

**RANCANG BANGUN *HELPDESK TICKETING SYSTEM*  
UNTUK PENANGANAN PERMINTAAN PENGGUNA TERHADAP  
DIVISI *INFORMATION TECHNOLOGY***

**SKRIPSI**

**Oleh :**

**Rayi Detriawan  
NIM. 16060123**



**JURUSAN TEKNIK INFORMATIKA  
SEKOLAH TINGGI TEKNOLOGI YBS INTERNASIONAL  
STT YBSI TASIKMALAYA  
2022**

## LEMBAR PENGESAHAN

Nama : Rayi Detriawan  
NIM : 16060123  
Judul : Rancang Bangun *Helpdesk Ticketing System* untuk penanganan permintaan pengguna terhadap Divisi *Information Technology*

Skripsi ini dipertahankan dalam Ujian Skripsi  
di depan tim penguji pada tanggal :

Skripsi ini Telah Disetujui dan Disahkan  
Tasikmalaya,

Pembimbing

**Arip, S.T., M.Kom.**  
**NID. 0410088603**

Mengetahui,  
Ketua Jurusan Teknik Informatika  
SEKOLAH TINGGI TEKNOLOGI YBS INTERNASIONAL  
TASIKMALAYA

**Siti Maesaroh, S.T., M.Kom.**  
**NID. 040459302**

## PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini:

Nama : Rayi Detriawan  
NIM : 16060123  
Judul : Rancang Bangun *Helpdesk Ticketing System* untuk penanganan permintaan pengguna terhadap Divisi *Information Technology*

Menyatakan dengan sebenarnya bahwa penulisan Skripsi ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri, baik untuk naskah laporan maupun kegiatan Programming yang tercantum sebagai bagian dari skripsi ini. Jika terdapat karya orang lain, saya akan mencantumkan sumber yang jelas.

Demikian pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini dan sanksi lain sesuai dengan peraturan yang berlaku di STT YBS Internasional Tasikmalaya.

Demikian pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Tasikmalaya,

Yang membuat pernyataan,

**Rayi Detriawan**  
**Nim. 16060123**

## ABSTRAK

CV Mitra Jaya Group merupakan perusahaan yang bergerak di bidang otomotif khususnya penjualan sepeda motor dan *spare part* resmi merk Honda. CV Mitra Jaya Group sendiri memiliki 34 cabang dealer resmi yang tersebar dipulau Jawa, Sumatera, Sulawesi dan Bali.

Berbagai macam masalah yang berkaitan dengan sistem *internal* maupun perangkat komputer sering ditemui pada cabang maupun *Head Office* CV Mitra Jaya Group. Sebagai contoh koreksi data atau perubahan data pada *database*, perbaikan perangkat komputer, cctv yang pengaduannya masih dilakukan lewat telepon atau *chatting* dan dinilai kurang efektif dalam menangani keluhan-keluhan tersebut. Permintaan yang diajukan terkadang datang dalam waktu yang bersamaan dan pengguna menuntut untuk bisa diselesaikan dalam waktu secepat mungkin. Hal tersebut dikarenakan pengguna tidak mengetahui permintaan lainnya yang diterima dan sedang dikerjakan oleh Divisi IT.

Oleh karena itu, penulis mengajukan *Helpdesk Ticketing System* yang dibuat untuk membantu Divisi IT dalam menyelesaikan permasalahan yang timbul diperusahaan. Secara garis besar aplikasi yang dikembangkan terdiri dari proses pencatatan keluhan, informasi status keluhan, informasi cara menyelesaikan keluhan, serta laporan. Metode yang digunakan pada pengembangan sistem ini yaitu dengan model *Waterfall* yang terdiri dari analisa kebutuhan sistem, desain sistem, penulisan kode program, pengujian program, dan pemeliharaan. Sistem didefinisikan dengan perancangan objek menggunakan konsep UML dan dibuat dengan berbasis *web* menggunakan bahasa pemograman PHP *framework* Laravel, *database* Mysql dan untuk mendapatkan layanan *helpdesk* yang lebih baik penulis memanfaatkan fitur gratis dari Google yaitu *Simple Mail Transfer Protocol* (SMTP) Gmail sebagai media notifikasi untuk meningkatkan kecepatan pengerjaan tiket.

**Kata Kunci :** *Helpdesk; SMTP; Waterfall; PHP; Framework; Laravel; Mysql*

## **KATA PENGANTAR**

Segala puji bagi Allah SWT yang telah memberikan rahmat-Nya sehingga tugas akhir ini dapat penulis selesaikan. Tugas akhir ini merupakan salah satu syarat kelulusan program pendidikan Sarjana (S1) pada program studi Teknik Informatika STT YBSI Tasikmalaya.

Tugas akhir ini membuat suatu perangkat lunak yang dapat memberikan layanan terhadap pengguna yang menggunakan sistem informasi di CV Mitra Jaya Group.

Penulisan tugas akhir ini tidak lepas dari dukungan bantuan dan bimbingan berbagai pihak. Dalam kesempatan ini penulis mengucapkan terima kasih dan penghargaan sebesar-besarnya kepada :

1. Bapak Dewanto Rosian Adhy, MT, Selaku Ketua STT YBSI Tasikmalaya.
2. Ibu Siti Maesaroh, S.T, M.Kom., Selaku Ketua Program Studi Teknik Informatika STT YBSI Tasikmalaya.
3. Bapak Arip, S.T, M.Kom., Selaku dosen pembimbing yang telah dengan sabar membimbing dan mengarahkan penulis dalam menyelesaikan skripsi.
4. Seluruh Dosen dan Karyawan STT YBSI Tasikmalaya.
5. Kedua orang tua dan kakak tercinta yang selalu memberikan dukungan, do'a, dan motivasi.
6. Rekan-rekan kerja CV Mitra Jaya Group yang telah membantu saya dalam menyelesaikan tugas akhir ini.
7. Rekan-rekan seperjuangan dari LP3I Tasikmalaya yang saya hormati.
8. Pihak yang telah membantu, sehingga penulis dapat menyelesaikan tugas akhir.

Penulis menyadari penulisan laporan Tugas Akhir ini masih jauh dari kesempurnaan, oleh karena itu kritik dan saran sangat penulis harapkan demi kesempurnaan laporan ini di kemudian hari.

Tasikmalaya, Juli 2021

Penulis

## DAFTAR ISI

## **DAFTAR GAMBAR**

## DAFTAR TABEL



## **DAFTAR ISTILAH**

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Beberapa tahun belakangan ini sistem informasi berbasis *web* di Indonesia semakin dirasakan keberadaannya dalam menunjang kegiatan-kegiatan sehari-hari. Pengolahan data secara manual tentunya tidak bisa mengimbangi kebutuhan akan penyajian informasi yang cepat, tepat dan akurat. Saat ini pengolahan data dengan cara manual dinilai kurang efektif untuk penyediaan laporan dan informasi bagi perusahaan yang sedang berkembang dan memiliki transaksi yang beragam.

CV Mitra Jaya Group merupakan perusahaan yang bergerak di bidang otomotif khususnya penjualan sepeda motor dan *spare part* resmi merk Honda. CV Mitra Jaya Group sendiri memiliki 34 cabang dealer resmi yang tersebar dipulau Jawa, Sumatera, Sulawesi dan Bali. Dalam perjalanan penggunaan sistem informasi maupun perangkat komputer sering mendapat permasalahan-permasalahan yang jika tidak ditangani akan mengakibatkan adanya kemunduran kinerja. Untuk menyelesaikan masalah tersebut, cabang mengajukan permintaan terhadap Divisi IT melalui telepon atau *chatting*.

Divisi IT adalah sebuah departemen yang bertanggung jawab dalam pembangunan dan pengembangan perangkat lunak maupun perbaikan *hardware*. Selain itu, Divisi IT juga bertanggung jawab penuh terhadap permintaan yang diajukan oleh pengguna sistem internal dan perangkat komputer. Permintaan yang diajukan terkadang datang dalam waktu yang bersamaan dan pengguna menuntut untuk bisa diselesaikan dalam waktu secepat mungkin. Hal tersebut dikarenakan pengguna tidak mengetahui permintaan lainnya yang diterima dan sedang dikerjakan oleh Divisi IT. Semua permasalahan dan permintaan tidak dapat terkontrol karena sistem yang ada saat ini masih menggunakan telepon atau *chatting*. Sehingga perbaikan seringkali mengalami keterlambatan dalam segi waktu dan akibatnya dapat berdampak terhadap kinerja beberapa pengguna.

Berdasarkan permasalahan yang ditemukan, maka diperlukan sebuah solusi agar kinerja pada perusahaan tidak terganggu dan bisa membantu Divisi IT dalam melakukan permintaan pengerjaan. Salah satu cara yang dapat dilakukan yaitu dengan diadakannya rancangan bangun *Helpdesk Ticketing System*. Aplikasi

tersebut diharapkan dapat mengatasi semua masalah yang timbul sehingga membuat pekerjaan dapat berjalan dengan baik dan meningkatkan kinerja pada perusahaan.

## **1.2 Perumusan Masalah**

Berdasarkan uraian latar belakang di atas, penulis mengambil beberapa rumusan masalah yang ada pada objek penelitian antara lain :

1. Bagaimana perancangan dan pembangunan *Helpdesk Ticketing System* untuk layanan permintaan terhadap Divisi IT?
2. Bagaimana cara menanggapi laporan permasalahan *user* dengan cepat dan efisien ?
3. Bagaimana mengatasi laporan-laporan permintaan yang tidak terdokumentasi dengan baik ?

## **1.3 Tujuan dan Manfaat Penelitian**

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut :

1. Merancang, membangun *Helpdesk Ticketing System* untuk layanan permintaan keluhan pengguna terhadap Divisi IT pada CV Mitra Jaya Group.
2. Meningkatkan efisiensi Divisi IT dalam menangani keluhan para pengguna.
3. Dapat memberikan solusi atas permasalahan yang muncul secara cepat.
4. Dapat memantau keluhan atau permintaan yang masuk dengan mudah.

Adapun manfaat-manfaat yang dapat diperoleh baik untuk penulis, perusahaan maupun bagi kampus, yaitu sebagai berikut :

1. Manfaat Bagi Penulis
  - Meningkatkan wawasan dan ilmu seputar teknologi sistem informasi maupun proses layanan permintaan pengguna di CV Mitra Jaya Group.
  - Mengimplementasikan materi yang dipelajari saat perkuliahan kedalam bentuk suatu project sehingga bisa dapat bermanfaat bagi perusahaan maupun masyarakat luas.
2. Manfaat Bagi Perusahaan
  - Perusahaan dapat meningkatkan kinerja karyawan.

- Memaksimalkan dan mengikuti perkembangan teknologi yang sudah berkembang.
  - Untuk memudahkan Divisi IT dalam menangani permintaan perubahan sistem atau data dari pengguna sistem.
3. Manfaat Bagi Kampus
- Menjadikan universitas yang mampu bersaing dalam hal mendidik mahasiswa unggul dalam ilmu teknologi informasi.
  - Menciptakan lulusan yang dapat menganalisa dan merancang sebuah sistem dan di buat dalam bentuk laporan penelitian.
  - Menciptakan Lulusan mahasiswa yang dapat berkontribusi bagi instansi maupun masyarakat langsung.

#### **1.4 Batasan Masalah**

Batasan masalah dalam penelitian ini yaitu :

1. Sistem tidak memiliki layanan registrasi, jika pengguna belum memiliki akun maka harus didaftarkan oleh SPV IT.
2. Aplikasi ini menggunakan bahasa pemograman PHP (*Hypertext Preprocessor*) dengan versi 7.2.4, *Framework* Laravel dengan versi 7.30.4, database Mysql dan menggunakan SMTP Google Mail untuk media notifikasi.
3. Hanya *Supervisor* yang bisa menentukan teknisi yang akan melayani permintaan pengguna.

#### **1.5 Metode Penelitian**

Penelitian ini dilakukan melalui dua tahapan yaitu pengumpulan data dan pengembangan sistem. Pengumpulan data dilakukan dengan metode wawancara dan studi literatur. Sedangkan pengembangan sistem dilakukan menggunakan metode *waterfall*.

##### **1.5.1 Metode Pengumpulan Data**

Metode pengumpulan data pada penelitian ini menggunakan dua tahap yaitu wawancara dan studi literatur.

1. Wawancara (*Interview*)

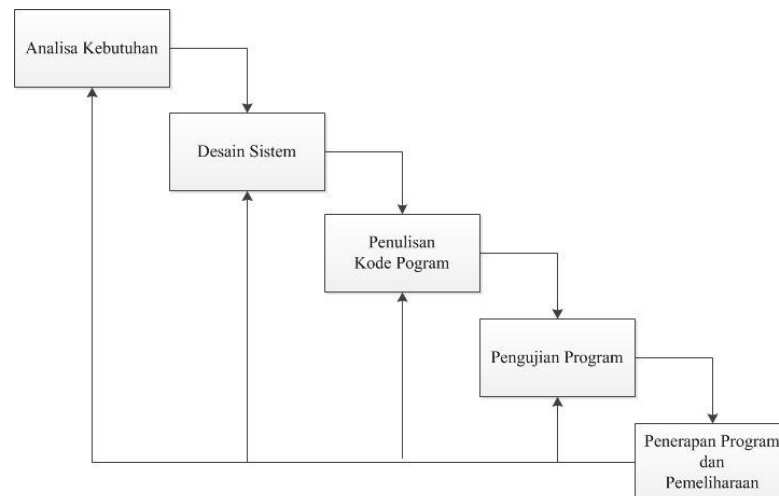
Merupakan suatu pengumpulan data yang dilakukan dengan cara tanya jawab atau dialog secara langsung dengan pihak-pihak yang terkait. Wawancara dilakukan dengan cara tanya jawab langsung kepada pihak yang bersangkutan dengan penelitian penulis di CV Mitra Jaya Group.

## 2. Studi Literatur

Penulis melakukan pencarian yang ada kaitannya dengan judul untuk mendapatkan data dengan cara mengumpulkan literatur, jurnal, paper, makalah dan berbagai referensi dari internet.

### 1.5.2 Metode Pengembangan Sistem

Metode pengembangan sistem dilakukan dengan menggunakan *Waterfall* Model. Dalam metode tersebut semua proses kegiatan dilakukan secara sekuensial dan juga saling berurutan. Proses pengembangan dilakukan dari mulai analisa kebutuhan, desain sistem, penulisan kode program, pengujian program dan pemeliharaan. *Waterfall* model menurut (Rosa A.S M.Shalahuddin, 2011) dapat dilihat pada Gambar 1.1.



Gambar 7.1 Model *Waterfall*

Tahapan *Waterfall* model menurut (Rosa AS M.Shalahuddin, 2011) terdiri dari 5 (Lima) tahapan yaitu : analisa kebutuhan, desain sistem, penulisan kode program, pengujian program dan pemeliharaan. Dalam penelitian ini dikaji hanya sampai tahap ke-4 yaitu tahap pengujian program karena aplikasi baru sampai tahap pengujian belum sampai tahap pemeliharaan, oleh karena itu tahapan yang ke-5 tidak dibahas. Detail tahapan *waterfall* pada Gambar 1.1, yaitu:

#### 1. Analisa Kebutuhan

Tahap ini melakukan analisis permasalahan pada komunikasi yang berjalan saat ini antara pengguna dengan divisi IT *Development* kemudian melakukan konsultasi untuk menemukan solusi.

## 2. Desain Sistem

Tahapan desain adalah merancang aplikasi berdasarkan hasil analisis yang telah dilakukan. Perancangan tersebut mencakup *Use Case Diagram*, *Entity Relationship Diagram*, *Class Diagram* dan perancangan antar muka aplikasi.

## 3. Penulisan Kode Program

Tahap penulisan kode program yaitu mengimplementasikan tahapan perancangan dalam bentuk *coding* aplikasi dan membuat desain aplikasi. Bahasa pemrograman yang digunakan adalah php dengan *framework* Laravel dan basis data menggunakan MySQL.

## 4. Pengujian Program

Tahapan pengujian program yaitu melakukan pengujian terhadap sistem yang telah dibuat. Pengujian aplikasi dalam penelitian ini diuji menggunakan *Black Box Testing* terhadap semua fungsi dalam aplikasi.

## 5. Pemeliharaan

Tahapan pemeliharaan yaitu pemeliharaan dan pengembangan aplikasi yang telah digunakan *user*. Pemeliharaan meliputi perbaikan *error* atau *bugs* di aplikasi, sedangkan pengembangan meliputi penambahan fitur baru dan penyempurnaan aplikasi sesuai kebutuhan user. Namun untuk saat ini penulis tidak membahas sampai pada tahap ini dikarenakan aplikasi belum sampai pada tahap pemeliharaan.

### 1.6 Sistematika Penulisan

Untuk mengetahui gambaran dari isi yang ada dalam penulisan ini, penulis membagi pembahasan ini dalam lima bab yang saling berhubungan. Adapun sistematika penulisannya adalah sebagai berikut:

## BAB I PENDAHULUAN

Bab ini menjelaskan tentang informasi umum yaitu, Latar Belakang, Rumusan Masalah, Tujuan dan Manfaat penelitian, Batasan Masalah, Metode Penelitian dan Sistematika Penelitian.

## **BAB II LANDASAN TEORI**

Bab ini membahas berbagai konsep dasar, teori-teori yang berkaitan dengan topik penelitian yang dilakukan dan hal-hal yang berguna dalam proses analisis permasalahan serta tinjauan terhadap penelitian. Bab ini meliputi referensi tentang Rancang Bangun, Aplikasi, *Helpdesk*, UML, PHP, Laravel, Konsep MVC, SMTP, *Bootsrap*, *Javascript*, CSS, Mysql dan *Testing Software*.

## **BAB III ANALISIS DAN PERANCANGAN**

Bab ini berisi tentang sistem yang sedang berjalan dan uraian mengenai perancangan sistem yang akan dibangun yang meliputi penjabaran mengenai konsep usulan yang digambarkan UML (*Use Case Diagram*, *Scenario Use Case*, *Activity Diagram*, *Sequence Diagram*, *Deployment Diagram*, *Component Diagram*, *State Chart Diagram*, *Class Diagram*), serta melakukan perancangan tampilan antarmuka *Helpdesk Ticketing System*.

## **BAB IV IMPLEMENTASI DAN PENGUJIAN**

Bab ini menjelaskan mengenai implementasi antarmuka, perangkat keras dan perangkat lunak yang digunakan dalam pembangunan *Helpdesk Ticketing System*, serta pengujian dengan menggunakan *Black box testing*.

## **BAB V PENUTUP**

Bab ini menguraikan beberapa simpulan dari pembahasan masalah pada bab-bab sebelumnya serta memberikan saran yang bisa bermanfaat bagi penyusun.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Rancang Bangun**

Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan (Pressman, 2002). Rancangan sistem adalah penentuan proses dan data yang diperlukan oleh sistem baru (McLeod, 2002). Perancangan adalah kegiatan yang memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik (Ladjamudin, 2005). Sedangkan pengertian bangun atau pembangunan sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian (Pressman, 2002). Bangun sistem adalah membangun sistem informasi dan komponen yang didasarkan pada spesifikasi desain (Whitten et al, 2004).

Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada.

#### **2.2 Aplikasi**

Program aplikasi (sering kali hanya disebut aplikasi) adalah program yang dibuat oleh peemakai yang ditujukan untuk melakukan tugas khusus. (Abdul Kadir, 2013).

Menurut (Jogiyanto, 2004), aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data. Aplikasi secara umum yaitu suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal.

Menurut (Wibowo, 2017), aplikasi adalah sekumpulan perintah program yang dibuat untuk mempermudah pengguna dalam melakukan tugas-tugas tertentu yang berhubungan dengan pengelolaan informasi-informasi sehingga menghasilkan *output* yang sesuai dengan kebutuhan pengguna.

Aplikasi secara umum yaitu proses dari secara manual yang ditransformasikan ke kompuer dengan membuat sistem atau program supaya data



diolah lebih berdaya guna secara optimal. Dari pengertian tersebut, dapat disimpulkan bahwa aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah atau melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.

### **2.3 Helpdesk**

*Helpdesk* pada dasarnya adalah sebuah *center point* dimana masalah atau *issue* dilaporkan dan diatur secara terurut dan diorganisasikan. Dari perspektif umum, *helpdesk* merupakan bagian pelengkap dari sebuah fungsi pelayanan, dan bertanggung jawab sebagai sumber dari pemecahan masalah atau *issue* lainnya. (*Helpdesk World*, 2012)

Menurut Wooten (2001), "*Helpdesk is a formal organization that provides support function to users of the companies product, services, or technology*". Jadi *helpdesk* merupakan bagian dari perusahaan yang menyediakan dokumen fungsi produk, servis atau teknologi dari perusahaan tersebut.

Menurut (Syukriah, 2007), *helpdesk system* merupakan suatu sistem digunakan untuk penanganan *problem management* yang mengacu kepada perusahaan tersebut sehingga melalui sistem *ticketing*, *incident* ataupun *problem managetnent* diakibatkan oleh *IT services* perusahaan dapat diidentifikasi dan dikonsolidasikan melalui berbagai media komunikasi yang tersedia di perusahaan, seperti telepon, email, dan juga web *interface* sehingga seluruh *incident* ataupun *problem management* dapat ditanggulangi dan diberikan solusinya atas permasalahan yang muncul.

Berdasarkan definisi-definisi di atas dapat disimpulkan bahwa *helpdesk* merupakan sebuah bagian atau program aplikasi dalam perusahaan yang menangani pertanyaan dan keluhan dari pengguna sistem informasi perusahaan dengan menyediakan informasi serta solusi yang diperlukan.

### **2.4 UML**

Menurut Sucipto (2011 : 195) UML atau (*Unified Modeling Language*) adalah cara sukses merombak analisa berorientasi objek dan desain, dan muncul pertama kali pada tahun 90'an. Itu ada setelah banyak pemikiran-pemikiran gabungan dari *Booch*, *Rumbaugh* (OMT) dan *Jacobson*, yang dianggap sebagai pendahulu munculnya UML. UML menembus proses standarisasi bersama OMG

(*Object Management Group*) dan sekarang menjadi standar pembuatan sistem yang sering dipakai.

UML disebut juga contoh bahasa yang terdiri dari banyak cara dan kaidah-kaidah yang sangat penting dalam perancangan dan desain suatu sistem, UML sebagai grafis utama untuk catatan cara mendesain dengan cepat dan prosedural. Dalam mendesain dan merancang sistem UML menganjurkan tahapan-tahapan dalam pengerjaannya. Karena UML adalah bagian yang sangat penting untuk dijadikan sebagai kaidah dalam perancangan dan desain sistem. Sebuah kepastian adalah bagian kunci terpenting dalam melakukan komunikasi untuk sebuah rancangan.

## **2.5 PHP (*Hypertext Preprocessor*)**

Menurut (Setiawan, 2017), PHP merupakan kependekan dari *Hypertext Preprocessor* yang merupakan sebuah bahasa *script* tingkat tinggi yang dipasang pada dokumen HTML. Sebagian besar sintaks PHP mirip dengan bahasa C, java, dan perl. Namun, PHP terdapat beberapa fungsi yang lebih spesifik. PHP digunakan untuk merancang yang sifatnya dinamis dan dapat bekerja secara otomatis.

Web dinamis yang bisa dibuat dengan menggunakan PHP adalah situs web yang bisa menyesuaikan tampilan konten tergantung kebutuhan atau situasi. Web dinamis juga dapat menyimpan data ke dalam *database*, membuat halaman yang berubah ubah sesuai dengan permintaan pengguna, dan lain sebagainya. Dalam pembuatan web, kode PHP biasanya disisipkan ke dalam dokumen HTM. PHP disebut juga sebagai *scripting language* karena memiliki fitur tersebut.

## **2.6 Laravel**

Pengertian *framework* menurut Naista adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah yang kompleks. Singkatnya, *framework* adalah wadah atau kerangka kerja dari sebuah *website* yang akan dibangun. Dengan menggunakan kerangka tersebut waktu yang digunakan dalam membuat website lebih singkat dan memudahkan dalam melakukan perbaikan.

Salah satu *framework* yang banyak digunakan oleh *programmer* adalah *framework* laravel. Laravel adalah *framework* berbasis PHP yang sifatnya *open source*, dan menggunakan konsep *model – view – controller*. Laravel berada di

bawah lisesni MIT *License* dengan menggunakan Github sebagai tempat berbagi code menjalankannya (Naista, 2017).

Dalam penggunaanya laravel memiliki beberapa kekurangan salah satunya yaitu ukuran file yang cukup besar. Di dalam laravel terdapat file yang sifatnya *default* seperti *vendor*. File tersebut tidak boleh dihapus sembarangan sehingga ukuran *website* yang dibuta berukuran cukup besar. Selain itu, dibutuhkan koneksi internet untuk instalasi dan mengunduh *library* laravel, dan PHP minimal versi 5.4 untuk menjalankannya (Naista, 2017).

Berikut adalah dasar-dasar laravel :

### 1. *Artisan*

*Artisan* adalah *command line* atau perintah yang dijalankan melalui terminal dan disediakan beberapa perintah perintah yang dapat digunakan selama melakukan pengembangan dan pembuatan aplikasi. Salah satu fungsi dari php artisan yaitu “*php artisan serve*”. *Php artisan serve* berfungsi untuk membuka website yang telah dibuat tanpa menggunakan *web server* lokal. Berikut adalah contoh salah satu penggunaan *artisan* dalam Laravel.

```
C:\xampp\htdocs\lara_helpdesk
λ php artisan serve
Laravel development server started: http://127.0.0.1:8000
[Sun Apr 4 10:59:06 2021] 127.0.0.1:62784 [200]: /assets/libs/datatables.net-buttons-bs4/css/buttons.bootstrap4.min.css
[Sun Apr 4 10:59:06 2021] 127.0.0.1:62786 [200]: /assets/libs/datatables.net-bs4/css/dataTables.bootstrap4.min.css
[Sun Apr 4 10:59:06 2021] 127.0.0.1:62787 [200]: /assets/css/bootstrap.min.css
[Sun Apr 4 10:59:06 2021] 127.0.0.1:62788 [200]: /assets/css/icons.min.css
[Sun Apr 4 10:59:06 2021] 127.0.0.1:62789 [200]: /assets/css/app.min.css
[Sun Apr 4 10:59:06 2021] 127.0.0.1:62790 [200]: /assets/libs/jquery/jquery.min.js
[Sun Apr 4 10:59:06 2021] 127.0.0.1:62791 [200]: /assets/libs/bootstrap/js/bootstrap.bundle.min.js
[Sun Apr 4 10:59:06 2021] 127.0.0.1:62792 [200]: /assets/libs/metismenu/metisMenu.min.js
[Sun Apr 4 10:59:06 2021] 127.0.0.1:62793 [200]: /assets/libs/simplebar/simplebar.min.js
[Sun Apr 4 10:59:06 2021] 127.0.0.1:62794 [200]: /assets/libs/node-waves/waves.min.js
[Sun Apr 4 10:59:06 2021] 127.0.0.1:62795 [200]: /assets/libs/apexcharts/apexcharts.min.js
[Sun Apr 4 10:59:06 2021] 127.0.0.1:62796 [200]: /assets/js/pages/dashboard.init.js
[Sun Apr 4 10:59:06 2021] 127.0.0.1:62799 [200]: /assets/libs/datatables.net-buttons/js/dataTables.buttons.min.js
```

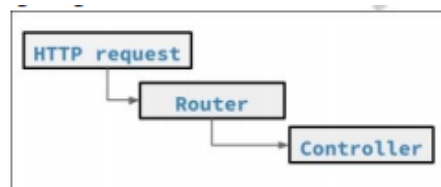
Gambar 6.1 *php artisan serve*

### 2. *Routing*

*Routing* adalah suatu proses yang bertujuan agar suatu item yang diinginkan dapat sampai ke tujuan. Dengan menggunakan *routing* dapat ditentukan halaman halaman yang akan muncul ketika dibuka oleh *user*. Pengaturan *routing* di laravel biasanya terletak di file *web.php*. File *web.php* terletak di dalam *folder routes*.

### 3. Controller

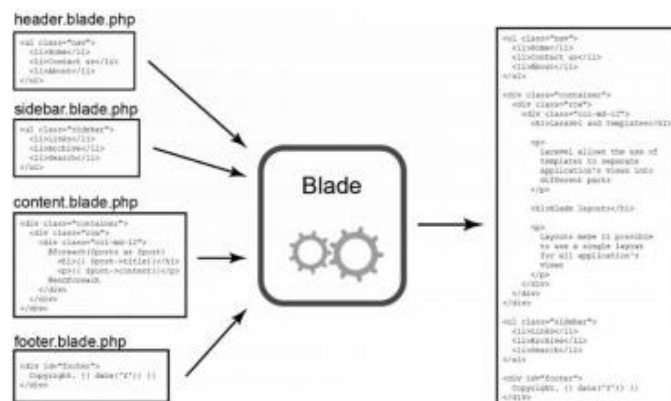
*Controller* adalah suatu proses yang bertujuan untuk mengambil permintaan, menginisialisasi, memanggil *model* untuk dikirimkan ke *view*. Ada dua cara membuat *controller* di laravel. Cara pertama adalah dibuat file *controller* secara manual dan dituliskan code *extends controller* di dalamnya. Cara kedua adalah dibuat file *controller* menggunakan *command line* dengan menuliskan “*php artisan make:controller nama\_file\_controller*”. Permintaan yang dibuat dalam laravel harus berada di dalam *controller*, kemudian dilempar melalui *routing* untuk mendapat permintaan yang diinginkan.



Gambar 6.2 Controller Laravel

### 4. View (blade templating)

*Blade* adalah *template engine* bawaan dari laravel. *Blade* memiliki kode-kode yang lebih mudah untuk menghasilkan laravel. Cara membuat *file.blade* dilakukan secara manual dengan membuat *nama\_file.blade.php* di dalam folder *views*. Di dalam *blade* dapat dibuat *template master* dan *template inheritance*. Pembuatan *template master* dan turunannya ini bertujuan agar elemen yang sama tidak ditulis secara berulang-ulang. Pada *template inheritance* diberikan kode “*extend (nama\_layout)*” dan *section (nama\_content)*”.



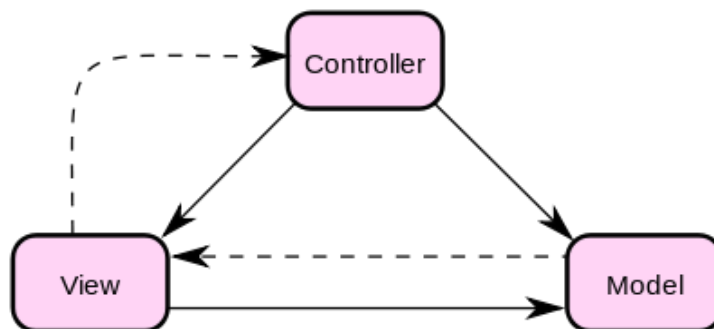
Gambar 6.3 Blade Templating

## 5. *Middleware*

*Middleware* adalah penengah Antara *request* yang masuk dengan *controller* yang dituju. Cara membuat *middleware* menggunakan *artisan* dengan mengetikkan “*php artisan make:middleware nama\_file*”. File *middleware* berada di dalam *folder middleware*.

## 2.7 Konsep MVC

*Model-View-Controller* atau MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*). Dalam implementasinya kebanyakan kerangka kerja (*framework*) dalam aplikasi web adalah berbasis arsitektur MVC. MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi web. Konsep MVC dapat dilihat pada Gambar 2.1.



Gambar 6.4 Konsep MVC

1. *Model* mewakili struktur data. *Model* merupakan bagian yang bertugas untuk mengatur, menyiapkan, memanipulasi, dan mengorganisir data (biasanya dari basis data). Tugas yang ia lakukan meliputi memasukkan data ke basis data, pembaruan data, menghapus data, dan lain-lain. *Model* menjalankan tugasnya berdasarkan instruksi dari *controller*.
2. *View* merupakan bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.
3. *Controller* merupakan bagian yang menjembatani *model* dan *view*. *Controller* berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web.

Dengan menggunakan metode MVC maka aplikasi akan lebih mudah untuk dirawat dan dikembangkan. Untuk memahami metode pengembangan aplikasi menggunakan MVC diperlukan pengetahuan tentang pemrograman berorientasi objek (*Object-oriented programming*).

## **2.8 SMTP (Simple Mail Transfer Protocol)**

*Simple Mail Transfer Protocol* (SMTP) merupakan tata cara standar untuk mengirim *e-mail*. *Simple Mail Transfer Protocol* (SMTP) bekerjasama dengan beberapa *server Post Office Protocol* (POP). Ketika pengirim mengirimkan pesan *e-mail*, komputer akan mengarahkan pesan tersebut kepada *server Simple Mail Transfer Protocol* (SMTP). *Server* mencari alamat *e-mail* tersebut ke *mail server* penerima dan menyimpannya di sana sampai orang yang dituju membukanya.

## **2.9 Bootstrap**

Menurut Husein Alatas (2013) Bootstrap merupakan *framework* HTML dan CSS untuk membangun desain web secara responsif. Artinya, tampilan web yang dibuat oleh bootstrap akan menyesuaikan ukuran layer dan browser yang kita gunakan baik di desktop, tablet ataupun *mobile device*. Dengan bootstrap kita juga bisa membangun *web* dinamis ataupun statis.

## **2.10 Javascript**

Menurut Kadir (2013) Javascript merupakan bahasa pemrograman khusus untuk halaman *web* agar halaman *web* menjadi lebih hidup. Menurut Kadir (2013)

Javascript yang tertulis dalam sebuah halaman web dapat mengetahui dan merespon perintah pemakai seperti gerakan *mouse*, *input form*, dan navigasi halaman HTML.

Menurut Kadir (2013) *Javascript* mampu menginterpretasikan teks yang dituliskan pada halaman HTML dan memberikan tampilan teks dialog penolakan apabila teks yang dituliskan salah. Menurut Kadir (2013) *Javascript* juga mampu mengeksekusi perintah seperti memainkan file suara, mengeksekusi sebuah *applet* atau berkomunikasi dengan “*plug-in*” lain sebagai respon terhadap dibukanya sebuah halaman web atau penutupan halaman tersebut oleh pengguna.

### 2.11 *Cascading Style Sheet (CSS)*

Menurut Kadir (2013) CSS merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. Menurut Kadir (2013) CSS bukan merupakan bahasa pemrograman. CSS umumnya dipakai untuk membuat format tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

Menurut Kadir (2013) CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna mouse over, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. Menurut Kadir (2013) CSS memungkinkan untuk menampilkan halaman yang sama dengan format yang berbeda.

### 2.12 *MySQL*

Menurut Sulhan (2006:18) *MySQL* merupakan perangkat lunak yang digunakan untuk membangun *database* yang sering digunakan di lingkungan linux. *MySQL* merupakan *software open source* yang berarti *free* digunakan. Menurut Kadir (2013) *MySQL* merupakan sebuah aplikasi *Relational Database Management Server* (RDBMS) yang sangat cepat dan kokoh dengan dukungan bebas lisensi terbuka (*open source*). Menurut Kadir (2013) *Database server* yang banyak digunakan dengan dukungan SQL sebagai bahasa dasar untuk *query*.

Menurut Kadir (2013) Data yang telah disimpan di komputer perlu pengelolaan dengan sistem manajemen basis data yang baik. Menurut Kadir (2013) Adapun keunggulan *MySQL* dalam mengolah database adalah sebagai berikut:

1. Kecepatan yang lebih baik dibandingkan *tools* RDBMS lainnya berdasarkan setiap peningkatan versi aplikasi.
2. Perintah yang mudah digunakan dengan bahasa SQL (*Structured Query Language*) sebagai bahasa standar database.
3. Biaya yang murah dengan dukungan lisensi terbuka (*open source*) sehingga dapat diimplementasikan dengan aturan GNU *public licence*.
4. Kapabilitas yang mumpuni dalam memproses data yang tersimpan dengan jumlah 50 juta record, 60.000 tabel dan 5.000.000.000 jumlah baris, dan mampu memproses sebanyak 32 indeks per-tabel.

5. Keamanan yang baik sehingga mampu menerapkan hak akses secara bertingkat dengan pengacakan kata (*encrypted password*), dukungan dalam pengacakan lapisan data.
6. Minim kesalahan (*bug*) dan dukungan *multi platform* yang dapat diimplementasikan pada setiap sistem operasi yang ada.

### **2.13 Testing Software**

Menurut Quadri dan Farooq (2010: 1), pengujian *software* adalah proses verifikasi dan validasi apakah sebuah aplikasi *software* atau program memenuhi persyaratan bisnis dan persyaratan teknis yang mengarahkan desain dan pengembangan dan cara kerjanya seperti yang diharapkan dan juga mengidentifikasi kesalahan yang penting yang digolongkan berdasarkan tingkat *severity* pada aplikasi yang harus diperbaiki.

Menurut Nidhra dan Dondeti (2012:1), pengujian *software* adalah teknik yang sering digunakan untuk verifikasi dan validasi kualitas suatu *software*. Pengujian *software* adalah prosedur untuk eksekusi sebuah program atau sistem dengan tujuan untuk menemukan kesalahan.

Kesimpulan yang dapat diambil dari pendapat-pendapat tersebut adalah pengujian *software* merupakan proses verifikasi dan validasi apakah *software* memenuhi *requirement* dan mengidentifikasi kesalahan-kesalahan yang ditemukan saat eksekusi program.

### **2.14 Black box testing**

Blackbox testing adalah tahap yang digunakan untuk menguji kelancaran program yang telah dibuat. Pengujian ini penting dilakukan agar tidak terjadi kesalahan alur program yang telah dibuat.

Menurut Rosa dan Salahuddin (2015:275) “*Blackbox testing* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”.

Menurut Rizky (2011:264) “*Blackbox testing* adalah tipe testing yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya”.

Sedangkan menurut Mustaqbal, dkk (2015:34) “*Black Box Testing* befokus pada spesifikasi fungsional dari perangkat lunak, kumpulan kondisi input dan melakukan pengetesan pada fungsional program”.



## **BAB III**

### **ANALISA DAN PERANCANGAN SISTEM**

#### **3.1 Analisa Sistem**

Pada awalnya penulis melakukan studi pendahuluan yaitu peneliti mempelajari objek penelitian kemudian menetapkan tujuan penelitian terhadap objek yang akan di teliti sembari menganalisa dan mempelajari literatur maupun studi pustaka yang berkaitan dengan analisa dan perancangan yang berkaitan dengan penulisan tugas akhir ini.

Setelah itu penulis menetapkan metode pengumpulan data untuk mendapatkan data-data yang dibutuhkan dalam perancangan sistem. Setelah berhasil mendapatkan data yang dibutuhkan melalui metode pengumpulan data yang telah ditetapkan maka penulis melanjutkan ke tahap analisa dan perancangan.

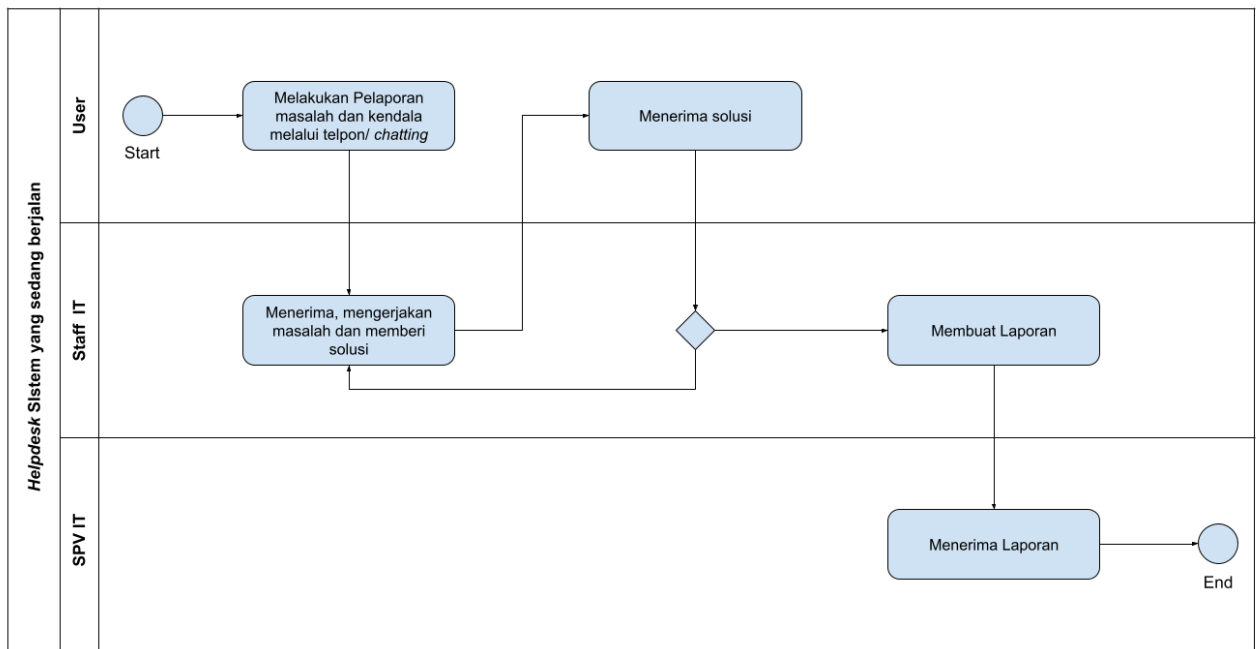
Tahap selanjutnya penulis membuat *user requirement* sebagai hasil dari analisa dan sebagai sistem yang diharapkan ada setelah proyek selesai. *User requirement* juga merupakan perincian kebutuhan-kebutuhan *user* terhadap sistem. Setelah membuat *user requirement*, penulis akan menyimpulkan hasil dari penelitian, merancang dan mulai membuat sistem.

##### **3.1.1 Analisa Sistem Yang Sedang Berjalan**

Analisa sistem yang sedang berjalan bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut dan masalah yang dihadapi sistem untuk dijadikan landasan usulan perancangan analisa sistem yang sedang berjalan yang dilakukan berdasarkan urutan kejadian yang ada.

Sistem yang sedang berjalan dilakukan keseluruhan secara manual yaitu pelaporan melalui telepon dan *chatting* sedangkan sistem yang diusulkan lebih ditekankan pada pelaporan melalui sistem yang akan dibuat agar kasus pelaporan terdokumentasi dengan baik. Dengan demikian pembuatan sistem ini diharapkan dapat membantu mengatasi kekurangan-kekurangan yang ada dalam hal pelaporan perbaikan sistem maupun kerusakan komputer.

Berikut ini adalah prosedur yang berjalan dalam penanganan permintaan yang terjadi di CV Mitra Jaya Group menggunakan model BPMN:



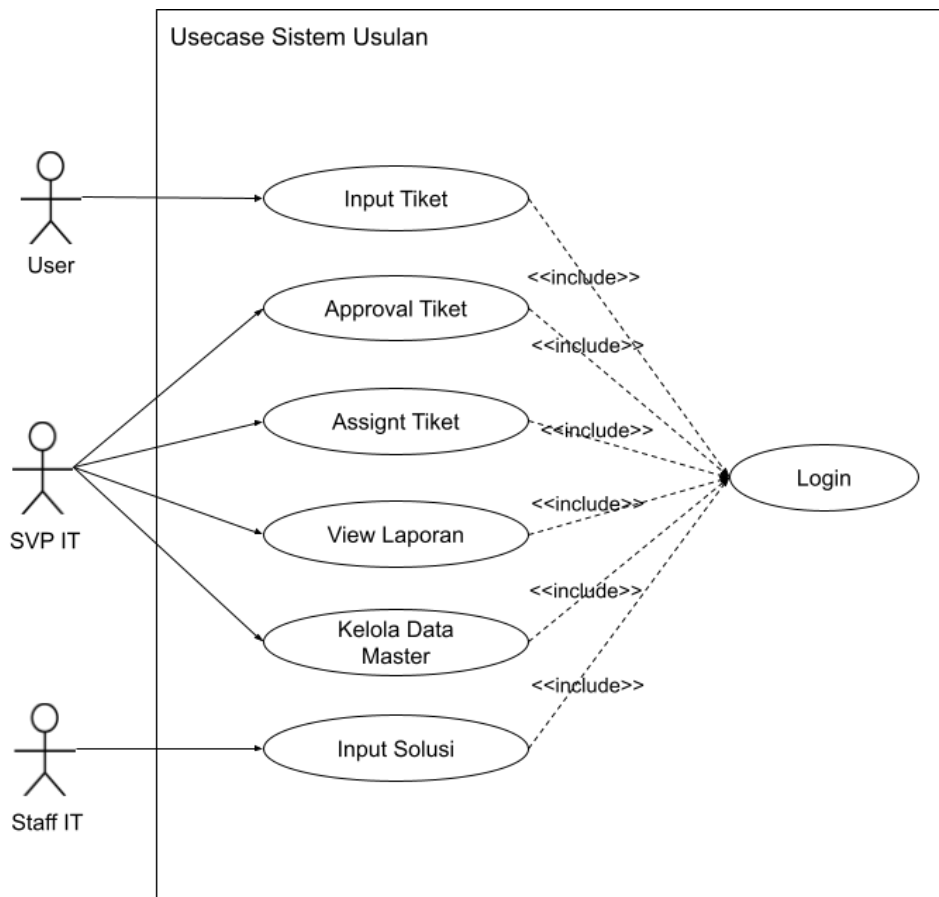
Gambar 3.1 Sistem Berjalan dengan Model BPMN

Kekurangan yang terdapat pada sistem yang sedang berjalan adalah sebagai berikut :

1. Media pelaporan hanya melalui telepon dan *chatting* sehingga kurang efektif karena tidak adanya data mengenai kerusakan sebelumnya.
2. Kesulitan dalam penilaian setiap karyawan pada divisi IT karena tidak semua karyawan turut bekerja dan tidak ada data siapa-siapa saja yang mengatasi permasalahan yang dilaporkan.
3. Kurang efektif apabila permintaan datang pada waktu yang bersamaan dan harus diselesaikan pada waktu yang sama karena *user* yang mengajukan permintaan tidak mengetahui permintaan dari *user* lainnya yang sedang dikerjakan oleh Divisi IT.

### 3.2 Perancangan Sistem Usulan

Dengan meninjau dan mengidentifikasi permasalahan yang sedang berjalan, maka dibutuhkan suatu sistem yang mampu mengatasi masalah tersebut. Oleh karena itu peneliti mengusulkan sebuah sistem yang dapat mengatasi masalah itu semua. Berikut diagram use case yang merupakan rancangan usulan *helpdesk ticketing system*.



Gambar 3.2 Use Case Sistem Usulan

Dimana use case tersebut dapat dideskripsikan sebagai berikut:

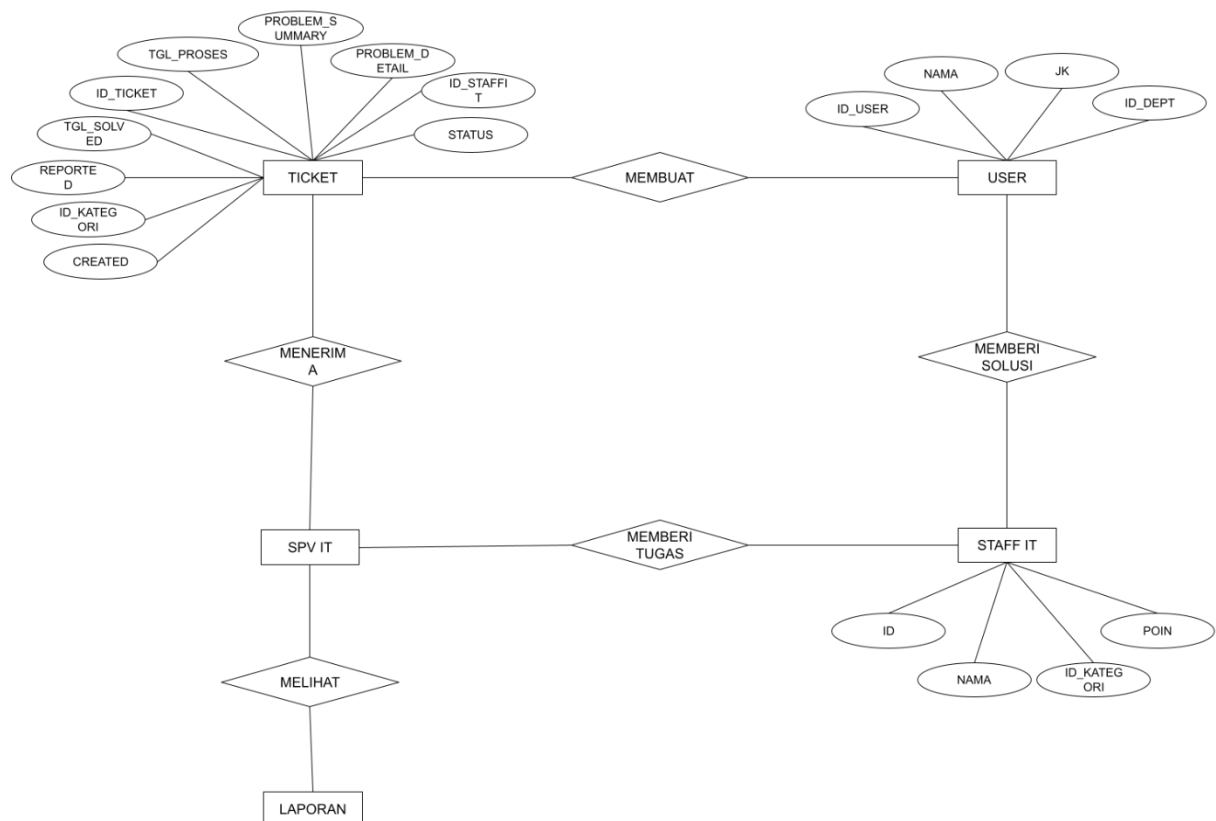
Tabel 3.2 Deskripsi use case sistem usulan

NO	Use case	Deskripsi	Aktor
1.	Login	Usecase menggambarkan kegiatan login kedalam sistem dengan memasukkan data pengguna untuk diverifikasi oleh sistem.	User, Staff IT, SVP IT
2.	Input Tiket	Usecase menggambarkan kegiatan memasukkan data tiket.	User
3.	Approval Tiket	Usecase menggambarkan kegiatan menyetujui data tiket.	SPV IT
4.	Assigned Tiket	Usecase menggambarkan kegiatan menugaskan teknisi	SPV IT

		untuk menyelesaikan masalah atau tiket.	
5.	View Laporan	<i>Usecase</i> menggambarkan kegiatan melihat laporan tiket.	SPV IT
6.	Kelola Data Master	<i>Usecase</i> menggambarkan kegiatan mengelola data master.	SPV IT
7.	Input Solusi	<i>Usecase</i> menggambarkan kegiatan memasukkan data solusi atau penyelesaian terhadap tiket kedalam sistem.	Staff IT

### 3.2.1 Rancangan ERD

ERD menggambarkan hubungan antara satu entitas yang memiliki sejumlah atribut dengan entitas yang lain dalam suatu sistem yang terintegrasi. Pada Gambar 3.3 menunjukan rancangan ERD yang dibuat untuk sistem *ticketing*. Dimana pada sistem *ticketing helpdesk* terdapat lima entitas yang saling terhubung dengan relasinya.

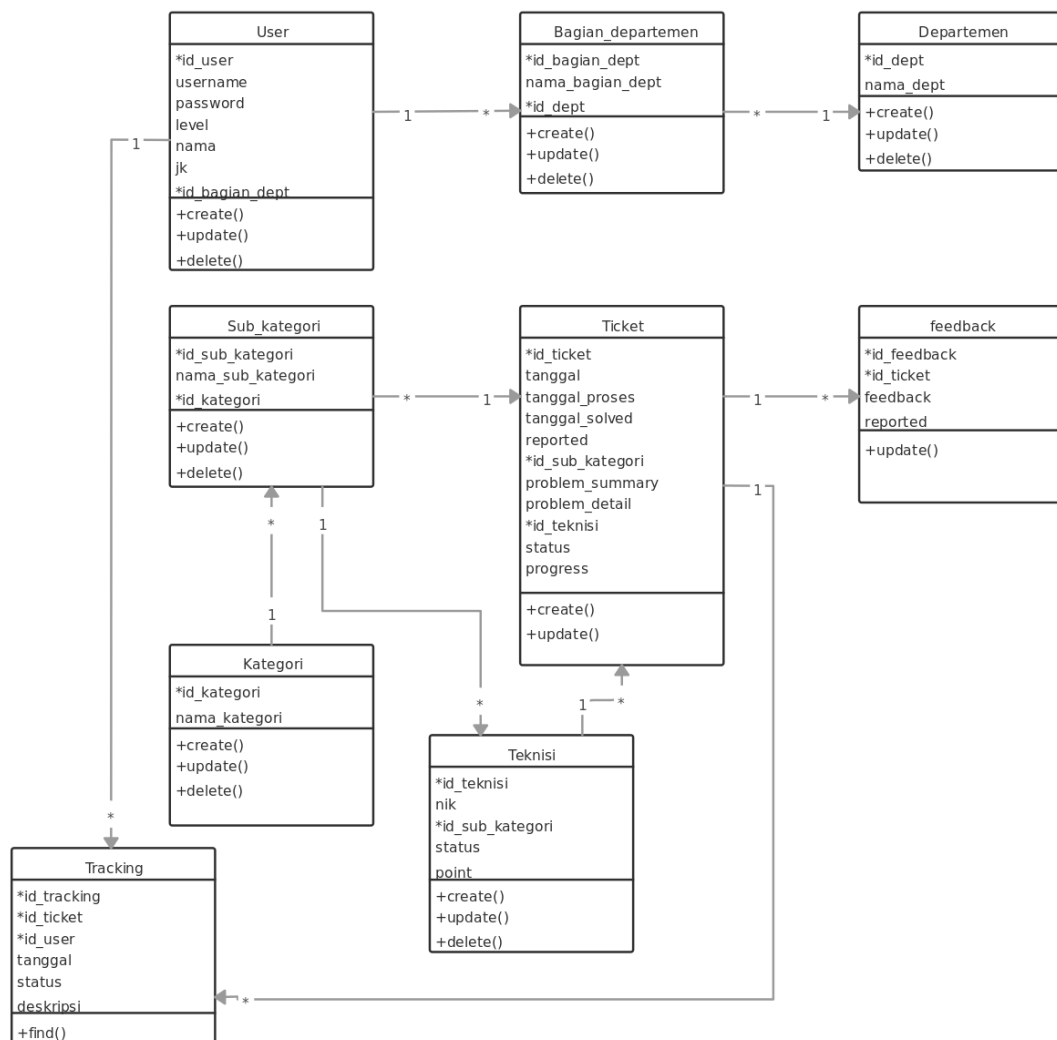


Gambar 3.3 Rancangan ERD

Pemodelan data pada ERD ini nantinya akan dikembangkan menjadi database pada sistem *ticketing helpdesk*.

### 3.2.2 Class Diagram

*Class diagram* digunakan untuk menggambarkan struktur sistem dari sisi pendefinisian kelas-kelas yang akan dibangun pada sistem. Pada Gambar 3.4 memperlihatkan rancangan *class diagram* untuk sistem *ticketing heldesk*. Dalam rancangan *class diagram* sudah ditentukan *property* apa saja yang akan berpengaruh terhadap jalannya sistem dari masing-masing *class* yang terbentuk.



Gambar 3.4 Class Diagram