

BAB II

LANDASAN TEORI

2.1 Ekologi Desk

Ekologi Desk yang terletak di daerah Pandeansari Sawitsari blok IV No.10 RT 10/ RW 63, Condongcatur, Depok, Sleman, Yogyakarta. Menyediakan *coworking space* yang mengangkat konsep *eco green*. Berdiri pada tahun 2016 lalu sampai saat ini yang bisa dibilang salah satu pionir *coworking space* yang ada di Yogyakarta.

Ekologi Desk menerima pesanan tempat di sesuaikan kebutuhan pelanggan. Pemesanan yang oleh pelanggan belum bisa dilakukan melalui website atau *online*, karena Ekologi Desk masih menggunakan cara pemesanan yang manual atau langsung. menggunakan sistem baru akan dimulai terlebih dahulu memfokuskan untuk pengelolaan pemesanan *coworking space*.

2.2 Sistem Informasi Manajemen

2.2.1 Sistem Informasi

Sistem informasi adalah suatu sistem buatan manusia secara umum terdiri atas kumpulan komponen berbasis computer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada pra pemakai (Gelinas, 1990).

Sistem informasi adalah suatu sistem didalam suatu oragnisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang yang mendukung fungsi orgasiswa yang bersifat manajerial dalam kegiatan strategi dari suatu organisasi untuk menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Sutabri, 2012).

2.2.2 Sistem Informasi Manajemen

Sistem informasi manajemen adalah serangkaian sub-sistem informasi yang menyeluruh dan terkoordinasi dan secara rasional terpadu yang mampu mentranformasi data sehingga menjadi informasi lewat serangkaian cara guna meningkatkan produktivitas yang sesuai dengan gaya dan sifat manajer atas dasar kriteria mutu yang telah ditetapkan (Scott, 2001).

SIM (sistem informasi manajemen) dapat didefinisikan sebagai kumpulan dari interaksi sistem-sistem informasi yang bertanggung jawab mengumpulkan dan mengolah data untuk menyediakan informasi yang berguna untuk semua tingkatan manajemen di dalam kegiatan perencanaan dan pengendalian (Pangestu, 2007).

2.3 Metode Prototyping

2.3.1 Pengertian Metode *Prototyping*

Proses pengembangan sistem seringkali menggunakan pendekatan prototipe (*prototyping*). Metode ini sangat baik digunakan untuk menyelesaikan masalah kesalahpahaman antara user dan analis yang timbul akibat user tidak mampu mendefinisikan secara jelas kebutuhannya (Mulyanto, 2009).

Metode *Prototyping* adalah suatu proses pembuatan software yang bersifat berulang dengan perencanaan cepat dimana terdapat umpan balik yang memungkinkan terjadinya perulangan dan perbaikan software sampai memenuhi kebutuhan dari pengguna. Dengan metode ini pengguna juga dapat memiliki suatu gambaran awal/dasar tentang program serta melakukan pengujian awal yang didasarkan pada konsep model kerja (*working model*). *Prototyping* mempunyai tujuan yaitu mengembangkan model awal software menjadi sebuah sistem yang final (Chandramita, 2019).

2.3.2 Proses-proses Metode *Prototyping*

Proses-proses dalam metode *prototyping* secara umum adalah sebagai berikut (Chandramita, 2019):

a. Pengumpulan Kebutuhan

Developer dan klien akan bertemu terlebih dahulu dan kemudian menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya.

b. Perancangan

Perancangan dilakukan dengan cepat dan rancangan tersebut mewakili semua aspek sistem yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.

c. Evaluasi *Prototype*

Klien akan mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan sistem.

2.3.3 Tahapan-tahapan Metode *Prototyping*

Untuk memodelkan sebuah sistem informasi dibutuhkan beberapa tahapan di dalam proses pengembangannya. Tahapan inilah yang akan menentukan keberhasilan dari sebuah sistem itu. Pengembang sistem harus memperhatikan tahapan dalam metode *prototyping* agar sistem akhirnya dapat diterima oleh pengguna. Dan tahapan-tahapan dalam *prototyping* adalah sebagai berikut (Chandramita, 2019):

- a. Pengumpulan Kebutuhan
Pelanggan dan pengembang bersama-sama mendefinisikan format dan kebutuhan keseluruhan sistem, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.
- b. Membangun *Prototyping*
Membangun *prototyping* dengan membuat perancangan sementara yang berpusat pada penyajian kepada pelanggan (misalnya dengan membuat input dan contoh outputnya).
- c. Evaluasi *Prototyping*
Evaluasi ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah keempat akan diambil. Jika tidak, maka *prototyping* diperbaiki dengan mengulang langkah 1, 2, dan 3.
- d. Mengkodekan Sistem
Dalam tahap ini *prototyping* yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.
- e. Menguji Sistem
Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan *White Box*, *Black Box*, *Basis Path*, pengujian arsitektur dan lain-lain.
- f. Evaluasi Sistem
Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika sudah, maka langkah ketujuh dilakukan, jika belum maka mengulangi langkah d dan e.
- g. Menggunakan Sistem
Sistem yang telah diuji dan diterima pelanggan siap untuk digunakan.

2.4 Teori Pengujian

Definisi pengujian perangkat lunak adalah proses untuk mencari kesalahan pada setiap item perangkat lunak, mencatat hasilnya, mengevaluasi setiap aspek pada setiap komponen sistem dan mengevaluasi semua fasilitas dari perangkat lunak yang dikembangkan. Terdapat 2 hal utama yang dilakukan dalam pengujian (Perry, 2006), yaitu :

1. Verifikasi adalah proses mengevaluasi suatu *system/component* untuk menentukan apakah suatu produk yang diselesaikan setelah fase pengembangan memenuhi kondisi

seperti yang telah ditetapkan pada awal pengembangan (saat menentukan spesifikasi) perangkat lunak.

2. Validasi adalah proses mengevaluasi suatu sistem atau komponen pada akhir atau selama masa pengembangan untuk menentukan apakah produk yang dihasilkan telah memenuhi kebutuhan-kebutuhan dan persyaratan tertentu yang diminta oleh *user*.

2.5 Black-Box Testing

Black-Box Testing merupakan Teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. *Blackbox Testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada informasi domain. *Blackbox Testing* memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program (Jaya, 2018).

Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Black Box Testing* cenderung untuk menemukan hal-hal berikut (Jaya, 2018):

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

2.6 User Acceptance Test

User Acceptance Testing merupakan pengujian yang dilakukan oleh *end-user* dimana *user* tersebut adalah *staff* karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya (Perry, 2006).