

# RANCANG BANGUN APLIKASI JEMPUT DAN ANTAR (*PICKUP AND RETURN*) BERBASIS ANDROID PADA PT. SINAR ELOK ABADI

Minda Mora Purba, Amin Nurdin

karo\_727@yahoo.com, aminndn@yahoo.co.id

## ABSTRAK

Jemput dan antar merupakan salah satu layanan yang ditawarkan dalam bisnis dibidang penjualan. Jemput dan antar juga menjadi suatu pilihan dalam layanan perbaikan. Layanan ini tentu memerlukan sistem yang bisa memudahkan serta mempercepat dalam proses pelaksanaannya. Pada PT. Sinar Elok Abadi layanan jemput dan antar diberikan untuk membuat pelanggan bisa mudah dalam proses perbaikan komputer serta printer mereka. Namun prosesnya masih dilakukan dengan cara konvensional, memerlukan beberapa tahapan seperti panggilan telpon kepada operator yang terbatas jumlahnya dan pencatatan oleh operator itu sendiri untuk permintaan penjemputan dan pengantaran barang. Permasalahan itu memunculkan gagasan untuk membuat aplikasi yang bisa diakses melalui telepon genggam sehingga mempermudah pelaksanaan layanan ini. Aplikasi ini dibuat agar pelanggan bisa langsung melakukan permintaan jemput barang dan admin akan mengatur jadwal penjemputan. Tentu saja aplikasi ini juga dibuat untuk mempermudah kurir dalam melaksanakan proses penjemputan atau pengantaran karena semua bisa dilihat dalam aplikasi. Aplikasi berjalan dalam *operating system* (OS) Android.

Kata Kunci: aplikasi, jemput dan antar, android, *pickup and return*

## 1. PENDAHULUAN

### 1.1 Latar belakang masalah

Kecanggihan teknologi yang terus berkembang pada saat ini, memaksa manusia untuk mengikutinya. Bahkan dirasa tidak akan cukup jika hanya mengikutinya, kita sebagai manusia juga harus mampu mengembangkannya agar tidak tertinggal dengan teknologi itu sendiri. Hampir semua hal telah menggunakan teknologi komputerisasi, itu digunakan untuk mempermudah pekerjaan. Dalam perkembangannya pula, muncul teknologi dalam bidang ponsel pintar atau *smartphone* terutama dalam sistem operasi Android. Hal itu semakin mempermudah pekerjaan yang telah terhubung dengan komputer dan diaplikasikan kedalam genggamannya melalui *smartphone*.

Perkembangan teknologi *smart-*

*phone* Android, mulai banyak dilirik oleh para pengusaha atau kalangan pebisnis pemula. Sistem aplikasi Android yang sedang berkembang dirasa cukup membuat pekerjaan semakin mudah, tanpa harus membawa sebuah *Personal Computer* (PC) atau Notebook, ada banyak pekerjaan yang bisa dilakukan hanya lewat layar *smartphone* kita.

Perseroan Terbatas (PT.) Sinar Elok Abadi (SEA) adalah sebuah perusahaan yang bergerak di bidang jasa *service* untuk produk Hewlett-Packard (HP). Perusahaan yang berdiri sejak 2014 ini sedang berusaha meningkatkan layanan bagi para *customer*. Salah satu layanan terbaru PT. SEA adalah *Pick Up and Return* (PUR). Dimana pengguna produk HP berupa notebook atau printer *Next Gen* yang ada di daerah Jakarta bisa menggunakan layanan ini dengan syarat

unit masih garansi (1 tahun setelah pembelian). PUR ini adalah layanan dimana pengguna produk HP (*user/customer*) bisa melakukan perbaikan tanpa harus datang ke tempat *service center*, *user* hanya perlu menghubungi *customer service* lalu kurir dari PT. SEA akan datang ke rumah untuk mengambil unitnya dan akan diantar kembali saat unit selesai diperbaiki. Untuk mempermudah layanan tersebut maka diperlukan adanya sistem yang bisa lebih mempermudah proses, baik bagi pelanggan maupun bagi pihak PT. SEA itu sendiri. Jika ada sebuah aplikasi Android yang bisa di-unduh untuk layanan ini, maka *user* tidak perlu lagi menghubungi dan menunggu konfirmasi kedatangan kurir, semua bisa dilihat dan dikerjakan melalui genggaman di *smartphone*.

Berdasarkan penjelasan pada latar belakang diatas, maka dilakukan penelitian untuk membangun suatu aplikasi android untuk antar dan jemput.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dijabarkan di atas, maka didapatkan rumusan masalah yang akan dibahas adalah sebagai berikut:

- a. Bagaimana merancang aplikasi yang bisa digunakan sebagai layanan terbaru berupa *Pick Up and Return* (PUR) di PT. SEA?
- b. Apakah aplikasi ini dapat mempermudah *customer* dalam menggunakan layanan ini?
- c. Apakah aplikasi ini mempermudah bagi PT. SEA itu sendiri?

## 1.3 Batasan Masalah

Batasan masalah agar pembahasan dan analisa yang dilakukan dapat terarah adalah sebagai berikut:

1. Sebagai dasar pembangunan aplikasi untuk layanan PUR.
2. Aplikasi ini akan terhubung dengan

sistem yang akan dijalankan melalui web oleh admin.

3. Sistem aplikasi yang ada akan digunakan untuk kurir dan juga customer.

## 1.4 Tujuan Penelitian

Ada beberapa tujuan yang ingin dicapai dalam penulisan penelitian pada PT. SEA sebagai berikut:

1. Mengetahui proses layanan PUR yang ada di PT. SEA.
2. Membangun aplikasi PUR guna memperbaiki sistem manual dimana *user/customer* harus menghubungi kemudian admin meminta kurir mengambil dari alamat yang disebutkan *user* dengan aplikasi yang bisa langsung diakses melalui *smartphone*.

## 2. LANDASAN TEORI

### 2.1 Pengertian Perancangan

Menurut Jogiyanto, perancangan adalah tahap setelah analisis dari siklus pengembangan sistem yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem<sup>1</sup>.

Sedangkan menurut Al-Bahra, perancangan adalah suatu kegiatan yang memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan *alternative* sistem yang baik<sup>2</sup>.

---

<sup>1</sup> Jogiyanto, (2005 : 196). Analisis dan Desain

<sup>2</sup> Al Bahri, (2005 : 39). Analisis dan Desain Sistem Informasi

Dari definisi diatas dapat disimpulkan bahwa perancangan merupakan tahap lanjut dari analisis sistem untuk membuat sebuah rancangan sistem baru yang lebih baik dari sebelumnya.

## 2.2 Pengertian Aplikasi

Aplikasi adalah program siap pakai yang dibuat oleh perusahaan komputer untuk para pemakai yang telah dirancang untuk membuat pengguna lebih produktif. Hal ini disimpulkan menurut pengertian dari beberapa ahli diantaranya menurut Sutarman yang mengatakan aplikasi merupakan program-program yang dibuat oleh suatu perusahaan komputer untuk para pemakai yang beroperasi dalam bidang umum, seperti pertokoan, komunikasi, penerbangan, perdagangan, dan sebagainya<sup>3</sup>.

Hendrayudi juga mengungkapkan aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu<sup>4</sup>.

## 2.3 Pengertian Pick Up and Return (Jemput dan Antar)

Jemput dan Antar (Pick Up and Return) adalah sebuah layanan yang ditawarkan oleh sebuah perusahaan baik dibidang penjualan barang atau jasa dimana *customer* melakukan pemesanan barang atau jasa melalui sebuah aplikasi atau telepon.

Layanan jemput dan antar biasanya digunakan oleh perusahaan jasa, karena layanan ini tidak berhubungan dengan penjualan barang sehingga tak perlu mengantarkan mengambil barang dan mengembalikannya lagi.

Layanan seperti ini mulai banyak diminati, karena mempunyai beberapa keuntungan diantaranya:

1. Mempermudah *customer* dalam mendapatkan barang atau jasa yang dibutuhkan tanpa perlu datang ke tempat barang atau jasa tersebut ada.
2. Menjadikan pihak perusahaan (barang atau jasa) baik dimata *customer* dan mampu meningkatkan daya jual produk atau jasa mereka.

## 2.4 Konsep Pemrograman

Secara umum pemrograman sendiri adalah kumpulan urutan perintah ke komputer untuk mengerjakan sesuatu, dimana perintah tersebut menggunakan bahasa yang dimengerti oleh komputer.

### 2.4.1 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek merupakan suatu pendekatan pemrograman yang menggunakan objek dan *class*<sup>5</sup>.

Dalam pemrograman beorientasi objek, ada beberapa hal yang harus diketahui yaitu antara lain:

1. *Class*, merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. *Class* yaitu template untuk membuat objek. *Class* merupakan prototype atau *blue print* yang mendefinisikan variable-variabel dan *method-method* secara umum. Objek merupakan hasil *instances* dari suatu *class*. Proses pembentukan objek dari suatu *class* disebut *instances*.
2. *Object*, merupakan segala sesuatu yang dapat dibedakan satu sama lainnya.
3. *Atribut*, adalah data yang membedakan antara objek satu dengan lainnya.
4. *Method, Behavior* atau tingkah laku

---

<sup>3</sup> Sutarman, (2009 : 147). Pengantar Teknologi

<sup>4</sup> Hendrayudi, (2009 : 143). VB 2008 untuk keperluan pemrograman

---

<sup>5</sup>

[http://repo.unnes.ac.id/dokumen/pemrograman-web/pw2\\_pertemuan10.pdf](http://repo.unnes.ac.id/dokumen/pemrograman-web/pw2_pertemuan10.pdf)

adalah hal-hal yang bisa dilakukan oleh objek dari suatu *class*. *Behavior* dapat digunakan untuk mengubah nilai atribut objek, menerima dari objek lain dan mengirim informasi ke objek lain untuk melakukan suatu tugas (*task*). Dalam *class*, *behavior* disebut juga *method*. *Method* sendiri adalah serangkaian pernyataan suatu *class* yang mempunyai suatu *task* tertentu. Cara objek berkomunikasi dengan objek lain dengan menggunakan *method*.

5. Enkapsulasi, adalah pembungkusan *variable* dan *method* dalam sebuah objek yang terlindungi serta menyediakan *interface* untuk mengakses *variable* tersebut. *Variable* dan *method* yang dimiliki oleh suatu objek bisa ditentukan hak aksesnya. Enkapsulasi berfungsi untuk memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam/dari sebuah objek dengan tidak layak, hanya *method* objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses *interface* yang menyebutkan bagai-mana objek lainnya dapat ber-interaksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.
6. Pewaris (*inheritance*), merupakan pewarisan atribut dan *method* dari sebuah *class* ke *class* lainnya. Secara singkat bisa diartikan teknik yang menyatakan bahwa anak dari objek akan mewarisi *data*/atribut dan *method* dari induknya langsung.
7. Polimorfisme, adalah kemampuan suatu objek untuk mempunyai lebih dari satu bentuk. Polimorfisme tidak tergantung pada pemanggilan sub rutin. Metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu dimana pesan tersebut dikirim.

#### 2.4.2 Diagram Unified Modelling Language (UML)

UML adalah sebuah bahasa berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis objek<sup>6</sup>.

Dalam hal ini, UML bukanlah merupakan bahasa pemrograman tetapi model-model yang tercipta berhubungan langsung dengan berbagai macam bahasa pemrograman, sehingga mungkin melakukan pemetaan (*mapping*) langsung dari model-model yang dibuat dengan UML dengan bahasa-bahasa pemrograman berorientasi objek seperti Java, Borlan Delphi, Visual Basic, c++ dan lain-lain.

Diagram UML adalah representasi grafis parsial (*view*) dari model sistem yang sedang didesain, tahap implementasi, atau sudah ada. Diagram UML berisi elemen grafis (*simbol*). UML terhubung dengan tepi (juga dikenal sebagai jalur atau arus) yang mewakili elemen dalam model UML dari sistem yang dirancang. Model UML dari sistem mungkin juga berisi dokumentasi lain seperti *use case* yang ditulis sebagai teks<sup>7</sup>.

Jenis diagram didefinisikan oleh simbol grafis utama yang ditunjukkan pada diagram. Misalnya, diagram di mana simbol utama di area isi adalah kelas adalah diagram kelas. Diagram yang menunjukkan *use case* dan actor adalah *use case diagram*. Diagram urutan menunjukkan urutan pertukaran pesan antara jalur kehidupan (*lifeline*).

Spesifikasi dari diagram UML

---

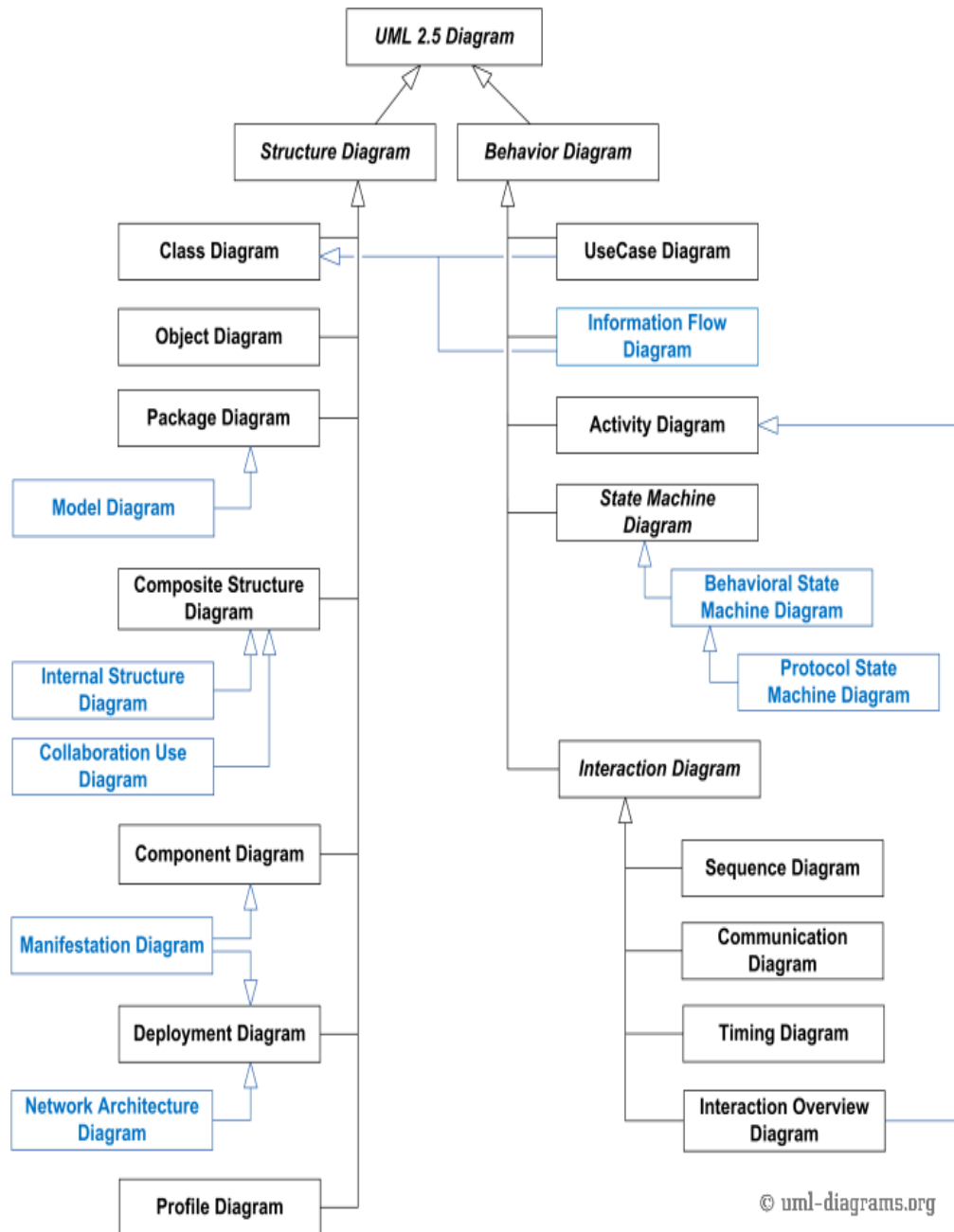
<sup>6</sup> Pengertian UML.

<http://informatika.web.id/pengertian-uml.htm>

<sup>7</sup> UML Diagram. <http://www.uml-diagrams.org/uml-25-diagrams.html>

mendefinisikan menjadi dua jenis diagram, yaitu diagram struktur (*structure diagram*) dan diagram perilaku (*behavior diagram*).

Diagram UML (2.5) bisa dikategorikan secara hierarkis seperti gambar di bawah ini:



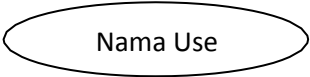
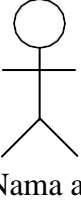
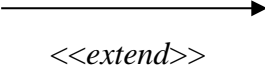
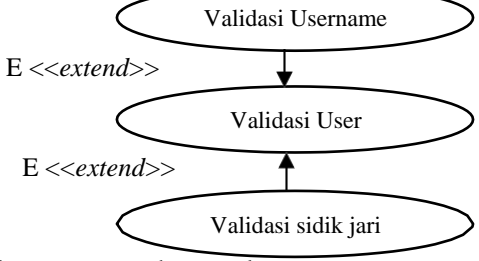
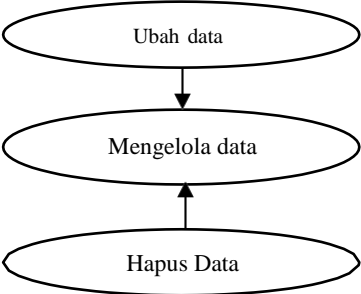
Gambar II. 1: UML Diagram

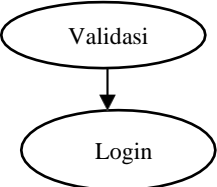
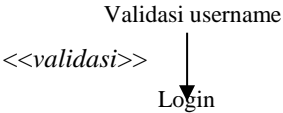
### 2.4.2.1 Use Case Diagram

*Use case* diagram atau dalam bahasa Indonesia dikenal dengan diagram use case merupakan salah satu diagram yang ada dalam UML (Unified Modeling Language) yang

digunakan untuk memodelkan aspek perilaku sistem dari sistem yang akan dibuat dan untuk merekam persyaratan fungsional sebuah sistem.

Tabel II. 1 : Simbol *use case diagram*

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsi-fungsi/proses-proses yang disediakan aplikasi sebagai uni-unit yang saling bertukar pesan antar unit atau aktor, biasanya diawali dengan menggunakan kata kerja diawal frase nama <i>use case</i>.</p>
<p>Aktor / <i>actor</i></p> 	<p>Orang, proses atau aplikasi lain yang berinteraksi dengan aplikasi yang akan dibuat itu sendiri. Jadi walaupun symbol dari aktor adalah gambar orang tapi aktor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Ektensi/<i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu.. Biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan. Misal:</p>  <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan</p>
<p>Generalisasi/<i>Generalization</i></p>	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p> 






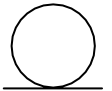
	Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)
<p>Menggunakan <i>include / uses</i></p> <p>-----▶</p> <p>————▶</p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p> <p>Adapun dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> adalah:</p> <ul style="list-style-type: none"> <li>• <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misalnya:</li> </ul> <div style="text-align: center;">  <pre> graph TD     Validasi((Validasi)) -- "&lt;&lt;include" --&gt; Login((Login)) </pre> </div> <ul style="list-style-type: none"> <li>• <i>Include</i> berarti <i>use case</i> tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misalnya:</li> </ul> <div style="text-align: center;">  <pre> graph TD     Validasi[Validasi username] -- "&lt;&lt;validasi&gt;&gt;" --&gt; Login((Login)) </pre> </div> <p>Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p> <p>Arah pada <i>include</i> mengarah pada <i>use case</i> yang dipakai.</p>

#### 2.4.2.2 Sequence Diagram

Adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah object.

Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object.

Tabel II. 2 : Simbol *sequence diagram*




Simbol	Deskripsi
<i>Lifeline</i> 	Objek entity antar muka yang saling berinteraksi.
<i>Actor</i> 	Digunakan untuk menggambarkan <i>user</i> .
<i>Message()</i> 	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
<i>Boundary</i> 	Digunakan untuk menggambarkan sebuah <i>form</i> .
<i>Control Class</i> 	Digunakan untuk menghubungkan <i>boundary</i> dengan <i>table</i> .
<i>Entity Class</i> 	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilaksanakan.

### 2.4.2.3 Activity Diagram



Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana mereka berakhir.

*Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Tabel II. 3 : Simbol *activity diagram*

Nama simbol	Simbol	Keterangan
Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i>		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.



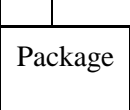

Penggabungan / <i>join</i>		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki status akhir.
Swimlane	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center;"><i>nama swimlane</i></p> </div> <p style="text-align: center;"><i>Atau</i></p> <div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; border-right: 1px solid black; border-bottom: 1px solid black;"> <div style="writing-mode: vertical-rl; transform: rotate(180deg); padding: 2px;">nama swimlane</div> <div style="flex-grow: 1; border-bottom: 1px solid black;"></div> </div> </div>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.



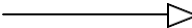
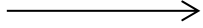
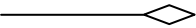
#### 2.4.2.4 Class Diagram

*Class diagram* adalah diagram yang menunjukkan *class-class* yang ada dari sebuah

sistem dan hubungannya secara logika. *Class diagram* menggambarkan struktur statis dari sebuah sistem.

Tabel II. 4 : Simbol *Class Diagram*

Nama simbol	Simbol	Deskripsi
<i>Package</i>		Package merupakan sebuah bungkusan dari satu atau lebih kelas.
Kelas	<div style="border: 1px solid black; padding: 5px;"> <p><b>nama_kelas</b></p> <hr/> <p>+attribute</p> <hr/> <p>+operation()</p> </div>	Kelas pada struktur sistem.
Antarmuka / <i>Interface</i>	 nama_interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.

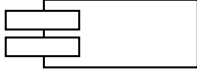



Asosiasi / <i>Association</i>		Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
Asosiasi berarah / directed association		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi juga biasanya disertai dengan multiplicity.
Generalisasi		Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
Kebergantungan / dependency		Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi / <i>Aggregation</i>		Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> ).

#### 2.4.2.5 Component Diagram

Component diagram merupakan penerapan software dari satu ataupun

lebih class, dan biasanya berupa file data atau .exe, source kode, table, dokumen dsb.

Tabel II. 5 : Simbol *component diagram*

Simbol	Deskripsi
Komponen 	Komponen sistem.
Kebergantungan 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
Antarmuka/ <i>Interface</i>  Nama_interface	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antara muka komponen agar tidak mengakses langsung komponen.
<i>Link</i> 	Relasi antar komponen





#### 2.4.2.6 Deployment Diagram

Deployment diagram adalah diagram yang digunakan memetakan software ke processing node. Menunjukkan konfigurasi elemen pemroses pada saat

run time dan software yang ada di dalamnya<sup>8</sup>.

<sup>8</sup> Deployment Diagram.  
<http://ocw.ui.ac.id/>

Tabel II. 6 : Simbol *deployment diagram*


Simbol	Dekripsi
<i>Package</i> 	Sebuah bungkus dari satu atau lebih <i>node</i> .
<i>Node</i> 	Biasanya mengacu pada perangkat keras (hardware), perangkat lunak yang tidak dibuat sendiri (software), jika di dalam <i>node</i> disertakan komponen untuk menkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.
Kebergantungan/ <i>dependency</i> 	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
<i>Link</i> 	Relasi antar <i>node</i> .

#### 2.4.2.7 Package Diagram

*Package diagram* menyediakan

cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML.

Tabel II. 7 : Simbol *package diagram*

Simbol	Deskripsi
<i>Package</i> 	Package merupakan sebuah bungkus dari satu atau lebih kelas atau elemen diagram UML.

## 2.5 Perangkat Lunak Pendukung

### 2.5.1 PHP (Hypertext Preprocessor)

#### a. Pengertian

Menurut Arief, PHP adalah Bahasa server-side-scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan server-side-scripting maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya akan dikirimkan ke browser dengan format HTML<sup>9</sup>.

Sedangkan menurut Nugroho, PHP atau singkatan dari Personal Home Page merupakan bahasa skrip yang tertanam dalam HTML untuk dieksekusi bersifat server side<sup>10</sup>. PHP termasuk dalam open source product, sehingga source code PHP dapat diubah dan didistribusikan secara bebas. Versi terbaru PHP dapat diunduh secara gratis melalui situs resmi PHP : <http://www.php.net>.

Salah satu keunggulan yang dimiliki PHP adalah kemampuannya

<sup>9</sup> Rudianto, Arif M. (2011). Pemrograman web dinamis menggunakan PHP

<sup>10</sup> Bunafit, Nugroho. (2006). Membuat Aplikasi Sistem Pakar dengan PHP

untuk melakukan koneksi ke berbagai macam software sistem manajemen basis data atau Database Management Sistem (DBMS), sehingga dapat menciptakan suatu halaman web dinamis. PHP mempunyai koneksitas yang baik dengan beberapa DBMS seperti Oracle, Sybase, mSQL, MySQL, Microsoft SQL Server, Solid, PostgreSQL, Adabas, FilePro, Velocis, dBase, Unix dbm, dan tidak terkecuali semua database ber-interface ODBC.

#### **b. Kelebihan**

- Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
- Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
- Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
- PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system.

#### **c. Tipe Data**

- Integer, adalah tipe data pada php yang berupa angka bulat seperti 1, 22, 100, 1000, tipe data ini sangat umum digunakan di bahasa pemrograman khususnya berkaitan dengan angka bulat.
- Float, adalah floating point atau real number adalah tipe data pada php

yang memiliki bagian desimal di akhir angka contohnya adalah 3,21 atau 4,5 dalam penulisan tipe data float didalam php bukan menggunakan koma (,) tetapi menggunakan titik (.).

- Boolean, adalah tipe data pada php yang paling sederhana dalam bahasa pemrograman apapun. karena tipe data ini hanya memiliki dua nilai yaitu true dan false. tipe data boolean sering kali digunakan pada operasi logika seperti kondisi if dan looping.
- String, adalah tipe data pada php yang berisi text dan karakter dimana bentuknya bisa kata atau kalimat.
- Array, berbeda dengan tipe data pada php seperti integer atau boolean, karena array adalah sebuah tipe data yang didalamnya terdiri dari kumpulan tipe data. Penulisan array terdiri dari kunci (key) dan nilai (value). key berfungsi sebagai penunjuk posisi dimana value tersimpan. tanda => berfungsi untuk memberikan nilai kepada key. untuk mengakses nilai dari array kita menggunakan kombinasi nama variable dan nilai key.
- Null, adalah sebuah tipe data spesial yang menunjukkan sebuah variabel belum memiliki nilai/data.

#### **2.5.2 Android**

Android merupakan sebuah sistem operasi untuk smartphone berbasis kernel linux yang merupakan platform terbuka sehingga pengembang dapat dengan bebas mengembangkan aplikasi. Seperti teknologi lainnya, Android muncul tidak langsung canggih seperti saat ini. Teknologinya yang bersifat open source, terus berkembang dan selalu terbuka untuk digunakan dan dikembangkan siapa saja. Mungkin inilah yang membuat Android begitu dicintai.

Versi beta muncul pertama kali pada november 2007, Android benar-benar dipasarkan dengan versi Android 1.0 pada September 2008 dengan kode nama Apple Pie. Android versi ini disematkan pada sebuah handphone HTC Dream. Pada versi ini fitur-fitur seperti Web Browser, Wifi, Bluetooth, SMS, Google Mail, Google Search, Media Player, Youtube Player, Google Maps, Google Calendar, Alarm, Calculator, Phone, Android Market, dan fitur standar smart-phone lainnya. Adapun versi-versi yang pernah dirilis adalah sebagai berikut:

1. Android versi 1.0, merupakan versi komersial pertama android, dirilis 23 September 2008 dengan kode nama Apple pie serta ukuran layar 320 x 480 HVGA. Perangkat Android pertama yang tersedia se-cara komersial adalah HTC Dream.
2. Android versi 1.1 (9 Maret 2009), Android ini telah didukung oleh Google Mail Service dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail dan pemberitahuan *email*.
3. Android versi 1.5 (*Cupcake*), diluncurkan google pada pertengahan Mei 2009. Masih dengan *Google Inc.* Android ini telah dilengkapi dengan *software development kit* dengan berbagai pembaruan termasuk penambahan beberapa fitur lain. Yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke youtube, *upload* gambar ke picasa langsung dari telepon, serta mendapat dukungan Bluetooth A2DP.
4. Android versi 1.6 (*Donut*), dirilis pada September 2009. Android versi ini menampilkan proses pencarian yang lebih baik dari versi-versi sebelumnya. Android Donut juga memiliki fitur-fitur tambahan seperti galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, *camcorder* dan game yang diintegrasikan, *text-to-speech engine*, *dial* kontak, teknologi *text to change speech*, *battery indicator*, dan kontrol *applet* VPN.
5. Android versi 2.0/2.1 (*Eclair*), diluncurkan pada 3 Desember 2009. Perubahan pada versi ini diantaranya adalah: pengoptimalan *hardware*, peningkatan *google maps* 3.1.2, perubahan pada *user interface* dengan browser baru dengan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3.2 MP, digital zoom, dan bluetooth 2.1. Android ini adalah android pertama yang mulai dipakai oleh banyak *smartphone*. Fitur utamanya tentu adalah perubahan total struktur dan tampilan *user interface*.
6. Android versi 2.2 (*Froyo: Frozen Yoghurt*), dirilis pada Mei 2010. Android ini memiliki kecepatan kinerja aplikasi 2 sampai 4 kali versi-versi sebelumnya. Selain itu ada perubahan fitur-fitur baru seperti dukungan *adobe flash* 10.1, integrasi v8 JavaScript engine yang dipakai *Google Chrome* yang mempercepat kemampuan rendering pada *browser*, pemasangan aplikasi pada SD Card, kemampuan *WiFi Hotspot portable* dan kemampuan *auto update* dalam aplikasi android market.
7. Android versi 2.3 (*Gingerbread*), diluncurkan pada Desember 2010. Perubahan umum yang terdapat pada android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antar muka (*Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek *audio* baru (*reverb*, dukungan kemampuan *Near Field Communication* (NFC))

- dan dukungan jumlah kamera yang lebih dari satu.
8. Android versi 3.0 (*Honeycomb*), rilis pada 10 Mei 2011, versi ini khusus dirancang untuk tablet. Versi Android yang dirancang khusus untuk device dengan layar besar seperti Tablet PC. Fitur baru pada Android Honeycomb antara lain dukungan terhadap *Processor multi-core* dan grafis dengan hardware *acceleration user interface* pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Tablet pertama yang memakai Honeycomb adalah tablet Motorola Xoom yang dirilis bulan Februari 2011.
  9. Android versi 4.0 (*Ice Cream Sandwich*), dirilis pada 16 Desember 2011, Pihak mengklaim Android Ice Cream Sandwich akan digunakan baik di smartphone ataupun tablet. Android Ice Cream Sandwich membawa fitur baru seperti membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.
  10. Android versi 4.1-4.3 (*Jelly Bean*), diluncurkan pada 9 Juli 2012. Keunggulan dan fitur baru versi 4.1 ini, diantaranya peningkatan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui Voice Search yang lebih cepat. Dilengkapi juga Google Now yang dapat memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi Android Jelly Bean 4.1 pertama kali digunakan dalam produk tablet asus, yakni Google Nexus 7. Jelly Bean 4.2 memiliki Fitur photo sphere untuk panorama, daydream sebagai screensaver, power control, lock screen widget, menjalankan banyak user (dalam tablet saja), widget terbaru. Android 4.2 pertama kali dikenalkan melalui LG Google Nexus 4. Google merilis Jelly Bean 4.3 pada 24 Juli 2013 di San Francisco. Nexus 7 generasi kedua adalah perangkat pertama yang menggunakan sistem operasi ini. Sebuah pembaruan minor dirilis pada tanggal 22 Agustus 2013.
  11. Android versi 4.4 (*KitKat*). Google mengumumkan Android 4.4 Kitkat (dengan izin dari Nestle dan Hershey) pada 3 September 2013, dirilis pada tanggal 31 Oktober 2013. Keunggulan diantaranya pembaruan antarmuka dengan berstatus navigasi transparan pada layar depan, optimasi Kitkat dengan spesifikasi perangkat yang lebih rendah, NFC Host Card Emulation Sebagai emulator kartu pintar, WebViews Counter API, Peningkatan tampilan mode layar penuh, penyeimbangan audio, dan peningkatan suara audio, dukungan Bluetooth *Message Access Profile* (MAP).
  12. Android versi 5.0-5.1 (*Lollipop*), dirilis tanggal 15 Oktober 2014. Beberapa perubahan signifikan terlihat di beberapa sektor, seperti pada panel notifikasi terdapat menu Quick Settings yang berguna untuk melakukan pengaturan cepat seperti WiFi dan Bluetooth, adanya fitur penghematan data, smooth dan lancar dalam menjalankan berbagai macam aplikasi berat maupun *multitasking*.
  13. Android versi 6.0 (*Marshmallow*), dirilis pada bulan September 2015. Beberapa fitur yang dapat terlihat

baru adalah Google Now dengan tampilan terbaru, App Permissions, *flex storage* untuk memindahkan aplikasi dari memori internal ke microSD, tampilan chrome custom tab, *Doze and Charging* untuk membantu stamina batre dan percepatan pengisian daya, pemindai sidik jari, penyempurnaan seleksi kata, pada versi ini juga pengguna bisa *uninstall*

beberapa aplikasi bawaan sehingga dapat mengoptimalkan kebutuhan kecuali aplikasi *setting*, Google play store, Gmail dan clock adalah aplikasi yang tidak bisa di *uninstall* serta beberapa fitur lainnya.

14. Android versi 7.0 (*Nougat*), resmi diperkenalkan pada bulan Juni 2016. Beberapa fitur android terbaru dapat dilihat pada table dibawah ini.

Tabel II. 8 : Versi Android 7.0 (Nougat)<sup>11</sup>

No.	Fitur	Keterangan
1.	Project Svelte	Memungkinkan penggunaan kinerja RAM yang lebih minim sehingga lebih hemat baterai.
2.	Virtual Reality	Virtual Reality alias VR merupakan salah satu teknologi yang tengah naik daun saat ini. Google ternyata juga tak mau kalah dengan meluncurkan platform VR bernama Google Daydream. Google ingin membangun sebuah ekosistem teknologi VR yang keren dan menyenangkan.
3.	Multitasking	pengguna Android bisa menjalankan dua aplikasi secara bersamaan.
4.	Night Mode	memungkinkan layar meredup secara otomatis saat digunakan sehingga mengurangi efek yang dapat merusak mata.
5.	Notifications	Pengguna bisa langsung membuka dan membalas pesan masuk langsung melalui notifikasi tanpa perlu membuka aplikasi. Selain itu juga terdapat fitur bundle notifikasi yang dapat mengelompokkan notifikasi-notifikasi tertentu.

<sup>11</sup> Android Nougat. [https://www.android.com/intl/id\\_id/versions/nougat-7-0/](https://www.android.com/intl/id_id/versions/nougat-7-0/)

### a. Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu-Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya:

- Sistem versi berbasis Gradle yang fleksibel
- Emulator yang cepat dan kaya fitur
- Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
- Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
- Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
- Alat pengujian dan kerangka kerja yang ekstensif
- Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
- Dukungan C++ dan NDK
- Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine

### b. Android SDK

Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk memulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Beberapa pendapat tentang android SDK diantaranya:

1. Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk memulai mengembangkan aplikasi pada platform Android menggunakan bahasa

pemrograman java<sup>12</sup>.

2. SDK Android merupakan sebuah tool dan alat bantu API (Application Programming Interface) yang diperlukan untuk mengembangkan aplikasi berbasis android yang menggunakan bahasa java<sup>13</sup>.

Alfa Satyaputra & Eva Maulina Aritonang (2014:41), Mengatakan bahwa: “SDK Android merupakan sebuah tool dan alat bantu API (Application Programming Interface) yang diperlukan untuk mengembangkan aplikasi berbasis android yang menggunakan bahasa java”. Dari penulisan diatas dapat di ambil kesimpulan bahwa Android SDK adalah alat bantu tool API (Application Programming Interface) yang digunakan untuk membuat aplikasi pada platform Android menggunakan bahasa pemrograman java.

### 2.5.3 Notepad++

Notepad++ adalah sebuah penyunting teks dan penyunting kode sumber yang berjalan di sistem operasi Windows. Notepad++ menggunakan komponen Scintilla untuk dapat menampilkan dan menyuntingan teks dan berkas kode sumber berbagai bahasa pemrograman. Notepad++ didistribusikan sebagai perangkat lunak bebas. Proyek ini dilayani oleh Sourceforge.net.

Beberapa bahasa pemrograman yang didukung oleh Notepad++ diantaranya adalah:

- Assembler
- C++
- HTML

---

<sup>12</sup> Nazarudin Safaat H (2012).

Pemrograman aplikasi mobile smartphone dan tablet PC berbasis android

<sup>13</sup> Alfa Satyaputra & Eva Maulina Aritonang (2014). Beginning Android Programming with ADT Budle



- Java
- PHP
- XML

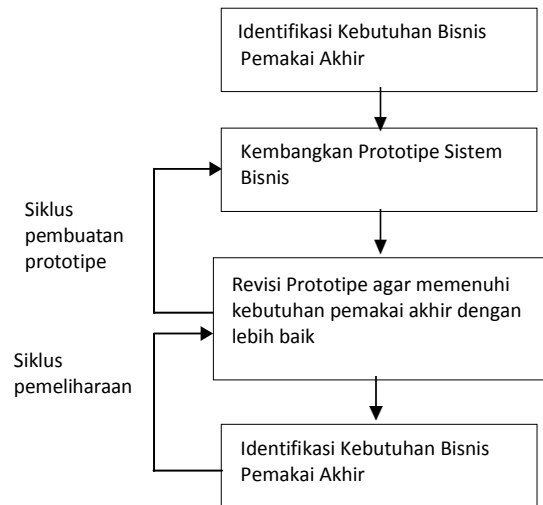
### 2.5.4 Prototype

Proses pengembangan sistem seringkali menggunakan pendekatan prototipe (*prototyping*). Metode ini sangat baik dalam menyelesaikan masalah kesalahpahaman antara *user* dan analis yang timbul akibat *user* tidak mampu mendefinisikan secara jelas kebutuhannya.

Sebagian *user* kesulitan mengungkapkan keinginannya untuk mendapatkan aplikasi yang sesuai kebutuhannya. Kesulitan ini yang perlu diselesaikan oleh analis dengan memahami kebutuhan *user* dan menterjemahkannya kedalam bentuk model (prototipe). Model ini selanjutnya diperbaiki secara terus menerus sampai sesuai dengan kebutuhan *user*<sup>14</sup>.

Fitur yang akan diimplementasikan pada prototipe sistem dapat dibatasi dengan Teknik pertikal atau horizontal. *Vertical Prototype* mengandung fungsi yang detail tetapi hanya untuk beberapa fitur yang terpilih, tidak pada keseluruhan fitur sistem. *Horizontal Prototype* mencakup seluruh fitur antarmuka pengguna namun tanpa fungsi pokok hanya berupa simulasi dan belum dapat digunakan untuk melakukan pekerjaan yang sebenarnya.

### Proses Pembuatan Prototype



Gambar II. 2 : *Prototype Metode*

Tahap-tahap sesuai metode *prototype* yang penulis lakukan dalam penelitian adalah sebagai berikut:

1. Pengumpulan Kebutuhan  
 Dalam tahap ini, penulis bertemu dengan calon pengguna aplikasi baik itu admin atau kurir untuk mendefinisikan kebutuhan seluruh perangkat lunak, mengidentifikasi semua kebutuhan dan garis besar sistem yang akan dibuat.
2. Membangun *Prototyping*  
 Tahap ini dilakukan dengan membuat perancangan sementara yang berpusat pada penyajian kepada pelanggan (membuat input dan contoh output).
3. Evaluasi *Prototyping*  
 Tahap ini dilakukan untuk mengetahui apakah prototipe itu sudah sesuai dengan keinginan atau belum, jika belum maka prototipe diperbaiki dengan mengulang langkah 1, 2, dan 3.
4. Mengkodekan Sistem  
 Tahap selanjutnya prototipe yang sudah disepakati diterjemahkan kedalam bahasa pemrograman.
5. Menguji Sistem

<sup>14</sup> Agus Mulyanto (2009). Sistem informasi konsep dan Aplikasi

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dulu sebelum digunakan. Pengujian ini dilakukan dengan *White Box* dan *Black Box*.

6. **Evaluasi Sistem**  
Pada tahap ini pelanggan atau *user* mengevaluasi apakah sistem sudah sesuai dengan yang diharapkan. Jika sudah maka langkah ketujuh dilakukan, jika belum maka mengulangi langkah 4 dan 5.
7. **Menggunakan Sistem**  
Sistem Aplikasi yang telah diuji dan diterima oleh pelanggan atau *user* siap untuk digunakan.

### **Kelebihan dan Kekurangan Prototype**

#### **a. Kelebihan Metode Prototype**

1. Komunikasi akan terjalin baik antara pengembang dan pelanggan
2. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan setiap pelanggannya
3. Pelanggan berperan aktif dalam proses pengembangan sistem
4. Lebih menghemat waktu dalam pengembangan sistem
5. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya

#### **b. Kekurangan Metode Prototype**

1. Pelanggan kadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jangka waktu lama.
2. Pengembang biasanya ingin cepat menyelesaikan proyek sehingga menggunakan algoritma dan bahasa pemrograman yang sederhana.
3. Hubungan pelanggan dengan komputer mungkin tidak meng-

gambarkan teknik perancangan yang baik.

### **2.5.5 Basis Data**

Basis data diartikan sebagai kumpulan data yang saling berhubungan satu sama lain, tersimpan di simpanan luar komputer dan menggunakan perangkat lunak tertentu untuk memanipulasinya.

#### **Pemanfaatan Basis Data**

Pemanfaatan basis<sup>15</sup> data dilakukan dengan tujuan yaitu:

1. *Speed* (Kecepatan dan Kemudahan)  
Pemanfaatan Database memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan (manipulasi) dan menampilkan kembali data tersebut dengan cepat dan mudah, dari pada kita menyimpan data secara manual.
2. *Space* (Efisien Ruang penyimpanan)  
Dengan Database penggunaan ruang penyimpanan data dapat dilakukan karena kita dapat melakukan penekanan jumlah pengulangan data dengan menerapkan sejumlah pengkodean.
3. *Accuracy* (Keakuratan)  
Pemanfaatan pengkodean atau pembentukan relasi antar data dengan penerapan aturan atau batasan tipe data dapat diterapkan dalam Database yang berguna untuk menentukan ketidakakuratan pemasukan atau penyimpanan.
4. *Security* (Keamanan)  
Dalam sejumlah sistem (aplikasi) pengelola database tidak menerapkan aspek keamanan dalam penggunaan database. Tetapi untuk sistem yang besar dan serius, aspek keamanan juga dapat diterapkan.

---

<sup>15</sup> Pengertian basis data.

<http://blog.unnes.ac.id/ayukwitantri/2016/02/25/pengertian-basis-data-tujuan-manfaat-dan-contoh-kasus-basis-data/>

Dengan begitu kita dapat menentukan siapa yang boleh menggunakan database dan menentukan jenis operasi-operasi apa saja yang boleh dilakukan.

5. *Consistant* (Terpeliharanya keselarasan data)

Apabila ada perubahan data pada aplikasi yang berbeda maka secara otomatis perubahan itu berlaku untuk keseluruhan.

6. *Shared* (Data dapat dipakai secara bersama)

Data dapat dipakai secara bersama-sama oleh beberapa program aplikasi (secara batch maupun on-line) pada saat bersamaan.

7. *Standardization* (Dapat diterapkan standarisasi)

Dengan adanya pengontrolan yang terpusat maka database dapat menerapkan standarisasi data yang disimpan sehingga memudahkan pemakaian, pengiriman maupun pertukaran data.

### 2.5.6 XAMPP

XAMPP ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai server yang berdiri sendiri (localhost), yang terdiri dari program MySQL database, Apache HTTP Server, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl.

Nama XAMPP merupakan singkatan dari X (empat sistem operasi), Apache, MySQL, PHP dan Perl. Program ini tersedia di bawah GNU General Public License dan bebas, adalah mudah untuk menggunakan web server yang dapat melayani tampilan halaman web yang dinamis. Jika ingin mendapatkan xampp dapat mendownload langsung dari situs resminya.

XAMPP merupakan singkatan dari:

- **X:** Program ini dapat dijalankan di banyak sistem operasi, seperti Windows, Linux, Mac OS, dan Solaris.
- **A: Apache**, server aplikasi Web. Apache tugas utama adalah untuk menghasilkan halaman web yang benar kepada pengguna terhadap kode PHP yang sudah dituliskan oleh pembuat halaman web. jika perlu kode PHP juga berdasarkan yang tertulis, database dapat diakses dulu (misalnya MySQL) untuk mendukung halaman web yang dihasilkan.
- **M: MySQL**, server aplikasi database. Pertumbuhannya disebut SQL singkatan dari Structured Query Language. SQL merupakan bahasa terstruktur yang difungsikan untuk mengolah database. MySQL dapat digunakan untuk membuat dan mengelola database dan isinya. Bisa juga memanfaatkan MySQL guna untuk menambahkan, mengubah, dan menghapus data dalam database.
- **P: PHP**, bahasa pemrograman web. Bahasa pemrograman PHP adalah bahasa pemrograman untuk membuat web yang server-side scripting. PHP digunakan untuk membuat halaman web dinamis. Sistem manajemen database yang sering digunakan dengan PHP adalah MySQL. PHP juga mendukung Pengelolaan sistem database Oracle, Microsoft Access, Interbase, d-base, PostgreSQL, dan sebagainya.
- **P: Perl**, bahasa pemrograman untuk semua tujuan, pertama kali dikembangkan oleh Larry Wall, mesin Unix. Perl dirilis pertama kali tanggal 18 Desember 1987 yang ditandai dengan keluarnya Perl 1. Pada versi-versi selanjutnya, Perl juga tersedia untuk berbagai sistem operasi Unix (SunOS, Linux, BSD, HP-UX), juga

tersedia untuk sistem operasi seperti DOS, Windows, PowerPC, BeOS, VMS, EBCDIC, dan PocketPC.

### 2.5.7 Internet

Secara sederhana, Internet adalah kumpulan dari jutaan komputer di seluruh dunia yang terkoneksi antara yang satu dengan yang lain. Media koneksi yang digunakan bisa melalui sambungan telpon, serat optik (fiber optic), kabel koaksial (coaxial cable), satelit atau dengan koneksi wireless.

### WWW (Word Wide Web)

Adalah nama yang diberikan untuk semua bagian Internet yang dapat diakses dengan software web browser. Sampai saat ini ada beberapa software web browser yang sering digunakan antara lain Microsoft Internet Explorer, Netscape Navigator, Mozilla dan Opera. World Wide Web atau WWW atau singkatnya web, terdiri dari jutaan situs web (web site) dan setiap web site terdiri banyak halaman web (web page). Halaman-halaman web ini tersebar di seluruh dunia di komputer-komputer server yang terhubung dengan Internet. Halaman Web adalah halaman yang tampak dari sebuah software web browser. Halaman ini seperti file dokumen word-processing, kecuali didalamnya dapat terkandung text, gambar bahkan animasi. Untuk dapat

membuat halaman web ini anda harus memiliki pengetahuan tentang skrip (script) HTML (hyper-text markup language).

### HTTP (Hyper Text Transfer Protocol)

Adalah suatu protokol yang digunakan untuk mengirim dokumen atau halaman dalam WWW atau World Wide Web. Sedangkan pengertian HTTP menurut kamus besar adalah protokol jaringan untuk didistribusikan, kolaboratif, sistem informasi hypermedia. HTTP adalah dasar dari komunikasi data untuk WWW.

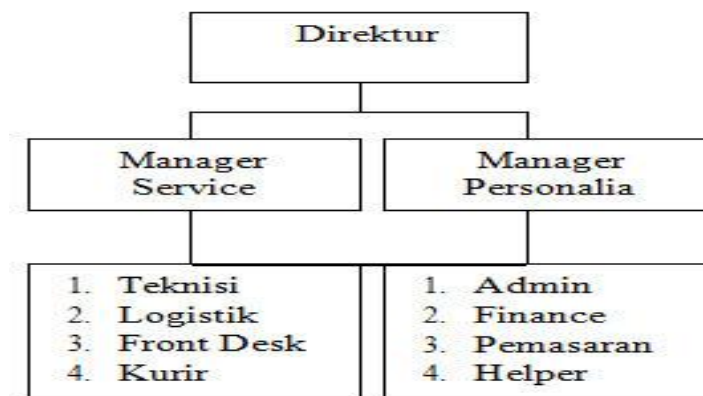
## 3. ANALISIS SISTEM BERJALAN

### 3.1 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem ini, akan memberikan gambaran tentang struktur organisasi perusahaan dengan aplikasi yang dirancang, permasalahan yang ada, serta apa yang dibutuhkan dari sistem tersebut yang belum ada.

#### 3.1.1 Struktur Organisasi Perusahaan

Struktur organisasi merupakan penentuan bagaimana suatu pekerjaan dikelompokkan atau dibagi-bagi secara formal. Struktur organisasi ini berfungsi sebagai suatu alur yang mengatur tugas dan wewenang agar pekerjaan dapat dilaksanakan secara terarah dan terkendali.



Gambar III. 1 : Struktur Organisasi Perusahaan

Keterangan fungsi dari tiap-tiap bagian adalah sebagai berikut:

1. Direktur
  - Mengawasi dan mengevaluasi perusahaan agar berjalan dengan baik.
2. Manager Service
  - Membantu proses service.
  - Mengoptimalkan layanan kepada *customer*.
  - Mengawasi ketersediaan *sparepart*.
  - Memastikan semua barang terkirim dan diterima dengan baik.
3. Manager Personalia
  - Membuat dan mengatur organisasi dalam perusahaan.
  - Mengendalikan pendapatan dan pengeluaran keuangan perusahaan.
  - Menyediakan dan mengurus sistem administrasi bagi karyawan.

### 3.1.2 Analisis Sistem Berjalan

Analisis ini difokuskan dalam laya-

nan jemput dan antar (*Pick Up and Return*) di PT. Sinar Elok Abadi. Dari analisis yang dilakukan, berikut adalah gambaran sistem yang berjalan pada layanan *Pick Up and Return*.

1. Proses pengajuan *service*.
  - a. Customer menghubungi PT. Sinar Elok Abadi
  - b. Menyampaikan kerusakan.
  - c. Memberitahu alamat penjemputan.
2. Proses penerimaan permintaan *service*.
  - a. Menerima telepon dari customer.
  - b. Mencatat kerusakan dan alamat kedalam sistem.
  - c. Menentukan kurir yang akan mengambil barang.
3. Proses penjemputan dan pengiriman barang yang akan dan telah diperbaiki.
  - a. Kurir menerima data dari admin alamat penjemputan.
  - b. Mengambil/mengantarkan barang dari customer.

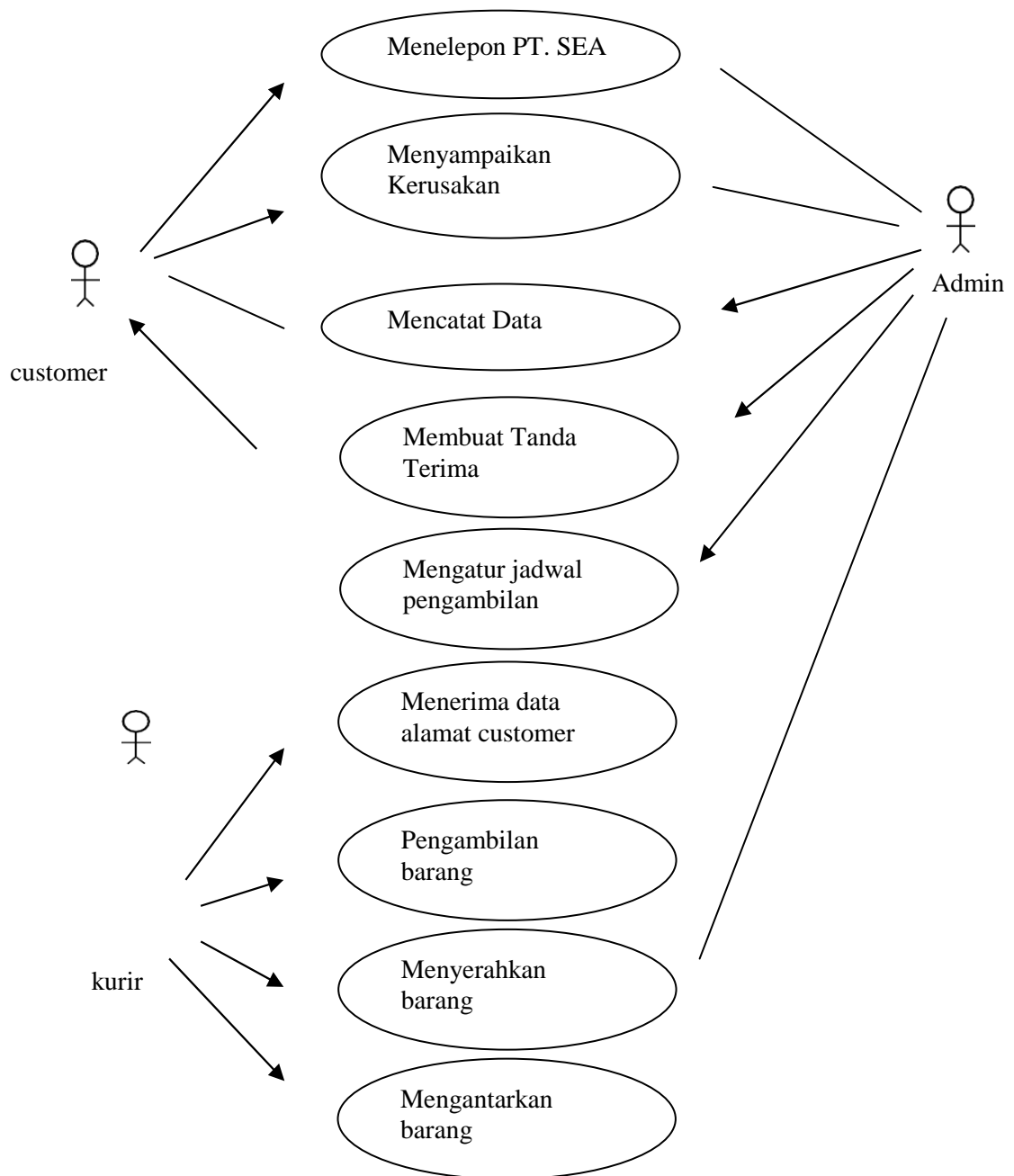
Tabel III. 1 : Aktor dalam sistem berjalan

No.	Aktor	Keterangan
1.	<i>Customer</i>	Orang yang akan mengajukan layanan <i>Pick Up and Return</i>
2.	Admin	Orang yang bekerja di PT. Sinar Elok Abadi dan mengurus bagian layanan <i>Pick Up and Return</i> (Biasa disebut admin <i>pick up and return</i> )
3.	Kurir	Orang yang bekerja di PT. Sinar Elok Abadi dan bertugas mengambil dan atau mengantar unit yang akan dan atau telah diperbaiki.

#### 3.1.2.1 Use Case Diagram Sistem Berjalan

Gambaran *use case diagram*

digunakan PT. SEA dalam layanan *pick up and return* yang sedang berjalan adalah sebagai berikut:



Gambar III. 2 : Use case diagram layanan *pick up and return*

### Skenario Use Case Request Pick Up and Return Sistem Berjalan

Skenario *use case* adalah narasi tentang aktivitas dalam suatu *use case* diagram. Adapun skenario *use case* untuk membuat aplikasi *pick up and return* adalah sebagai berikut:

Nama use case : Pengajuan layanan *pick up and return*.

Aktor : Customer.

Deskripsi : Meliputi proses pengajuan layanan, pemberian info kerusakan dan alamat penjemputan/pengantaran.

Tabel III. 2 : Skenario *use case* pengajuan layanan pick up and return

Aktor	Sistem
1. <i>Customer</i> melakukan pengajuan layanan dengan memberikan info-info yang dibutuhkan oleh pihak PT. Sinar Elok Abadi	2. Admin akan mencatat kerusakan dan alamat <i>customer</i> serta data lain yang dibutuhkan.
	3. Admin melakukan penjadwalan penjemputan barang.
	4. Admin menentukan kurir yang akan mengambil barang dari <i>customer</i> .

Nama use case : Pencatatan layanan *pick up and return*.

Aktor : Admin.

Deskripsi : Meliputi proses pencatatan data kerusakan dan data customer.

Tabel III. 3 : Skenario pencatatan layanan *pick up and return*

Aktor	Sistem
	1. Admin menerima laporan kerusakan dan data customer.
	2. Data tercatat dalam sistem untuk selanjutnya dibuatkan jadwal.
3. Admin membuat jadwal penjemputan dan penentuan kurir mana yang akan mengambil barang.	

Nama use case : Penjemputan/pengantaran unit customer.

Aktor : Kurir.

Deskripsi : Meliputi proses penjemputan/pengantaran.

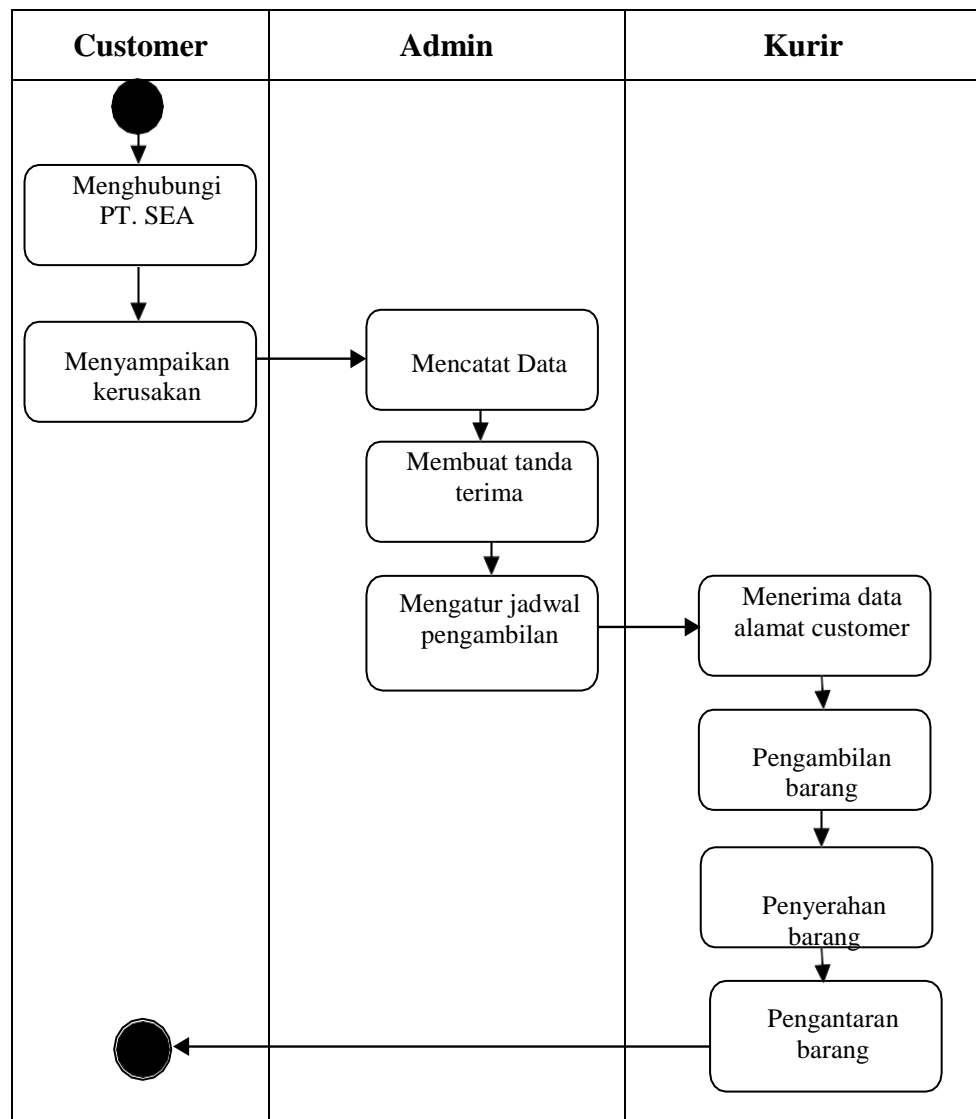
Tabel III. 4 : Skenario penjemputan/pengantaran barang

Aktor	Sistem
	1. Data dan jadwal dicatat dan ditentukan oleh admin.
2. Kurir menerima data alamat customer dari admin dan melakukan penjemputan / pengantaran barang.	

### 3.1.2.2 Activity Diagram Sistem Pick Up and Return yang sedang Berjalan

Hasil analisis penulis dari penelitian di PT. Sinar Elok Abadi, diketahui

bentuk activity diagram sistem yang berjalan secara garis besar adalah sebagai berikut:



Gambar III. 3 : Activity diagram sistem berjalan

### Kelebihan dan Kekurangan Sistem Berjalan

#### Kelebihan Sistem Berjalan

1. Penjemputan dan pengantaran barang sesuai dengan data dari customer dan sesuai jadwal yang telah disepakati (oleh *customer* dan PT. Sinar Elok Abadi)
2. Admin mencatat semua data yang diperlukan dalam proses layanan *pick up and return*.

#### Kekurangan Sistem Berjalan

1. Lambatnya proses pengajuan karena *customer* harus menelpon sedangkan admin hanya ada satu orang,

menghambat saat ada beberapa *customer* melakukan pengajuan di waktu yang sama.

2. Kurir masih harus menunggu data dari admin dalam proses penjemputan, saat dalam perjalanan kurir tidak bisa langsung ke tempat *customer* kedua dan seterusnya karena data ada di admin sehingga membuat waktu kurir banyak terbuang untuk pulang dan pergi mengambil data serta barang.

### 3.1.3 Analisis Kebutuhan Sistem Perusahaan

Dari hasil penelitian dan ditemukan



berbagai masalah yang ada, penulis menyimpulkan bahwa perusahaan membutuhkan suatu aplikasi yang bisa membantu layanan *pick up and return* lebih efektif, seperti:

1. Aplikasi untuk pengajuan layanan *pick up and return* yang bisa diakses melalui aplikasi *smartphone* sehingga tidak perlu mengantri saat melakukan panggilan saat admin sedang melayani *customer* di panggilan yang lain.
2. Aplikasi bagi kurir dalam menerima data dan jadwal pengambilan/pengantaran barang.

#### 4. PERANCANGAN DAN IMPLEMENTASI SISTEM

##### 4.1 Perancangan Sistem

###### 4.1.1 Perancangan Proses Bisnis

Berikut adalah perancangan proses-proses yang penulis usulkan:

1. Perancangan proses pengajuan perbaikan
  - a. Customer melakukan pengunduhan aplikasi melalui google play store
  - b. Customer melakukan registasi pada aplikasi
  - c. Customer melakukan pengejuan

- perbaikan dengan mengisi form yang tersedia pada aplikasi
2. Perancangan proses penerimaan permintaan perbaikan
  - a. Kurir menerima pemberitahuan permintaan perbaikan
  - b. Kurir mengambil barang sesuai info dari aplikasi
3. Perancangan proses penjemputan dan pengiriman barang yang akan dan telah diperbaiki
  - a. Admin menerima dan mengganti status proses perbaikan
  - b. Setelah selesai, Admin mengganti proses perbaikan menjadi selesai
  - c. Kurir menerima pemberitahuan proses perbaikan selesai dan selanjutnya melakukan pengantaran unit yang sudah selesai sesuai informasi dari aplikasi

###### 4.1.2 Use Case Diagram Sistem Usulan

Pembuatan *use case diagram* sistem usulan mempunyai beberapa langkah yang penulis lakukan dalam proses penelitian, diantaranya:

###### 1. Identifikasi Actor

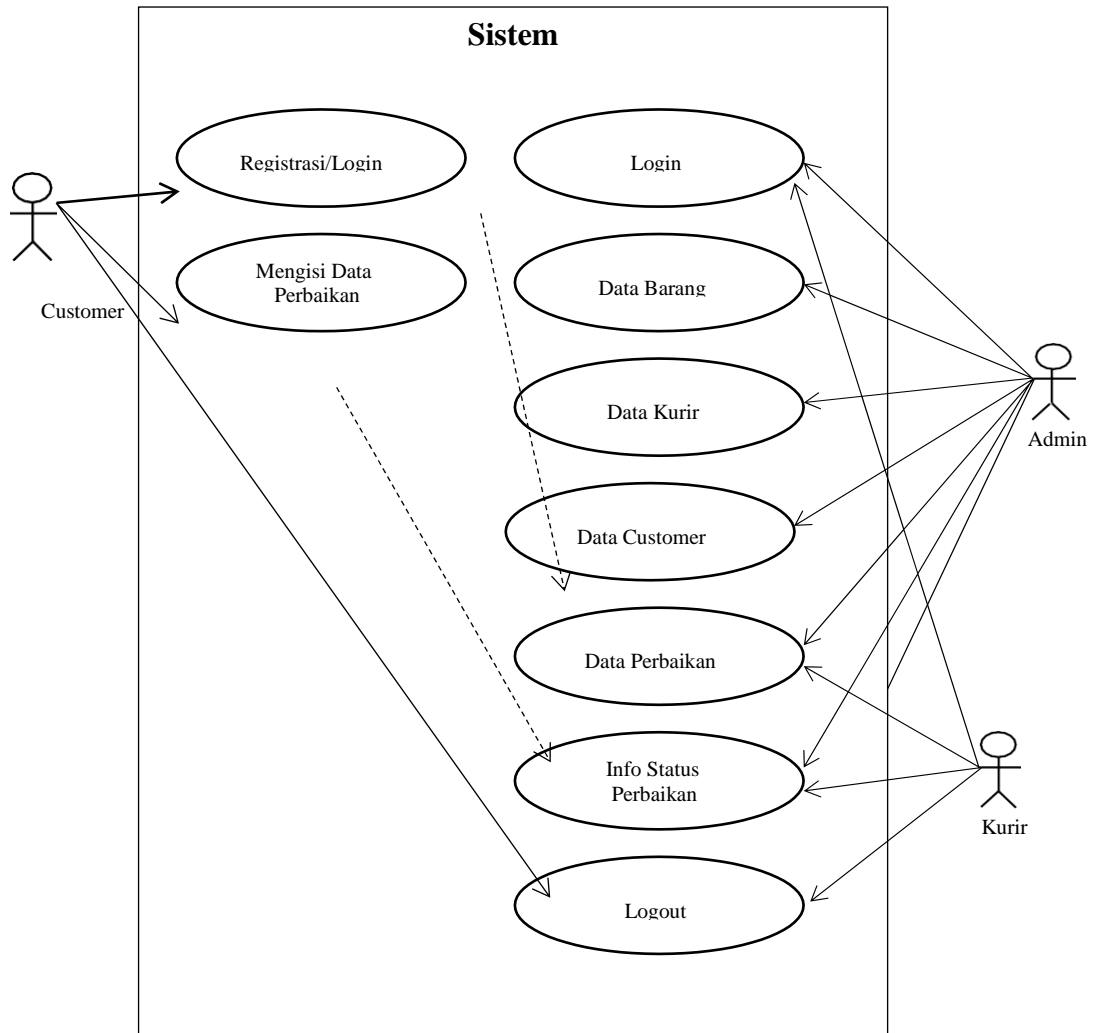
*Actor* yang ada dalam sistem usulan adalah sebagai berikut:

Tabel IV.1 : Aktor dalam sistem usulan

No	Actor	Keterangan
1.	Customers	Orang yang akan melakukan perbaikan unit
2.	Admin	Orang yang melakukan proses pengolahan penerimaan dan atau pengiriman dalam proses jemput atau antar
3.	Kurir	Orang yang menerima tugas proses penjemputan unit yang akan diperbaiki dan pengantaran unit yang sudah selesai.

2. Identifikasi *use case*

Untuk *use case diagram* yang diusulkan dapat dilihat pada gambar berikut:



Gambar IV.1 : *Use case* usulan

Tabel IV.2 : Skenario Use Case Sistem Usulan

No	Nama Use Case	Keterangan	Aktor
1	Login	Menggambarkan kegiatan login dari semua pengguna	Admin, Kurir, Customer
2	Data Barang	Menggambarkan kegiatan yang berisi data barang	Admin
3	Data Kurir	Menggambarkan kegiatan yang berisi data Kurir	Admin
4	Data Customer	Menggambarkan kegiatan yang berisi data Customer	Admin
5	Data Perbaikan	Menggambarkan kegiatan yang berisi data Perbaikan	Admin, Kurir, Customer
6	Info Status Perbaikan	Menggambarkan kegiatan yang berisi info status perbaikan	Admin, Kurir, Customer

1. Skenario Use Case Login yang diusulkan

Skenario use case merupakan narasi tentang aktivitas dalam suatu use case diagram. Adapun skenario use case login dalam sistem yang diusulkan adalah sebagai berikut:

Nama Use Case : Login

Aktor : Admin, Kurir, Customer

Deskripsi : Meliputi Pengisian *username* dan *password*

Tabel IV.3 : Skenario use case login yang diusulkan

Aktor	Sistem
1. Admin/Kurir mengisi username dan password (Untuk customer perlu registrasi untuk mendapatkan username dan login)	2. Sistem memproses username dan password di database aplikasi pick up and return
	3. Jika username dan password dimasukan dengan benar sesuai database maka sistem akan menampilkan menu utama, jika belum maka akan muncul peringatan kesalahan pada password atau email.
4. Admin mengelola menu barang, kurir, customer, perbaikan, status perbaikan	

2. Skenario Use Case Data Barang diusulkan

Skenario use case merupakan narasi tentang aktivitas dalam suatu use case diagram. Adapun skenario use case data barang dalam sistem yang diusulkan adalah sebagai berikut:

Nama Use Case : Data Barang

Aktor : Admin

Deskripsi : Meliputi Pengolahan data barang beserta jenisnya

Tabel IV.4 : Skenario use case Data Barang yang diusulkan

Aktor	Sistem
1. Admin mengisi data barang dan atau jenisnya	2. Sistem memproses data kedalam database barang yang ada
	3. Sistem akan menampilkan hasil pengolahan data yang dilakukan oleh aktor
4. Admin melihat data yang telah Diolah	

3. Skenario Use Case Data Kurir diusulkan

Skenario use case merupakan narasi tentang aktivitas dalam suatu use case diagram. Adapun skenario use case data kurir dalam sistem yang diusulkan adalah sebagai berikut:

Nama Use Case : Data Kurir

Aktor : Admin

Deskripsi : Meliputi pengolahan data kurir

Tabel IV.5 : Skenario use case Data Kurir yang diusulkan

Aktor	Sistem
1. Admin mengisi data kurir	2. Sistem memproses data kedalam database kurir yang ada
	3. Sistem akan menampilkan hasil pengolahan data yang dilakukan oleh aktor
4. Admin melihat data yang telah diolah	

4. Skenario Use Case Data Customer diusulkan

Skenario use case merupakan narasi tentang aktivitas dalam suatu use case diagram. Adapun skenario use case data customer dalam sistem yang diusulkan adalah sebagai berikut:

Nama Use Case : Data Customer

Aktor : Admin

Deskripsi : Meliputi pengolahan data customer

Tabel IV.6 : Skenario use case Data Customer yang diusulkan

Aktor	Sistem
1. Admin melihat dan mengubah bila ada kesalahan pada data customer	2. Sistem memproses data kedalam database customer yang ada
	3. Sistem akan menampilkan hasil pengolahan data yang dilakukan oleh aktor
4. Admin melihat data yang diolah	

5. Skenario Use Case Data Perbaikan diusulkan

Skenario use case merupakan narasi tentang aktivitas dalam suatu use case diagram. Adapun skenario use case data perbaikan dalam sistem yang diusulkan adalah sebagai berikut:

Nama Use Case : Data Perbaikan

Aktor : Admin, Customer, Kurir

Deskripsi : Meliputi pengisian data perbaikan dan proses penjemputan dan atau pengantaran hasil perbaikan

Tabel IV.7 : Skenario use case Data Perbaikan yang diusulkan

Aktor	Sistem
1. Customer memberikan data perbaikan	2. Sistem memproses data perbaikan yang selanjutnya muncul pada sistem tampilan kurir
3. Kurir menerima data perbaikan dalam sistem, kemudian melakukan penjemputan barang untuk Diperbaiki	4. Sistem mengirim data perbaikan kepada Admin
5. Admin menerima data perbaikan dan mengubah status perbaikan	

6. Skenario Use Case Info Status Perbaikan diusulkan

Skenario use case merupakan narasi tentang aktivitas dalam suatu use case diagram. Adapun skenario use case info status perbaikan dalam sistem yang diusulkan adalah sebagai berikut:

Nama Use Case : Info Status Perbaikan  
 Aktor : Admin, Customer, Kurir  
 Deskripsi : Meliputi pesan aplikasi untuk status perbaikan

Tabel IV.8 : Skenario use case Info Status Perbaikan yang diusulkan

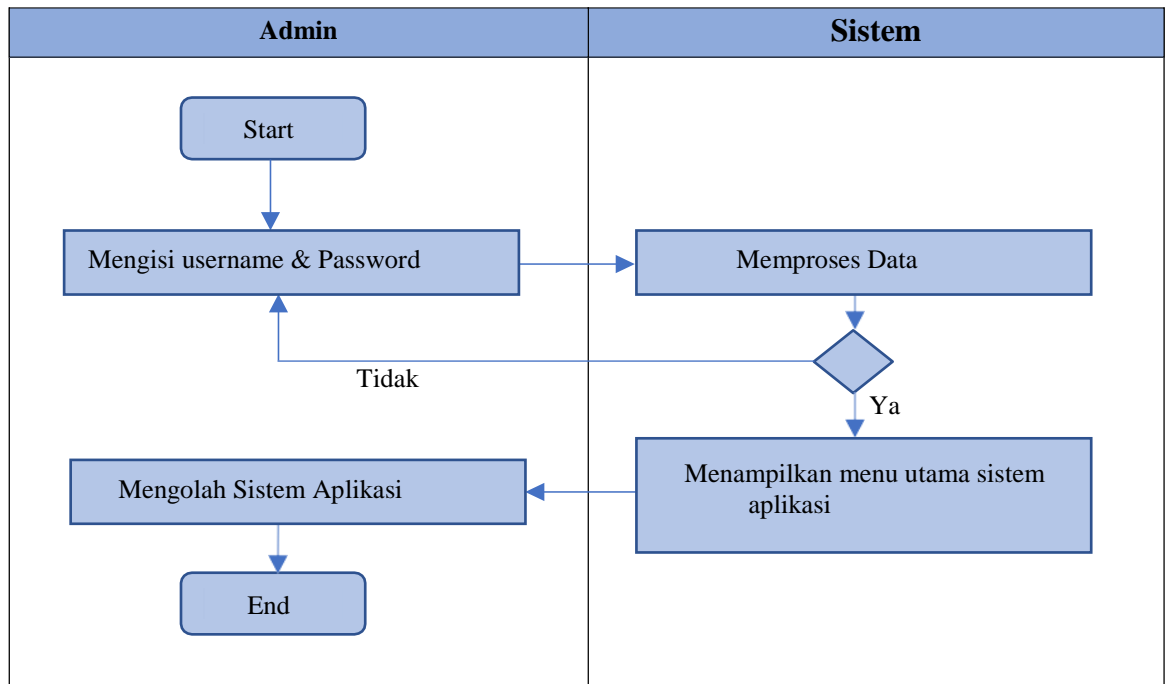
Aktor	Sistem
1. Kurir mendapatkan info Permintaan Perbaikan dalam sistem, dan melakukan penjemputan	2. Customer menerima info penjemputan barang yang akan diperbaiki
3. Admin menerima data perbaikan dalam sistem, setelah kurir mengambil barang dari customer	4. Admin mengubah status perbaikan yang akan diinfokan kepada customer
5. Kurir menerima data perbaikan selesai dan melakukan pengantaran ke alamat customer	6. Customer menerima laporan info status perbaikan

#### 4.1.3 Activity Diagram Sistem Usulan

Beberapa *activity diagram* yang diusulkan adalah sebagai berikut:

##### 1. Activity Diagram Login yang diusulkan

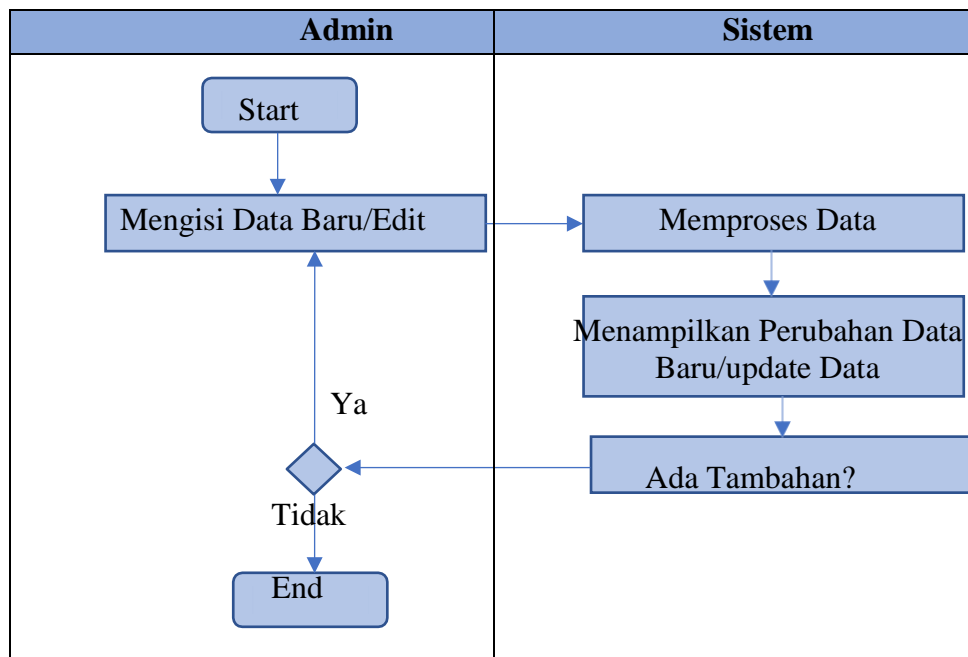
Adapun activity diagram dalam login yang diusulkan adalah sebagai berikut:



Gambar IV.2. : Activity diagram login yang diusulkan

##### 2. Activity Diagram Pengisian Data yang diusulkan

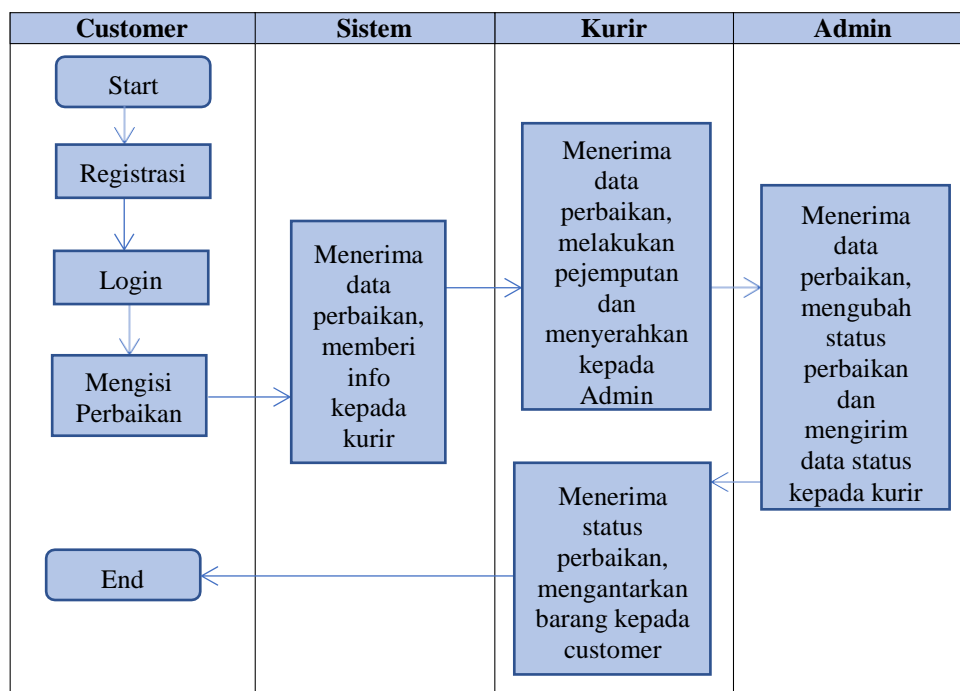
Adapun activity diagram dalam login yang diusulkan adalah sebagai berikut:



Gambar IV.3. : Activity diagram Pengisian Data yang diusulkan

3. Activity Diagram Perbaikan yang diusulkan

Adapun activity diagram dalam login yang diusulkan adalah sebagai berikut:

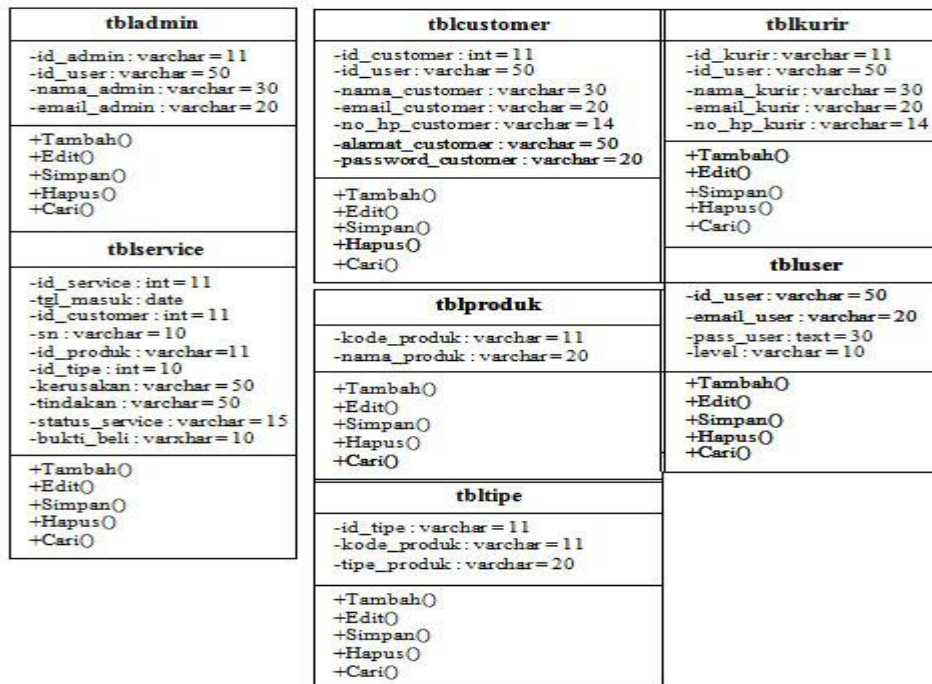


Gambar IV.4. : Activity diagram Perbaikan yang diusulkan

4.1.4 Class Diagram Sistem Usulan

Diagram kelas (*class diagram*) adalah diagram untuk menampilkan beberapa kelas yang ada dalam sistem

yang sedang dikembangkan. Adapun perancangan untuk menentukan objek yang terlibat adalah sebagai berikut:

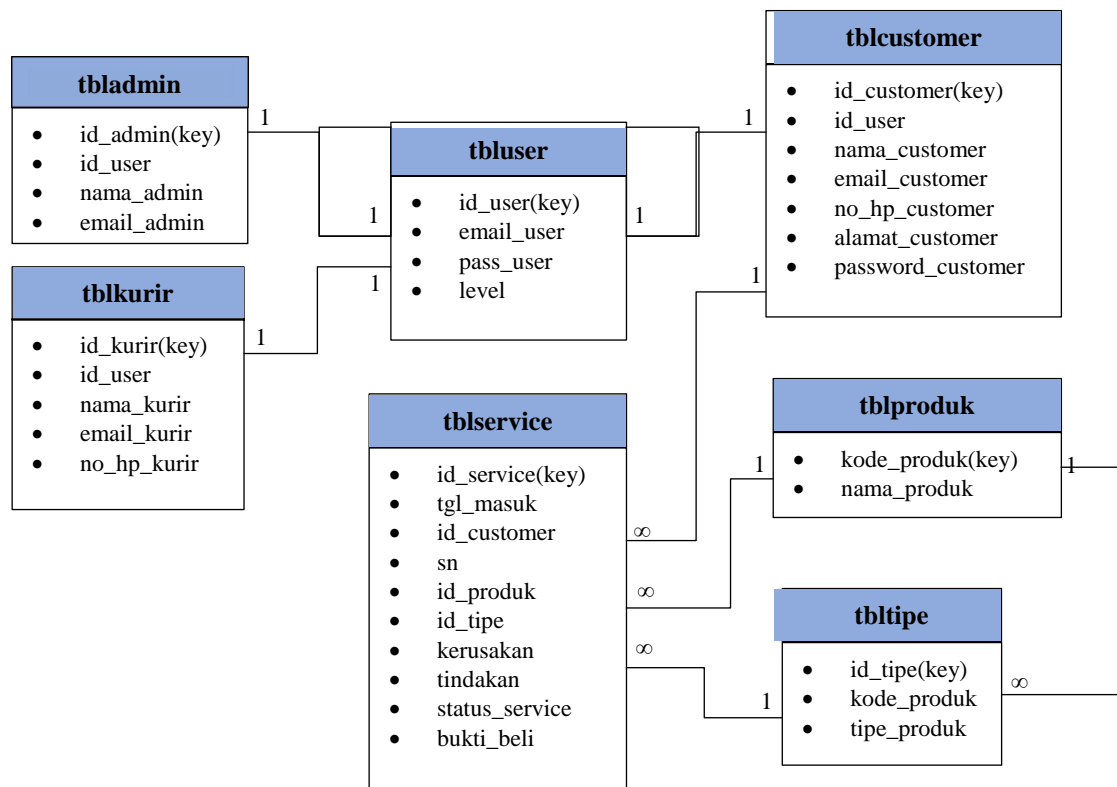


Gambar IV.5. : Kelas diagram sistem yang diusulkan

#### 4.1.5 Relasi Antar Kelas

Relasi adalah koneksi semantik antar kelas. Dengan Relasi, suatu kelas memungkinkan mengetahui atribut-

atribut, operasi-operasi serta relasi-relasi yang dimiliki kelas lainnya. Berikut adalah atribut dan relasi antar kelas yang terbentuk:



Gambar IV.6. : Relasi antar kelas/Entity Relationship Diagram (ERD)

#### 4.1.6 Perancangan Basis Data

Basis data dalam sistem aplikasi *pick up and retur* mempunyai spesifikasi dalam tiap tabelnya, penjelasannya adalah sebagai berikut:

1. Tabel Admin

Nama File : admin

Primary Key : id\_admin

Tabel IV.9 : Spesifikasi basis data tabel admin

No	Nama Field	Tipe	Ukuran	Keterangan
1	id_admin	varchar	11	Kode Admin
2	nama_admin	varchar	30	Nama Admin
3	email_admin	varchar	20	Email Admin

2. Tabel Kurir

Nama File : kurir

Primary Key : id\_kurir

Tabel IV.10 : Spesifikasi basis data tabel kurir

No	Nama Field	Tipe	Ukuran	Keterangan
1	id_kurir	varchar	11	Kode Kurir
2	nama_kurir	varchar	30	Nama Kurir
3	email_kurir	varchar	20	Email Kurir
4	no_hp_kurir	varchar	14	No Hp Kurir

3. Tabel Barang (produk)

Nama File : produk

Primary Key : id\_produk

Tabel IV.11 : Spesifikasi basis data tabel produk

No	Nama Field	Tipe	Ukuran	Keterangan
1	id_produk	varchar	11	Kode Produk
2	nama_produk	varchar	20	Nama Produk

4. Tabel Perbaikan (service)

Nama File : service

Primary Key : id\_service

Tabel IV.12 : Spesifikasi basis data tabel service

No	Nama Field	Tipe	Ukuran	Keterangan
1	id_service	integer	11	Kode Perbaikan
2	tgl_masuk	date		Tanggal Masuk
3	id_customer	integer	10	Kode Customer
4	Sn	varchar	10	Serial Number
5	id_produk	varchar	11	Kode Produk
6	id_tipe	integer	10	Kode Tipe



7	kerusakan	varchar	50	Kerusakan
8	tindakan	varchar	50	Tindakan
9	status_service	varchar	15	Status Perbaikan
10	bukti_beli	varchar	10	Bukti Beli

5. Tabel Tipe

Nama File : tipe

Primary Key : id\_tipe

Tabel IV.13 : Spesifikasi basis data tabel tipe produk

No	Nama Field	Tipe	Ukuran	Keterangan
1	id_tipe	integer	10	Kode tipe
2	id_produk	varchar	11	Kode Produk
3	tipe_produk	varchar	20	Tipe Produk

6. Tabel User

Nama File : user

Primary Key : id\_user

Tabel IV.14 : Spesifikasi basis data tabel user

No	Nama Field	Tipe	Ukuran	Keterangan
1	id_user	varchar	11	Kode user
2	nama_user	varchar	30	Nama user
3	email_user	varchar	20	Email user
4	password_user	varchar	20	Password User
5	level_user	varchar	10	Level user

7. Tabel Customer

Nama File : customer

Primary Key : id\_customer

Tabel IV.15 : Spesifikasi basis data tabel customer

No	Nama Field	Tipe	Ukuran	Keterangan
1	id_customer	integer	10	Kode Customer
2	nama_customer	varchar	30	Nama Customer
3	email_customer	varchar	20	Email Customer
4	no_hp_customer	varchar	14	No HP Customer
5	alamat_customer	varchar	50	Alamat Customer
6	password_customer	varchar	20	Password Customer

#### 4.1.7 Component Diagram

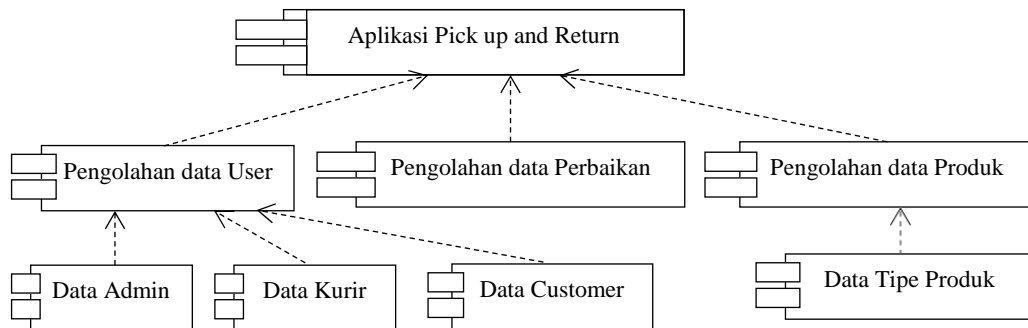
Diagram Komponen (*component diagram*) dibuat untuk menunjukkan ketergantungan diantara komponen dalam suatu sistem. Diagram komponen fokus

pada komponen sistem yang dibutuhkan dan ada di dalam sistem.

Adanya diagram komponen membuat orang-orang yang bertanggungjawab

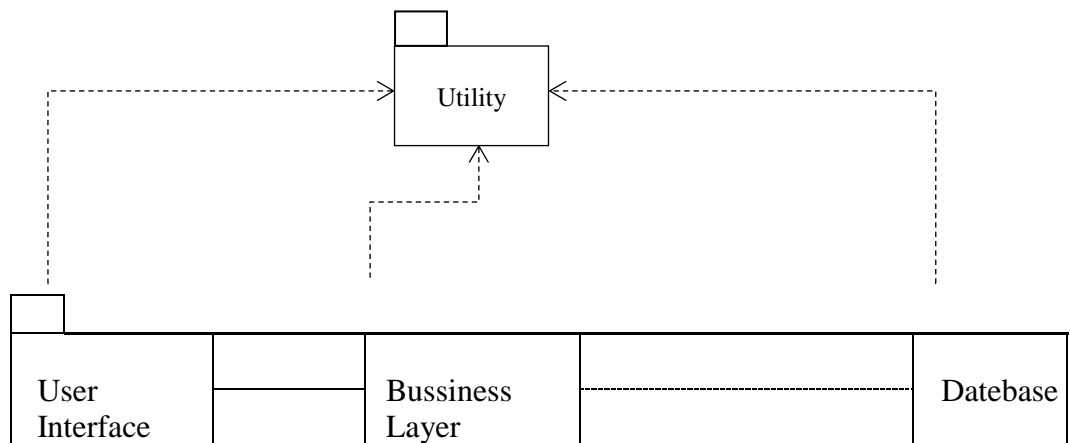
dan menyebarkan komponen antar mesin (*deplyoying*) akan mengetahui pustaka kode mana yang sudah ada dan berkas

tersukses mana yang akan diciptakan saat kode dikompilasi.



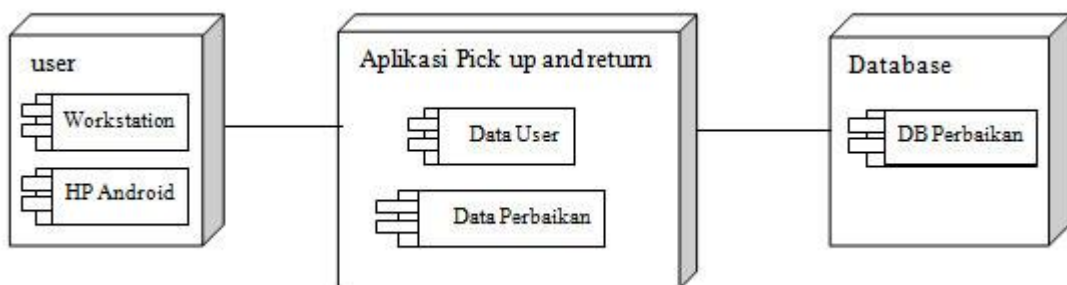
Gambar IV.7. : Komponen Diagram sistem yang diusulkan

#### 4.1.8 Package Diagram



Gambar IV.8. : Package Diagram Aplikasi Pick up and Return

#### 4.1.9 Deployment Diagram



Gambar IV.9. : Deployment Diagram Aplikasi Pick up and Return

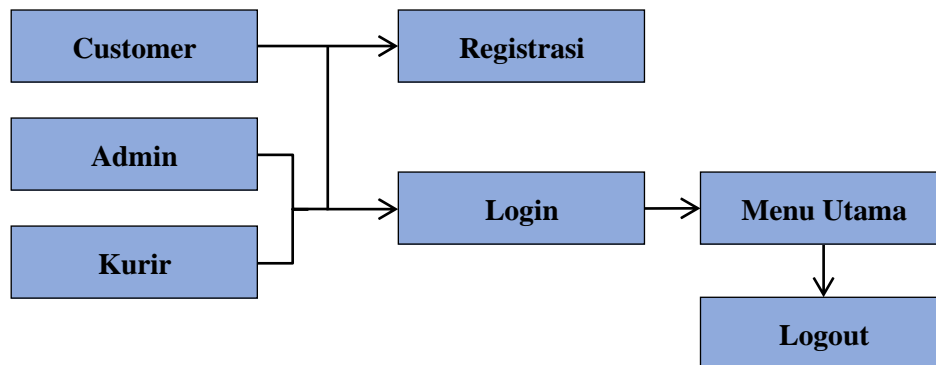
#### 4.1.10 Perancangan Antar Muka

Perancangan antar muka (*interface*) bertujuan untuk membuat tampilan sistem bersifat komunikatif dan mudah digunakan oleh pemakai (*user*), hal ini sejalan dengan tujuan dibuatnya sebuah aplikasi.

#### A. Struktur Menu

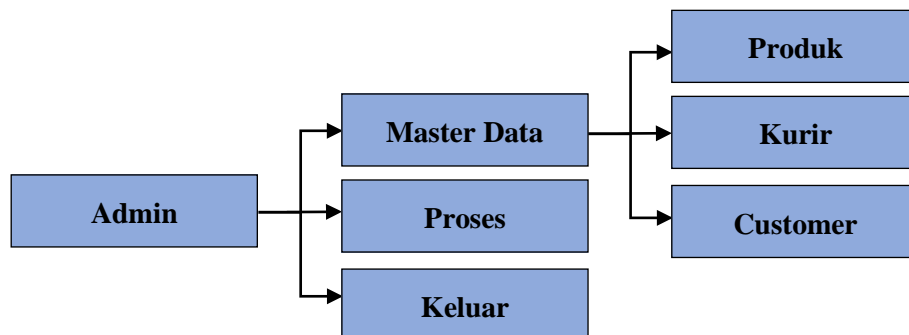
Dalam rancangan ini, pilihan menu yang ada mengintegrasikan sebuah data dalam sistem. Masing-masing *user* dibedakan melalui hak aksesnya.

##### 1. Struktur Menu Utama



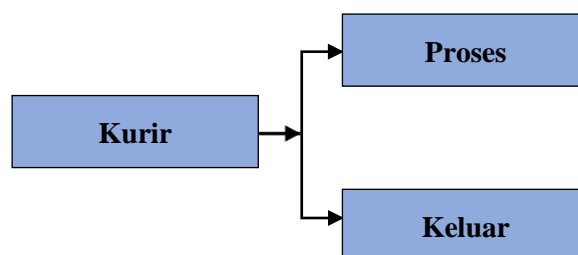
Gambar IV.10. : Struktur Menu Utama Aplikasi

##### 2. Struktur Menu Admin



Gambar IV.11. : Struktur Menu Admin

##### 3. Struktur Menu Kurir



Gambar IV.12. : Struktur Menu Kurir

#### 4. Struktur Menu Customer



Gambar IV.13. : Struktur Menu Customer

#### B. Perancangan Form

Perancangan form dipergunakan sebagai rancangan tampilan untuk penggunaan sistem dan aplikasi. Beberapa rancangan tampilan form adalah sebagai berikut:

##### 1. Form Login

Bukan hanya bagi admin yang mengatur proses *pick up and return*, form login sendiri digunakan bagi customer sebelum melakukan transaksi atau bagi kurir untuk melihat kemana lokasi pengambilan atau pengantaran barang yang akan di perbaiki.

Gambar IV.14. : Rancangan Form Login

##### 2. Form Registrasi

Form Registrasi ini diperuntukan bagi customer yang akan melakukan perbaikan, sebelum melakukan perbaikan diharuskan untuk mengisi data melalui registrasi yang selanjutnya bisa *login* dan melakukan proses perbaikan.

Gambar IV.15. : Rancangan Form Registrasi

##### 3. Form Menu Utama

Ini adalah tampilan menu utama pada sistem admin *pick up and return* dimana dalam menu ini mengatur proses status perbaikan antar jempuit.

Perbaikan	Produk	Kurir	Customer	Admin
XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX
-	-	-	-	-

Gambar IV.16. : Rancangan Form Menu Utama

##### 4. Form Menu Master Data

Form menu master data adalah rancangan untuk tampilan menu bagi admin dalam pengolahan produk, kurir, customer ataupun tipe produk.

DATA					Keluar
Profil Perusahaan	XXXXXX				
	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX
	-	-	-	-	-
	<input type="button" value="Tamba"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>				
Pick up and Return					

Gambar IV.17.: Rancangan Form Master Data

## 5. Form Menu Pengolahan Perbaikan

Form menu service berada dalam menu bagi admin untuk pengolahan proses *service* setelah case dibuat oleh customer.

Data Perbaikan					Keluar
XXXXXX	XXXXXX	XXXXXX	XXXXXX	Status	
-	-	-	-	Ubah ▼	
-	-	-	-	Ubah ▼	
Pick up and Return					

Gambar IV.18. : Rancangan Form Perbaikan

## 4.2 Implementasi Sistem

Pada tahap ini penulis menguraikan

### 1. Pengujian *Black Box*

Tabel IV.16 : Tabel Pengujian *Black Box*

No	Rancangan Proses	Hasil yang diharapkan	Hasil
1.	Halaman Awal	Tampilan awal saat program pertama kali dijalankan	Ok
2.	Halaman Registrasi	Masuk ke halaman registrasi	Ok
3.	Halaman Login	Masuk ke halaman login	Ok
4.	Menu Utama	Masuk ke menu utama	Ok
5.	Sub menu data produk	Menampilkan data produk	Ok
6.	Sub menu data kurir	Menampilkan data kurir, tambah, edit dan hapus	Ok
7.	Sub menu data <i>customer</i>	Menampilkan data <i>customer</i> , tambah, edit dan hapus	Ok

beberapa tahapan, yaitu sebagai berikut:

1. Tahap Penulisan *script*
2. Tahap Pengujian
3. Tahap Pengunggahan ke Hosting

### 4.2.1 Penulisan Script

Pembuatan sistem ini menggunakan bahasa pemrograman Hypertext Preprocessor (PHP) dan dalam pengolahan text nya menggunakan notepad++ serta XAMPP sebagai *local host* yang didalamnya meliputi MySQL yang digunakan sebagai basis datanya, kemudian dengan bantuan bootstrap untuk desainnya, dan tentu saja software Android Studio dalam pembuatan aplikasi androidnya.

### 4.2.2 Pengujian

Dalam memastikan program bebas dari kesalahan, maka dilakukan proses pengujian. Proses ini untuk mengetahui serta menyelesaikan masalah yang timbul pada proses sistem yang berjalan.

Pengujian dilakukan setelah sistem selesai dan dilakukan test. Hal ini ditujukan untuk mencari kelemahan atau kesalahan yang masih mungkin terjadi. Proses pengujian dilakukan dengan dua acara, yaitu pengujian *white box* dan *black box*.

8.	Perbaiki	Menampilkan <i>case</i> perbaikan, edit status (proses antar/jempu, selesai/dalam proses)	Ok
----	----------	---	----

## 2. Pengujian White Box

Tabel IV.16 : Tabel Pengujian White Box

Test Case	Hasil yang diharapkan	Koreksi
<p>1. Memeriksa error and event handling pada saat user login</p> <pre> 1 &lt;?php 2 session_start(); 3 include "dbcon.php"; 4 include "config.php"; 5 \$error = ""; 6 if(\$_POST){ 7     \$email_user = \$_POST['email_user']; 8     \$password = \$_POST['password']; 9     \$sql = "select * from user where level='superadmin' 10     AND email_user='\$email_user' 11     and password_user='\$password'"; 12     \$query = \$db-&gt;query(\$sql); 13     \$rs = \$query-&gt;fetch_array(MYSQLI_ASSOC); 14 15 16     if(\$query-&gt;num_rows &gt; 0){ 17         \$_SESSION['SESSION_ID'] = \$rs['id_user']; 18         \$_SESSION['level'] = \$rs['level']; 19     } 20     else{ 21         \$error = "Email atau password tidak cocok"; 22     } 23 } 24 25 \$session_id = (isset(\$_SESSION['SESSION_ID'])) ? \$_SESSION['SESSION_ID'] : null; 26 define ('SESSION_ID',\$session_id); 27 28 if(SESSION ID){ 29     header('Location: '. base_url, true); 30     exit; 31 } 32 33 &gt; </pre>	<p>1. Error handling lengkap</p> <p>2. Tersedia event handler untuk event handling</p>	<p>-</p> <p>-</p>
<p>2. Memeriksa error and event handling pada saat koneksi database</p> <pre> 1 APP_NAME=PUR 2 APP_ENV=local 3 APP_KEY=base64:7NYzrGyS0XNwe30aCbVV3f5W354XPd8rOXvd8OUYRs0= 4 APP_DEBUG=true 5 APP_LOG_LEVEL=debug 6 APP_URL=http://localhost 7 8 DB_CONNECTION=mysql 9 DB_HOST=127.0.0.1 10 DB_PORT=3306 11 DB_DATABASE=pur 12 DB_USERNAME=root 13 DB_PASSWORD= </pre>	<p>1. Error handling lengkap</p> <p>2. Tersedia event handler untuk event handling</p>	<p>-</p> <p>-</p>

### 4.2.3 Koneksi ke Internet

Sistem *pick up and return* sendiri merupakan sistem aplikasi berbasis web maka dari itu proses selanjutnya adalah mengunggahnya ke internet agar dapat pula terhubung dengan android. Ada beberapa cara pengunggahan file ke internet, diantaranya menggunakan File Manager melalui CPanel atau *Upload* menggunakan software FTP. Disini penulis akan memberikan cara untuk pengunggahan dengan cara yang pertama yaitu menggunakan File Manager melalui CPanel.

Cara Upload File Website dengan File Manager

1. Pastikan kita sudah memiliki *account* di web hosting
2. Login ke Cpanel



Gambar IV.19. : Upload Menggunakan CPanel 1

3. Upload dulu basis data (*database*) dari program kita, dengan klik phpMyAdmin



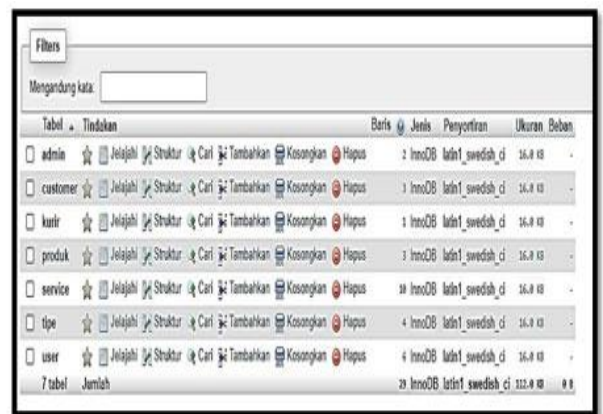
Gambar IV.19. : Upload Menggunakan CPanel 2

4. Pada jendela phpMyAdmin, *upload* file *database* kita dengan format *.sql*



Gambar IV.19. : Upload Menggunakan CPanel 3

5. Sampai muncul tabel *database* kita dalam jendela phpMyAdmin



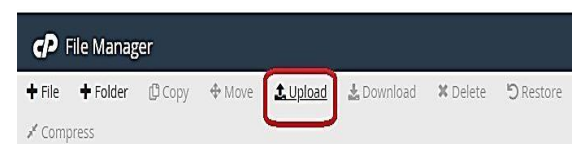
Gambar IV.19. : Upload Menggunakan CPanel 4

6. Setelah *database* masuk kedalam cpanel php MyAdmin, langkah selanjutnya adalah memasukan file php kita kedalam file manager.



Gambar IV.18. : Upload Menggunakan CPanel 5

7. Pilih *upload* pada jendela *file manager*



Gambar IV.19. : Upload Menggunakan CPanel 6

8. Sara *upload* bisa dengan *Drop files* atau *select files*. Tunggu sampai proses selesai.



Gambar IV.19. : Upload Menggunakan CPanel 7

9. Setelah proses selesai, maka gantilah *base url* pada program kita dengan domain yang sudah kita miliki.

Berikutnya kita daftarkan Aplikasi Android ke Google Play dengan langkah sebagai berikut:

1. Membuat akun di Google Play Developer Console (Ini cukup jika kita ingin mengunggah aplikasi gratis)
2. Di halaman utama Google Play Developer Console klik tombol “Publish an Android App on Google Play”
3. Memilih “Bahasa Indonesia” Sebagai *Default Language*.
4. Klik “Upload APK”
5. Pada bagian “Store Listing” di detail aplikasi, masukan informasi tentang aplikasinya.
6. *Setting* Aplikasi menjadi “gratis”.
7. Setelah bagian “APK”, “Store Listing” dan “Pricing & Distribution” tercentang

hijau, maka aplikasi bisa dirilis.

## 5. PENUTUP

### 5.1 Kesimpulan

Setelah semua tahap penelitian dilakukan, maka penulis dapat mengambil kesimpulan sebagai berikut:

1. Penelitian ini mampu merancang dan membangun sebuah sistem aplikasi layanan terbaru dalam proses perbaikan antar jemput (*Pickup and Return*) pada PT. Sinar Elok Abadi, Jakarta.
2. Sistem aplikasi ini mampu membuat mempermudah *customer* saat akan melakukan perbaikan, karena tidak perlu datang cukup menggunakan aplikasi lalu unit dijemput oleh kurir, kemudian akan diantarkan bila sudah selesai diperbaiki.
3. Sistem aplikasi ini bagi PT. Sinar Elok Abadi dapat mempermudah dalam pendaftaran data perbaikan.

### 5.2 Saran

Dalam upaya untuk mengembangkan Sistem Aplikasi *Pickup and Return* berbasis Android ini, ada beberapa hal yang penulis sarankan:

1. Sistem aplikasi ini bisa digabungkan agar bisa diakses juga oleh tim teknisi ketika melakukan proses perbaikan.
2. Sistem aplikasi ini bisa ditambahkan fitur *chat* agar bisa melakukan *chat*/obrolan kepada teknisi melalui aplikasi ini ketika melakukan proses perbaikan.

## Daftar Pustaka

1. Jogiyanto. 2005. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Penerbit Andi
2. Al-Bahri bin Lajadmudin. 2005. *Analisis Sistem Informasi*, Graha Ilmu.
3. Sutarmam. 2009. *Pengantar Teknologi Informasi*. Jakarta : Bumi Aksara
4. Hendrayudi. 2009. *VB 2008 untuk keperluan pemrograman*. Yogyakarta : Andi
5. [http://repo.unnes.ac.id/dokumen/pemrograman-web/pw2\\_pertemuan10.pdf](http://repo.unnes.ac.id/dokumen/pemrograman-web/pw2_pertemuan10.pdf)
6. <http://informatika.web.id/pengertian-uml.htm>
7. <http://www.uml-diagrams.org/uml-25-diagrams.html>
8. <http://ocw.ui.ac.id/>
9. Arif M. Rudianto. 2011. *Pemrograman Web Dinamis Menggunakan PHP*. Yogyakarta : Andi
10. Bunafit, Nugroho. 2006. *Membuat Aplikasi Sistem Pakar menggunakan PHP*. Yogyakarta : Ardana Media
11. [https://www.android.com/intl/id\\_id/versions/nougat-7-0/](https://www.android.com/intl/id_id/versions/nougat-7-0/)



12. Nazaridin Safaat Harahap. 2012. *Pemograman Aplikasi Mobile Smartphone dan tablet PC Berbasis Android*. Bandung : Informatika
13. Alfa Satyaputra & Eva Maulina Aritonang. 2014. *Beginning Android Programming with ADT Budle*. Jakarta : PT. ElexMedia Komputindo
14. Mulyanto, Agus. 2009. *Sistem Informasi Konsep dan Aplikasi*. Yogyakarta : Pustaka Pelajar
15. <http://blog.unnes.ac.id/ayukwitantri/2016/02/25/pengertian-basis-data-tujuan-manfaat-dan-contoh-kasus-basis-data/>

