

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Penelitian yang dilakukan oleh Prasojo Herdy Sutanto yang berjudul **“Perancangan System Stok Barang Di Warehouse Berbasis Web”**. Didalam proses pendataan stok di gudang PT. Mulia Andalan Gemilang masih banyak kendala perhitungan stok barang di gudang. Perhitungan dan menentukan jenis produk yang telah habis sangat sulit dilakukan, hal ini di sebabkan karena penggunaan system perhitungan yang masih manual dan belum menerapkan teknologi informasi yang tepat. Dalam proses memonitor dan *controlling* stok gudang PT. Mulia Andalan Gemilang saat ini sangat membutuhkan sebuah teknologi sistem informasi yang moderen yaitu *monitoring* dan *controlling* persediaan secara terkomputerisasi, sehingga memudahkan manajer ataupun *owner* (pemilik) untuk mengontrol dan memonitor persediaan stok barang yang masih ada. Jika dikaitkan dengan kebutuhan yang lebih jauh seperti pembuatan BEP perusahaan tentu saja dapat mendukung jika sistem dengan teknologi modern diterapkan. [2]

Pada saat ini sistem informasi persediaan *stock* barang yang berjalan masih menggunakan sistem manual, hal ini akan menyulitkan ketika manajer keuangan membutuhkan laporan *stock* gudang ataupun seorang *owner* ingin melihat jumlah barang tersisa yang ada di gudang, maka dimulai dari manajer produksi yang memberikan perintah kepada *staff* gudang untuk memberikan laporan *stock* gudang dan kemudian selanjutnya manajer produksi akan meneruskan laporan *stock* gudang tersebut kepada manajer keuangan. Pekerjaan itu di lakukan secara rutin oleh pihak produksi. Karena sistem informasi *monitoring* yang di gunakan saat ini masih berjalan secara manual dan kurang efektif juga kurang efisien, serta kurangnya keakuratan informasi data yang dilaporkan oleh pihak manajer sehingga mudah terjadi kesalahan dalam pembuatan laporan *stock* gudang dan sulitnya manajer

dalam mengontrol persediaan barang jika terjadi permintaan penjualan dalam jumlah tertentu.

Penelitian yang dilakukan oleh Michael Adi Swasono dan Agung Tri Prastowo yang berjudul **“Analisa dan Perancangan Sistem Informasi Pengendalian Persediaan Barang”**. Dalam pengolahan persediaan toko Maranatha Elektrik akan menemui beberapa kendala yaitu jumlah data persediaan yang salah, tidak ada pengarsipan dokumen masuk dan keluarnya dokumen di gudang, dan catatan persediaan akhir yang salah. Penjaga harus hitung dan hitung dan laporkan keterlambatan karena gudang. Pencatatan ulang, sering terjadi kelebihan persediaan, sulit untuk menentukan harga pokok barang yang dijual dan diperbarui, persediaan tidak dikoreksi, barang yang dibeli dan digunakan, informasinya tidak terlalu relevan dan akurat, komunikasi antara pemilik toko dan petugas gudang tidak lancar. Asumsi arus biaya FIFO mendefinisikan bahwa unit pertama yang dibeli adalah unit pertama yang terjual. FIFO juga disebut metode LISH (*last-in, still-here*). Metode masuk pertama keluar pertama berbeda dengan metode masuk terakhir keluar pertama. Metode masuk terakhir keluar pertama merupakan kelebihan dari metode ini. Dalam metode ini barang pertama yang dibeli lebih dulu akan dikirim out dulu, sehingga mengurangi biaya perawatan barang. Nilai persediaan. Metode ini tidak sesuai dengan peraturan perpajakan dan tidak berlaku untuk perusahaan komoditas dan produk jadi. Metode rata-rata biasa disebut dengan metode rata-rata tertimbang. Hipotesis arus biaya rata-rata menghitung harga bahan yang terkandung dalam persediaan berdasarkan biaya rata-rata barang yang sama yang tersedia selama periode waktu tertentu. Metode average membagi antara biaya barang yang tersedia untuk dijual dengan jumlah unit yang tersedia. Sehingga persediaan akhir dan beban pokok penjualan dapat dihitung dengan harga rata-rata. Metode *average* merupakan titik tengah atau perpaduan dari metode FIFO dan LIFO. Jadi kelebihan dan kekurangan metode ini berada diantara LIFO dan FIFO. Yang menjadi maksud diantara kelebihan dan kekurangan metode ini berada diantara metode LIFO dan FIFO yaitu metode *Average* menyediakan keuntungan menentukan biaya HPP akurat dan memberikan aliran fisik persediaan

sesungguhnya, yaitu barang yang lebih lama harusnya dikeluarkan pertama/dijual terlebih dahulu. [3]

Penelitian yang dilakukan oleh Steven Sanjaya, Jasmir, Despita Meisak yang berjudul **“Perancangan Sistem Informasi Stok Barang Berbasis Web Pada PT. Jambi Agung Lestari”**. Sistem informasi stok barang berbasis *web* merupakan bagian yang sangat penting agar dapat membantu perusahaan dalam mengelola stok barang yang terdapat di dalam gudang perusahaan. Sistem informasi adalah sekumpulan prosedur organisasi yang dilaksanakan untuk mencapai suatu tujuan yaitu memberikan informasi bagi pengambil keputusan dan untuk mengendalikan organisasi. Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. Sedangkan untuk *website* sendiri adalah “*website* adalah kumpulan laman yang saling terhubung di internet. *Web* mengizinkan pemberian *highlight* (penyorotan atau penggaris bawahan) pada kata-kata atau gambar dalam sebuah dokumen untuk menghubungkan atau menunjuk ke media lain seperti dokumen, frase, *movie clip*, atau *file* suara.

PT. Jambi Agung Lestari adalah perusahaan yang bergerak di bidang pengadaan barang dan jasa. Selama ini, pencatatan stok barang masih menggunakan sistem manual oleh admin gudang dengan kertas sebagai medianya. Sistem seperti ini membuat pencarian data stok barang atau pembuatan laporan memakan waktu yang lama. Selain itu, kesalahan dapat sering terjadi dalam menangani data-data yang banyak seperti ketidakcocokan antara data barang yang tercatat dengan stok fisik yang ada. Data barang juga rentan untuk hilang, terselip, basah, ataupun robek.

Perusahaan yang bergerak dibidang pengadaan barang dan jasa membutuhkan sistem yang berfokus pada bagian *warehouse* atau gudang. Sehingga *admin* gudang akan dipermudah dalam pengelolaan persediaan barang atau stok agar dapat berjalan lebih baik dan efisien, mengurangi kesalahan saat input barang

masuk dan keluar yang dilakukan secara manual, serta menghasilkan informasi yang lebih akurat dan juga cepat dicari.

Oleh karena itu, diperlukan sebuah perancangan sistem informasi stok barang berbasis *web* yang akan dilakukan pada PT. Jambi Agung Lestari dengan menggunakan metode *waterfall* sebagai metode pengembangan sistem, bahasa pemrograman PHP dan *database* MySQL. PHP dirancang untuk dapat bekerja sama dengan *database server* dan dibuat sedemikian rupa sehingga pembuatan dokumen HTML yang dapat mengakses database menjadi begitu mudah. Dengan PHP, kita dapat merubah situs kita menjadi sebuah aplikasi berbasis *web*, tidak lagi hanya sekedar sekumpulan halaman statik, yang jarang diperbaharui. [4]

2.2. Landasan Teori

2.2.1. Sistem

Sistem merupakan kumpulan dari subsistem / bagian / komponen yang saling bekerja sama secara harmonis untuk mencapai tujuan atau sasaran tertentu. Sistem adalah suatu kumpulan komponen yang saling terhubung untuk mencapai satu tujuan tertentu. Berdasarkan definisi diatas dapat disimpulkan bahwa sistem merupakan kumpulan dari beberapa subsistem yang bekerja sama untuk mencapai tujuan. [5]

2.2.2. Informasi

Informasi adalah data yang telah terorganisasi dan memiliki kegunaan ataupun manfaat. Definisi lain informasi adalah hasil pengolahan data yang memberikan arti dan manfaat. Menurut definisi diatas dapat peneliti simpulkan bahwa informasi merupakan data yang sudah diolah dan dapat bermanfaat. [5]

2.2.3. Sistem Informasi

Sistem informasi adalah cara-cara yang diorganisasi untuk mengumpulkan memasukkan dan mengolah serta menyimpan data dan cara-cara yang diorganisasi untuk menyimpan, mengelola, mengendalikan dan melaporkan informasi sedemikian rupa sehingga sebuah organisasi dapat mencapai tujuan yang telah

ditetapkan. Definisi lain dari sistem informasi adalah kumpulan dari sub-sub sistem baik fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi yang berguna. Menurut definisi diatas peneliti menyimpulkan bahwa sistem informasi adalah kumpulan dari subsistem yang bekerja sama dalam mengolah data untuk mencapai tujuan tertentu dan menghasilkan informasi yang bermanfaat. [5]

2.2.4. Website

Website adalah kumpulan dari halaman-halaman situs, yang terkandung dalam sebuah *domain* atau *subdomain*, yang tempatnya berada di dalam *World Wide Web* (*WWW*) di dalam *internet*, *website* juga dapat diartikan sebagai sebuah halaman yang berisi data, baik data *text*, gambar, suara dan lainnya yang dapat diakses secara *online*. ada banyak model pengembangan sistem yang bisa dimanfaatkan untuk membangun *website* salah satunya model *Prototyping*. *Website* atau disingkat *web*, dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa *text*, gambar, *video*, *audio*, dan animasi lainnya yang disediakan melalui jalur koneksi *internet*. *Website* adalah apa yang anda lihat via *browser*, sedangkan yang disebut *web* sebenarnya adalah sebuah aplikasi *web*, karena melakukan *action* tertentu dan membantu anda melakukan kegiatan tertentu. [6]

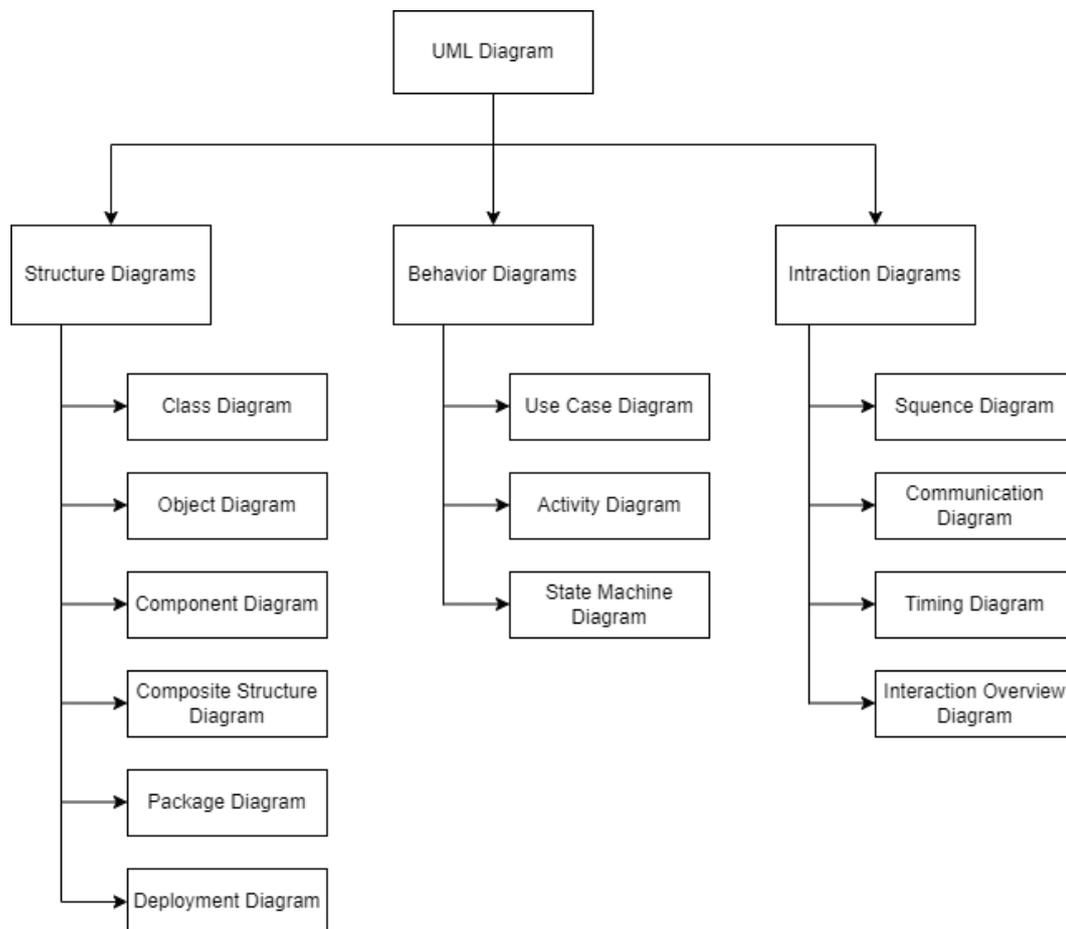
2.2.5. Inventory

Definisi persediaan adalah bahan baku yang tersedia yang akan digunakan dalam produksi, atau barang-barang yang masih dalam proses produksi, atau barang jadi siap jual yang tersedia. Menurut peneliti lain, persediaan merupakan semua jenis barang milik organisasi yang diolah, dikirim ke konsumen dan siap dijual kepada konsumen. Persediaan adalah barang-barang yang disimpan untuk digunakan atau dijual pada masa yang akan datang. Persediaan merupakan barang-barang yang tersimpan yang dapat digunakan dalam proses produksi, atau dijual atau dapat digunakan untuk tujuan tertentu sebagai operasi bisnis perusahaan. [7]

2.2.6. UML (*Unified Modeling Language*)

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks – teks pendukung. UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Diagram UML (*Unified Modeling Language*) terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam – macam diagram tersebut dapat dilihat pada gambar di bawah.

[8]



Gambar 2. 1 UML (*Unified Modeling Language*) Diagram

Berikut ini penjelasan singkat dari pembagian kategori tersebut :

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. [8]

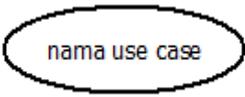
2.2.7. Use Case Diagram

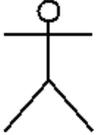
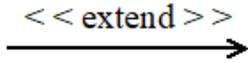
Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan di buat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem informasi yang akan di buat. Ada dua hal utama pada *use case* yaitu:

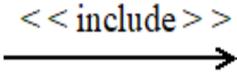
1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan di buat.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai *unit-unit* yang saling bertukar pesan antar *unit* atau aktor.

Berdasarkan definisi diatas, *Use case* diagram merupakan suatu pemodelan untuk mendeskripsikan sebuah interaksi atau kelakuan antara satu atau lebih *actor* dengan sistem informasi yang akan di buat. Berikut adalah simbol-simbol yang ada pada *use case* diagram : [8]

Tabel 2. 1 Daftar Simbol *Use Case Diagram*

Simbol	Nama	Keterangan
	<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling berkaitan bertukar pesan antar unit atau <i>actor</i> : biasanya di nyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
	<i>Actor</i>	Orang, Proses, atau sistem lain yang berinteraksi dengan sistem informasi yang

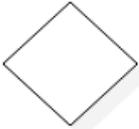
		<p>akan di buat di luar sistem informasi yang akan di buat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang: biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
	<p>Ektensi</p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.</p>
	<p>Generalisasi</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.</p>

	<i>Include</i>	Relasi <i>use case</i> tambahkan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.
---	----------------	--

2.2.8. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu di perhatikan disini bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram *activity diagram* : [8]

Tabel 2. 2 Daftar Simbol *Activity Diagram*

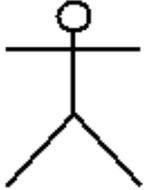
Simbol	Nama	Keterangan
	Status Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah staus awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
	Percabangan / <i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari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

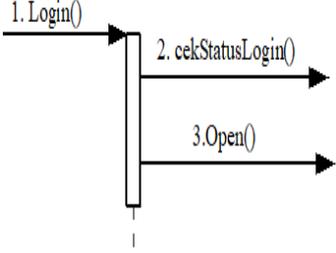
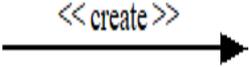
	Penggabungan / <i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas di gabungkan menjadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

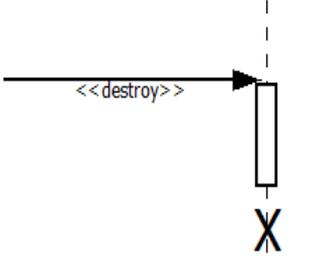
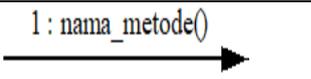
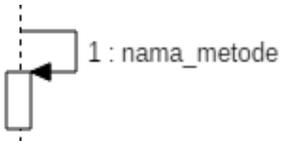
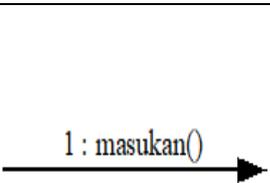
2.2.9. Sequence Diagram

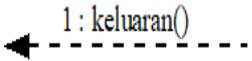
Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang di kirimkan dan diterima antar objek. Membuat diagram sekuen juga di butuhkan untuk melihat skenario yang ada pada *use case*. Berikut adalah simbol-simbol yang ada pada *sequence diagram* : [8]

Tabel 2. 3 Daftar Simbol *Sequence Diagram*

Simbol	Nama	Keterangan
 <p>Atau</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">nama_aktor</div> <p>Tanpa Waktu Aktif</p>	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan di buat diluar sistem informasi yang akan dibuat itu sendiri.Walaupun symbol dari actor adalah gambar orang,tapi actor belum tentu merupakan orang.
	Garis hidup / <i>Lifeline</i>	Menyatakan kehidupan suatu objek
<div style="border: 1px solid black; padding: 2px; display: inline-block;">Nama_objek : Nama_kelas</div>	Objek	Menyatakan objek yang berinteraksi pesan.

	<p>Waktu Aktif</p>	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.misalnya :</p>  <p>Maka cek StatusLogin() dan Open() dilakukan dalam metode Login(). Aktor tidak memiliki waktu aktif.</p>
	<p>Pesan tipe <i>Create</i></p>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang di buat.</p>

	<p>Pesan tipe <i>Destroy</i></p>	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain. Arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>.</p>
	<p>Pesan tipe <i>call</i></p>	<p>Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri.</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang di panggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
	<p>Pesan tipe <i>Send</i></p>	<p>Menyatakan bahwa suatu objek mengirimkan data/ masukan/ Informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>

	Pesan tipe <i>Return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu. Arah panah mengarah pada objek yang menerima kembalian.
---	-----------------------------	---

2.2.10. Class Diagram

Rosa A. Sukanto dan M. Shalahudin (2016), Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. [8]

Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas *main*

Kelas yang memiliki fungsi awal di eksekusi ketika sistem di jalankan.

2. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang di ambil dari pendefinisian *use case* (*controller*)

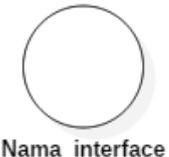
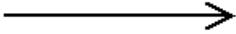
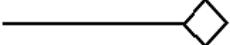
Kelas yang menangani fungsi-fungsi yang harus ada di ambil dari pendefinisian *use case*.

4. Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang di gunakan untuk memegang atau membungkus data yang menjadi sebuah kesatuan yang di ambil maupun akan disimpan ke basis data.

Berikut simbol-simbol yang ada pada diagram kelas:

Tabel 2. 4 Daftar Simbol *Class Diagram*

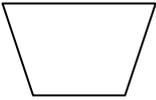
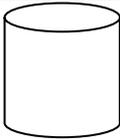
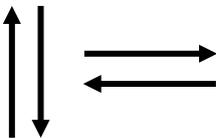
Simbol	Nama	Keterangan
	Kelas	Kelas pada struktur sistem.
	Antarmuka / <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
	Asosiasi / <i>association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Asosiasi berarah <i>/directed Association</i>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Generalisasi	Relasi antarkelas dengan makna generalisasi - spesialisasi (umum-khusus)
	Kebergantungan <i>/ dependency</i>	Relasi antarkelas dengan makna kebergantungan antar kelas.
	Agregasi/ <i>aggregation</i>	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

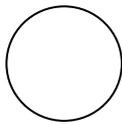
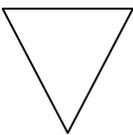
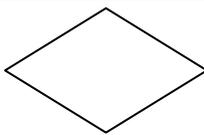
2.2.11. Flowmap

Flowmap adalah campuran peta dan *flowchart* yang menunjukkan pergerakan benda dari satu lokasi ke lokasi lain. Seperti jumlah orang migrasi, jumlah barang yang diperdagangkan atau jumlah paket dalam jaringan. *Flowmap* menolong analis dan *programmer* untuk memecahkan masalah ke dalam segmen atau bagian yang lebih kecil dan menolong dalam menganalisis alternatif – alternatif dalam pengoperasian. Fungsi *flowmap* yaitu mendefinisikan hubungan antara bagian (pelaku proses), proses (manual/ berbasis komputer) dan aliran data (dalam bentuk dokumen masukan dan keluaran). [9]

Berikut simbol dari *flowmap* : [10]

Tabel 2. 5 Daftar Simbol *Flowmap*

Simbol	Keterangan
	Simbol Dokumen Menunjukkan dokumen <i>input</i> dan <i>output</i>
	Simbol Kegiatan Manual Menunjukkan Kegiatan atau pekerjaan manual
	Simbol Proses Menunjukkan kegiatan proses dari operasi program komputer
	Simbol Keyboard Menunjukkan input yang menggunakan <i>on-line keyboard</i>
	Simbol Harddisk Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain
	Simbol Garis Alir Menunjukkan arus dari setiap proses

	<p>Simbol Penghubung</p> <p>Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain</p>
	<p>Simbol Arsip</p> <p>Menunjukkan pengarsipan <i>file</i> tanpa menggunakan komputer</p>
	<p>Simbol Keputusan</p> <p>Digunakan untuk suatu penyeleksian kondisi di dalam program</p>

2.2.12. Basis Data

Basis Data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi. Basis Data merupakan data yang terintegrasi, yang diorganisasi untuk memenuhi kebutuhan para pemakai di dalam suatu organisasi. Berdasarkan diatas, dapat disimpulkan bahwa basis data merupakan data yang dapat didesain dan berintegrasi sehingga dapat memenuhi kebutuhan user dalam perusahaan atau organisasi. [11] Basis Data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau *informasi* dan membuat informasi tersedia saat dibutuhkan”. [8] Dapat didefinisikan bahwa basis data merupakan suatu tempat penyimpanan data yang terkomputerisasi dan dapat diakses dengan mudah saat dibutuhkan.

2.2.13. PHP (*Hypertext Preprocessor*)

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah penelitian HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, *Java*, dan *Perl* serta mudah untuk dipelajari. PHP merupakan bahasa *scripting server side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya,

server yang akan menerjemahkan skrip *program*, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Adapun pengertian lain PHP adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode-kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML". [12]

2.2.14. MySQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database* nya. MySQL bersifat *free* dengan lisensi GNU *General Public License* (GPL). Dengan adanya keadaan ini maka anda dapat menggunakan *software* ini dengan bebas tanpa perlu harus takut dengan lisensi yang ada. MySQL termasuk jenis RDBMS (*Relational Database Management System*). Itulah sebabnya istilah *table*, baris, kolom digunakan pada MySQL. Pada MySQL sebuah *database* mengandung satu atau sejumlah *table*.

MySQL adalah sebuah *database management system* (DBMS) populer yang memiliki fungsi sebagai *relational database management system* (RDBMS). Selain itu MySQL *software* merupakan suatu aplikasi yang sifatnya *open source* serta *server* basis data MySQL memiliki kinerja sangat cepat, *reliable*, dan mudah untuk digunakan serta bekerja dengan arsitektur *client server* atau *embedded systems*. Dikarenakan faktor *open source* dan populer tersebut maka cocok untuk mendemonstrasikan proses replikasi basis data.[12]

2.2.15. Database Management Systems (DBMS)

DBMS adalah perangkat lunak yang menangani semua pengaksesan *database*. DBMS merupakan kumpulan *file* yang saling berkaitan bersama-sama dengan program untuk pengelolaannya.[11] Sistem manajemen *database* atau *database management system* (DBMS) adalah merupakan suatu sistem *software* yang memungkinkan seorang *user* dapat mendefinisikan, membuat, dan memelihara serta menyediakan akses terkontrol terhadap data. *Database* sendiri adalah sekumpulan data yang berhubungan dengan secara logika dan memiliki

beberapa arti yang saling berpautan. Keuntungan dari *Database Management System* adalah :

1. Pengulangan Data Berkurang. Pengulangan data atau repetisi berarti bahwa kolom data yang sama (misal : alamat seseorang) muncul berkali-kali dalam *file* yang berbeda dan terkadang dalam *format* yang berbeda. Dalam sistem pemrosesan yang lama, *file-file* yang berbeda akan mengulang data yang sama sehingga memboroskan ruang penyimpanan.
2. Integritas Data Meningkat. Integritas tidak akurat dalam DBMS, berkurangnya pengulangan berarti meningkatkan kesempatan integritas data, karena semua perubahan hanya dilakukan di satu tempat.
3. Keamanan Data Meningkat. Meskipun berbagai departemen bisa berbagi pakai data, namun akses ke informasi bisa dibatasi hanya untuk pengguna tertentu. Hanya dengan menggunakan *password* maka informasi finansial, medis, dan nilai mahasiswa dalam *database* sebuah universitas tersedia hanya bagi mereka yang memiliki hak untuk mengetahuinya.
4. Kemudahan Pemeliharaan Data. DBMS menawarkan prosedur standar untuk menambahkan, mengedit dan menghapus rekaman, juga untuk memvalidasi pemeriksaan untuk memastikan bahwa data yang tepat sudah dimasukkan dengan benar dan lengkap ke dalam masing - masing jenis kolom.[12]

2.2.16. Framework Laravel

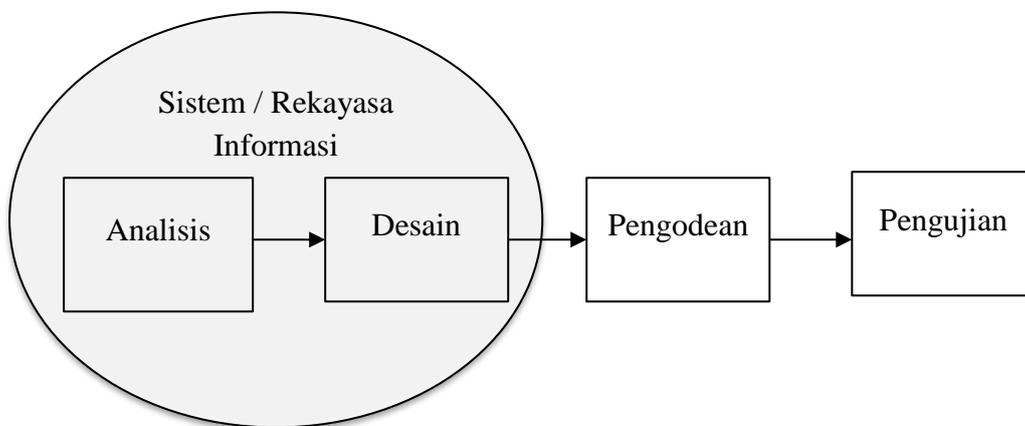
Laravel adalah sebuah *framework* untuk membuat aplikasi berbasis *web* dengan bahasa pemrograman PHP. *Laravel* merupakan salah satu *framework* PHP yang paling diminati dalam pengembangan aplikasi *back end* dengan menggunakan bahasa pemrograman PHP. *Laravel* memiliki banyak fitur yang sangat mendukung dalam mempercepat pembuatan aplikasi atau sistem informasi berbasis *web*, dimana *laravel* dapat digunakan baik sebagai *back end* dan *front end* atau hanya digunakan sebagai *back end* saja".[13]

2.2.17. Waterfall

Metode pengembangan *software* adalah suatu kerangka kerja yang digunakan untuk menstrukturkan, merencanakan, dan mengendalikan proses pengembangan suatu sistem informasi. [14]

Dalam penelitian ini penulis menggunakan metode *waterfall*, metode ini sesuai dengan kebutuhan penelitian penulis. Metode *waterfall* merupakan model yang awal digunakan dan sangat umum pada proses pembuatan project pada instansi ataupun industri yang besar. [15]

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahanan pendukung (*support*). Berikut adalah gambar model air terjun : [8]



Gambar 3. 1 Ilustrasi Model / Metode *Waterfall*

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mempesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak representasi antarmuka, dan prosedur pengodean.

3. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*Support*) atau Pemeliharaan (*Maintenance*)

Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk perangkat lunak baru. [8]

2.2.18. Black Box Testing

Pengujian sistem merupakan langkah terpenting dalam pembuatan sistem. Pengujian sistem dilakukan karena setiap orang pasti bisa membuat kesalahan apa lagi saat membuat sistem yang rumit. Langkah ini dilakukan untuk menguji apakah sistem yang telah kita buat sudah layak untuk digunakan dan sesuai dengan kebutuhan, sehingga sangat perlu dilakukan pengujian untuk mengurangi kesalahan pada sistem yang dibangun yang dapat menyebabkan kerugian.

Black box testing merupakan teknik pengujian *software* yang fokus pada spesifikasi fungsi-fungsi yang ada pada perangkat lunak yang dikembangkan. *Black box testing* cenderung dapat menemukan beberapa hal seperti fungsional yang tidak benar atau tidak ada, kesalahan struktur data, kesalahan akses basis data, kesalahan

antar muka, kesalahan *performance* serta kesalahan inisialisasi dan terminasi. Proses *Black Box Testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap *form* nya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan oleh perusahaan. [16]

Berikut beberapa tujuan *black box testing* adalah :

1. Fungsi yang salah atau hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.
6. Validasi fungsional.
7. Kesensitifan *system* terhadap nilai *input* tertentu.
8. Batasan suatu data.

Kesalahan ditemukan pada 5 poin pertama diatas. Tujuan utama pengujian adalah untuk mencari kesalahan dalam perangkat lunak untuk menghindari suatu kegagalan, atau jika sudah terlanjur gagal maka dapat diperbaiki. Karena itu pengujian *black box* memiliki langkah pertama untuk memecahkan suatu masalah. [17]

2.2.19. User Acceptance Testing (UAT)

UAT (*User Acceptance Test*) adalah suatu proses pengujian yang dilakukan oleh pengguna dengan hasil *output* sebuah dokumen hasil uji yang dapat dijadikan bukti bahwa *software* sudah diterima dan sudah memenuhi kebutuhan yang diminta. UAT tidak jauh beda dengan kusioner pada tahap awal pembuatan aplikasi. [18]

Selain itu , *User Acceptance Testing* atau UAT adalah tahapan dalam siklus pengujian software untuk memastikan produk yang dikembangkan sesuai dengan kebutuhan dan harapan *end user*. *End user* adalah individu atau kelompok yang akan secara langsung menggunakan perangkat lunak dalam kegiatan sehari-harinya.

Mereka memiliki pemahaman mendalam tentang kebutuhan dan ekspektasi terhadap *software*, sehingga menjadi pihak yang paling efektif untuk menguji apakah perangkat lunak memenuhi kebutuhan dan harapan tersebut. Tujuan utama UAT adalah untuk memastikan bahwa sistem (yang telah melewati berbagai tahap pengujian) dapat beroperasi dengan efektif di dunia nyata dan sesuai dengan kriteria bisnis. UAT berperan penting dalam siklus pengembangan perangkat lunak karena alasan-alasan berikut : [19]

1. Validasi kebutuhan *user*: UAT memastikan *software* yang dikembangkan sesuai dengan kebutuhan dan ekspektasi end user. Hal ini membantu dalam memvalidasi bahwa *software* dapat memenuhi tujuan bisnis yang diharapkan.
2. Mengidentifikasi *bug*: Melalui UAT, *bug* dan masalah yang mungkin belum terdeteksi selama tahapan pengujian sebelumnya dapat diidentifikasi dan diperbaiki sebelum peluncuran produk.
3. Mengurangi risiko dan biaya: dengan mengidentifikasi dan memperbaiki masalah sebelum peluncuran, UAT dapat mengurangi risiko kegagalan *software* di pasar dan biaya tambahan yang mungkin diperlukan untuk perbaikan setelah peluncuran.
4. Meningkatkan kepuasan *user*: UAT memungkinkan *end user* terlibat langsung dalam pengembangan *software*, sehingga memastikan produk akhir sesuai dengan kebutuhan dan harapan mereka yang berdampak pada kepuasan pengguna.
5. Melindungi reputasi: peluncuran *software* berkualitas tinggi dan bebas dari *bug* dapat membantu dalam mempertahankan dan melindungi reputasi *developer* atau bisnis. [19]

Organisasi Internasional untuk Standardisasi (ISO) 9126 mengusulkan beberapa karakteristik uji kualitas *software*. Pengujian ISO 9126 merupakan pengujian untuk mengetahui kualitas perangkat lunak yang diuji coba dari segi fungsionalitas, reliabilitas, kegunaan, efisiensi, pemeliharaan, dan portabilitas[10]. Rumus pengukuran pada ISO 9126 adalah :

$$P (\text{Presentase}) = \frac{\text{Skor Aktual } (f)}{\text{Skor Ideal } (n)} \times 100\%$$

Keterangan :

1. Skor aktual merupakan pilihan dari semua responden dari kuesioner yang telah diberikan.
2. Skor ideal diasumsikan bahwa semua responden memilih skor tertinggi dari semua jawaban.

Rensis Likert telah mengembangkan skala pengukuran yang diberi nama Skala Likert dan telah dipublikasikan tahun 1932. Skala ini menggunakan ukuran ordinal sehingga dapat membuat *ranking* dari responden. Tanggapan untuk setiap *item* instrumen pada skala Likert memiliki kriteria penilaian dari positif sampai dengan *negative*, kata yang digunakan misalnya: tidak baik, baik, cukup, kurang baik dan tidak baik. Tabel skala *likert* ditunjukkan pada tabel dibawah ini [20] :

Tabel 2. 6 Skala *Likert*

Jumlah Presentase / Skor (%)	Kriteria
20,00 % - 36,00 %	Tidak Baik
36,01 % - 52,00 %	Kurang Baik
52,01 % - 68,00 %	Cukup
68,01 % - 84,00 %	Baik
84,01 % - 100 %	Sangat Baik

Tabel 2. 7 Bobot Jawaban

Jawaban	Bobot
A. Sangat : Mudah/Bagus/Sesuai/Jelas	5
B. Mudah/Bagus/Sesuai/Jelas	4
C. Netral	3
D. Cukup : Sulit/Bagus/Sesuai/Jelas	2
E. Sangat : Sulit/Jelek/Tidak Sesuai/Tidak Jelas	1