

**LAPORAN PRAKTEK KERJA LAPANGAN  
SISTEM MANAJEMEN PRODUK DAN TICKETING  
BAGIAN MODUL TICKETING  
DI PT TEKNOLOGI APLIKASI SEJAHTERA**



**Disusun Oleh:**  
Givandra Haikal Adjie  
240601211130063

**DEPARTEMEN INFORMATIKA  
FAKULTAS SAINS DAN MATEMATIKA  
UNIVERSITAS DIPONEGORO  
SEMARANG  
2023**

## ABSTRAK

Manajemen ticketing adalah bagian integral dari platform yang mengelola aspek-aspek terkait dengan aplikasi yang telah diluncurkan oleh PT Teknologi Aplikasi Sejahtera. Sebagai perusahaan yang menghasilkan berbagai produk sistem informasi untuk kliennya, PT Teknologi Aplikasi Sejahtera memerlukan sistem informasi berbasis web untuk mengelola ticketing system yang membantu dalam menyelesaikan bug dan memberikan notifikasi untuk setiap tiket yang dilaporkan oleh pengguna. Pengembangan dan pengujian website ini mengikuti metode waterfall dan black box testing. Proses dimulai dengan analisis kebutuhan melalui wawancara dengan pihak perusahaan untuk menentukan kebutuhan pengguna dan persyaratan sistem. Website dikembangkan menggunakan bahasa PHP dengan framework Laravel dan diintegrasikan dengan REST API. SQL digunakan sebagai bahasa untuk basis data relasional, dengan PostgreSQL sebagai database management system-nya. Antarmuka website didesain intuitif untuk memudahkan pengguna dalam mengajukan tiket dan melacak status tiket yang dilaporkan. Selain itu, website ini juga dilengkapi dengan fitur notifikasi WhatsApp untuk tiket baru, memastikan pengguna mendapatkan informasi terkini mengenai status tiket mereka. Dengan adanya sistem ini, efisiensi proses manajemen tiket di PT Teknologi Aplikasi Sejahtera dapat ditingkatkan secara signifikan, memungkinkan penyelesaian masalah yang lebih cepat dan komunikasi yang lebih baik antara pengguna dan tim pengembang.

**Kata kunci :** Customer service, Manajemen produk, Ticketing, Perangkat lunak, Bug, Notifikasi

## ABSTRACT

Ticketing is a crucial aspect of the platform that manages applications launched by PT Teknologi Aplikasi Sejahtera. As a company that produces various information system products for its clients, PT Teknologi Aplikasi Sejahtera requires a web-based information system to efficiently handle the ticketing system, which aids in resolving bugs and provides notifications for every ticket reported by users. The development and testing of this ticketing website follows the waterfall and black box testing methods. The process begins with a needs analysis conducted through interviews with the company to determine user needs and system requirements. The website is developed using the PHP language with the Laravel framework and is integrated with REST API. SQL is employed for relational databases, with PostgreSQL as the database management system. The interface is designed to be intuitive, facilitating users in submitting tickets and tracking their status. Additionally, the website includes features like WhatsApp notifications for new tickets and comprehensive product documentation management. These enhancements significantly improve the efficiency of the ticket management process at PT Teknologi Aplikasi Sejahtera, ensuring quicker resolution of issues and better communication between users and the development team.

**Keywords :** Customer service, Product management, Ticketing, Software, Bug, Notification

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala berkat rahmat dan hikmat-Nya penulis dapat menyelesaikan Laporan Praktek Kerja Lapangan “Sistem Manajemen Produk dan Ticketing bagian Modul Ticketingdi PT Teknologi Aplikasi Sejahtera”. Penulis menyadari dalam menyelesaikan kegiatan PKL ini sangatlah sulit tanpa bantuan dan bimbingan dari berbagai pihak. Bersama ini, penulis menyampaikan terima kasih kepada:

1. Bapak Prof. Dr. Yos Johan Utama, S.H., M.Hum, selaku Rektor Universitas Diponegoro Semarang;
2. Ibu Prof. Dr. Widowati, S.Si., M.Si., selaku dekan Fakultas Sains dan Matematika yang telah memberikan izin untuk melakukan Praktek Kerja Lapangan di PT Teknologi Aplikasi Sejahtera;
3. Bapak Dr. Aris Puji Widodo, S.Si, M.T. selaku Ketua Departemen Informatika yang telah membantu dalam proses perizinan PKL di PT Teknologi Aplikasi Sejahtera;
4. Ibu Beta Noranita, S.Si., M.Kom. selaku Dosen Pembimbing PKL yang telah membimbing penulis hingga terselesaikannya PKL ini;
5. Bapak Sandy Kurniawan, S.Kom., M.Kom. selaku Koordinator PKL Departemen Informatika yang telah memberikan bimbingan serta arahan mengenai pelaksanaan PKL;
6. PT Teknologi Aplikasi Sejahtera;
7. Serta semua pihak yang telah terlibat membantu kelancaran dan pelaksanaan dalam kegiatan ini.

Penulis menyadari bahwa masih banyak kekurangan dalam laporan PKL ini. Maka dari itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga laporan ini dapat bermanfaat bagi semua pihak.

Semarang, ...

Givandra Haikal Adjie



## **DAFTAR ISI**

## **DAFTAR GAMBAR**

## **DAFTAR TABEL**

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Kemajuan teknologi dan digitalisasi yang cepat telah memberikan banyak kemudahan dalam menjalankan pekerjaan, termasuk bagi perusahaan dan bisnis yang menggunakan teknologi dalam operasional mereka. Untuk tetap bersaing di pasar yang kompetitif, perusahaan harus terus beradaptasi dengan teknologi, termasuk dengan membangun berbagai sistem informasi untuk menjaga keunggulan mereka. PT Teknologi Aplikasi Sejahtera, sebuah perusahaan pengembang perangkat lunak, juga mengalami peningkatan dalam intensitas manajemen produk dan layanan pelanggan seiring dengan pertumbuhan perangkat lunak yang mereka hasilkan setiap tahunnya.

Namun, meskipun telah menghasilkan beragam aplikasi untuk kliennya, manajemen tiket dari klien masih dilakukan secara manual melalui WhatsApp dengan perantara salah satu karyawan disana yang mengurus bagian ini sebagai tugas tambahan. Hal ini menimbulkan tantangan efisiensi dalam mengakses informasi serta kesalahan manusia.

Oleh karena itu, diperlukan pembuatan sebuah sistem informasi berbasis web yang bertujuan untuk meningkatkan efisiensi dalam manajemen bug dan pelaporan, serta penyimpanan dokumentasi produk. Berdasarkan analisis awal, terdapat beberapa bagian yang dapat dikerjakan secara terpisah, salah satunya adalah modul ticketing yang akan difokuskan pada manajemen ticket. Modul ini mencakup pengelolaan tiket dari pembuatan, administrasi internal, hingga penutupan, serta berfungsi sebagai perantara antara pihak eksternal dan internal. Sistem informasi yang kami rencanakan akan menggunakan PHP dengan framework Laravel sebagai backend, PostgreSQL sebagai manajemen basis data, dan React.js sebagai frontend.

### 1.2. Rumusan Masalah

Berdasarkan latar belakang dan permasalahan yang telah dibahas, maka rumusan masalah dari proyek PKL ini adalah bagaimana merancang dan membuat Sistem Manajemen Produk dan Ticketing modul Ticketing di PT Teknologi Aplikasi Sejahtera.

### 1.3. Tujuan

Tujuan dilaksanakannya praktik kerja lapangan ini adalah untuk menghasilkan Sistem Manajemen Produk dan Ticketing PT Teknologi Aplikasi Sejahtera yang dapat membantu permasalahan pengelolaan ticket yang masih dilakukan secara manual.

### 1.4. Manfaat

Sedangkan untuk manfaat dilaksanakannya praktik kerja lapangan ini adalah agar PT Teknologi Aplikasi Sejahtera dapat menggunakan Sistem Manajemen Produk dan Ticketing ini dalam meningkatkan efisiensi dari mengelola ticket lebih baik serta menyediakan tempat untuk mengarsipkan ticket yang sudah selesai secara terpusat.

### **1.5. Ruang Lingkup**

Ruang lingkup dalam praktik kerja lapangan ini adalah Sistem Manajemen Produk dan Ticketing yang berfokus pada modul ticketing.

### **1.6. Sistematika Penulisan**

Berdasarkan latar belakang dan permasalahan yang telah dibahas, maka berikut adalah rumusah masalah dari projek yang dibuat:

#### **BAB I PENDAHULUAN**

Bab ini berisikan mengenai informasi perusahaan tempat kegiatan Praktek Kerja Lapangan dilaksanakan, yaitu PT. Teknologi Aplikasi Sejahtera disertai dengan profil instansi, visi, misi, dan struktur organisasi.

#### **BAB II LANDASAN TEORI**

Bab ini membahas mengenai landasan teori yang digunakan dalam pembangunan Laporan Praktek Kerja Lapangan pada Sistem Manajemen Produk dan Ticketing Pada PT. Teknologi Aplikasi Sejahtera.

#### **BAB III ANALISIS KEBUTUHAN DAN PERANCANGAN**

Bab ini menjelaskan tentang pembahasan yang meliputi deskripsi umum perangkat lunak, analisis, dan desain rancangan Laporan Praktek Kerja Lapangan pada Sistem Manajemen Produk dan Ticketing Pada PT. Teknologi Aplikasi Sejahtera.

#### **BAB IV IMPLEMENTASI DAN PENGUJIAN**

Bab ini menjelaskan mengenai implementasi berdasarkan rancangan sistem dan pengujian dari sistem yang telah dibentuk, yaitu Laporan Praktek Kerja Lapangan pada Sistem Manajemen Produk dan Ticketing Pada PT. Teknologi Aplikasi Sejahtera.

#### **BAB V PENUTUP**

Bab ini membahas kesimpulan dari Praktek Kerja Lapangan yang sudah dilakukan dan saran penulis untuk pengembangan lebih lanjut mengenai sistem yang telah dibuat.

## **BAB II**

### **TINJAUAN PERUSAHAAN**

Bagian ini akan membahas terkait dengan informasi seputar PT. Teknologi Aplikasi Sejahtera yang berupa profil instansi, visi dan misi, serta struktur organisasinya.

#### **2.1. Profil Instansi**

PT Teknologi Aplikasi Sejahtera (TAS) merupakan perusahaan pengembang teknologi yang berkantor di Semarang, PT Teknologi Aplikasi Sejahtera berkomitmen untuk menjadi perusahaan teknologi informasi terintegrasi dengan support system yang unggul, inovatif dan terpercaya. PT Teknologi Aplikasi Sejahtera menyediakan layanan pembuatan web apps, mobile apps, hingga IoT (Internet Of Things). Informasi detail terkait dengan PT Teknologi Aplikasi Sejahtera dapat dilihat pada tabel 2.1 dibawah ini.

Tabel 2.1: Informasi PT Teknologi Aplikasi Sejahtera

|               |   |
|---------------|---|
| Nama Instansi | PT Teknologi Aplikasi Sejahtera   |
| Alamat Kantor | Jl. Plamongan Indah Blok E2 No. 17, Batursari, Kec. Mranggen, Kabupaten Demak 59567 |
| Telepon       | 0895-3271-75587   |
| Email         | teknosejahtera@gmail.com  |
| Website       | <a href="https://teknosejahtera.co.id/">https://teknosejahtera.co.id/</a>           |

#### **2.2. Visi**

PT. Teknologi Aplikasi Sejahtera memiliki 4 Visi, yaitu:

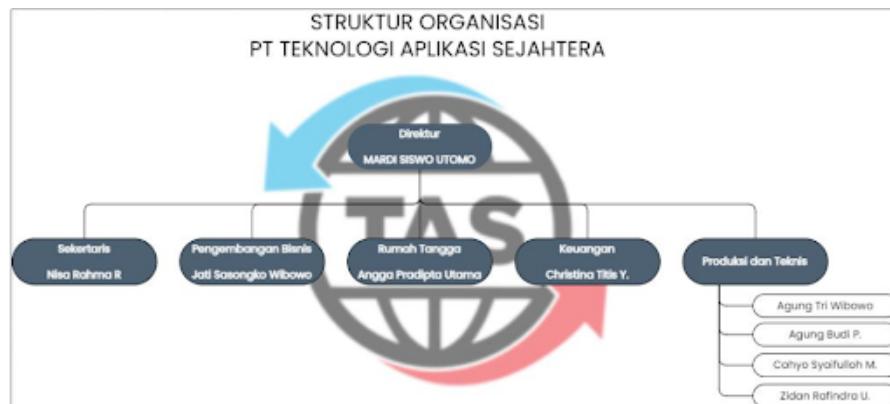
1. *Smart Innovation*
2. *Excellence Integration*
3. *Trust Integrity*
4. *Express Delivery Orientation*

#### **2.3. Misi**

PT Teknologi Sejahtera memiliki visi untuk menjadi perusahaan teknologi informasi terintegrasi yang terkemuka dengan support system yang unggul, inovatif dan terpercaya sehingga memberikan manfaat sebesar-besarnya bagi bangsa indonesia maupun pengguna secara luas dengan menjawab serta mempersiapkan kebutuhan

## 2.4. Struktur Organisasi

Pada gambar ?? di bawah ini merupakan struktur organisasi PT Teknologi Aplikasi Sejahtera. Dipimpin oleh Bapak Mardi Siswo Utomo selaku Direktur, dengan beberapa divisi di bawahnya, antara lain sekretaris, pengembangan bisnis, rumah tangga, keuangan, serta produksi dan teknis. Selama pelaksanaan praktek kerja lapangan, saya ditempatkan di divisi produksi dan teknis, di bawah bimbingan supervisor lapangan, Bapak Zidan Rafindra U. Fokus utama divisi ini adalah pada pengembangan sistem informasi.



Gambar 2.1: Struktur PT. Teknologi Aplikasi Sejahtera

## **BAB III**

### **LANDASAN TEORI**

#### **3.1 Sistem Informasi**

Sistem informasi (SI) merupakan penunjang penting dari proses bisnis sebuah organisasi yang memfasilitasi komunikasi dan koordinasi di antara berbagai area fungsional, dan memungkinkan pertukaran data serta akses data dengan mudah di seluruh proses bisnis ?. SI memainkan peran penting dalam tiga bidang:

##### **1. Menjalankan proses bisnis**

Sistem Informasi (SI) membantu organisasi menjalankan bisnis secara efisien dan efektif. SI biasanya tertanam dalam proses bisnis suatu organisasi dan memainkan peran penting dalam melaksanakan proses bisnis tersebut. Jika SI tidak berfungsi, proses bisnis akan terhambat atau tidak dapat dijalankan. SI membantu menjalankan bisnis dengan memberitahu pengguna kapan saatnya untuk menyelesaikan tugas dengan menyediakan data yang diperlukan untuk menyelesaikan tugas tersebut. Dalam beberapa kasus SI menyediakan cara untuk menyelesaikan tugas tersebut.

##### **2. Menangkap dan menyimpan data yang sudah diproses**

Proses bisnis menghasilkan data seperti tanggal, waktu, produk jumlah, jumlah, harga, dan alamat, serta siapa melakukan apa, kapan, dan di mana. Menangkap dan menyimpan data tersebut biasa disebut dengan transaksi data. Beberapa data dihasilkan dan secara otomatis ditangkap oleh SI. Data ini adalah data terkait siapa yang melakukan menyelesaikan suatu kegiatan, kapan, dan di mana. Data lain yang dihasilkan di luar SI dapat dimasukkan dengan berbagai cara, seperti entri manual hingga otomatis seperti automation tools atau SI lain yang terhubung dengan SI yang bersangkutan.

##### **3. Memantau kinerja proses bisnis**

Peran terakhir SI adalah untuk melakukan monitoring terhadap semua alur bisnis yang terjadi dalam sistem. SI dapat memberitahu pengguna seberapa efektif proses bisnis yang sedang dijalankan. Proses monitoring ini dapat dilakukan dengan mengevaluasi kegiatan bisnis yang ada.

#### **3.2 Aplikasi Berbasis Web**

Aplikasi berbasis web (disingkat webapp) adalah suatu aplikasi yang diakses menggunakan penjelajah web melalui suatu jaringan seperti Internet atau intranet. Webapp juga merupakan suatu aplikasi perangkat lunak komputer yang dikodekan

dalam bahasa yang didukung web browser (seperti Javascript, PHP, Python, Ruby, dll) dan bergantung pada aplikasi web browser tersebut untuk menampilkan aplikasi.

Aplikasi web menjadi populer karena kemudahan tersedianya aplikasi klien untuk mengaksesnya, penjelajah web, yang kadang disebut sebagai suatu thin client. Kemampuan untuk memperbarui dan memelihara aplikasi web tanpa harus mendistribusikan dan menginstalasi perangkat lunak pada kemungkinan ribuan komputer klien merupakan alasan kunci popularitasnya.

### **3.3 Unified Modeling Language (UML)**

UML (Unified Modelling Language) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Awal mulanya, UML diciptakan oleh Object Management Group dengan versi awal 1.0 pada bulan Januari 1997.

UML dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem perangkat lunak ?. UML diharapkan mampu mempermudah pengembangan perangkat lunak serta memenuhi semua kebutuhan pengguna dengan efektif, lengkap, dan tepat. Hal itu termasuk faktor-faktor scalability, robustness, security, dan sebagainya.

#### **3.3.1 Use Case Diagram**

Use Case Diagram adalah satu jenis dari diagram UML yang menggambarkan interaksi antara sistem dan aktor, Use Case dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya (Hengki, 2017). Use Case merupakan sesuatu yang mudah dipelajari. Langkah awal untuk melakukan pemodelan perlu adanya suatu diagram yang mampu menjelaskan aksi aktor dengan aksi dalam sistem itu sendiri, seperti yang terdapat pada Use Case.

#### **3.3.2 Activity Diagram**

Activity diagram adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau sistem), orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut (Hengki, 2017). Activity diagram adalah salah satu contoh diagram dari UML dalam pengembangan dari Use Case.

#### **3.3.3 Sequence Diagram**

Sequence diagram merupakan diagram yang menjelaskan interaksi objek yang berdasarkan urutan waktu, sequence diagram juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada use case diagram (Hengki, 2017). Sequence dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu,

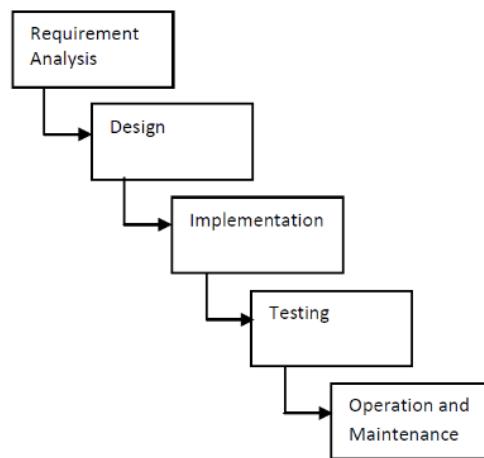
seperti yang tertera pada Use Case diagram.

### 3.4 Metode Perangkat Lunak

Metode pengembangan perangkat lunak yang digunakan oleh penulis yaitu OOAD (Object Oriented Analysis and Design). OOAD adalah metode analisis yang memeriksa requirement dari sudut pandang kelas-kelas dan objek yang ditemui dalam ruang lingkup permasalahan dan mengarah pada arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

### 3.5 Model Proses Waterfall

Model waterfall adalah model pengembangan perangkat lunak statis yang memiliki pendekatan secara linier, berurutan, dan wajibkan penggunaanya untuk menyelesaikan satu aktivitas sebelum lanjut ke aktivitas lainnya (Adenowo & Adenowo, 2013). Fowler (2004) memecah model waterfall menjadi beberapa bagian berdasarkan aktivitas yang dilakukan: analisis kebutuhan, desain, coding, dan pengujian. Sedangkan Pressman (2005) membagi model waterfall menjadi: communication (inisiasi proyek dan pengumpulan persyaratan), planning (perkiraan, penjadwalan, dan tracking), modeling (analisis dan desain), construction (coding dan testing), dan deployment (delivery, support, dan feedback). Terakhir ada Shari L. Pfleeger & Joanne M. Atlee (2006) yang membagi model waterfall menjadi: requirement analysis, system design, coding, unit and integration testing, system testing, acceptance testing, dan operation and maintenance. Dari beberapa pendapat ahli tersebut, dapat disimpulkan bahwa model waterfall dapat dibagi menjadi lima tahap: requirement analysis, design, implementation, testing, dan operation and maintenance.



Gambar 3.1: Tahapan dalam model waterfall (Pfleeger Atlee, 2006)

1. Requirement analysis. Pada tahap ini pihak pembuat perangkat lunak, yaitu

developer, melakukan konsultasi dengan pihak pengguna perangkat lunak, yaitu klien. Dalam konsultasi ini, kedua belah pihak akan mendiskusikan tentang fitur, constraint, dan tujuan dari dibuatnya perangkat lunak tersebut. Hasil dari diskusi tersebut kemudian akan dipaparkan secara rinci dalam sebuah dokumen yang berguna sebagai spesifikasi perangkat lunak. Tahap ini mencakup proses identifikasi dan penjelasan

2. Design. Di tahap ini terjadi pengalokasian hardware dan software yang akan digunakan dengan mendesain keseluruhan arsitektur perangkat lunak sesuai dengan spesifikasi yang telah dibuat. Proses desain mencakup pengidentifikasi dan pendeskripsian abstraksi fundamental dari perangkat lunak yang dibuat.
3. Implementation. Tahap ini merupakan tahap yang pelaksanaannya memerlukan resource paling banyak. Karena pada tahap ini terjadi realisasi desain perangkat lunak menjadi sekumpulan program atau program unit (contohnya seperti front end dan back end). Seluruh program unit yang telah dibuat kemudian diintegrasikan menjadi sebuah perangkat lunak yang utuh.
4. Testing. Pengetesan dilakukan agar perangkat lunak dapat dipastikan memenuhi persyaratan yang sudah didesain sebelumnya. Setelah perangkat lunak sudah dipastikan berjalan sesuai dengan yang diinginkan, perangkat lunak kemudian akan diserahkan kepada klien.
5. Operation and Maintenance. Biasanya (walaupun belum tentu), tahap ini adalah tahap terpanjang dari keseluruhan proses pengembangan perangkat lunak. Setelah sistem diinstal dan dapat digunakan oleh klien, pihak developer akan mulai melakukan pemantauan. Hal ini dilakukan agar ketika klien menemukan kesalahan baru yang tidak ditemukan pada tahap testing, pihak developer dapat langsung merilis pembaruan perangkat lunak yang memperbaiki kesalahan tersebut. Selain itu jika sewaktu-waktu terdapat requirement baru yang dimana perangkat lunak yang dibuat bergantung pada requirement tersebut, pihak developer juga dapat merilis pembaharuan perangkat lunak kepada klien.

### **3.6 Bahasa pemrograman**

Bahasa pemrograman yang digunakan dalam sistem manajemen produk dan ticketing bagian modul ticketing di PT Teknologi Aplikasi Sejahtera antara lain:

#### **3.8.1. PHP**

Hypertext Preprocessor (PHP) adalah bahasa server-side scripting yang menyatu dengan HTML untuk membuat halaman Website yang dinamis. PHP banyak dipakai untuk pemrograman situs Website dinamis karena PHP merupakan

server-side scripting, maka sintaks dan perintah-perintah PHP akan dieksekusi oleh server kemudian hasilnya dikirim ke browser dalam format HTML.

### 3.8.2. CSS

Cascading Style Sheets (CSS) adalah bahasa stylesheet yang digunakan untuk menggambarkan tampilan sebuah dokumen yang ditulis dalam HTML atau XML (termasuk dialek XML seperti SVG, MathML, atau XHTML). CSS menjelaskan bagaimana elemen-elemen seharusnya ditampilkan di layar, di kertas, dalam suara, atau di media lainnya (Mozilla, 2024).

### 3.8.3. JavaScript

JavaScript (JS) adalah bahasa pemrograman yang ringan yang diinterpretasi dengan fitur first-class functions. JavaScript merupakan bahasa pemrograman berbasis prototipe, multi-paradigma, single-threaded, dinamis, yang mendukung gaya pemrograman berorientasi objek, imperatif, dan deklaratif (Mozilla, 2024).

## 3.7 RESTful API

RESTful API adalah API berbasis website yang menggunakan teknologi REST dan menggunakan format JSON (JavaScript Object Notation), yaitu sebuah format pertukaran data yang menggunakan metode http GET, POST, PUT, dan DELETE untuk menerima, membuat, memperbarui dan menghapus resource (Ramadan Handaga, 2019). API (Application Programming Interface) sendiri merupakan cara untuk berinteraksi secara terprogram dengan komponen perangkat lunak atau sumber daya yang terpisah. Adanya API membuat produk atau layanan dapat terhubung dengan produk dan layanan lainnya tanpa harus tahu bagaimana cara penerapannya. Dalam pengimplementasiannya RESTful API memiliki 2 sisi yakni front end dan back end. Penjelasan masing masing bagian dapat dilihat sebagai berikut

### 3.8.1. Front end

### 3.8.2. Back end

**3.8 Laravel** Laravel adalah framework web berbasis PHP yang free dan open-source untuk membangun aplikasi web berkualitas tinggi. Laravel diciptakan oleh Taylor Otwell dan ditujukan untuk pengembangan aplikasi web yang mengikuti pola arsitektur model–view–controller (MVC) dan berbasis pada framework PHP lain, Symfony. Beberapa fitur dari Laravel termasuk sistem packaging yang modular dengan dependency manager. Laravel menyediakan berbagai cara untuk mengakses basis data relasional dari menggunakan SQL query secara langsung atau menggunakan wrapper dengan syntactic sugar pada library Eloquent. Laravel juga menyediakan berbagai script yang sangat membantu tugas tugas repetitif dan deployment.

### **3.9 ReactJS**

ReactJS adalah library JavaScript front-end yang free dan open-source untuk membangun antarmuka pengguna berbasis komponen. ReactJS dipelihara oleh Meta dan komunitas pengembang (Paul, 2014).

React dapat digunakan untuk mengembangkan single-page application (SPA), aplikasi mobile, atau server-side rendered application dengan framework seperti Next.js. Karena React hanya peduli dengan antarmuka pengguna dan merender komponen ke DOM, aplikasi React mengandalkan library untuk routing dan fungsionalitas sisi client lainnya. Keunggulan utama dari React adalah bahwa React hanya merender kembali bagian-bagian halaman yang telah berubah, menghindari merender ulang yang tidak perlu dari elemen-elemen DOM yang tidak berubah.

### **3.10 PostgreSQL**

PostgreSQL yang juga dikenal sebagai Postgres adalah sistem manajemen basis data relasional (RDBMS) yang free dan open-source yang menekankan pada kemampuan extensibility dan kompatibilitas dengan SQL. PostgreSQL memiliki fitur transaksi dengan properti atomicity, consistency, isolation, durability (ACID). Postgres juga memiliki fitur untuk memperbarui view secara otomatis, triggers, stored procedures. Ini didukung pada semua sistem operasi utama, termasuk Linux, FreeBSD, OpenBSD, macOS, dan Windows.

### **3.11 Pengujian Perangkat Lunak**

Pengujian perangkat lunak adalah proses penting dalam pengembangan perangkat lunak yang bertujuan untuk mengidentifikasi kecacatan dalam aplikasi atau sistem yang sedang dikembangkan. Menurut Sommerville, (2011), pengujian perangkat lunak tidak hanya mencakup penemuan bug atau kesalahan, tetapi juga verifikasi bahwa perangkat lunak memenuhi persyaratan yang ditentukan dan bekerja seperti yang diharapkan dalam semua kondisi yang diperkirakan pengguna. Sommerville menjelaskan bahwa pengujian perangkat lunak harus sistematis dan dapat diulang. Hal ini mencakup beberapa tingkat pengujian, yaitu:

#### **1. Pengujian Unit**

Fokus pada masing-masing komponen atau unit perangkat lunak untuk memastikan bahwa setiap bagian bekerja secara benar secara individual. Pengujian ini sering kali dilakukan oleh pengembang yang menggunakan framework pengujian yang ada untuk mengisolasi setiap bagian dan memeriksa fungsionalitasnya.

#### **2. Pengujian Integrasi**

Menilai kombinasi unit-unit perangkat lunak dan modul untuk mendeteksi kegagalan dalam interaksi antara bagian-bagian yang terintegrasi. Pengujian ini penting untuk menangani masalah yang muncul dari interaksi antar modul yang tidak diperkirakan selama pengujian unit.

3. Pengujian Sistem Memeriksa sistem perangkat lunak lengkap untuk memastikan bahwa ia memenuhi semua persyaratan dan spesifikasi. Pengujian sistem mencakup evaluasi fitur, keamanan, dan kinerja sistem dalam kondisi yang mirip dengan apa yang akan dihadapi di dunia nyata.
4. Pengujian penerimaan

Pengujian penerimaan dilakukan oleh pengguna atau klien untuk memverifikasi bahwa sistem sesuai dengan kebutuhan bisnis dan siap untuk dioperasikan. Pengujian ini seringkali merupakan pengujian final sebelum perangkat lunak dinyatakan siap untuk diproduksi dan diluncurkan.

## **BAB IV**

### **ANALISIS DAN PERANCANGAN**

Dalam pengembangan sistem ticketing yang memiliki lingkup yang relatif kecil seperti yang modul ticket ini cocok untuk menggunakan metode waterfall adalah karena karakteristiknya yang sesuai dengan lingkup proyek yang terbatas. Metode waterfall terkenal karena pendekatannya yang linear dan terstruktur, di mana setiap tahap pengembangan dilakukan secara berurutan, mulai dari analisis kebutuhan hingga pengiriman produk akhir. Dalam kasus ticketing yang memiliki lingkup yang relatif kecil, pendekatan ini memberikan kejelasan dan keteraturan dalam proses pengembangan. Dengan demikian, pengembangan dapat dengan mudah memahami langkah-langkah yang diperlukan dan fokus pada tujuan yang ditetapkan.

#### **4.1. Requirement Elicitation**

Analisis kebutuhan digunakan untuk mengidentifikasi kebutuhan sistem yang akan dibangun. Pada bagian ini, terdapat penjelasan mengenai deskripsi umum sistem, kebutuhan fungsional, kebutuhan non-fungsional, use case diagram, activity diagram, dan sequence diagram

##### **4.1.1. Deskripsi Sistem**

Modul Ticket dalam Sistem Informasi Manajemen Produk dan Ticketing di PT TAS adalah sebuah sistem yang dirancang untuk mengelola data bisnis dan transaksi terkait manajemen ticketing. Sistem ini memungkinkan berbagai aktor untuk berinteraksi dalam proses pembuatan, verifikasi, penggerjaan, dan penyelesaian ticket. Adapun beberapa role yang terlibat dalam sistem ini meliputi internal user sebagai product manager, internal user sebagai developer, dan external user sebagai PIC (Person in Charge) pembuatan ticket. Setiap ticket dalam sistem ini memiliki beberapa status, yaitu pending verification, in progress, done, closed, dan rejected. Berikut adalah alur transaksi yang terjadi pada setiap tahap ticket:

###### **1. Pending verification**

Status ini merupakan status ketika ticket dibuat oleh external user dan menunggu verifikasi dari product manager apakah ticket ini valid atau tidak. Jika ticket ini valid, ticket akan dialokasikan developer yang akan mengerjakannya dan ticket akan berubah statusnya menjadi in progress. Jika ticket tidak valid, status ticket akan berubah menjadi rejected.

## 2. In progress

Ticket saat status ini sedang dikerjakan para developer, tiap developer yang mengerjakan ticket ini juga memiliki status penggerjaan yang dapat dilihat oleh user lain. Ketika semua developer sudah ok, product manager dapat mengecek apakah penggerjaan ticket sudah sesuai atau belum. Jika sesuai, ticket akan masuk ke tahap done untuk divalidasi pembuat ticket. Jika tidak sesuai, status penggerjaan developer akan direset dan status ticket tidak berubah.

## 3. Done

Ticket pada status ini akan dicek oleh external user apakah ticket yang sudah diajukan sudah benar benar sesuai. Jika terdapat ketidaksesuaian, external user dapat bisa mengajukan revisi dan ticket akan kembali menjadi in progress. Jika sesuai ticket akan menjadi closed.

## 4. Closed

Ticket dengan status ini berarti sudah selesai dan akan masuk ke dalam arsip.

## 5. Rejected

Ticket dengan status ini tidak sampai ke tahap penggerjaan dan masuk ke dalam arsip.

Dari alur transaksi yang sudah diidentifikasi, sistem memiliki tiga aktor yakni external user sebagai PIC pembuatan ticket, internal user sebagai product manager, dan internal user sebagai developer. Ketiga aktor ini memiliki peran dan tanggung jawab yang berbeda dalam pengelolaan ticket. Berikut adalah deskripsi singkat mengenai peran masing-masing aktor beserta aksi yang dapat mereka lakukan.

### 1. External User / PIC

External user merupakan pegawai dari luar yang bertanggung jawab untuk membuat ticket yang akan diproses oleh tim internal. External user berperan sebagai pihak yang mengajukan ticket terhadap tim internal untuk diperbaiki. External user juga bertanggung jawab untuk memeriksa hasil penggerjaan ticket dan memastikan bahwa kebutuhan mereka telah terpenuhi. Berikut merupakan aksi yang dapat dilakukan oleh seorang external user pada modul ticketing:

- A. Membuat ticket baru.
- B. Memeriksa ticket yang sudah diajukan.
- C. Memvalidasi hasil penggerjaan ticket pada status done.

- D. Mengajukan revisi jika hasil penggerjaan tidak sesuai.
- E. Menutup ticket jika penggerjaan sudah sesuai.
- F. Berkomunikasi dengan product manager

### 2. Internal User (Product Manager)

Product manager berperan sebagai penghubung antara external user dan developer. Product manager bertanggung jawab untuk memverifikasi validitas ticket yang diajukan oleh external user dan mengalokasikan ticket yang valid kepada developer. Product manager juga memastikan bahwa hasil penggerjaan ticket oleh developer sesuai dengan kebutuhan yang diajukan oleh external user sebelum mengubah status ticket. Berikut merupakan aksi yang dapat dilakukan oleh seorang product manager pada modul ticketing:

- A. Menerima dan memverifikasi ticket baru.
- B. Memvalidasi ticket.
- C. Mengalokasikan tenaga kerja ke ticket yang valid.
- D. Memeriksa hasil penggerjaan developer.
- E. Mengubah status ticket berdasarkan hasil verifikasi dan pengecekan.
- F. Berkomunikasi dengan external user

### 3. Internal User (Developer)

Developer bertanggung jawab untuk mengerjakan ticket yang telah dialokasikan oleh product manager. Developer bekerja untuk menyelesaikan masalah atau kebutuhan yang tertera pada ticket. Developer juga mengupdate status penggerjaan ticket sehingga dapat dipantau oleh product manager dan external user. Berikut merupakan aksi yang dapat dilakukan oleh seorang developer pada modul ticketing:

- A. Menerima tugas ticket dari product manager.
- B. Mengerjakan ticket sesuai dengan deskripsi dan kebutuhan.
- C. Menyelesaikan penggerjaan dan menginformasikan product manager untuk verifikasi.

#### **4.1.2. Kebutuhan Fungsional**

Sistem Informasi Manajemen Produk dan Ticketing bagian modul ticketing di PT. Teknologi Aplikasi Sejahtera memiliki beberapa kebutuhan fungsional yang harus dipenuhi untuk memastikan kinerja yang optimal dan mendukung pengguna dalam menjalankan tugas-tugas mereka dengan efisien. Kebutuhan fungsional ini dirancang untuk memenuhi proses bisnis dan alur kerja yang terlibat dalam manajemen ticketing. Daftar kebutuhan fungsional dapat dilihat pada tabel ??.

Tabel 4.1: Kebutuhan fungsional

| SRS ID         | Deskripsi   |
|----------------|---|
| SRS-TKT-PM-01  | Sistem dapat menampilkan list tiket yang dibuat oleh pihak eksternal berdasarkan produk yang menjadi tanggung jawabnya untuk Internal user (PM) |
| SRS-TKT-PM-02  | Sistem dapat memfilter list tiket berdasarkan status tiket untuk Internal user (PM)   |
| SRS-TKT-PM-03  | Sistem dapat memberikan detail tiket yang sudah ada untuk Internal user (PM)  |
| SRS-TKT-PM-04  | Sistem dapat mem-verify tiket yang dibuat oleh pihak eksternal dan mengalokasikan developer yang bertanggung jawab untuk Internal user (PM)     |
| SRS-TKT-PM-05  | Sistem dapat menolak tiket yang dibuat oleh pihak eksternal dan memberikan alasan menolak untuk Internal user (PM)                              |
| SRS-TKT-PM-06  | Sistem dapat mengusulkan revisi tiket yang sudah dikerjakan oleh developer dan memberikan alasan revisi untuk Internal user (PM)                |
| SRS-TKT-PM-07  | Sistem dapat berkomunikasi dengan pihak eksternal (pembuat tiket) melalui whatsapp untuk Internal user (PM)                                     |
| SRS-TKT-PM-08  | Sistem dapat mengubah status tiket menjadi done untuk Internal user (PM)  |
| SRS-TKT-DEV-01 | Sistem dapat memberikan list data tiket yang diassign oleh PM berdasarkan produk yang menjadi tanggung jawabnya untuk Internal user (DEV)       |
| SRS-TKT-DEV-02 | Sistem dapat memfilter list tiket berdasarkan status tiket untuk Internal user (DEV)  |
| SRS-TKT-DEV-03 | Sistem dapat memberikan detail tiket yang sudah ada untuk Internal user (DEV)   |
| SRS-TKT-DEV-04 | Sistem dapat mengganti status penggerjaan tiket menjadi done dari tiket yang diassign oleh PM untuk Internal user (DEV)                         |

| SRS ID         | Deskripsi   |
|----------------|---|
| SRS-TKT-PIC-01 | Sistem dapat menampilkan list tiket yang dibuat oleh dirinya berdasarkan product yang menjadi tanggung jawabnya untuk External user       |
| SRS-TKT-PIC-02 | Sistem dapat memfilter list tiket berdasarkan status tiket untuk External user  |
| SRS-TKT-PIC-03 | Sistem dapat membuat tiket baru berdasarkan produk yang menjadi tanggung jawabnya untuk External user                                     |
| SRS-TKT-PIC-04 | Sistem dapat menampilkan detail tiket yang telah dibuat untuk External user   |
| SRS-TKT-PIC-05 | Sistem dapat mengubah status tiket yang sudah selesai menjadi closed untuk External user  |
| SRS-TKT-PIC-06 | Sistem dapat mengubah status tiket yang sudah selesai kembali menjadi in progress jika masih ada yang belum terpenuhi untuk External user |
| SRS-TKT-PIC-07 | Sistem dapat berkomunikasi dengan pihak internal (PM) melalui WA untuk External user  |

#### 4.1.3. Kebutuhan Non-Fungsional

Setelah mengidentifikasi kebutuhan fungsional, langkah selanjutnya adalah mengidentifikasi kebutuhan non-fungsional. Kebutuhan non-fungsional ini berfungsi sebagai elemen pendukung yang memastikan bahwa sistem dapat berjalan dengan efektif dan efisien dalam memenuhi kebutuhan fungsionalnya. Daftar kebutuhan non-fungsional untuk sistem ini disajikan dalam tabel ??.

Tabel 4.2: Kebutuhan Non-Fungsional

| No | Deskripsi  |
|----|--|
| 1  | Sistem harus kompatibel dan dapat dijalankan secara optimal di web browser Google Chrome |
| 2  | Sistem harus memiliki response time yang tidak lebih dari 3 detik                        |

#### 4.1.4. Daftar Use Case

Sistem Informasi Manajemen Produk dan Ticketing bagian ticketing di

PT. Teknologi Aplikasi Sejahtera melibatkan beberapa use case yang mencakup berbagai interaksi antara pengguna dengan sistem. Use case ini menggambarkan berbagai skenario di mana aktor-aktor dalam sistem berinteraksi untuk mencapai tujuan tertentu. Deskripsi mendetail mengenai setiap use case dijelaskan pada tabel ??.

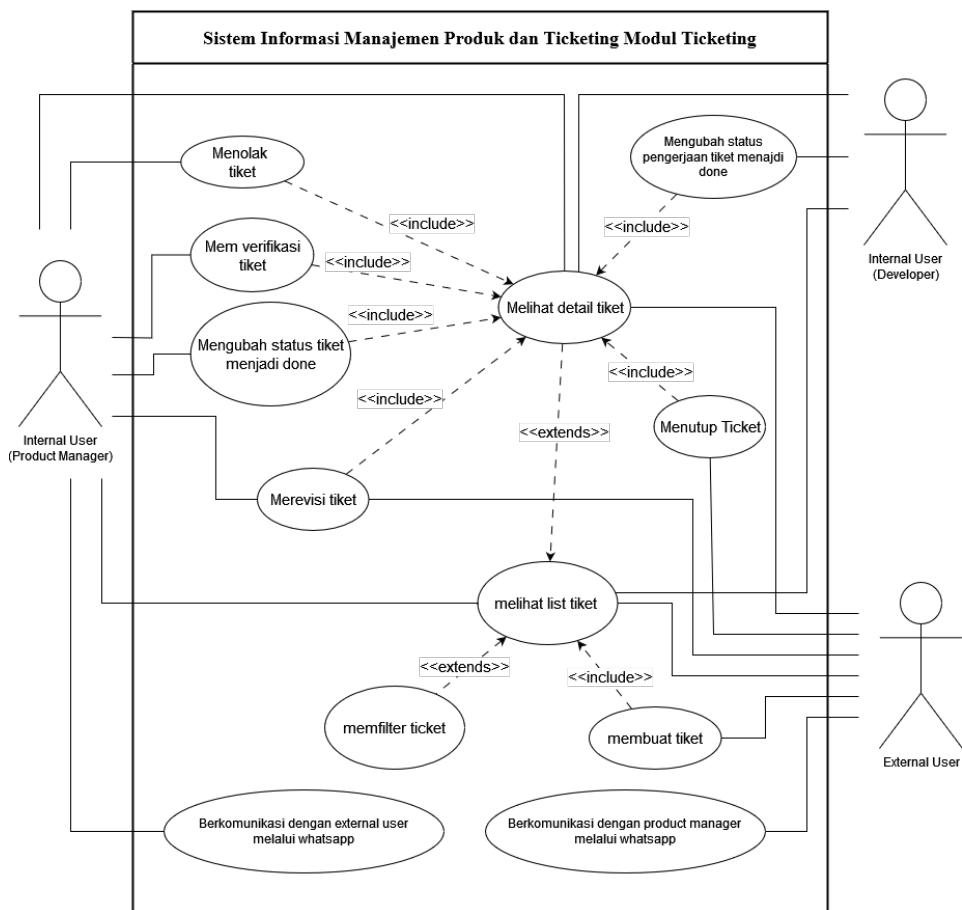
Tabel 4.3: Daftar Use Case

| Aktor                           | Deskripsi  |
|---------------------------------|--|
| Internal User (Product Manager) | Dapat melihat list tiket yang dibuat oleh pihak eksternal berdasarkan product yang menjadi tanggung jawabnya |
|                                 | Dapat memfilter list tiket berdasarkan status tiket  |
|                                 | Dapat melihat detail tiket yang sudah ada  |
|                                 | Dapat mem-verify tiket yang dibuat oleh pihak eksternal dan mengalokasikan developer yang bertanggung jawab  |
|                                 | Dapat menolak tiket yang dibuat oleh pihak eksternal dan memberikan alasan menolak                           |
|                                 | Dapat mengusulkan revisi tiket yang sudah dikerjakan oleh developer dan memberikan alasan revisi             |
|                                 | Dapat berkomunikasi dengan pihak eksternal (pembuat tiket) melalui whatsapp                                  |
|                                 | Dapat mengubah status tiket menjadi done   |
| Internal User (Developer)       | Dapat melihat list data tiket yang diassign oleh PM berdasarkan produk yang menjadi tanggung jawabnya        |
|                                 | Dapat memfilter list tiket berdasarkan status tiket  |
|                                 | Dapat melihat detail tiket yang sudah ada  |
|                                 | Dapat mengganti status penggerjaan tiket menjadi done dari tiket yang diassign oleh PM                       |
| External User (PIC)             | Dapat melihat list tiket yang dibuat oleh dirinya berdasarkan product yang menjadi tanggung jawabnya         |
|                                 | Dapat memfilter list tiket berdasarkan status tiket  |
|                                 | Dapat membuat tiket baru berdasarkan produk yang menjadi tanggung jawabnya                                   |

| Aktor | Deskripsi   |
|-------|---|
|       | Dapat melihat detail tiket yang telah dibuat                  |
|       | Dapat mengubah status tiket yang sudah selesai menjadi closed |
|       | Dapat berkomunikasi dengan pihak internal (PM) melalui WA     |

#### 4.1.5. Use Case Diagram

Dari tabel yang telah dibuat pada daftar use case, kita dapat melihat hubungan interaksi antara aktor-aktor dalam sistem dengan menggunakan use case diagram. Berikut merupakan use case diagram yang menggambarkan interaksi tersebut yang dapat dilihat pada gambar ??



Gambar 4.1: Use Case Diagram Ticketing

## 4.2. Analysis

### 4.2.1. Activity Diagram

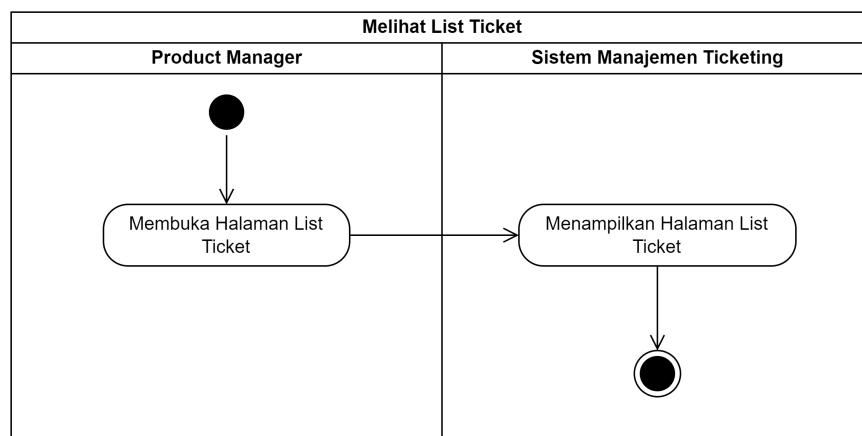
Berdasarkan use case yang telah ditentukan, berikut adalah activity diagram untuk Sistem Informasi Manajemen Produk dan Ticketing bagian ticketing di PT. Teknologi Aplikasi Sejahtera

#### 4.2.1.1. Activity Diagram Internal User (Product Manager)

##### 1. Activity Diagram Melihat List Ticket

ID Activity : AD-PM-01

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk melihat list ticket oleh seorang internal user dengan role product manager yang ditunjukkan pada gambar ??.

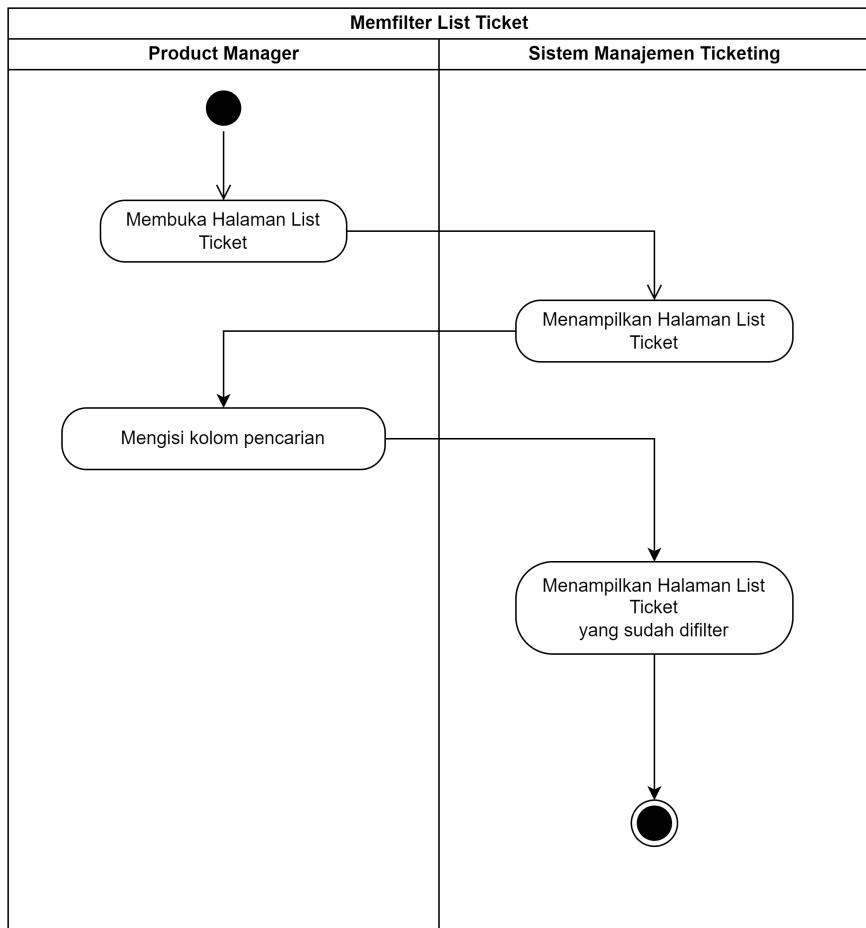


Gambar 4.2: Activity Diagram Melihat List Ticket (PM)

##### 2. Activity Diagram Memfilter Ticket

ID Activity : AD-PM-02

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk memfilter ticket oleh seorang internal user dengan role product manager yang ditunjukkan pada gambar ??

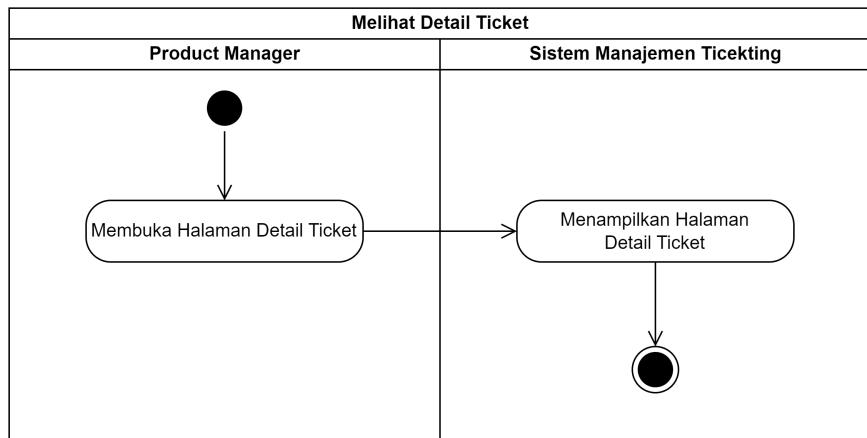


Gambar 4.3: Activity Diagram Memfilter List Ticket (PM)

### 3. Activity Diagram Melihat Detail Ticket

ID Activity : AD-PM-03

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk melihat detail ticket oleh seorang internal user dengan role product manager yang ditunjukkan pada gambar 4.4.

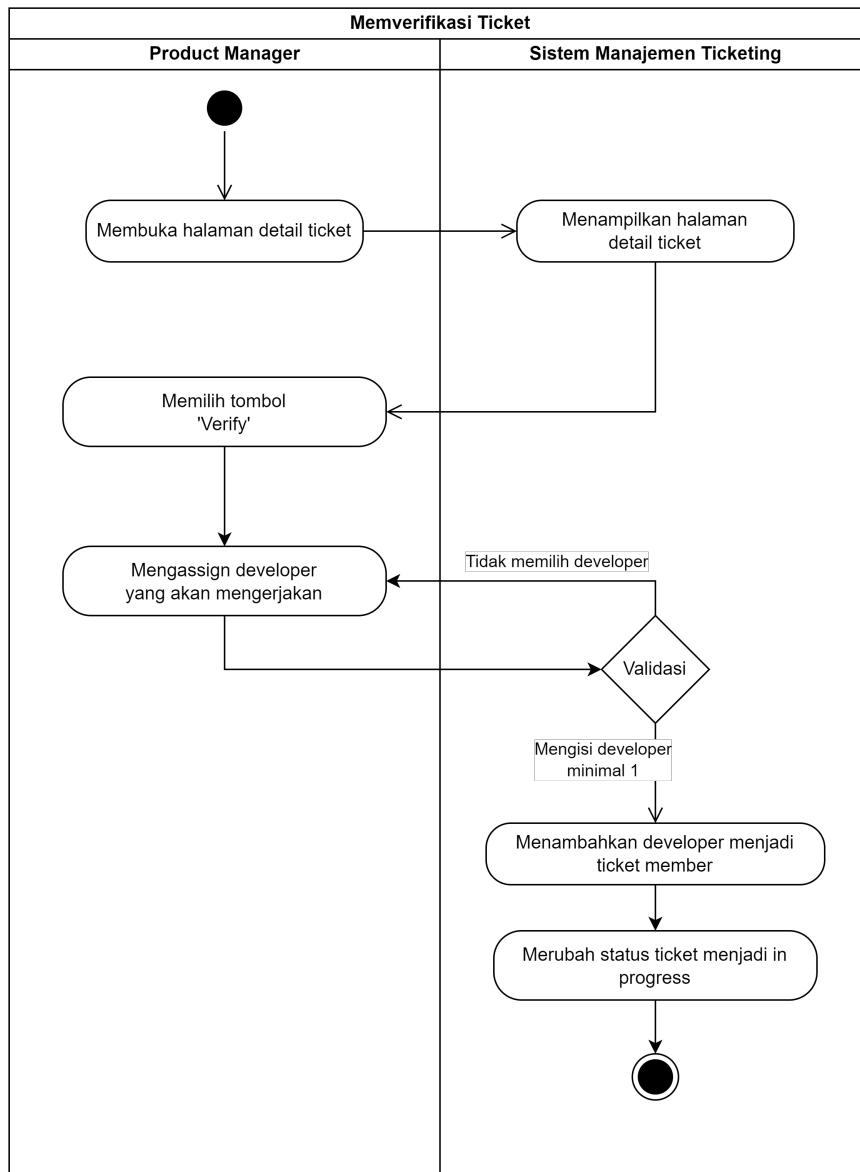


Gambar 4.4: Activity Diagram Melihat Detail Ticket (PM)

#### 4. Activity Diagram Memverifikasi Ticket

ID Activity : AD-PM-04

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk memverifikasi ticket yang dibuat external user oleh seorang internal user dengan role product manager yang ditunjukkan pada gambar 4.5.

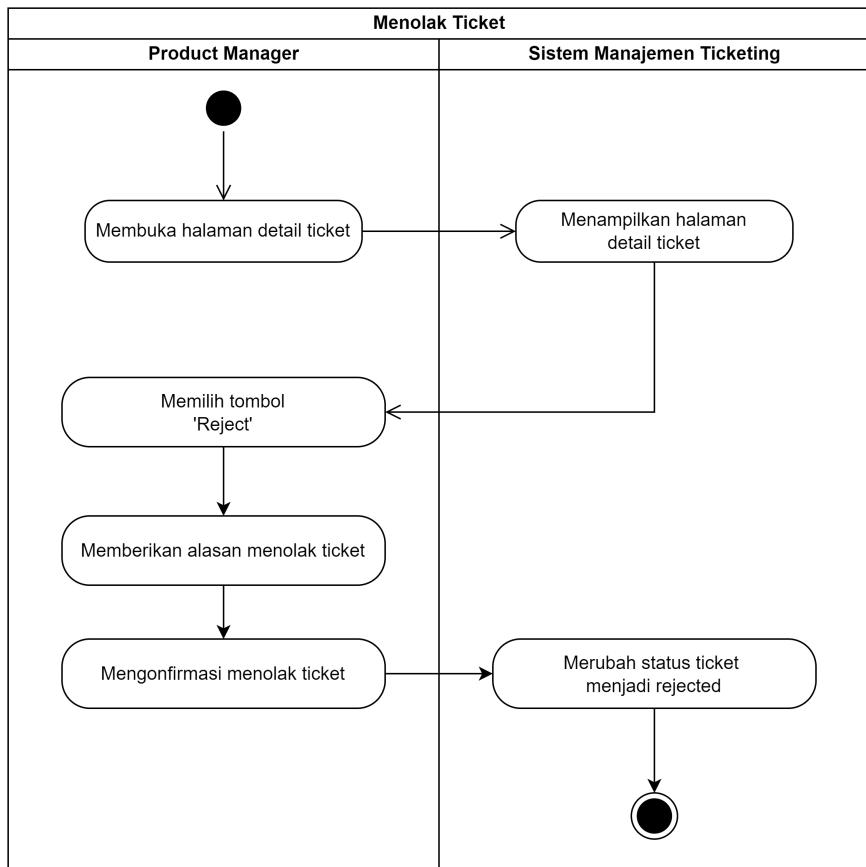


Gambar 4.5: Activity Diagram Memverifikasi Ticket (PM)

## 5. Activity Diagram Menolak Ticket

ID Activity : AD-PM-05

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk menolak ticket yang dibuat external user oleh seorang internal user dengan role product manager yang ditunjukkan pada gambar 4.6.

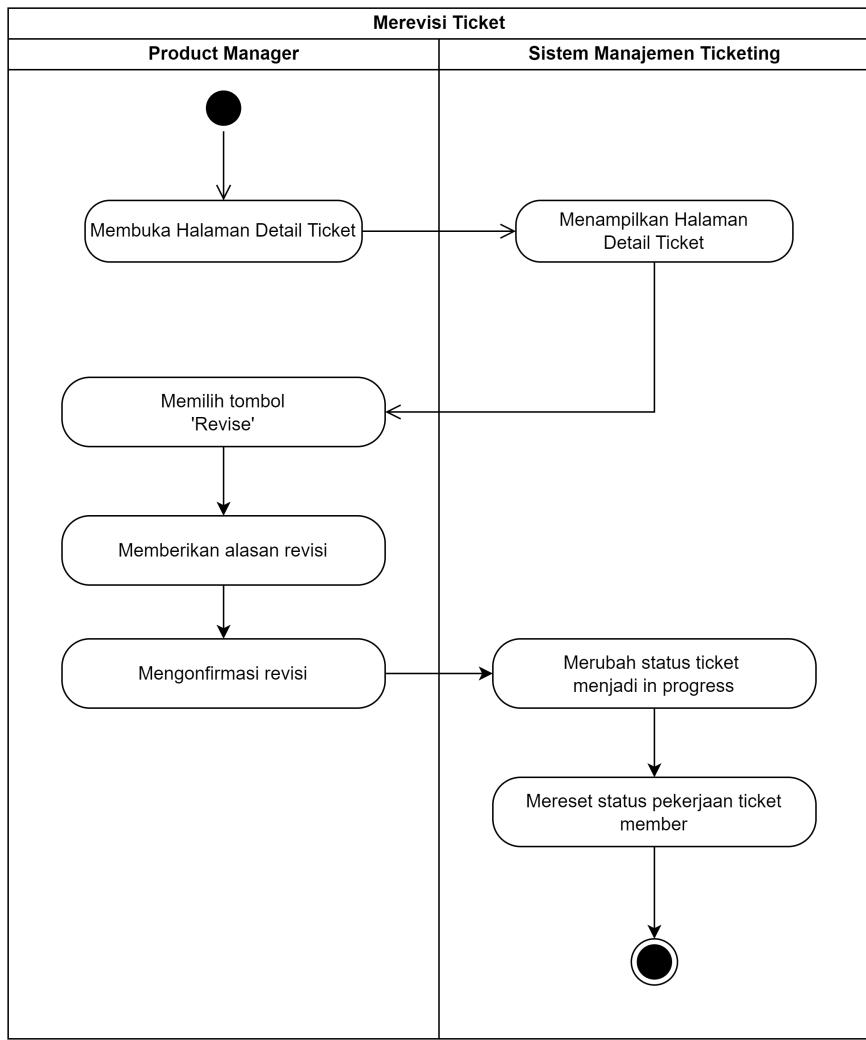


Gambar 4.6: Activity Diagram Menolak Ticket (PM)

## 6. Activity Diagram Merevisi Ticket

ID Activity : AD-PM-06

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk merevisi ticket yang telah dikerjakan developer oleh seorang internal user dengan role product manager yang ditunjukkan pada gambar 4.7.

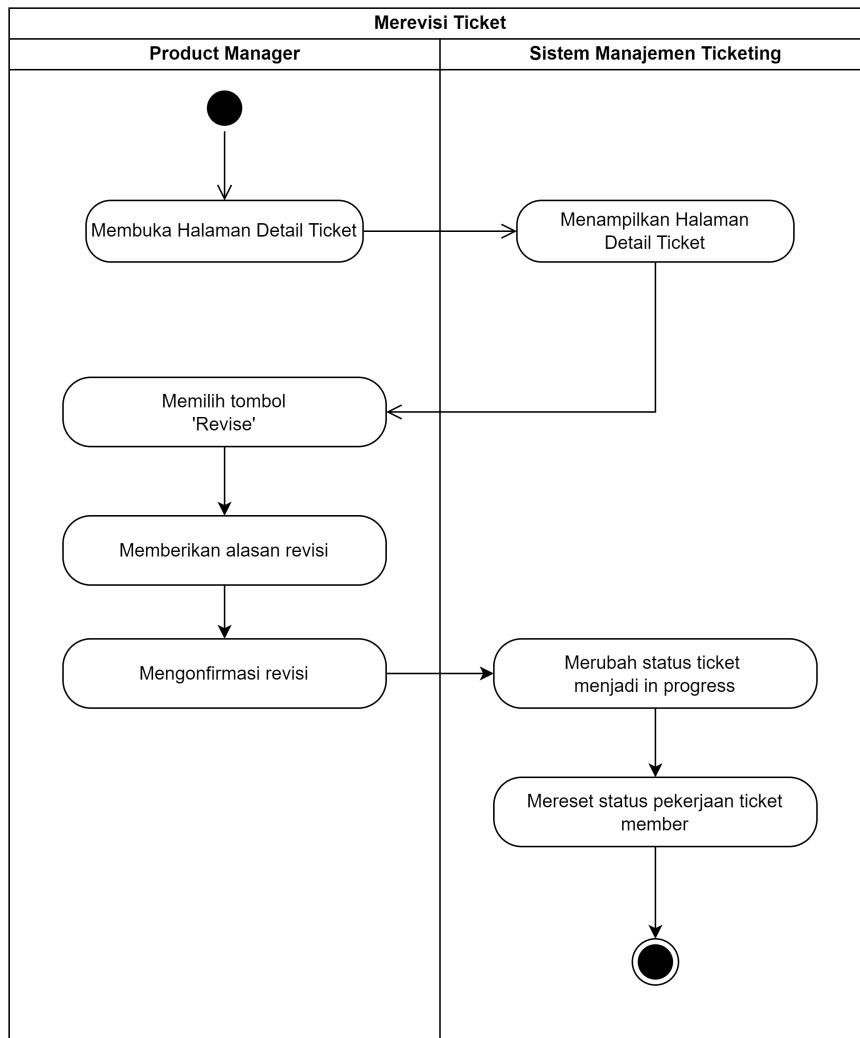


Gambar 4.7: Activity Diagram Merevisi Ticket (PM)

#### 7. Activity Diagram Berkomunikasi dengan external user

ID Activity : AD-PM-07

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk berkomunikasi dengan external user oleh seorang internal user dengan role product manager yang ditunjukkan pada gambar 4.8.

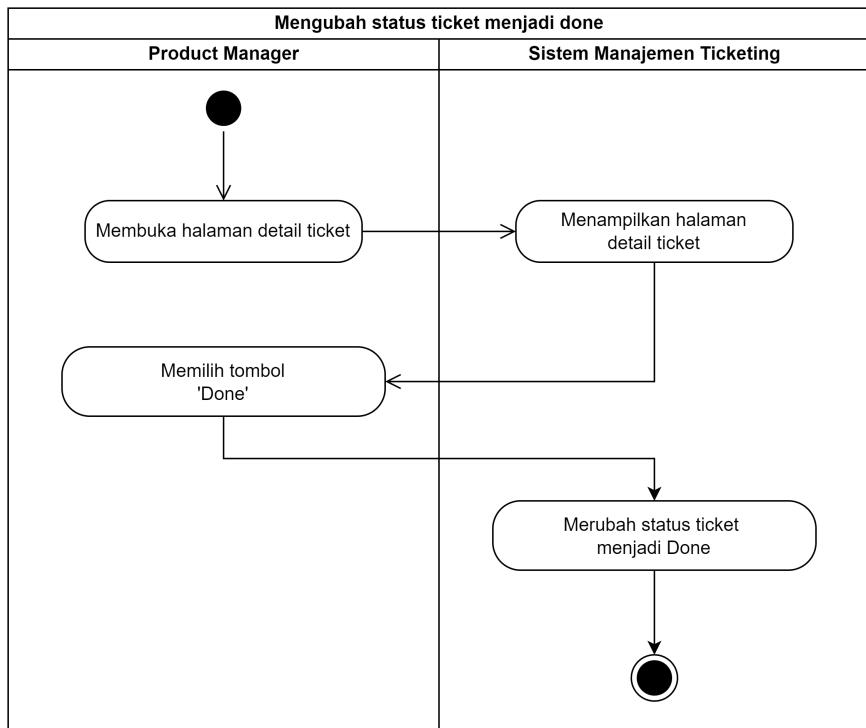


Gambar 4.8: Activity Diagram Berkomunikasi Dengan PM Melalui Whatsapp (PM)

#### 8. Activity Diagram Mengubah Status Ticket Menjadi Done

ID Activity : AD-PM-08

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk mengubah status ticket menjadi done oleh internal user dengan role product manager yang ditunjukkan pada gambar 4.9.



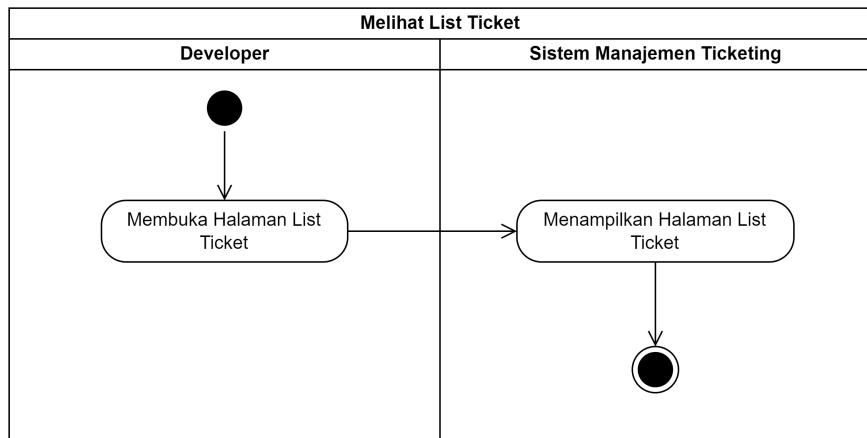
Gambar 4.9: Activity Diagram Mengubah Status Ticket Menjadi Done (PM)

#### **4.2.1.2. Activity Diagram Internal User (Developer)**

##### 1. Activity Diagram Melihat List Ticket

ID Activity : AD-DEV-01

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk melihat list ticket oleh internal user dengan role developer yang ditunjukkan pada gambar ??.

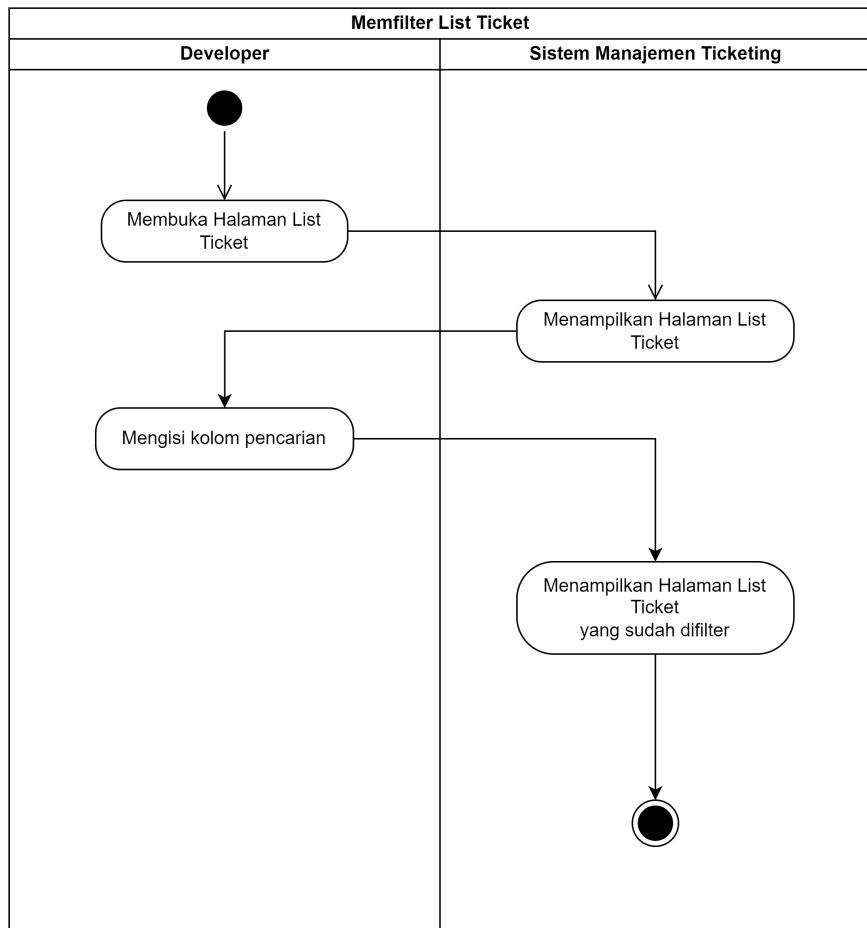


Gambar 4.10: Activity Diagram Melihat List Ticket (DEV)

## 2. Activity Diagram Memfilter Ticket

ID Activity : AD-DEV-02

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk memfilter list ticket oleh internal user dengan role developer yang ditunjukkan pada gambar ??.

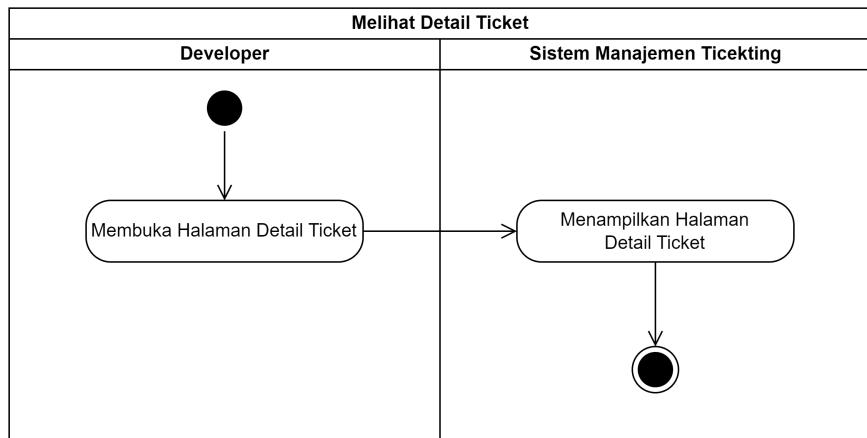


Gambar 4.11: Activity Diagram Memfilter List Ticket (DEV)

### 3. Activity Diagram Melihat Detail Ticket

ID Activity : AD-DEV-03

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk melihat detail ticket oleh internal user dengan role developer yang ditunjukkan pada gambar ??.

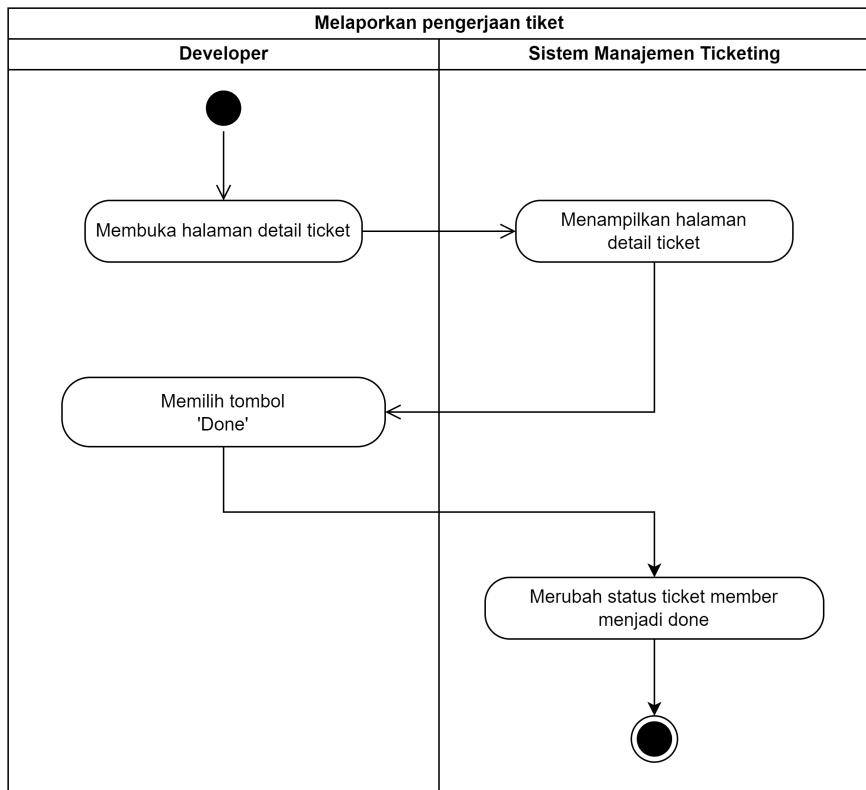


Gambar 4.12: Activity Diagram Melihat Detail Ticket (DEV)

#### 4. Activity Diagram Melaporkan Pekerjaan Ticket

ID Activity : AD-DEV-04

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk melaporkan penggerjaan ticket oleh internal user dengan role developer yang ditunjukkan pada gambar ??.



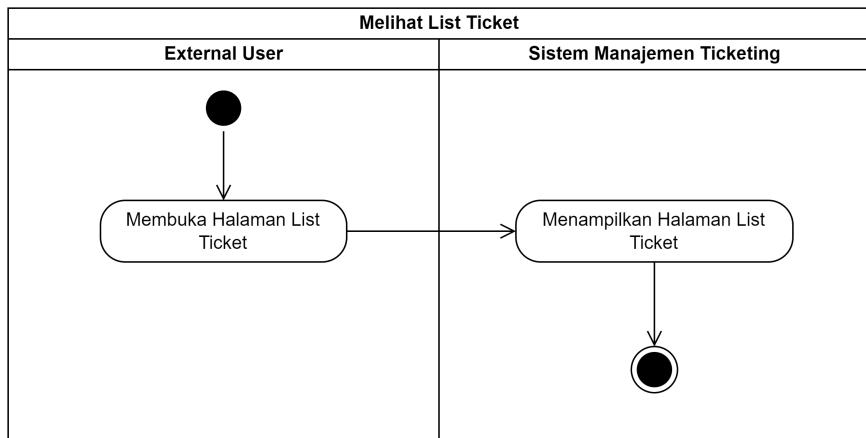
Gambar 4.13: Activity Diagram Melaporkan Pengerajan Ticket (DEV)

#### 4.2.1.3. Activity Diagram External User

##### 1. Activity Diagram Melihat List Ticket

ID Activity : AD-PIC-01

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk melihat list ticket oleh seorang external user yang ditunjukkan pada gambar ??.

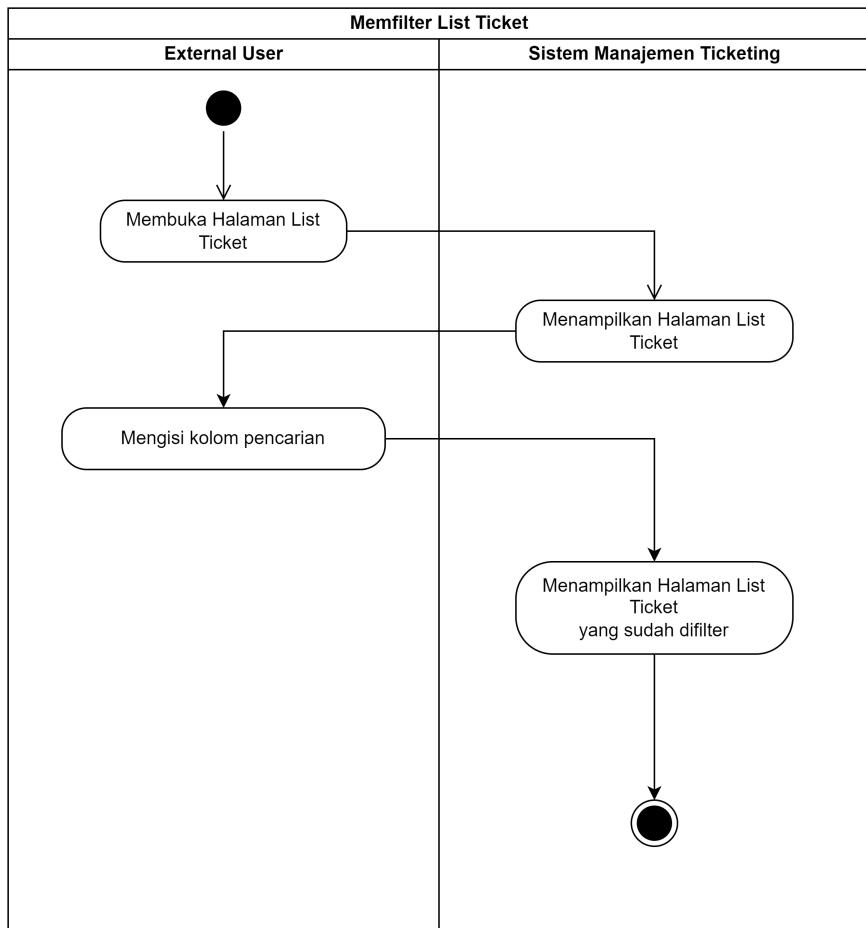


Gambar 4.14: Activity Diagram Melihat List Ticket (External User)

## 2. Activity Diagram Memfilter List Ticket

ID Activity : AD-PIC-02

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk memfilter ticket oleh seorang external user yang ditunjukkan pada gambar ??.

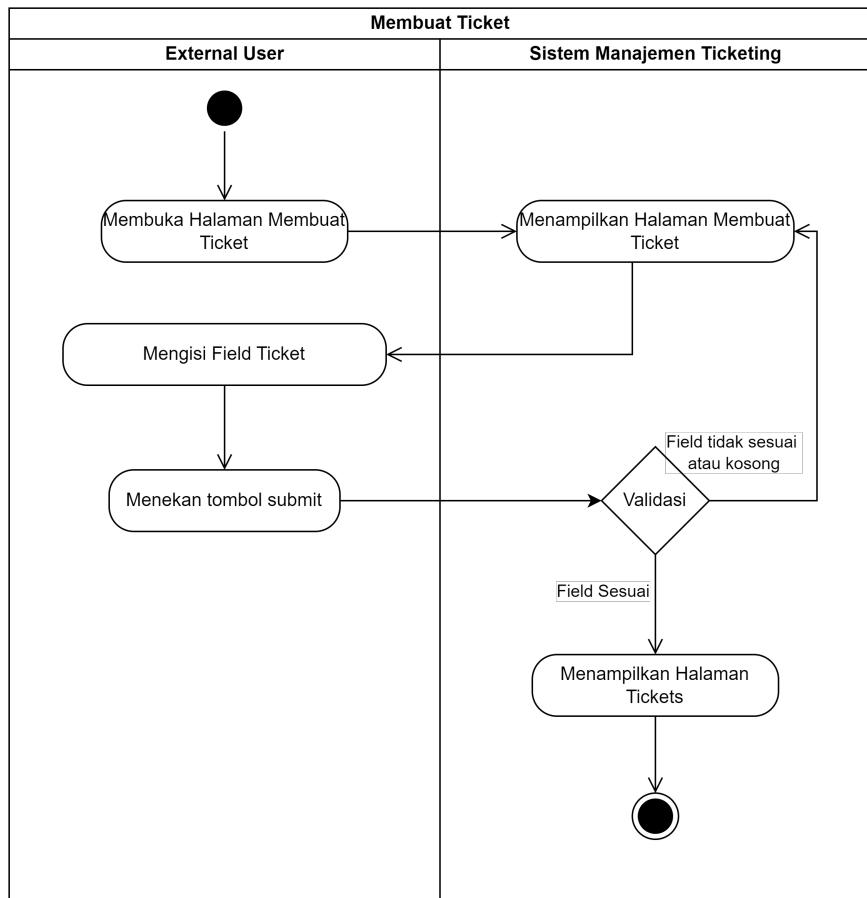


Gambar 4.15: Activity Diagram Memfilter List Ticket (External User)

### 3. Activity Diagram Membuat Ticket

ID Activity : AD-PIC-03

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk membuat ticket oleh seorang external user yang ditunjukkan pada gambar ??.

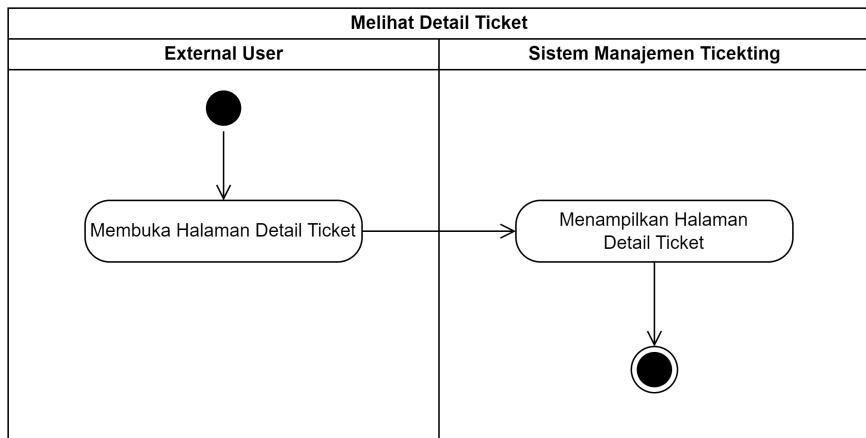


Gambar 4.16: Activity Diagram Membuat Ticket (External User)

#### 4. Activity Diagram Melihat Detail Ticket

ID Activity : AD-PIC-04

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk melihat detail ticket oleh seorang external user yang ditunjukkan pada gambar ??.

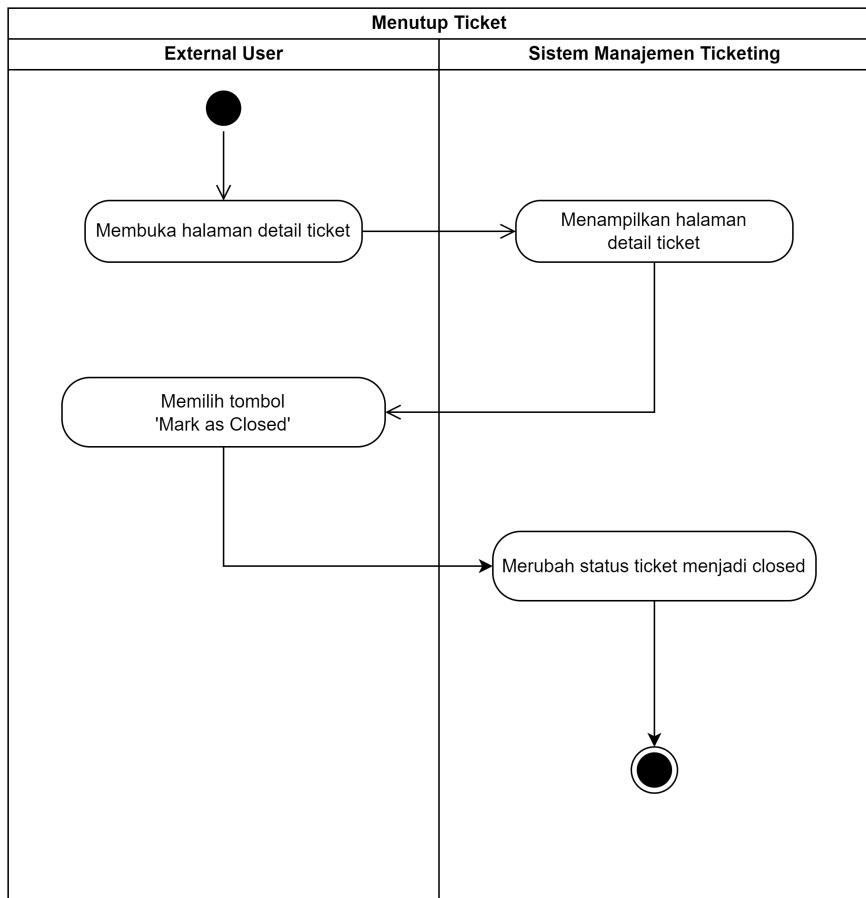


Gambar 4.17: Activity Diagram Melihat Detail Ticket (External User)

##### 5. Activity Diagram Menutup Ticket

ID Activity : AD-PIC-05

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk menutup ticket yang telah dibuat oleh seorang external user dan dikerjakan internal user yang ditunjukkan pada gambar ??.

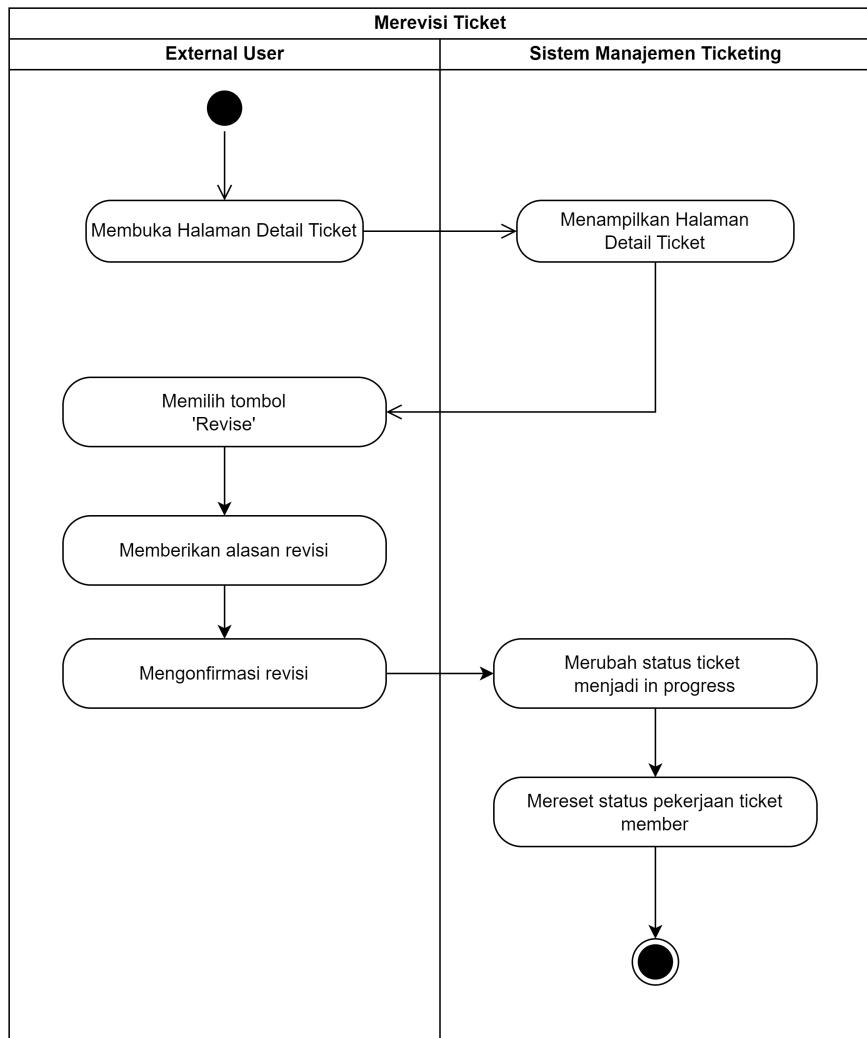


Gambar 4.18: Activity Diagram Menutup Ticket (External User)

#### 6. Activity Diagram Merevisi Ticket

ID Activity : AD-PIC-06

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk merevisi ticket yang telah dibuat oleh seorang external user dan dikerjakan internal user yang ditunjukkan pada gambar ??.

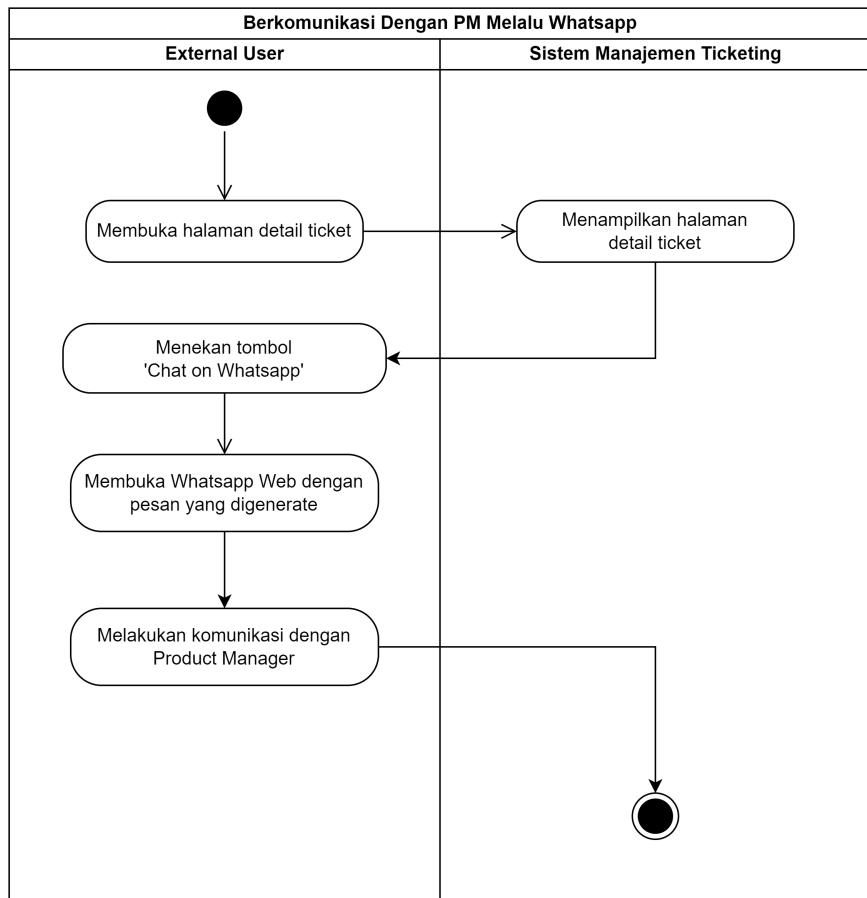


Gambar 4.19: Activity Diagram Merevisi Ticket (External User)

#### 7. Activity Diagram Berkomunikasi dengan Product Manager

ID Activity : AD-PIC-07

Deskripsi : Activity Diagram berikut merupakan sebuah proses aktivitas yang dilakukan untuk berkomunikasi dengan Product Manager oleh seorang external user yang ditunjukkan pada gambar ??.



Gambar 4.20: Activity Diagram Berkommunikasi Dengan PM Melalui Whatsapp (External User)

#### **4.2.2. Sequence Diagram**

Berdasarkan activity diagram yang telah dibuat, berikut adalah sequence diagram untuk Sistem Informasi Manajemen Produk dan Ticketing bagian ticketing di PT. Teknologi Aplikasi Sejahtera

##### **4.2.2.1. Sequence Diagram Internal User (Product Manager)**

###### **1. Sequence Diagram Melihat List Ticket**

ID Sequence : SQ-PM-01

Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk melihat list ticket oleh seorang internal user dengan role product manager.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

Class Boundary : Ticket List Page

Class Control : Ticket Controller

Class Entity : User, Employee, Ticket, Ticket Member, Product Member, Ticket Category, Ticket Status, dan Ticket Priority

###### **2. Sequence Diagram Memfilter List Ticket**

ID Sequence : SQ-PM-02

Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk memfilter ticket oleh seorang internal user dengan role product manager.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan ??:

Class Boundary : TicketPage  
Class Control : TicketController  
Class Entity : User, Employee, Ticket, Ticket Member, Product Member, Ticket Category, Ticket Status, dan Ticket Priority

### 3. Sequence Diagram Melihat Detail Ticket

ID Sequence : SQ-PM-03  
Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk melihat detail ticket oleh seorang internal user dengan role product manager.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

Class Boundary : Ticket List Page dan Ticket Details Page  
Class Control : TicketController  
Class Entity : Ticket, Product, Company, Ticket Category, Ticket Status, Ticket Priority, Ticket Member, Product Member, Ticket Member Role, dan Employee

### 4. Sequence Diagram Memverifikasi List Ticket

ID Sequence : SQ-PM-04  
Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk memverifikasi ticket yang dibuat external user oleh seorang internal user dengan role product manager.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

Class Boundary : Ticket Details Page dan Ticket Verify Form  
Class Control : Ticket Controller  
Class Entity : Ticket, Product Member, dan Ticket Member

## 5. Sequence Diagram Menolak Ticket

ID Sequence : SQ-PM-05

Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk menolak ticket yang dibuat external user oleh seorang internal user dengan role product manager.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

Class Boundary : Ticket Details Page dan Ticket Reject Form

Class Control : Ticket Controller

Class Entity : Ticket

## 6. Sequence Diagram Merevisi Ticket

ID Sequence : SQ-PM-06

Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk merevisi ticket yang telah dikerjakan developer oleh seorang internal user dengan role product manager.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

Class Boundary : Ticket Details Page dan Ticket Revise Form

Class Control : TicketController

Class Entity : Ticket

## 7. Sequence Diagram Berkomunikasi dengan external user

ID Sequence : SQ-PM-07

Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk berkomunikasi dengan external user oleh seorang internal user dengan role product manager.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan

pada gambar ??:

Class Boundary : Ticket Details Page  
Class Control : TicketController  
Class Entity : Ticket, Ticket Member, Product Member, dan Employee

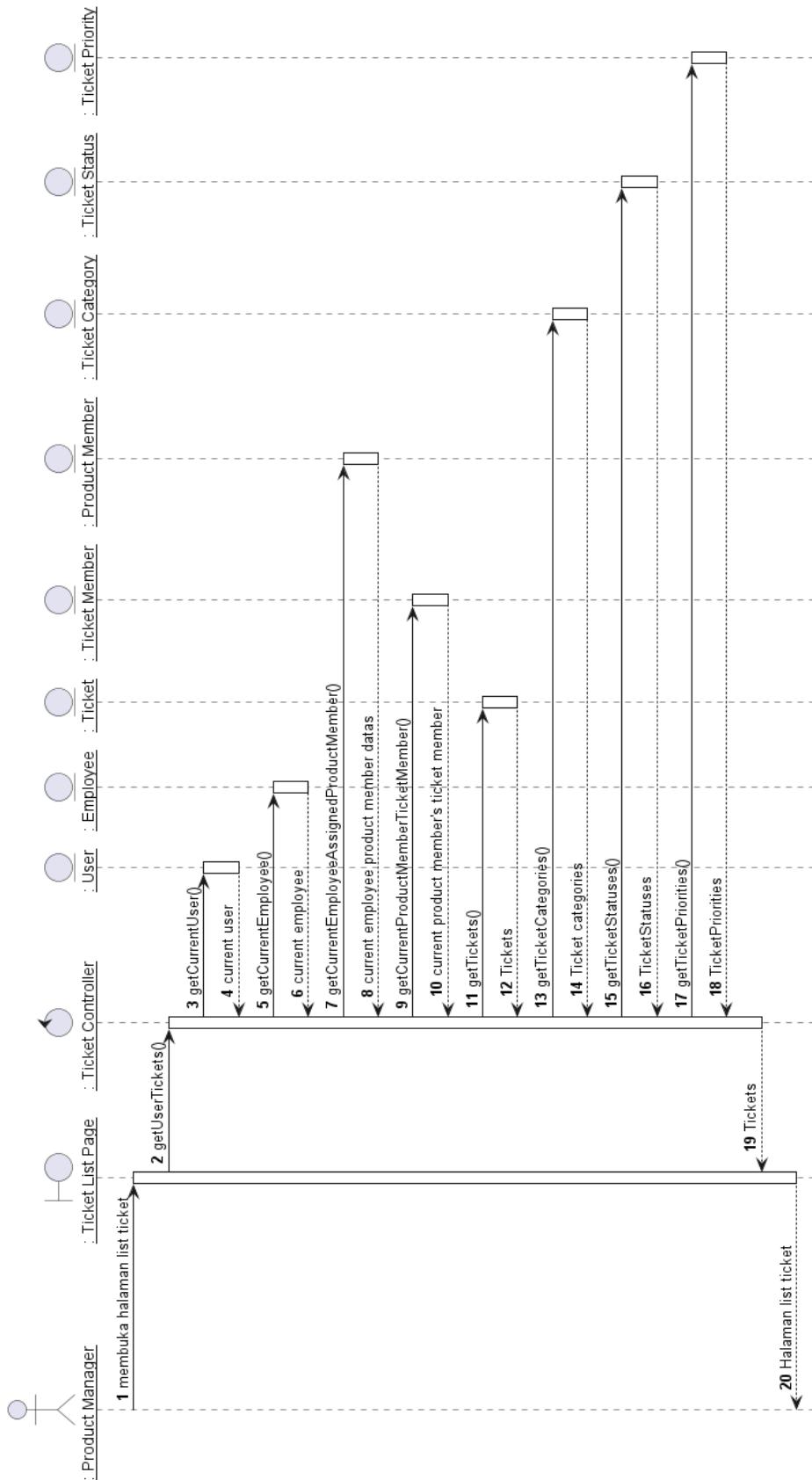
#### 8. Sequence Diagram Mengubah Status Ticket Menjadi Done

ID Sequence : SQ-PM-08

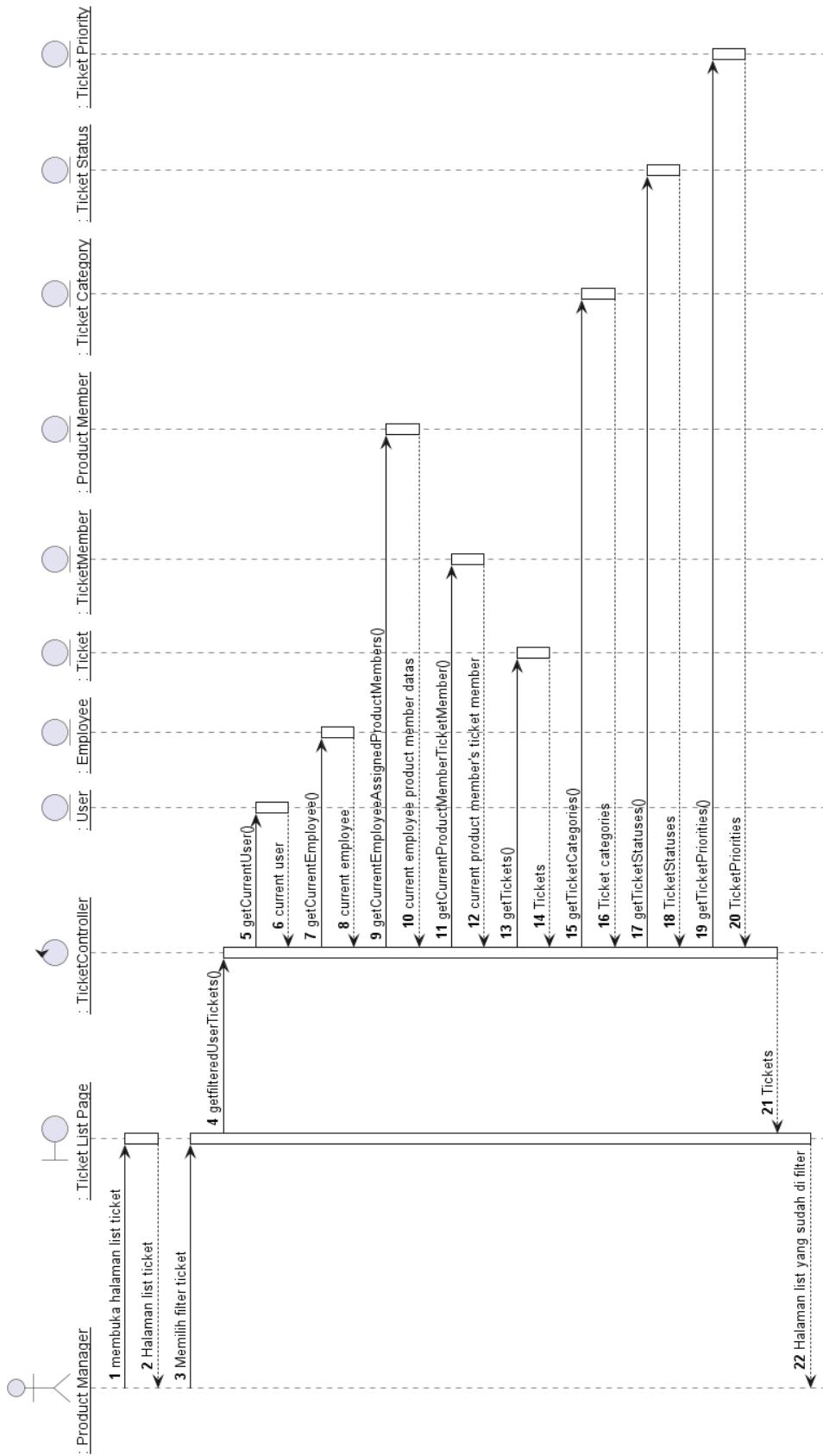
Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk mengubah status ticket menjadi done oleh internal user dengan role product manager.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

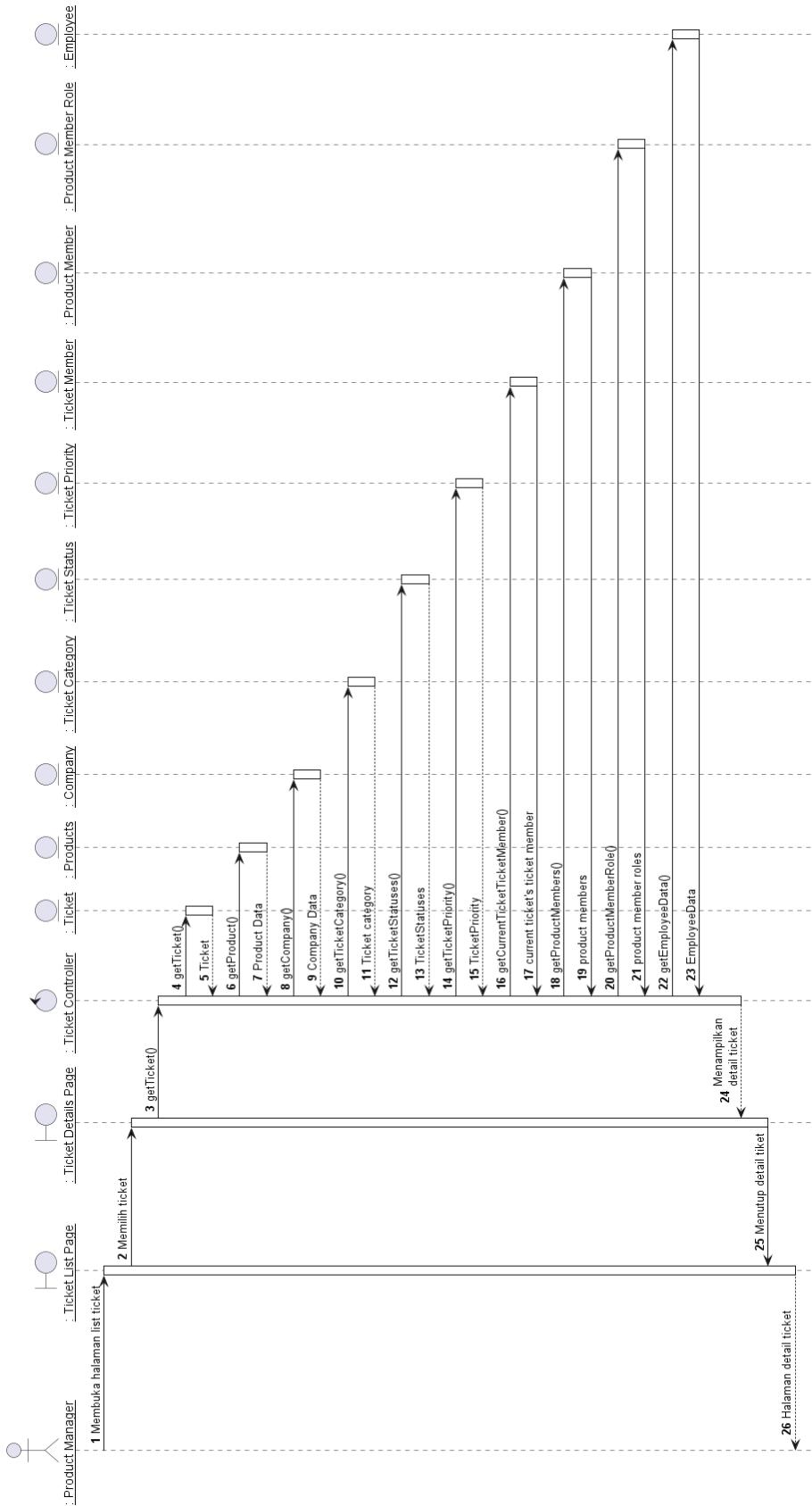
Class Boundary : Ticket Details Page  
Class Control : TicketController  
Class Entity : Ticket



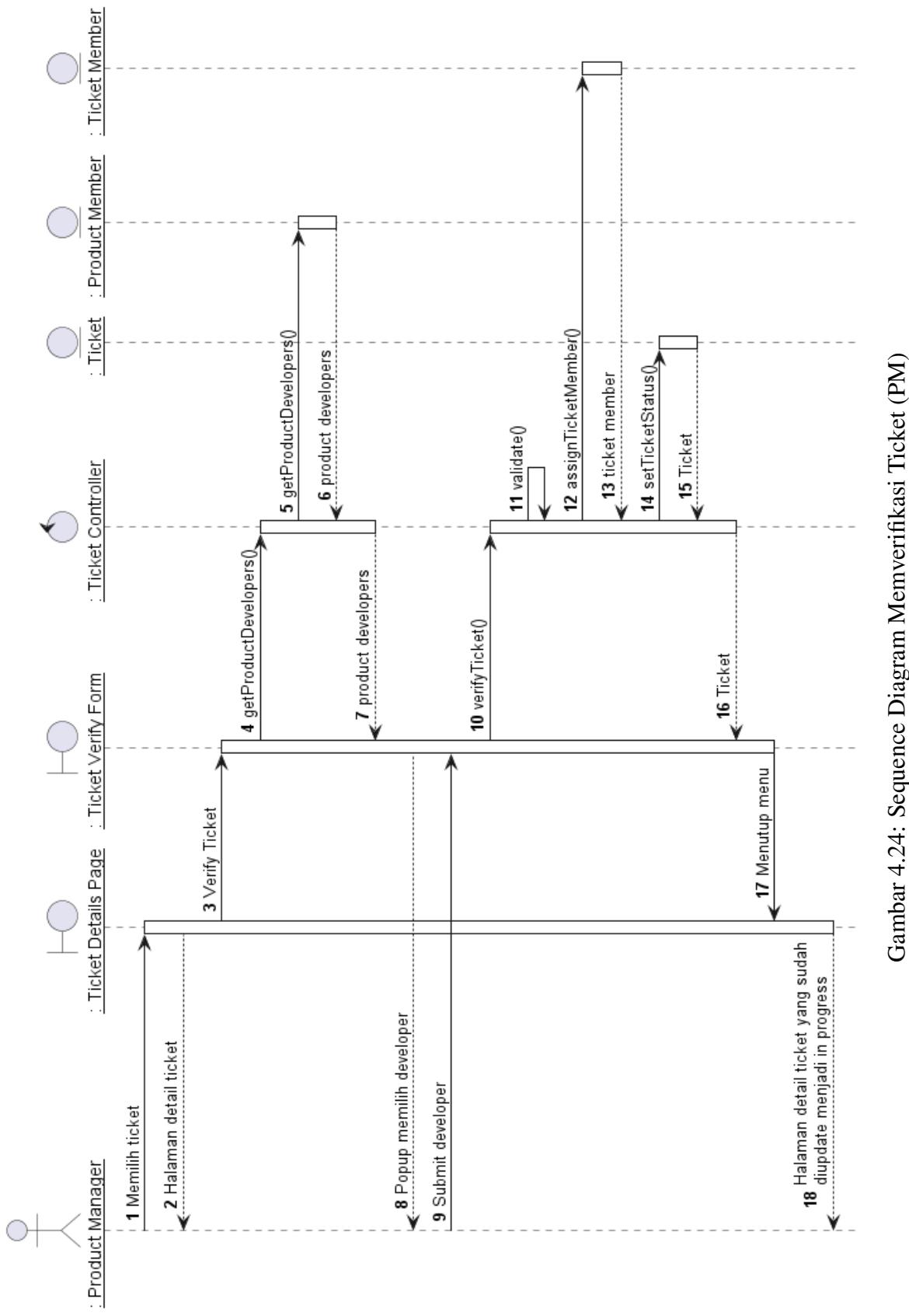
Gambar 4.21: Sequence Diagram Melihat List Ticket (PM)



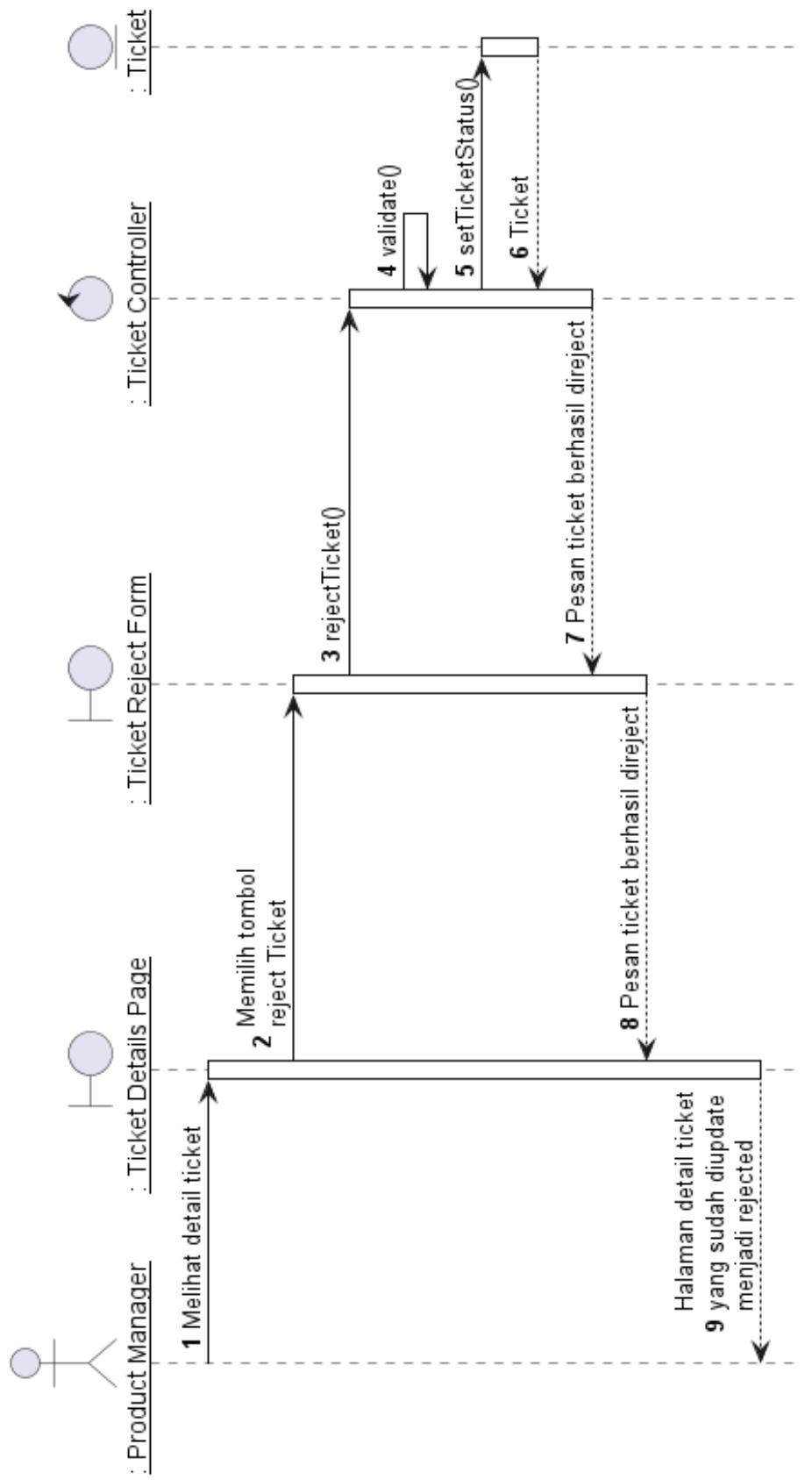
Gambar 4.22: Sequence Diagram Memfilter List Ticket (PM)



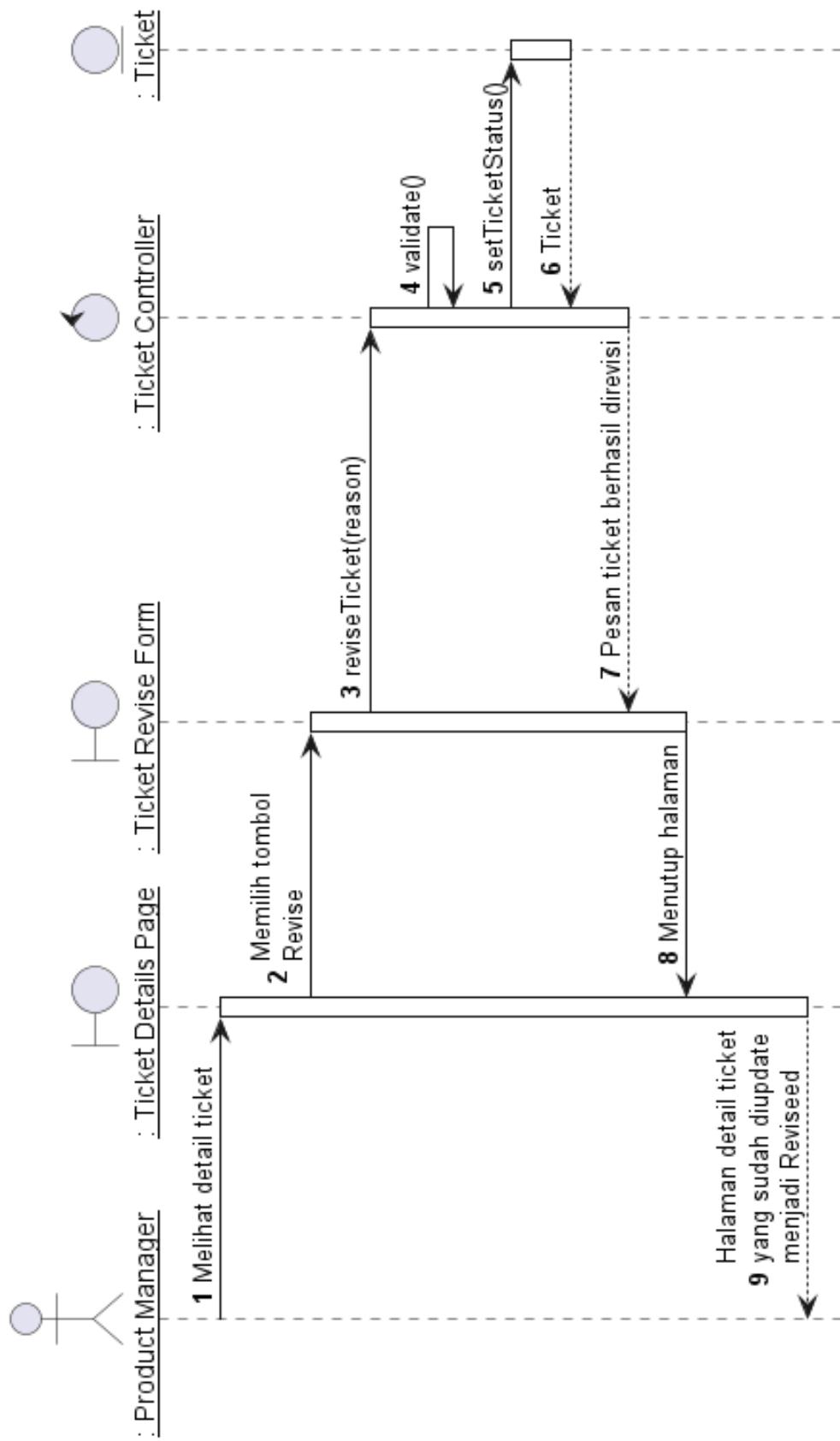
Gambar 4.23: Sequence Diagram Melihat Detail Ticket (PM)



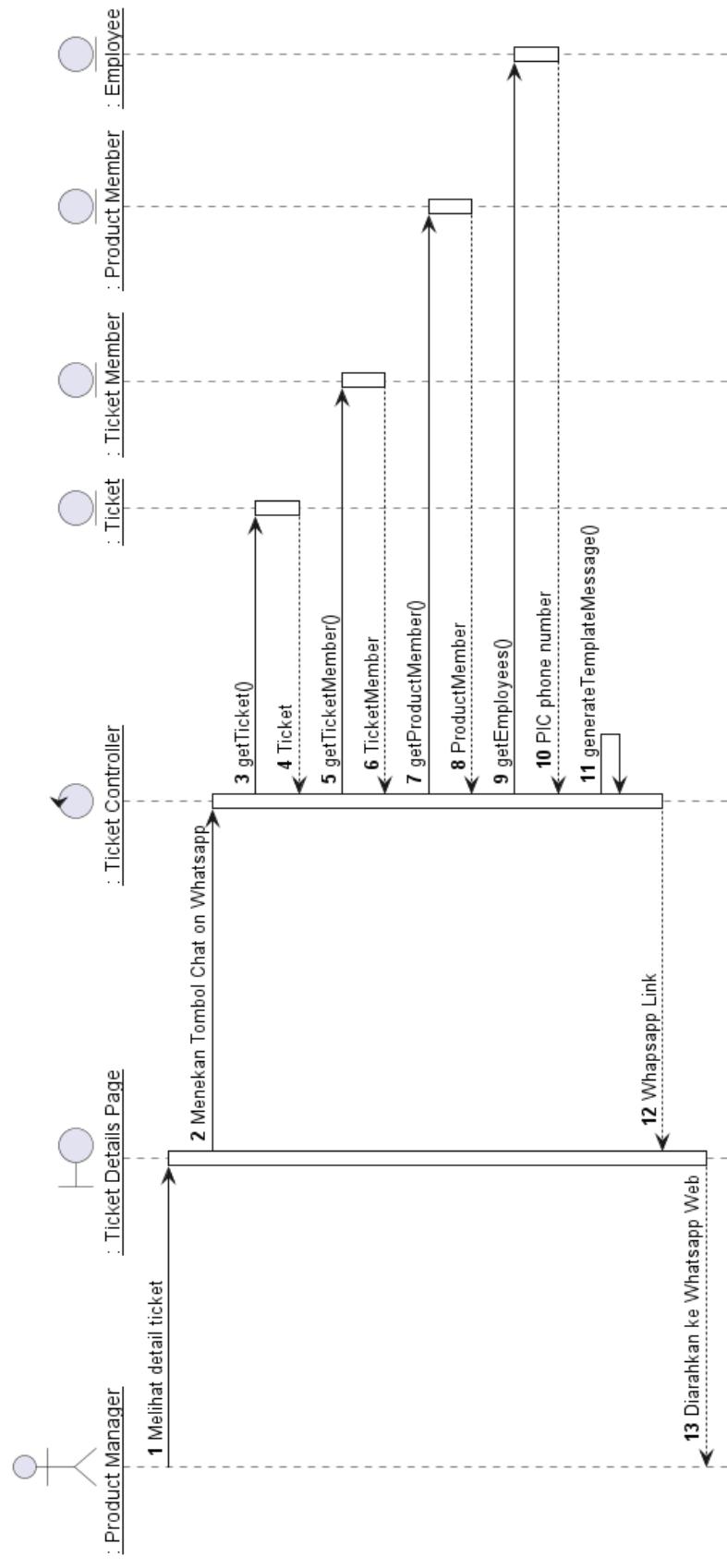
Gambar 4.24: Sequence Diagram Memverifikasi Ticket (PM)



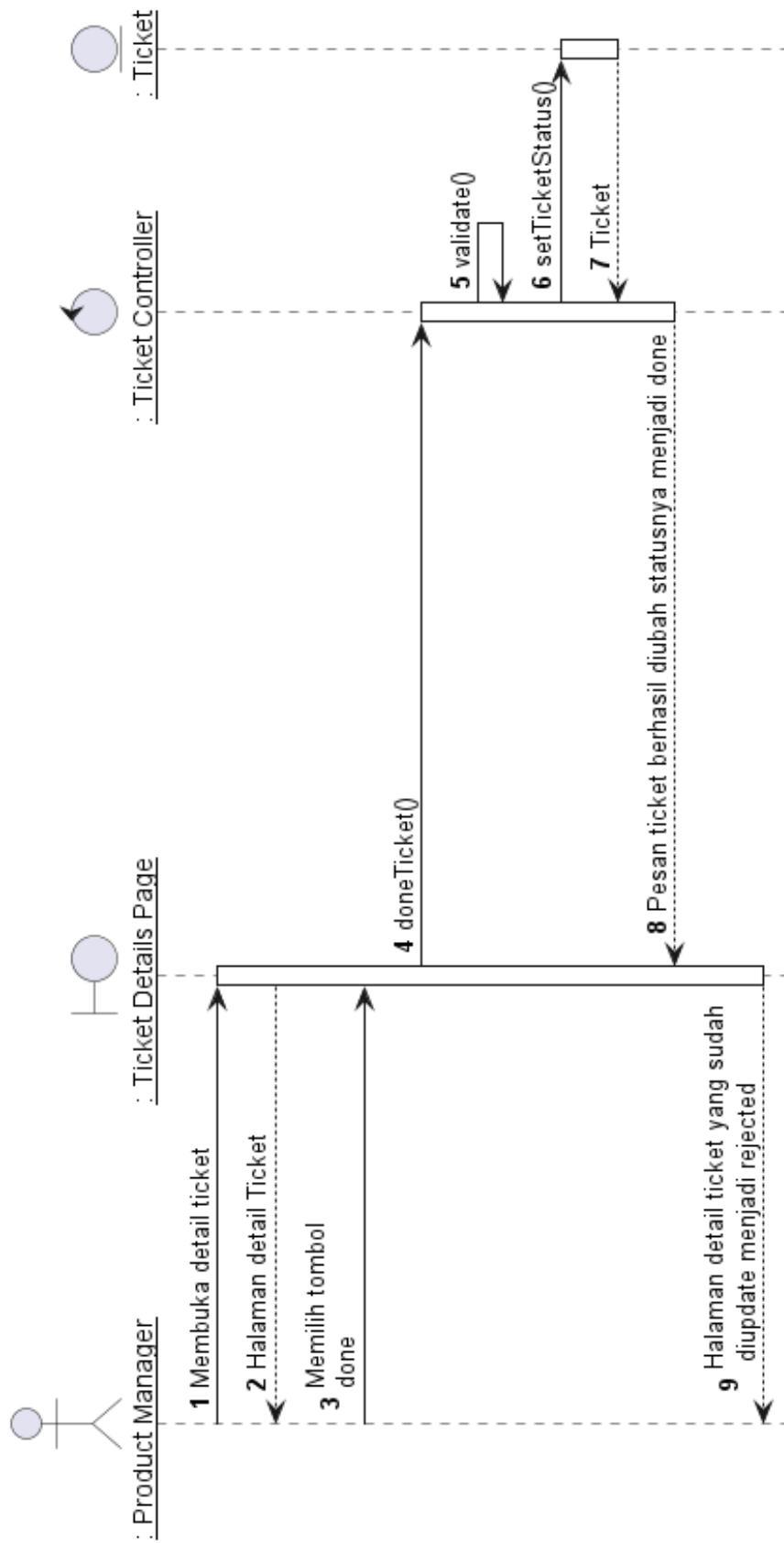
Gambar 4.25: Sequence Diagram Menolak Ticket (PM)



Gambar 4.26: Sequence Diagram Merevisi Ticket (PM)



Gambar 4.27: Sequence Diagram Berkomunikasi Dengan PM Melalui WhatsApp (PM)



Gambar 4.28: Sequence Diagram Mengubah Status Ticket Menjadi Done (PM)

#### **4.2.2.2. Sequence Diagram Internal User (Developer)**

##### **1. Sequence Diagram Melihat List Ticket**

ID Sequence : SQ-DEV-01

Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk melihat list ticket oleh internal user dengan role developer.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

Class Boundary : Ticket List Page

Class Control : Ticket Controller

Class Entity : User, Employee, Ticket, Ticket Member, Product Member, Ticket Category, Ticket Status, dan Ticket Priority

##### **2. Sequence Diagram Memfilter Ticket**

ID Sequence : SQ-DEV-02

Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk memfilter list ticket oleh internal user dengan role developer.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

Class Boundary : TicketPage

Class Control : TicketController

Class Entity : User, Employee, Ticket, Ticket Member, Product Member, Ticket Category, Ticket Status, dan Ticket Priority

##### **3. Sequence Diagram Melihat Detail Ticket**

ID Sequence : SQ-DEV-03

Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk melihat detail ticket oleh internal user dengan role developer.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

Class Boundary : Ticket List Page dan Ticket Details Page

Class Control : TicketController

Class Entity : Ticket, Product, Company, Ticket Category, Ticket Status, Ticket Priority, Ticket Member, Product Member, Ticket Member Role, dan Employee

#### 4. Sequence Diagram Melaporkan Pengerjaan Ticket

ID Sequence : SQ-DEV-04

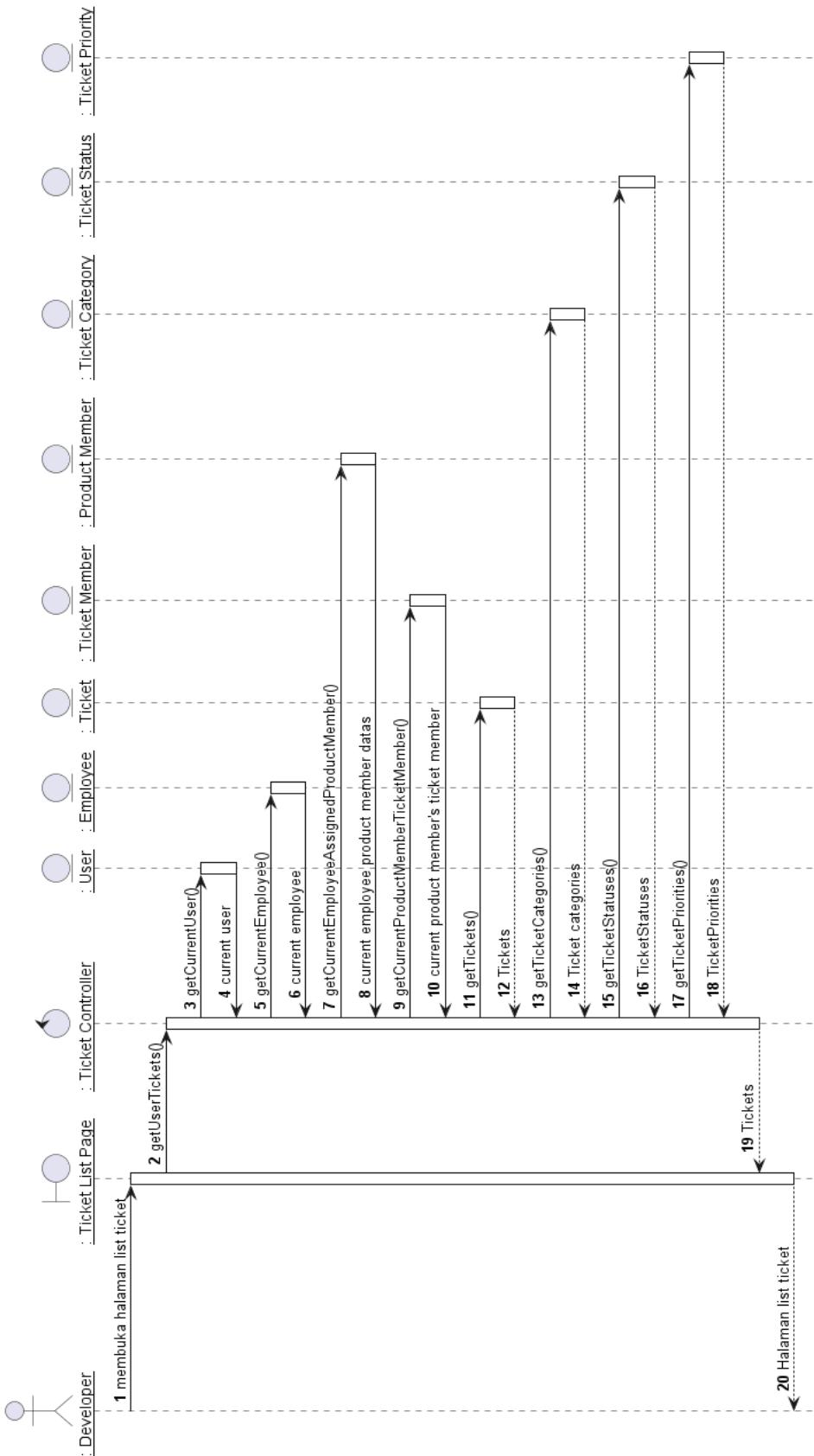
Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk melaporkan pengerjaan ticket oleh internal user dengan role developer.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

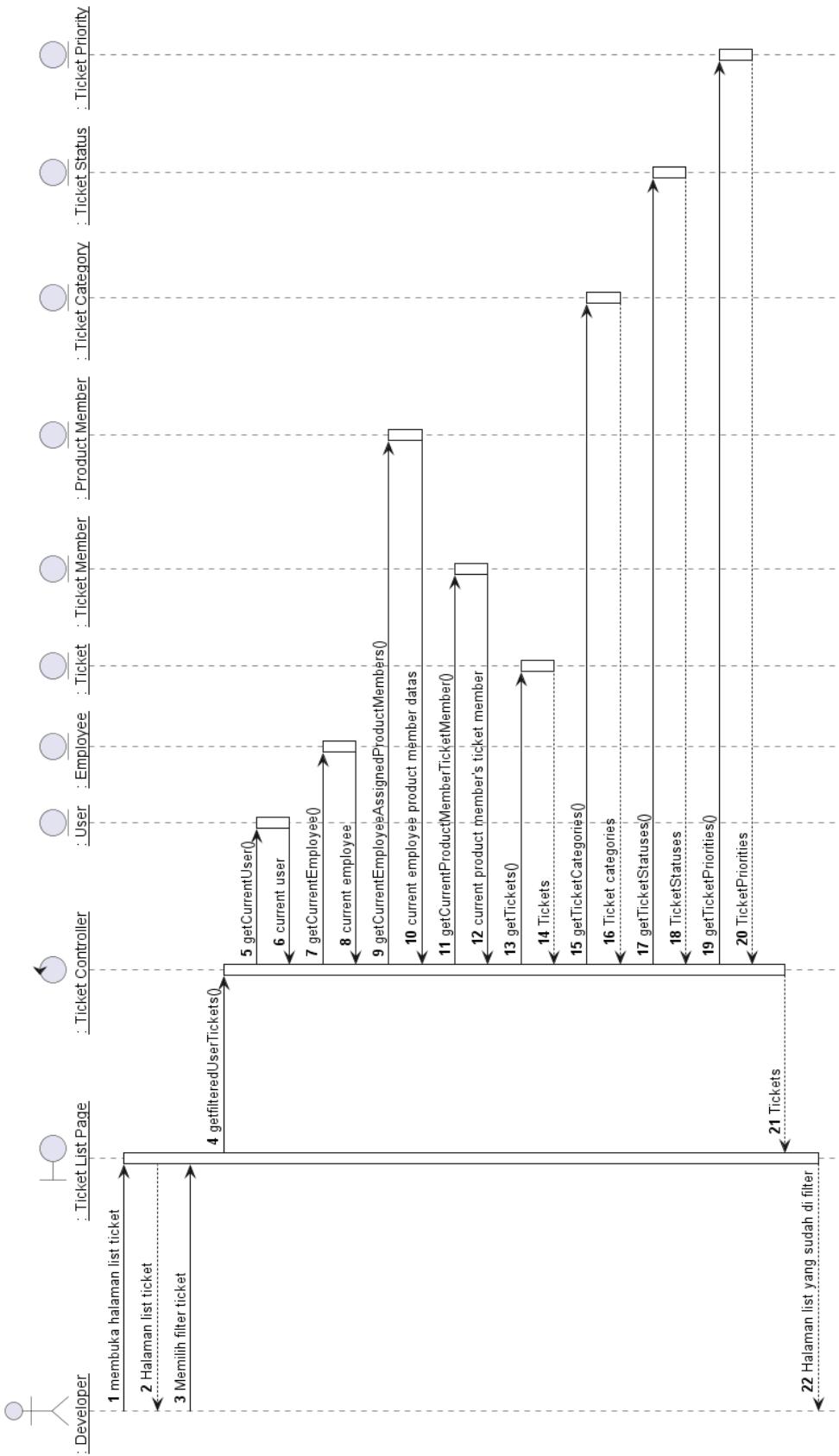
Class Boundary : Ticket Details Page

Class Control : Ticket Controller

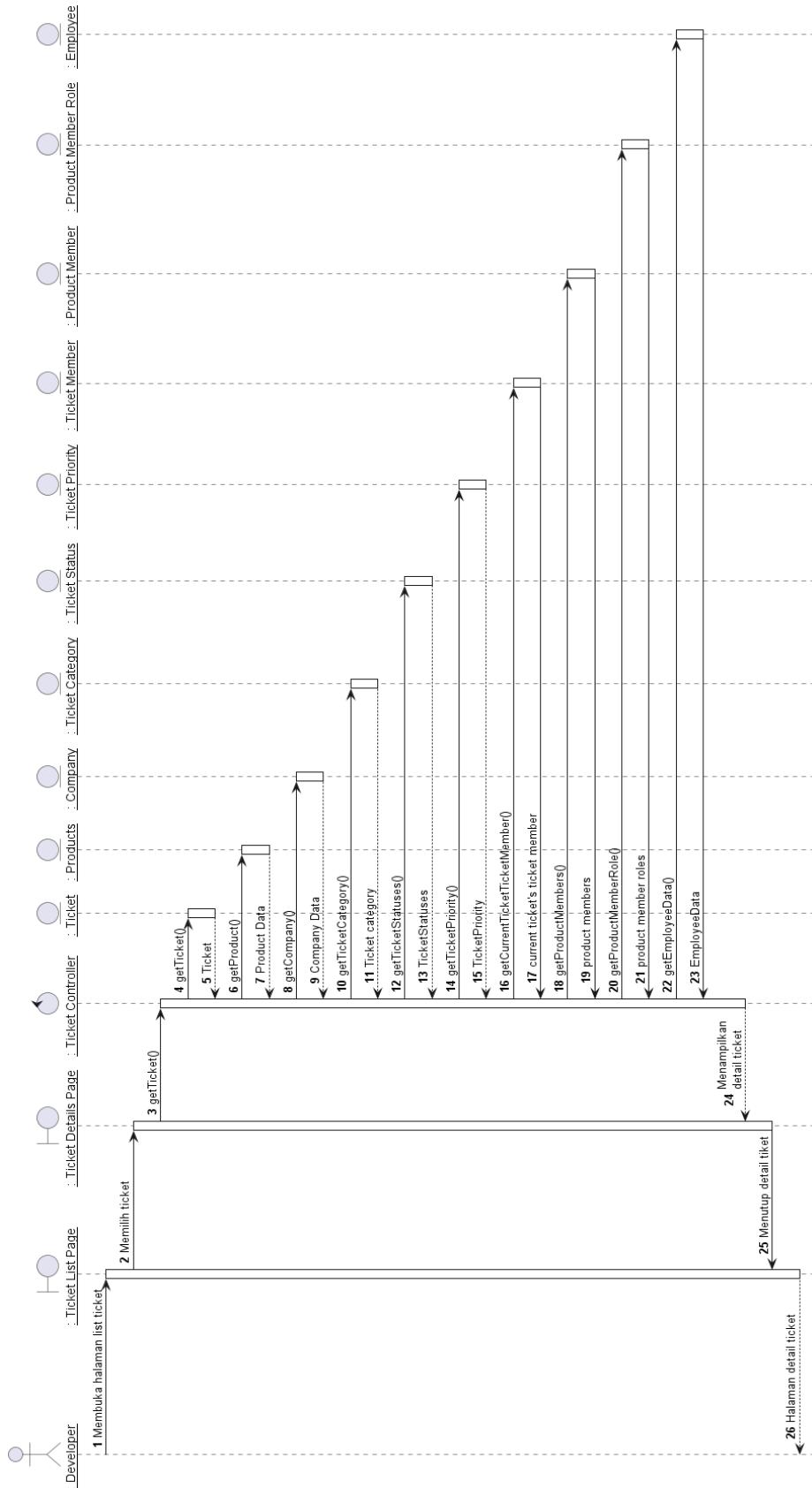
Class Entity : Ticket dan Ticket Member



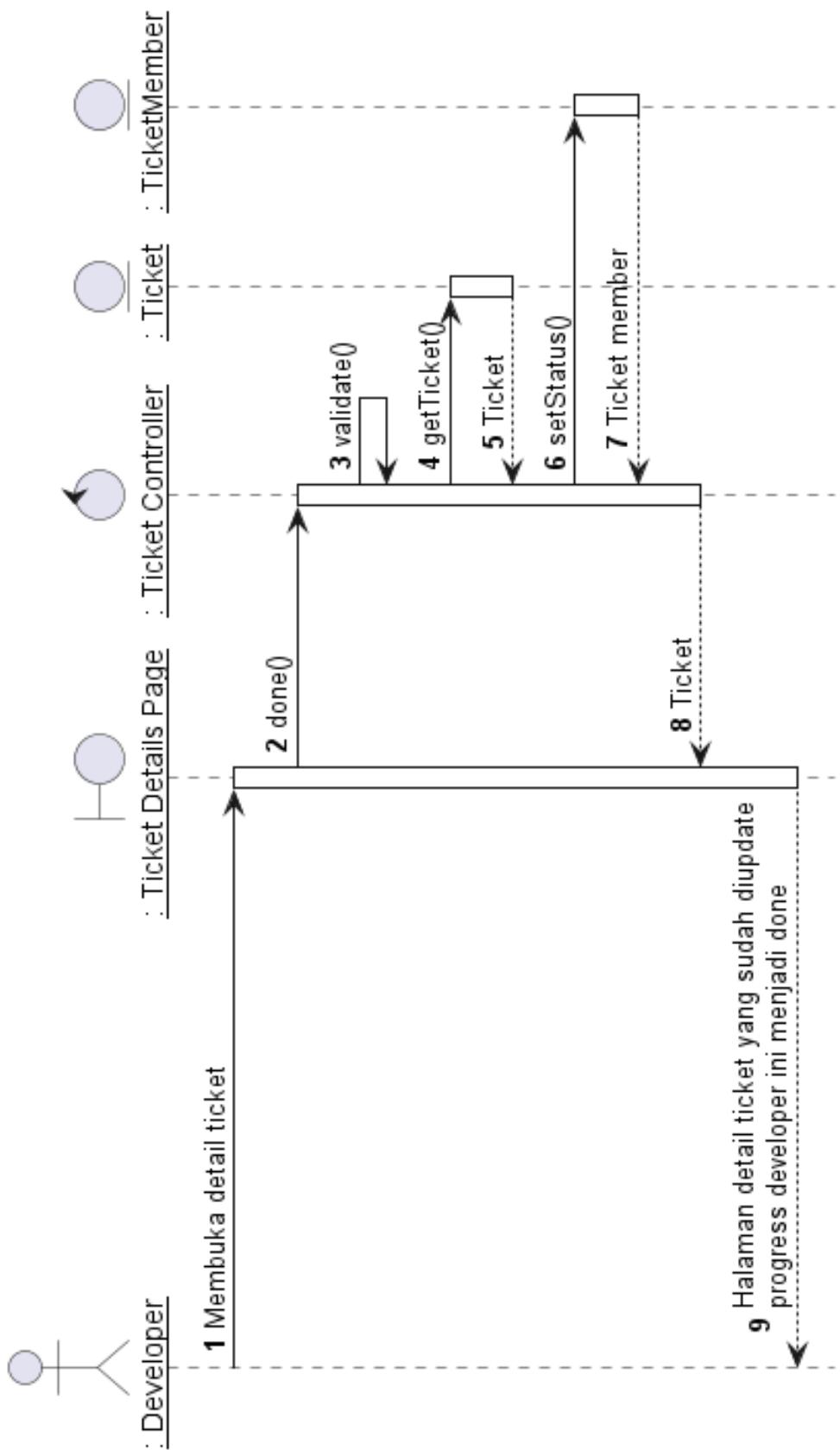
Gambar 4.29: Sequence Diagram Melihat List Ticket (DEV)



Gambar 4.30: Sequence Diagram Memfilter List Ticket (DEV)



Gambar 4.31: Sequence Diagram Melihat Detail Ticket (DEV)



Gambar 4.32: Sequence Diagram Melaporkan Pengerjaan Ticket (DEV)

#### **4.2.2.3. Sequence Diagram External User**

##### **1. Sequence Diagram Melihat List Ticket**

ID Sequence : SQ-PIC-01

Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk melihat list ticket oleh seorang external user.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

Class Boundary : Ticket List Page

Class Control : Ticket Controller

Class Entity : User, Employee, Ticket, Ticket Member, Product Member, Ticket Category, Ticket Status, dan Ticket Priority

##### **2. Sequence Diagram Memfilter List Ticket**

ID Sequence : SQ-PIC-02

Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk memfilter ticket oleh seorang external user.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

Class Boundary : TicketPage

Class Control : TicketController

Class Entity : User, Employee, Ticket, Ticket Member, Product Member, Ticket Category, Ticket Status, dan Ticket Priority

##### **3. Sequence Diagram Membuat Ticket**

ID Sequence : SQ-PIC-03

Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk membuat ticket oleh seorang external user.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

Class Boundary : Ticket List Page dan Ticket Create Page

Class Control : TicketController

Class Entity : Ticket, Product, Ticket Category, Ticket Status, Ticket Priority, Ticket Member, Product Member, dan Employee

#### 4. Sequence Diagram Melihat Detail Ticket

ID Sequence : SQ-PIC-04

Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk melihat detail ticket oleh seorang external user.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

Class Boundary : Ticket List Page dan Ticket Details Page

Class Control : TicketController

Class Entity : Ticket, Product, Company, Ticket Category, Ticket Status, Ticket Priority, Ticket Member, Product Member, Ticket Member Role, dan Employee

#### 5. Sequence Diagram Menutup Ticket

ID Sequence : SQ-PIC-05

Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk menutup ticket oleh seorang external user.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan

pada gambar ??:

- Class Boundary : Ticket Details Page
- Class Control : Ticket Controller
- Class Entity : Ticket

#### 6. Sequence Diagram Merevisi Ticket

- ID Sequence : SQ-PIC-06
- Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk merevisi ticket oleh seorang external user.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

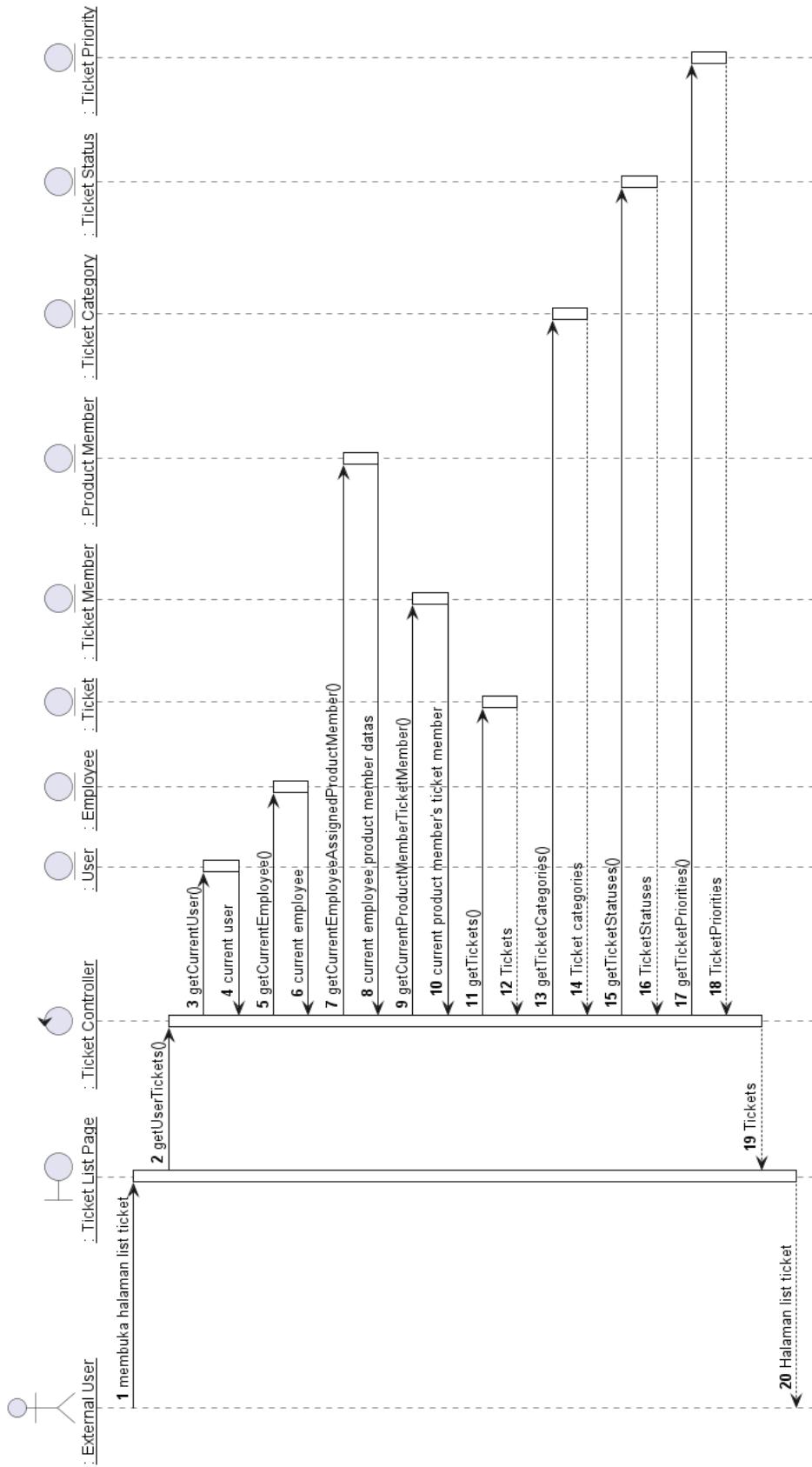
- Class Boundary : Ticket Details Page dan Ticket Revise Form
- Class Control : TicketController
- Class Entity : Ticket

#### 7. Sequence Diagram Berkommunikasi dengan Product Manager

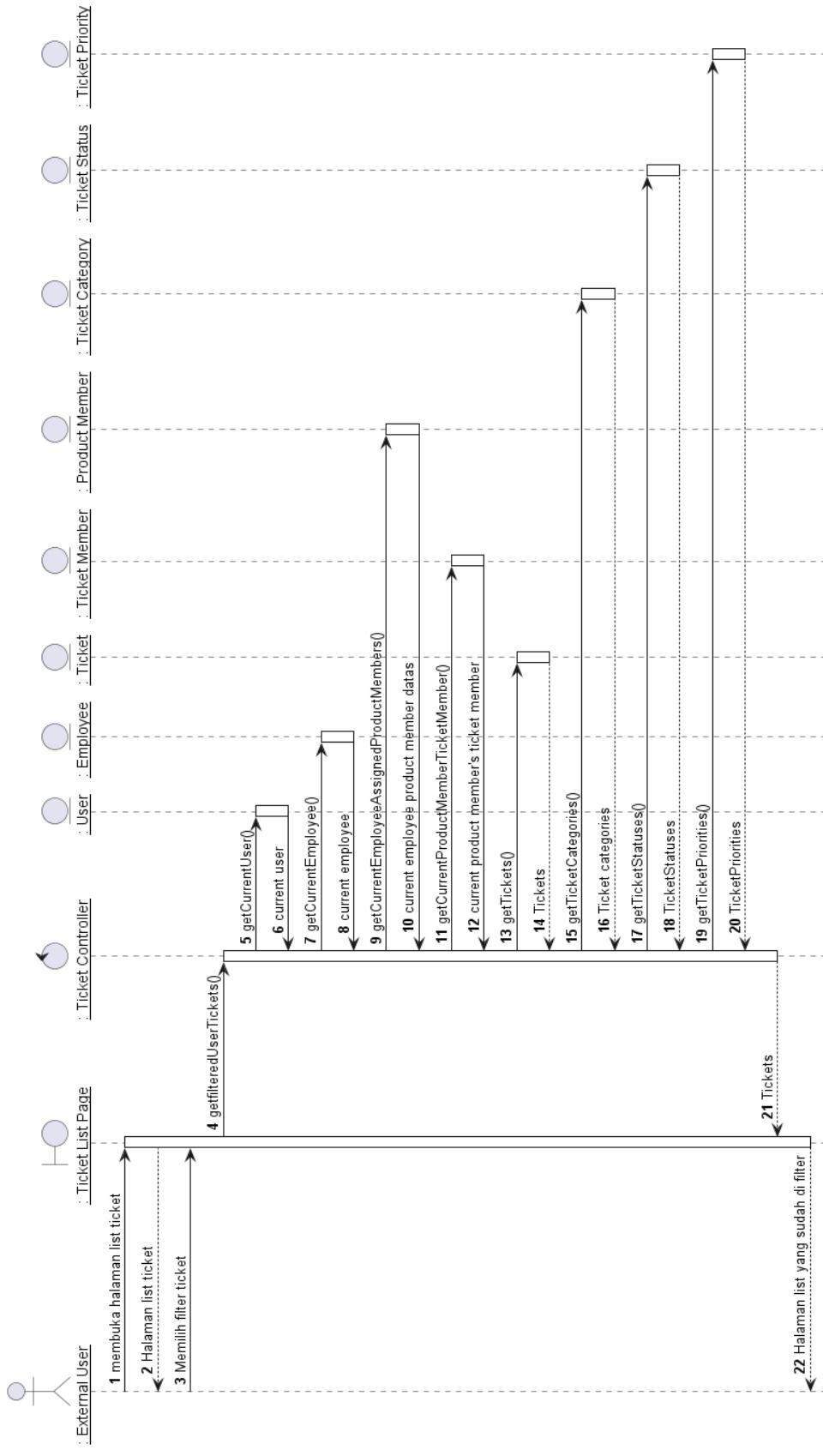
- ID Sequence : SQ-PIC-07
- Deskripsi : Sequence Diagram yang ditunjukkan pada gambar ?? merupakan sebuah proses aktivitas yang dilakukan untuk berkomunikasi dengan Product Manager oleh seorang external user.

Berikut merupakan class class yang ada pada sequence diagram yang ditunjukkan pada gambar ??:

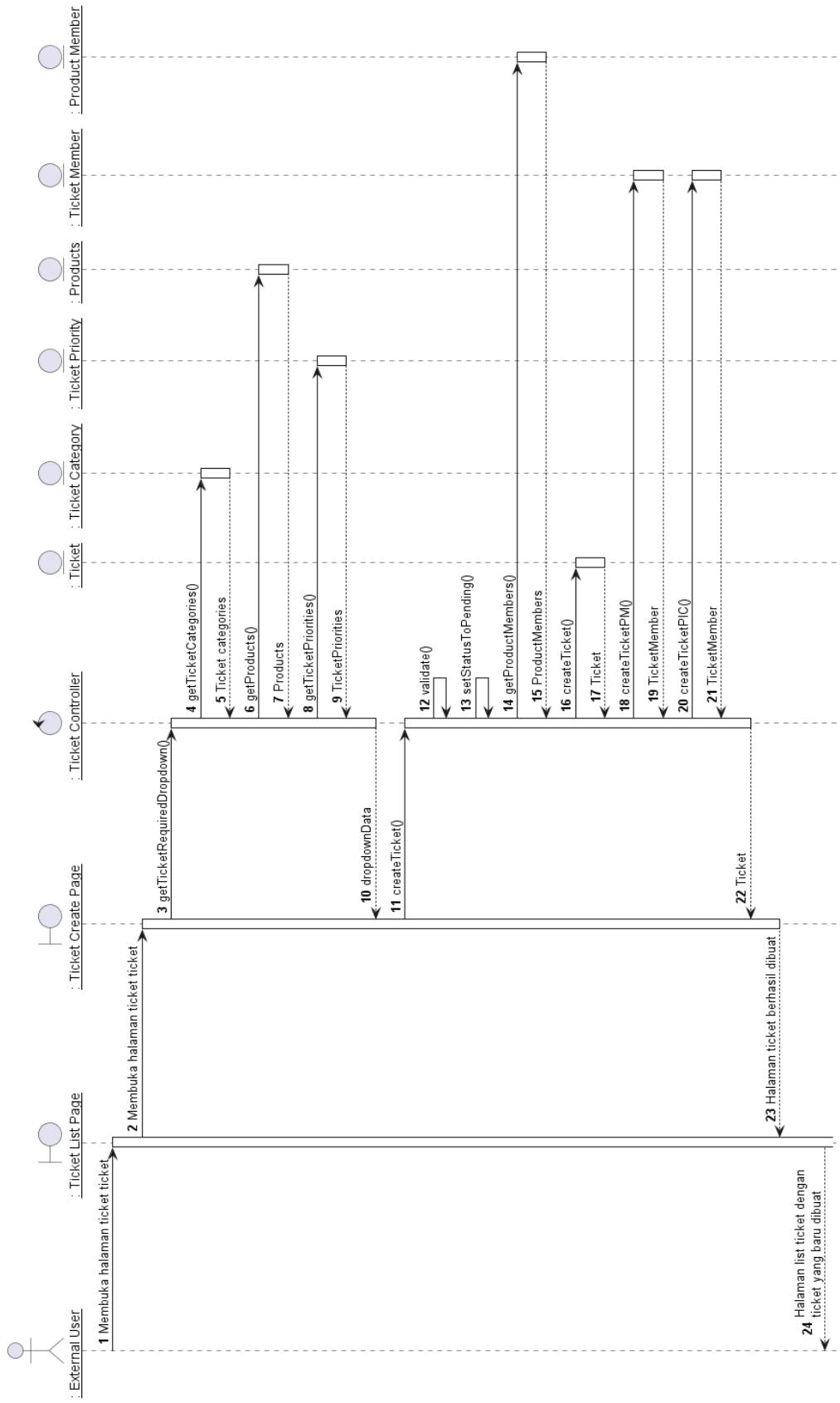
- Class Boundary : Ticket Details Page
- Class Control : TicketController
- Class Entity : Ticket, Ticket Member, Product Member, dan Employee



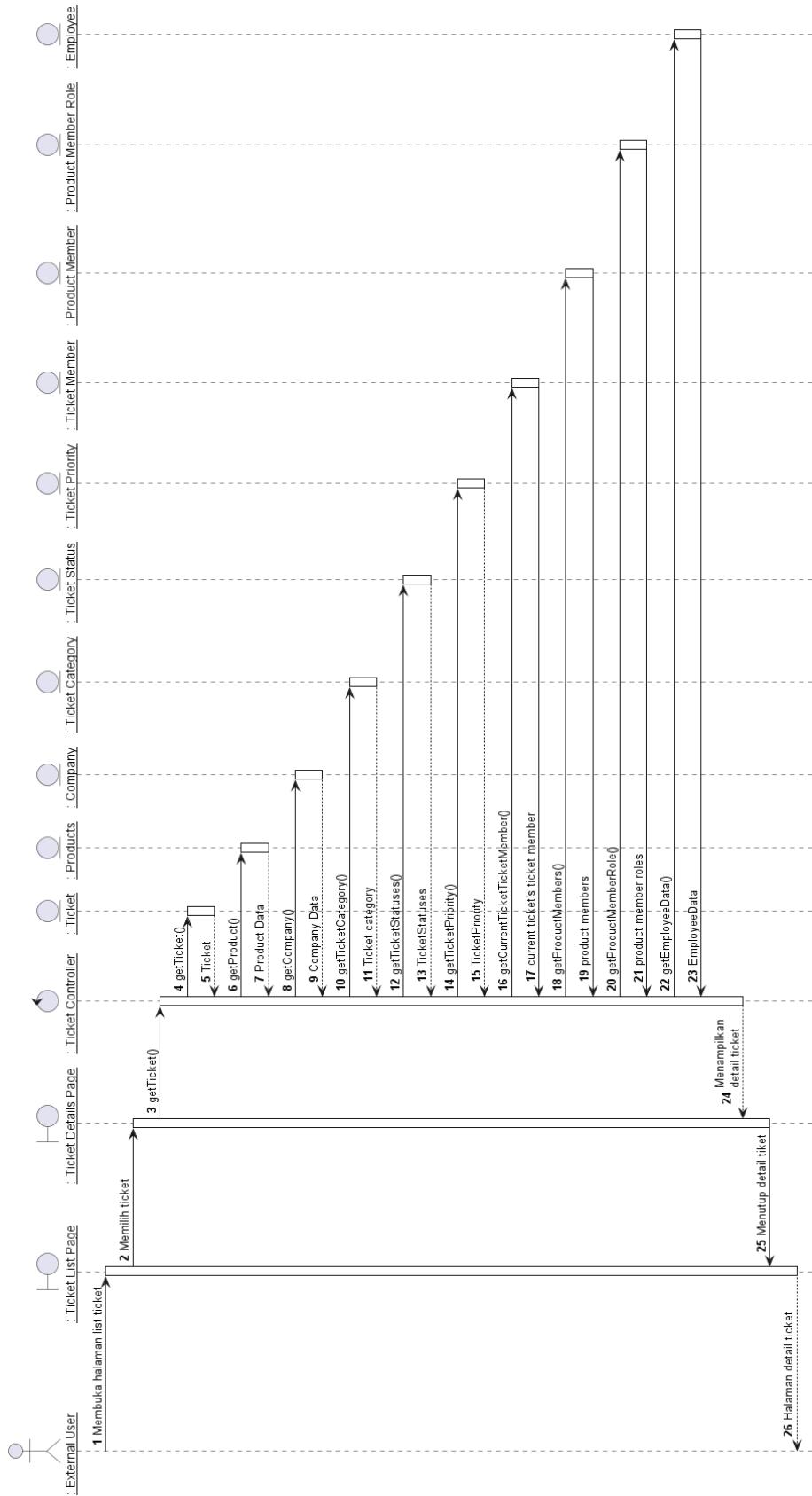
Gambar 4.33: Sequence Diagram Melihat List Ticket (External User)



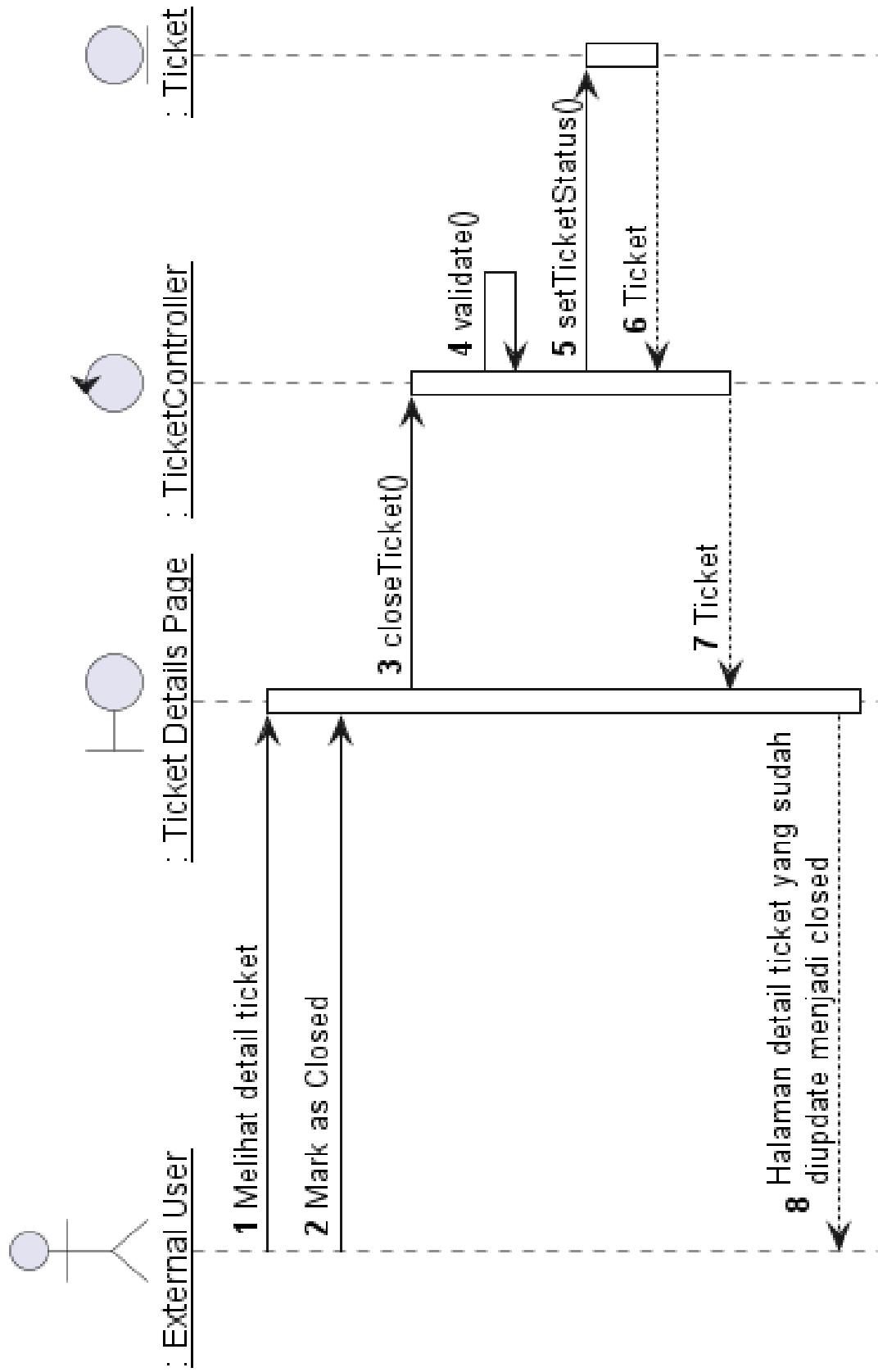
Gambar 4.34: Sequence Diagram Memfilter List Ticket (External User)



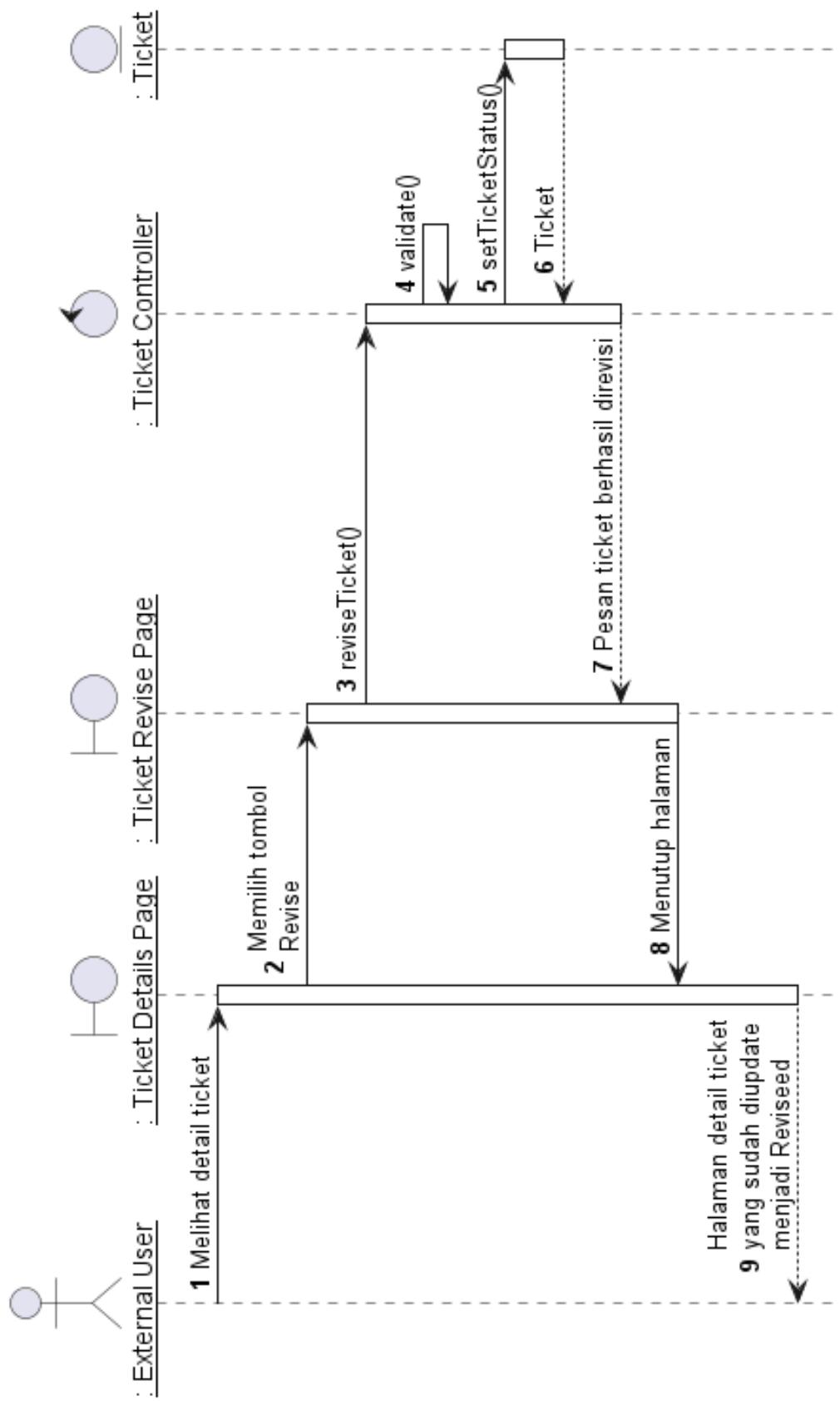
Gambar 4.35: Sequence Diagram Membuat Ticket (External User)



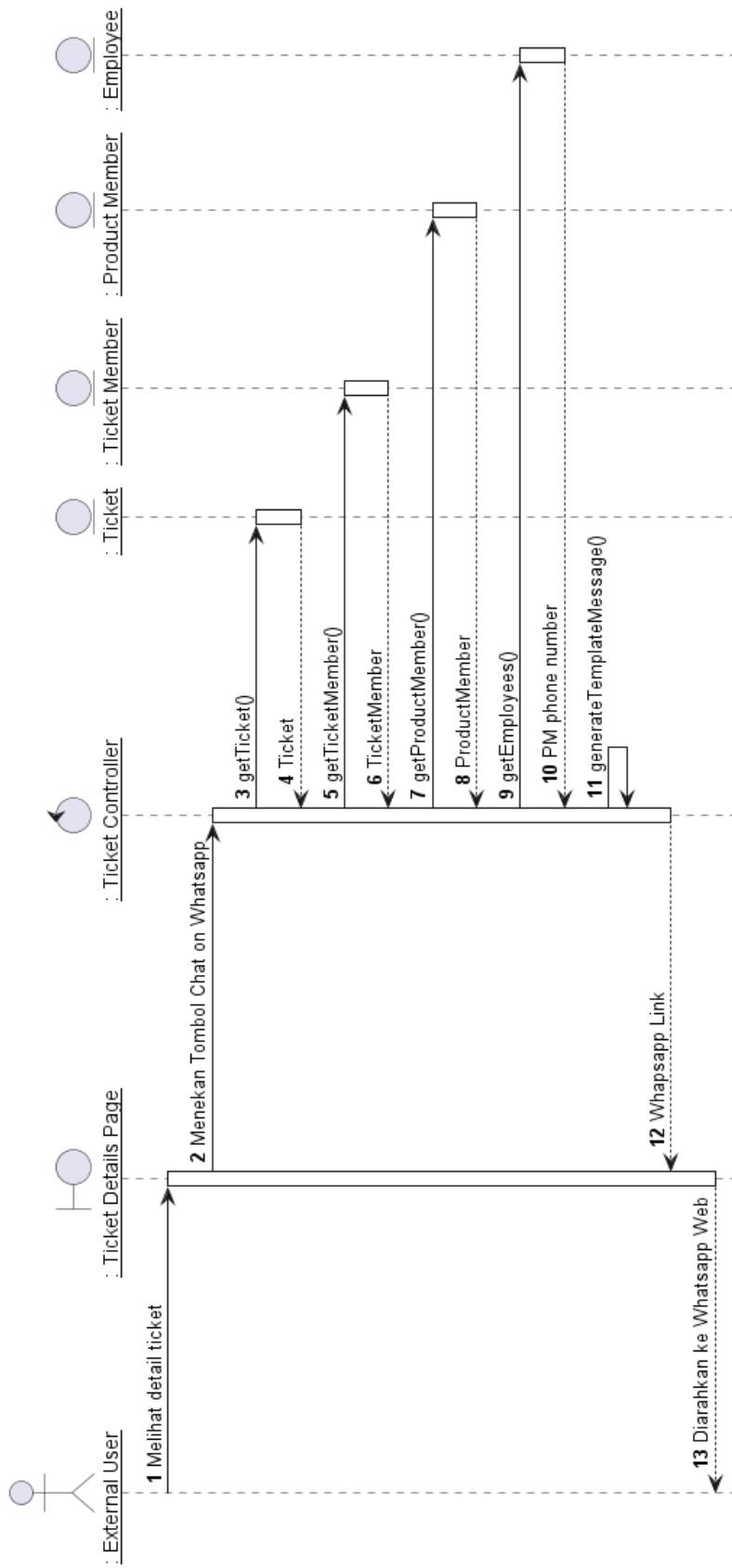
Gambar 4.36: Sequence Diagram Melihat Detail Ticket (External User)



Gambar 4.37: Sequence Diagram Menutup Ticket (External User)



Gambar 4.38: Sequence Diagram Merevisi Ticket (External User)



Gambar 4.39: Sequence Diagram Berkommunikasi Dengan PM Melalui Whatsapp (External User)

#### **4.2.3. Class Diagram**

Class diagram merupakan diagram yang menggambarkan kelas-kelas yang akan digunakan dalam perangkat lunak beserta hubungan antar kelas. Setiap kelas memiliki nama kelas, atribut, dan metode dari kelas yang dibuat. Bagian ini akan membahas class diagram dari aplikasi Sistem Informasi Manajemen Produk dan Ticketing bagian modul ticketing yang telah dianalisa di bagian sequence diagram.

##### **4.3.1. Identifikasi Class**

Identifikasi class class yang ada pada Sistem Informasi Manajemen Produk dan Ticketing bagian modul ticketing dapat dilihat pada tabel ??

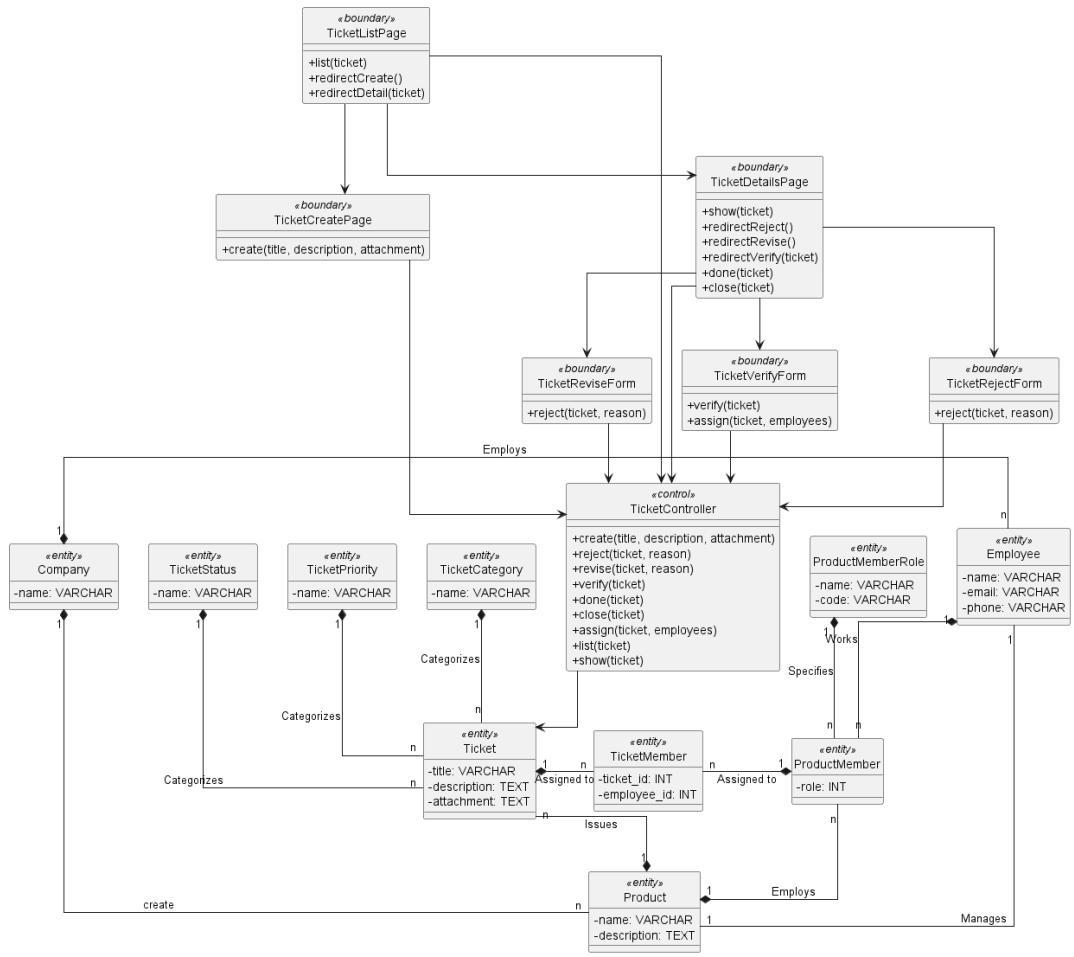
Tabel 4.4: Identifikasi Class

| No. | Stereotype | Nama Class   | Deskripsi  |
|-----|------------|--------------|--|
| 1   | Entity     | User         | Entitas yang menyimpan informasi pengguna login. Seorang user menyimpan informasi login dan memiliki relasi ke employee yang dimiliki user ini.  |
| 2   | Entity     | Employee     | Entitas yang menyimpan informasi karyawan. Entitas ini berelasi 1:m dengan Product sebagai product manager dan berelasi 1:m dengan ProductMember sebagai member dari sebuah Product                                      |
| 3   | Entity     | Product      | Entitas yang menyimpan informasi suatu produk. Entitas ini berelasi 1:m dengan ticket untuk aggregasi ticket pada product serta berelasi 1:m dengan ProductMember untuk aggregasi employee yang bekerja pada product ini |
| 4   | Entity     | Ticket       | Entitas yang menyimpan informasi ticket, berelasi m:n dengan ProductMember karena tiap product member bisa saja mengerjakan banyak ticket dan tiap ticket bisa saja dikerjakan banyak product member.                    |
| 5   | Entity     | TicketMember | Entitas yang menyimpan informasi mengenai ProductMember yang mengerjakan suatu ticket. Class ini juga menyimpan status penggerjaan dari Employee yang mengerjakan ticket tersebut  |

| No. | Stereotype | Nama Class          | Deskripsi   |
|-----|------------|---------------------|---|
| 6   | Entity     | TicketCategory      | Entitas yang menyimpan informasi mengenai kategori dari suatu ticket.   |
| 7   | Entity     | TicketStatus        | Entitas yang menyimpan informasi mengenai status dari suatu ticket.   |
| 8   | Entity     | TicketPriority      | Entitas yang menyimpan informasi mengenai prioritas dari suatu ticket.  |
| 9   | Entity     | ProductMember       | Entitas yang memberikan role dari employee yang bekerja di suatu product  |
| 10  | Control    | TicketController    | Class control yang menyediakan manipulasi data dan perantara antara entity dan boundary ticketing   |
| 11  | Boundary   | Ticket Create Page  | Page yang menyediakan form pembuatan ticket   |
| 12  | Boundary   | Ticket Details Page | Page yang menampilkan detail ticket serta transaksi ticket berupa verify, reject, done, dan close juga menyediakan form assign developer. |
| 13  | Boundary   | Ticket List Page    | Page yang menampilkan list ticket yang berguna sebagai mulai alur transaksi ticketing.  |
| 14  | Boundary   | Ticket Revise Form  | Form yang menampilkan field alasan untuk usulan revisi ticket.  |
| 15  | Boundary   | Ticket Reject Form  | Form yang menampilkan field alasan untuk menolak ticket.  |
| 16  | Boundary   | Ticket Verify Form  | Form yang menampilkan field dropdown yang digunakan untuk memilih employee yang akan mengerjakan ticket.                                  |

#### 4.3.2. Class Diagram

Setelah mendefinisikan class yang terlibat pada identifikasi class, tahap selanjutnya adalah memodelkan class diagram berupa visualisasi yang terdiri atas entitas, method dan relasi antar class yang dapat dilihat pada gambar ??.



Gambar 4.40: Class Diagram Modul Ticketing

### 4.3. Design

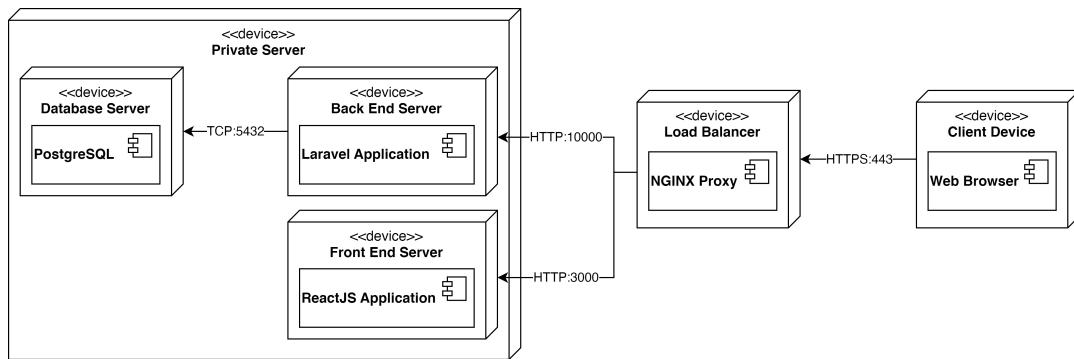
Eksplorasi mendalam mengenai desain arsitektural dari sistem yang sedang dikembangkan akan dilakukan. Untuk memastikan pemahaman yang komprehensif mengenai integrasi dan interaksi antar-komponen dalam aplikasi, dua bentuk diagram utama akan diperkenalkan: Package Diagram dan Deployment Diagram. Package Diagram akan menguraikan struktur organisasi dan modularisasi dari komponen-komponen sistem, menunjukkan pemecahan dan pengelompokan fungsi-fungsi sistem. Deployment Diagram akan menyediakan perspektif terkait dengan distribusi fisik sumber daya komputasi yang digunakan oleh aplikasi serta cara berkomunikasi antar-komponen di lingkungan produksi. Pemaparan ini bertujuan untuk mendukung analisis terhadap aspek skalabilitas, performa, dan keamanan sistem, sehingga memberikan gambaran yang lebih jelas dan menyeluruh tentang kerangka kerja arsitektural yang dipilih.

#### 4.3.1. Package Diagram

Package Diagram digunakan untuk mengilustrasikan organisasi struktural serta hubungan antar komponen yang ada dalam Sistem IT Service Desk di PT Teknologi Aplikasi Sejahtera.

#### 4.3.2. Deployment Diagram

Deployment diagram ini memberikan visualisasi tentang bagaimana aplikasi di-deploy dalam lingkungan produksi, termasuk perangkat keras dan perangkat lunak yang digunakan dalam mendistribusikan dan menjalankan komponen aplikasi. Deployment diagram Sistem Informasi Manajemen Produk dan Ticketing dapat dilihat pada gambar ??.



Gambar 4.41: Deployment Diagram

Diagram diatas menunjukkan bagian bagian utama deployment Sistem Informasi Manajemen Produk dan Ticketing. Penjelasan untuk masing masing komponen akan dijelaskan sebagai berikut:

##### 1. Private Server

Private server merupakan device utama yang digunakan sebagai host dari backend server, frontend server, dan database server. Mesin ini memiliki kemampuan komputasi yang tinggi dengan ram yang cukup besar. Alasan private server melakukan virtualisasi ini adalah untuk memungkinkan isolasi antar service dan dapat mengalokasikan sumber daya dengan efisien, sehingga setiap service yang dibuat server dapat dikelola secara independen dan fleksibel. Penggunaan satu mesin seperti ini dapat memudahkan untuk memanage service yang ada pada satu device. Namun, dengan hanya ada satu mesin saja skalabilitas aplikasi menjadi terbatas.

##### 2. Database Server dan PostgreSQL

Database server perangkat yang dirancang khusus untuk menyediakan layanan database kepada backend server. Database server dipisah dengan backend

dengan tujuan tim pengembang dapat menggunakan database bersama tanpa menghiraukan masa pemeliharaan server backend. Database server beroperasi sebagai virtual machine di bawah private server.

Database server memiliki satu layanan yakni PostgreSQL. PostgreSQL akan digunakan sebagai sumber penyimpanan persistent backend server.

### 3. Backend Server dan Laravel Application

Backend server merupakan perangkat yang akan digunakan sebagai penyedia layanan backend dari Sistem Informasi Manajemen Produk dan Ticketing. Backend server akan lebih berfokus pada manipulasi data yang persisten serta mengatur logika bisnis. Untuk implementasinya, tim pengembang memilih menggunakan PHP dengan framework Laravel sebagai backend.

### 4. Frontend Server dan React.JS Application

Frontend server merupakan perangkat yang akan digunakan sebagai penyedia layanan frontend dari Sistem Informasi Manajemen Produk dan Ticketing. Berbeda dari backend, frontend lebih berfokus kepada bagaimana data disajikan dan ditampilkan. Untuk implementasinya, tim pengembang memilih menggunakan React.JS sebagai framework frontend.

### 5. Load Balancer dengan NGINX Proxy

Load balancer digunakan sebagai penyeimbang beban requests yang masuk. Load balancer akan di-host di layanan cloud yang memiliki IP publik. Selain sebagai penyeimbang beban, mesin ini juga akan digunakan sebagai gateway akses ke bagian private server karena private server tidak memiliki IP publik yang statis. Dengan menggunakan NGINX proxy, mesin dapat meneruskan request sesuai service yang dituju meskipun berada di IP yang sama namun dengan domain yang berbeda.

### 6. PC Client dan Web Browser

PC client merupakan device yang digunakan pengguna untuk mengakses Sistem Informasi Manajemen Produk dan Ticketing. Pengguna akan memasukkan url ke dalam navigasi web browser yang akan melakukan request ke frontend server dan backend server melalui load balancer.

#### 4.3.3. Desain Antarmuka Aplikasi

Berikut merupakan penjelasan rancangan desain antarmuka Sistem Informasi Manajemen Produk dan Ticketing bagian ticketing di PT. Teknologi Aplikasi Sejahtera:

##### 1. Halaman List Ticket

Pada halaman list ticket akan ditampilkan list ticket yang user sedang login memiliki tanggung jawab. Semisal sebagai PM maka seluruh tiket product yang

di assign akan muncul dan jika sebagai external user maka semua ticket yang ia buat akan muncul di sini. Sebagai tambahan user dapat melakukan searching menggunakan kolom search. User juga bisa melakukan filter berdasarkan statusnya. User juga bisa melakukan sorting berdasarkan seberapa baru ticket itu dibuat. Jika user merasa tampilan dalam satu halaman terlalu sedikit atau terlalu banyak, user juga bisa mengatur banyaknya ticket yang ditampilkan dalam satu halaman. Khusus untuk external user, akan muncul tombol new ticket untuk membuat ticket baru. Tampilan mockup ui halaman list ticket dapat dilihat pada gambar ??.

Gambar 4.42: Prototype List Ticket

## 2. Halaman Membuat Ticket

Untuk external user yang ingin membuat ticket, akan diteruskan ke halaman ini, user akan mengisi informasi yang diperlukan untuk membuat ticket dengan field yang wajib adalah project, tingkat prioritas, judul atau nama ticket, serta deskripsi permasalahan. User dapat menambahkan attachment untuk memberikan tambahan informasi jika tidak dapat dijelaskan menggunakan kata kata. Tampilan mockup ui

halaman pembuatan ticket dapat dilihat pada gambar ??.

The screenshot shows a web-based application interface titled 'TAS IT Desk'. On the left, there is a sidebar with three menu items: 'Dashboard', 'Tickets' (which is currently selected and highlighted in grey), and 'Products'. The main content area is titled 'Tickets / Create' and contains a form titled 'Create New Ticket'. The form includes fields for 'Project \*' (a dropdown menu), 'Priority \*' (another dropdown menu), 'Title \*' (a text input field), 'Describe the issue \*' (a large text area), and 'Attachment' (a file upload input field with a 'Select file' button). At the bottom of the form are two buttons: 'Save' and 'Cancel'.

Gambar 4.43: Prototype Create Ticket

### 3. Halaman Detail Ticket

Halaman ini adalah halaman di mana transaksi ticket berlangsung. Fungsionalitas halaman ini akan berubah tergantung role user dan kondisi ticket. Ketika ticket menunggu verifikasi, hanya product manager yang dapat memverifikasi ticket tersebut atau menolak ticket tersebut. Jika product manager memilih untuk memverifikasi ticket tersebut, akan muncul popup untuk memilih developer mana yang akan mengerjakan ticket tersebut. Developer yang mengerjakan ticket tersebut dapat melaporkan pekerjaannya dengan menekan tombol done. Setelah semua developer sudah OK, product manager dapat melakukan pengecekan apakah pekerjaan sudah selesai dan benar. Jika belum progress pekerjaan developer akan direset ulang. Jika sudah, ticket akan ke tahap selanjutnya untuk divalidasi dari sisi client atau pembuat ticket. Jika tidak sesuai maka akan dikembalikan ke fase penggerjaan, jika sesuai ticket akan menjadi closed.

Tampilan mockup ui halaman detail ticket dapat dilihat pada gambar ??.

The mockup shows the 'TAS IT Desk' application interface. On the left is a sidebar with 'Dashboard', 'Tickets' (which is selected), and 'Products'. The main area is titled 'Tickets / Details' and shows a ticket for 'myHolic / BRJHLC002' with the subject 'Back button still missing'. A detailed description discusses the absence of a back button and its impact on user frustration. To the right of the ticket details are four action buttons: 'Done' (with a checkmark icon), 'Verify Ticket' (with a checkmark icon), 'Reject Ticket' (with a cross icon), and 'Chat on WhatsApp' (with a WhatsApp icon). Below these buttons is a summary of ticket metadata: Project: myHolic, Company: PT Burjo Holic Kuliner, Created by: Givandra, Status: In Progress, Priority: Normal Priority, Created: 3 January 2024 21:53:17, and Modified: 5 January 2024 10:25:33. At the bottom, there is a section for 'Assigned Developers Status' with three entries: Ridwan Kurniawan (Not OK), Sean Gonzalez (OK), and Asep Panji Lesmana (OK).

Gambar 4.44: Prototype Ticket Details

#### 4. Popup Verify Ticket

Popup ini digunakan untuk memverifikasi ticket yang masuk ke list ticket product manager. Sebelum bisa memverifikasi ticket, product manager perlu mengassign developer yang akan mengerjakan ticket tersebut terlebih dahulu. Dropdown dari assign developer akan dibuat sebagai dropdown yang bisa dilakukan search. Ketika memilih developer product manager dapat memilih lagi dan yang sudah dipilih akan menjadi list yang ditunjukan didalam input box.

## Verify Ticket

Assign Developers \*

|  | + Developer   |
|--|---|
|  Ridwan Kurniawan   |  |
|  Sean Gonzalez      |  |
|  Asep Panji Lesmana |  |

Cancel

Save

Gambar 4.45: Prototype Popup Verify Ticket

### 5. Popup Reject Ticket

Popup ini digunakan untuk mereject ticket yang masuk ke list ticket product manager. Sebelum menolak ticket product manager akan memberikan alasan kenapa ticket tidak layak untuk dikerjakan. Product manager akan menjelaskan dalam bentuk teks yang disediakan dalam bentuk input text.

## Reject Ticket

Reason \*

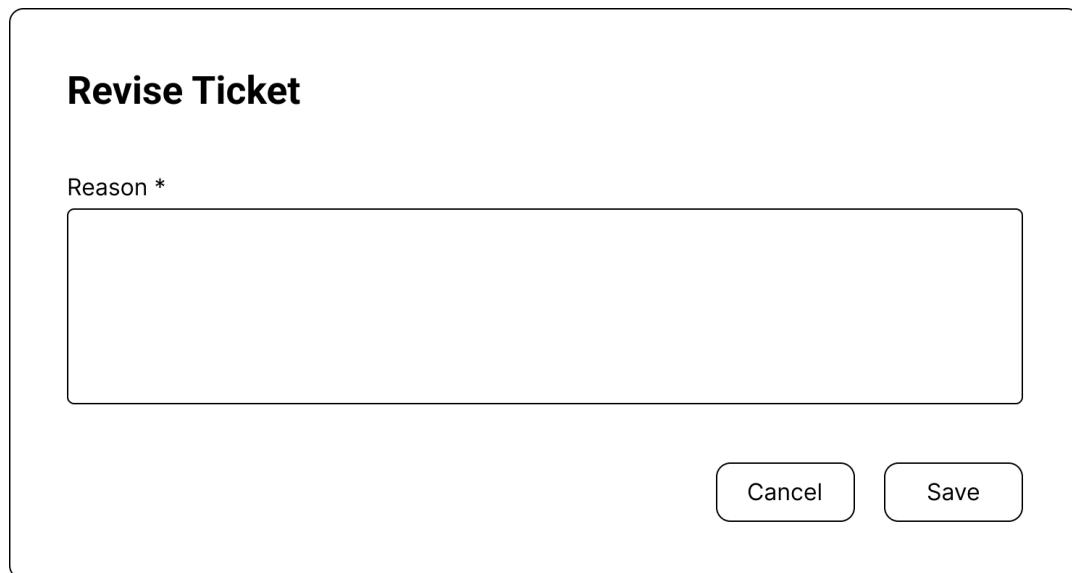
Cancel

Save

Gambar 4.46: Prototype Popup Reject Ticket

## 6. Popup Revise Ticket

Popup ini digunakan untuk merevisi ticket yang masuk ke list ticket external user dan product manager. Sebelum merevisi ticket pengguna akan memberikan alasan kenapa ticket masih kurang. Pengguna akan menjelaskan dalam bentuk teks yang disediakan dalam bentuk input text.



Gambar 4.47: Prototype Popup Revise Ticket

## **BAB V**

### **HASIL DAN PEMBAHASAN**

#### **5.1. Implementasi Sistem**

Spesifikasi perangkat keras dan lunak yang digunakan dalam mengembangkan Sistem Informasi Manajemen Produk dan Ticketing bagian ticketing di PT. Teknologi Aplikasi Sejahtera.

Tabel 5.1: Informasi Hardware pengembangan

|                            |   |
|----------------------------|---|
| CPU                        | Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz              |
| Sistem Operasi             | Microsoft Windows 10 Home Single Language build 19045 |
| Bahasa pemrograman         | PHP dan Javascript                                    |
| Framework frontend         | React.js  |
| Framework backend          | Laravel   |
| Database Management System | PostgreSQL  |
| IDE                        | Visual Studio Code                                    |
| Web browser                | Google Chrome   |

#### **5.1.1. Implementasi Back End**

Berikut adalah sebagian fungsi yang diimplementasikan di backend menggunakan PHP dengan framework Laravel. Pengimplementasian kode menggunakan gaya campuran antara ORM Eloquent dan query builder.

##### 1. Create Ticket

Create ticket merupakan fungsi yang tujuannya membuat ticket. Langkah pertama dari create ticket adalah validasi input. Input yang diambil lewat request dipastikan sudah memenuhi standar. Setelah divalidasi akan diambil product manager dari product dari ticket yang bersangkutan melalui ProductMember. Setelah mendapatkan data product manager, kita akan membuat Ticket, kemudian dua TicketMember awal yakni Product Manager dan PIC. Setelah itu ticket yang baru dibuat akan dikirim sebagai return value.

---

```
public function create(Request $request) {  
    $user = auth('api')->user();  
    $employee = $user->employee;  
    $request->validate([
```

```

        'name' => 'required',
        'attachment' => 'required',
        'content' => 'required',
        'product_id' => 'required',
        'ticket_category_id' => 'required',
        'ticket_priority_id' => 'required',
        'ticket_status_id' => 'required',
    ]);

$request->merge(['created_by' => $user->id]);
$request->merge(['updated_by' => $user->id]);

$raw = HomebrewQueryBuilder::select(
    $request, ProductMember::class,
    [], [], [
        "product_id = $request->product_id"
    ]
);

$res = collect(DB::select($raw['query']));
$pm = $res->firstWhere('product_member_role_id', 1);
$issuer = $res->firstWhere('employee_id', $employee->id);
$ticket = Ticket::factory()->create($request->all());

$ticket_member = TicketMember::factory()->create([
    'ticket_id' => $ticket->id,
    'product_member_id' => $pm->product_member_id,
]);
$ticket_member = TicketMember::factory()->create([
    'ticket_id' => $ticket->id,
    'product_member_id' => $issuer->product_member_id,
]);


---


return json_encode([
    'ticket' => $ticket,
]);
}

```

## 2. List Ticket

List ticket merupakan fungsi untuk mendapat list ticket yang ditugaskan ke sebuah user. Fungsi ini menyediakan fitur list sekaligus filter yang digunakan oleh user. List bekerja dengan mendapatkan id employee user terlebih dahulu. Employee id ini akan digunakan sebagai filter tambahan untuk mendapatkan product member user. Setelah mendapatkan semua product member user, kita akan mengambil semua data ticket yang dikerjakan oleh product member dengan melakukan query ticket member dengan tambahan data yang diperlukan dengan filter product member yang sudah kita dapatkan sebelumnya

---

```
public function get_ticket_list_by_logged_in_user(Request $request) {
    $user = auth('api')->user();
    $user_id = $user->id;
    $employee_id = $user->employee->id;

    $productMembers = ProductMember::where('employee_id',
                                            $employee_id)->get();
    $product_member_ids = $productMembers
        ->pluck('id')->toArray();
    $product_member_id_commas = implode(", ",
                                         $product_member_ids);

    $additional_filter = [];

    if ($request->filled('status'))
        array_push($additional_filter,
                   "ticket_statuses.name = '$request->status'");
    if ($request->filled('category'))
        array_push($additional_filter,
                   "ticket_categories.name = '$request->category');

    $raw = HomebrewQueryBuilder::select(
        $request,
        TicketMember::class,
        [
            ],
            [], array_merge([
                "ticket_members.product_member_id
                IN ($product_member_id_commas)"
```

```

        ], $additional_filter)
    );

    // Do the query
    $res = DB::select($raw['query']);
    $count = DB::select($raw['query_count'])[0]->count;

    return [
        'data' => $res,
        'total' => $count,
        'limit' => $raw['limit'],
        'page' => $raw['page'],
        'totalPage' => ceil($count / $raw['limit']),
        'data' => $res
    ];
}

}

```

---

### 3. Show Ticket

Show ticket merupakan fungsi yang digunakan untuk mendapatkan detail sebuah ticket. Fungsi ini berawal dari mendapatkan employee id user kemudian melakukan query data yang diperlukan untuk ticket. Setelah melakukan query kita perlu merapihkan data karena hasil query bentuknya adalah tabel yang memiliki kolom duplikat karena hasil join dengan ticket member, kita akan mengambil data yang duplikat dengan mendefinisikan kolom kolomnya pada variabel `$common`. Variabel ini digunakan untuk melakukan filter kolom yang duplikat. Setelah mendapatkan data duplikat ticket, kita akan mengambil data ticket member dengan mendefinisikan `$ticket_member_columns` yang akan digunakan untuk memfilter data ticket member. Dua data dari data duplikat dan data ticket member akan digabungkan dan dikembalikan ke sisi frontend.

---

```

public function show(Request $request, $id) {
    $user = auth('api')->user();
    $employee = $user->employee;
    $employee_id = $employee->id;

    $raw = HomebrewQueryBuilder::select(

```

```

$request,
Ticket::class,
[
    'employee_name' => 'employees.name',
    'created_by_employee_name' => 'c_employees.name',
    'created_by_user_id' => 'u_created.id',
    'updated_by_employee_name' => 'u_employees.name',
    'updated_by_user_id' => 'u_updated.id',
    'employee_id' => 'employees.id',
    'employee_phone' => 'employees.phone',
    'ticket_member_status' => 'ticket_members.status'
], [
    'employees' => [
        JOIN::L_TABLE => 'product_members',
        JOIN::L_FKEY => 'employee_id',
    ],
    'users u_created' => [
        JOIN::L_FKEY => 'created_by',
    ],
    'users u_updated' => [
        JOIN::L_FKEY => 'updated_by',
    ],
    'employees c_employees' => [
        JOIN::L_TABLE => 'u_created',
        JOIN::L_FKEY => 'employee_id',
    ],
    'employees u_employees' => [
        JOIN::L_TABLE => 'u_updated',
        JOIN::L_FKEY => 'employee_id',
    ]
], [
    "tickets.id = $id",
]
);

// Do the query
$res = DB::select($raw['query']);

```

```

$common = [
    'ticket_id', 'ticket_name', 'ticket_content',
    'ticket_description', 'ticket_attachment',
    'ticket_status_id', 'ticket_status',
    'ticket_status_hex', 'ticket_category_id',
    'ticket_category', 'ticket_category_hex',
    'ticket_priority_id', 'ticket_priority',
    'ticket_priority_hex', 'product_id', 'product_name',
    'company_id', 'company_name', 'created_by_employee_name',
    'created_by_user_id', 'updated_by_employee_name',
    'updated_by_user_id', 'created_at',
];

// sampling base ticket columns
$base = array_filter((array) $res[0], function($v)
    use ($common) {
        return in_array($v, $common);
}, ARRAY_FILTER_USE_KEY);

$ticket_member_info = array_map(function ($item) {
    $ticket_member_columns = [
        'product_member_role_id',
        'product_member_role_name',
        'employee_id',
        'employee_name',
        'employee_phone',
        'ticket_member_status',
    ];

    return array_filter((array) $item, function($v)
        use ($ticket_member_columns) {
            return in_array($v, $ticket_member_columns);
}, ARRAY_FILTER_USE_KEY);
}, $res);

return [
    "query" => $raw['query'],
    "message" => "Success",
    "data" => $base
];

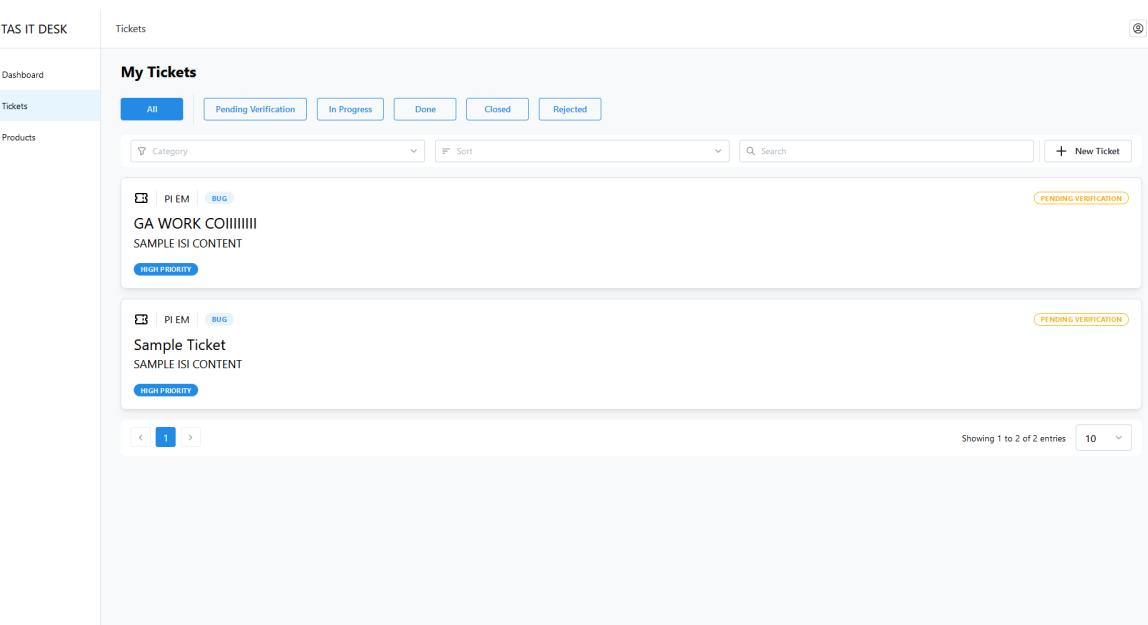
```

```
] ;  
}  
-----
```

## 5.1.2. Implementasi Front End

### 5.1.1.1. Antarmuka List Ticket

Halaman list ticket merupakan halaman utama dari modul ticketing di Sistem Informasi Manajemen Produk dan Ticketing. Pada halaman ini akan ditampilkan informasi mengenai ticket yang user tersebut memiliki tanggung jawab. Di halaman ini user dapat diberikan opsi untuk mencari dan memfilter kategori tiket yang ingin dicari. Sistem memberikan opsi untuk memfilter berdasarkan kategori dan status. User juga dapat mengurutkan berdasarkan kronologi pembuatan dan modifikasi. Pada halaman ini jika user merupakan seorang external user, sistem akan menampilkan tombol untuk membuat ticket baru yang akan diarahkan ke halaman membuat ticket. Desain antarmuka list ticket dapat dilihat pada gambar ??.



Gambar 5.1: Antarmuka List Ticket

### 5.1.1.2. Antarmuka Membuat Ticket

Halaman pembuatan ticket merupakan halaman yang berasal dari halaman list ticket. Halaman ini hanya dapat diakses oleh seorang external user. Pada halaman ini user akan diberikan form untuk mengisikan informasi yang diperlukan untuk membuat ticket seperti product apa, prioritas dari

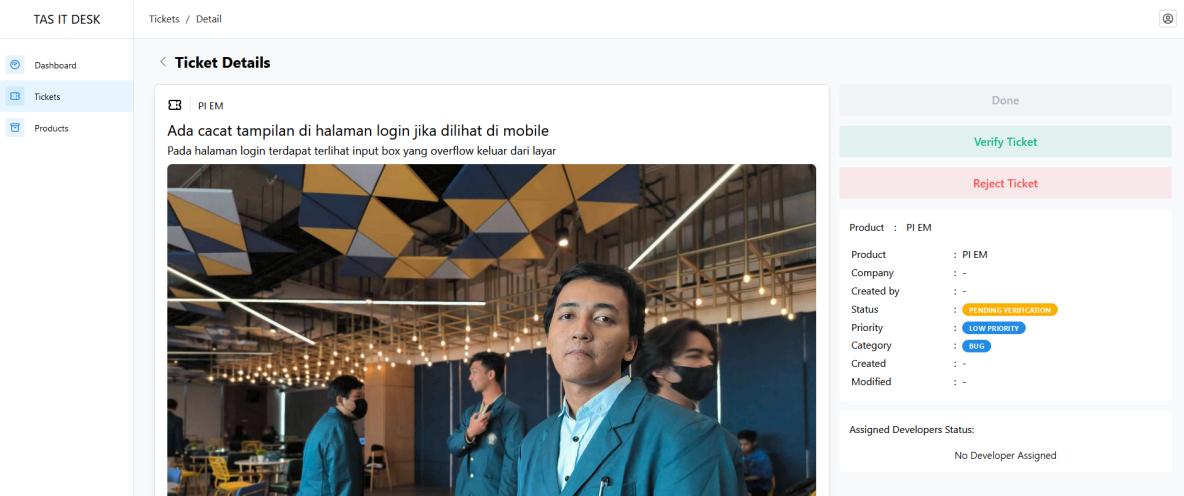
ticket tersebut, kategori ticket, judul dari ticket, serta deskripsi kenapa ticket tersebut perlu dibuat.

The screenshot shows the 'TAS IT DESK' application. On the left is a sidebar with 'Dashboard', 'Tickets' (which is selected and highlighted in blue), and 'Products'. The main area is titled 'Tickets / Create' and has a sub-section 'Create Ticket'. It contains fields for 'Project \*' (set to 'PI EM'), 'Priority \*' (set to 'Low Priority'), 'Ticket Category \*' (set to 'BUG'), 'Title \*' (containing the text 'Ada cacat tampilan di halaman login jika dilihat di mobile'), and a 'Describe the issue \*' text area (containing the text 'Pada halaman login terdapat terdapat input box yang overflow keluar dari layar'). At the bottom are two buttons: a green 'Create' button and a red 'Cancel' button.

Gambar 5.2: Antarmuka Membuat Ticket

### 5.1.1.3. Antarmuka Detail Ticket

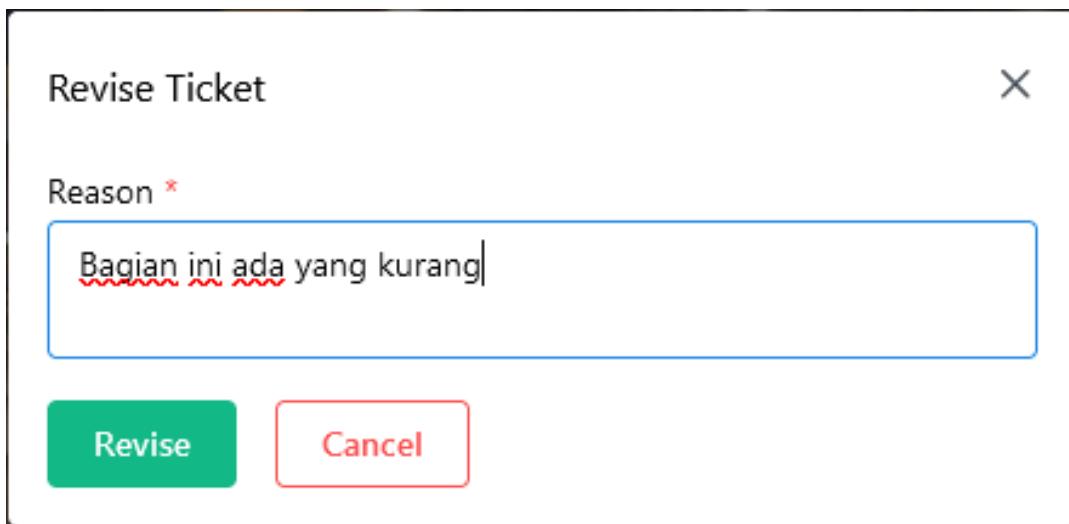
Halaman detail ticket memberikan informasi keseluruhan ticket tersebut. Informasi yang disampaikan berupa deskripsi ticket, judul ticket, status ticket, prioritas ticket, kategori ticket, dari product apa, dari company apa, siapa yang membuat, kapan dibuat, kapan ada aktivitas terbaru di ticket tersebut, siapa yang mengerjakan, serta status penggerjaannya. User dengan role external user dapat merevisi dan menutup ticket yang ia buat. User dengan role product manager dapat memverifikasi atau mereject ticket yang baru dibuat, setelah dikerjakan product manager juga dapat memvalidasi pekerjaan developer dengan mengubah status menjadi done atau merevisi ulang. User dengan role developer hanya dapat mengubah status penggerjaan menjadi done untuk tugasnya.



Gambar 5.3: Antarmuka Detail Ticket

#### 5.1.1.4. Antarmuka Form Revisi Ticket

Antarmuka form revisi merupakan tampilan popup yang keluar ketika external user menerima ticket dengan status done. Ketika external user merasa hasil yang sudah dikerjakan masih belum memenuhi maka external user dapat merevisi ticket dengan mengisi form alasan merevisi ticket.

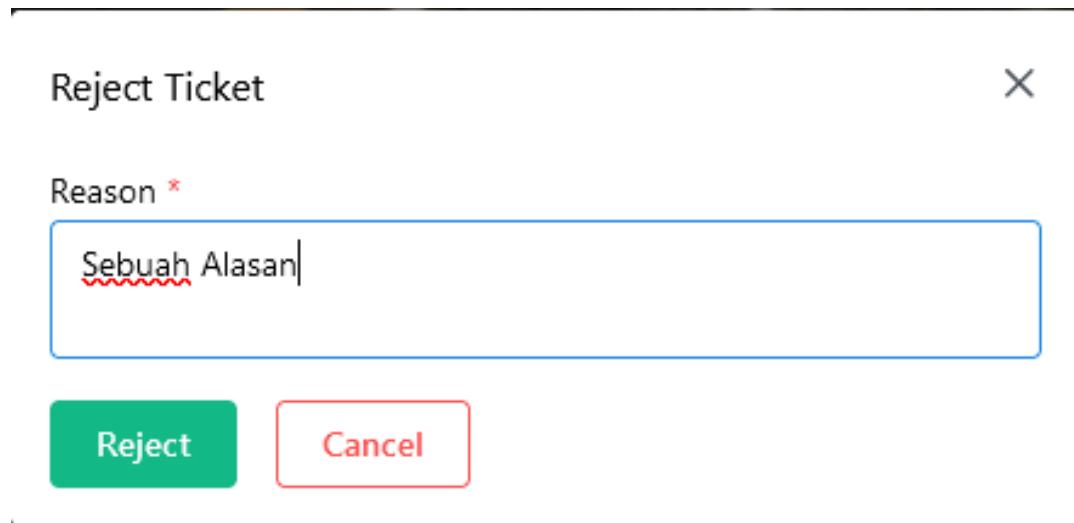


Gambar 5.4: Antarmuka Popup Revise Ticket

#### 5.1.1.5. Antarmuka Form Reject Ticket

Antarmuka form revisi merupakan tampilan popup yang keluar ketika product manager menerima ticket dengan status pending verification. Ketika product manager merasa ticket yang diajukan masih terlalu umum atau

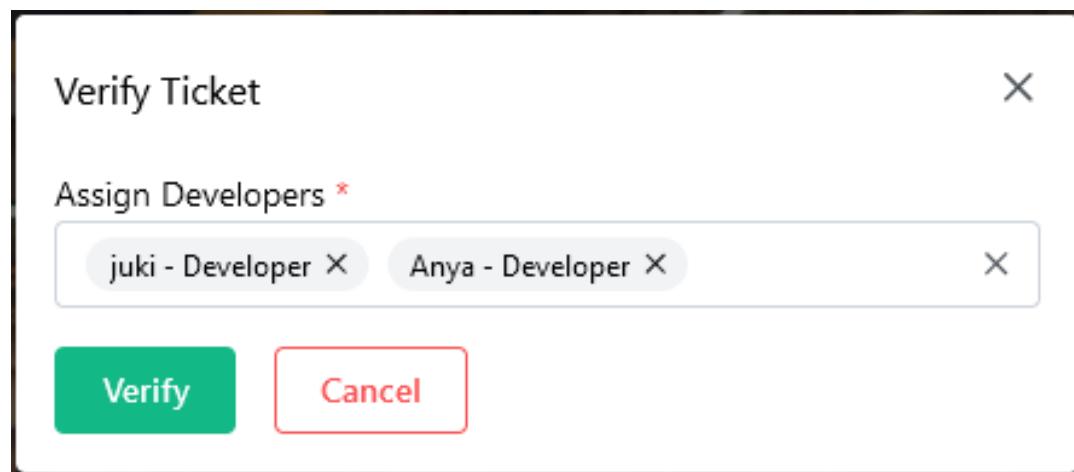
tidak cukup jelas, maka product manager bisa menolak dan memberikan masukkan untuk external user untuk membuat ticket baru atau memberitahu permasalahan sudah diluar dari kontrak.



Gambar 5.5: Antarmuka Popup Reject Ticket

#### 5.1.1.6. Antarmuka Form Verify Ticket

Antarmuka form revisi merupakan tampilan popup yang keluar ketika product manager menerima ticket dengan status pending verification. Sebelum product manager bisa memverifikasi ticket, product manager perlu mengalokasikan employee yang akan mengerjakan ticket tersebut.



Gambar 5.6: Antarmuka Popup Verify Ticket

## 5.2. Pengujian Sistem

### **5.2.1. Metode Pengujian**

Prosedur Pengujian dirancang untuk mengatur persiapan pengujian dengan cara yang terstruktur dan sistematis, memudahkan pelaksanaan pengujian secara efektif. Langkah-langkah dalam prosedur pengujian meliputi:

1. Menyiapkan perangkat lunak yang akan diuji serta perangkat keras yang dibutuhkan untuk menjalankan pengujian.
2. Menentukan hasil yang diharapkan untuk setiap item pengujian berdasarkan skenario yang telah direncanakan dalam dokumen rencana pengujian.
3. Menetapkan kriteria untuk mengukur keberhasilan pengujian.
4. Membuat kasus pengujian dan mendokumentasikan hasilnya.
5. Mengidentifikasi dan mendokumentasikan segala cacat atau kesalahan yang ditemukan selama pengujian.
6. Mengimplementasikan perbaikan atas cacat yang ditemukan.

### **5.2.2. Rencana Pengujian Sistem**

Pengujian perangkat lunak dilaksanakan untuk memverifikasi kesesuaian sistem dengan kebutuhan fungsional yang ditetapkan dan untuk memastikan sistem beroperasi seperti yang diharapkan. Pengujian ini akan menggunakan metode black box, yang merupakan teknik pengujian dimana sistem diuji dengan memasukkan data dan mengamati outputnya tanpa mempertimbangkan proses internal sistem. Proses ini bertujuan untuk menilai apakah output sesuai dengan kebutuhan fungsional yang diharapkan. Detail dari rencana pengujian ini disajikan dalam ??

Tabel 5.2: Rencana Pengujian Sistem

| Aktor | Komponen Pengujian    | Butir Uji  |
|-------|-----------------------|--|
| PM    | Melihat list ticket   | <ol style="list-style-type: none"><li>1. Pengguna dapat melihat list ticket yang merupakan tanggung jawabnya</li></ol> |
|       | Memfilter list ticket | <ol style="list-style-type: none"><li>1. Pengguna dapat memfilter list tiket berdasarkan status tiket</li></ol>        |

| Aktor | Komponen Pengujian   | Butir Uji  |
|-------|----------------------|--|
|       | Melihat detail tiket | <ol style="list-style-type: none"> <li>Pengguna dapat melihat detail tiket yang sudah ada</li> </ol>   |
|       | Memverifikasi ticket | <ol style="list-style-type: none"> <li>Pengguna dapat mem-verify tiket dengan status pending verification</li> <li>Pengguna berusaha mem-verify tiket tanpa mengassign developer</li> <li>Pengguna tidak dapat mem-verify tiket dengan status selain pending verification</li> </ol> |
|       | Menolak ticket       | <ol style="list-style-type: none"> <li>Pengguna dapat menolak tiket dengan status pending verification</li> <li>Pengguna tidak mengisi alasan penolakan</li> <li>Pengguna tidak dapat menolak tiket dengan status selain pending verification</li> </ol>                             |
|       | Revisi ticket        | <ol style="list-style-type: none"> <li>Pengguna dapat mengusulkan revisi tiket dengan status done</li> <li>Pengguna tidak mengisi alasan revisi</li> <li>Pengguna tidak dapat mengusulkan revisi tiket dengan status selain done</li> </ol>  |
|       | Komunikasi ticket    | <ol style="list-style-type: none"> <li>Pengguna dapat berkomunikasi dengan pihak eksternal (pembuat tiket) melalui whatsapp</li> </ol>   |

| Aktor | Komponen Pengujian     | Butir Uji   |
|-------|------------------------|---|
|       | Done ticket            | <ol style="list-style-type: none"> <li>1. Pengguna dapat mengganti status ticket menjadi done dengan status in progress</li> <li>2. Pengguna tidak dapat mengganti status ticket menjadi done dengan status selain in progress</li> </ol> |
| DEV   | Melihat list ticket    | <ol style="list-style-type: none"> <li>1. Pengguna dapat melihat list ticket yang merupakan tanggung jawabnya</li> </ol>  |
|       | Memfilter list ticket  | <ol style="list-style-type: none"> <li>1. Pengguna dapat memfilter list tiket berdasarkan status tiket</li> </ol>   |
|       | Melihat detail tiket   | <ol style="list-style-type: none"> <li>1. Pengguna dapat melihat detail tiket yang sudah ada</li> </ol>   |
|       | Melaporkan Penggerjaan | <ol style="list-style-type: none"> <li>1. Pengguna dapat melaporkan penggerjaan ketika status in progress</li> <li>2. Pengguna tidak dapat melaporkan penggerjaan ketika status selain in progress</li> </ol>                             |
| PIC   | Melihat list ticket    | <ol style="list-style-type: none"> <li>1. Pengguna dapat melihat list ticket yang merupakan tanggung jawabnya</li> </ol>  |
|       | Memfilter list ticket  | <ol style="list-style-type: none"> <li>1. Pengguna dapat memfilter list tiket berdasarkan status tiket</li> </ol>   |
|       | Membuat ticket         | <ol style="list-style-type: none"> <li>1. Pengguna dapat membuat ticket</li> <li>2. Pengguna tidak mengisi beberapa field</li> </ol>  |

| Aktor | Komponen Pengujian   | Butir Uji   |
|-------|----------------------|---|
|       | Melihat detail tiket | <ol style="list-style-type: none"> <li>Pengguna dapat melihat detail tiket yang sudah ada</li> </ol>  |
|       | Menutup ticket       | <ol style="list-style-type: none"> <li>Pengguna dapat menutup tiket dengan status pending verification</li> <li>Pengguna tidak dapat menutup tiket dengan status selain pending verification</li> </ol>                                     |
|       | Revisi ticket        | <ol style="list-style-type: none"> <li>Pengguna dapat mengusulkan revisi tiket dengan status done</li> <li>Pengguna tidak mengisi alasan revisi</li> <li>Pengguna tidak dapat mengusulkan revisi tiket dengan status selain done</li> </ol> |
|       | Komunikasi ticket    | <ol style="list-style-type: none"> <li>Pengguna dapat berkomunikasi dengan pihak internal (PM) melalui whatsapp</li> </ol>  |

### 5.2.3. Pelaksanaan Pengujian Sistem

Pengujian untuk Sistem Manajemen Produk dan Ticketing Modul Ticketing di PT Teknologi Aplikasi Sejahtera dijalankan berdasarkan rancangan yang telah diuraikan dalam tabel ???. Hasil dari pengujian ini dapat dilihat pada tabel ?? yang disajikan di bawah ini.

Tabel 5.3: Pelaksanaan Pengujian

| Identifikasi  | Butir Uji   | Prosedur pengujian  | Masukan       | Hasil yang diharapkan  | Hasil yang diperoleh  | Kesimpulan |
|---------------|---|---|---------------|--|---|------------|
| SRS-TKT-PM-01 | Pengguna dapat melihat list ticket tanggung jawabnya          | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Halaman ticket ditampilkan</li> </ol>   |               | Sistem menampilkan list ticket tanggung jawab user yang login  | Ditampilkan list ticket tanggung jawab user   | diterima   |
| SRS-TKT-PM-02 | Pengguna dapat memfilter list ticket berdasarkan status tiket | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih filter ticket</li> <li>4. Halaman ticket yang telah difilter ditampilkan</li> </ol> | status ticket | Sistem menampilkan list ticket tanggung jawab user yang difilter berdasarkan filter status yang dipilih user | Ditampilkan list ticket tanggung jawab user yang difilter berdasarkan filter status yang dipilih user | diterima   |
| SRS-TKT-PM-03 | Pengguna dapat melihat detail tiket yang sudah ada            | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket</li> <li>4. Halaman detail ticket ditampilkan</li> </ol>                     |               | Sistem menampilkan detail ticket yang dipilih user   | Ditampilkan detail ticket yang dipilih user   | diterima   |

| <b>Identifikasi</b> | <b>Butir Uji</b>   | <b>Prosedur pengujian</b>  | <b>Masukan</b> | <b>Hasil yang diharapkan</b>   | <b>Hasil yang diperoleh</b>   | <b>Kesimpulan</b> |
|---------------------|--|--|----------------|--|---|-------------------|
| SRS-TKT-PM-04       | Pengguna dapat mem-verify tiket dengan status pending verification | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket dengan status pending verification</li> <li>4. Menekan tombol verify</li> <li>5. Popup assign developer ditampilkan</li> <li>6. Memilih developer</li> <li>7. Menekan tombol verify</li> <li>8. Halaman detail ticket diperbarui menjadi in progress</li> </ol> | product member | Sistem merubah status ticket menjadi in progress dan developer diassign ke ticket tersebut | Ticket status berubah menjadi in progress dan developer yang dipilih terassign ke ticket tersebut | diterima          |
| SRS-TKT-PM-04       | Pengguna berusaha mem-verify tiket tanpa mengassign developer      | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket dengan status pending verification</li> <li>4. Menekan tombol verify</li> <li>5. Popup assign developer ditampilkan</li> <li>6. Menekan tombol verify tanpa memilih developer</li> </ol>  |                | Sistem tidak memperbolehkan verifikasi ticket  | Pengguna tidak dapat memverifikasi tiket dengan tombol verify tidak dapat ditekan                 | diterima          |

| Identifikasi  | Butir Uji   | Prosedur pengujian  | Masukan       | Hasil yang diharapkan   | Hasil yang diperoleh  | Kesimpulan |
|---------------|---|---|---------------|---|---|------------|
| SRS-TKT-PM-04 | Pengguna tidak dapat mem-verify tiket dengan status selain pending verification | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket dengan status selain pending verification</li> <li>4. Menekan tombol verify</li> </ol>   |               | Sistem tidak memperbolehkan verifikasi ticket                               | Pengguna tidak dapat memverifikasi tiket dengan tombol verify tidak dapat ditemukan | diterima   |
| SRS-TKT-PM-05 | Pengguna dapat menolak tiket dengan status pending verification                 | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket dengan status pending verification</li> <li>4. Menekan tombol reject</li> <li>5. Popup reject ticket ditampilkan</li> <li>6. Mengisi alasan menolak</li> <li>7. Menekan tombol reject</li> <li>8. Halaman detail ticket diperbarui menjadi rejected</li> </ol> | reject reason | Sistem merubah status ticket menjadi rejected dan alasan reject ditampilkan | Ticket status berubah menjadi in progress dan alasan reject ditampilkan             | diterima   |

| <b>Identifikasi</b> | <b>Butir Uji</b>   | <b>Prosedur pengujian</b>  | <b>Masukan</b> | <b>Hasil yang diharapkan</b>                 | <b>Hasil yang diperoleh</b>   | <b>Kesimpulan</b> |
|---------------------|--|--|----------------|--|---|-------------------|
| SRS-TKT-PM-05       | Pengguna tidak mengisi alasan penolakan                                      | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket dengan status selain pending verification</li> <li>4. Menekan tombol reject</li> <li>5. Popup reject ticket ditampilkan</li> <li>6. Menekan tombol reject tanpa mengisi alasan menolak</li> </ol> |                | Sistem tidak memperbolehkan penolakan ticket | Pengguna tidak dapat menolak tiket dengan tombol reject tidak dapat ditekan   | diterima          |
| SRS-TKT-PM-05       | Pengguna tidak dapat menolak tiket dengan status selain pending verification | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket dengan status selain pending verification</li> <li>4. Menekan tombol reject</li> </ol>  |                | Sistem tidak memperbolehkan penolakan ticket | Pengguna tidak dapat menolak tiket dengan tombol reject tidak dapat ditemukan | diterima          |

| <b>Identifikasi</b> | <b>Butir Uji</b> | <b>Prosedur pengujian</b>                        | <b>Masukan</b>  | <b>Hasil yang diharapkan</b> | <b>Hasil yang diperoleh</b>  | <b>Kesimpulan</b>  |          |
|---------------------|------------------|--|---|------------------------------|--|--|----------|
|                     | SRS-TKT-PM-06    | Pengguna dapat merevisi tiket dengan status done | 1. Mengakses Sistem<br>2. Membuka halaman ticket<br>3. Memilih ticket dengan status done<br>4. Menekan tombol revise<br>5. Popup revise ticket ditampilkan<br>6. Mengisi alasan merevisi<br>7. Menekan tombol revise        | revise reason                | Sistem merubah status ticket menjadi in progress dan alasan revise ditampilkan | Ticket status berubah menjadi in progress dan alasan revise ditampilkan      | diterima |
| 101                 | SRS-TKT-PM-06    | Pengguna tidak mengisi alasan merevisi           | 1. Mengakses Sistem<br>2. Membuka halaman ticket<br>3. Memilih ticket dengan status selain done<br>4. Menekan tombol revise<br>5. Popup revise ticket ditampilkan<br>6. Menekan tombol revise tanpa mengisi alasan merevisi |                              | Sistem tidak memperbolehkan merevisi ticket                                    | Pengguna tidak dapat merevisi tiket dengan tombol revise tidak dapat ditekan | diterima |

| <b>Identifikasi</b> | <b>Butir Uji</b>  | <b>Prosedur pengujian</b>   | <b>Masukan</b> | <b>Hasil yang diharapkan</b>                                    | <b>Hasil yang diperoleh</b>  | <b>Kesimpulan</b> |
|---------------------|---|---|----------------|---|--|-------------------|
| SRS-TKT-PM-06       | Pengguna tidak dapat merevisi tiket dengan status selain done                         | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket dengan status selain done</li> <li>4. Menekan tombol revise</li> </ol>   |                | Sistem tidak memperbolehkan merevisi ticket                     | Pengguna tidak dapat merevisi tiket dengan tombol revise tidak dapat ditemukan | diterima          |
| SRS-TKT-PM-07       | Pengguna dapat berkomunikasi dengan pihak eksternal (pembuat ticket) melalui whatsapp | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket</li> <li>4. Menekan Chat on WhatsApp</li> <li>5. Pengguna akan diarahkan ke whatsapp dengan template message</li> </ol>        |                | Sistem mengarahkan pengguna ke whatsapp dengan template message | Pengguna diarahkan ke whatsapp dengan template message                         | diterima          |
| SRS-TKT-PM-08       | Pengguna dapat mengganti status ticket menjadi done dengan status in progress         | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket dengan status in progress</li> <li>4. Menekan tombol done</li> <li>5. Halaman detail ticket diperbarui menjadi done</li> </ol> |                | Sistem merubah status ticket menjadi done                       | Ticket status berubah menjadi done   | diterima          |

| <b>Identifikasi</b> | <b>Butir Uji</b> | <b>Prosedur pengujian</b>  | <b>Masukan</b>  | <b>Hasil yang diharapkan</b>                                     | <b>Hasil yang diperoleh</b>  | <b>Kesimpulan</b> |
|---------------------|------------------|--|---|--|--|-------------------|
|                     | SRS-TKT-PM-08    | Pengguna tidak dapat mengganti status ticket menjadi done dengan status selain in progress | 1. Mengakses Sistem<br>2. Membuka halaman ticket<br>3. Memilih ticket dengan status selain in progress<br>4. Menekan tombol done  | Sistem tidak memperbolehkan mengganti status ticket menjadi done | Pengguna tidak dapat mengganti status tiket menjadi done dengan tombol done tidak dapat ditemukan            | diterima          |
| 103                 | SRS-TKT-DEV-01   | Pengguna dapat melihat list ticket tanggung jawabnya                                       | 1. Mengakses Sistem<br>2. Membuka halaman ticket<br>3. Halaman ticket ditampilkan   | Sistem menampilkan list ticket tanggung jawab user yang login    | Ditampilkan list ticket tanggung jawab user  | diterima          |
|                     | SRS-TKT-DEV-02   | Pengguna dapat memfilter list ticket berdasarkan status tiket                              | 1. Mengakses Sistem<br>2. Membuka halaman ticket<br>3. Memilih filter ticket<br>4. Halaman ticket yang telah difilter ditampilkan | status ticket  | Sistem menampilkan list ticket tanggung jawab user yang difilter berdasarkan filter status yang dipilih user | diterima          |
|                     | SRS-TKT-DEV-03   | Pengguna dapat melihat detail tiket yang sudah ada   | 1. Mengakses Sistem<br>2. Membuka halaman ticket<br>3. Memilih ticket<br>4. Halaman detail ticket ditampilkan                     |  | Sistem menampilkan detail ticket yang dipilih user   | diterima          |

| <b>Identifikasi</b> | <b>Butir Uji</b>   | <b>Prosedur pengujian</b>   | <b>Masukan</b> | <b>Hasil yang diharapkan</b>                                  | <b>Hasil yang diperoleh</b>  | <b>Kesimpulan</b> |
|---------------------|--|---|----------------|---|--|-------------------|
| SRS-TKT-DEV-04      | Pengguna dapat melaporkan penggerjaan ketika status in progress              | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket dengan status in progress</li> <li>4. Memilih tombol done</li> <li>5. Halaman detail ticket diperbarui dengan status penggerjaan menjadi done</li> </ol> |                | Sistem mengubah status penggerjaan pengguna menjadi done      | Status penggerjaan ticket pengguna menjadi done  | diterima          |
| SRS-TKT-DEV-04      | Pengguna tidak dapat melaporkan penggerjaan ketika status selain in progress | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket dengan status in progress</li> <li>4. Memilih tombol done</li> </ol>   |                | Sistem melarang pelaporan penggerjaan pengguna                | Pengguna tidak dapat mengganti melaporkan penggerjaan karena tombol done tidak dapat ditemukan | diterima          |
| SRS-TKT-PIC-01      | Pengguna dapat melihat list ticket tanggung jawabnya                         | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Halaman ticket ditampilkan</li> </ol>   |                | Sistem menampilkan list ticket tanggung jawab user yang login | Ditampilkan list ticket tanggung jawab user  | diterima          |

| <b>Identifikasi</b> | <b>Butir Uji</b>  | <b>Prosedur pengujian</b>   | <b>Masukan</b>  | <b>Hasil yang diharapkan</b>   | <b>Hasil yang diperoleh</b>   | <b>Kesimpulan</b> |
|---------------------|---|---|---|--|---|-------------------|
| SRS-TKT-PIC-02      | Pengguna dapat memfilter list ticket berdasarkan status tiket | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih filter ticket</li> <li>4. Halaman ticket yang telah difilter ditampilkan</li> </ol>   | status ticket   | Sistem menampilkan list ticket tanggung jawab user yang difilter berdasarkan filter status yang dipilih user | Ditampilkan list ticket tanggung jawab user yang difilter berdasarkan filter status yang dipilih user | diterima          |
| SRS-TKT-PIC-03      | Pengguna dapat membuat ticket                                 | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih menu membuat ticket</li> <li>4. Mengisi field</li> <li>5. Menekan tombol create</li> <li>6. Ticket berhasil dibuat</li> </ol> | company,<br>project,<br>judul ticket,<br>deskripsi ticket,<br>attachment, | Sistem membuat   | Pengguna tidak dapat membuat tiket dengan tombol revise tidak dapat ditemukan                         | diterima          |
| SRS-TKT-PIC-03      | Pengguna tidak mengisi beberapa field                         | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih menu membuat ticket</li> <li>4. Tidak Mengisi field</li> <li>5. Menekan tombol create</li> </ol>                              |   | Sistem tidak memperbolehkan membuat ticket   | Pengguna diberikan warning field yang kosong dan ticket tidak dibuat                                  | diterima          |

| <b>Identifikasi</b> | <b>Butir Uji</b>   | <b>Prosedur pengujian</b>   | <b>Masukan</b> | <b>Hasil yang diharapkan</b>                       | <b>Hasil yang diperoleh</b>   | <b>Kesimpulan</b> |
|---------------------|--|---|----------------|--|---|-------------------|
| SRS-TKT-PIC-04      | Pengguna dapat melihat detail tiket yang sudah ada             | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket</li> <li>4. Halaman detail ticket ditampilkan</li> </ol>   |                | Sistem menampilkan detail ticket yang dipilih user | Ditampilkan detail ticket yang dipilih user                                   | diterima          |
| SRS-TKT-PIC-05      | Pengguna dapat menutup ticket                                  | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket dengan status done</li> <li>4. Menekan tombol close</li> <li>5. Halaman detail ticket diperbarui menjadi closed</li> </ol> |                | Sistem merubah status ticket menjadi closed        | Ticket status berubah menjadi done  | diterima          |
| SRS-TKT-PIC-05      | Pengguna tidak dapat menutup ticket dengan status selain close | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket dengan status selain done</li> <li>4. Menekan tombol close</li> </ol>  |                | Sistem tidak memperbolehkan menutup ticket         | Pengguna tidak dapat menutup ticket karena tombol close tidak dapat ditemukan | diterima          |

| <b>Identifikasi</b> | <b>Butir Uji</b> | <b>Prosedur pengujian</b>                        | <b>Masukan</b>  | <b>Hasil yang diharapkan</b> | <b>Hasil yang diperoleh</b>  | <b>Kesimpulan</b>  |          |
|---------------------|------------------|--|---|------------------------------|--|--|----------|
| 107                 | SRS-TKT-PIC-06   | Pengguna dapat merevisi tiket dengan status done | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket dengan status done</li> <li>4. Menekan tombol revise</li> <li>5. Popup revise ticket ditampilkan</li> <li>6. Mengisi alasan merevisi</li> <li>7. Menekan tombol revise</li> </ol>  | revise reason                | Sistem merubah status ticket menjadi in progress dan alasan revise ditampilkan | Ticket status berubah menjadi in progress dan alasan revise ditampilkan      | diterima |
|                     | SRS-TKT-PIC-06   | Pengguna tidak mengisi alasan merevisi           | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket dengan status selain done</li> <li>4. Menekan tombol revise</li> <li>5. Popup revise ticket ditampilkan</li> <li>6. Menekan tombol revise tanpa mengisi alasan merevisi</li> </ol> |                              | Sistem tidak memperbolehkan merevisi ticket                                    | Pengguna tidak dapat merevisi tiket dengan tombol revise tidak dapat ditekan | diterima |

| <b>Identifikasi</b> | <b>Butir Uji</b>   | <b>Prosedur pengujian</b>  | <b>Masukan</b> | <b>Hasil yang diharapkan</b>                                    | <b>Hasil yang diperoleh</b>  | <b>Kesimpulan</b> |
|---------------------|--|--|----------------|---|--|-------------------|
| SRS-TKT-PIC-06      | Pengguna tidak dapat merevisi tiket dengan status selain done            | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket dengan status selain done</li> <li>4. Menekan tombol revise</li> </ol>  |                | Sistem tidak memperbolehkan merevisi ticket                     | Pengguna tidak dapat merevisi tiket dengan tombol revise tidak dapat ditemukan | diterima          |
| SRS-TKT-PM-07       | Pengguna dapat berkomunikasi dengan pihak internal (PM) melalui whatsapp | <ol style="list-style-type: none"> <li>1. Mengakses Sistem</li> <li>2. Membuka halaman ticket</li> <li>3. Memilih ticket</li> <li>4. Menekan Chat on WhatsApp</li> <li>5. Pengguna akan diarahkan ke whatsapp dengan template message</li> </ol> |                | Sistem mengarahkan pengguna ke whatsapp dengan template message | Pengguna diarahkan ke whatsapp dengan template message                         | diterima          |

## **BAB VI**

### **KESIMPULAN**

Bab ini merangkum hasil dan pengalaman dari praktik kerja lapangan, serta menyajikan saran untuk perbaikan dan pengembangan berdasarkan observasi dan kegiatan selama praktik tersebut.

#### **6.1. Kesimpulan**

Melalui Praktik Kerja Lapangan (PKL) yang dilakukan di PT Teknologi Aplikasi Sejahtera, penulis telah berhasil mengembangkan Sistem Manajemen Produk dan Ticketing untuk bagian modul Ticketing. Sistem yang telah dibuat diharapkan akan meningkatkan efisiensi dalam pengelolaan produk dan penanganan ticketing, sehingga dapat meningkatkan kualitas pelayanan di PT Teknologi Aplikasi Sejahtera.

#### **6.2. Saran**

Berdasarkan pengalaman masa pengembangan selama PKL, berikut adalah beberapa saran untuk pengembangan lebih lanjut dari sistem yang telah dibuat:

1. Mengimplementasikan fitur pencetakan ticket yang sudah maupun sedang dikerjakan. Fitur ini akan membantu dalam pembuatan laporan dan dokumentasi secara fisik.
2. Mengintegrasikan fitur ticketing dengan github issues.
3. Memberikan fitur push notifications menggunakan email.
4. Memberikan fitur diskusi di web pada halaman ticket.