

# RANCANG BANGUN APLIKASI PROFIT PENJUALAN MOTOR BERBASIS DESKTOP KONSEP ARSITEKTUR *MODEL VIEW CONTROLLER* (MVC)

Ibnu Alfitra Salam<sup>\*</sup>, Kamal Prihandani<sup>2</sup>, Intan Purnamasari<sup>3</sup>

Program Studi Informatika, Universitas Singaperbangsa Karawang

*Riwayat artikel:*

*Received: 13 Agustus 2023*

*Accepted: 28 Agustus 2023*

*Published: 11 September 2023*

## Keywords:

profit;  
*waterfall*;  
java;  
mvc.

## Correspondent Email:

1910631170085@student.unsi  
ka.ac.id

**Abstrak.** Profit merupakan indikator utama pada suatu perusahaan untuk menilai performa rasio keuangannya. Sebagian besar perusahaan sudah mengenali konsep dari profitabilitas, namun untuk pengelolaan profit dengan menggunakan aplikasi masih terbilang masih kurang. Sering kali terjadi kesalahan perhitungan pada sistem sehingga perhitungan kurang akurat. Melihat permasalahan yang terjadi sebuah aplikasi berbasis desktop bisa dijadikan alternatif dalam perhitungan profit perusahaan. Dengan menggunakan metode pengembangan *waterfall* pada perancangan sebuah sistem dapat berlangsung secara cepat dan tepat. Pemrograman yang digunakan yaitu Java dengan *tool* Netbeans IDE 8.2 dan MySQL sebagai tempat penyimpanan data yang diproses. Dari pengujian yang dilakukan dengan teknik *basis path* dan perhitungan *cyclomatic complexity* didapatkan hasil keberhasilan 100% untuk sistem tersebut. Angka tersebut sudah mencapai sempurna untuk sebuah sistem yang dikembangkan dan dapat diterapkan dengan baik kepada perusahaan.

**Abstract.** Profit is the main indicator for a company to assess the performance of its financial ratios. Most companies already recognize the concept of profitability, but profit management using applications is still lacking. Often there are calculation errors in the system so that the calculations are less accurate. Seeing the problems that occur in a desktop-based application can be used as an alternative in calculating company profits. By using the *waterfall* development method in designing a system, it can take place quickly and precisely. The programming used is Java with the Netbeans IDE 8.2 tool and MySQL as the storage location for processed data. From the tests carried out using the *basis path* technique and *cyclomatic complexity* calculations, a 100% success result was obtained for the system. This figure has reached perfect for a system that is developed and can be applied properly to the company.

## 1. PENDAHULUAN

Profit adalah indikator utama kelangsungan bisnis, sebuah rasio keuangan yang sering digunakan untuk menilai performa suatu perusahaan. Profit telah menjadi salah satu perhatian utama perusahaan dan manajer mereka. Profit sering kali dianggap sebagai

salah satu persyaratan penting untuk menjaga keberlanjutan dan mencapai kesuksesan dalam jangka panjang suatu perusahaan. Penjelasan lain tentang pentingnya profit adalah pengaruhnya terhadap pertumbuhan ekonomi, lapangan kerja, inovasi, dan perubahan teknologi. Oleh karena itu, pengelolaan

keuangan yang baik sangat penting agar perusahaan dapat memperoleh profit yang optimal dan mengelola risiko keuangan dengan baik. Namun, pengelolaan keuangan yang baik tidak selalu mudah, terutama bagi pemilik usaha kecil dan menengah yang mungkin memiliki sumber daya dan pengelolaan keuangan yang terbatas.

Menurut Fanny[1], sebagian besar perusahaan telah mengenali pentingnya konsep profitabilitas, namun demikian, perusahaan-perusahaan tersebut mungkin tidak memiliki pengetahuan tentang cara meningkatkan laba dan faktor-faktor apa yang berdampak pada hal tersebut. Dalam beberapa tahun terakhir, kemajuan teknologi telah membuat pengelolaan keuangan menjadi lebih mudah dan terjangkau bagi pemilik bisnis kecil dan menengah. Salah satu cara di mana teknologi telah membantu pengelolaan keuangan adalah melalui pengembangan aplikasi profit yang memungkinkan pemilik bisnis untuk memonitor dan mengelola keuangan mereka dengan lebih efektif.

Penggunaan aplikasi profit dapat memberikan banyak keuntungan bagi perusahaan dalam mengelola keuangan dan meningkatkan profitabilitas bisnisnya. Beberapa keuntungan penggunaan aplikasi profit bagi perusahaan diantaranya meningkatkan efisiensi pengelolaan keuangan, mempermudah pemantauan kinerja keuangan, meningkatkan akurasi perhitungan profit, meningkatkan transparansi keuangan, dan mempercepat proses pelaporan keuangan. Penggunaan aplikasi profit dapat memberikan banyak keuntungan bagi perusahaan dalam mengelola keuangan dan meningkatkan profitabilitas bisnisnya. Oleh karena itu, banyak perusahaan untuk menggunakan aplikasi profit sebagai salah satu alat dalam mengelola keuangan dan membuat keputusan bisnis yang lebih tepat.

Berdasarkan analisis aplikasi profit yang sudah dilakukan pada CV. Tridjaya Anugerah Motor, aplikasi ini masih memiliki beberapa aspek yang perlu ditingkatkan dan ditambahkan. Dimana hal tersebut menyebabkan tidak sesuainya hasil akhir dari sistem dengan laporan yang seharusnya. Dari hasil wawancara langsung dengan pengguna yang beroperasi langsung dengan sistemnya, tampilan *interface* masih banyak ruang kosong

yang jarang dipakai, sering kali harus memasukkan data kembali karena tidak dapat menghapus data yang sudah dimasukkan ke dalam tabel, dan sering kali perhitungan akhir profit gross yang tertera pada sistem berubah-ubah yang disebabkan oleh fungsi tombol yang mengalami bug.

Beberapa penelitian sebelumnya yang telah menciptakan berbagai aplikasi menggunakan *platform* desktop yang mampu memenuhi kebutuhan seperti yang telah dianalisis sebelumnya. Seperti, penelitian yang merancang sebuah aplikasi untuk mengatur penggunaan peralatan di jurusan pendidikan teknik elektro melalui komputer desktop. Penelitian tersebut menghasilkan aplikasi yang sesuai dengan tujuan dari segi fungsionalnya dan memberikan kemudahan dalam pengelolaan pendataan peralatan yang ada[2]. Ada juga penelitian yang merancang aplikasi pengelolaan data penjualan sembako berbasis desktop. Aplikasi tersebut dapat memudahkan admin dalam mengelola penjualan sembako dengan fitur yang sudah sesuai dengan analisis yang sudah dilakukan oleh peneliti[3].

Berdasarkan permasalahan yang sudah dipaparkan maka dalam penelitian ini akan berusaha untuk membangun aplikasi desktop yang memenuhi kebutuhan yang telah dianalisis sebelumnya. Dengan berbasis desktop aplikasi akan sulit diakses dari pihak luar atau jarak jauh untuk mengurangi pengambilan database yang penting. Aplikasi juga sulit disebarluaskan karena harus melakukan instalasi terlebih dahulu dan penyesuaian agar bisa berjalan lancar. Kehadiran sistem aplikasi yang memadai dan akurat dapat mengurangi kesalahan yang tidak diinginkan dan dapat meningkatkan efisiensi dan kecepatan operasional perusahaan. Maka dari itu, penelitian ini akan membuat suatu sistem berjudul “Rancang Bangun Aplikasi Profit Penjualan Motor Berbasis Desktop Konsep Arsitektur *Model View Controller* (MVC)”.

## 2. TINJAUAN PUSTAKA

### Rancang Bangun

Rancang bangun merupakan kegiatan/proses yang dilakukan untuk mengilustrasikan cara kerja suatu proses bisnis dengan membuat gambaran melalui diagram seperti use case diagram. *System Development Life Cycle* (SDLC) adalah suatu metode yang umum

digunakan dalam proses desain atau pengembangan sistem programmer yang sudah ada. Tahapan SDLC-nya terdiri dari lima bagian yaitu penelitian, analisis, desain, pengkodean, dan pemeliharaan serta evaluasi[4].

Rancang bangun adalah proses mengimplementasikan hasil analisis menjadi perangkat lunak dan memodifikasi atau memperbaiki sistem yang telah ada. Rancang bangun melibatkan proses merencanakan dan membuat gambar sketsa atau menyusun beberapa elemen yang terpisah menjadi satu kesatuan yang konsisten dan dapat berfungsi dengan baik[5].

Pada dasarnya, rancang bangun adalah proses merencanakan untuk membangun atau mengatur kembali struktur yang sudah ada, dengan tujuan menghasilkan suatu proyek yang baru.

#### **Aplikasi**

Aplikasi adalah suatu sistem atau program yang dirancang dengan tujuan untuk memfasilitasi tugas-tugas manusia[6]. Aplikasi juga bisa dikatakan sebagai perangkat lunak yang menggabungkan berbagai macam fitur sesuai dengan kebutuhan perusahaan. Menurut para ahli, kata "aplikasi" memiliki banyak arti. Berikut adalah beberapa penjelasan mengenai program yang diberikan oleh beberapa pakar yang cukup terkenal:

##### 1. Ali Zaki dan *Smitdev Community*

Menurut Ali Zaki dan *Smitdev Community*, aplikasi merupakan suatu elemen yang berfungsi sebagai alat untuk memproses data atau melaksanakan berbagai tugas lainnya seperti pembuatan atau perubahan dokumen dan berkas.

##### 2. Sri Widianti

Sri Widianti menjelaskan bahwa aplikasi merupakan sebuah *software* yang berfungsi sebagai antarmuka dari suatu sistem yang digunakan untuk mengelola berbagai data sehingga dapat menjadi informasi bermanfaat bagi pengguna dan sistem yang terkait.

##### 3. Harip Santoso

Menurut Harip Santoso, aplikasi merupakan sekumpulan berkas (seperti kelas, formulir, dan laporan) yang dibuat dengan tujuan untuk menjalankan berbagai fungsi terkait.

##### 4. Yuhefizar

Menurut Yuhefizar, aplikasi adalah perangkat lunak yang dirancang dan

dikembangkan dengan tujuan untuk memenuhi kebutuhan pengguna ketika melakukan suatu tugas khusus.

##### 5. Hengky W. Pramana

Berdasarkan pendapat Hengky W. Pramana, aplikasi merupakan sebuah program komputer yang secara khusus dirancang untuk memenuhi keperluan berbagai macam aktivitas atau pekerjaan, seperti bisnis, periklanan, layanan publik, *game*, dan berbagai jenis kebutuhan lainnya yang dibutuhkan oleh manusia.

#### **Desktop**

Aplikasi desktop merupakan perangkat lunak atau aplikasi yang dimiliki oleh komputer desktop (PC dan laptop) dan dapat berjalan tanpa mengharuskan koneksi internet (*offline*). Untuk dapat menggunakannya, pengguna perlu memasangnya terlebih dahulu di dalam sistem operasi laptop atau komputer. Program desktop dikembangkan dengan tiga bahasa pemrograman yang berbeda, yakni .Net, Java dan Delphi. Pemrograman .Net mencakup beberapa bahasan pemrograman seperti Visual Basic (VB), C dan C# (C Sharp).

Aplikasi desktop memiliki berbagai fitur yang bervariasi tergantung pada kebutuhan dan tujuan penggunaannya. Keuntungan pada saat menggunakan aplikasi desktop diantaranya kinerja yang lebih cepat, kemampuan akses *offline*, fungsi yang lebih lengkap, dan lebih aman untuk digunakan.

#### **Java**

Bahasa pemrograman Java merupakan hasil karya pertama dari James Gosling. Bahasa Java merupakan hasil pengembangan dari bahasa pemrograman C, Java sangat populer saat ini karena penggunaan *multi platform*, dapat digunakan di berbagai jenis sistem informasi, dan Java memiliki library yang lengkap[7].

Java juga memiliki banyak fitur yang sangat berguna untuk mengembangkan aplikasi, termasuk platform standar yang menyediakan sejumlah besar fungsi dan *library* yang dapat digunakan pengembang dan platform khusus untuk mengembangkan aplikasi web dan server. Selain itu, Java juga mendukung *multi-threading*, memungkinkan pengembang membuat program yang dapat menjalankan banyak proses secara bersamaan. NetBeans 8.2 merupakan salah satu IDE terpopuler dan sering digunakan oleh programmer dalam mengembangkan perangkat

lunak baik dalam bahasa pemrograman Java, PHP, dan C/C++.

### Model View Controller (MVC)

Menurut Badiyanto, ada tiga jenis komponen untuk aplikasi MVC yaitu:

#### 1. Model

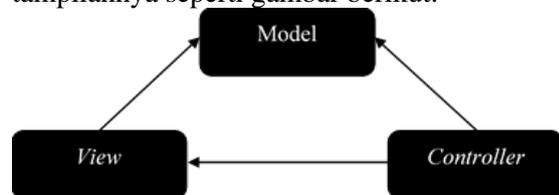
Model adalah kelas logika proses dan kelas perangkat lunak terkait. Model adalah objek yang tidak berisi informasi tentang antarmuka pengguna. Model juga merupakan kelas yang berisi metode/fungsi dan digunakan untuk menyimpan data dan aturan bisnis yang relevan.

#### 2. View

View adalah kumpulan kelas yang mewakili elemen antarmuka pengguna, tampilan memiliki nama yang digunakan untuk mengidentifikasi file skrip tampilan saat dipanggil oleh fungsi rendering. Nama view sama dengan nama berkas skrip view-nya.

#### 3. Controller

Controller adalah kelas antarmuka tampilan model yang digunakan untuk berkomunikasi antara model dan view. Controller memiliki fungsi standar. Jika permintaan pengguna tidak menentukan fungsi mana yang harus dijalankan, program akan menjalankan fungsi standar. Setelah alur kerja MVC dipetakan, tampilannya seperti gambar berikut:



Gambar 1. Konsep Model View Controller

### White Box Testing

White box testing merupakan sebuah teknik pengujian aplikasi yang melibatkan pemahaman terhadap struktur kontrol dalam level komponen desain, dengan tujuan untuk menghasilkan kasus uji. White box memiliki beberapa metode dalam pengujian, seperti: *Data Flow Testing*, *Control Flow Testing*, *Basic Path/Pat Testing*, dan *Loop Testing*. Dalam White Box Testing, seorang penguji harus mengetahui kode sumber yang akan diuji. White box testing memiliki potensi untuk menemukan kekurangan dalam implementasi aplikasi. Tes ini dapat diterapkan pada tingkat integrasi, unit dan sistem[8].

### Black Box Testing

Pressman menyatakan bahwa *black box testing* mengkhhususkan diri pada fitur-fitur yang harus dipenuhi oleh perangkat lunak. Dengan demikian, metode *black box testing* memperbolehkan pengembang untuk menciptakan serangkaian situasi input yang akan menguji segala persyaratan fungsional dari program. *Black box testing* bukanlah pengganti pengujian *white box*, tetapi merupakan pendekatan tambahan untuk mendeteksi kesalahan lain selain melalui metode *white box testing*[9]. *Black box testing* berupaya menemukan bug dalam beberapa kategori, antara lain:

1. Fungsi yang salah atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau database eksternal
4. Kesalahan kinerja
5. Kesalahan inisialisasi dan terminasi

### System Development Life Cycle (SDLC) Model Waterfall

Metode SDLC adalah pendekatan sistematis dalam pengembangan sistem yang terdiri dari serangkaian tahapan yang saling terkait dan berurutan. Metode ini juga dikenal sebagai metode siklus hidup perangkat lunak. Metode SDLC digunakan untuk mengembangkan sistem informasi, baik itu aplikasi desktop, aplikasi *mobile*, maupun aplikasi web. Ada beberapa model SDLC yang sering digunakan di antaranya yaitu Model *Waterfall*, Model *Spiral*, Model RAD (*Rapid Application Development*), dan Model *Agile*.

Penggunaan metode SDLC sangat penting dalam pengembangan sistem informasi karena dapat membantu tim pengembang dalam mengorganisasi pekerjaan, memastikan kualitas hasil akhir, dan mengoptimalkan penggunaan sumber daya. Selain itu, metode SDLC juga membantu dalam menjamin keamanan sistem dan mengurangi risiko kesalahan pada pengembangan aplikasi.

Metode *waterfall* adalah metode yang paling sering digunakan dalam proses pengembangan sistem informasi. Model ini termasuk ke dalam model umum *software engineering* dan diperkenalkan pertama kali pada tahun 1970 oleh Winston Royce. Oleh karena itu, seringkali dianggap sudah ketinggalan zaman, padahal model ini menjadi pilihan utama dalam dunia rekayasa perangkat lunak. Model ini mengadopsi pendekatan yang

teratur dan berurutan. Dikenal sebagai *waterfall* karena langkah-langkahnya harus menunggu sampai langkah sebelumnya selesai dan dilakukan secara berurutan[10]. Setiap langkah saling terkait, karena keluaran dari langkah pertama menjadi masukan dari langkah berikutnya, sehingga pekerjaan dari setiap langkah harus dilakukan dengan benar. Penggunaan metode ini lebih terorganisir, karena setiap langkah memiliki proses dan dokumen, sehingga dapat dikembangkan dengan mudah sesuai kebutuhan pengguna[11].

Adapun tahapan yang ada pada tahapan model *waterfall* sebagai berikut:

#### 1. *Requirement Analysis*/Analisis Kebutuhan

Tahapan analisis kebutuhan adalah tahapan awal dalam pengembangan sistem. Pada tahapan ini, tim pengembang melakukan identifikasi kebutuhan dan memahami masalah yang ingin dipecahkan oleh sistem yang akan dikembangkan. Selain itu, tim pengembang juga melakukan pengumpulan data dan informasi yang diperlukan untuk membangun sistem yang efektif. Tahapan ini melibatkan kerja sama antara pengguna dan pengembang untuk memastikan kebutuhan pengguna terpenuhi.

#### 2. *System Design*/Desain Sistem

Setelah tahapan analisis selesai, tim pengembang mulai merancang solusi untuk memecahkan masalah yang telah diidentifikasi. Tahapan desain sistem ini mencakup desain arsitektur sistem, desain basis data, desain *user interface*, dan desain algoritma yang digunakan. Desain yang dihasilkan pada tahapan ini akan menjadi acuan untuk mengembangkan sistem pada tahapan selanjutnya.

#### 3. *Coding*/Pengkodean

Tahapan pengkodean melibatkan tim pengembang untuk memulai pembuatan kode program aplikasi yang sesuai dengan desain yang telah disusun sebelumnya. Program dijalankan menggunakan bahasa pemrograman spesifik yang disesuaikan dengan keperluan.

#### 4. *Testing*/Pengujian

Setelah sistem selesai diperbarui, langkah pengujian dilakukan guna memastikan bahwa sistem beroperasi sejalan dengan standar yang telah ditetapkan.. Tahapan pengujian ini mencakup pengujian fungsional, pengujian kinerja, pengujian keamanan, dan pengujian

integrasi. Setelah pengujian selesai dilakukan, maka sistem siap untuk diimplementasikan.

#### 5. *Maintenance*/Pemeliharaan

Tahapan pemeliharaan adalah tahapan di mana sistem yang telah diimplementasikan diperbaiki dan ditingkatkan secara berkala. Pada tahapan ini, tim pengembang melakukan perbaikan *bug*, meningkatkan fungsionalitas sistem, serta mengoptimalkan kinerja sistem.

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

UML ialah suatu bahasa yang digunakan untuk menentukan, menggambarkan, membangun, dan mencatat komponen-komponen perangkat lunak. Artefak-arterfak ini mencakup model-model, penjelasan, atau perangkat lunak dalam sistem perangkat lunak, seperti aplikasi bisnis dan lainnya. Di samping itu, UML merupakan satu bentuk bahasa pemodelan yang menerapkan prinsip orientasi objek. Grady Booch, James Rumbaugh, dan Ivar Jacobson menciptakan UML dalam kerangka *Rational Software Corps*. UML menawarkan simbol-simbol untuk membantu dalam menggambarkan sistem dari sudut pandang yang berbeda. UML digunakan bukan hanya dalam pemodelan perangkat lunak, tetapi juga dalam bidang lain yang memerlukan pemodelan.[12].

#### **Use Case Diagram**

*Use case diagram* dipakai untuk menggambarkan interaksi antara sistem dengan aktor yang terlibat dalam menggunakan sistem. *Use case diagram* mengilustrasikan situasi penggunaan sistem dengan visual, yang membantu tim pengembang memahami kebutuhan pengguna dan memastikan bahwa sistem yang dikerjakan memenuhi kebutuhan tersebut.

Tabel 1. Simbol *Use Case Diagram*

Simbol	Nama	Deskripsi
	<i>Actor</i>	Mewakili peran individu, sistem lain, atau alat ketika berinteraksi dengan skenario penggunaan

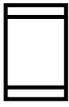
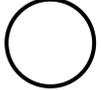
	<i>Use Case</i>	Interaksi sistem dan aktor dipresentasikan melalui abstraksi.
	<i>Association</i>	Penghubung antara aktor dan <i>use case</i> dapat dijelaskan sebagai abstraksi.
	<i>Generalisasi</i>	Menunjukkan kemampuan khusus aktor untuk dapat terlibat dalam skenario penggunaan
<i>&lt;include&gt;</i> 	<i>Include</i>	Menunjukkan bahwa satu <i>use case</i> sepenuhnya memuat fungsionalitas dari <i>use case</i> lainnya
<i>&lt;extend&gt;</i> 	<i>Extend</i>	Menyatakan bahwa sebuah <i>use case</i> menjadi tambahan fungsional dari <i>use case</i> lainnya apabila suatu kondisi dipenuhi.

Sumber: <https://www.jagoanhosting.com>

### **Class Diagram**

*Class diagram* digunakan untuk mewakili struktur sistem dengan menggunakan kelas-kelas yang memiliki kaitan satu sama lain. *Class diagram* menunjukkan atribut dan metode dari setiap kelas, serta hubungan antara kelas-kelas tersebut. *Class diagram* juga dapat digunakan untuk memodelkan pola-pola desain yang digunakan dalam pengembangan sistem.

Tabel 2. Simbol *Class Diagram*

Simbol	Nama	Deskripsi
	<i>Class</i>	Kelas yang ada dalam kerangka sistem
 nama_ <i>interfa</i> <i>ce</i>	<i>Interface</i>	Mirip dengan konsep <i>interface</i> dalam pemrograman berbasis objek
	<i>Association</i>	Hubungan antara kelas dengan arti umum, biasanya didukung dengan kemungkinan adanya hubungan berganda.
	<i>Directed Association</i>	Hubungan antara kelas dengan arti kelas yang satu digunakan oleh kelas yang lain, asosiasi seringkali juga dilengkapi dengan <i>multiplicity</i>
	<i>Generalisasi</i>	Hubungan antara kelas-kelas dalam konteks ini melibatkan konsep <i>generalisasi-spesialisasi</i> , yang berarti menghubungkan konsep umum dengan konsep yang lebih spesifik
	<i>Dependency</i>	Hubungan antara kelas satu dengan yang lain

	<i>Aggregation</i>	memiliki arti penting sebagai saling bergantung satu sama lain Relasi di antara kelas dengan signifikansi seluruh-bagian
--	--------------------	---

Sumber: Jurnal (Heriyanto, 2018)

**Activity Diagram**

*Activity diagram* dipergunakan untuk menggambarkan jalannya tugas dalam sebuah proses bisnis atau fungsi di dalam sistem. *Activity diagram* menunjukkan aktivitas-aktivitas yang harus dilakukan dalam proses bisnis, serta alur kerja dari aktivitas-aktivitas tersebut. *Activity diagram* membantu tim pengembang dalam memahami proses bisnis secara visual, sehingga memudahkan dalam pengembangan sistem yang efektif dan efisien.

Tabel 3. Simbol *Activity Diagram*

Simbol	Nama	Deskripsi
	<i>Action</i>	Untuk menjelaskan tindakan yang simpel dan tidak dapat diuraikan
	<i>Activity</i>	Untuk menggambarkan suatu tindakan yang mudah dilakukan dan sulit dijelaskan secara rinci
	<i>Object Node</i>	Objek yang terhubung dengan kumpulan aliran objek dapat diwakili
	<i>Control Flow</i>	Menunjukkan langkah-langkah pelaksanaan
	<i>Object Flow</i>	Menggambarkan bagaimana suatu objek bergerak dari satu aktivitas atau tindakan ke

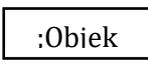
	<i>Initial Node</i>	aktivitas atau tindakan lainnya Mengindikasikan permulaan suatu rangkaian tindakan atau kegiatan
	<i>Final-Activity Node</i>	Untuk menghentikan semua aliran kontrol atau aliran objek pada suatu aktivitas (atau tindakan)
	<i>Final-Flow Node</i>	Untuk menghentikan aliran kontrol atau aliran objek tertentu
	<i>Decision Node</i>	Dalam rangka mewakili kondisi pengujian, tujuan utamanya adalah memastikan bahwa aliran kontrol atau objek hanya berjalan dalam satu arah.
	<i>Merge Node</i>	Untuk kembali menggabungkan jalur keputusan yang terbentuk dengan menggunakan simpul keputusan

Sumber: <https://www.nesabamedia.com>

**Sequence Diagram**

Sequence diagram digunakan untuk memodelkan urutan dari aktivitas-aktivitas dalam sebuah proses yang melibatkan beberapa objek dalam sistem. Sequence diagram menunjukkan bagaimana pesan atau perintah diproses oleh objek-objek dalam sistem. Sequence diagram membantu tim pengembang dalam memahami bagaimana objek-objek dalam sistem berinteraksi satu sama lain dalam urutan tertentu.

Tabel 4. Simbol *Sequence Diagram*

Simbol	Nama	Fungsi
	<i>Object</i>	Komponen utama <i>Sequence Diagram</i>
	<i>Actor</i>	Menggambarkan individu yang sedang terlibat dalam komunikasi dengan suatu sistem
	<i>Entity Class</i>	Menggambarkan ikatan antara kegiatan yang akan dilaksanakan
	<i>Boundary Class</i>	Menggambarkan suatu representasi visual atau gambaran dari bentuk atau format yang disajikan
	<i>Control Class</i>	Menjelaskan hubungan antara batas dengan tabel
	<i>Life Line</i>	Mendeskrripsikan awal dan akhir dari suatu pesan
	<i>Message</i>	Menggambarkan pengiriman pesan

Sumber: <https://devyferdiansyah.com>

### State Diagram

*State diagram* adalah bentuk visualisasi yang digunakan dalam *Unified Modeling Language* (UML) untuk mengilustrasikan tingkah laku suatu entitas atau sistem. Sebuah *state diagram* menjelaskan bagaimana suatu objek atau sistem berperilaku atau bereaksi terhadap peristiwa tertentu di sekitarnya. Dalam *state diagram*, suatu objek atau sistem dinyatakan sebagai keadaan yang dapat berubah atau berubah menjadi keadaan lain.

Tabel 5. Simbol *State Diagram*

Simbol	Nama	Keterangan
	<i>State</i>	Keadaan atau kondisi suatu objek atau sistem pada waktu tertentu. Setiap <i>state</i> dapat dijelaskan dengan deskripsi atau label yang menjelaskan keadaan atau sifat dari <i>state</i> tersebut.
	<i>Start State</i>	Lingkaran yang mewakili mulainya sebuah objek
	<i>End State</i>	Lingkaran yang mewakili berakhirnya sebuah objek
	<i>Transition</i>	Panah yang mewakili jalur antara status objek yang berbeda. Sebuah <i>state</i> dapat memiliki transisi yang mengarah kembali ke dirinya sendiri.

### Database

Basis data atau *database* adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat dikontrol oleh program komputer untuk mengambil informasi dari *database* tersebut. Sistem basis data dapat dipelajari dalam ilmu komputer. Konsep dasar *database* adalah kumpulan *record* atau informasi. *Database* memiliki deskripsi terstruktur dari tipe fakta yang disimpan di dalamnya, deskripsi ini disebut skema. Skema menggambarkan objek yang direpresentasikan dalam *database* dan hubungan antara objek tersebut [13].

*Database* secara umum adalah suatu susunan atau kumpulan *record* yang disimpan dalam komputer yang saling berhubungan yang dapat digunakan sebagai sumber sistem informasi yang berkesinambungan untuk memenuhi kebutuhan informasi yang optimal bagi pengguna. Rekaman basis data masih sering ditampilkan dalam bentuk teks biasa untuk informasi pengguna guna memungkinkan akses yang lebih mudah oleh kriptanalis dan

memberikan peluang untuk membocorkan, berbagi, atau memodifikasi rekaman basis data ini[14].

### 3. METODE PENELITIAN

#### Objek Penelitian

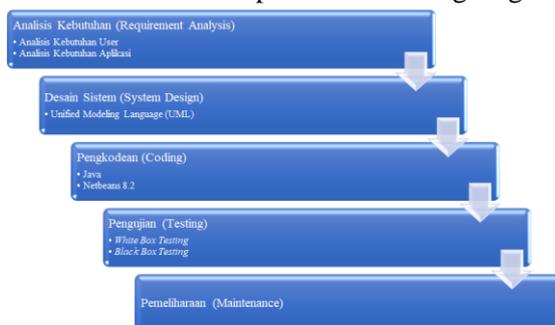
Objek pada penelitian ini adalah perancangan aplikasi profit penjualan motor untuk CV. Tridjaya Anugerah Sukses. Penelitian ini dilakukan untuk memudahkan admin yang mengelola penjualan motor dan pembelian motor dalam melakukan pekerjaannya. Serta dapat membuat laporan yang sesuai dengan pekerjaan yang sudah dilakukannya dengan menggunakan aplikasi profit ini.

#### Metodologi Penelitian

Metode yang digunakan dalam penelitian ini adalah rekayasa perangkat lunak dengan menggunakan model *waterfall*. Model *waterfall* adalah suatu pendekatan yang mengembangkan perangkat lunak secara berurutan, dimana setiap tahapan harus diselesaikan sebelum dapat melanjutkan ke tahapan berikutnya. Untuk tahapan yang ada pada model *waterfall* adalah melakukan analisis kebutuhan dan dilanjut dengan tahap desain sistem, pengkodean, pengujian, dan terakhir pemeliharaan.

#### Rancangan Penelitian

Berikut adalah tahapan yang akan dilaksanakan selama penelitian berlangsung:



Gambar 2. Alur Penelitian

#### Requirement Analysis/Analisis Kebutuhan

Pada tahap ini dilakukan sebuah analisis untuk mengetahui permasalahan yang dialami oleh CV. Tridjaya Anugerah Sukses sehingga diperoleh sebuah solusi yang tepat sasaran. Tahapan yang ada pada analisis kebutuhan diantaranya analisis kebutuhan *user*, analisis kebutuhan aplikasi, dan analisis sistem.

#### a. Analisis Kebutuhan *User*

Pada analisis ini, dilakukan interaksi langsung dengan administrator yang bertanggung jawab atas aspek keuangan perusahaan seperti transaksi pembelian dan penjualan yang dilakukan..

#### b. Analisis Kebutuhan Aplikasi

Pada analisis ini, dilakukan evaluasi terhadap para pengguna dalam penelitian. Hal yang dievaluasi meliputi kebutuhan fungsional dan kebutuhan non fungsional.

#### System Design/Desain Sistem

Pada tahapan ini, kebutuhan yang sudah didapat dari hasil analisis direpresentasikan ke dalam bentuk desain agar lebih mudah dalam tahapan selanjutnya. Perancangan desain menggunakan dibantu dengan UML dengan berbagai diagram seperti *use case diagram*, *class diagram*, *activity diagram*, *sequence diagram*, dan *state diagram*.

#### Coding/Pengkodean

Pada tahapan ini akan melakukan pengkodean sistem yang sudah dianalisis dan dibuat desainnya. Implementasi yang dilakukan pada tahapan ini menggunakan bahasa pemrograman Java dengan bantuan *software* Netbeans 8.2.

#### Testing/Pengujian

Pada langkah ini, dilakukan upaya untuk menemukan kesalahan atau kekurangan yang mungkin ada pada sistem yang telah dibuat. Pengujian yang dilaksanakan pada program aplikasi, yaitu:

#### a. White Box Testing

Pengujian *White Box* adalah pengujian yang dilakukan dengan memeriksa secara detail bagaimana sistem berfungsi di dalamnya. Tujuan dari pengujian ini yaitu untuk memastikan kode program sudah berfungsi seutuhnya tanpa ada kendala yang tidak diinginkan.

#### b. Black Box Testing

Pengujian *Black Box* adalah pengujian yang dilakukan tanpa melihat struktur internal kode program, pengujian ini dilakukan dengan sudut pandang pengguna yang berhadapan langsung dengan *interface* aplikasi. Tujuan dari pengujian ini adalah memastikan tidak terdapat kendala pada saat aplikasi sudah digunakan oleh penggunanya.

### Maintenance/Pemeliharaan

Pada tahapan terakhir ini adalah sebuah pemeliharaan sistem yang sudah dipakai oleh pengguna agar sistem dapat dilakukan perbaikan apabila sudah digunakan lama dan mendapatkan kendala. Selain perbaikan, penambahan fungsi lain bisa dilakukan dalam aplikasi maka akan diperbaharui atau dilakukan analisis ulang.

## 4. HASIL DAN PEMBAHASAN

### Hasil

Dalam penelitian yang dilakukan terhadap sistem profit penjualan di admin CV. Tridjaya Anugerah Sukses, ditemukan hasil berdasarkan pengamatan dan intervisi. Dalam penelitian ini, dijelaskan hasil dari penelitian tentang sistem profit penjualan yang sedang berlangsung serta sistem keuntungan penjualan yang diajukan. Dalam merencanakan dan membangun aplikasi ini, pengembangan akan menggunakan bahasa pemrograman Java dan mengikuti metode *System Development Life Cycle* (SDLC) dengan model *Waterfall*. Proses pembangunan aplikasi akan dilakukan dengan tahapan-tahapan berikut:

### Analisis Kebutuhan

Hasil analisa dari metode pengumpulan data yang sudah dilakukan adalah sebagai berikut:

#### 1. Analisis kebutuhan *user*

Terdapat seorang *user* yang berperan dalam aplikasi yang dibuat yaitu admin. Peran admin disini sebagai pengelola profit penjualan dari data penjualan yang sudah dilakukan. Data penjualan tersebut kemudian disatukan dengan data kasir untuk menghitung profit yang dihasilkan dealer tersebut.

#### 2. Analisis kebutuhan aplikasi

Dalam penelitian ini, dilakukan analisis untuk memahami kebutuhan yang dihadapi oleh pengguna aplikasi profit penjualan.

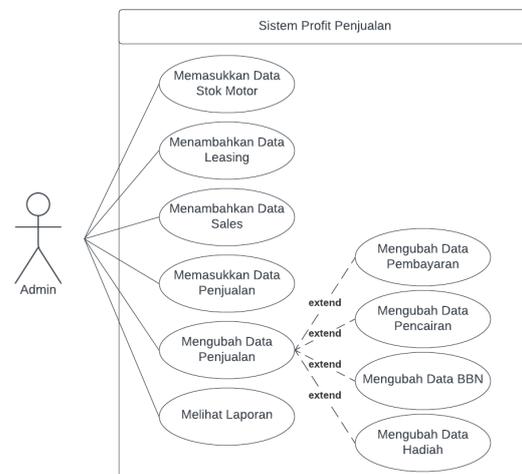
Melakukan analisis kebutuhan fungsional bertujuan untuk memperoleh pemahaman yang lebih jelas mengenai jenis kebutuhan yang diperlukan oleh pengguna dalam aplikasi profit penjualan. Dari hasil wawancara yang dilakukan sesuai dengan analisis kebutuhan pengguna, ditemukan beberapa kebutuhan bagi pengguna pada aplikasi profit penjualan sebagai berikut:

- Mempermudah dalam perhitungan profit penjualan menggunakan aplikasi berbasis desktop.
- Mempermudah dalam membuat laporan penjualan yang dapat dikelompokkan berdasarkan *leasing*, *bbn*, dan *pencairan*.

### Desain Sistem

Dalam merancang desain sistem ini menggunakan pemodelan UML diantaranya adalah: *use case diagram*, *activity diagram*, *sequence diagram*, *state diagram*. Pada desain database menggunakan *class diagram*.

#### Use Case Diagram

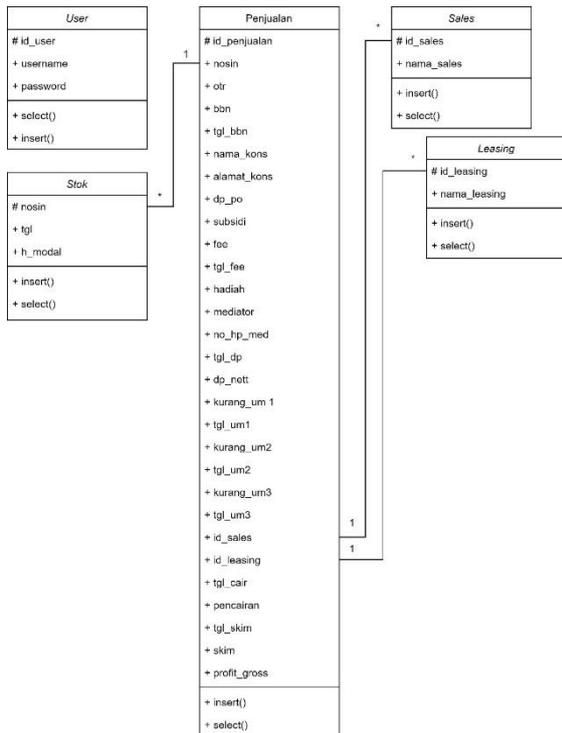


Gambar 3. Use case diagram

*Use case diagram* yang terdapat pada Gambar 3 adalah skema umum yang digunakan dalam merancang sistem yang akan dikembangkan dalam penelitian ini.

#### Class Diagram

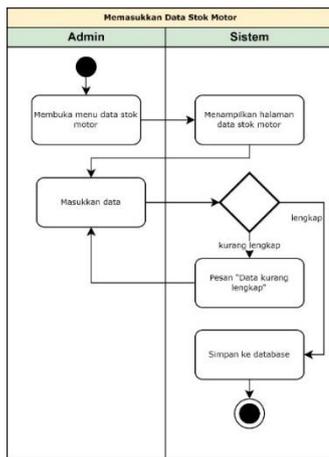
*Class diagram* pada Gambar 4 merupakan susunan dari database sistem yang akan dibuat. Dimulai dari *database user* yang diperlukan untuk masuk ke dalam sistem. Dan *database stok*, *sales*, dan *leasing* untuk mendukung penginputan yang akan dimasukkan ke dalam *database penjualan*.



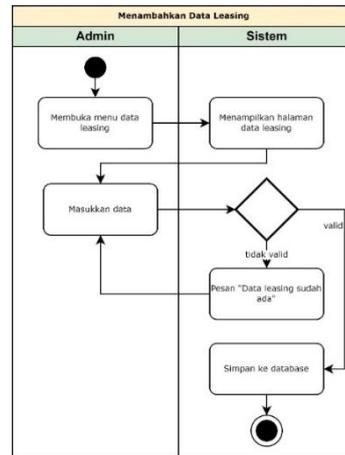
Gambar 4. Class Diagram

**Activity Diagram**

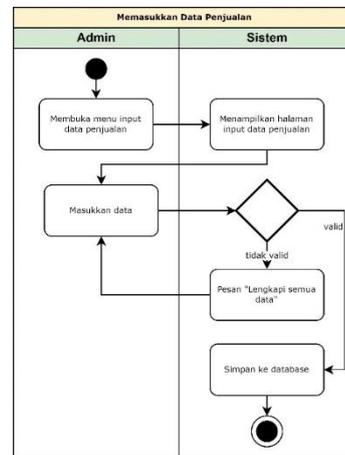
Untuk mempelajari alur interaksi pengguna dengan sistem dibutuhkan diagram yang bisa menggambarkan alur interaksi dari case yang ada pada Use case diagram. Activity diagram dibawah ini akan menggambarkan alur interaksi antara user dengan sistem sebagai berikut.



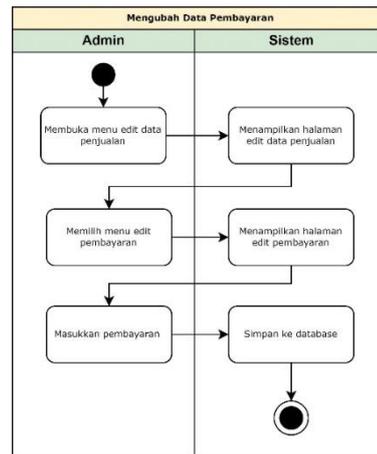
Gambar 5. Activity Diagram Menambahkan Data Leasing



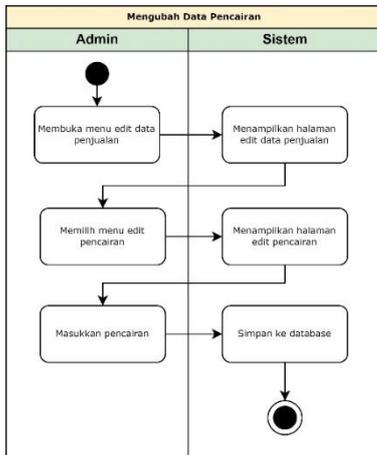
Gambar 6. Activity Diagram Menambahkan Data Sales



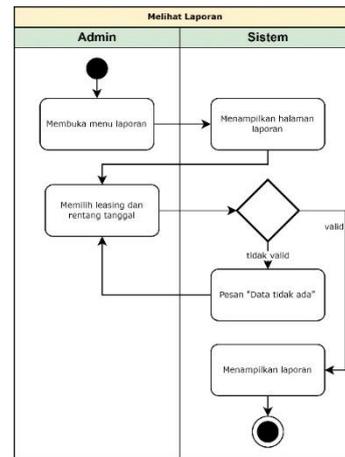
Gambar 7. Activity Diagram Memasukkan Data Penjualan



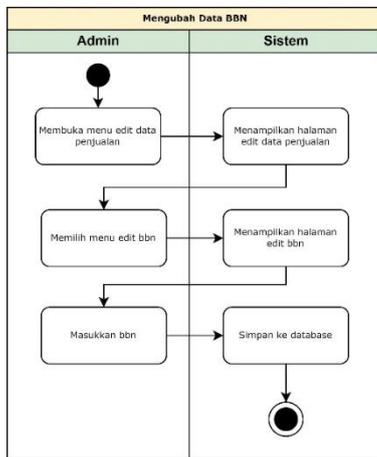
Gambar 8. Activity Diagram Merubah Data Pembayaran



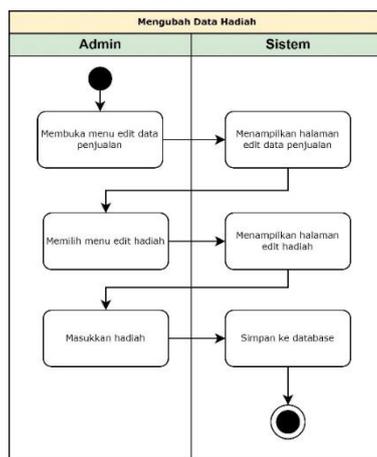
Gambar 9. Activity Diagram Merubah Data Pencairan



Gambar 12. Activity Diagram Melihat Laporan



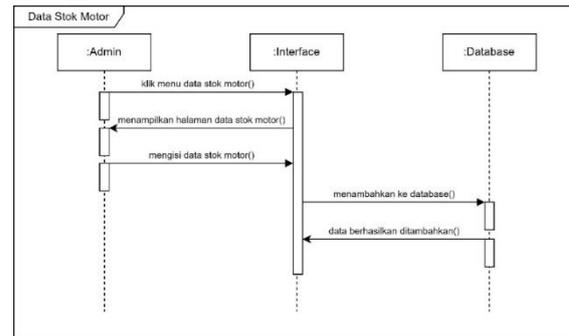
Gambar 10. Activity Diagram Merubah Data BBN



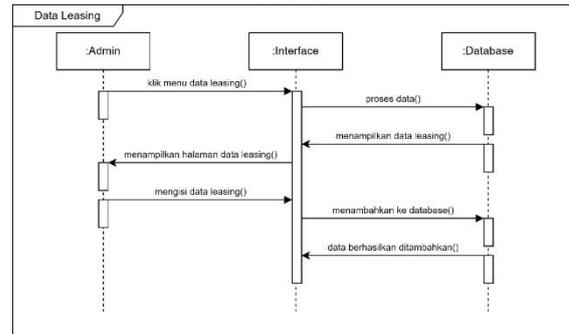
Gambar 11. Activity Diagram Merubah Data Hadiah

Sequence Diagram

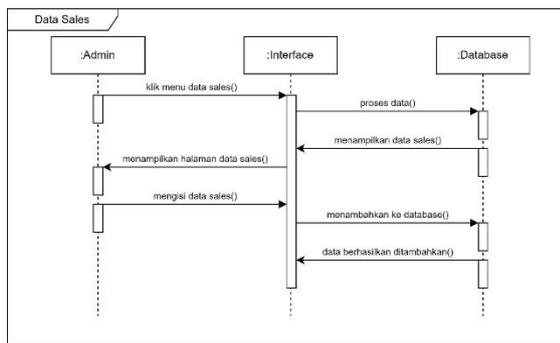
Sequence Diagram merupakan sebuah gambaran suatu integrasi antar objek disekitar sistem.



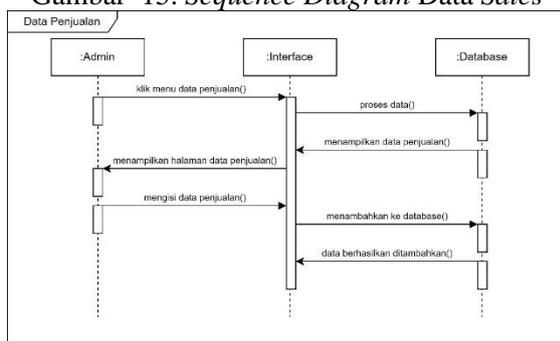
Gambar 13. Sequence Diagram Data Stok Motor



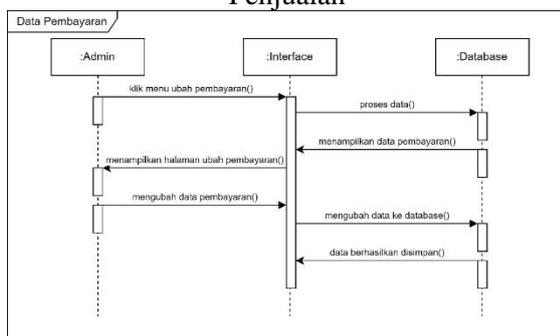
Gambar 14. Sequence Diagram Data Leasing



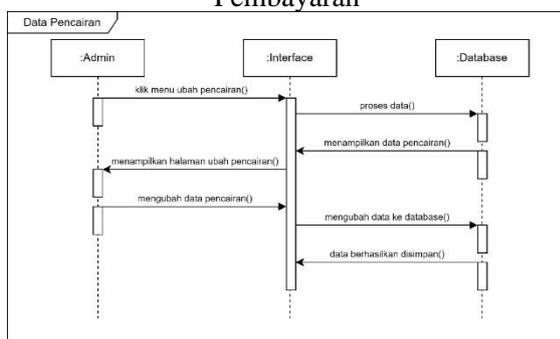
Gambar 15. Sequence Diagram Data Sales



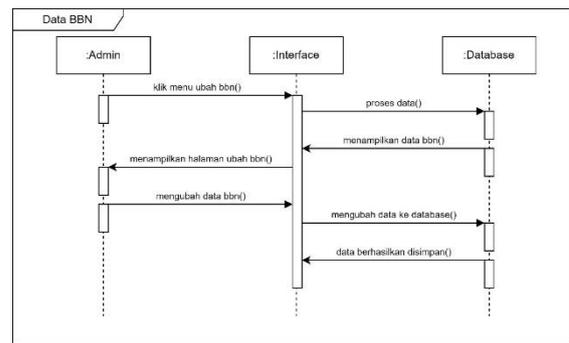
Gambar 16. Sequence Diagram Data Penjualan



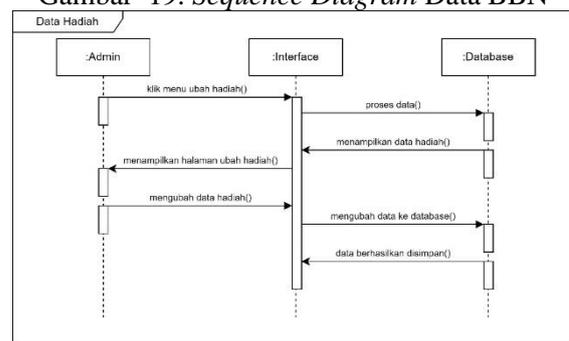
Gambar 17. Sequence Diagram Data Pembayaran



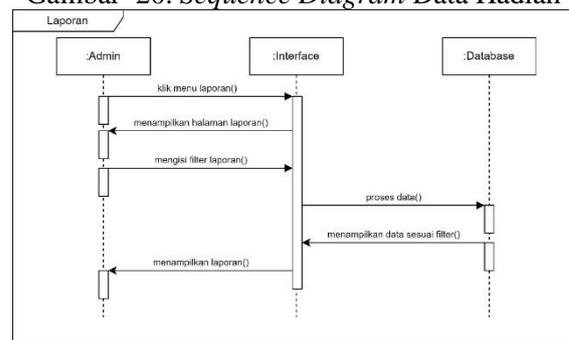
Gambar 18. Sequence Diagram Data Pencairan



Gambar 19. Sequence Diagram Data BBN



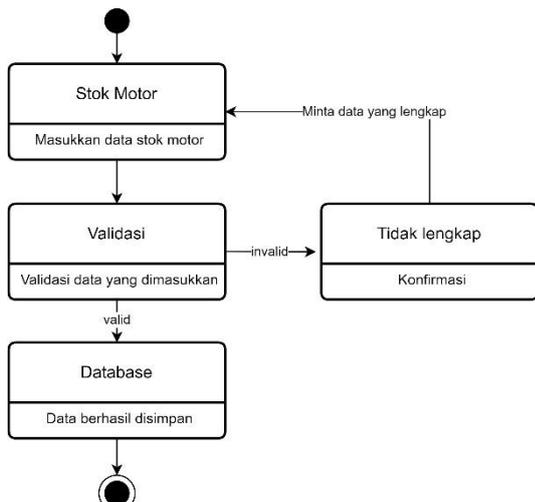
Gambar 20. Sequence Diagram Data Hadiah



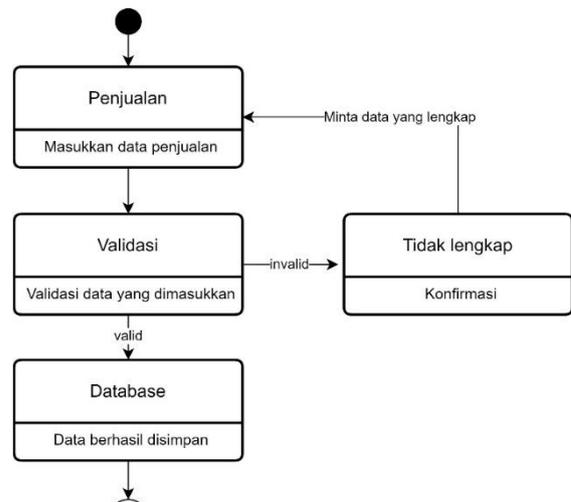
Gambar 21. Sequence Diagram Laporan

**State Diagram**

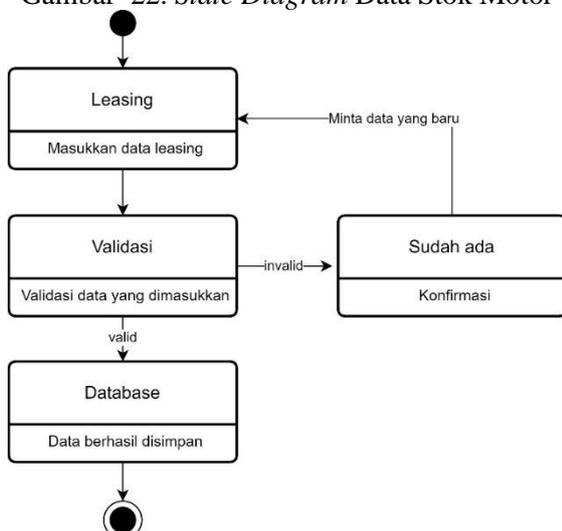
State diagram merupakan perilaku sebuah sistem pada saat digunakan oleh penggunanya. Dengan penggambaran perilaku sistem dengan diagram ini dimaksudkan untuk mempermudah dalam memahami apa yang terjadi pada saat sistem digunakan.



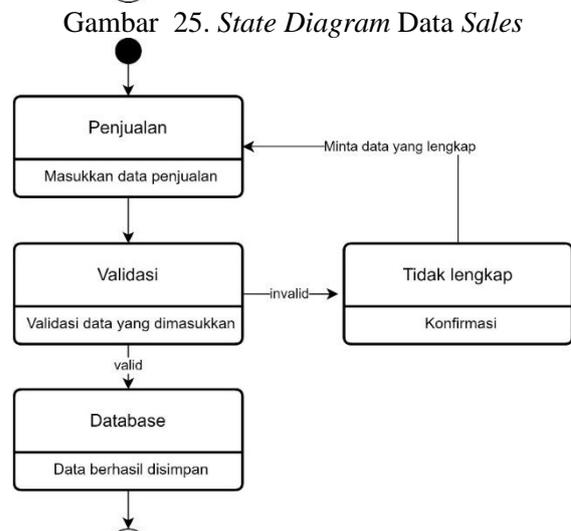
Gambar 22. State Diagram Data Stok Motor



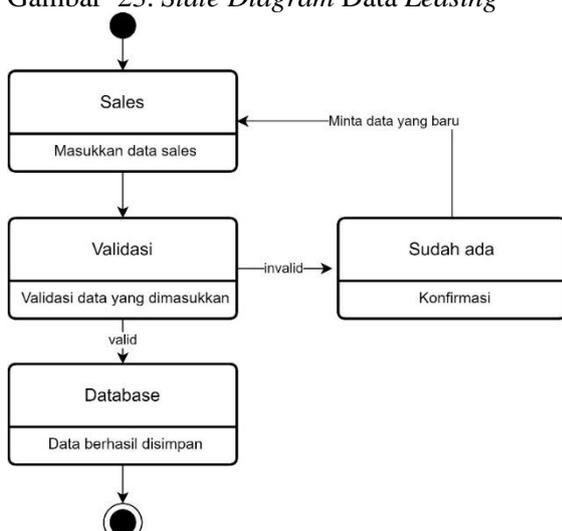
Gambar 25. State Diagram Data Sales



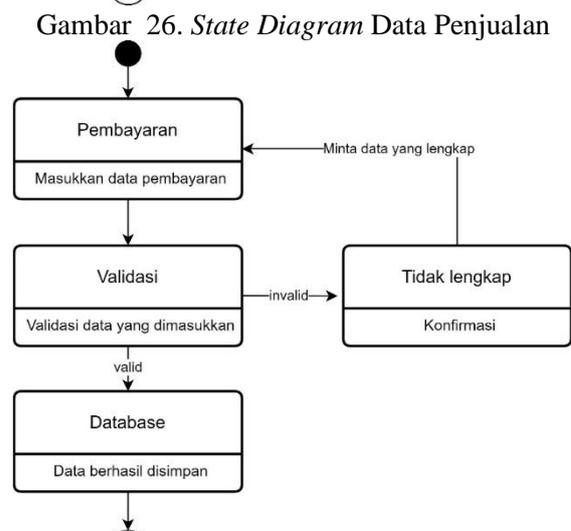
Gambar 23. State Diagram Data Leasing



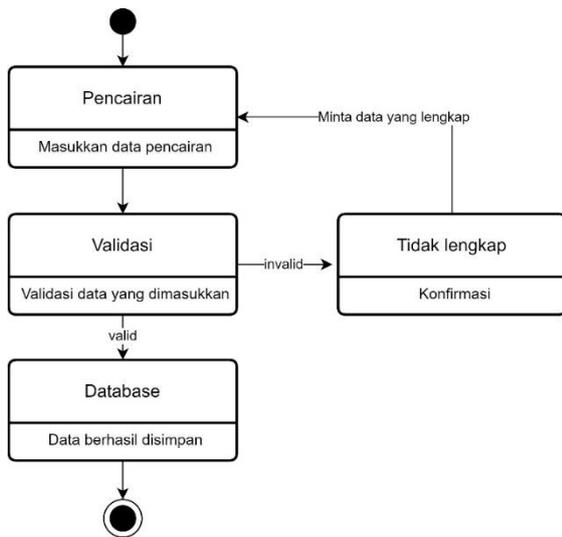
Gambar 26. State Diagram Data Penjualan



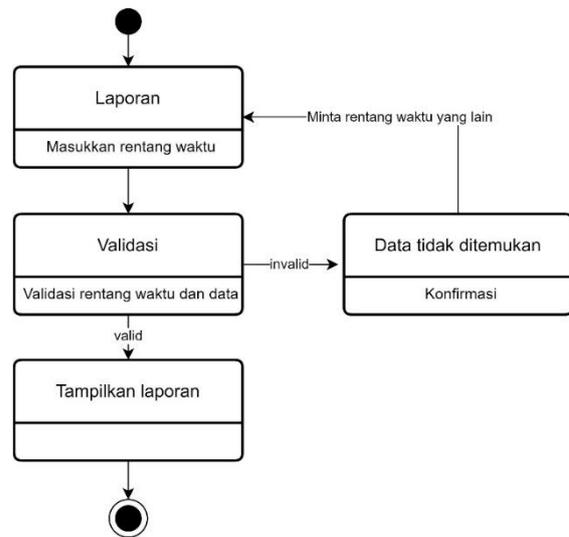
Gambar 24. State Diagram Data Sales



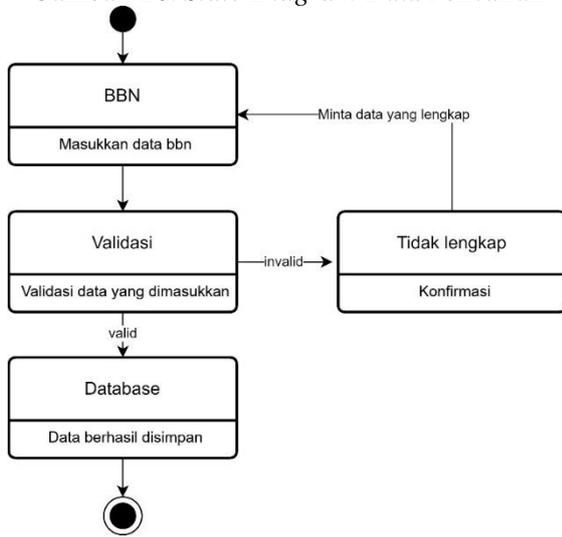
Gambar 27. State Diagram Data Pembayaran



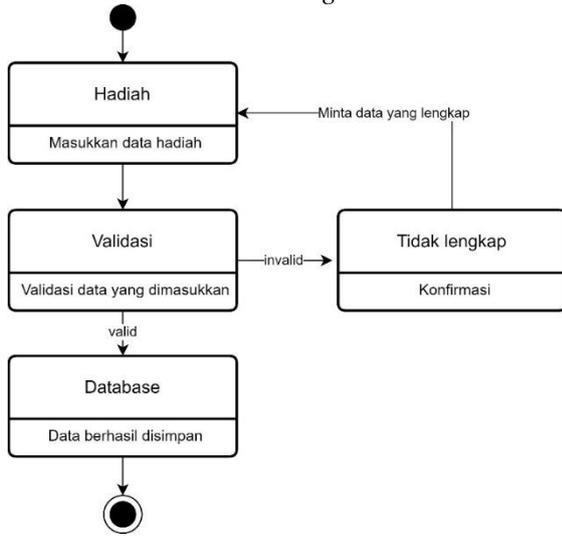
Gambar 28. State Diagram Data Pencairan



Gambar 31. State Diagram Melihat Laporan



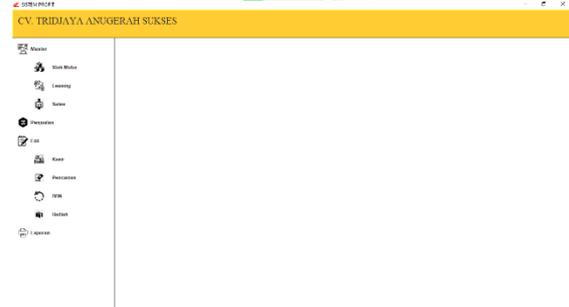
Gambar 29. State Diagram Data BBN



Gambar 30. State Diagram Data Hadiah

**Pengkodean**

Pada fase pengkodean, penulis menciptakan sebuah sistem menggunakan bahasa pemrograman Java melalui aplikasi Netbeans IDE 8.2. Dengan demikian, sistem yang dibangun dapat berjalan dalam format aplikasi desktop. Serta menggunakan *database MySQL Workbench 8.0* untuk menyimpan data yang diproses oleh sistem yang dibuat.



Gambar 32. Halaman Dashboard



Gambar 33. Halaman Data Penjualan

**Pengujian**

Pengujian aplikasi ini dilakukan bertujuan untuk mengecek kembali masing-masing fitur dan fungsi yang terdapat pada aplikasi serta mengetahui jika terdapat

kesalahan. Pengujian atau testing aplikasi yang telah dibuat menggunakan pengujian *Black Box Testing* dan *White Box Testing*.

a. *Black Box Testing*

*Black Box Testing* adalah metode pengujian perangkat lunak yang bertujuan untuk memverifikasi bahwa aplikasi beroperasi sebagaimana yang diinginkan. Berikut hasil *Black Box Testing* pada aplikasi sistem profit telah diperoleh.

Tabel 6 Hasil Uji Halaman Data Stok

Input	Plannin g	Response System	Result
Hasil Uji Normal			
Menambahkan data stok motor yang belum ada di sistem	Data berhasil tersimpan	Data berhasil tersimpan	[✓] Terima [ ] Tolak
Hasil Uji Tidak Normal			
Menambahkan data stok motor yang sudah ada di sistem	Data tidak tersimpan dan muncul peringatan bahwa stok sudah pernah dimasukkan	Data tidak tersimpan dan muncul peringatan bahwa stok sudah pernah dimasukkan	[✓] Terima [ ] Tolak

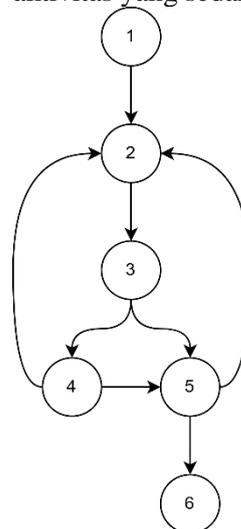
Tabel 7 Hasil Uji Halaman Data Penjualan

Input	Plannin g	Response System	Result
Hasil Uji Normal			
Menginput data dengan nomor mesin yang belum terjual	Data berhasil tersimpan	Data berhasil tersimpan	[✓] Terima [ ] Tolak

Input	Plannin g	Response System	Result
Hasil Uji Tidak Normal			
Menginput data dengan nomor mesin yang sudah terjual	Muncul pesan bahwa stok atau unit sudah terjual	Muncul pesan bahwa stok atau unit sudah terjual	[✓] Terima [ ] Tolak

b. *White Box Testing*

*White Box Testing* adalah suatu metode pengujian perangkat lunak yang bertujuan untuk memverifikasi bahwa semua jalur atau skenario dalam kode program telah berhasil tanpa kegagalan. Pengujian dilaksanakan dengan menggunakan metode *basis path testing* dan perhitungan untuk menentukan tingkat *error* suatu skenario berdasarkan jumlah *cyclomatic complexity* yang dimilikinya. Dengan demikian, skenario yang akan dilakukan dapat terlihat melalui jalur yang akan dibuat dari beragam aktivitas yang sedang berlangsung.



Gambar 34. *Flowgraph* Halaman Data Stok Motor

Pada Gambar 34 jumlah jalur yang terbentuk dapat dikalkulasikan menggunakan rumus perhitungan *cyclomatic complexity*.

$$V(G) = E - N + 2$$

$$V(G) = 8 - 6 + 2 = 4$$

Berdasarkan perhitungan, ditemukan bahwa terdapat 4 jalur yang dihasilkan dari *flowgraph*. Dari rangkaian yang telah diidentifikasi, sebuah *test case* dibuat untuk menguji apakah hasil yang diperoleh berhasil atau gagal. Daftar uji yang telah disusun ditampilkan dalam Tabel 4.21.

Tabel 8 *Test Case* Halaman Data Stok Motor

No.	Kegiatan	Hasil yang diharapkan	Hasil	Keterangan
1	Tambah data stok motor dengan data yang lengkap	Data berhasil disimpan	Data berhasil disimpan	Berhasil
2	Tambah data stok motor dengan data yang tidak lengkap	Terdapat pesan <i>error</i> dan harus mengisikannya yang kosong	Terdapat pesan <i>error</i> dan harus mengisikannya yang kosong	Berhasil
3	Langsung proses simpan dengan menamahkan data dalam tabel	Data berhasil disimpan	Data berhasil disimpan	Berhasil
4	Langsung proses simpan tanpa menamahkan data dalam tabel	Terjadi <i>error</i> karena tidak ada data yang dimasukkan	Terjadi <i>error</i> karena tidak ada data yang dimasukkan	Berhasil

Setelah melaksanakan semua skenario, hasil pengujian yang telah dilakukan pada semua skenario yang telah ditentukan dapat dilihat dalam Tabel 9 sebagai ringkasan.

Dengan menggunakan informasi mengenai *node* (N), *edge* (E), *cyclomatic complexity* (CC), berhasil (B), dan gagal (G).

Tabel 9 Ringkasan Hasil Pengujian Skenario

Skenario	Jumlah				
	N	E	CC	B	G
1	6	8	4	4	0
2	4	4	2	2	0
3	4	4	2	2	0
4	3	4	1	1	0
5	3	4	1	1	0
6	3	4	1	1	0
7	3	4	1	1	0
8	3	4	1	1	0
9	7	8	3	3	0

Dari hasil pengujian di Tabel 9, terdapat 16 jalur yang muncul dari 9 skenario yang telah ditentukan. Sehingga, terbentuklah sebuah diagram hasil yang terlihat seperti Gambar 35.



Gambar 35. Hasil Pengujian

Berdasarkan hasil pengujian pada Gambar 35 dapat dilihat bahwa tingkat keberhasilan dari pengujian *white box* mencapai 100%. Dapat dikatakan bahwa untuk sistem yang diuji tidak ada kegagalan yang terjadi sesuai dengan skenario yang digunakan.

#### Pemeliharaan

Pada tahapan pemeliharaan akan dilakukan apabila sistem sudah dijalankan dengan waktu yang cukup lama. Pemeliharaan melibatkan beberapa komponen seperti memperbaiki kesalahan dalam sistem, mengubah unit sistem yang sudah ada, dan

meningkatkan sistem sesuai dengan kebutuhan yang ada.

### Pembahasan

Penelitian yang dilakukan ini menggunakan metode *waterfall*. Aplikasi dibuat dengan konsep *model view controller* menggunakan tool Netbeans IDE 8.2 dan bahasa pemrograman Java. Untuk pengujian dilakukan dengan *black box testing* dan *white box testing* teknik basis path. Hasil pengujian menggunakan teknik *basis path* mencapai angka keberhasilan 100% dengan perhitungan *cyclomatic complexity*.

### 5. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan dalam pembangunan sistem profit penjualan motor sebagai penunjang pengelolaan data profit, maka dapat disimpulkan sebagai berikut:

1. Fitur yang disediakan dalam sistem profit sudah mencapai semua target yang diharapkan. Dengan adanya sistem profit dapat memudahkan admin dalam melakukan tugasnya dengan lebih baik lagi dan lebih mudah.
2. Perancangan sistem dengan menggunakan metode *waterfall* bisa dikata cepat selesai karena setiap tahapan harus sudah selesai terlebih dahulu apabila ingin melanjutkan ke tahapan selanjutnya. Perancangan menggunakan bahasa Java dapat diterapkan dengan baik dan menghasilkan sebuah aplikasi berbasis desktop.
3. Dengan adanya sistem profit ini berpengaruh terhadap perhitungan secara otomatis oleh sistem agar lebih akurat lagi dan tidak menghabiskan waktu yang cukup lama.

### UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini. (*The author would like to thank the related parties who have provided support for this research.*)

### DAFTAR PUSTAKA

[1] Fanny, Sufiyati, and M. F. D. Indrajati, "Faktor yang Mempengaruhi Profitability pada Perusahaan Manufaktur yang Terdaftar Di

- BEI," vol. 2, pp. 1036–1046, 2020.
- [2] D. V. Karim, "Rancang Bangun Aplikasi Manajemen Peralatan Pada Jurusan Pendidikan Teknik Elektro Berbasis Desktop," *JUKI J. Komput. dan Inform.*, vol. 2, no. 1, pp. 1–5, 2020, doi: 10.53842/juki.v2i1.22.
- [3] S. Esabella, W. Satru, and M. Haq, "Rancang Bangun Aplikasi Pengelolaan Data Penjualan Sembako Berbasis Desktop Untuk Ud. Kerta Mandala Sumbawa Besar," *J. Inform. Teknol. dan Sains*, vol. 3, no. 1, pp. 294–300, 2021, doi: 10.51401/jinteks.v3i1.982.
- [4] R. Taufiq, R. R. Ummah, I. Nasrullah, and A. A. Permana, "Rancang Bangun Sistem Informasi Penggajian Pegawai Berbasis Web di Madrasah Ibtidaiyah Nurul Huda Kota Tangerang," *J. Inform. Univ. Pamulang*, vol. 4, no. 4, p. 119, 2019, doi: 10.32493/informatika.v4i4.3951.
- [5] A. N. Nurhayati, A. Josi, and N. A. Hutagalung, "Rancang Bangun Aplikasi Penjualan dan Pembelian Barang Pada Koperasi Kartika Samara Grawira Prabumulih," *J. Teknol. dan Inf.*, vol. 7, no. 2, pp. 13–23, 2018, doi: 10.34010/jati.v7i2.490.
- [6] B. Huda and B. Priyatna, "Penggunaan Aplikasi Content Management System (CMS) Untuk Pengembangan Bisnis Berbasis E-commerce," *Systematics*, vol. 1, no. 2, p. 81, 2019, doi: 10.35706/sys.v1i2.2076.
- [7] T. H. F. Harumy, J. Sitorus, and M. Lubis, "Sistem Informasi Absensi Pada Pt . Cospar Sentosa Jaya Menggunakan Bahasa Pemrograman Java," *J. Tek. Inform.*, vol. 5, no. 1, pp. 63–70, 2018, [Online]. Available: <http://download.garuda.kemdikbud.go.id/article.php?article=1054639&val=15787&title=SIS tem Informasi Absensi Pada Pt Cospar Sentosa Jaya Menggunakan Bahasa Pemrograman Java>
- [8] M. F. Londjo, "Implementasi White Box Testing Dengan Teknik Basis Path Pada Pengujian Form Login," *J. Siliwangi*, vol. 7, no. 2, pp. 35–40, 2021, [Online]. Available: <https://jurnal.unsil.ac.id/index.php/jssainstek/article/view/4086>
- [9] A. Utomo, Y. Sutanto, E. Tiningrum, and E. M. Susilowati, "Pengujian Aplikasi Transaksi Perdagangan Menggunakan Black Box Testing Boundary Value Analysis," *J. Bisnis Terap.*, vol. 4, no. 2, pp. 133–140, 2020, doi: 10.24123/jbt.v4i2.2170.
- [10] Risald and L. S. Lafu, "Implementasi Sistem Penjualan Online Berbasis E-Commerce Pada Usaha Ukm Ike Suti Menggunakan Metode Waterfall," *J. Inf. Technol.*, vol. 1, no. 1, pp. 37–42, 2021, doi: 10.32938/jitu.v1i1.1393.
- [11] M. S. Rumetna and T. N. Lina, "Sistem

- Informasi Kampung Wisata Arborek Dengan Metode Waterfall,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 3, p. 305, 2018, [Online]. Available: <http://ejournal-binainsani.ac.id/index.php/ISBI/article/view/1393>
- [12] A. F. Prasetya, Sintia, and U. L. D. Putri, “Perancangan Aplikasi Rental Mobil Menggunakan Diagram UML ( Unified Modelling Language ),” *J. Ilm. Komput. Terap. dan Inf.*, vol. 1, no. 1, pp. 14–18, 2022, [Online]. Available: <http://journal.polita.ac.id/index.php/politati/article/view/98>
- [13] A. Andaru, “Pengertian Database Secara Umum,” *Proc. 1970 25th Annu. Conf. Comput. Cris. How Comput. are Shap. our Futur. ACM 1970*, pp. 1–7, 2018, doi: 10.1145/1147282.1147284.
- [14] N. R. Yanti, Alimah, and D. A. Ritonga, “Implementasi Algoritma Data Encryption Standard Pada Penyandian Record Database,” *J-SAKTI (Jurnal Sains Komput. dan Inform.*, vol. 2, no. 1, p. 23, 2018, doi: 10.30645/j-sakti.v2i1.53.