

Bab ini akan membahas teori-teori terkait implementasi sistem informasi *helpdesk* yang bersumber dari jurnal, artikel dan buku yang bertujuan menjadi dasar keilmuan dalam menyelesaikan tugas akhir.

### **2.1 Institut Teknologi Kalimantan**

Disahkan pada 6 Oktober 2014 oleh Presiden yang menjabat saat itu, Institut Teknologi Kalimantan (ITK) resmi menjadi perguruan tinggi negeri di Balikpapan, Kalimantan Timur berlandaskan hukum Peraturan Presiden No. 125 Tahun 2014 tentang Pendirian Institut Teknologi Kalimantan (Lembaran Negara Republik Indonesia Tahun 2014 No. 254). Kegiatan akademik dimulai sejak penerimaan mahasiswa baru tahun 2012-2013 dan proses belajar mengajar yang dilakukan di Institut Teknologi Sepuluh Nopember (ITS). Saat ini ITK telah memiliki 17 program studi yang dibagi kedalam 5 jurusan. Adapun visi misi ITK ialah:

- Visi:  
Menjadi perguruan tinggi yang unggul dan berperan aktif dalam pembangunan Nasional melalui pemberdayaan potensi daerah Kalimantan pada tahun 2025.
- Misi:
  - ✓ Menyelenggarakan proses pendidikan tinggi yang berbasis pada penguasaan ilmu pengetahuan dan teknologi.
  - ✓ Berperan aktif dalam penelitian untuk menghasilkan inovasi proses dan produk sebagai upaya untuk memperkaya serta memperkuat ilmu pengetahuan dan teknologi.
  - Membangun kerjasama dan kontribusi pada pengabdian masyarakat yang didasarkan pada hasil penelitian dan potensi daerah untuk meningkatkan kesejahteraan masyarakat.

## 2.2 Manajemen Layanan TI

Manajemen layanan TI adalah konsep yang memungkinkan organisasi memaksimalkan nilai bisnis dari penggunaan teknologi informasi. Manajemen layanan TI juga merupakan sebuah penyediaan layanan pelanggan yang berkualitas dengan memastikan kebutuhan dan harapan pelanggan terpenuhi. Layanan memberikan nilai ke pelanggan dengan memfasilitasi, sehingga pelanggan mendapatkan kepuasan yang dia inginkan, tanpa harus mengeluarkan biaya dan menanggung resiko. Penerapan dan pengelolaan Layanan TI Berkualitas yang memenuhi kebutuhan Bisnis. (ITIL, 2015).

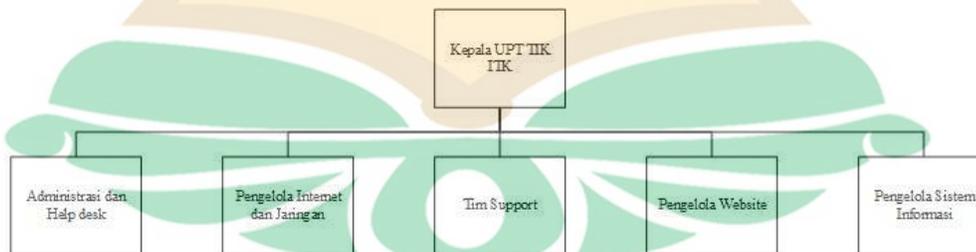
*Service operation* atau operasi layanan merupakan bagian dari manajemen layanan TI. Operasi Layanan adalah fase dalam siklus hidup manajemen layanan yang bertanggung jawab atas aktivitas bisnis normal. Operasi layanan dapat dipandang sebagai 'pabrik' TI. Ini menyiratkan fokus yang lebih dekat pada aktivitas dan infrastruktur sehari-hari yang digunakan untuk memberikan layanan. Tujuan utama dari operasi layanan adalah untuk memberikan dan mendukung layanan. Pengelolaan prasarana dan kegiatan operasional harus senantiasa mendukung tujuan ini (ITIL, 2015).

*Helpdesk* adalah unit fungsional yang terdiri dari sejumlah staf khusus yang bertanggung jawab untuk menangani berbagai acara layanan, sering kali dilakukan melalui panggilan telepon, antarmuka web, atau peristiwa infrastruktur yang dilaporkan secara otomatis (ITIL, 2015). *Helpdesk* merupakan *stage* awal yang dapat dihubungi oleh *end user* apabila mereka menjumpai masalah, *helpdesk* akan berupaya menyelesaikannya, namun bila tidak berhasil, *helpdesk* akan menugaskan kepada staf yang lebih ahli. *Helpdesk* akan mengkoordinir penanganan masalah. *End user* dapat menanyakan perkembangan aduan mereka pada *helpdesk*. Mereka dibatasi untuk langsung menghubungi divisi *support* spesialis. *Helpdesk* menjadi fasilitator komunikasi bagian IT dan pengguna lainnya, membuat prioritas pengerjaan masalah dan merespon aduan. Staf *helpdesk* harus memiliki pengetahuan yang luas agar masalah dapat dikelompokkan dan diberikan kepada tim solusi yang tepat. Diharapkan dengan menjadikan *helpdesk* titik utama pelaporan *user*, penanganan menjadi lebih terkoordinasi dan pengetahuan setelah solusi diimplementasikan dapat disimpan dengan baik. *Helpdesk* sebaiknya

didukung oleh aplikasi tertentu untuk membantu *tracking* pelaporan, insiden, dan eskalasi masalah. *Software* diharapkan dapat menyimpan pengetahuan dari solusi yang didapat, melakukan prioritas pengerjaan, serta pengkategorian masalah. *Helpdesk* yang baik sering kali dapat mengkompensasi kekurangan di tempat lain dalam organisasi TI, tetapi *Helpdesk* yang buruk (atau kurangnya *Helpdesk*) dapat memberikan kesan yang buruk tentang organisasi TI yang sangat efektif (ITIL, 2015). *Helpdesk* atau *ticketing* juga merujuk kepada sistem manajemen untuk membantu menangani kebutuhan konsumen terkait pelayanan, pertanyaan, aduan atau komplain dengan memanfaatkan sistem penomoran (*request ticket*) untuk memudahkan penelusuran terhadap tindakan penyelesaian yang dikoordinasi oleh suatu tim.

### 2.3 Unit Pelaksana Teknis Teknologi Informasi Dan Komunikasi

Unit Pelaksana Teknis Teknologi Informasi dan Komunikasi (UPT TIK) adalah unit pelaksana teknis di bidang pengembangan dan pengelolaan teknologi informasi dan komunikasi (NN, 2011). UPT TIK berfungsi untuk mendukung organisasi mencapai tujuannya dengan menunjang kebutuhan teknologi dan informasi. Divisi yang terdapat di dalam organisasi UPT ITK dapat disesuaikan sesuai kebutuhan organisasi dan jumlah anggota UPT TIK. Umumnya UPT TIK disusun atas Kepala UPT TIK, admin, *developer*, dan operator. Unit Pelaksana Teknis Teknologi Informasi dan Komunikasi Institut Teknologi Kalimantan (UPT TIK ITK) merupakan pelaksana teknis pengembangan dan pengelolaan TIK di lingkungan ITK.



Gambar 2.1 Struktur Anggota UPT TIK ITK

Struktur anggota UPT TIK ITK dapat dilihat dari Gambar 2.1 yang terdiri atas kepala UPT TIK, administrasi dan *helpdesk*, pengelola internet dan jaringan, pengelola *website*, dan pengelola sistem informasi.

**Tabel 2.1 Uraian Tugas Anggota UPT TIK ITK**

Posisi	Uraian Tugas
Kepala	<ul style="list-style-type: none"> <li>a. Mengambil keputusan</li> <li>b. Menyusun rencana, program dan anggaran UPT</li> <li>c. Berkomunikasi dengan civitas akademik mengenai keluhan, saran dan masukan pemanfaatan SI/TI</li> <li>d. Bertanggungjawab kepada pejabat ITK mengenai keputusan yang diambil di dalam UPT TIK</li> <li>e. Berkoordinasi dengan unit lain</li> </ul>
Administrasi dan <i>Helpdesk</i>	<ul style="list-style-type: none"> <li>a. Pelaksana administrasi UPT</li> <li>b. Menginventaris asset UPT TIK dan infrastruktur teknologi informasi ITK</li> <li>c. Berkoordinasi dengan unit lain</li> <li>d. Pelaksana perizinan peminjaman lab komputer</li> <li>e. Pelaksana pemberian layanan TIK kepada civitas akademik</li> <li>f. Menerima dan menyelesaikan pengaduan masalah terkait layanan UPT TIK</li> <li>g. Membuat laporan pelaksanaan kegiatan Administrasi dan <i>Helpdesk</i></li> <li>h. Melaksanakan tugas dari Kepala</li> </ul>
Pengelola Internet dan Jaringan	<ul style="list-style-type: none"> <li>a. Melakukan perancangan, pengelola dan pengembangan infrastruktur jaringan dan komputer</li> <li>b. Melakukan perancangan dan pengelolaan sistem keamanan dan perangkat jaringan</li> <li>c. Mengelola <i>bandwith</i>, <i>routing</i> dan <i>firewall</i> serta alat jaringan lainnya</li> <li>d. Merancang dan menganalisis kebutuhan jaringan komputer dan infrastruktur SI/TI</li> <li>e. Merancang prosedur pengamanan jaringan komputer dan infrastruktur SI/TI</li> </ul>

	<p>www.itk.ac.id</p> <p>f. Membangun jaringan komputer dan infrastruktur SI/TI</p> <p>g. Membangun dan mengelola koneksi internet</p> <p>h. Membuat laporan pelaksanaan kegiatan pengelolaan internet dan jaringan</p> <p>i. Melaksanakan tugas dari Kepala</p>
<i>Team Support</i>	<p>a. Melakukan ngawasan dan perawatan perangkat keras</p> <p>b. Melakukan perbaikan perangkat keras</p> <p>c. Melakukan pengawasan dan perawatan kelistrikan</p> <p>d. Melakukan perbaikan kelistrikan</p> <p>e. Menganalisis kebutuhan perangkat keras dan kelistrikan</p> <p>f. Membantu pengelola internet dan jaringan dalam menganalisis kebutuhan internet dan jaringan</p> <p>g. Membuat laporan pelaksanaan kegiatan <i>team support</i></p> <p>h. Melaksanakan tugas dari Kepala</p>
<i>Pengelola Website</i>	<p>a. Berkomunikasi dengan unit-unit lain dalam pemanfaatan dan kebutuhan <i>website</i></p> <p>b. Menyusun standar pengembangan <i>website</i></p> <p>c. Merencanakan pengembangan dan menganalisis kebutuhan <i>website</i></p> <p>d. Membangun <i>website</i></p> <p>e. Mengelola konten <i>website</i></p> <p>f. Memelihara dan melakukan pengembangan <i>website</i></p> <p>g. Memberikan bantuan teknis operasional <i>website</i> kepada unit lain</p> <p>h. Membuat laporan pelaksanaan kegiatan pengelolaan <i>website</i></p> <p>i. Melaksanakan tugas dari Kepala</p>
<i>Pengelola Sistem Informasi</i>	<p>a. Berkomunikasi dengan unit-unit lain dalam pemanfaatan dan kebutuhan sistem informasi</p> <p>b. Menyusun standar pengembangan sistem informasi dan pemanfaatan data</p> <p>www.itk.ac.id</p>

www.itk.ac.id	<ul style="list-style-type: none"> <li>c. Merencanakan pengembangan dan menganalisis kebutuhan sistem informasi</li> <li>d. Membangun sistem informasi</li> <li>e. Mengelola konten sistem informasi</li> <li>f. Memelihara dan melakukan pengembangan sistem informasi</li> <li>g. Melayani <i>request</i> data dan informasi dari unit kerja lain</li> <li>h. Memberikan bantuan teknis operasional sistem informasi</li> <li>i. Membuat laporan pelaksanaan pengelolaan sistem informasi</li> <li>j. Melaksanakan tugas dari Kepala</li> </ul>
---------------	---

Tabel 2.1 merupakan uraian tugas dari anggota UPT TIK ITK. UPT TIK menyediakan layanan kepada civitas akademik ITK berupa pembuatan *e-mail* baru dan reset *password e-mail, hosting* dan *subdomain*, peminjaman lab komputer, penambahan layanan, pengajuan layanan dan layanan internet. Selain layanan diatas, civitas akademik juga dapat melaporkan kendala maupun pertanyaan mengenai fasilitas TI ITK. UPT TIK ITK dapat dihubungi melalui tautan [ict.itk.ac.id](http://ict.itk.ac.id) atau *e-mail* [itsupport@itk.ac.id](mailto:itsupport@itk.ac.id).

#### 2.4 *Reuse-based Software Engineering*

Rekayasa perangkat lunak berbasis penggunaan kembali (*reuse-based*) merupakan strategi rekayasa perangkat lunak yang fokus kepada penggunaan kembali perangkat lunak yang sudah ada untuk mengembangkan perangkat yang dibutuhkan. Dengan perangkat lunak *reuse-based*, produksi dan pemeliharaan perangkat lunak menjadi lebih murah dan kualitas perangkat yang lebih baik. (Sommerville, 2016).

Perangkat lunak *reuse-based* adalah pendekatan pengembangan dengan memaksimalkan penggunaan kembali perangkat lunak yang ada. Terdapat berbagai macam *reuse-based software* dengan ukuran yang berbeda (Sommerville, 2016):

1. *System reuse* adalah sistem lengkap, yang mungkin terdiri dari beberapa program aplikasi, dapat digunakan kembali sebagai bagian dari sistem.

2. *Application reuse* adalah sebuah aplikasi yang dapat digunakan berulang kali dengan mengkonfigurasi aplikasi sesuai kebutuhan pelanggan.
3. *Component reuse*, adalah komponen aplikasi, baik objek tunggal maupun ukuran subsistem yang dapat digunakan kembali.
4. *Object and function reuse*, adalah komponen perangkat lunak yang mengimplementasikan fungsi tunggal, seperti kelas objek dapat digunakan kembali atau fungsi matematika. *Developer* menggunakan kembali *class* dan *function* di *library* dengan menautkannya dengan kode pada aplikasi yang akan dikembangkan.

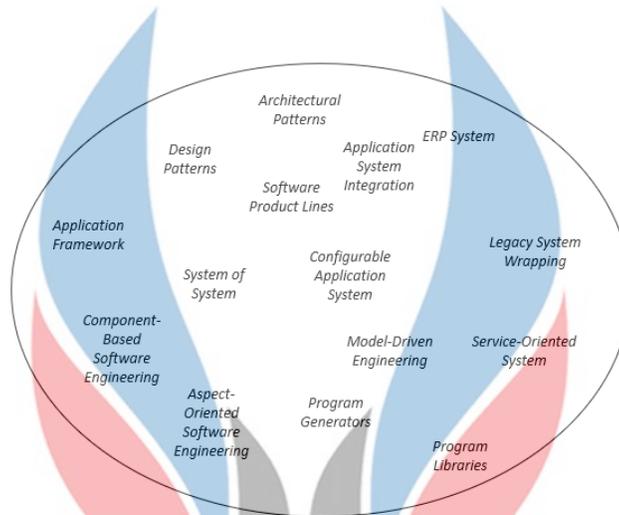
Selanjutnya, keuntungan dari perangkat lunak *reuse* ditunjukkan pada Tabel 2.2 berikut ini

**Tabel 2.2 Keuntungan Perangkat Lunak Reuse \*)**

<b>Keuntungan</b>	<b>Penjelasan</b>
<i>Effective use of specialists</i>	<i>Reusable software</i> memungkinkan <i>application specialist</i> terhindar dari membangun perangkat yang sama berulang-ulang.
<i>Lower development costs</i>	Semakin besar <i>software</i> yang dikembangkan maka semakin besar pula biaya yang harus dikeluarkan.
<i>Standards compliance</i>	Beberapa standar yang ada dapat digunakan sebagai sekumpulan komponen yang dapat digunakan kembali. Sehingga dengan menggunakan standar yang sudah ada dapat meningkatkan kepercayaan dan mengurangi kesalahan.
<i>Accelerated development</i>	Menggunakan kembali <i>software</i> yang ada dapat mempercepat produksi sistem dikarenakan mengurangi waktu pengembangan dan validasi sistem.
<i>Increased dependability</i>	<i>Reused software</i> , yang telah digunakan dan dicoba pada sistem, lebih dapat dipercaya dibandingkan <i>software</i> baru. Hal ini dikarenakan kesalahan desain dan implementasi sudah pasti ditemukan dan diperbaiki.
<i>Reduced process risk</i>	Biaya <i>software</i> yang sudah ada dapat diketahui sehingga mengurangi kemungkinan biaya yang membengkak apabila membuat sistem baru.

\*) (Sommerville, 2016)

Banyak teknik yang telah dikembangkan untuk mendukung penggunaan perangkat lunak *reuse-based*, dengan menitikberatkan fakta bahwa sistem dalam aplikasi yang sama adalah serupa dan memiliki potensi untuk digunakan kembali.



**Gambar 2.2 Landscape Konsep Reused-Based Software Engineering (Sommerville, 2016)**

Gambar 2.2 menetapkan sejumlah pendekatan yang dapat digunakan untuk menerapkan perangkat lunak *reuse-based*. Pendekatan diatas akan dijelaskan pada Tabel 2.3

**Tabel 2.3 Pendekatan Untuk Menerapkan Software Reuse-Based \*)**

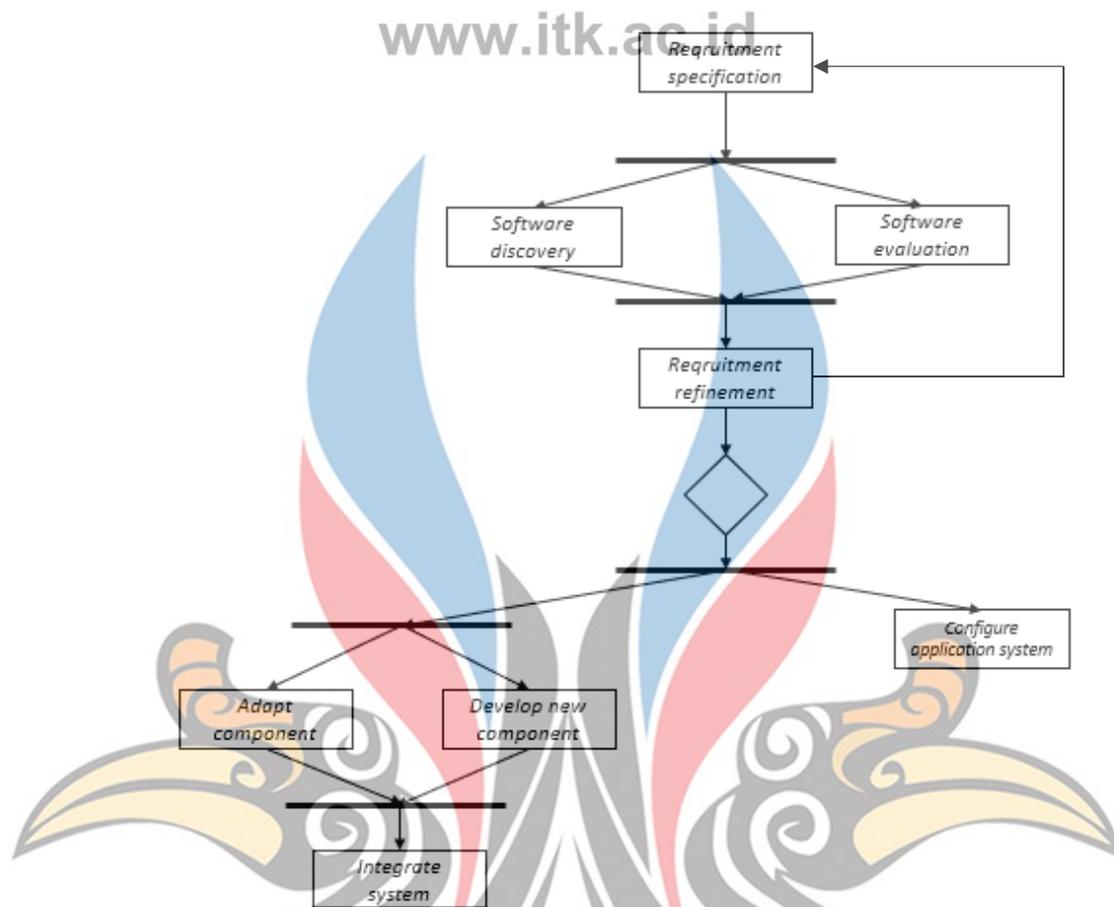
<b>Pendekatan</b>	<b>Deskripsi</b>
<i>Design pattern</i>	Abstraksi umum yang terdapat pada aplikasi direpresentasikan sebagai <i>design pattern</i> yang menunjukkan obyek abstrak dan konkrit serta interaksi yang ada
<i>Application frameworks</i>	Koleksi dari kelas abstrak dan konkrit yang diadaptasi dan dikembangkan untuk membuat sistem aplikasi
<i>Service-oriented system</i>	Sistem dikembangkan dengan menghubungkan layanan bersama, yang dapat disediakan oleh pihak eksternal
<i>System of system</i>	Dua atau lebih sistem terdistribusi yang terintegrasi untuk membentuk sistem baru.
<i>Component-based software engineering</i>	Sistem dibangun dengan mengintegrasikan komponen-komponen yang sesuai dengan standar komponen-model
<i>Aspect-oriented software engineering</i>	Komponen bersama disusun menjadi sebuah aplikasi di tempat berbeda ketika program dikompilasi

<i>Program generators</i>	Sebuah sistem generator menanamkan pengetahuan tentang jenis aplikasi dan digunakan untuk sistem di domain tersebut dari model sistem yang disediakan pengguna
<i>Model-driven engineering</i>	Perangkat lunak direpresentasikan sebagai <i>domain model</i> dan <i>implementation independent model</i> . Kode dihasilkan dari model tersebut.
<i>Configurable application system</i>	Sistem umum yang didesain sehingga bisa dikonfigurasi sesuai dengan kebutuhan pelanggan.
<i>Software product lines</i>	Sebuah tipe aplikasi yang digeneralisasikan di sekitar arsitektur umum sehingga dapat disesuaikan untuk pelanggan yang berbeda.
<i>Architectural patterns</i>	Arsitektur perangkat lunak standar yang mendukung jenis sistem aplikasi umum. Digunakan sebagai dasar aplikasi
<i>Application system integration</i>	Dua atau lebih sistem aplikasi terintegrasi untuk menyediakan fungsionalitas yang lebih besar.
<i>ERP system</i>	Sistem berskala besar yang meliputi fungsionalitas bisnis secara umum dan aturan yang dikonfigurasi untuk sebuah organisasi
<i>Legacy system wrapping</i>	Sekumpulan antarmuka dan menyediakan akses ke <i>legacy system</i> melalui antarmuka tersebut
<i>Program libraries</i>	<i>Library</i> dari <i>class</i> dan <i>function</i> yang mengimplementasikan abstraksi yang umum digunakan yang tersedia secara umum

\*) (Sommerville, 2016)

#### 2.4.1 Tahapan *Reuse-oriented Software Engineering*

Model proses umum untuk pengembangan *reuse-based* ditunjukkan pada Gambar 2.3. Jika dibandingkan dengan tahapan pengembangan rekayasa perangkat lunak lain, pengembangan *reuse-based* memiliki tahap awal dan tahap akhir yang serupa, namun memiliki perbedaan pada tahap di dalamnya (Sommerville, 2016).



**Gambar 2.3 Tahapan Reuse-oriented Software Engineering (Sommerville, 2016)**

Berikut adalah penjelasan tiap tahapan pada Gambar 2.3:

1. *Requirements specification*, persyaratan awal untuk sistem diusulkan, tidak harus diuraikan secara rinci tetapi harus mencakup singkat deskripsi persyaratan penting dan fitur sistem yang diinginkan.
2. *Software discovery and evaluation*, pencarian dilakukan untuk komponen dan sistem yang menyediakan fungsionalitas yang diperlukan. Komponen dan sistem kandidat dievaluasi untuk melihat apakah memenuhi persyaratan esensial dan apakah secara umum cocok untuk digunakan dalam sistem.
3. *Requirements refinement*, persyaratan disempurnakan menggunakan informasi tentang komponen yang dapat digunakan kembali dan aplikasi yang telah ditemukan.

4. *Application system configuration*, jika sistem aplikasi *off-the-shelf* memenuhi persyaratan yang tersedia, selanjutnya dapat dikonfigurasi untuk digunakan untuk membuat sistem baru.
5. *Component adaptation and integration*, jika tidak ada sistem *off-the-shelf*, komponen individu yang dapat digunakan kembali dapat dimodifikasi dan komponen baru dikembangkan, untuk kemudian diintegrasikan untuk membuat sistem

#### **2.4.2 COTS Product Reuse**

Produk aplikasi *reuse* adalah sistem perangkat lunak yang bisa dikonfigurasi sesuai dengan kebutuhan berbagai pengguna tanpa mengubah *source code* sistem. Sistem aplikasi dikembangkan oleh vendor sistem untuk pasar umum, dikembangkan secara khusus untuk pelanggan individu. Produk sistem ini dikenal sebagai produk *reuse* COTS (*commercial-off-the-shelf*). Perangkat lunak produk COTS dapat dengan mudah dijumpai pada berbagai macam produk *server* dan perangkat lunak *desktop*. Perangkat lunak COTS memiliki fungsi dan banyak fitur sehingga memiliki keunggulan untuk digunakan di *environment* yang berbeda atau menjadi bagian dari aplikasi lain. Hal ini karena produk ini didesain untuk penggunaan umum. Produk *open source* dapat dikategorikan sebagai perangkat lunak COTS karena produk *open source* digunakan tanpa mengganti atau melihat *source code* produk. Produk COTS diadaptasi dengan menggunakan mekanisme konfigurasi yang sudah ada yang memungkinkan fungsionalitas dari sistem dapat disesuaikan dengan kebutuhan spesifik pengguna (Sommerville, 2016).

#### **2.5 Open Source Software untuk Helpdesk**

Pengembangan *open source* adalah pendekatan pengembangan perangkat lunak di mana *source code* sistem perangkat lunak bersifat publik dan pengguna eksternal di dorong untuk berpartisipasi dalam pengembangan sistem. Melalui internet perekrutan sukarelawan menjadi lebih besar. Banyak dari mereka juga merupakan pengguna *code*. Pada prinsipnya setidaknya, setiap kontributor proyek *open-source* dapat melaporkan dan memperbaiki *bug* dan mengusulkan fitur dan

fungsionalitas baru. Namun, dalam praktiknya, sistem *open-source* masih bergantung pada kelompok *developer* inti yang mengontrol perubahan pada perangkat lunak. Sistem operasi Linux adalah sistem *server* yang paling banyak digunakan, seperti juga *server web* Apache *open source*. Produk *open-source* penting dan umum lainnya adalah Java, Eclipse IDE, dan sistem manajemen *database* MySQL (Sommerville, 2016).

Aplikasi *open source* biasanya murah atau bahkan gratis. Namun, jika pengguna menginginkan dokumentasi dan *support*, pengguna mungkin harus membayarnya, tetapi biayanya biasanya cukup rendah. Manfaat utama lainnya dari penggunaan produk *open-source* adalah bahwa sistem *open-source* yang banyak digunakan sangat andal. Aplikasi *open-source* memiliki populasi pengguna yang besar yang bersedia memperbaiki masalah sendiri daripada melaporkan masalah ini ke pengembang dan menunggu rilis baru dari sistem. *Bug* ditemukan dan diperbaiki lebih cepat.

Berbagai aplikasi tersedia dalam produk *open-source*, salah satunya aplikasi *helpdesk*. Tiga aplikasi *helpdesk open source* diantaranya adalah LiveAgent, UVdesk, dan osTicket. Tabel 2.4 merupakan perbandingan ketiganya berdasarkan kategori versi PHP, sistem operasi, sistem *database*, web server, serta *deployment*.

**Tabel 2.4 Perbandingan Tiga Open Source Helpdesk**

Kategori	osTicket	UVdesk	LiveAgent
<i>PHP version</i>	7.3 or higher	7.2 or higher	7.2 or higher
<i>Operating System</i>	Windows, Ubuntu	Windows, Ubuntu, Mac	Unix, Windows, Mac
<i>Database system</i>	MySQL, MariaDB	MySQL, MariaDB	MySQL PDO or MySQLi
<i>Web Server</i>	Apache, IIS, Nginx	Apache	Apache, Nginx, or any other which supports PHP
<i>Deployment</i>	Web-Based, Cloud, SaaS	Web-Based, Cloud, SaaS	Web-Based, Cloud, SaaS,

## 2.6 osTicket

*Helpdesk ticketing system* memungkinkan pelanggan untuk membuat dan memantau permintaan *support* sebagai tiket dan memungkinkan staf *helpdesk* untuk mengorganisir, mengatur dan merespon tiket-tiket. Tiket berisikan semua informasi terkait permintaan *support* pelanggan. osTicket adalah sistem tiket dukungan *open source* yang dirancang untuk meningkatkan efisiensi *customer support* dengan menyediakan *tools* yang dibutuhkan staf untuk memberikan dukungan yang cepat, efektif dan terukur. Sistem menyediakan pemberitahuan otomatis melalui email saat tiket dibuat, direspons, dll. Selain itu, sistem memungkinkan staf untuk menambahkan catatan internal ke tiket. Topik bantuan yang dapat dikonfigurasi, penugasan staf yang bertanggung jawab untuk setiap tiket, tanggal jatuh tempo, departemen, prioritas, dll. Terakhir, pelanggan dapat mengakses sistem untuk melacak status tiket mereka dan membalasnya (Pugibet, 2011).

osTicket merupakan produk dari Enhancesoft yang merupakan perusahaan perangkat lunak yang fokus pada menyediakan perangkat lunak helpdesk untuk meningkatkan pemberdayaan bisnis. Klien dari Enhancesoft antara lain Intel, Kobelco, G-W Publisher, dan *City National Rochdale*. osTicket memiliki 3 edisi, yaitu *open source*, *cloud-hosted*, dan *enterprise appliance*. Saat penelitian ini dilakukan, osTicket *open source* terbaru memiliki seri v1.15.1 dengan dokumentasi seri 1.15 (osTicket, 2020). Adapun fitur yang dimiliki oleh osTicket edisi *open source* adalah:

- a. *Custom field*, fitur ini memungkinkan admin helpdesk menyesuaikan formulir tiket sehingga memperoleh data yang dibutuhkan untuk membantu menyelesaikan masalah pengguna.
- b. *Ticket filters*, fitur ini memungkinkan admin helpdesk menentukan rute tiket masuk ke departemen atau agen yang tepat.
- c. *Help topics*, fitur ini merupakan pengelompokan jenis aduan. Sehingga tiket akan mudah dikelompokkan berdasarkan jenis aduan.

- d. *Assign, transfer, dan referral*, fitur ini akan memudahkan admin *helpdesk* untuk memastikan tiket ditangani dengan optimal dengan menentukan agen atau tim agen yang tepat. *Assign, transfer* dan *referral* akan dicatat pada utas tiket untuk dapat melacak apa yang terjadi pada tiket.
- e. *Costumer portal, end user dapat log in* menggunakan *e-mail* dan nomor tiket atau mereka dapat mendaftarkan profil untuk mendapatkan akses ke tiket yang mereka buat.
- f. *Ticket locking*, dengan fitur ini, tiket akan terkunci otomatis jika sedang dikerjakan oleh seorang agen, sehingga agen lain tidak bisa mengerjakannya. Fitur ini bertujuan untuk menghindari duplikasi agen.
- g. *Auto responder*, fitur ini memungkinkan untuk membuat respon otomatis apabila tiket baru saja dibuat.

Untuk meng-*install* osTicket, disarankan menggunakan PHP 5.6 (atau di atasnya) dan MySQL 5.0 (atau di atasnya)

## 2.7 Diagram Use Case

*Unified modeling language* (UML) diperkenalkan sebagai bahasa pemodelan *software* pada tahun 1997 dan telah menjadi bahasa baku (Larman, 2005). UML memodelkan sistem dengan pendekatan berorientasi objek. *Object Management Group* (OMG) merupakan institusi yang merilis konsep pemodelan spesifikasi UML untuk mendukung penggunaan UML yang tinggi dan terstandarisasi (OMG, 2017). Ada banyak *tools* dalam pembuatan UML antara lain UML Designer, StarUML, dan ArgoUML.

UML memiliki banyak diagram untuk menjelaskan pengembangan dinamis dan statik sistem (OMG, 2017). Satu diantara berbagai diagram yang kerap digunakan dalam mempresentasikan kebutuhan perangkat adalah diagram *use case* (UC). Diagram UC menjelaskan hubungan antara sistem dengan aktor. *Use case* menunjukkan fungsionalitas sistem dan aktor merupakan pelaku. Penjelasan detail dari interaksi *use case* dan aktor dijelaskan dalam *use case scenario* secara lengkap dan valid. Penjelasan secara kontekstual dalam UC *scenario* adalah pemodelan UC utama dan diagram UC adalah pelengkap. Pada kenyataannya, banyak pengembang

yang memandang sebelah mata tahap perencanaan pengembangan dan fokus pada pembangunan sistem (Pressman, 2020).

UC *scenario* menjelaskan sekumpulan skenario interaksi secara tekstual yang dapat ditampilkan dengan beberapa format, *brief*, *casual*, dan *fully dressed*. Bagian-bagian penting dalam UC *scenario* adalah aktor primer, prakondisi, alur utama, alur alternatif, dan kondisi akhir (Ramadhan, 2018). Pada aktor dan *use case* dapat ditambah penomoran tertentu untuk memudahkan identifikasi aktor dan *use case* pada diagram sistem yang cukup kompleks.

Diagram UC mempresentasikan visual interaksi antara sistem dan aktor. Aktor ditunjukkan dengan simbol *stick-man* dengan keterangan nama kata benda di bawahnya dan UC digambarkan dengan simbol elips dengan kata kerja aktif di dalamnya (Ramadhan, 2018). Aktor dan *use case* dapat memiliki spesifikasi relasi berbeda. *Base* UC dapat berelasi dengan UC lain atau disebut *supplier* UC dalam bentuk *include* atau *extend*. Relasi *include* dan *extend* tidak menjelaskan urutan eksekusi. *Extend* berarti fungsionalitas *base* UC diperluas oleh *supplier* UC. *Include* berarti fungsionalitas *base* UC hanya bisa dipenuhi dengan bantuan *supplier* UC. Aktor dikatakan aktor induk jika memiliki relasi *inheritance* dengan aktor lain, yang merupakan turunan dari aktor induk (Kurniawan, 2018).

Dari berbagai kekeliruan dalam membuat diagram *use case*, terdapat beberapa aspek yang perlu mendapat perhatian dalam menentukan relasi antar aktor dan relasi aktor dan *use case*. Terdapat beberapa kesalahan yang biasa terjadi dalam menginisiasi UC *log in* dan *log out* (Kurniawan, 2018):

- a. Kesalahan saat akan menggambarkan UC pertama memerlukan fungsionalitas dari UC kedua sebagai syarat menggunakan relasi *include*, sehingga diagram terkesan memiliki urutan eksekusi UC. Kesalahan ini sering terjadi pada *use case log in*. Rekomendasi perbaikannya adalah dengan penambahan aktor baru sebagai abstraksi pemakai perangkat lunak yang belum tervalidasi melalui proses *log in*.
- b. Kesalahan lain saat menetapkan aktor dari UC yang kurang sesuai dengan penjelasan UC *scenario*, tentang prakondisi dan kondisi akhir. Kekeliruan ini sering terjadi pada penentuan aktor *log in*. UC *log in* bertujuan untuk memvalidasi aktor yang mengakses sistem valid atau tidak. Sehingga

prakondisinya aktor belum dikenali sebagai aktor yang valid dan kondisi akhir yang harus terpenuhi adalah teridentifikasi sebagai aktor valid. Namun, pada kenyataannya masih banyak perancang menggambarkan UC *log in* dan *log out* berelasi dengan satu aktor yang sama. Rekomendasi perbaikannya adalah penggunaan aktor yang belum teridentifikasi untuk menginisiasi UC *log in*. Dan aktor yang telah teridentifikasi oleh sistem yang menginisiasi UC *log out*.

## 2.8 *User Acceptance Testing*

*User acceptance testing* atau pengujian penerimaan pengguna adalah jenis pengujian yang dilakukan untuk memverifikasi apakah produk dikembangkan sesuai standar dan kriteria yang ditentukan dan memenuhi semua persyaratan yang ditentukan oleh pelanggan. Jenis pengujian ini umumnya dilakukan oleh pengguna atau pelanggan di mana produk dikembangkan secara eksternal oleh pihak lain. *User acceptance testing* dianggap sebagai salah satu prosedur pengujian yang paling penting sebelum sistem akhirnya dikirim atau diserahkan kepada *end-user* (Mathur, 2011).

## 2.9 Pelatihan *End-user*

Pelatihan *end-user* dilakukan untuk membantu menciptakan *end-user* yang lebih produktif dan kompetitif. Pelatihan *end-user* sering kali berpusat pada masalah sehingga individu yang menerima pelatihan cenderung menemukan sistem teknologi yang lebih mudah digunakan. Selain itu, program yang menerapkan prinsip andragogi seperti menangani relevansi langsung dengan kinerja pekerjaan lebih mungkin untuk meningkatkan persepsi individu tentang kegunaan sistem dibandingkan mereka yang tidak menerima pelatihan (Marshall, 2008).

## 2.10 Penelitian Terdahulu

Tabel 2.6 adalah rangkuman hasil penelitian terdahulu yang memiliki keterkaitan dengan penelitian yang telah dilakukan.

Tabel 2.5 Penelitian Terdahulu

No	Judul	Permasalahan	Metode	Kesimpulan
1	Aplikasi <i>helpdesk</i> dalam Peran Layanan <i>Service-desk</i> pada PT SHS (Rachmawati, 2020)	PT. SHS mengalami kesulitan dalam menyediakan wadah untuk menampung <i>issue</i> yang terjadi pada piranti keras dan piranti lunak sistem yang telah diimplementasikan pada <i>user</i> .	Menggunakan metodologi berorientasi objek yaitu <i>iteration waterfall</i> .	Aplikasi <i>helpdesk</i> bertujuan untuk membantu <i>user</i> dan teknisi dalam mengkomunikasikan masalah yang berkaitan dengan aplikasi ERP dan pirantinya, dan untuk mendata masalah serta penanganannya, baik yang dapat dilakukan sendiri oleh <i>user</i> atau <i>teknisi</i> dari jarak jauh.
2	<i>Implemetation Helpdesk System Using Information Technology Infrastructure Library Framework on Software Company</i> (Girsang, 2018)	Dalam menjalankan layanan IT, PT. Boon Software masih menggunakan cara manual. <i>Service catalogue</i> tidak tersedia, <i>problem management</i> tanpa pemberian prioritas, belum ada <i>change management</i> , dan <i>service level agreements</i> yang tidak terdefinisi.	Mengimplementasi sistem <i>helpdesk</i> perusahaan dengan <i>information technology infrastructure library</i> (ITIL) <i>framework</i> sebagai standar referensi untuk mengatasi permasalahan yang berhubungan dengan layanan TI	Pengimplementasian ITIL menghasilkan: masalah dapat dikelompokkan dan dapat diidentifikasi <i>support teams</i> yang tepat, masalah dicatat dan diselesaikan secara sistematis sehingga staf tetap sesuai prosedur, setiap perubahan dapat dicatat dalam sistem dan akan mempengaruhi keseluruhan proses,

				peningkatan layanan yang disediakan kepada pengguna
3	<i>Analysis and Design Information System Development of IT Care Application in Astra Credit Company</i> (Sari Y. A., 2019)	ACC telah menggunakan IT <i>Helpdesk</i> . Namun seiring berkembangnya proses bisnis dan perubahan kebutuhan perusahaan, sistem tersebut tidak lagi dapat mendukung kebutuhan proses bisnis	Dilakukan proses <i>redesain</i> dan analisis dilakukan melalui metodologi <i>system development life cycle</i> (SDLC) dengan pendekatan <i>object oriented analysis and design</i> menggunakan <i>unified model language</i> (UML) <i>diagram</i> .	Analisis dan <i>re-design</i> menghasilkan penambahan fitur waktu pada <i>cart of account</i> , fitur <i>urgency level</i> dan <i>urgency point</i> , menentukan staf yang bertanggungjawab, peningkatan <i>re-attachment</i> pada <i>ticket</i> dan fitur <i>staff performance reports</i> dan <i>ticket completion reports</i>
4	Rancang Bangun Aplikasi <i>Hepldesk</i> dengan Menggunakan Pendekatan <i>Knowledge Management System</i> pada Seksi Teknisi PT. Indah Kiat Pulp & Paper Tbk. (Irawan, 2017)	Aduan kerusakan dengan <i>form reparation</i> , dan keterbatasan pengetahuan yang dimiliki oleh beberapa teknisi baru.	Pendekatan <i>knowledge management system</i> dengan <i>Waterfall</i> model, dirancang dengan diagram UML, bahasa pemrogramman PHP, dan <i>database</i> MySQL.	Perancangan aplikasi <i>helpdesk</i> menggunakan pendekatan yang digunakan yaitu <i>Knowledge Management System</i> .
5	Analisis Dan Perancangan <i>Helpdesk Ticketing</i>	Laporan tindak perbaikan yang dilakukan dicatat menggunakan	<i>Plan</i> , tahap <i>Do</i> , <i>Check</i> , dan <i>Action</i> . Pada penelitian ini	Usulan aplikasi <i>Helpdesk Ticketing System</i> menggunakan

	System Untuk Mengelola Tindak Perbaikan Perangkat Komputer Dan Jaringan Pada PT. Len Industri (Persero) Menggunakan Metodologi PDCA ( <i>Plan-Do-Check-Action</i> ) (Iswara, 2018)	kertas sehingga data akan rentan untuk rusak, hilang data dan penumpukan data.	metodologi PDCA yang digunakan hanya sampai pada tahap <i>Check</i> .	<i>open source</i> osTicket beserta prosedur dalam melakukan permintaan perbaikan perangkat komputer dan jaringan, tanggapan dari permintaan, dan laporan data tindak perbaikan.
6	Penerapan <i>Helpdesk Ticketing System</i> Dalam Penanganan Keluhan Penggunaan Sistem Informasi Berbasis <i>Web</i> (Baharudin, 2019)	<i>User</i> datang langsung ke kantor teknisi IT untuk melaporkan keluhannya, <i>user</i> juga bingung harus melapor kepada siapa ketika teknisi IT tidak berada ditempat sehingga laporan yang masuk tidak terdokumentasi dengan baik.	Pendekatan pengembangan sistem ini menggunakan metode <i>prototype</i> , dirancang memanfaatkan <i>web application</i> dan MySQL.	Aplikasi <i>Helpdesk Ticketing</i> menyampaikan teknisi dalam memberikan solusi dari laporan-laporan <i>user</i> , memudahkan <i>user</i> dalam melakukan laporan, menyediakan <i>self-service</i> , laporan menjadi lebih terorganisir.
7	Perancangan Aplikasi Sistem Helpdesk Berbasis Website Pada Direktorat Teknologi	Sering kali ditemukan berbagai masalah dalam penggunaan sistem informasi di lingkup kerja Unhas, yang	Tahapan penelitian yaitu pengumpulan data, membentuk kerangka berpikir, merancang kebutuhan	Unhas membutuhkan divisi <i>helpdesk</i> yang berfungsi sebagai bagian pertama yang harus dihubungi oleh <i>end user</i> bila

	Informasi Universitas Hasanuddin Makassar (Suaib, 2018)	apabila tidak ditangani akan menyebabkan kemunduran kerja.	aplikasi, dan desain sistem	mereka mengalami gangguan jaringan. Maka dari itu dibuat perancangan aplikasi <i>helpdesk</i> .
8	Rancang Bangun Sistem Informasi "IT Helpdesk" Berbasis Website Pada PT. Pelindo Marine Service (Syahputra, 2020)	Permasalahan pada perangkat ITK menyebabkan terhambatnya kegiatan operasional perusahaan. Penanganan perlu dilaksanakan secara cepat, tepat dan sesuai prioritas prosedur.	Aplikasi IT <i>Helpdesk</i> dibangun dengan metode AHP ( <i>analytical hierarchy process</i> ). Metode ini digunakan untuk menentukan prioritas penanganan utama dengan perhitungan atas nilai kriteria dan nilai alternatif	Aplikasi ini dapat membantu perusahaan dalam mengelola aduan dan meratakan beban kerja staf serta menangan seluruh proses prioritas penanganan aduan permasalahan perangkat TIK berdasarkan kriteria kerusakan.
9	Pembangunan Sistem Informasi <i>Helpdesk Ticketing System</i> Menggunakan <i>Django Framework</i> (Studi Kasus: SMK Saradan) (Mudhar, 2015)	Proses pencatatan aduan masih sederhana sehingga mempengaruhi kinerja staf tata usaha dan kurikulum. Berisiko duplikasi data.	Metode pengumpulan data ialah studi pustaka, observasi dan wawancara. Metode pembangunan menggunakan RAD yang dimodelkan dengan UML. Fase RAD ialah perencanaan syarat, perancangan, konstruksi dan pelaksanaan.	Penelitian ini menghasilkan sistem informasi <i>helpdesk</i> yang membantu staf tata usaha dan kurikulum dalam menyelesaikan masalah dan mendapatkan solusi dari aduan.

10	Implementasi Pengembangan <i>Smart Helpdesk</i> di UPT TIK UNS Menggunakan Algoritma <i>Naive Bayes Classifier</i> (Suryono, 2017)	Saat gangguan teknis, civitas mengajukan keluhan melalui telpon maupun datang secara langsung ke UPT TIK UNS. Kebanyakan aduan sama dan memiliki solusi yang hampir sama pula.	Tahap penelitian: analisa kebutuhan, perancangan, implementasi (Algoritma <i>Naive Bayes Classifier</i> dilakukan berdasarkan proses <i>supervised learning</i> ), uji coba, penarikan kesimpulan.	Pengembangan <i>smart helpdesk online</i> agar peran manusia atau operator dalam meminimalisir kesalahan klasifikasi yang dilakukan. Selain itu, peran operator sebagai perantara atau penyalur tiket tersebut bisa ditiadakan dengan adanya <i>smart helpdesk</i> tersebut.
11	Sistem Informasi IT <i>Helpdesk</i> pada Kejaksaaan Tinggi Jawa Tengah (Oktaviyani, 2019)	Di Kejaksaaan Tinggi Jateng, permasalahan yang dilaporkan masih diselesaikan secara manual.	Metode peneltian yaitu <i>waterfall</i> dengan tahap <i>requirement, design, implementation, verification</i> dan <i>maintenance</i> .	Dengan sistem Informasi IT <i>Helpdesk</i> , semua data disimpan terstruktur dan sistematis serta data dapat di akses dengan cepat.
12	Aplikasi Informasi Teknologi <i>Helpdesk</i> PT Survindo Dwi Putra Berbasis Intranet (A.N, 2015)	PT Survindo Dwi Putra belum memiliki aplikasi pengaduan keluhan terhadap gangguan informasi teknologi.	Menggunakan metode <i>waterfall</i> berurutan di mulai dari penentuan masalah, analisa kebutuhan, perancangan implementasi, integrasi, uji coba sistem, dan pemeliharaan.	Hasil penelitian adalah aplikasi ini dapat mengatasi jarak antara <i>user</i> dan teknisi serta dapat mengatasi pencarian tiket keluhan yang belum bisa diatasi secara manual sehingga dapat meningkatkan efisiensi waktu dan tenaga.

13	Rancang Bangun Aplikasi IT <i>Helpdesk</i> Berbasis Web Pada PT. Pupuk Sriwidjaja Palembang (Tahrirulwathann, 2015)	Tidak adanya prosedur pasti dalam proses pengaduan kepada departemen TI, proses penyelesaian masalah tidak beraturan, dan proses pendataan yang kurang baik.	Metode <i>waterfall</i> untuk pembangunan sistem dan mengadopsi metode Delbecq untuk menghitung prioritas masalah	Aplikasi dapat mengolah data pengaduan/keluhan beserta data penyelesaiannya, laporan aktifitas pelayanan keluhan serta laporan pendukung lainnya
14	Rancang Bangun Aplikasi <i>Helpdesk</i> (A-Desk) Berbasis Web Menggunakan <i>Framework</i> Laravel (Studi Kasus di PT PDAM Surya Sembada Kota Surabaya) (Mediana, 2018)	Data data pengaduan sering hilang bahkan rusak. Data yang masuk harus dianalisis terlebih dahulu agar perbaikan yang dilakukan sesuai dengan keahlian teknisi.	Metode yang digunakan adalah RAD dengan tahapan: <i>requirement planning, design system, dan implementation.</i>	Aplikasi <i>helpdesk</i> berhasil diimplementasikan dibuktikan dengan hasil kuisisioner kepuasan 60%-84%.
15	Teori dan Implementasi Sistem Informasi <i>Helpdesk</i> Berbasis Web (Ramadhan, 2018)	Saat ini proses penanganan masalah yang terjadi masih kurang baik karena proses penyampaian masalah melalui lisan sehingga petugas terkadang lupa untuk mencatat masalah yang	Metode penelitian yang digunakan adalah RAD.	Proses penanganan masalah yang terjadi masih kurang baik sehingga diusulkan untuk mengentri secara langsung ke dalam aplikasi <i>helpdesk</i> setiap keluhan yang masuk melalui fitur entri

dikeluhkan dan penanganan masalah menjadi tidak terkontrol.

masalah/keluhan. Aplikasi juga mempermudah proses penanganan masalah dengan fitur pengelompokan masalah.

Penelitian pertama adalah Aplikasi *Helpdesk* dalam Peran Layanan *Service-desk* Pada PT SHS yang ditulis oleh Arifah Rachmawati pada tahun 2020 dimana ditemukan permasalahan PT. SHS mengalami kesulitan dalam menyediakan wadah untuk menampung *issue* yang terjadi pada piranti keras dan piranti lunak sistem yang telah diimplementasikan pada *user*. PT SHS terkendala dalam mendokumentasi masalah-masalah pada piranti-piranti tersebut. Jika terdapat masalah yang berisiko tinggi mungkin saja tercatat secara berulang atau tidak tercatat sama sekali. Solusi yang ditawarkan adalah penerapan sistem *helpdesk*. Aplikasi *helpdesk* bertujuan untuk membantu pengguna dan teknisi dalam mengkomunikasikan masalah yang berkaitan dengan aplikasi ERP dan pirantinya, dan untuk mendata masalah serta penanganannya, baik yang dapat dilakukan sendiri oleh pengguna atau teknisi dari jarak jauh. Aplikasi akan dibangun menggunakan metode *waterfall*. (Rachmawati, 2020).

Selanjutnya, penelitian dari Girsang, dkk pada tahun 2018 dengan judul *Implemetation Helpdesk System Using Information Technology Infrastructure Library Framework on Software Company*. Permasalahan yang dihadapi adalah Dalam menjalankan layanan IT, PT. Boon *Software* masih menggunakan cara manual. *Service catalogue* tidak tersedia, *problem management* hanya merekam aduan tanpa pemberian prioritas, belum ada *change management* sehingga tidak diketahui masalah akan mempengaruhi layanan TI atau tidak, dan *service level agreements* yang tidak terdefinisi. Solusi yang diberikan ialah pengimplementasian ITIL menghasilkan: masalah dapat dikelompokan dan dapat diidentifikasi *support teams* yang tepat (*service catalogue*), masalah dicatat dan diselesaikan secara

sistematis sehingga staf tetap sesuai prosedur (*problem management*), setiap perubahan dapat dicatat dalam sistem dan akan mempengaruhi keseluruhan proses (*change management*), peningkatan layanan yang disediakan kepada pengguna (*service level agreement*) (Girsang, 2018).

Kemudian, penelitian dari Sari, dkk pada tahun 2019 dengan judul *Analysis and Design Information System Development of IT Care Application in Astra Credit Company*. Masalah yang dihadapi adalah Astra Credit Company (ACC) telah menggunakan IT Helpdesk. Namun, seiring berkembangnya proses bisnis dan perubahan kebutuhan perusahaan, sistem tersebut tidak lagi dapat mendukung kebutuhan proses bisnis. Untuk itu, penulis memberikan solusi untuk melakukan proses *redesain* dan analisis dilakukan melalui metodologi *system development life cycle* (SDLC) dengan pendekatan *object oriented analysis* and desain menggunakan *unified model language* (UML) diagram. Kesimpulan yang diperoleh adalah analisis dan *re-design* menghasilkan penambahan fitur waktu pada *cart of account*, fitur *urgency level* dan *urgency point*, menentukan staf yang bertanggungjawab, meningkatkan proses *re-attachment* pada *ticket* yang sudah selesai dan penambahan fitur *staff performance reports* dan *ticket completion reports* (Sari Y. A., 2019)

Irawan, dkk pada tahun 2017 melakukan penelitian berjudul Rancang Bangun Aplikasi *Helpdesk* dengan Menggunakan Pendekatan *Knowledge Management System* pada Seksi Teknisi PT. Indah Kiat Pulp & Paper Tbk. Masalah yang berusaha diselesaikan adalah karyawan memberikan laporan kerusakan tidak langsung kepada teknisi, harus mengisi *form reparation*, laporan kerusakan yang masuk secara bersamaan mengakibatkan teknisi kesulitan dalam menyesuaikan dengan teknisi yang tersedia dan keterbatasan pengetahuan beberapa teknisi baru. Untuk menghadapi masalah tersebut, penulis melakukan rancang bangun aplikasi *helpdesk* dengan *Waterfall model*, menggunakan pendekatan *knowledge management system*, dirancang dengan diagram UML, bahasa pemrograman PHP, dan *database MySQL*. Peran *knowledge management system* dalam aplikasi *helpdesk* untuk membantu penyebaran pengetahuan karyawan melalui sistem yang merekam setiap informasi yang diberikan oleh teknisi untuk dapat dipelajari dan digunakan oleh teknisi lain (Irawan, 2017).

Penelitian dari Iswara, dkk pada tahun 2018 berjudul Analisis Dan Perancangan *Helpdesk Ticketing System* Untuk Mengelola Tindak Perbaikan Perangkat Komputer Dan Jaringan Pada PT. Len Industri (Persero) Menggunakan Metodologi PDCA (*Plan-Do-Check-Action*). Pada PT. Len Industri (Persero), laporan tindak perbaikan yang dilakukan dicatat menggunakan kertas sehingga data akan rentan untuk hilang, rusak, dan penumpukan data. Untuk itu, penulis memberikan solusi untuk membangun aplikasi *helpdesk* menggunakan *open source* dan metodologi PDCA. Melalui metodologi ini, penulis melakukan perancangan pembangunan aplikasi karena penelitian hanya sampai pada tahap *check*. Selain perancangan, *output* dari penelitian ini adalah prosedur aduan, penyelesaian masalah dan laporan data tindak perbaikan. (Iswara, 2018).

Selanjutnya penelitian berjudul Penerapan *Helpdesk Ticketing System* Dalam Penanganan Keluhan Penggunaan Sistem Informasi Berbasis *Web* oleh Baharudin, dkk pada tahun 2019. Permasalahan yang dihadapi pada penelitian ini adalah *user* harus datang langsung ke kantor teknisi IT untuk melaporkan keluhannya, *user* juga bingung harus melapor kepada siapa ketika teknisi IT tidak berada ditempat sehingga laporan yang masuk tidak terdokumentasi dengan baik. Untuk itu, penulis mengusulkan untuk membangun aplikasi *Helpdesk Ticketing* membantu teknisi dalam memberikan solusi dari laporan-laporan *user*, memudahkan *user* dalam melakukan laporan, menyediakan *self-service*, laporan menjadi lebih terorganisir. Penelitian ini menggunakan metode *prototype*, dirancang memanfaatkan *web application* dan MySQL (Baharudin, 2019).

Kemudian, penelitian berjudul Perancangan Aplikasi Sistem *Helpdesk* Berbasis *Website* Pada Direktorat Teknologi Informasi Universitas Hasanuddin Makassar oleh Suaib, dkk tahun 2018. Sering kali ditemukan berbagai masalah dalam penggunaan sistem informasi di lingkup kerja Unhas, yang apabila tidak ditangani akan menyebabkan kemunduran kerja. Maka dari itu, diusulkan solusi membangun aplikasi *helpdesk* yang berfungsi sebagai divisi pertama yang perlu dihubungi oleh *end user* bila mereka mendapatkan masalah/gangguan jaringan. Maka dari itu dibuat perancangan aplikasi *helpdesk*. Tahapan penelitian ini yaitu pengumpulan data, membentuk kerangka berpikir, merancang kebutuhan aplikasi, dan desain sistem (Suaib, 2018).

Penelitian berjudul Rancang Bangun Sistem Informasi “IT Helpdesk” Berbasis Website Pada PT. Pelindo Marine Service oleh Syahputra, dkk pada tahun 2020 memiliki permasalahan pada perangkat IT menyebabkan terhambatnya kegiatan operasional perusahaan. Penanganan perlu dilaksanakan secara cepat, tepat dan sesuai prioritas prosedur. Solusi yang diberikan adalah rancang bangun aplikasi IT helpdesk dibangun dengan metode AHP (*analytical hierarchy process*). Metode ini digunakan untuk menentukan prioritas penanganan utama dengan perhitungan atas nilai kriteria dan nilai alternatif. Aplikasi ini dapat membantu perusahaan dalam mengelola aduan dan meratakan beban kerja staf serta menangan seluruh proses prioritas penanganan aduan permasalahan perangkat TI berdasarkan kriteria kerusakan (Syahputra, 2020).

Penelitian berikutnya berjudul Pembangunan Sistem Informasi Helpdesk Ticketing System Menggunakan Django Framework (Studi Kasus: SMK Saradan) oleh Mudhar pada tahun 2015. Masalah yang diangkat adalah proses pencatatan aduan masih sederhana sehingga mempengaruhi kinerja staf tata usaha dan kurikulum dan berisiko duplikasi data. Penelitian ini menghasilkan sistem informasi helpdesk yang membantu staf tata usaha dan kurikulum dalam menyelesaikan masalah dan mendapatkan solusi dari aduan (Mudhar, 2015).

Penelitian berjudul Implementasi Pengembangan Smart Helpdesk di UPT TIK UNS Menggunakan Algoritma Naive Bayes Classifier oleh Suryono, dkk pada tahun 2017 mengangkat permasalahan saat gangguan teknis yang terjadi tidak terdeteksi oleh UPT TIK UNS, civitas sering mengajukan keluhan melalui telepon maupun datang secara langsung ke UPT TIK UNS. Kebanyakan aduan sama dan memiliki solusi yang hampir sama pula. Pencatatan keluhan kurang mendetail sehingga dokumentasi keluhan dan masalah kerap tidak lengkap. Solusi yang diberikan adalah pengembangan smart helpdesk online. Melalui sistem ini, peran operator sebagai perantara atau penyalur tiket bisa ditiadakan. Tahap penelitian: analisa kebutuhan, perancangan, implementasi (Algoritma Naive Bayes Classifier dilakukan berdasarkan proses supervised learning), uji coba, penarikan kesimpulan (Suryono, 2017).

Selanjutnya penelitian berjudul Sistem Informasi IT Helpdesk pada Kejaksaan Tinggi Jawa Tengah oleh Oktaviyani, dkk pada 2019 dengan permasalahan saat ini

di Kejaksaan Tinggi Jateng, penyelesaian masalah diselesaikan secara manual dan sistem pengontrolan terhadap keluhan yang dilaporkan tidak dapat dilakukan secara cepat. Solusi yang diberikan adalah penggunaan sistem informasi IT *helpdesk* yang dapat menyimpan data dengan sistematis, terstruktur dan dapat diakses dengan cepat (Oktaviyani, 2019).

Penelitian berjudul Aplikasi Informasi Teknologi *Helpdesk* PT Survindo Dwi Putra Berbasis Intranet oleh A.N Pradana pada tahun 2015 mengangkat permasalahan PT Survindo Dwi Putra belum memiliki aplikasi pengaduan keluhan terhadap gangguan informasi teknologi. Solusi yang diberikan adalah membangun aplikasi *helpdesk* dengan menggunakan metode *waterfall* berurutan di mulai dari penentuan masalah, analisa kebutuhan, perancangan implementasi, integrasi, uji coba sistem, dan pemeliharaan. Hasil penelitian adalah aplikasi dapat mengatasi jarak antara pengguna dan teknisi serta dapat mengatasi pencarian tiket keluhan yang belum bisa diatasi secara manual (A.N, 2015).

Kemudian penelitian dari Tias Tahrirulwathan pada tahun 2015 dengan judul Rancang Bangun Aplikasi IT *Helpdesk* Berbasis Web Pada PT. Pupuk Sriwidjaja Palembang. Permasalahan yang diangkat adalah manajemen pelayanan pengaduan masalah yang ada saat ini terbilang kurang baik, hal ini dapat dilihat dari tidak adanya prosedur pasti dalam proses pengaduan kepada departemen TI. Selain itu juga, proses penyelesaian masalah juga menjadi tidak beraturan karena tidak adanya sistem antrian. Laporan kepada pihak atasan juga menjadi tidak lengkap karena proses pendataan yang kurang baik. Untuk itu, solusi yang diberikan adalah pembangunan aplikasi IT *Helpdesk* berbasis *website* pada PT. Pupuk Sriwidjaja Palembang yang diharapkan dapat membantu manajemen proses pelayanan pengaduan/keluhan mengenai masalah-masalah operasional SI dan infrastruktur TI. Metode pengembangan yang digunakan adalah *waterfall* dan metode *Delbecq* yang merupakan metode untuk menghitung prioritas masalah pada saat menentukan urutan antrian keluhan yang akan diselesaikan. Hasil dari penelitian ini adalah aplikasi dapat mengolah data pengaduan/keluhan beserta data penyelesaiannya dan juga dapat membuat laporan aktifitas pelayanan keluhan serta laporan pendukung lainnya yang dapat menjadi pertimbangan manajer dalam memperhatikan kinerja

hardware. Selain itu aplikasi membuat sistem antrian keluhan menjadi lebih teratur (Tahrirulwathan, 2015).

Selanjutnya penelitian dari Mediana dengan judul Rancang Bangun Aplikasi *Helpdesk* (A-Desk) Berbasis Web Menggunakan *Framework* Laravel (Studi Kasus di PT PDAM Surya Sembada Kota Surabaya) pada 2018. Masalah yang diangkat dalam penelitian ini adalah dalam pelayanan aduan, data-data pengaduan sering hilang bahkan rusak. Data yang masuk harus dianalisis terlebih dahulu agar perbaikan yang dilakukan sesuai dengan keahlian teknisi. Solusi yang diberikan adalah implementasi aplikasi *helpdesk* yang diberi nama A-desk. Aplikasi akan dibangun menggunakan metode *rapid application development* (RAD), *framework* Laravel, dan basis data *progreSQL*. Hasil dari implementasi adalah didapat kesesuaian dari pengguna TI dengan rata-rata jawaban adalah 60%-80% dengan interval perolehan kuisioner termasuk kategori setuju (Mediana, 2018).

Penelitian terakhir berjudul Teori dan Implementasi Sistem Informasi *Helpdesk* Berbasis Web oleh Ramadhan tahun 2018. Saat ini proses penanganan masalah pada PT. Aseanindo kurang baik karena proses penyampaian masalah melalui lisan sehingga petugas kerap lupa mencatat masalah yang dilaporkan dan penanganan masalah menjadi tidak terkontrol. Adapun tujuan perlunya sistem *helpdesk* agar memudahkan *helpdesk* untuk mencatat masalah yang diterima dan menugaskan staf sesuai dengan masalah yang diterima. Metode yang digunakan adalah *rapid application development* (RAD). Metode pengumpulan data adalah wawancara, analisa dokumen, observasi, dan studi perpustakaan. Sistem menggunakan *database MySQL* dan bahasa pemrograman PHP (Ramadhan, 2018).

Berdasarkan permasalahan yang diangkat dalam penelitian ini dan juga pengetahuan-pengetahuan dari penelitian sebelumnya, maka solusi dari penelitian ini adalah mengimplementasikan *helpdesk ticketing system* ITK IT Support pada UPT TIK Institut Teknologi Kalimantan menggunakan *software open source* osTicket. Adapun metodologi yang digunakan adalah adaptasi dari *reuse-based software engineering* dengan tahap-tahap sebagai berikut, yaitu persiapan, studi literatur, identifikasi masalah, *requirement specification*, *selection and procurement*, *system design*, *development and integration*, *system validation*, *training*, serta penarikan kesimpulan dan saran.