

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Tinjauan Pustaka**

Hardina (2017) dengan judul penelitian “Analisis dan Perancangan Sistem Informasi Gerakan Terpadu Pusat Kesehatan Hewan Kabupaten Bangka Berbasis *Website*”. Pusat kesehatan hewan Kabupaten Bangka adalah suatu organisasi pemerintahan yang bergerak di bidang kesehatan hewan yang bertugas untuk menyediakan pelayanan bagi kesehatan hewan di Kabupaten Bangka. Banyak data yang memerlukan cara kerja yang lebih cepat dan mudah dalam pengolahannya sehingga petugas tidak mengalami kesulitan dalam pengaksesan data karena sebelumnya masih bersifat manual. Oleh karena itu, penulis memiliki inisiatif untuk membantu petugas agar mempermudah dalam mengelola dan menyimpan data organisasi dengan konsep air terjun (*waterfall*) yang memiliki 3 tahap, tahap pertama Perencanaan Sistem, tahap kedua Analisis Sistem, tahap ketiga Desain dan Perancangan Sistem. Hasil penelitian adalah terbentuknya suatu sistem, aplikasi berdasarkan konsep air terjun (*waterfall*) untuk meningkatkan kinerja administrasi pusat kesehatan hewan Kabupaten Bangka yang lebih tertata dalam pengelolaannya.

Kusumastuti, dkk (2017) melakukan penelitian dengan judul “Sistem Informasi Pusat Kesehatan Hewan Kota Cimahi”. Pusat kesehatan hewan (Puskeswan) Kota Cimahi merupakan lembaga kesehatan yang berfokus pada bidang pelayanan kesehatan hewan. Tidak adanya informasi secara *real time* mengenai ketersediaan stok obat sehingga berakibat pada keterlambatan dalam penanganan pasien dan menyebabkan pelayanan yang diberikan oleh Pusat Kesehatan Hewan Kota Cimahi menjadi kurang optimal, selain itu kurangnya informasi mengenai tenaga medik hewan di Kota Cimahi menjadi salah satu masalah yang terjadi. Metodologi yang digunakan dalam penelitian ini adalah *waterfall* yang terdiri dari tahapan identifikasi kebutuhan, desain, implementasi, pengujian dan dokumentasi. Dengan adanya sistem informasi pusat kesehatan hewan Kota Cimahi diharapkan dapat mengintegrasikan data pasien, data rekam medik dan data obat yang ada di pusat

kesehatan hewan sehingga dapat memberikan notifikasi ketersediaan obat secara *real time* dan *website* yang membantu dalam memberikan informasi mengenai tenaga medik hewan di Kota Cimahi.

Ozzy Guardiola Fittifaldy (2017) dengan judul “Sistem Informasi Pelayanan Kesehatan Hewan Pada UPT Klinik Hewan Bandung Berbasis *Web*”. Sistem pelayanan kesehatan yang sedang berjalan pada UPT Klinik Hewan Bandung selama ini masih berjalan manual, diantaranya yaitu saat pendaftaran hewan masih dilakukan dengan pencatatan nama hewan, jenis hewan, umur hewan, dan keluhan penyakit masih ditulis dalam selembar kertas. Begitu juga pada bagian pemeriksaan, dokter harus mencatat data penyakit hewan pada file kertas rekam medik kemudian disimpan dan diurutkan sesuai nama hewan dan jenis hewan yang sudah datang sebelumnya, sedangkan proses penghitungan jumlah ketersediaan obat, obat masuk, dan obat keluar dihitung satu persatu dan dicatat dengan menuliskan daftar obat pada tabel yang telah disediakan, dari uraian diatas diperlukan suatu sistem terkomputerisasi yang lebih efektif untuk menunjang pelayanan kepada masyarakat yang lebih baik, dan untuk mempermudah pekerjaan pegawai dalam melakukan pekerjaannya.

Suhartanto (2016) melakukan penelitian dengan judul “Pembuatan *Website* UPT Puskesmas Delanggu Menggunakan Php Dan MySQL”. Pada penelitian tersebut, Suhartanto menggunakan php dan database MySQL yang akan menggantikan sistem lama yaitu secara manual. Karena sistem ini bertujuan sebagai sumber informasi UPT Puskesmas Delanggu, maka sistem ini dibuat secara sederhana dan mudah digunakan. Sistem ini memiliki fasilitas-fasilitas antara lain adalah home, profil, pelayanan, dan berita.

Tinjauan pustaka pada penelitian ini mengumpulkan informasi tentang penelitian-penelitian terdahulu untuk mendapatkan pembandingan dan ide atau gagasan yang nantinya dapat diterapkan dalam penelitian ini, dapat dicermati bahwa pada perkembangan teknologi dalam implementasinya pembuatan sistem informasi sangat dibutuhkan dari berbagai sektor, dari beberapa metode yang digunakan mulai dari analisa sampai implementasi dalam sistem informasi berupa *website* yang dapat membantu dalam proses publikasi informasi maupun layanan admin-

istrasi berupa informasi publik yang diterapkan pada penelitian “Analisis dan Perancangan Sistem Informasi Gerakan Terpadu Pusat Kesehatan Hewan Kabupaten Bangka Berbasis *Website*”. Pada penelitian lain menunjukkan bahwa *website* dapat membantu proses pelayanan informasi secara *real time* dan *website* dapat membantu dalam memberikan informasi mengenai tenaga medik hewan, meminimalisir terjadinya *redundancy* yang dibahas detail pada penelitian judul “Sistem Informasi Pusat Kesehatan Hewan Kota Cimahi”. Kemudian pada penelitian dengan judul “Sistem Informasi Pelayanan Kesehatan Hewan Pada UPT Klinik Hewan Bandung Berbasis *Web*” menunjukkan *website* dapat memberikan optimalisasi pelayanan kepada masyarakat yang lebih baik, disamping sangat berguna untuk mempermudah pekerjaan pegawai dalam melakukan pekerjaannya. Sehingga dapat disimpulkan dalam penelitian ini mengambil topik bahasan dalam merancang dan implementasi *Website* UPT Puskesmas Dinas Perikanan dan Peternakan Kabupaten Karanganyar menggunakan metode *User Experience* dan *Heuristic Evaluation* dalam proses pengujiannya.

## 2.2. Kerangka Pemikiran

Penelitian yang dilakukan pada perancangan dan implementasi *website* UPT Puskesmas Kabupaten Karanganyar dibutuhkan beberapa tahapan yang harus dilalui untuk dapat menghasilkan *website* yang sesuai dengan kebutuhan pengguna. Kerangka pemikiran seperti yang terlihat pada Gambar 2.1.

### 1. Latar Belakang

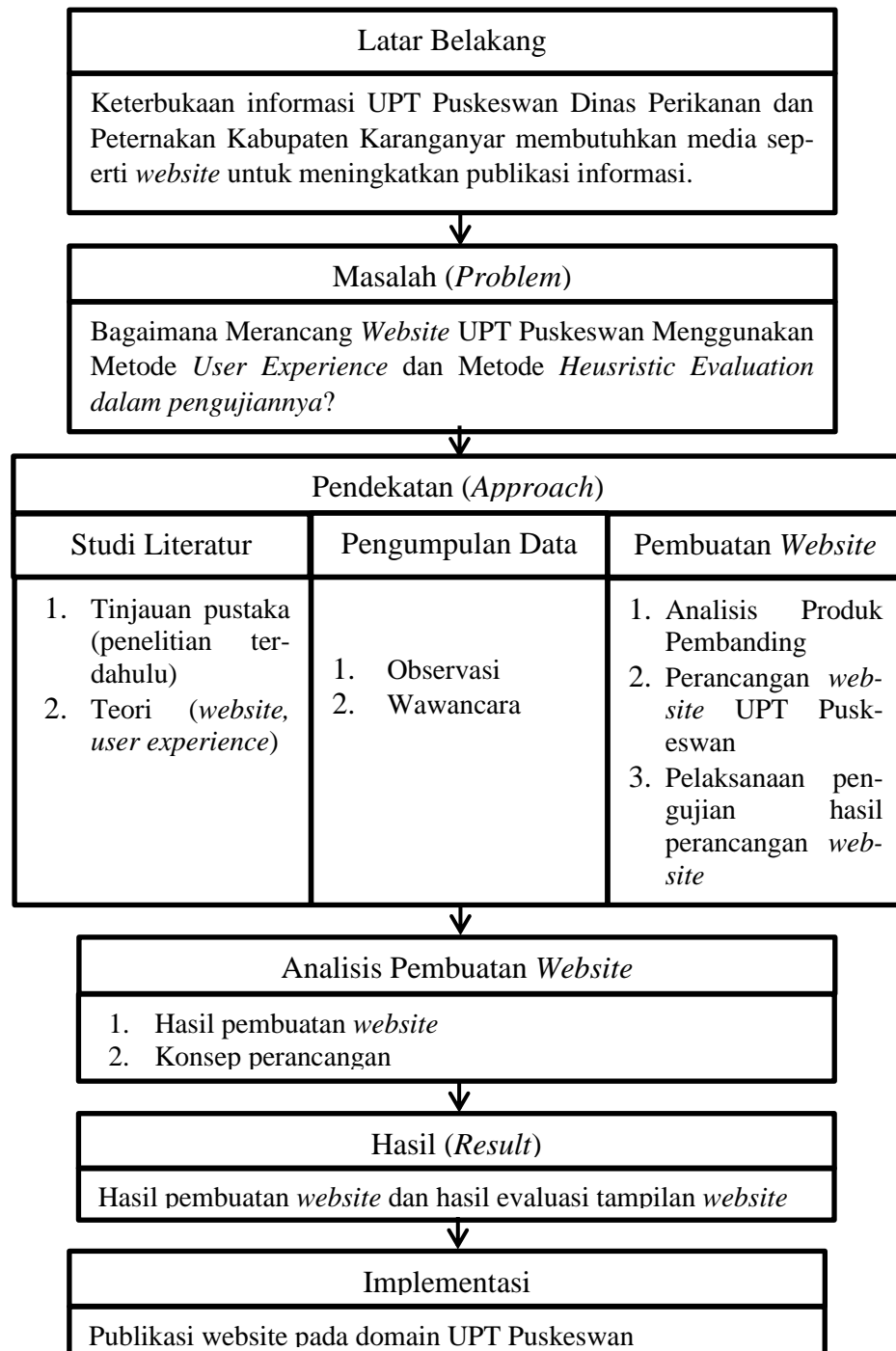
Pokok permasalahan yang mendasari perlu dilakukan perancangan *user interface website* UPT Puskesmas Dinas Perikanan dan Peternakan Kabupaten Karanganyar.

### 2. Masalah

Masalah yang akan diselesaikan dalam penelitian adalah Bagaimana Merancang *Website* UPT Puskesmas Menggunakan Metode *User Experience* dan *Heuristic Evaluation* dalam pengujiannya?

### 3. Pendekatan

Pendekatan penelitian terdiri dari studi literatur tentang penelitian terdahulu dan kajian teori tentang perancangan *user experience*. Pengumpulan data penelitian dilakukan melalui observasi lapangan pada UPT Puskesmas.



Gambar 2.1. Kerangka Pemikiran

#### 4. Analisis Pembuatan *Website*

Analisa dan proses perancangan pada pembuatan *website* UPT Puskesmas, dimulai dengan melakukan konsep perancangan untuk *website* yang telah dibuat sampai membuat hasil pengujian *website*.

#### 5. Hasil Penelitian

Hasil penelitian merupakan hasil dari analisis pengujian. Hasil penelitian dan hasil evaluasi tampilan (*user experience*) selanjutnya dapat digunakan untuk pengembangan *website* UPT Puskesmas.

#### 6. Implementasi

Publikasi *website* pada domain UPT Puskesmas Dinas Perikanan dan Peternakan Kabupaten Karanganyar.

### 2.3. Teori-Teori Pendukung

Penyusunan Tugas Akhir memerlukan suatu referensi pendukung yang digunakan sebagai landasan teori agar penelitian dapat berjalan dengan benar dan tidak menyimpang dari kaedah ilmu pengetahuan yang ada. Landasan teori diperoleh dari berbagai sumber dan literatur yang mempublikasikan pendapat beberapa ilmuwan yang digunakan sebagai pendukung pembahasan masalah dalam penelitian Tugas Akhir. Berikut ini beberapa diantaranya:

#### 2.3.1. *Website*

*Web* dapat diartikan sebagai sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa *text*, gambar, *video*, *audio*, dan animasi lainnya yang disediakan melalui jalur koneksi *internet*. *Website* atau sering disebut situs merupakan kumpulan halaman *web* yang dijalankan dari suatu alamat *web domain* (Abdullah, 2015).

*Website* merupakan sekumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman (Bekti, 2015:1).

### 2.3.2. *User Interface*

Antar muka (*User Interface*) sangat diperlukan dalam suatu pembuatan sistem. Dengan adanya antar muka, diharapkan pengguna akan mudah memahami apa yang harus dilakukan dan menghindari terjadinya kebingungan saat menggunakan sistem. Desain *user interface* adalah sebuah media yang menciptakan komunikasi yang efektif antara manusia dengan komputer (Abdullah, 2015).

Perencanaan *user interface* memiliki beberapa hal yang perlu diperhatikan dalam membuat desain antarmuka yang antara lain sebagai berikut:

1. Konsisten

Konsisten mencakup tata letak menu, jenis huruf, warna tulisan, bentuk *prompt* yang digunakan, layar bantuan, dan sebagainya.

2. *Shortcut* atau pintasan untuk pengguna

Menyediakan fitur *shortcut* bagi pengguna yang sudah ahli untuk meningkatkan kecepatan interaksi dan menghindari kejenuhan dengan cara mengurangi prosedur yang harus dilalui ketika melakukan suatu tugas.

3. Umpan Balik yang informatif

Pengguna perlu untuk melihat konsekuensi dari aksi yang mereka lakukan. Jika seorang pengguna memasukkan sebuah perintah atau melakukan sebuah aksi dan komputer tidak menunjukkan apakah perintah atau aksi tersebut diproses atau tidak, hal ini akan membuat pengguna kebingungan.

4. Desain dialog untuk menghasilkan penutupan

Berinteraksi dengan sebuah komputer seperti sebuah percakapan. Setiap tugas harus memiliki awal, tengah, dan akhir.

5. Memberikan penawaran penanganan kesalahan yang sederhana

Perlu adanya suatu mekanisme penanganan kesalahan yang sederhana dan mudah dimengerti oleh pengguna ketika pengguna melakukan sebuah kesalahan, sehingga pengguna dapat memperbaikinya dengan dipandu oleh sistem.

6. Mudah kembali ke tindakan sebelumnya

Adanya fitur pembalikan aksi yang dapat membuat pengguna menjadi lebih cemas karena pengguna mengetahui bahwa setiap kesalahan yang terjadi sebagai akibat dari aksi yang bisa dibalikan ke keadaan sebelum aksi tersebut dilakukan.

7. Mendukung tempat pengendali internal

Hal ini diperlukan untuk meningkatkan kepuasan dari pengguna dimana kepuasan pengguna akan meningkat jika pengguna memiliki perasaan bahwa pengguna yang memegang kendali atas komputer dan bukan sebaliknya.

8. Mengurangi beban memori jangka pendek

Keterbatasan ingatan manusia membutuhkan tampilan yang sederhana atau banyak tampilan halaman yang sebaiknya disatukan, serta diberikan cukup waktu pelatihan untuk kode dan urutan tindakan.

### **2.3.3. *User Experience* (Pengalaman Pengguna)**

UX (*User experience*) merupakan suatu metode yang memperhatikan masukan dan saran dari pengalaman pengguna dalam merancang *website*. Perancangan bukan menurut pembuat yang beranggapan bagus melainkan orang lain sebagai pengguna. *User experience* secara umum dipahami sebagai keadaan yang selalu berubah, mengingat keadaan internal dan emosional seseorang selalu berubah selama menggunakan produk dan setelah menggunakan produk (Farida, L. D. 2016)

UX (*User experience*) adalah semua tentang pengguna termasuk di dalamnya adalah emosi, kepercayaan, preferensi, persepsi, tanggapan fisik, perilaku yang terjadi sebelum penggunaan, selama penggunaan, dan setelah penggunaan. *User Experience* adalah totalitas/keseluruhan efek yang dirasakan pengguna sebagai hasil interaksi dan konteks penggunaan dari sebuah sistem, *device*, atau produk, termasuk pengaruh dari *usability*, *usefulness*, dan dampak emosional selama interaksi berlangsung (Farida, L. D. 2016).

*User Experience* memperhatikan empat poin utama, yaitu:

1. *Usability*

Komponen pragmatis (praktis dan berguna bagi umum) dari UX, termasuk efektivitas, efisiensi, produktivitas, kemudahan dalam menggunakan, kemudahan untuk mempelajari, ketahanan, dan aspek dari kepuasan pengguna.

2. *Usefulness*

Komponen UX dimana fungsionalitas sistem memberikan kemampuan untuk menyelesaikan tujuan dari suatu pekerjaan/permainan.

3. *Functionality*

Kekuatan untuk melakukan pekerjaan/permainan tersebut dengan fitur-fitur dan kapabilitas tanpa memperhatikan antarmuka pengguna.

4. *Emotional Impact*

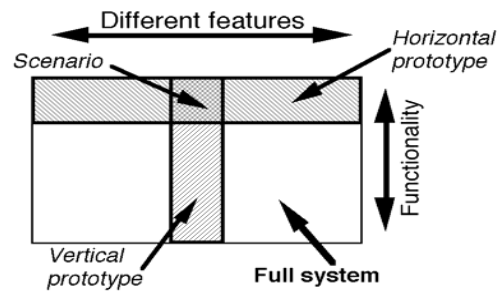
Komponen efektif dari UX yang mempengaruhi perasaan pengguna. Termasuk di dalamnya efek kepuasan, kesenangan, penggunaan yang menyenangkan, estetika, disukai, sesuatu yang baru, originalitas, sensasi, keren, menjanjikan, menarik, dan dapat melibatkan faktor-faktor emosi yang lebih mendalam seperti ekspresi diri, identitas, perasaan kontribusi untuk dunia, dan kebanggaan memiliki.

#### **2.3.4. *Prototype***

Suatu gambaran dari sebuah sistem yang menunjukkan ide dari pengembang tentang bagaimana sistem akan berjalan apabila telah dibuat dalam bentuk yang telah selesai. Proses pembuatan *prototype* disebut *prototyping*. Dasar dari pemikiran pembuatan *prototype* adalah membuat *prototype* secepat mungkin, bahkan dalam waktu semalam, lalu memperoleh umpan balik dari pengguna yang akan memungkinkan *prototype* tersebut diperbaiki kembali dengan sangat cepat. *Prototype* menghasilkan versi awal sebuah produk atau sistem yang dapat di konstruksi ulang dengan cepat dan murah, sesuatu yang dapat mewakili sistem jadi (Farida, L. D. 2016).



Berdasarkan fitur dan fungsinya *prototype* dibagi menjadi empat jenis, yaitu *horizontal prototype*, *vertical prototype*, *T prototype* dan *local prototype*. Ilustrasi jenis *prototype* terhadap *full system* seperti pada Gambar 2.2.



Gambar 2.2. Ilustrasi Jenis *Prototype* Terhadap *Full System* (Sumber:

<https://www.umsl.edu/>)

#### 1. *Horizontal Prototype*

Mencakup seluruh fitur antarmuka dari sistem tetapi memiliki fungsionalitas yang rendah atau tanpa fungsi pokok, biasanya adalah berupa simulasi dan belum dapat digunakan untuk melakukan pekerjaan sesungguhnya.

#### 2. *Vertical Prototype*

Fokus utama terletak pada fungsionalitas yang detail pada salah satu atau beberapa fitur antarmuka. Fitur antarmuka yang disimulasikan lebih sedikit, tetapi lebih rinci.

#### 3. *T Prototype*

Penggabungan keunggulan yang terdapat pada *horizontal prototype* dan *vertical prototype*. *Prototype* yang dibuat memiliki fitur antarmuka yang lengkap (*horizontal prototype*) dan simulasi yang rinci pada salah satu fungsi (*vertical prototype*).

#### 4. *Local Prototype*

*Prototype* yang dibuat memiliki salah satu fitur antarmuka dan fungsionalitas yang disimulasikan juga terbatas/tidak rinci. *Local prototype* biasanya digunakan untuk melakukan evaluasi terhadap alternatif desain, contohnya adalah pada ikon atau huruf yang akan digunakan.

### 2.3.5. HTML (*Hypertext Markup Language*)

Sebuah bahasa markah untuk membuat halaman *web* dan bahasa yang digunakannya masih sangat standar seperti salah satu fungsinya untuk membuat tabel, menambahkan objek suara, video dan animasi adalah pengertian dari HTML. "*Hypertext Markup Language*" atau HTML adalah bahasa yang digunakan pada dokumen sebagai bahasa untuk pertukaran dokumen *web*. Pengertian diatas dapat disimpulkan bahwa HTML adalah sebuah dokumen yang berisikan tag atau beberapa elemen dan atribut untuk menampilkan halaman pada *web browser* (Hidayatullah & Kawistara, 2017).

### 2.3.6. Javascript

Bahasa pemrograman yang berbentuk kumpulan skrip yang memiliki fungsi untuk memberikan tampilan agar tampak lebih interaktif pada dokumen web adalah pengertian dari *Javascript*. "*Javascript*" adalah suatu bahasa pemrograman yang dikembangkan untuk dapat berjalan pada *web browser*. Pengertian diatas dapat disimpulkan bahwa *Javascript* adalah bahasa pemrograman untuk memberikan kemampuan tambahan ke dalam bahasa pemrograman HTML (Hidayatullah & Kawistara, 2017).

### 2.3.7. CSS (*Cascading Style Sheet*)

Sebuah *website* bisa terdiri dari berpuluh-puluh bahkan bisa beratus-ratus halaman. Jika setiap kita ingin merubah halaman *web* tersebut kita harus mengubah formatnya satu per satu maka akan membuat kita repot. *Cascading Style Sheet* atau yang biasa kita sebut dengan CSS sangat tepat untuk menyimpan format dan menggunakannya kapanpun kita inginkan. CSS atau yang memiliki kepanjangan *Cascading Style Sheet*, merupakan suatu bahasa pemrograman *web* yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web, sehingga tampilan *web* akan lebih rapih, terstruktur, dan seragam. Suatu teknologi yang digunakan untuk mempermudah pengguna dalam mengubah secara keseluruhan warna, tampilan dan dikembangkan untuk menata gaya pengaturan halaman *web* biasa kita kenal dengan sebutan CSS. Pengertian diatas dapat disimpulkan bahwa CSS (*Cascading Style Sheet*) adalah bahasa pem-

rograman yang dikembangkan untuk mempercantik halaman *web* (Hidayatullah & Kawistara, 2017).

### **2.3.8. Basis Data**

Himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah merupakan definisi dari Basis Data. Secara lebih lengkap pemanfaatan Basis Data dilakukan untuk memenuhi tujuan kecepatan, kemudahan, efisiensi ruang penyimpanan, keakuratan, ketersediaan, kelengkapan, keamanan, dan pemakaian bersama. “Basis Data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”. Jadi, Basis Data adalah kumpulan data yang saling berhubungan agar dapat diakses oleh *software* tertentu (Hidayatullah & Kawistara, 2017).

### **2.3.9. MySQL (*My Structure Query Language*)**

Menurut Hidayatullah & Kawistara (2017:175) “MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi *web*”. Sedangkan aplikasi perangkat lunak yang bertugas untuk menjalankan fungsi pengolahan data disebut MySQL. Pertama MySQL dikembangkan oleh MySQL AB yang kemudian diakuisisi Sun Microsystem dan terakhir MySQL dikelola oleh Oracle Corporation. Dapat disimpulkan bahwa MySQL adalah Perangkat lunak atau software yang mengelola SQL.

### **2.3.10. PHP (*Hypertext Preprocessor*)**

Hidayatullah & Kawistara (2017:223) dalam bukunya yang berjudul “Pemrograman Web”. Mengemukakan bahwa “*PHP Hypertext Preprocessor*” adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*”. PHP memiliki sifat *server side scripting* sehingga untuk menjalankan PHP harus menggunakan *web server side*.



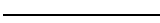

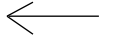
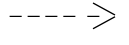
### 2.3.11. UML (*Unified Modelling Language*)

(Rosa & Shalahuddin, 2015) dijelaskan pada perkembangan teknologi perangkat lunak diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Pada perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak, yaitu UML. Untuk membangun *website* UPT Puskesmas Dinas Perikanan dan Peternakan Kabupaten Karanganyar peneliti menggunakan 4 macam diagram UML, yaitu:

#### 1. *Use Case Diagram*

*Use Case* atau diagram *Use Case* merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol *Use Case* diagram dapat dilihat pada Tabel 2.1.



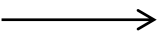
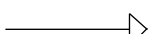
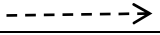

Tabel 2.1 Simbol-simbol *Use Case Diagram*

NO:	SIMBOL	NAMA	KETERANGAN
1.		<i>Actor</i>	Menspesifikasi himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Use Case</i>	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan kata kerja diawal frase nama <i>use case</i> .
3.		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		<i>Extend</i>	Menspesifikasi bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
5.		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum – khusus) 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> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

## 2. Class Diagram

Diagram kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi. Kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas (Rosa & Shalahuddin, 2015). Simbol *Class Diagram* dapat dilihat pada Tabel 2.2.



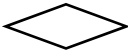


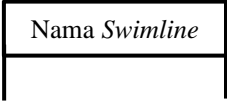
Tabel 2.2 Simbol-simbol *Class Diagram*

NO:	SIMBOL	NAMA	KETERANGAN
1.	<div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">Nama kelas</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">+ Atribut</div> <div style="border: 1px solid black; padding: 2px;">+ Operasi()</div>	<i>Class</i>	Kelas pada struktur system
2.		<i>Interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.		<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain biasanya juga disertai dengan <i>multiplicity</i> .
5.		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisai (umum-khusus).
6.		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.		<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> ).

## 3. Activity Diagram

*Activity Diagram* menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah diagram aktivitas menggambarkan tentang aktivitas yang dilakukan oleh sistem dan bukan yang dilakukan aktor (Rosa & Shalahuddin, 2015). Simbol *Activity Diagram* ditujukan pada Tabel 2.3.




Tabel 2.3 Simbol-simbol *Activity Diagram*

NO:	GAMBAR	NAMA	KETERANGAN
1.		Status awal	Status awal aktivitas sistem.
2.		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Status akhir	Status akhir yang dilakukan sistem.
6.		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

#### 4. *Sequence Diagram*

*Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek (Rosa & Shalahuddin, 2015). Simbol *sequence diagram* dapat dilihat pada Tabel 2.4.


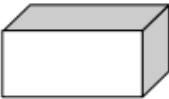


Tabel 2.4. Simbol-simbol *Sequence Diagram*

NO:	SIMBOL	NAMA	KETERANGAN
1.		Aktor	Menggambarkan orang yang berinteraksi dengan sistem.
2.		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
3.		Objek	Menyatakan objek yang berinteraksi pesan.
4.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi.
5.		Pesan tipe create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.		Pesan tipe call	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.

## 5. Deployment Diagram

*Deployment Diagram* atau *physical Diagram* menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik (Rosa & Shalahuddin, 2015). *Deployment Diagram* atau *physical Diagram* dapat dilihat pada Tabel 2.5.


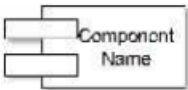

Tabel 2.5. Simbol-simbol *Deployment Diagram*

NO:	SIMBOL	NAMA	KETERANGAN
1.		<i>Package</i>	Merupakan sebuah bungkus dari satu atau lebih mode.
2.		<i>Node</i>	Biasanya mengacu pada perangkat keras ( <i>hardware</i> ), perangkat lunak yang tidak dibuat sendiri ( <i>software</i> ), jika didalam node disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikut sertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.
3.		<i>Kebergantungan/dependency</i>	Kebergantungan antar node, arah panah mengarah pada node yang dipakai.
4.		<i>Link</i>	Relasi antar node

## 6. Component Diagram

*Component Diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain (Rosa & Shalahuddin, 2015). *Component Diagram* dapat dilihat pada Tabel 2.6.

Tabel 2.6. Simbol-simbol *Component Diagram*

NO:	SIMBOL	NAMA	KETERANGAN
1.		<i>Package</i>	Merupakan sebuah bungkus dari satu atau lebih mode.
2.		<i>Komponen</i>	Komponen system
3.		Kebergantungan/ <i>dependency</i>	Kebergantungan antar node, arah panah mengarah pada node yang dipakai.
4.	(Antarmuka/ <i>interface</i> )	Antarmuka/ <i>interface</i>	Sama dengan konsep interface pada pemrograman berorientasi objek, yaitu sebagai antarmuka pada setiap komponen.

### 2.3.12. Heuristic Evaluation

*Heuristic evaluation* itu sendiri merupakan panduan, prinsip umum, atau aturan yang dapat menuntun keputusan rancangan atau digunakan untuk mengkritik suatu keputusan yang sudah diambil. Evaluasi Heuristik diusulkan oleh Nielsen dan Molich, hampir sama dengan *Cognitive Walkthrough* tetapi sedikit terstruktur dan sedikit terarah. Pada pendekatan ini sekumpulan kriteria *usability* atau *heuristic* diidentifikasi dan perancangan dilaksanakan misalnya dimana kriteria dilanggar (Savitri & Ispani, 2015).

Tujuan dari evaluasi heuristik adalah untuk memperbaiki perancangan secara efektif. Evaluator melakukan evaluasi melalui kinerja dari serangkaian tugas dengan perancangan dan dilihat kesesuaiannya dengan kriteria setiap tingkat. Jika ada kesalahan terdeteksi maka perancangan dapat ditinjau ulang untuk memperbaiki masalah ini sebelum tingkat implementasi (Savitri & Ispani, 2015).



Evaluasi Heuristik sangat baik digunakan sebagai teknik evaluasi desain, karena lebih mudah untuk menemukan atau menentukan masalah *usability* yang muncul. Untuk menggunakan evaluasi ini dibutuhkan *software* yang akan diteliti atau *storyboard* untuk sistem yang akan dibuat (Savitri & Ispani, 2015).

Setelah analisis terhadap objek sesuai dengan prinsip metode yang digunakan telah selesai dilakukan, maka tahap selanjutnya dibuat laporan mengenai pembahasan atas hasil analisis yang telah didapat serta kesimpulan dan saran dari analisis tersebut. Berikut ini adalah 10 prinsip metode *heuristic evaluation* yang digunakan beserta keterangan-keterangan mengenai setiap prinsip-prinsip dari *heuristic evaluation*.

- 1) *Visibility of system status (feedback)*, terdapat informasi terhadap status sistem saat ini dalam bentuk pesan dan waktu yang sesuai.
- 2) *Match between system and the real world*, bahasa sistem sesuai dengan bahasa yang digunakan pengguna.
- 3) *Use Control and Freedom*, pengguna mempunyai kendali dan kebebasan dalam mengambil suatu keputusan.
- 4) *Consistency and Standards*, standar yang digunakan sistem harus konsisten.
- 5) *Error Prevention*, terdapat sistem yang mencegah terjadinya kesalahan.
- 6) *Recognition Rather than Recall*, membantu pengguna dalam mengenali, mendiagnosa, dan mengatasi masalah.
- 7) *Flexibility and Efficient of Use*, sistem dapat mengakomodasikan pengguna yang ahli dan pengguna pemula.
- 8) *Aesthetic and Minimalist Design*, memberikan informasi yang relevan serta tampilan yang sesuai dengan kebutuhan sistem.
- 9) *Help users recognize, dialogue, and recovers from errors*, memberikan bantuan bagi pengguna dalam mengenal, berdialog, serta memperbaiki kesalahan.
- 10) *Help and Documentation*, sistem memiliki dokumentasi yang relevan serta fitur *help* yang baik.