CHAPTER 2

THEORETICAL FOUNDATION

2.1 Theoretical Foundation

2.1.1 System Analysis and Design

Witten provides a description of System Analysis and Design (SAD) which is a process of analyzing business problems and requirements for Information System (IS), followed by designing IS that fulfill business requirements using appropriate technology.

Similar to definition above, Turban describes system analysis as a process of requirement identification and analysis of business requirement using a standardized method. [4] The outputs of this step are namely: (1) the business problem, (2) the motive, (3) output of solution, and (4) requirements.

Purpose of system analysis is to analyze existing system, identify business requirements, and analyze several candidate solutions. While, the purpose of system design is to develop physical model and to implement it further using chosen technology to improve the existing system. [3 - 5]

Furthermore, Turban argued that system analysis stage should produce key information as follow:

- Thorough outlook of existing system
- Solution destined for current problem
- User requirements for a new system

The challenge in implementation of system analysis which Witten defines is to determine boundary between end of system analysis and beginning of system design. In this subject, Turban gives the guideline to identify which activities that are parts of system analysis and parts of system design. These activities can be included into system analysis if the purpose of those activities is to define *what* system lack to complete business problem. On the other hand, activities are parts of system design if those purposes are to define *how* new system will achieve the goal.

Furthermore, Turban argued that the design phase main key deliverable is the technical design that covers following:

- System outputs, inputs and user interfaces
- Hardware, software, databases, telecommunications, personnel, and procedures
- How these system pieces are integrated

Similarly, Satzinger added that system analysis should include system requirements to produce information on what system should do independently to particular technology. Benefit of these technology independencies is that developers can focus to what services system should provide rather than to how it is going to be implemented. Satzinger proposed several approaches: (1) structured analysis, (2) information engineering, and (3) object oriented analysis.

2.1.1.1. Structured Analysis

Following Satzinger's statement, one popular technique in system analysis method is structured analysis. Which helps developer clearly describe systems

require (processing requirements), required data stored (data requirements), desirable inputs and outputs, and overview of system work to achieve tasks. Witten realized that it has evolved into a technique that model data and interface building blocks as a secondary emphasis. Furthermore, Satzinger explains that model to be used with structured analysis approach is called Data Flow Diagram (DFD) that shows inputs, processes, storage, and outputs of way the system works together. [6]

By using graphical model such as DFD, structured analysis could be defines as a model-driven and process-centered technique either to analyze an existing system, identify process and data requirement for the new system, or both. [3]

In the end, most people thought system development should begin only after organization completed an overall strategic system planning to ensure system are comprehensive and coordinated. Therefore, they wanted more advance approach to determine which system should be built and initial requirements models that ensure system compatibility. [6]

These demands initiate developers turned to an adjustment of structured development: information engineering.

2.1.1.2. Information Engineering

Demands for refinement in structured analysis weaknesses creates more advance technique called information engineering. This approach begins by overall strategic planning to define organization's information system requirement to conduct its business (the application architecture plan). It also breaking down and describing explanation of

business functions and activities that system should support, system's data entities information that needed to be stored and technological infrastructure that organization plans to use to support IS. [6]

There are several similarities between structured analysis and information engineering, both approach includes process in data requirements. Satzinger concludes process dependency diagram which is the processing model of information engineering are very similar to DFD, but the difference is, it focuses more on processes that depends on other processes, and less on data inputs and outputs; this method are process-sensitive technique. Similarly, by referring to Witten explanation, another difference is information engineering defined as a data-centered model because it emphasizes the study and requirements analysis of data requirements before those of process and interface requirements.

Moreover, Information engineering approach is said to be a better adjustment in many aspects compared to structured approach. With it, developers initiate a more rigorous and comprehensive methodology. However, in some other cases, even though many recent IS projects using object-oriented technology, traditional approach that merges key aspects from two previous approaches into one, is still widely used for IS development. [6]

2.1.1.3. Object Oriented Analysis

Turban, on the other hand describes an entirely different approach to IS methodology which is object-oriented analysis (OOA). Development process in this

approach begins with feasibility study and analysis of existing system. This approach does not introduce any processes or programs and there are definitely no data entities or files, instead it will consist of objects.

Objects are important element in OOA, which represents a tangible real-world entity, i.e. costumer. Therefore, this approach defines all relational objects needed for the improved system, including their properties (data values), operations (behaviors), and their interaction to achieve system objectives. Meanwhile, Witten explains definition of the object itself, it is a specific data and processes that create, read, update and delete that data are integrated into constructs.

Witten continues to state that OOA fulfill demanding need of a new system analysis approach that corresponds to recent trend of using object technology to build new applications. The technique itself integrates data and process concerns into a constructs called objects.

Purpose of performing OOA is to achieve a better understanding of system and its requirement. By performing object modeling, author describe functions of system graphically, identifying the business objects, its behavior and their relationship with each other.

This approach uses models to illustrate system's objects from various perspectives such as structure and behavior, Unified modeling language (UML) provides the best graphical syntax for it. [3]

2.1.2. Unified Modeling Language

Denis describes by the early years, object-oriented methods were slow to become generally used. It's because many different companies and tool vendors implement it in diverse ways. Therefore, three industry leaders worked side by side to create a standard set of diagramming techniques known as Unified Modeling Language (UML). They try to build a tool that facilitates a standard language of object-based terms and a diagramming technique which sufficient enough to model any systems development project from analysis through implementation.

Other cases analyzed by Turban, state that a model is essential when developing a complex software system. This is where UML takes part. It is a standard language to specify, visualize, construct, and document elements in object-oriented software systems. To state that it's a standard language, UML does not dictate a method of developing system-it only helps to ease the reuse of elements because it provide a widespread set of notations that can be used for all types of software projects.

Moreover, he states that UML offers different choices of diagramming technique to model a system which works more like blueprints. Analyst and developers would not be confused to learn the language, because all of the option uses same syntax and notation. UML evolve to be powerful language for developers because it's constant notation and ability to integrate among diagramming techniques.

They are various common UML diagrams in OOA as follows:

- **Use case diagram.** It is the key building block of UML. Use case diagram shows description of external users and define what way of interaction users expect with system. Other function of use case is to capture system business

requirements. The use case narrative is used in addition to textually explain sequence of steps of each interaction.

- **Class diagram.** It shows the static nature of a system at class level. Class diagram describe system's object arrangement. They illustrate relationships between those object classes modeled in system.
- **Object diagram.** It is similar to class diagram, but instead illustrating object classes, it models actual object instances, showing value instance's attributes. Object diagram are not often to be used, it only used when actual instances attribute of classes will give better understanding of the model.
- **Sequences Diagram.** It illustrates interaction between classes of a use case or operation, by given time sequence. They model behavior of classes in a use case. Sequence diagram graphically define processes of objects interaction using messages within a certain rule of time sequence.
- **State Diagram.** State diagrams are used to describe sequence of states that an object can assume, events that affect an object to change from state to state, and results of significant actions that occur. Those dynamic behaviors of a particular object are well illustrated in state diagram.

2.1.2.1 Use Case

Use cases are the major drivers for UML diagramming techniques. By breaking down the scope of system functionality into many smaller statements, use case has proven to be an excellent technique to give better understanding and document system

requirement. It is best describe as a technique which provides understandable terminology that demands a high level of user involvement by describing system functionality from external users point of view. [3]

Related to Witten statement, Satzinger added that Use case also describes an event response activity carried out by the system. However, It could not be done if two most important concepts are not fulfilled, which are there must be a person involved, and person uses the system.

Looking back to the activities when author perform OOA, the approach that commonly used to model system functionality aspects is Use case modeling. It provides graphical description of a business event, actor who takes part, and system interaction according to series of events. Importance of use case itself lies on the scenario it tells, both automated and manual for purpose of completing a single business task. [3]

Use case shows interaction between actors and system. The interaction itself includes whole different sequences of individual steps to accomplish business requirements. Scenarios consist of many different sequences. Thus, it describes and identifies unique set of internal activities within use case. [6]

Use case diagram simply model main functions of the system and actors that interact with it. There are several of elements worn by use case diagram for describing, furthermore explained in figure 2.1:

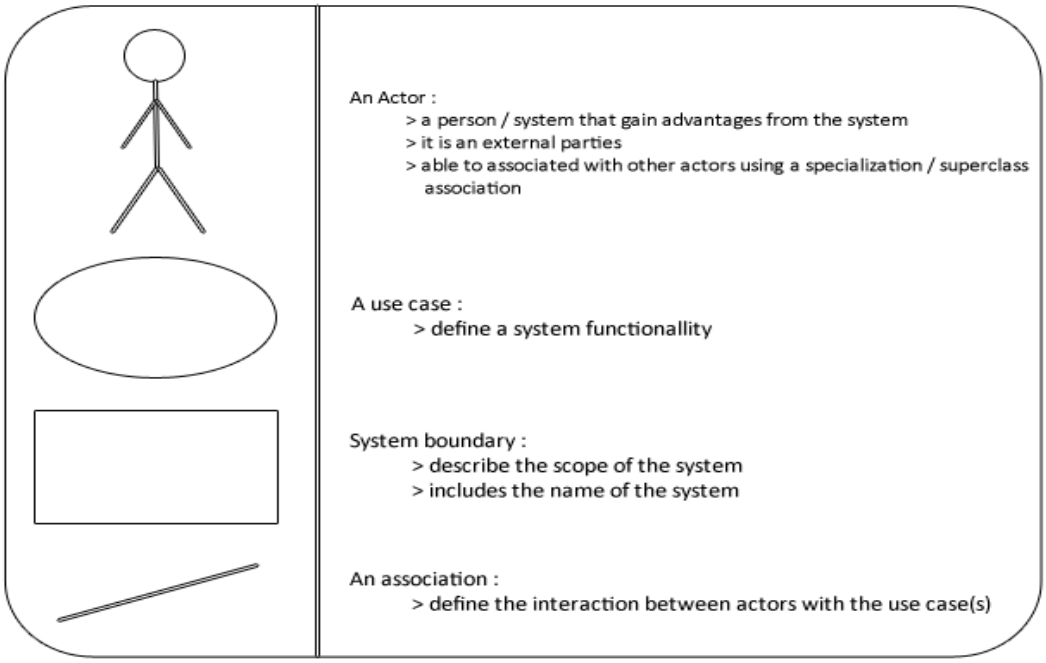

Figure 2.1 Use case Diagram explanatory [source: Turban]

In order to create a well-illustrate use case diagram, there could be many approaches, including prototyping, interview from user and business analyst point of view, and many miscellaneous fact finding technique. To describe creation of use case diagram, better to examine it step-by-step:

Step 1: Identify Use Cases. First step to create an excellent use case diagram is by going through process of identifying use cases that takes part in system's functionality. This process includes interviewing users, and analyzing system context model diagram that illustrates interaction between actors and system itself.

Step 2: Construct a Use Case Model. After the entire important elements of use case have been identified, system boundary will be formed to separate use cases

from actors. Subsystems resulted from system separation holds important role in providing better understanding of system architecture, and to define development strategy. The subsystems represent logical functional areas of business processes.

Step 3: Document the Use Case Course of Events. For the next step, every each use case identified, inserted it into system boundary. There are a constraint involving use case's total number in one system boundary, there could be no more than eight use case identified in it. If definitions of use case are more than that, it should be grouped into packages in order to give better understanding of diagram and maintain model's level of complexity.

This stage main function is to add special associations onto the model. This step is needed to define business events and requirements. It will then refine as the stage goes on to buff up more detail based on result from fact finding we earn throughout the development process.

Step 4: Define the Analysis Use Cases. After all use case's requirements have been defined and approved by users, it will be refined by giving more detailed information. This is done to specify system functionality. It results in free implementation details analysis use case that still could be further refined.

2.1.2.2 Sequence Diagram

It is used to show the interactions between objects in a sequential order of the occurrence of those interactions. The main purpose of a sequence diagram is to define event sequences that result in some desired outcome. The focus is more on the order in which messages occur. Nevertheless, most sequence diagrams will communicate what

messages are sent between a system's objects as well as the order in which they occur. The diagram conveys this information along the horizontal and vertical dimensions: the vertical dimension shows, top down, the time sequence of messages/calls as they occur, and the horizontal dimension shows, left to right, the object instances that the messages are sent to.

During the requirements phase of a project, analysts can take use cases to the next level by providing a more formal level of refinement. When that occurs, use cases are often refined into one or more sequence diagrams. The sequence diagram is a good diagram to use to document system's requirements and to map out the system's design. The reason the sequence diagram is so useful is because it shows the interaction logic between the objects in the system in the time order that the interactions take place. [14]

2.1.2.3 Class Diagram

Once the major conceptual relationship between objects already identified, it's time to document and organize it by using the next major diagramming technique which is the class diagram. It is a static model that supports the static view of the evolving system. Class diagram illustrates the association or relationship between objects and still remain constant in the system over time. [5] It has the similar approach with Entity Relationship Diagram (ERD) but different in the detail of description.

There are several cases occurs when class diagram have to illustrates all the classes and relationship for a real-world system, and it will definitely become very complex; instead of providing a better understanding what happen is actually the reverse. Two ways to simplify class diagram is using view and with the use of

packages. View as the first option, function as the amount of information displayed limitation. Views itself are subsets of information contained in the entire model. Second, packages function to create a more simplified and understandable diagram and still keep the level of complexity at the reasonable level. [5]

The concept of constructing this diagram could be similar to an iterative process which the analyst start this process by drawing a rough version of it and then by the time goes on, it keeps on refined until it reaches perfect model. The detailed steps to construct the diagram are:

Step 1: Identify Associations and Multiplicity. In this step we need to define each of class information that we want to capture. Other word for it is to identify associations that occur between object and classes. Association in this case describes what each class should know about each other, which in the end will support cross reference between objects.

Step 2: Identify Generalization/Specialization Relationship. Second step to construct class diagram is to determine whether there occur any generalization/specialization relationship. If its occur, it is added by drawing an association lines, by going through every class and determine what kind of relationship they have, and the number of instances involved in it. Understand that definition of relationship in this case similar as classification hierarchies, consist of super and sub object.

The advantage in generalization/specialization relationship is that it allows the usage of inheritance, which gives the authority to reuse objects and programming code.

Step 3: Identify Aggregation Relationship. Next step is to determine whether any aggregation or composition relationship occurs. It is defined as a whole/part relationship,

which determines one object to be part of the other object, and the others are the container. Thus, it doesn't imply any inheritance method.

Step 4: Prepare the Class Diagram. Last step to construct class diagram is to prepare or identifying the classes. By identifying it means we will choose and show which classes that are needed for the system as a whole. The diagram lastly will reflect the associations, and relationships discovered in previous steps.

2.1.3. Entity Relationship Diagram

To discuss the crucial aspect in building a good database, Entity Relationship Diagram (ERD) provides the best alternative modeling technique that illustrates information or data that is created, stored, and used by a business system. In terms of entities and relationship describe by it. Even though entities that are related to each other are usually placed close together, ERD implies no order. [5]

Agreeing to Witten statement, to represent all instances of a similar entities group, we need a concept that called entities. It is the building block for a data model whether it is a class of person, place, event, or thing about which data is collected. Each entity has their short description that gives information on which are called identifier. Multiple instances could exist in one entity. Instance itself is a single occurrence of an entity, so it's very important to differentiate between an entity and its instances.

Furthermore, Denis describes entity's definition, in which we want to store data, should be identified or specified with pieces of data called attributes. It is a descriptive type of information that captured from an entity. Meanwhile, there are lots of attributes that could be added in, It is important to sort information that really useful to business

process to be included in the model. One or more attributes could serve as identifier. It can uniquely identify one instance of an entity.

Witten added by knowing that one entity could have thousands or millions of instances, we need to uniquely identify them to distinguish one another. Thus, every entity must have a key or an identifier. It is an attribute, or group of attribute, that describes a unique value of each entity instance.

Theoretically, entities and attributes do not isolate each other. They interact with and impact one another to support business requirements and achieve the goal. The term relationship comes up which is a natural business association that occur between each entities, it may represents any events that have an impacting effect between them. To graphically illustrate the relationship is by using lines that connecting entities and they can be interpreted in both directions. [3]

2.1.4. Data Flow Diagram

This chapter discusses the refinement of all requirements definition and use cases into a process model. It is a representation of business system operation and a technique to illustrate processes that are performed as well as data activities among them. Process model could be useful to document the current system or even the on-going-development system, whether it computerized or not. [5]

Witten defines Data Flow Diagram (DFD) as one of the tool or technique that commonly used to model a process. DFD describes data passes inside the system and also business process performed by that system. Despite the name is more focused on data implication, this technique focuses on activities that are being performed.

Meanwhile, Satzinger gives other definition for DFD as a graphical system model that display all main requirements for an IS in one diagram, including every aspects of working system. With minimum training people range from end users, management, and all IS workers could easily read and understand DFD because it's a graphical model.

There are several elements of DFD, which includes a set of symbols and syntax rules, representing processes, data flows, data stores, and external entities that could be seen in figure 2.2:

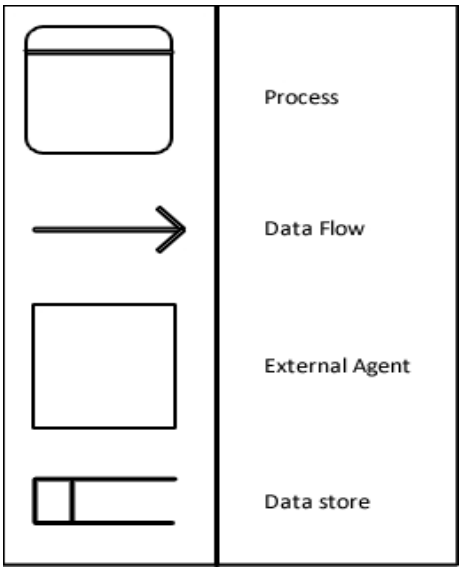

Figure 2.2 Data Flow Diagram explanatory [Source Witten]

Process

Process is an activity or a function that performed to achieve some specific business reason. It can be either manual or computerized. It also explains step-by-step instructions followed by transformation from inputs onto outputs. Each processes forms only one activity, and they always have a unique identification or description.

Data Flow

It performs either as an inputs or outputs from one process onto another. Data flow represents data movement among processes, data stores, and external agents. Each process has at least one input data flow because it is difficult to generate an output if there is no any input.

External Agent

A person or organization, outside the system boundary, but interacts with system. It acts as the source or destination of data outside system boundary. However, person who performs a process is often described in process description but never on DFD itself.

Data Store

It is a collection of data that stored in some way. Data store includes self description and identification with definite at least one input in each of data store. Every input to data store means that there are data or information being added into it, and every output from data store indicates that there is information being retrieved from it.

2.1.5 Data Management

Ramakrishnan states that a good system must have well managed information system. This information is derived from raw facts known as data. Data are efficient when they are managed in a database. Database is a collection of data, typically describing activities of one or more related organizations. Data management main function is to store data in files and write application - specific code to manage it.

2.1.5.1 Database Management System

DBMS as describe by Date is the software layer between physical database itself and system users. [8] All requests for access to database are handled by DBMS such as adding, removing and updating data. Furthermore, Silberchatz stated that Database Management Systems (DBMS) is a collection of interrelated data and a collection of programs to access that data. [9]

The advantages of DBMS are discussed by Connolly as: [10]

- Using database approach to controls amount of redundancy by integrating the files so that multiple copies of same data are not stored.
- Database can be shared by all authorized users in the entire organization.
- Database integrity to ensure validity and consistency of stored data. Integrity is usually expressed in terms of constraints, which are consistency rules that database are not permitted to violate.
- Database security that protects database from unauthorized users.
- Reduced development time and increased programmer productivity.

2.1.5.2 DBMS Functions

Briefly described by Connolly, there should be eight services provided by any full-scale DBMS. Such as:

- **Data storage, retrieval, and update**. Users must be able to store, retrieve and update data in the database.

- **A user-accessible catalog**. A catalog must be provided in which descriptions of data items are stored and accessible to users.
- **Transaction support**. A DBMS must provide mechanisms which will ensure either that all the updates corresponding to a given transaction are made or that none of them is made.
- **Concurrency control services**. A DBMS must guarantee that the database is updated correctly when multiple users are updating the database at the same time.
- **Recovery services**. A DBMS must furnish a recovering mechanism for database in some event that database is damaged in a way.
- **Authorization services**. A DBMS must ensure that only authorized users can access the database.
- **Support for data communication**. A DBMS must be capable of integrating with communication device.
- **Integrity services**. A means to ensure that both data in database and changes to data obey certain rules.

2.1.5.3 Structured Query Language (SQL)

SQL is a database language that is convenient and flexible to enable users to use similar command structure and syntax across different DBMS. Based on Connolly explanatory, SQL is a database language that allows a user to:

- Create database and relation structures.

- Perform basic data management tasks, such as insertion, modification, and deletion of data from relations.
- Perform both simple and complex queries.

As a language, ISO SQL standard has two major components, which are: (1) Data Definition Language (DDL) for defining database structure and controlling access to data. (2) Data Manipulation Language (DML) for retrieving and updating data. Furthermore, Silberchatz added that besides DLL and DML SQL have several other components which are:

- **View definition**. SQL DDL include commands for defining views.
- **Transaction control**. SQL includes command for specifying beginning and ending of a transaction.
- **Embedded SQL and Dynamic SQL**. Define how SQL statements can be embedded within general purpose languages, such as C, C++, COBOL, and Pascal and so on.
- **Integrity**. The SQL DDL include commands for specifying integrity constraints that data stored in database must satisfy.
- **Authorization**. The SQL DDL include commands for specifying access rights to relations and views.

By referencing to Date, basic structure of an SQL expression consists of three clauses: select, from, and where.

- S**elect** clause corresponds to the projection operation of relational algebra. It is used to list attributes desired in result of a query.

- F**rom** clause corresponds to the Cartesian-product operation of relational algebra. It lists the relations to be scanned in expression evaluation.
- W**here** clause corresponds to the selection predicate relational algebra. It consists of a predicate involving attributes of relations that appear in *from* clause.

2.1.6. C#

To analyze existing system and to design new system requires basic theories of SAD, tools for making conceptual design and data management concept. However, to physically build system based on analysis and design, it requires standardized tools and frameworks to enable creation of a fully running system. Tools that are going to be used in this thesis project is based on Microsoft web development framework using C# as the core programming language, using ASP.NET 2.0 as development tools, ADO.NET as data access library.

C# pronounced "see-sharp" is a hybrid of C and C++. It is a Microsoft programming language developed to compete with Sun's Java language. C# is an object-oriented programming language used with XML-based Web services on .NET platform and designed for improving productivity in Web applications development. C# boasts type-safety, garbage collection, simplified type declarations, versioning and scalability support, as well as other features that make developing solutions faster and easier, especially for COM+ and Web services [11].

2.1.7. ASP.NET 2.0

ASP.NET is a set of Web development tools offered by Microsoft. Programs like Visual Studio .NET and Visual Web Developer allow Web developers to create dynamic websites using a visual interface. Of course, programmers can write their own code and scripts and incorporate it into ASP.NET websites as well.

ASP.NET is built on .NET framework, which provides an application program interface (API) for software programmers. The .NET development tools can be used to create applications for both Windows operating system and Web. Programs like Visual Studio .NET provide a visual interface for developers to create their applications, which makes .NET a reasonable choice for designing Web-based interfaces as well [12].

2.1.8. ADO.NET

The simplest definition of ADO.NET is that it is Microsoft's new .NET object library for data access. This is the most recent of Microsoft's access methods (previous products were DAO, RDO and ADO). But in a more fundamental way, this is a new revolution in software.

Microsoft's data technologies have been among their most successful products and a big part of the reason why is wrapped up in the word "standard". ADO.NET can be used with all ODBC data sources, XML, Oracle, SQL Server [13].

2.1.9 Internet

Our application is using web based technologies because it is easier to deploy in web environment or in a local windows network using a server. In this section we will define several terms that is essential to run a web application, which are internet, Hyper Text Transfer Protocol (HTTP) and Hyper Text Markup Language (HTML).

Based on Turban definition, Internet is a global network of computer networks. It links computing resources of businesses, government, and educational institutions using a common computer communication protocol, TCP/IP.

2.1.9.1 HTTP

HTTP defined by Kaufman is the protocol for retrieving web pages. It is a stateless request / response protocol. Client machine makes a request, and in response is the content of that page.

Mitchell similarly added that HTTP is the Hypertext Transfer Protocol which provides a communication standard for Web browsers and servers and also describes the technical specification of a network protocol that software must implement. [24]

2.1.9.2 Hyper Text Markup Language (HTML)

One other component of Internet stated by Turban is HTML (Hypertext markup language). He added it definition which is a simple language that useful for displaying static content to viewers. HTML has very limited capabilities for interacting with viewers or for providing information that is continually being updated. It is not suitable for collecting information, such as names and addresses, or for providing animation or changing information such as stock quotes.

2.1.9.3 Cascading Style Sheets (CSS)

Invented in 1997, Cascading Style Sheets (CSS) is widely used among browsers and Web developers. Style sheets are a very powerful tool for a website developer. It gives the chance to be completely consistent with the look and feel of your web pages, and giving the developer more control over the layout and design than just using HTML. With CSS, designers and users can create style sheets that define how different elements, such as headers and links, appear. [21, 22]

2.1.10 CRM

SMEs needed a methodology that will provide them with rapid customer's report that will notice them what customer status is to increase their profit. By keeping track of this information, SMEs can decide which promotion type suitable for their loyal customer. Therefore, Customer Relationship Manager (CRM) comes up. Edelstein defines CRM as a methodology in managing all customers' interactions, which requires their information and prospect to be more efficiently manage interactions with them in all stages of relationship. [17]

Moreover, Edelstein added the three stages which known as customer life cycle:

- Acquiring customers
- Increase in customer value
- Sort the good customers

Supporting Edelstein theory, Turban added that CRM is a business strategy to select and manage customers to optimize long term value. It requires a customer centric

business philosophy and culture to support effective marketing, sales, and service processes. [18]

SMEs in one hand needs CRM feature to be able to contend with market competition, but it also have the limitations which could cost them major blow if it not well managed. The benefit of CRM explained by Turban is that it will try to provide wide range choices of products and services, fast problem resolution and response, easy and quick access to information. This beneficial factor definitely will give competitive advantage over the competitor. Turban added the limitations of CRM which are companies will have hard time in justifying the expense of CRM because they cannot calculate the exact cost number it needed to lure and bring more customers to buy their products.

CRM consist of 3 types of activities defined by Turban:

1. **Operational.** Consist of customer service, order management, invoice or billing, sales and market automation.
2. **Analytical.** Consist of capturing, storing, extracting, processing, interpreting, and reporting customer's data to user, who then will analyze them as needed to produce better marketing plan.
3. **Collaborative.** Deals with all necessary communication, coordination, and collaboration between vendors and customers.

CRM have one main aim in its program, which is converting what it is called customer's satisfaction into customer's loyalty. Through Griffin research, customer's satisfactory of our products do not guarantee their repeat purchases and increase in our sales number, which then he discovered that customer's loyalty is the essential part in

gaining competitive advantage over our competitor because in the end what customer really want is cheap price. [19]

Research also states that more than fifty percent customers that satisfied with one product will switch suppliers or vendors without hesitation. This data emerges SMEs to build a good brand identity that different with their competitors and provide a good competitive price which will makes them on top of the competition. Griffin added that increase in loyalty can bring cost saving in a company in at least six areas:

1. **Reducing marketing cost.** Customer acquisition requires more money.
2. **Lower transaction cost.** Including less contract negotiation and order processing.
3. **Reduced customer turnover expenses.** Fewer customers to replace.
4. **Increased cross-selling success.** Leading to larger share of customer.
5. **More positive word-of-mouth.** Assuming the loyal customers is also satisfied.
6. **Reduced failure cost.** Reduce in rework, warranty claims and so forth that caused by customers dissatisfactory from our work and service.

Griffin also emphasize in good employee managerial. This has the advantage in improving loyalty inside and outside the company, while directly reducing the training cost. Loyal employee will attract more loyal customer because in several cases, loyal customer always wants to buy a product from the same employee, in other words they already have a bond and trust. Loyal employee will give them trustable advice each time the customers confused in choosing a product.

There are seventeen rules of CRM which can be sorted to four major rules according to Ellington. First, CRM is more than a product. By choosing to implement a CRM, a company has taken a dramatic step forward in its customer commitment so as to

generate more profit. Patience is the key in CRM. Given CRM enough project time, it will repay this patience by delivering more than the company expected. Second, customers are everywhere. In the recent days, the idea of customer definition has widened to include a broad-range of end users or different kinds of corporate information. Every element in the surrounding can be a customer according to its needs. Third, bigger is not always better. CRM is a wide regulation. The solutions are built to deliver and match different levels of functionality, complexity, structures, methods of working and business size. Fourth, different solutions are made for different companies. Solutions should be adjusted according to its needs, which involve company size. Solutions made for large company cannot be implemented as well in small company. [23]

Critical factors to make key decision making in CRM successfully implemented, divided by five points. [25]

1. When evaluating a CRM solution, decision making cannot be part ways with the ability to integrate with the existing technology.
2. A solution must match to fit a company's exact needs. This should be implemented according to company's capabilities.
3. To successfully implement a CRM, integration to legacy data and low adoption rate by users should be considered as well.
4. ROI is the main point in implementing a CRM.
5. By setting aggressive goals for CRM projects, companies expect increase in sales revenue.

2.1.11 Statistic

Newbold states, to improve the sales and profit income to a company, statistical calculation is required. It counts the probability of transaction a customer made. By knowing it, it will help to acknowledge what user really wants and needs. The result also shows time and frequency a transaction initiated, in months and days.

He then continues to state that, various processes of statistics could be used to summarize samples of data. This will enable to construct probability models called sampling distribution. Statistical procedures focus on drawing inferences about large populations of items using a small sample of the items.

Instead using the whole population, samples are taken due to cost measurement of every item in population will be prohibitive. Sample collected from a population so that valid statements can be made about the population as a whole. The ideal sample is a *simple random sample.*

It's important to differentiate between population attributes and the corresponding sample quantities, which mean an attribute of a population is in a fixed number. By drawing a random sample from the population and computing the sample mean, the interferences about the attribute can be concluded. The distribution of possible sample outcomes provides a basis for inferential statements about the sample. The probability distribution of the values could take over all possible sampling of the same number of observations drawn from the population. This statistic is called *sampling distribution.* [20]

$$\mu = ?_i / n$$

μ = Sample mean

$?_i$ = Sample

n = Sample size

The sampling distribution becomes concentrated closer to the population mean as the sample size increases.

2.2 Theoretical Framework

2.2.1 System Development Life Cycle

Every project need a structured work outline, in this case, System Development Life Cycle (SDLC) is one of the standard framework used to create common project. Turban states that system development refers to set of activities that create systems for effective and efficient information processing.

An SDLC shows the major steps, over time, of an information systems development project. Moreover, Turban also added that there are an eight-stage, or groups of major tasks SDLC. The stages are:

- **Stage 1: Project Initiation.** Projects often start when a manager has a problem or sees an opportunity related to the area where he or she works. The manager call IS and requests that a formal planning process be initiated to discover ways that can help organization to meet its objectives. Sometimes the IS group initiates projects that will improve its own operations or deal with common problems among the user areas.
- **Stage 2: System Analysis and Feasibility Studies.** Stage 2 consists of two phases of analysis: system analysis and feasibility studies.

- **System analysis.** It is the phase that develops a thorough understanding of the existing organization, its operation, and the situation that is causing a problem. System analysis method include observation, review of documents, interviews, and performance measurement
- **Feasibility Studies.** It calculates the probability of success of the proposed solution; they may be run several times throughout the systems development life cycle.

- **Stage 3: Logical Analysis and Design.** The emphasis at third stage is the design of a system forms business user's point of view. The analyst identifies information requirements and specifies processes and generic IS functions such as input, output, and storage, rather than writing programs or identifying hardware.
- **Stage 4: Development or actual acquisition**. Logical design of the new system guides actual development or acquisition, just as blueprints guide the construction of a new building.
- **Stage 5: Implementation**. It is obviously an important stage; the system can fail here even if it has all the specified functionality. The project team should plan the implementation very carefully, to avoid problem that could lead to failure or user resistance. Users require training in the mechanics of system to reduce frustration and to minimize productivity losses in transition period. In most cases, implementing a new system requires a conversion from a previous system.
- **Stage 6: Operation.** After a successful conversion, system will operate for an indefinite period of time, until the system is no longer adequate or necessary, or cost effective

- **Stage 7: Post-Audit Evaluation.** An organization should perform a post-audit to evaluate all its larger systems projects after their completion. Post-audits introduce an additional element of discipline into development process. If the implementation was successful, an audit should occur after system's operations have stabilized. If the project failed, audit should be done as soon as possible after the failure.
- **Stage 8: Maintenance.** Every system needs two regular kinds of maintenance: bugs fixing and regular system updating. Maintenance is expensive; accounting for up to 80 percent of organizational IS budgets. Therefore it is important that the design and development stages produce systems that are easy to maintain and flexible enough to handle future expansion upgrading and capacity increases.

Similarly, Rob explains that SDLC traces the history (life cycle) of an information system. Traditional SDLC is divided into five phases: planning, analysis, detailed system design, implementation, and maintenance.

2.2.2. Object Oriented Development

By using other variation means of approach, Object Oriented Development explained by Turban is based on fundamentally different views of computer systems than that found in traditional SDLC approaches. Traditional approaches provide specific step-by-step instructions in form of computer programs, in which programmers must specify every procedural detail.

An Object-oriented (OO) system begins not with the task to be performed, but with real world aspects that must be modeled to perform that task. Object technology enables the development of purchasable, sharable, and reusable information assets

(objects) existing in a worldwide network of interoperable interorganizational information systems.