

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Mohammad Subekti (2011) dengan judul Analisis Dan Perancangan Sistem Informasi Eksekutif Pemasaran Pada Distributor Alat Tulis Kantor: Studi Kasus Pada Benza Prima. Sistem informasi eksekutif (SIE) telah banyak digunakan dalam perusahaan dan institusi untuk membantu para eksekutif mengambil keputusan secara cepat dan akurat bagi kebutuhan bisnis organisasi. Untuk membuat sebuah sistem informasi eksekutif pemasaran bagi eksekutif Benza Prima sebuah penelitian dilakukan. Sistem informasi eksekutif pemasaran dirancang dapat menghasilkan laporan yang bersifat analitik dan laporan perbandingan dalam bentuk pivot tabel dan grafik untuk analisis pemasaran. Fact finding techniques digunakan untuk mencari informasi dengan melakukan survei langsung ke perusahaan untuk mempelajari kebutuhan perusahaan dan prosedur berjalan. Perancangan sistem informasi eksekutif menggunakan metode *Object Oriented Analysis and Design* (OOAD). Penelitian ini menghasilkan suatu laporan bagi eksekutif Benza Prima dalam bentuk pivot tabel dan grafik yang dapat membantu eksekutif melakukan pengambilan keputusan. Sistem informasi eksekutif pemasaran yang dirancang diharapkan dapat memberikan informasi analitik kepada eksekutif Benza Prima dalam melakukan perencanaan strategi pemasaran.

Berdasarkan penelitian yang dilakukan oleh M. Hadi Prayitno (2017) dengan judul Sistem Informasi Eksekutif Pemasaran Dengan Metode *Drill Down*. Hal ini dapat dilakukan dengan menciptakan sebuah system informasi eksekutif, termasuk dalam hal pemasaran. Sistem informasi eksekutif pemasaran yang dibangun dengan menggunakan metode *Drill Down* yang berfungsi untuk menampilkan informasi secara detail. Sistem akan menampilkan informasi berupa grafik dan table serta informasi yang bias dilihat secara terperinci. Sistem akan menjadi lebih efektif dan efisien jika menggunakan program berbasis web sebagai alat bantu. Data pemasaran disimpan ke dalam database guna mempermudah user dalam pencarian data. Sistem Informasi Eksekutif Pemasaran ini akan mempermudah seorang eksekutif dalam membuat sebuah keputusan guna kelangsungan perusahaan. Memperhatikan hal diatas, dan agar sistem informasi eksekutif pemasaran dibangun sesuai dengan kebutuhan perusahaan, agar lebih terarah, maka harus dibangun dengan kaidah *System Development Life Cycle* (SDLC) dengan harapan system informasi eksekutif pemasaran ini dapat digunakan oleh perusahaan.[1]

Berdasarkan penelitian yang dilakukan oleh Muhammad Ropianto (2018) dengan judul Perancangan Dashboard Sebagai Sistem Informasi Di Dinas Perumahan Rakyat, Pemukiman Dan Pertamanan Kota Batam. Metode pembuatan sistem yang digunakan adalah *Object Oriented Analysis and Design* (OOAD) dengan pemodelan sistem menggunakan Unified Modelling Language (UML) dan semi-formal interview sebagai data pendukung hasil penggunaan sistem dalam penyajian informasi. Hasil menunjukkan bahwa sistem ini bekerja dengan baik

dan dapat digunakan sebagai sebuah sistem informasi terpadu yang dapat menyajikan data secara visual dan menyeluruh kepada pihak eksekutif sebagai pengambil keputusan dan memudahkan pengguna dalam menginput dan mengolah data yang ada. Selain itu, perancangan sistem dashboard ini dirasa lebih ergonomis, responsive, interaktif, dan akurat pada praktek penggunaannya.[2]

Penelitian yang dilakukan oleh Hasanuddin (2016) dengan judul Sistem Informasi Keuangan Dengan Metode *Object Oriented Analysis Design*. *Object Oriented Analysis Design* (OOAD) adalah metode analisis yang memeriksa requirements dari sudut pandang kelas-kelas dan objek yang ditemui dalam ruang lingkup permasalahan yang mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem. OOAD merupakan cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas.[3]

Penelitian yang dilakukan oleh Harisuddin (2016) dengan judul Sistem Informasi Laundry Berbasis Metode OOAD. Seiring dengan majunya perkembangan teknologi, maka perusahaan laundry tersebut perlu melakukan peningkatan pelayanannya. Untuk mengelola berbagai macam aktivitas yang dilakukan oleh perusahaan laundry diperlukan suatu sistem informasi yang akurat dan keberadaan sistem informasi yang dapat diandalkan merupakan hal yang vital dalam perkembangan perusahaan tersebut. Sayangnya hampir semua perusahaan laundry belum memiliki aplikasi yang memadai untuk mempermudah pengelolaan laundrynya. Sehingga masih ada masalah dalam beberapa hal seperti kurangnya

komunikasi antara pegawai dengan pelanggan atau ketidaksesuaian antara pesanan dengan permintaan pelanggan. Terkait hal tersebut, maka kami membuat sistem aplikasi laundry berbasis website dengan menggunakan metode *Object Oriented Analysis and Design* (OOAD) untuk menangani permasalahan yang di hadapi perusahaan laundry tersebut. Serta untuk mendukung pengembangan sistem yang dibangun kami menggunakan Unified Model Language (UML) berupa *use case diagram*, *activity diagram*, dan *sequence diagram*. Sistem informasi ini diterapkan pada Bibi's Laundry untuk memberikan beberapa layanan seperti pemesanan secara *online*, komunikasi dengan staff secara *online*, dan beberapa layanan lain yang dapat memberikan kemudahan bagi pelanggan dan pegawai dari laundry tersebut.[4]

II.2. Landasan Teori

II.2.1. Sistem

Sebuah sistem terdiri dari berbagai unsur yang saling melengkapi dalam mencapai tujuan dan sasaran. Unsur-unsur yang terdapat dalam sistem itulah yang disebut dengan subsistem. Subsistem-subsistem tersebut harus saling berhubungan dan berinteraksi melalui komunikasi yang relevan sehingga sistem dapat bekerja secara efektif dan efisien (Eka Iswandy ; 2015 : 72).[5]

II.2.2 Informasi

Informasi merupakan data yang telah diorganisir sehingga memberikan arti dan nilai kepada penerimanya. Sedangkan menurut Jogiyanto, Informasi adalah

data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Dapat dikatakan bahwa data merupakan bahan mentah, sedangkan informasi adalah bahan jadi atau bahan yang telah siap digunakan, Jadi, sumber dari informasi adalah data (Fanny Andalia ; 2015 : 93).[6]

Informasi merupakan data yang telah proses sedemikian rupa sehingga meningkatkan pengetahuan orang yang menggunakan data tersebut. Berdasarkan beberapa pengertian informasi dari para ahli, maka dapat disimpulkan bahwa informasi adalah data yang telah diproses menjadi bentuk yang bernilai bagi penerimanya dan bermanfaat dalam setiap pengambilan keputusan (Rudi Hermawan ; 2016 : 32).[7]

II.2.3. Sistem Informasi Eksekutif

Sistem Informasi Eksekutif (EIS) adalah sebuah sistem yang menyediakan informasi secara cepat bagi eksekutif atas keseluruhan kinerja perusahaan serta membantu eksekutif dalam mengambil keputusan yang tepat. Sistem informasi eksekutif merupakan salah satu sistem informasi yang sangat dibutuhkan untuk manajerial perusahaan saat ini. Modul sistem informasi eksekutif ini diperuntukkan bagi top-level management dalam mengontrol dan mengawas kinerja perusahaan yang dipimpinnya secara ringkas, terintegrasi, mudah dipahami, dan dalam berbagai tingkatan rincian (Imam Firmansyah ; 2017 : 367) [8]

Sistem Informasi Eksekutif (EIS) adalah salah satu jenis manajemen sistem informasi untuk memudahkan dan mendukung keterangan dan pembuatan

keputusan yang dibutuhkan eksekutif senior dengan menyediakan kemudahan akses terhadap informasi baik dari dalam maupun dari luar yang relevan dengan tujuan organisasi. Ini biasanya dipertimbangkan sebagai bentuk dari sistem pendukung keputusan (SPK). EIS menekankan kepada tampilan gambar dan interface yang mudah digunakan oleh pengguna. EIS menawarkan laporan yang kuat dan kemampuan menelusuri. Secara umum, EIS adalah perusahaan lebar SPK yang membantu para eksekutif menganalisa, membandingkan, dan menyoroti variabel penting sehingga mereka dapat memonitor kinerja dan mengidentifikasi kesempatan dan masalah. EIS dan teknologi gudang data berkumpul di pasar. (Delpiah Wahyuningsih ; 2017 : 97).[9]

II.2.4. Metode OOAD (*Object Oriented Analisis Design*)

Menurut Muslihudin dan Oktafianto (2016:55) Pemrograman berorientasi objek bekerja dengan baik ketika bersama dengan object oriented analysis and design process (OOAD) dan mengatakan jika kita membuat program berorientasi objek tanpa OOAD, ibarat membangun rumah tanpa terlebih dahulu menganalisis apa saja yang dibutuhkan oleh rumah itu, tanpa perencanaan, tanpa blueprint, tanpa menganalisis ruangan apa saja yang diperlukan, berapa besar rumah yang akan dibangun dan sebagainya. Konsep OOAD mencakup analisis dan desain sebuah system dengan pendekatan objek, yaitu analisis berorientasi objek atau Object Oriented Analysis (OOA) dan desain berorientasi objek atau Object Oriented Design (OOD).[10]

II.2.5. PHP

PHP (*Personal Home Page*) adalah pemrograman (interpreter) adalah proses penerjemahan baris sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. Definisi lain mengenai PHP adalah skrip bersifat *server-side* yang ditambahkan ke dalam HTML. PHP sendiri merupakan singkatan dari *Personal Home Page Tools*. Skrip ini akan membuat suatu aplikasi dapat diintegrasikan ke dalam HTML sehingga suatu halaman web tidak lagi bersifat statis, namun menjadi bersifat dinamis. Berdasarkan beberapa pengertian PHP tersebut dapat disimpulkan bahwa PHP (*Hypertext Preprocessor*) adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang dapat ditambahkan ke dalam HTML (Supono : 2016 : 3) [11]

II.2.6. Database

Database atau biasa disebut basis data merupakan kumpulan data yang saling berhubungan. Data tersebut biasanya terdapat dalam tabel - tabel yang saling berhubungan satu sama lain, dengan menggunakan field/kolom pada tiap tabel yang ada (Agus Prayitno ; 2015 : 2).[12]

Penekanan file pada basis data adalah kemampuan untuk mengakses data dengan cepat dan efisien dalam menggunakan media simpanan luarnya. Faktor yang mempengaruhi hal ini adalah organisasi dari file basis data. Organisasi file

basis data ini mencoba meningkatkan struktur dari data antara satu file dengan file yang lainnya (Junidar ; 2015 : 19) [13].

II.2.7. MySQL

MySQL merupakan salah satu sistem database yang sangat handal karena menggunakan sistem SQL. Pada awalnya SQL berfungsi sebagai bahasa penghubung antara program *database* dengan bahasa pemrograman yang kita gunakan. Dengan adanya SQL maka para pemrogram jaringan dan aplikasi tidak mengalami kesulitan sama sekali di dalam menghubungkan aplikasi yang mereka buat. Setelah itu SQL dikembangkan lagi menjadi sistem database dengan munculnya MySQL. MySQL merupakan database yang sangat cepat, beberapa user dapat menggunakan secara bersamaan dan lebih lengkap dari SQL. MySQL merupakan salah satu *software* gratis yang dapat di-download melalui situsnya. MySQL merupakan sistem manajemen database, relasional sistem database dan *software open source*. (Stendy B. Sakur ; 2015 : 58) [14]

II.2.8. Teknik Normalisasi

Salah satu topik yang cukup kompleks dalam dunia manajemen *database* adalah proses untuk menormalisasi tabel-tabel dalam *database relasional*. Dengan normalisasi kita ingin mendesain *database relasional* yang terdiri dari tabel-tabel berikut :

1. Berisi data yang diperlukan.
2. Memiliki sesedikit mungkin redundansi.

3. Mengakomodasi banyak nilai untuk tipe data yang diperlukan.
4. Mengefisienkan update.
5. Menghindari kemungkinan kehilangan data secara tidak disengaja/tidak diketahui.

Alasan utama dari normalisasi *database* minimal sampai dengan bentuk normal ketiga adalah menghilangkan kemungkinan adanya “*insertion anomalies*”, “*deletion anomalies*”, dan “*update anomalies*”. Tipe-tipe kesalahan tersebut sangat mungkin terjadi pada *database* yang tidak normal.

a. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

b. Bentuk normal tahap pertama (1” Normal Form)

Definisi :

Sebuah table disebut 1NF jika :

- 1) Tidak ada baris yang duplikat dalam tabel tersebut.
- 2) Masing-masing cell bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

c. Bentuk normal tahap kedua (2nd normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.

d. Bentuk normal tahap ketiga (3rd normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi $X \rightarrow A$, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- 1) X haruslah superkey pada tabel tersebut.
- 2) Atau A merupakan bagian dari primary key pada tabel tersebut.

e. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

f. Boyce Code Normal Form (BCNF)

- 1) Memenuhi 1st NF

Relasi harus bergantung fungsi pada atribut *superkey* (Kusrini ; 2014 : 39-43) [15]


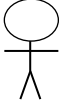


II.2.9. Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (Indra Griha Tofik Isa ; 2017 : 141) [16]

a. *Use case Diagram*

Use case diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*). sehingga pembuatan *use case diagram* lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case diagram* mempresentasikan sebuah interaksi antara aktor dengan sistem.

Tabel II.1. Simbol *Use case*

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>Use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use case</i>, tetapi tidak memiliki control terhadap <i>Use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>Use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidिकासikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>Use case</i> yang</p>

	menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
----->	<i>Include</i> , merupakan di dalam <i>Use case</i> lain (<i>required</i>) atau pemanggilan <i>Use case</i> oleh <i>Use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
<-----	<i>Extend</i> , merupakan perluasan dari <i>Use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Indra Griha Tofik Isa ; 2017 : 141)

b. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Tabel II.2. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4




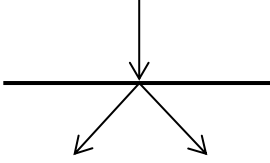
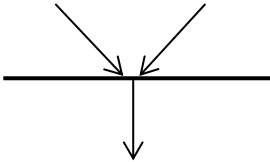
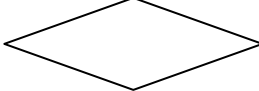

(Sumber : Indra Griha Tofik Isa ; 2017 : 141)

c. Activity Diagram

Activity Diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity Diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa

eksekusi. *Activity Diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *Activity Diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Tabel II.3. Simbol Activity Diagram

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>Activity Diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Indra Griha Tofik Isa ; 2017 : 141)

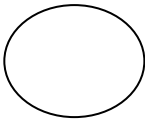
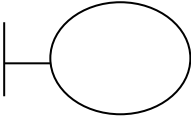
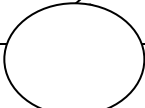
d. *Sequence Diagram*

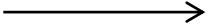
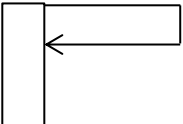

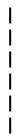
Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence Diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah

yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab

	kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Indra Griha Tofik Isa ; 2017 : 141)