

BAB II

LANDASAN TEORI

2.10. Sistem

Sistem adalah jaringan dari pada elemen-elemen yang saling berhubungan, membentuk satu kesatuan untuk melaksanakan suatu tujuan pokok dari sistem tersebut. (Jogiyanto, 2010).

2.2 Sisteminformasi

Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan 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.(Wicaksono dan Widodo.,2020)

2.3 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Rosa & Shalahudin, 2014).




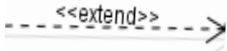

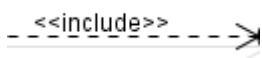
a. Usecase Diagram

Menurut Rosa dan Shalahuddin (2014) mengungkapkan *Use Case Diagram* Pemodelan untuk kelakuan (*Behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Use Case diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use Case* diagram mendeskripsikan sebuah interaksi antara satu atau lebih

aktor dengan sistem informasi yang akan dibuat. Simbol-simbol yang ada pada *Use Case* diagram dapat dilihat pada tabel 2.2 berikut :

Tabel 2.1 Simbol *Usecase Diagram*

No	Simbol	Keterangan
1.	<i>UseCase</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor.
2.	<i>Actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, walaupun simbol aktor adalah orang namun aktor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	<i>Association</i> 	Komunikasi antara aktor dan <i>Use Case</i> yang berpartisipasi pada usecase atau <i>Use Case</i> memiliki interaksi dengan aktor.
4.	Ekstensi/ <i>Extend</i> 	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>Use Case</i> tambahan itu.
5.	<i>Generalization</i> 	Hubungan generalisasi dan spesialisasi antara dua buah <i>Use Case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
6.	<i>Include</i> 	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>Use Case</i> ini.


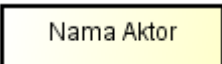

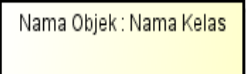
b. Use Case Skenario

Menurut Rosa dan Shalahuddin (2014) *Use case scenario* Sebuah dokumentasi terhadap kebutuhan fungsional dari sebuah sistem. Model *use case (use case scenario)* menggambarkan bagaimana sistem bekerja secara keseluruhan dan pada skenario tertentu sehingga jika sebuah *object* merupakan sesuatu yang menyediakan layanan maka harus dapat dibuktikan bahwa semua layanan tersebut ada dalam suatu *use case*.


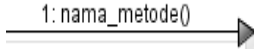

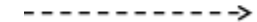
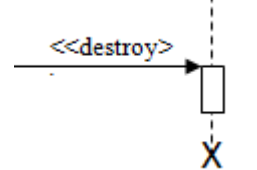
c. Sequence Diagram

Menurut Rosa dan Shalahudin(2014) *sequenceDiagram* menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek atau message yang dikirimkan dan diterima antar objek.Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *usecase* beserta metode-metode yang dimiliki kelas. Simbol – simbol yang ada pada *sequence diagram* dapat dilihat pada tabel 2.3 berikut ini :

Tabel 2.2 Simbol *Sequence Diagram*

Simbol	Keterangan
<p>Aktor</p>  <p>atau</p>  <p>Tanpa waktu aktif</p>	<p>Orang,proses,atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang;biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama aktor.</p>
<p>Garis hidup</p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> 	<p>Menyatukan objek yang berinteraksi pesan</p>

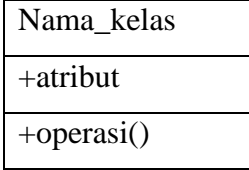


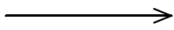
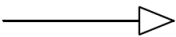

Tabel 2.2 Simbol *Sequence Diagram* (*Lanjutan*)

Simbol	Keterangan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
Pesan tipe <i>create</i> <code><<create>></code>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe <i>call</i> <code>1: nama_metode()</code> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
Pesan tipe <i>send</i> <code>1: masukan()</code> 	Menyatakan bahwa suatu objek mengirim data /masukan/informasi ke objek lainnya , arah panah mengarah pada objek yang dikirim.
Pesan tipe <i>return</i> <code>1: keluaran</code> 	Menyatakan bahwa suatu objek yang telah menjalankan sesuatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu.
<code><<destroy>></code> 	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> .

d. *Class Diagram*

Menurut Rosa dan Shalahudin (2014) *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. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Simbol – simbol yang ada pada *class diagram* dapat dilihat pada Tabel 2.5.4 berikut ini :

Tabel 2.3 Simbol *Class Diagram*

Simbol	Keterangan
Kelas/Class 	Kelas pada struktur sistem
Antar muka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek
Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum ,asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah/ <i>directed</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan atau <i>Dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.

(Sumber: Rosa danShalahuddin, 2014)


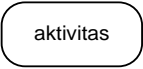



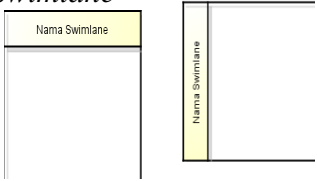
e. *Activity Diagram*

Menurut Rosa dan Shalahuddin (2014) *Activity Diagram* Menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

Perlu diperhatikan disini adalah bahwa *diagram* aktifitas menggambarkan aktifitas sistem bukan apa yang dilakukan aktor, jadi aktifitas yang dapat dilakukan oleh

sistem. Simbol – simbol yang ada pada *activity diagram* dapat dilihat pada tabel berikut ini :

Tabel 2.4 Simbol *Activity Diagram*

Simbol	Keterangan
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status 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 digabungkan 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.

(Sumber : Rosa dan Shalahudhin, 2014)

2.11. HTML (*HyperText Markup Language*)

HyperText Markup Language (HTML) merupakan sebuah bahasa *scripting* yang berguna untuk menuliskan halaman web, pada halaman web HTML dijadikan sebagai bahasa script dasar yang berjalan bersama sebagai bahasa *scripting* pemrograman lainnya, dengan menggunakan *macromedia dreamweaver 8* kode HTML tidak perlu diketik, karena semua perintahnya diwujudkan secara icon base. Tetapi bagi seorang pengembang aplikasi web, kemampuan penguasaan terhadap kode-kode HTML sangat diperlukan. Secara umum tugas-tugas yang dapat dilakukan oleh HTML adalah :

1. Mengontrol tampilan dari *web page* dan isinya.
2. Mempublikasikan web secara *online*.
3. Membuat *online form*.

2.12. *Hypertext Preprocessor*(PHP)

PHP yang merupakan singkatan dari:*Hypertext Preprocessor*, adalah bahasa pemrograman yang mana filenya diletakkan di *server* dan seluruh Prosesnya dikerjakan di *server*, kemudian hasilnya yang dikirimkan ke *client*, tempat pemakai menggunakan *browser*. *PHP* bekerja didalam sebuah dokumen *HTML*(*Hypertext Markup Language*) untuk dapat menghasilkan isi dari sebuah halaman *web* sesuai dengan permintaan. Dengan *PHP*, kita dapat merubah situs kita menjadi sebuah aplikasi berbasis *web*, tidak lagi hanya sekedar sekumpulan halaman statik, yang jarang diperbaharui.

Yang membedakan *PHP* dengan bahasa pemrograman lain adalah adanya tag penentu, yaitu diawali dengan “<? ” atau “<?php” dan diakhiri dengan “? >”. Jadi kita bebas menempatkan *skrip PHP* dimanapun dalam dokumen *HTML* yang telah kita. Beberapa keuntungan menggunakan *PHP* adalah sebagai berikut :

1. *PHP* tersedia secara gratis dan bersifat *open source* sehingga kodenya dengan dimodifikasi dan dikompilasi sesuai kebutuhan.
2. Kecepatan yang tinggi, terutama jika dipadukan dengan *web server apache* sebagai modul *server* .

3. Stabil, *PHP* bersifat *open source*, *source code PHP* terus dikembangkan oleh banyak orang untuk mendapatkan kinerja yang lebih baik dan stabil.
4. Keamanan yang baik dan *PHP* menyediakan banyak level keamanan yang dapat diatur melalui file konfigurasi.
5. *PHP* dapat dikomasi agar dapat berjalan pada banyak *platform*, termasuk *unix* dan *windows*.

2.13. MySQL

MySQL adalah database yang menghubungkan script PHP menggunakan perintah *query* dan *escapscharacter* yang sama dengan PHP. MySQL menjadi sangat populer karena MySQL bersifat *free* (tidak perlu membayar dalam menggunakannya) pada berbagai platform (unix/windows).

Pada sebuah *database* yang mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. Pada saat instalasi, secara *default* MySQL akan membuat sebuah *database* bernama *mysql*. Salah satu isi dari *database* ini adalah tabel *user*, tabel ini berisi nama dan *password user* yang dapat mengakses data pada *database* yang dibuat di *mysql*.

2.14. Basis Data (Database)

Basis data (*Database*) adalah kumpulan dari berbagai data yang saling berhubungan satu dengan lainnya. Basis data tersimpan di perangkat keras, serta dimanipulasi dengan menggunakan perangkat lunak. Pendefinisian basis data meliputi spesifikasi dari tipe data, struktur dan batasan dari data atau informasi yang akan disimpan.

Database merupakan salah satu komponen penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna. Penyusunan basis data meliputi proses memasukkan data kedalam media penyimpanan data, dan diatur dengan menggunakan perangkat Sistem Manajemen Basis Data (*Database Management System - DBMS*).

Database menjadi penting karena munculnya beberapa masalah bila tidak menggunakan data yang terpusat, seperti adanya duplikasi data, hubungan antar data

tidak jelas, organisasi data dan *update* menjadi rumit. Jadi tujuan dari pengaturan data dengan menggunakan *database* adalah :

- a. Menyediakan penyimpanan data untuk dapat digunakan oleh organisasi saat sekarang dan masa yang akan datang.
- b. Cara pemasukan data sehingga memudahkan tugas operator dan menyangkut pula waktu yang diperlukan oleh pemakai untuk mendapatkan data serta hak-hak yang dimiliki terhadap data yang ditangani.
- c. Pengendalian data untuk setiap siklus agar data selalu *up-to-date* dan dapat mencerminkan perubahan spesifik yang terjadi di setiap sistem
- d. Pengamanan data terhadap kemungkinan penambahan, modifikasi, pencurian dan gangguan-gangguan lain.

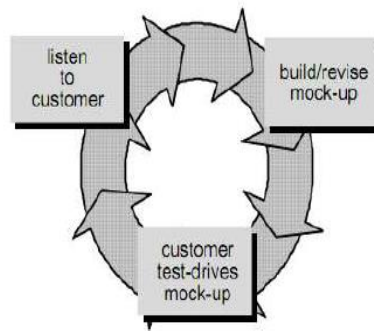
2.15. Macromedia Dreamweaver

Dreamweaver adalah suatu bentuk program editor *web* yang dibuat oleh macromedia, merupakan editor yang lengkap dan dapat digunakan untuk membuat animasi sederhana yang berbentuk *layer*. Dengan adanya program ini seorang programmer *web* dapat dengan mudah membuat dan mendesain *web* nya tanpa susah-susah mengetik script-script format lainnya.

Dreamweaver memiliki dua bentuk layar, yaitu bentuk halaman design dan halaman code yang akan mempermudah dalam menambahkan *script* yang berbasis *PHP* maupun *Javascript*. Selain mendukung pembuatan *web* yang berbasis *HTML*, *dreamweaver* juga mendukung program-program *web* yang lain diantaranya *PHP*, *ASP*, *Perl*, *Javascript* , dan lain-lain.

2.16. Metode Pengembangan Sistem

Metode yang digunakan dalam penelitian ini adalah *prototype modeling* (Pressman, 1997) dengan alur terlihat pada gambar di bawah ini



Gambar 2.1 Prototype Modeling

Proses *prototype modeling* pada gambar di atas akan dijelaskan sebagai berikut :

a. Pengumpulan Kebutuhan

Pengumpulan kebutuhan dilakukan dengan melakukan pertemuan antara pengguna dengan *develover*. Kustomer pada penelitian kali ini ialah kepala pemasaran dari perusahaan. Sedangkan *develover* ialah penulis sendiri selaku subjek yang melakukan penelitian.

b. Perancangan Prototype

Setelah kebutuhan sistem terkumpul, maka mulai dirancang *prototype* sistem. Perancangan dilakukan cepat dan rancangan mewakili semua aspek sistem yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.

c. Evaluasi Prototype

Pengguna mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan *software*. Ketiga proses di atas dilakukan beberapa kali sampai pengguna puas terhadap *prototype* yang dibangun. Ketika pengguna merasa puas atas *prototype* yang dibangun, maka kebutuhan sistem telah tergambarkan seluruhnya dan sistem siap dikembangkan menjadi perangkat lunak.

d. Membangun Software

Prototype yang telah diterima oleh pengguna dikembangkan dan disempurnakan menjadi *software*. *Software* dibangun menggunakan pendekatan berbasis objek (OOP).

e. Menguji *Software*

Pengujian *software* dilakukan berdasarkan metode *black box testing*. Melakukan pengujian kepada semua fungsional sistem.

f. Implementasi

Setelah pengujian dilakukan dan hasilnya positif, maka *software* siap untuk diaplikasikan. Proses pengaplikasian *software* dilakukan dengan memasukan data-data secara menyeluruh

2.10. Tinjauan Pustaka

Adapun tinjauan pustaka yang digunakan oleh penulis adalah sebagai berikut:

a. PERANCANGAN SISTEM INFORMASI PENJUALAN MAKANAN KUCING DAN ANJING BERBASIS WEB

JUMIKA Vol 6 No. 2 (2019) – PISSN : 2355-7494 – EISSN : 2541- 6316

Penulis Andi Taufik, Sistem Informasi, STMIK Nusa Mandiri. Jurnal ini membahas Petshop merupakan tempat yang menjual peralatan, perlengkapan untuk hewan peliharaan serta menjual hewan peliharaan. Dewasa ini usaha petshop semakin berkembang. Agar suatu petshop lebih diminati oleh para pecinta hewan, maka dibutuhkan suatu kelebihan dari petshop tersebut untuk menarik minat para pecinta hewan. Siego Petshop masih menggunakan sistem penjualan yang manual, seperti konsumen harus datang sendiri ke lokasi, pencatatan data penjualan dan stok masih manual menggunakan kertas atau microsoft excel sehingga pencarian data memakan waktu yang lama, dan juga dapat terjadi kehilangan data.

b. Sistem Pakar Diagnosa Penyakit Kucing Berbasis Web Menggunakan Metode Forward Chaining.

Jurnal Speed – Sentra Penelitian Engineering dan Edukasi – Volume 10 No 2 - 2018. Penulis Acmad Nurhadi, dari Akademi Manajemen Informatika dan Komputer, BSI Pontianak. Jurnal ini memiliki identifikasi masalah Kurangnya informasi tentang penyakit kucing dan juga kurangnya kesadaran tentang pentingnya memelihara kesehatan kucing peliharaan mengakibatkan

banyaknya kucing yang tidak terjaga kesehatannya. Dengan menggunakan aplikasi berbasis web, informasi dari suatu pakar akan mudah didapat oleh pengguna, tanpa harus datang pada seorang ahli/pakar yang ahli pada bidangnya. Dengan adanya akses online berbasis web maka pemelihara kucing dapat mendiagnosa kemungkinan penyakit kucing yang diderita kucing peliharaan sebelum mengambil tindakan lebih lanjut seperti konsultasi ke dokter di klinik hewan.