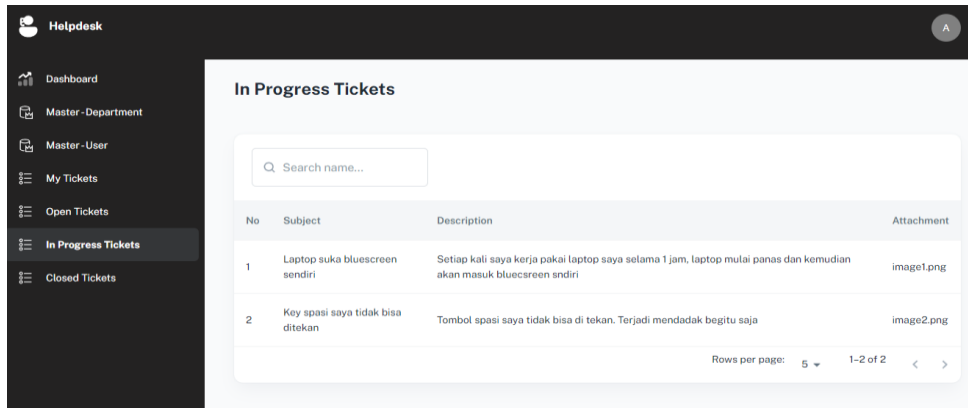
Pada Gambar 13 menggambarkan halaman *Open Tickets*. Halaman ini menyajikan informasi tiket laporan yang menunggu untuk diproses. Halaman *open tickets* hanya dapat diakses oleh agen, yang dimana halaman ini memiliki fitur penting yaitu memproses laporan yang masuk dari pengguna.



Gambar 13. Halaman *Open Tickets*

g. Halaman *In Progress Tickets*:

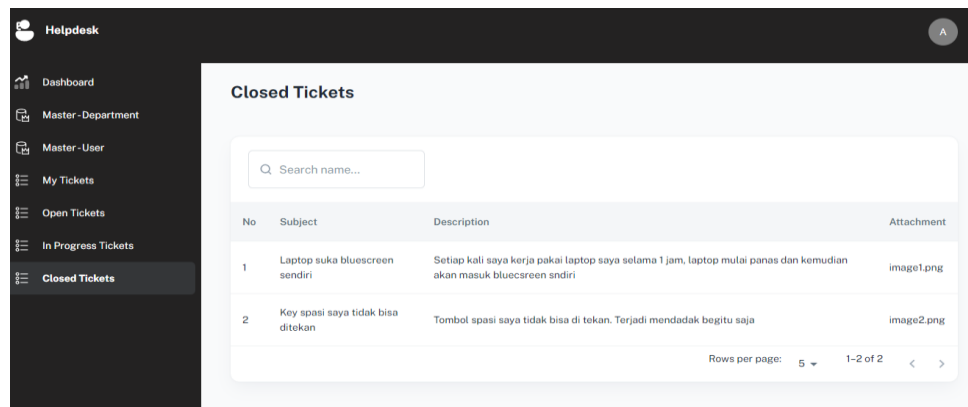
Pada Gambar 14 menampilkan antarmuka halaman *In Progress Tickets*. Halaman ini menyajikan informasi tiket laporan yang sudah diproses dan menunggu untuk diselesaikan. Halaman *in progress tickets* hanya dapat diakses oleh agen, yang dimana halaman ini memiliki fitur penting yaitu menyelesaikan laporan yang masuk dari pengguna.



Gambar 14. Halaman *In Progress Tickets*

h. Halaman *Closed Tickets*:

Pada Gambar 15 merupakan halaman *Closed Tickets*. Halaman ini menyajikan informasi tiket laporan yang telah selesai diproses. Halaman *closed tickets* hanya dapat diakses oleh agen, yang dimana halaman ini memiliki fitur untuk melihat sejarah laporan pengguna yang terselesaikan.



Gambar 15. Halaman *Closed Tickets*

3.4 Testing

Setelah tahap *coding*, langkah berikutnya adalah pengujian menggunakan metode *black-box testing* dengan *Usability Testing*. Proses ini melibatkan penilaian kemudahan pengguna dalam menggunakan sistem dan seberapa baik sistem yang dibangun dalam memenuhi kebutuhan pengguna. Hasil pengujian sangat memuaskan dan sesuai dengan sasaran awal, dengan beberapa hasil pengujian terdokumentasikan dalam Tabel 3.

Tabel 3. Hasil Pengujian *Black Box*

No	Pengujian	Hasil
1	Memastikan proses login mudah dipahami dan berhasil	Berhasil Login
2	Admin dapat membuat master data departemen, dan pengguna	Data departemen dan pengguna berhasil dibuat
3	Pengguna dapat membuat tiket laporan dan memeriksa kemudahan dalam membuat tiket	Tiket berhasil dibuat
4	Agen dapat memproses tiket laporan yang masuk dengan mudah dan berhasil	Tiket dapat dilihat dan diproses dengan baik
5	Menilai keefektifan antarmuka pengelolaan tiket	Berhasil mengelola tiket
6	Memeriksa kemudahan dalam melihat dan menerima tiket laporan	Berhasil melihat dan menerima tiket dengan baik

4. KESIMPULAN

Berdasarkan pembahasan dari penelitian ini penulis telah berhasil mengembangkan sebuah aplikasi berbasis web yaitu *helpdesk ticketing system* yang terbukti membantu pengguna dalam memantau laporan masalah hingga terselesaikan dan meningkatkan efektivitas pelaporan keluhan pengguna. Hasil pengujian fungsi aplikasi dengan metode pengujian

blackbox telah sesuai dengan skenario dan kebutuhan pengguna. Dengan demikian, metode *Extreme Programming (XP)* yang digunakan dari tiga jenis metode pada penelitian ini terbukti efektif dalam membantu pengembangan *helpdesk ticketing system*. XP yang responsif terhadap perubahan kebutuhan pengguna dan fokus pada kualitas kode serta pengujian, telah memberikan hasil yang memuaskan dalam memenuhi kebutuhan pengguna dan menciptakan sistem yang efektif. Berdasarkan hasil wawancara bahwa pendekatan XP sangat sesuai untuk pengembangan sistem seperti *helpdesk ticketing system* yang memerlukan adaptabilitas, kerjasama tim, dan pengiriman cepat.

REFERENCES

- [1] K. M. Darryl Aglibar, G. T. Christopher Alegre, G. I. Del Mundo, and K. O. Francis Duro, "Ticketing System: A Descriptive Research on the Use of Ticketing System for Project Management and Issue Tracking in IT Companies," *International Journal of Computing Sciences Research*, vol. 7, 2023, doi: 10.25147/ijcsr.2017.001.1.90.
- [2] F. Gohil and M. Vikash Kumar, "Ticketing System," *International Journal of Trend in Scientific Research and Development (IJTSRD)*, vol. 3, no. 4, pp. 155-156, 2019.
- [3] P. Mauliana, W. Wiguna, and A. Y. Permana, "Pengembangan E-Helpdesk Support System Berbasis Web di PT Akur Pratama," *JURNAL RESPONSIF*, vol. 2, no. 1, pp. 19-29, 2020.
- [4] M. Saepuloh Politeknik Piksi Ganesha and J. Management Informatika Jl Jend Gatot Subroto, "Perancangan Sistem Informasi Manajemen Helpdesk Berbasis Web Dengan Framework Codeigniter Dan MySQL," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 4, 2021.
- [5] U. A. Rachmawati, S. Adam, and H. Alwi, "PEMBANGUNAN HELPDESK TICKETING SYSTEM BERBASIS WEB (STUDI KASUS: UNIVERSITAS YARSI)," *Jurnal Teknologi Informasi YARSI (JTIY)*, vol. 6, no. 1, pp. 19-24, 2019.
- [6] R. M. Bahrudin, M. Ridwan, and H. S. Darmojo, "Penerapan Helpdesk Ticketing System Dalam Penanganan Keluhan Penggunaan Sistem Informasi Berbasis Web," *JUTIS*, vol. 7, no. 1, 2019.
- [7] K. Lesmana and Y. Ramdhani, "PERANCANGAN HELPDESK TICKETING DAN PROJECT MANAGEMENT SYSTEM MENGGUNAKAN METODE SCRUM (STUDI KASUS: PT IHSAN SOLUSI INFORMATIKA)," *PROTEKTIF*, vol. 2, no. 1, Juni 2021.
- [8] R. Tarigan, I. Kusosi, and A. Usri, "Perancangan Aplikasi Helpdesk Ticketing System Pada PT. Indonesia Nippon Seiki," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 11, no. 1, pp. 9-18, Jan. 2022, doi: 10.32736/sisfokom.v11i1.1271.
- [9] M. Chendra and M. Siahaan, "Perancangan Aplikasi Sistem Informasi Akuntansi Berbasis Website Pada Royale Pet Shop & Care Batam Menggunakan Codeigniter", 4th Natl. Conf. Community Serv. Proj. 2022.
- [10] S. I. Adam, J. H. Moedjahedy, and O. Lengkong, "Pengembangan IT Helpdesk Ticketing Sistem Berbasis Web di Universitas Klabat Development of Web-based IT Helpdesk Ticketing System at Universitas Klabat," *Cogito Smart Journal |*, vol. 6, no. 2, 2020.
- [11] N. Dwivedi, Katiyar Devesh, and Goel Gaurav, "A Comparative Study of Various Software Development Life Cycle (SDLC) Models," 2022.
- [12] M. N. Arifin and D. Siahaan, "Structural and Semantic Similarity Measurement of UML Use Case Diagram," *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, vol. 11, no. 2, p. 88, Jul. 2020, doi: 10.24843/kjiti.2020.v11.i02.p03.
- [13] K. ' Afiifah, Z. Fira Azzahra, A. D. Anggoro, D. Redaksi, R. Akhir, and D. Online, "Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database: Sebuah Literature Review," *JURNAL INTECH*, vol. 3, no. 1, pp. 8-11, 2022.
- [14] R. I. Borman, A. T. Priandika, and A. R. Edison, "Implementasi Metode Pengembangan Sistem Extreme Programming (XP) pada Aplikasi Investasi Peternakan," *Jurnal Sistem dan Teknologi Informasi (Justin)*, vol. 8, no. 3, p. 272, Jul. 2020, doi: 10.26418/justin.v8i3.40273.
- [15] A. Shrivastava, I. Jaggi, N. Katoch, D. Gupta, and S. Gupta, "A Systematic Review on Extreme Programming," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jul. 2021. doi: 10.1088/1742-6596/1969/1/012046.
- [16] M. Hron and N. Obwegeser, "Why and how is Scrum being adapted in practice: A systematic review," *Journal of Systems and Software*, vol. 183, Jan. 2022, doi: 10.1016/j.jss.2021.111110.
- [17] V. Hema, S. Thota, S. Naresh Kumar, C. Padmaja, C. B. Rama Krishna, and K. Mahender, "Scrum: An Effective Software Development Agile Tool," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing Ltd, 2020. doi: 10.1088/1757-899X/981/2/022060.
- [18] K. Saeedi and A. Visvizi, "education sciences Software Development Methodologies, HEIs, and the Digital Economy," 2021, doi: 10.3390/educsci.
- [19] I. K. Raharjana, D. Siahaan, and C. Fatichah, "User Stories and Natural Language Processing: A Systematic Literature Review," *IEEE Access*, vol. 9, pp. 53811-53826, 2021, doi: 10.1109/ACCESS.2021.3070606.
- [20] N. K. Al Ghanmi and N. S. M. Jamail, "Integrating scrum development process with ux design flow," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 6, pp. 2630-2636, Dec. 2020, doi: 10.11591/eei.v9i6.2484.