

## Rancang Bangun Aplikasi *Ticketing Service Desk* Teknologi Informasi Berbasis *Website*

Ruswan Hidayat, Nizirwan Anwar

<sup>1,2</sup> Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Esa Unggul

E-mail: [ruswan.hidayat@gmail.com](mailto:ruswan.hidayat@gmail.com)<sup>1</sup>, [nizirwan.anwar@esaunggul.ac.id](mailto:nizirwan.anwar@esaunggul.ac.id)<sup>2</sup>

### ABSTRAK

*Service Desk* adalah titik kontak tunggal (*single point of contact/SPOC*) antara pengguna layanan TI (Teknologi Informasi) dan penyedia layanan TI. *Service Desk* adalah pintu masuk utama bagi pengguna untuk mendapatkan bantuan terkait teknologi informasi. Mereka bertindak sebagai perantara antara pengguna dan tim IT lainnya, memastikan bahwa masalah diselesaikan dengan efisien dan efektif. Penelitian ini bertujuan untuk merancang dan membangun aplikasi *ticketing* bagi unit *service desk*, dengan menerapkan kesepakatan tingkat layanan pada aplikasi tersebut. Aplikasi yang dibangun diharapkan dapat menjadi sarana penunjang bagi kegiatan di *service desk* sekaligus dapat memberikan kepastian bagi pengguna layanan terkait waktu yang diperlukan sampai dengan laporan permasalahan ditindaklanjuti.

**Kata kunci :** *Service Desk, Rancang Bangun, Service Level Agreement, Titik Kontak Tunggal, Tiket, Layanan Teknologi Informasi*

### ABSTRACT

*Service Desk* is a single point of contact (*SPOC*) between IT (*Information Technology*) service users and IT service providers. *Service Desks* are the main entry point for users to get information technology-related assistance. They act as intermediaries between users and other IT teams, ensuring that problems are resolved efficiently and effectively. This research aims to design and build a *ticketing* application for the *service desk* unit, by applying service level agreement to the application. The application built is expected to be a means of supporting activities at the *service desk* as well as being able to provide certainty for service users regarding the time required until the problem report is followed up.

**Keyword :** *Service Desk, Design and Build, Service Level Agreement, Single Point of Contact, Ticketing, IT Service*

## 1. PENDAHULUAN

PT. XYZ merupakan salah satu perusahaan yang bergerak di bidang finansial. Perusahaan ini berkedudukan di Jakarta dengan banyak kantor cabang di kota-kota besar di Indonesia. Untuk mendukung jalannya proses bisnis utama, PT. XYZ menyediakan layanan teknologi informasi dan komunikasi dalam bentuk

Sistem Informasi terpusat. PT. XYZ memiliki sebuah divisi khusus yang bertugas dalam operasional layanan TIK tersebut.

Selain penyediaan layanan TIK, dalam rangka menjalankan fungsi teknologi informasi yang berkesinambungan, divisi IT perusahaan memiliki unit khusus yang berperan sebagai *service desk* bagi

pengguna layanan TIK. *Service desk* bertugas untuk menerima dan menindaklanjuti semua insiden, pertanyaan, dan permintaan teknis terkait layanan TIK dari semua pengguna internal, yaitu pegawai baik di kantor pusat maupun cabang yang menggunakan layanan TIK yang disediakan oleh PT.XYZ.

Aktivitas unit *service desk* telah berjalan dan didukung oleh aplikasi tersendiri. Namun, seiring berjalannya waktu, ditemui beberapa kendala dalam berjalannya fungsi *service desk*, seperti kebutuhan organisasi yang berubah, misalnya perubahan mekanisme kerja di internal unit *service desk* itu sendiri, serta kendala-kendala lain yang berkaitan dengan pengguna layanan, seperti kepastian akan tindak lanjut pelaporan permasalahan atau jaminan penyelesaian pelaporan permasalahan. Tindak lanjut atas pelaporan permasalahan terkadang menjadi lama dan pengguna tidak memiliki kepastian kapan permasalahannya akan terselesaikan. Hal ini terjadi karena aplikasi yang digunakan saat ini dibangun sebelum diberlakukannya *Service Level Agreement* (SLA), sehingga tidak ada kontrol atau pengingat kepada petugas di unit *service desk*. Ketika aturan formal terkait kesepakatan tingkat layanan (SLA) ini diterapkan, aplikasi belum mendukung fungsi tersebut.

Oleh karena itu, penulis berpikir untuk mengembangkan aplikasi *internal support* yang baru bagi unit *service desk*, dengan menerapkan standar waktu layanan pada aplikasi tersebut. Diharapkan aplikasi ini dapat menjadi sarana penunjang bagi kegiatan di *service desk*.

Berdasarkan uraian latar belakang diatas, maka ada beberapa permasalahan yang dapat diidentifikasi, yakni sebagai berikut:

1. Bagaimana merancang dan membangun sebuah aplikasi yang dapat digunakan dalam media pelaporan tiket ke *service desk*?
2. Bagaimana agar pelayanan *service desk* dapat terlaksana dengan baik dan tepat waktu, dengan mengimplementasikan SLA pada aplikasi *service desk*?

Tujuan dari penelitian ini yaitu membuat sebuah aplikasi yang dapat digunakan oleh pengguna layanan sebagai media pelaporan tiket ke unit *service desk*. Selain itu, dengan adanya pembobotan pada tiket dari pengguna, setiap tiket yang dilaporkan dapat dipantau dan diselesaikan sesuai dengan SLA yang berlaku.

Manfaat dari penelitian ini yaitu, menyediakan kepastian bagi pengguna layanan, berapa lama waktu yang diperlukan sampai dengan tiket ditindaklanjuti atas tiket-tiket yang dilaporkan ke *service desk*. Bagi unit *service desk* sendiri, aplikasi ini dapat membantu memantau tunggakan atas tiket-tiket yang dilaporkan, sekaligus membantu memantau kinerja bagi pihak manajemen.

## 2. LANDASAN TEORI

### Rancang Bangun

Menurut (Jogiyanto, 2019) rancang bangun (desain) adalah tahap dari setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan-kebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dari suatu sistem.

## Aplikasi

Menurut (Abdurahman, 2014), aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Sederhananya, dapat disimpulkan bahwa aplikasi adalah jenis perangkat lunak yang memungkinkan pengguna melakukan tugas tertentu, baik itu pada perangkat komputer, perangkat seluler, atau perangkat lainnya.

## Service Desk

Menurut kerangka ITIL (Axelos.com, 2019), definisi dari *service desk* adalah: “*The single point of contact between the service provider and the users. A typical service desk manages incidents and service requests, and also handles communication with the users*”. *Service desk* adalah suatu unit atau pusat komunikasi dimana pengguna bisa mendapatkan bantuan dari penyedia layanan teknologi informasi. Bantuan yang dimaksud dapat berupa pertanyaan, penyelesaian insiden ataupun pemenuhan permintaan layanan. Namun, apapun jenis bantuan yang disediakan, tujuan utama dari sebuah *service desk* adalah memberikan layanan secara tepat waktu kepada pengguna.

## Aplikasi Service Desk

Aplikasi *service desk* merupakan aplikasi yang digunakan oleh pengguna layanan teknologi informasi untuk menyampaikan keluhan, kendala maupun permintaan kepada penyedia layanan teknologi informasi. Aplikasi *service desk* berfungsi sebagai sistem pelaporan terpusat bagi pengguna, yang bertujuan

memudahkan pengguna dalam pengajuan dan pemantauan tiket yang diajukan. Selain itu, aplikasi ini juga membantu unit *service desk* dalam berbagai aspek, seperti pelayanan kepada pengguna, manajemen insiden, maupun monitoring kinerja, sehingga pelayanan yang diberikan oleh unit *service desk* lebih efektif dan efisien.

## Service Level Agreement (SLA)

Menurut (Atan & Haider, 2016) *Service Level Agreement* (SLA) merupakan sebuah alat yang dipakai dalam sebuah kesepakatan secara formal dengan pelanggan yang dalam hal ini adalah seputar pelayanan yang diterima dan diberikan dengan sejumlah ukuran performansinya.

## Unified Modelling Language (UML)

UML adalah bahasa untuk menspesifikasi, mem-visualisasi, membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah bahasa pemrograman yang menggunakan konsep orientasi objek (Destriana et al., 2021).

## Use Case Diagram dan Use Case Scenario

Menurut (Julianto & Setiawan, 2019), *use case diagram* merupakan diagram yang menggambarkan hubungan antara aktor dengan sistem. Diagram ini sangat penting dalam mengatur dan memodelkan perilaku suatu sistem. Terdapat tiga unsur dalam komponen *use case diagram* menurut (Ansori, 2020) yaitu:

1. Sistem  
Sistem dalam use case diagram menggambarkan dalam bentuk persegi yang memberi batas use case dengan antar hubungan di luar sistem. Sistem pada prinsipnya dapat diberikan penamaan atau label yang telah disesuaikan dengan nama pelabelan tersebut.
2. Aktor  
Peran aktor untuk menginformasikan atau menyampaikan informasi kepada sistem atau sebaliknya sistem dapat memberikan informasi kepada aktor atau keduanya bersamaan saling memberikan atau menerima informasi.
3. Use Case  
*Use Case* adalah uraian atau penjelasan mengenai fungsional sistem agar konsumen juga desainer saling memahami atas kebutuhan fungsi-fungsi didalam sistem yang sedang dirancang atau dikembangkan.

#### **Activity Diagram**

*Activity diagram* merupakan pemodelan yang menggambarkan sebuah sistem kerja dari sebuah objek atau sebuah sistem, sebuah *activity diagram* digambarkan dengan sebuah alur secara terstruktur proses kerja dari *use case* yang sedang diproses dari titik awal sampai titik akhir, setiap aktivitas digambarkan dengan notasi-notasi sesuai fungsinya (Kusumanto & Tomponu, 2011).

#### **Sequence Diagram**

Menurut (Nugroho, 2010) *sequence diagram* adalah diagram yang dibuat untuk mengetahui alur dari interaksi antar objek.

#### **Class Diagram**

Sebuah *class diagram* akan menunjukkan bagaimana skema dari arsitektur sebuah sistem yang sedang dirancang (Kendal, 2009). *Class diagram* digambarkan dengan *class* yang berisi atribut dan *method*, setiap *class* akan

dihubungkan dengan sebuah garis disebut asosiasi.

#### **PHP**

Menurut (Solichin, 2016), PHP merupakan salah satu bahasa pemrograman berbasis web yang ditulis oleh dan untuk pengembang web. PHP pertama kali dikembangkan oleh Rasmus Lerdorf, seorang pengembang software dan anggota tim Apache, dan dirilis pada akhir tahun 1994.

#### **MariaDB**

Dikutip dari (Jaya, 2022), *MariaDB* merupakan jenis *database* yang sumbernya terbuka (*open source*). Setiap orang dapat dengan bebas mengunduh kode sumber dan dapat mengembangkannya jika mereka mau. Sejak pertama rilis, *MariaDB* telah mendapat sambutan yang besar dari para pengembang perangkat lunak, terutama pengembang *website*.

### **3. METODOLOGI**

#### **Jenis Penelitian**

Penelitian ini dilakukan dengan menggunakan metode kualitatif. Menurut Creswell, penelitian kualitatif adalah pendekatan kualitatif untuk pengumpulan data, analisis, interpretasi, dan penulisan laporan berbeda dari pendekatan kuantitatif tradisional, pengambilan sampel secara sengaja, pengumpulan data terbuka, analisis teks atau gambar, representasi informasi dalam gambar dan tabel, dan interpretasi pribadi dari temuan semua menginformasikan metode kualitatif (Cresswel, 2010).

Untuk tahapan pengembangan perangkat lunak, metode yang digunakan adalah metode *prototype*. Metode *prototype* merupakan metode pengembangan perangkat lunak yang memungkinkan adanya interaksi antara pengembang sistem dengan pengguna sistem, sehingga dapat mengatasi

ketidakserasian antara pengembang dan pengguna (R. R. Pressman, 2012).

### Alur Penelitian

Berikut ini merupakan bagan dari tahap-tahap penelitian ini:



Gambar 1 Alur penelitian

1. Identifikasi Masalah  
Sebelum memulai penelitian, terlebih dahulu ditentukan kasus yang akan diteliti. Kemudian, kegiatan dimulai dengan identifikasi permasalahan yang terjadi pada lokasi studi kasus.
2. Analisis Kebutuhan  
Dari hasil identifikasi masalah, penulis mempelajari kondisi sistem yang berjalan saat ini.
3. Perancangan  
Berdasarkan kebutuhan yang telah didefinisikan sebelumnya, dilanjutkan dengan desain perancangan aplikasi sebagai *guideline* sebelum implementasi pemrograman. Desain aplikasi dibuat dengan menggunakan UML diagram.
4. Implementasi  
Kemudian, tahapan dilanjutkan dengan proses implementasi pemrograman.
5. Pengujian  
Penelitian dilanjutkan kembali ke tahapan pengujian untuk mengetahui apakah aplikasi yang telah dibuat, dapat berfungsi sebagaimana mestinya. Pengujian direncanakan menggunakan metode *Blackbox Testing*.
6. Evaluasi  
Proses dilanjutkan di tahap akhir, dilakukan proses evaluasi untuk menilai apakah aplikasi yang telah dibangun, sudah sesuai dengan kebutuhan dan harapan *stakeholder*.

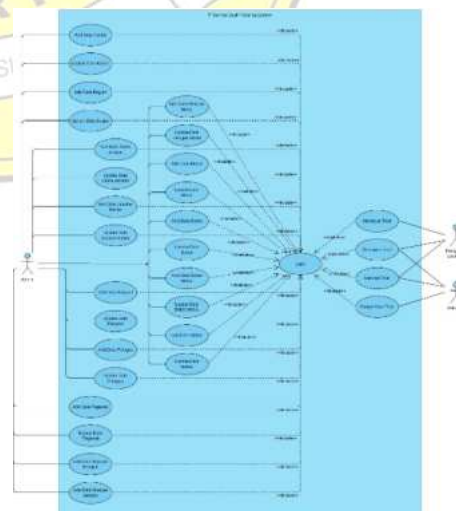
### Metode Pengumpulan Data

Pada penelitian ini, pengumpulan data dilakukan dengan beberapa metode, antara lain:

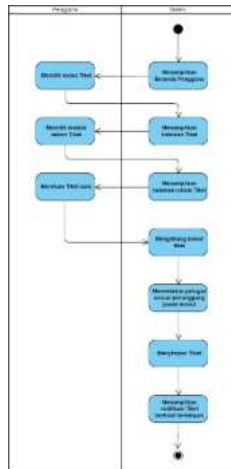
1. Studi literatur: Menurut Zed dikutip dari (Kartiningrum, 2015), metode studi literatur adalah serangkaian kegiatan yang berkenaan dengan metode pengumpulan data pustaka, membaca dan mencatat, serta mengelolah bahan penelitian. Studi literatur bertujuan untuk membangun landasan teori, kerangka berpikir, memahami permasalahan serta menentukan hipotesis penelitian, dengan bersumber kepada buku-buku ilmiah, hasil penelitian, peraturan dan sumber tertulis lainnya.
2. Observasi langsung: mengamati dan mempelajari secara langsung aplikasi *existing* serta proses bisnis yang berjalan saat ini.
3. Wawancara: selain melakukan observasi langsung, penulis juga melakukan wawancara secara langsung dengan pegawai yang bertugas pada unit *service desk*.

### Perancangan

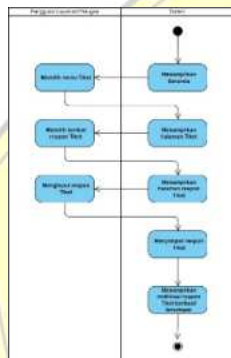
Berikut ini merupakan usulan diagram untuk sistem yang akan dibangun.



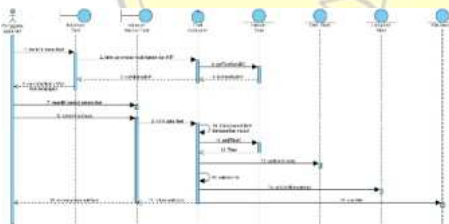
Gambar 2 Usulan Use Case Diagram



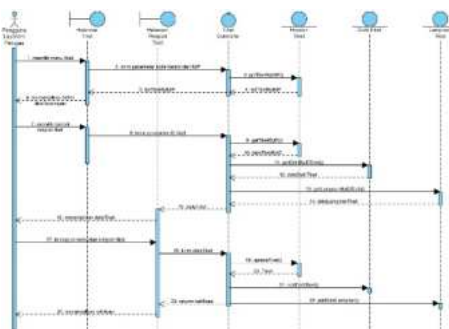
Gambar 3 Usulan Activity Diagram



Gambar 4 Usulan Activity Diagram



Gambar 5 Usulan Sequence Diagram



Gambar 6 Usulan Sequence Diagram

#### 4. HASIL DAN PEMBAHASAN

##### Implementasi User Interface - Admin

Pada hak akses administrator, tampilan aplikasi berisi manajemen data referensi yang digunakan untuk operasional aplikasi ticketing. Semua menu tersebut memiliki desain tampilan yang sama untuk semua jenis data referensi. Secara umum, jenis tampilan pada hak akses administrator terbagi menjadi 3, yaitu halaman daftar data, tambah data baru dan ubah data. Berikut ini merupakan tampilan antarmuka pengguna pada hak akses administrator.

##### 1. Tampilan Daftar Data Referensi

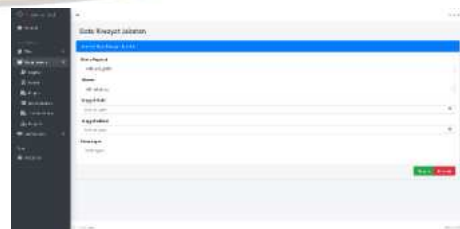
Pada halaman ini, menampilkan data-data yang telah tersimpan pada aplikasi.



Gambar 7 Tampilan menu Data Referensi

##### 2. Tampilan Tambah Data

Pada halaman ini, menampilkan form untuk menambahkan data referensi baru. Administrator dapat mengakses halaman ini dengan menekan tombol tambah data pada setiap halaman data referensi.



Gambar 8 Tampilan menu Tambah Data Referensi

3. Tampilan Ubah Data

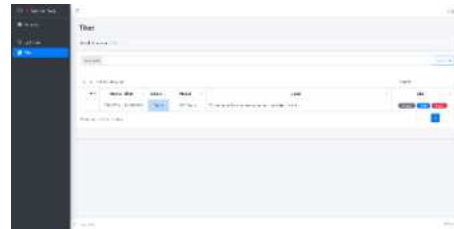
Pada halaman ini, menampilkan *form* untuk memperbarui data referensi. Administrator dapat mengakses halaman ini dengan menekan tombol edit pada kolom aksi di setiap halaman data referensi.



Gambar 9 Tampilan menu Edit Data Referensi

3. Tampilan Pencarian Tiket

Pada halaman ini, menampilkan *form* untuk melakukan pencarian tiket.



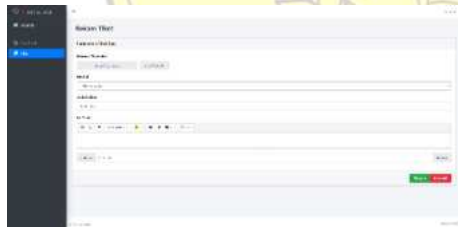
Gambar 12 Tampilan menu Pencarian Tiket

**Implementasi User Interface – Pengguna Layanan**

Berikut beberapa tampilan aplikasi pada hak akses pengguna layanan.

1. Tampilan Rekam Tiket

Pada halaman ini, menampilkan *form* untuk merekam pembuatan tiket baru.



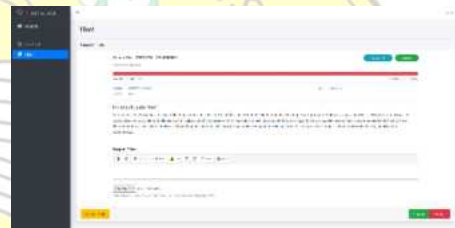
Gambar 10 Tampilan menu Rekam Tiket

**Implementasi User Interface – Petugas**

Berikut beberapa tampilan aplikasi pada hak akses petugas.

4. Tampilan Respon Tiket

Pada halaman ini, menampilkan *form* untuk merespon tiket.



Gambar 13 Tampilan menu Respon Tiket - Petugas

2. Tampilan Respon Tiket

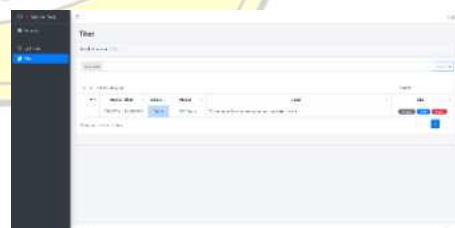
Pada halaman ini, menampilkan *form* untuk merespon tiket.



Gambar 11 Tampilan menu Respon Tiket - Pengguna

5. Tampilan Pencarian Tiket

Pada halaman ini, menampilkan *form* untuk melakukan pencarian tiket.



Gambar 14 Tampilan menu Pencarian Tiket

6. Tampilan Alih Tiket

Pada halaman ini, menampilkan *form* untuk melakukan pengalihan tiket.



Gambar 15 Tampilan menu Alih Tiket

### Pengujian Fungsionalitas

Pengujian fungsionalitas adalah pengujian fungsi-fungsi yang berjalan pada aplikasi berdasarkan skenario tertentu. Pada penelitian ini, proses pengujian dilakukan dengan metode *black box*. *Black box testing* menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program (A & Shalahuddin, 2018). Pengujian fungsionalitas dijelaskan pada tabel berikut.

Tabel 1 Hasil pengujian fungsional

No	Skenario	Hasil yang diharapkan	Ket
1.	Inisiasi aplikasi	Menampilkan halaman <i>login</i> aplikasi	Sesuai
2.	<i>Login</i> aplikasi	1. Aktor dapat meng- <i>input</i> <i>username</i> dan <i>password</i> 2. Aktor dapat <i>login</i> ke aplikasi 3. Aplikasi menampilkan halaman beranda sesuai dengan hak akses <i>user</i>	Sesuai
Pegguna Layanan			
3.	Akses Halaman Tiket	1. Menampilkan halaman yang berisi daftar tiket 2. Daftar yang ditampilkan merupakan tiket yang direkam sesuai dengan pengguna yang mengakses aplikasi	Sesuai
4.	Merekam Tiket	1. Aplikasi menampilkan halaman perekaman tiket 2. Pengguna dapat memilih modul pada <i>dropdown</i> yang tersedia 3. Pengguna dapat meng- <i>input</i> judul dan isi tiket 4. Pengguna dapat menambahkan <i>attachment</i> 5. Aplikasi menyimpan tiket kemudian	Sesuai

		menampilkan halaman daftar tiket	
5.	Merespon Tiket	1. Aplikasi menampilkan halaman respon tiket 2. Pengguna dapat meng- <i>input</i> isi respon 3. Pengguna dapat menambahkan <i>attachment</i> 4. Aplikasi menyimpan tiket kemudian menampilkan halaman daftar tiket	Sesuai
6.	Melihat Detil Tiket	Aplikasi menampilkan halaman detil tiket	Sesuai
7.	Mencari Tiket	1. Aplikasi menampilkan halaman pencarian tiket 2. Pengguna dapat meng- <i>input</i> <i>keyword</i> pencarian 3. Aplikasi melakukan pencarian dan menampilkan daftar hasil pencarian tiket	Sesuai
8.	Menghapus Tiket	1. Aplikasi menampilkan permintaan konfirmasi hapus tiket 2. Aplikasi menghapus tiket sesuai dengan pilihan pengguna	Sesuai
Petugas			
9.	Akses Halaman Tiket	1. Menampilkan halaman yang berisi daftar tiket yang perlu ditindaklanjuti 2. Daftar yang ditampilkan merupakan tiket sesuai dengan <i>assignment</i>	Sesuai
10.	Merespon Tiket	1. Aplikasi menampilkan halaman respon tiket 2. Aplikasi menampilkan sisa waktu respon tiket 3. Petugas dapat meng- <i>input</i> isi respon 4. Petugas dapat menambahkan <i>attachment</i> 5. Aplikasi menyimpan tiket kemudian menampilkan halaman daftar tiket	Sesuai
11.	Melihat Detil Tiket	Aplikasi menampilkan halaman detil tiket	Sesuai
12.	Mencari Tiket	1. Aplikasi menampilkan halaman pencarian tiket 2. Petugas dapat meng- <i>input</i> <i>keyword</i> pencarian 3. Aplikasi melakukan pencarian dan	Sesuai



		menampilkan daftar hasil pencarian tiket	
13	Mengalihkan Tiket	<ol style="list-style-type: none"> <li>1. Aplikasi menampilkan halaman pengalihan tiket</li> <li>2. Petugas dapat memilih modul sesuai dengan <i>dropdown</i> yang tersedia</li> <li>3. Aplikasi meng-<i>update</i> data tiket, mendistribusikan ulang dan kemudian menampilkan daftar tiket</li> </ol>	Sesuai
Administrator			
14.	Akses Halaman Referensi	Aplikasi menampilkan halaman yang berisi data referensi tersimpan	Sesuai
15.	Menambah Referensi	<ol style="list-style-type: none"> <li>1. Aplikasi menampilkan halaman tambah data referensi</li> <li>2. Administrator dapat mengisi <i>form</i> tambah data referensi</li> <li>3. Aplikasi menyimpan data referensi kemudian menampilkan halaman daftar data referensi</li> </ol>	Sesuai
16.	Meng- <i>edit</i> Referensi	<ol style="list-style-type: none"> <li>1. Aplikasi menampilkan halaman <i>edit</i> data referensi</li> <li>2. Administrator dapat mengubah data sesuai <i>form edit</i> data referensi</li> <li>3. Aplikasi meng-<i>update</i> data referensi kemudian menampilkan halaman daftar data referensi</li> </ol>	Sesuai
17.	Menghapus Referensi	<ol style="list-style-type: none"> <li>1. Aplikasi menampilkan permintaan konfirmasi hapus data referensi</li> <li>2. Aplikasi menghapus data referensi sesuai dengan pilihan administrator</li> </ol>	Sesuai
18.	Mengalihkan Tiket	<ol style="list-style-type: none"> <li>1. Aplikasi menampilkan halaman pengalihan tiket</li> <li>2. Administrator dapat memilih modul sesuai dengan <i>dropdown</i> yang tersedia</li> <li>3. Administrator dapat memilih petugas sesuai dengan <i>dropdown</i> yang tersedia</li> <li>4. Aplikasi meng-<i>update</i> data tiket, mendistribusikan ulang</li> </ol>	Sesuai

		dan kemudian menampilkan daftar tiket	
19.	Akses Halaman Distribusi Tiket	Aplikasi menampilkan halaman yang berisi daftar tiket <i>unassigned</i>	Sesuai
20.	Distribusi Tiket	Aplikasi mendistribusikan tiket <i>unassigned</i> sesuai dengan modul tiket	Sesuai

## 5. KESIMPULAN

Berdasarkan uraian pada bab-bab sebelumnya, maka dapat ditarik kesimpulan dari penelitian ini yaitu sebagai berikut:

1. Penelitian ini menghasilkan sebuah aplikasi berbasis *web* yang dapat digunakan sebagai media pelaporan permasalahan, dari pengguna layanan kepada unit kerja TIK. Aplikasi yang dibangun telah dilengkapi dengan penerapan standar waktu penyelesaian tiket.
2. Dengan telah diterapkannya standar waktu layanan, dapat memberikan informasi kepada pengguna terkait waktu penyelesaian tiket. Penerapan standar waktu pada aplikasi diharapkan dapat menjadi alternatif solusi dalam peningkatan pelayanan kepada pengguna secara tepat waktu.

## DAFTAR PUSTAKA

- A, R., & Shalahuddin, M. (2018). Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. In *Informatika Bandung*.
- Abdurahman, H. dan A. R. R. (2014). Aplikasi Pembayaran Secara Kredit Pada Bank Yudha Bahakti, Jurnal Computech & Bisnis. *Jurnal Computech & Bisnis*.
- Ansori, A. (2020). Pengertian Use Case Diagram : Tujuan, Fungsi, Simbol, dan Contohnya. In *Www.Ansoriweb.Com*.
- Atan, R., & Haider, H. (2016). Service Level Agreement (SLA) Enforcement: Is it Impactful? *Indian Journal of Science and Technology*. <https://doi.org/10.17485/ijst/2016/v9i48/109302>
- Axelos.com. (2019). *ITIL ® Foundation ITIL 4 edition Glossary*.
- Cresswel, J. W. (2010). Research Design; Pendekatan Kualitatif, Kuantitatif Dan Mixed. In *Yogyakarta: Pustaka Pelajar* (Edisi Keem). Pustaka Pelajar.
- Destriana, R., Husain, S. M., Handayani, N., & Siswanto, A. T. P. (2021). Diagram UML Dalam Membuat Aplikasi Android Firebase. In *Studi Kasus Aplikasi Bank Sampah". Deepublish* (Vol. 1, Issue 1). Deepublish.
- Jaya, S. A. P. K. (2022). *Pembangunan Aplikasi Berbagi Wi-Fi Gratis*.
- Jogiyanto. (2019). Analisis dan Desain Sistem Informasi, Sistem Informasi. *Yogyakarta: Andi Offset*.
- Julianto, S., & Setiawan, S. (2019). Perancangan Sistem Informasi Pemesanan Tiket Bus Pada Po. Handoyo Berbasis Online. *Jurnal Intra Tech*, 3(2), 11–25. <https://doi.org/10.37030/jit.v3i2.56>
- Kartiningrum, E. D. (2015). Panduan Penyusunan Studi Literatur. *Lembaga Penelitian Dan Pengabdian Masyarakat Politeknik Kesehatan Majapahit, Mojokerto*, 1–9.
- Kendal, S. (2009). *Object oriented programming using Java*.
- Kusumanto, R. D., & Tompunu, A. N. (2011). PENGOLAHAN CITRA DIGITAL UNTUK MENDETEKSI OBYEK MENGGUNAKAN PENGOLAHAN WARNA MODEL NORMALISASI RGB. In *Seminar Nasional Teknologi Informasi & Komunikasi Terapan*.
- Nugroho, A. (2010). Rekayasa Perangkat Lunak Menggunakan UML dan Java - Adi Nugroho - Google Books. *Rekayasa Perangkat Lunak Menggunakan UML & Java*.
- R. R. Pressman. (2012). Rekayasa Perangkat Lunak satu Pendekatan Praktisi. In *Skripsi Teknik Informatika: Vol. Edisi 7* (Edisi 7). Andi.
- Solichin, A. (2016). Pemrograman Web dengan PHP dan MySQL - Achmad Solichin - Google Buku. *Universitas Budi Luhur*.