BAB I
TINJAUAN PUSTAKA

I.1 Tinjauan Tempat Penelitian

I.1.1 Sejarah Instansi

Sebelum Kabupaten Bandung berdiri, daerah Bandung dikenal dengan sebutan "Tatar Ukur". Pada masa pemerintahan Dipati Ukur, Tatar Ukur merupakan suatu wilayah yang cukup luas, mencakup sebagian besar wilayah Jawa Barat, terdiri atas sembilan daerah yang disebut "Ukur Sasanga". Setelah Kerajaan Sunda-Pajajaran runtuh (1579/1580) akibat gerakan Pasukan Banten dalam usaha menyebarkan agama Islam di daerah Jawa Barat, Tatar Ukur menjadi wilayah kekuasaan Kerajaan Sumedanglarang, penerus Kerajaan Pajajaran.

Ketika Kerajaan Sumedang Larang diperintah oleh Raden Suriadiwangsa, anak tiri Geusan Ulun dari Ratu Harisbaya, Sumedanglarang menjadi daerah kekuasaan Mataram sejak tahun 1620. Sejak itu status Sumedanglarang pun berubah dari kerajaan menjadi Kabupaten dengan nama Kabupaten Sumedang. Mataram menjadikan Priangan sebagai daerah pertahanannya di bagian Barat terhadap kemungkinan serangan Pasukan Banten dan atau Kompeni yang berkedudukan di Batavia, karena Mataram di bawah pemerintahan Sultan Agung (1613-1645) bermusuhan dengan Kompeni dan konflik dengan Kesultanan Banten.

Setelah "pemberontakan" Dipati Ukur dianggap berakhir pada tahun 1623, Sultan Agung menyerahkan kembali jabatan Bupati Wedana Priangan kepada Pangeran Dipati Rangga Gede yang telah bebas dari hukumannya. Selain itu juga dilakukan reorganisasi pemerintahan di Priangan untuk menstabilkan situasi dan kondisi daerah tersebut. Daerah Priangan di luar Sumedang dan Galuh dibagi menjadi tiga Kabupaten, yaitu Kabupaten Bandung, Kabupaten Parakanmuncang dan Kabupaten Sukapura dengan cara mengangkat tiga kepala daerah dari Priangan yang dianggap telah berjasa menumpas pemberontakan Dipati Ukur.

Kabupaten Bandung berada di bawah pengaruh Mataram sampai akhir tahun 1677. Kemudian Kabupaten Bandung jatuh ke tangan Kompeni. Hal itu terjadi akibat perjanjian Mataram - Kompeni (perjanjian pertama) tanggal 19-20

Oktober 1677. Di bawah kekuasaan Kompeni (1677-1799), Bupati Bandung dan Bupati lainnya di Priangan tetap berkedudukan sebagai penguasa tertinggi di Kabupaten, tanpa ikatan birokrasi dengan Kompeni. Sistem pemerintahan Kabupaten pada dasarnya tidak mengalami perubahan, karena Kompeni hanya menuntut agar bupati mengakui kekuasaan Kompeni, dengan jaminan menjual hasil-hasil bumi tertentu kepada VOC.

Hingga berakhirnya kekuasaan Kompeni - VOC akhir tahun 1779, Kabupaten Bandung beribukota di Krapyak. Selama itu Kabupaten Bandung diperintah secara turun-temurun oleh enam orang bupati. Tumenggung Wiraangunangun (merupakan bupati pertama) angkatan Mataram yang memerintah sampai tahun 1681. Lima bupati lainnya adalah bupati angkatan Kompeni yakni Tumenggung Ardikusumah yang memerintah tahun 1681-1704, Tumenggung Anggadireja I (1704-1747), Tumenggung Anggadireja II (1747-1763), R. Anggadireja III dengan gelar R.A. Wiranatakusumah I (1763-1794) dan R.A. Wiranatakusumah II yang memerintah dari tahun 1794 hingga tahun 1829. Pada masa pemerintahan Bupati R.A. Wiranatakusumah II, ibukota Kabupaten Bandung dipindahkan dari Karapyak ke Kota Bandung. [11]

Instansi pada Pemerintah Kabupaten Bandung yang bertanggung jawab untuk melakukan pengelolaan informasi adalah Badan Perpustakaan, Arsip dan Pengelolaan Informasi yang disingkat BAPAPSI. *Website* Pemerintah Kabupaten Bandung merupakan bagian dari tanggung jawab BAPAPSI.

BAPAPSI dibentuk berdasarkan Peraturan Daerah Kabupaten Bandung No. 21 tahun 2007 tentang Pembentukan Organisasi Lembaga Teknis Daerah Kabupaten Bandung Terhitung 1 April 2008. Berdasarkan Perda tersebut Kantor Perpustakaan Daerah, Kantor Arsip Daerah, Badan Pengembangan Informasi Daerah, dan Kantor Pengolahan Data Elektronik dimerger menjadi SKPD baru dengan nama BAPAPSI (Badan Perpustakaan, Arsip dan Pengembangan Sistem Informasi).

Diharapkan dengan penggabungan ini, BAPAPSI dapat menjadi satu lembaga yang kuat dan dapat diandalkan dalam penyediaan informasi yang berkualitas baik untuk kebutuhan internal maupun eksternal melalui optimalisasi

pelayanan perpustakaan, kearsipan dan pelayanan berbasis multimedia (media cetak dan media elektronik).

I.1.2 Logo Instansi

Logo pemerintah Kabupaten Bandung dapat dilihat pada Gambar I.1.



Gambar I.1 Logo Pemerintah Kabupaten Bandung

Lambang berbentuk perisai terbagi atas empat bagian, yaitu [12]:

1. Bagian kanan atas berlatar kuning emas, dengan gambar gunung berwarna hijau, melambangkan bahwa Kabupaten Bandung termasyhur karena tanahnya yang subur di daerah bergunung-gunung, dan sebagai ciri memiliki gunung Tangkuban Perahu yang sangat terkenal dengan legenda Sangkuriang.
2. Bagian melintang bergerigi merupakan bentuk bendungan kokoh kuat berwarna hitam. Melambangkan masyarakat Kabupaten Bandung memiliki pendirian yang kokoh dan kuat, baik secara fisik dalam membendung hawa nafsu.
3. Pohon kina berwarna hijau dan berlatar belakang merah, melambangkan di Kabupaten Bandung kaya akan air, baik air maupun air danau. Kabupaten Bandung di lintasi oleh sungai Citarum, sungai Cikapundung, dab sungai-sungai kecil lainnya. Kabupaten Bandung danau/situ Patengang, Situ Cileunca, Situ Lembang, Situ Ciburuy, dan danau-danau lainnya.
4. Dibawah perisai tertulis dalam pita kuning "REPEH RAPIH KERTARAHARJA" yang mempunyai arti sebagai berikut
    a. REPEH : Suasana hidup yang aman dan tenteram.

b. RAPIH : Suasana kehidupan yang rukun dan tertib dalam lingkungan yang bersih, sehat dan asri.
c. KERTARAHARJA : Tatanan kehidupan yang sejahtera lahir dan batin secara seimbang serasi adil dan merata.

I.1.3 Visi dan Misi

Visi dari Pemerintah Daerah Kabupaten Bandung adalah "Terwujudnya Kabupaten Bandung yang Maju, Mandiri dan Berdaya Saing, melalui Tata Kelola Pemerintahan yang Baik dan Pemantapan Pembangunan Perdesaan, Berlandaskan Religius, Kultural dan Berwawasan Lingkungan".

Untuk mewujudkan Visi Pemerintah Daerah Kabupaten Bandung, maka harus ditetapkan juga Misi yang harus mendapatkan perhatian seksama di mana tugas yang diemban oleh Pemerintah Kabupaten Bandung adalah:

1. Meningkatkan profesionalisme birokrasi.
2. Meningkatkan kualitas SDM (pendidikan dan kesehatan) yang berlandaskan Iman dan takwa serta melestarikan budaya Sunda.
3. Memantapkan pembangunan perdesaan.
4. Meningkatkan keamanan dan ketertiban wilayah.
5. Meningkatkan ketersediaan infrastruktur dan keterpaduan tata ruang wilayah.
6. Meningkatkan ekonomi kerakyatan yang berdaya saing.
7. Memulihkan keseimbangan lingkungan dan menerapkan pembangunan berkelanjutan.

Selain visi dan misi dari Pemerintah Kabupaten Bandung, BAPAPSI sebagai instansi mempunyai visi dan misi khusus yang berusaha dilakukan sebagai tujuan dari instansi.

Visi dari Badan Perpustakaan Arsip dan Pengembangan Sistem Informasi adalah “Terwujudnya pelayanan kepada masyarakat yang optimal melalui informasi yang berkualitas tahun 2015”

Untuk mewujudkan visi dari Badan Perpustakaan Arsip dan Pengembangan Sistem Informasi maka disusun misi. Misi dari BAPAPSI adalah sebagai berikut :

1. Meningkatkan peran perpustakaan sebagai sarana untuk mendukung terwujudnya pelayanan bagi masyarakat.
2. Mengembangkan dan mengoptimalkan peran perpustakaan-perpustakaan binaan, baik di tingkat desa/kelurahan/kecamatan ataupun SKPD di lingkungan Kabupaten Bandung.
3. Meningkatkan aksesibilitas dan partisipasi masyarakat terhadap informasi melalui penyediaan sarana/prasarana yang memadai.
4. Meningkatkan diseminasi informasi publik yang terbuka melalui multimedia dan penguatan kemitraan.
5. Membangun masyarakat yang berbudaya informasi melalui pemberdayaan forum komunikasi masyarakat.
6. Menciptakan koordinasi dan kerja sama yang sinergis dengan berbagai pihak untuk meningkatkan kualitas dan kuantitas informasi sehingga menjadi salah satu sumber informasi utama bagi pimpinan dan publik.
7. Menyelenggarakan sistem pengelolaan kearsipan yang efektif dan efisien.
8. Meningkatkan kualitas SDM aparatur di internal BAPAPSI.

I.1.4 Satuan Kerja Perangkat Daerah

Dalam lingkungan Pemerintah Daerah Kabupaten Bandung, terdapat satuan kerja perangkat daerah yang merupakan perangkat pemerintah sebagai pelaksana fungsi yang harus berkoordinasi untuk penyelenggaraan pemerintah daerah. Berikut merupakan daftar SKPD yang ada di Pemerintah Daerah Kabupaten Bandung.

1. Inspektorat
2. Sekretariat
    a. Sekretariat Daerah
    b. Sekretariat DPRD
3. Dinas
    a. Dinas Bina Marga
    b. Dinas Kependudukan dan Catatan Sipil
    c. Dinas Kesehatan
    d. Dinas Koperasi, UKM, Perindustrian & Perdagangan

e. Dinas Pemuda, Olah Raga & Pariwisata
f. Dinas Pendapatan dan Pengelolaan Keuangan
g. Dinas Pendidikan dan Kebudayaan
h. Dinas Perhubungan
i. Dinas Pertanian
j. Dinas Pertanian, Perkebunan, dan Kehutanan
k. Dinas Perumahan, Penataan Ruang dan Kebersihan
l. Dinas Peternakan & Perikanan
m. Dinas Sosial
n. Dinas Sumber Daya Air, Pertambangan dan Energi
o. Dinas Tenaga Kerja

4. Badan
    a. Badan Perencanaan Pembangunan Daerah
    b. Badan Kepegawaian, Pendidikan dan Pelatihan
    c. Badan Perpustakaan, Arsip dan Pengembangan Sistem Informasi
    d. Badan KB dan Pemberdayaan Perempuan
    e. Badan Pengendalian Lingkungan Hidup
    f. Badan Ketahanan Pangan dan Pelaksana Penyuluhan
    g. Badan Penanaman Modal dan Perijinan
    h. Badan Pemberdayaan Masyarakat dan Pemerintahan Desa
    i. Badan Penanggulangan Bencana Daerah
5. Kantor Kesatuan Bangsa, Politik dan Perlindungan Masyarakat
6. Satpol PP
7. RSUD

I.2 Landasan Teori

I.2.1 E-Government

Istilah *E-Government* mempunyai definisi sebagai memanfaatkan internet dan w*orld-wide-web* untuk menyampaikan informasi pemerintahan dan layanan kepada penduduk berdasarkan laporan evaluasi *e-government* pada anggota-anggota PBB [13]. Berdasarkan buku Wirtz dan Daiser, ada 3 faktor kesuksesan *e-*

goverment, kesuksesan yang berpusat pada pengguna, kesuksesan *e-service,* dan kesuksesan IT (*information technology*) [1].

Kesuksesan yang berpusat pada pengguna, terdiri dari 5 aspek yang berhubungan satu sama lain diantaranya adalah :

1. Kemudahan penggunaan (*usability*)
2. Daya guna
3. Desain website atau daya tarik visual.
4. Dukungan dalam penggunaan
5. Penggunaan layanan *e-government* yang *multichannel.*

Kesuksesan *e-service* dibagi menjadi dua bagian, *service-oriented* dan *Function-oriented*. *Service-oriented* tertuju pada pertanyaan layanan apa yang harus diberikan, sedangkan *function-oriented* tertuju pada bagaimana sebuah layanan harus diberikan.

Sama halnya dengan kesuksesan yang berpusat pada pengguna, kesuksesan IT juga terdiri dari 5 aspek, diantaranya :

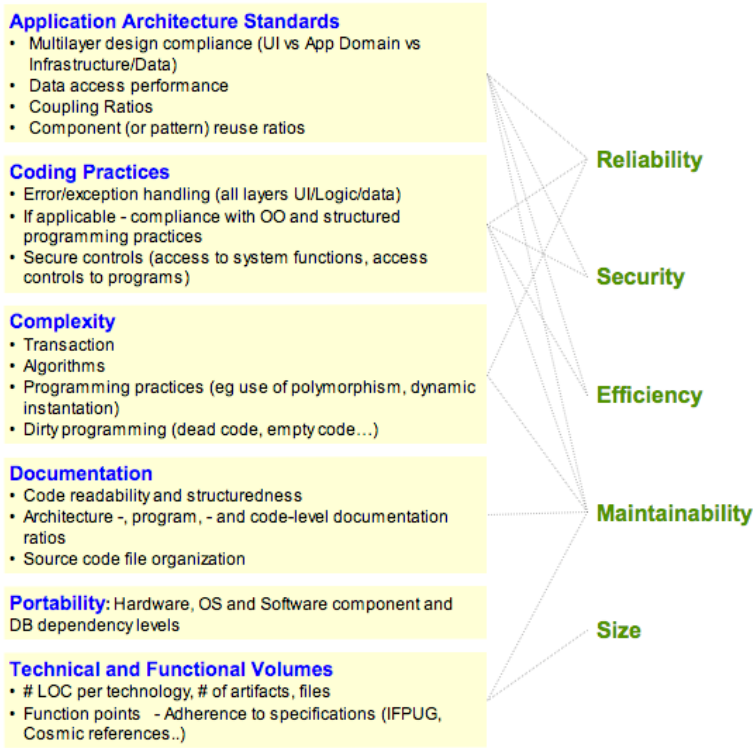
1. Keamanan IT atau privasi
2. Kualitas sistem IT
3. *Responsiveness* dari website dan infrastruktur IT.
4. Kinerja IT atau *reliability*
5. Layanan *multichannel* yang mulus dan tanpa kerusakan media.

I.2.2 Software Quality

Kualitas merupakan konsep yang rumit, tetapi merupakan sesuai yang dapat diukur. Kualitas dapat didefinisikan tergantung pada sudut pandang yang digunakan, misalkan jika menggunakan sudut pandang pengguna, maka kualitas terpenuhi ketika tujuan pengguna dapat diselesaikan dengan baik dan efektif. *Software quality* atau kualitas perangkat lunak adalah sebuah proses dalam perangkat lunak yang diterapkan efektif sehingga menciptakan produk yang bermanfaat dan dapat diukur oleh tim produksi dan penggunanya [14].

Analisis kualitas perangkat lunak juga dapat dinilai dari segi struktural yang didapat dari analisis terhadap kode sumber, skema basis data, dan *framework* [15]. Berdasarkan ISO 9126 terdapat beberapa aspek yang digunakan untuk

mengukur kualitas perangkat lunak, diantaranya adalah *Functionality*, *reliability*, *usability*, *efficiency*, *maintainability*, dan *portability* [14]. Terdapat pula aspek kualitas perangkat lunak berdasarkan OMG atau *Object Management Group*, aspek-aspek tersebut dapat dilihat pada Gambar I.2.



Gambar I.2 Aspek-aspek kualitas perangkat lunak dan hubungan dengan atribut terukur [15]

I.2.3 User Usability

Istilah *usability* diperkenalkan pada awal tahun 1980an yang merupakan pengganti dari istilah *user-friendly* yang merupakan konotasi kabur dan subjektif. Terdapat 3 sudut pandang untuk bagaimana *usability* harus diukur [16]:

1. Sudut pandang berorientasi produk, *usability* dapat diukur untuk istilah-istilah atribut ergonomis dari produk.
2. Sudut pandang berorientasi pengguna, *usability* dapat diukur untuk usaha mental dan prilaku pengguna.
3. Sudut pandang kinerja, *usability* dapat diukur dengan menguji bagaimana pengguna berinteraksi dengan produk dengan beberapa penekanan pada *ease-of-use* (semudah apa produk digunakan), dan *acceptability*

(mempertimbangkan apakah produk akan digunakan dalam kehidupan nyata atau tidak).

Berdasarkan standar ISO 9241 *part* 11 tahun 1998 mengenai *Ergonomics of Human System Interaction*, *usability* didefinisikan sebagai sejauh mana produk [layanan atau lingkungan] dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi dan kepuasan digunakan dalam konteks tertentu [17].

I.2.4 User Interface (UI)

Perancangan *user interface* atau *user interface design* dan *user interface* mempunyai definisi masing-masing. UI *design* merupakan satu bagian dari ilmu interaksi manusia dan komputer (HCI), HCI sendiri merupakan sebuah kajian, perencanaan, dan perancangan terhadap bagaimana manusia dan komputer bekerja bersama untuk mencapai kebutuhan manusia secara efektif. Dan *user interface* atau antarmuka merupakan bagian dari komputer atau perangkat lunak yang dapat dilihat, didengar, dan disentuh [18].

Web *user interface* dibagi dalam 2 kategori, yaitu desain *web page* (halaman web) dan desain *web application* (aplikasi berbasis web). Desain *web page* berpusat pada penempatan konten dan informasi pada antarmuka, sedangkan *web application* berpusat pada antarmuka aplikasi tersebut. Kedua kategori mempunyai ciri yang sama, yaitu kaya akan desain grafis dan informasi [18].

I.2.5 User Experience (UX)

Bidang *user experience* atau yang sering disingkat dengan UX, merupakan bidang yang berhubungan dengan mempelajari, merancang, dan mengevaluasi pengalaman yang didapatkan oleh seseorang ketika menggunakan sebuah sistem. Karena cakupannya yang luas, sehingga definisi UX bergantung pada sudut pandang yang diambil [19].

Dari sudut pandang UX sebagai fenomena, UX merupakan pengalaman yang dirasakan pengguna, yang dipengaruhi oleh pengalaman sebelumnya dan harapan yang ingin dipenuhi berdasarkan pengalaman tersebut, dan berupa memori yang tertanam pada pengguna setelah selesai menggunakan sistem tersebut.

Terdapat beberapa faktor yang mempengaruhi UX dari sudut pandang ini, faktor tersebut dikategorikan ke dalam 3 macam, yaitu [19]

1. Konteks, konteks terdiri dari konteks sosial (tujuan menggunakan sistem untuk mendapat pekerjaan), konteks fisik (menggunakan sistem di dalam mobil yang mengebut), *Task* konteks, dan konteks informasi dan teknologi.
2. Pengguna, UX dinamis karena pengguna yang menggunakan sistem dinamis.
3. Sistem, persepsi pengguna terhadap sistem akan sangat memengaruhi UX.

I.2.6 Hubungan User Experience dan User Interface

User interface UI dan *user experience* UX merupakan dua hal yang berbeda, tetapi keduanya mempunyai hubungan erat satu sama lain. Sebagaimana dipaparkan pada subbab sebelumnya I.2.4 dan I.2.5 definisi keduanya sangat berbeda, UX merupakan bagian dari ilmu perancangan yang berpusat pada manusia, erat kaitannya dengan bagaimana emosi, sikap, dan prilaku seseorang menggunakan sistem tersebut [19]. Sedangkan UI berpusat pada bagaimana fungsi-fungsi yang ada pada sistem akan ditampilkan secara efektif sehingga pengguna dapat melakukan kontrol pada sistem dan lebih mendekati pada rancangan visual [18] [20].

Perancangan user interface pada sebuah sistem merupakan bagian dari bidang user experience. Dan untuk melakukan perancangan user interface membutuhkan analisis user experience, tetapi perancangan UX dapat dilakukan tanpa melibatkan UI [20].

I.2.7 Standar Pengkodean

Standar pengkodean atau *coding standard* merupakan standar yang diterapkan mengenai bagaimana kode ditulis. Standar dibuat dan disepakati dengan tujuan mempermudah kode tersebut dimengerti oleh banyak *programmer* sehingga akan mempermudah proses pengembangan sistem. Tetapi setelah diteliti, standar pengkodean juga berpengaruh pada performa sebuah situs web [5].

Standard pengkodean berbeda-beda tergantung pada bahasa yang digunakan. Untuk website standar yang digunakan untuk bahasa HTML dan CSS

adalah standar dari W3C atau *World Wide Web Consortium*, untuk Zend Framework PHP standar yang digunakan adalah Zend Framework Coding Standard for PHP.

I.2.7.1 Standar Pengkodean HTML

HTML merupakan bahasa *markup* yang digunakan untuk membangun website. Sebagai bahasa yang digunakan untuk membangun website, HTML mempunyai standar penulisan yang harus dipenuhi agar website tersebut berfungsi dengan baik. Standar penulisan kode HTML dikeluarkan oleh *World Wide Web Consortium*, yang merupakan organisasi internasional yang mengembangkan standar web. Seluruh daftar error HTML dapat dilihat pada alamat https://validator.w3.org/docs/errors.html. Berikut merupakan kesalahan yang paling sering ditemukan melalui HTML validator [21].

Tabel I.1 Kesalahan pengkodean HTML

Kesalahan	Perbaikan
HTML dalam elemen <script>	Berdasarkan rekomendasi HTML 4, penulisan *tag* HTML di dalam elemen *script* harus dihindari dan termasuk *error*.
Ampersand dalam URL	Penulisan tanda *ampersand* & dalam Url adalah tidak valid, seperti berikut <a href=”img.cgi?chapter=1&Election=2>. Disrankan menggunakan & untuk mengganti tanda ampersand.
Kesalahan *nesting*	Kesalahan dalam menutup tag bersarang seperti berikut Judul
Menggunakan *lowercase* DOCTYPE	Doctype pada awal dokumen merupakan *tag case sensitive* dan harus menggunakan *uppercase* seperti berikut <!DOCTYPE ...>
Tidak terdapat *sub-elemen* dalam elemen head	Error yang terjadi ketika tidak terdapat *tag title* di dalam elemen head
Missing ALT Text	Atribut Alt dalam beberapa elemen, seperti <img> wajib digunakan dan dicantumkan
Missing Close omittag	Untuk *tag tag* yang tidak menggunakan *closing tag*, atau disebut dengan *omittag* wajib diakhiri dengan />. *Error* terjadi apabila *tag* hanya diakhiri dengan >.
Missing scriptype	Penggunaan *tag script* tanpa menggunakan atribut *type*, *script* wajib menggunakan atribut *type*, seperti berikut <script type=”text/Jquery”>.

Kesalahan	Perbaikan
Element yang tidak sesuai dengan versi dokumen HTML	Versi HTML yang digunakan tidak mendukung elemen tertentu, sehingga penggunaan tipe dokumen sangat menentukan.

I.2.7.2 Standar Pengkodean CSS

CSS atau *Cascading Style Sheet* merupakan bahasa *style sheet* yang digunakan untuk mendeskripsikan presentasi atau tampilan dari dokumen yang ditulis dalam bahasa *markup*, seperti HTML. Standar penulisan CSS dikeluarkan oleh *World Wide Web Consortium* dan dapat diakses secara online menggunakan *validator tools* yang beralamat di https://jigsaw.w3.org/css-validator/. Berikut merupakan kesalahan umum pada pengkodean CSS [22].

Tabel I.2 Kesalahan pengkodean CSS

Kesalahan	Perbaikan
Selector yang tidak ditutup	Deklarasi *style* di dalam *selector* diawali dengan kurawal buka '{' dan harus diakhir dengan kurawal tutup '}'. Terkadang deklarasi *style* tidak diakhiri dengan kurawal tutup.
Tidak menggunakan *style* global	Untuk style pada elemen-elemen yang sering digunakan, perlu di deklarasikan secara global agar penggunaan *class* tidak berlebihan.
Nama class dan di yang tidak unik	Penggunaan nama class yang atau di yang umum, sehingga kurang spesifik.
Tidak menggunakan kode *shorthand* untuk value dan warna.	Untuk beberapa style, seperti padding dapat menggunakan kode *shorthand* yang lebih pendek dan menghemat line.
Kesalahan *positioning*	Penggunaan style *position* yang berlebihan

I.2.7.3 Standar Pengkodean JavaScript

Kode JavaScript pada situs menggunakan *framework* jQuery, sehingga standar pengkodean yang digunakan adalah *Jquery Coding standards* yang dapat diakses secara online pada alamat https://contribute.jquery.org/style-guide/js/. Standar tersebut dikeluarkan oleh jquery Foundation sebagai acuan untuk membuat kode Jquery dengan *framework* jquery. Berikut adalah beberapa hal yang standar kode JQuery [8].

Tabel I.3 JQuery Foundation JavaScript Style Guide

Topik	Penjelasan

Spacing	Indentasi menggunakan tab
	Tidak ada spasi pada ujung baris
	Baris sebaiknya tidak lebih dari 80 karakter, dan tidak boleh lebih dari 100 karakter termasuk indentasi, kecuali jika mengandung URL atau regex
	If, Elbe, for, while, dan Etty selalu dilengkapi kurawal '{}' dan lebih dari satu baris
	Operator khusus unary, seperti ++, tidak boleh menggunakan spasi
	, ; dan : (untuk definisi objek) tidak boleh didahului dengan spasi.
	; digunakan sebagai penghenti dan harus berada diakhir baris
	? dan : untuk kondisional harus terdapat spasi pada kedua sisi
	{}, [], fn() untuk construct yang kosong, tidak boleh diisi spasi
Constructor	Fungsi konstruktor harus selalu dipanggil dengan list argumen
Equality	Pemeriksaan persamaan, menggunakan '===' terkecuali untuk pemeriksaan abstrak dan null menggunakan '=='
Comments	Komentar selalu diawali dengan baris kosong
Quote	Menggunakan kutip ganda
Naming Conventions	Fungsi dan variabel harus sebuah kata yang mewakili isinya dan menggunakan camelCase, kecuali untuk iterator pada perulangan.
DOM Node Rules	.nodeName adalah nama yang digunakan dibandingkan .tagName
Switch	Penggunaan switch, tidak direkomendasi tetapi baik digunakan untuk jumlah Chase yang banyak. Ketika mengiakan switch, gunakan break pada tiap Chase, dan ratakan case dengan switch.

I.2.7.4 Standar Pengkodean PHP

Zend Framework merupakan salah satu framework yang digunakan untuk membangun aplikasi website ataupun website dengan bahasa PHP. Zend framework mempunyai coding standar untuk bahasa PHP yang harus dipenuhi. Coding standar tersebut terdiri dari 4 hal utama yaitu, PHP *File formatting*, *naming conventions*, *coding style* dan *inline documentation* [9].

Tabel I.4 Standar Kode PHP dari Zend

Topik	**Kategori**	**Penjelasan**
File Formatting	*General*	*File* yang hanya berisi kode PHP, tidak perlu menggunakan tag penutup "?>".
	Indentation	Identasi harus terdiri dari 4 spasi, tidak menggunakan tab.
	Line Termination	Baris ditutup dengan karakter *linefeed* tunggal.
	Maximum Line Length	Karakter dalam tiap baris kode, dibawah 80 karakter.
Naming Convention	*Class*	Nama kelas diawali dengan direktori tempat kelas disimpan. Nama kelas hanya menggunakan karakter alfanumerik, dengan pemisah menggunakan *underscores*. Apabila terdiri lebih dari satu kata, huruf pertama pada tiap kata harus kapital.
	Filenames	Hanya menggunakan karakter alfanumerik, *underscore* (_) dan *dash* (-). Dilarang menggunakan spasi. Semua *File* yang mengandung kode PHP wajib menggunakan ekstensi .php
	Functions and methods	Nama fungsi hanya menggunakan karakter alfanumerik, *underscore* tidak disarankan. Nama fungsi menggunakan format *camelCase*. Nama fungsi harus jelas sesuai dengan apa yang dilakukannya. Untuk OOP, *instance* harus menggunakan prefiks *get* atau *set*. Dalam *methods*, untuk objek *Private* dan *protected* huruf pertama dari nama variabel harus berupa *underscore* (_). *Method* public, tidak boleh meng*and*ung *underscore*.
	Variabel	Nama variabel mempunyai aturan yang sama dengan nama fungsi. Nama variabel harus menunjukan nilai yang ditampungnya.
	Konstanta	Nama konstanta dapat terdiri dari karakter alfanumerik, *underscore*, dan nomor. Semua huruf yang digunakan harus kapital. Kata dipisahkan menggunakan *underscore*. Konstanta di definisikan sebagai anggota kelas dengan *const modifier*.
Coding Style	PHP *Demarcation*	Kode PHP harus selalu diapit oleh PHP tag <?php ?>. Pengecualian untuk

Topik	**Kategori**	**Penjelasan**
		File dengan hanya kode PHP di dalamnya, tidak diperkenankan menggunakan *tag* penutup ?>
	String	*String Literals* Isi dari sebuah *string* yang memang berupa *string*, diapit oleh t*and*a apostrop atau kutip satu (‘). Apabila dalam *string* terdapat t*and*a apostrop, maka digunakan t*and*a kutip g*and*a (“)
		String Concatenation Penggabuangan dua *string* atau lebih, dilakukan menggunakan operator titik (.)
	Array	*Numerically indexed Array* Indeks tidak bernilai negatif, dan dimulai dari 0. Nilai *Array* dipisahkan dengan koma.
	Class Declaration	Nama kelas harus sesuai dengan konvensi zend framework. Kurung kurawal ditempatkan dibawah baris nama kelas. Harus terdapat blok dokumentasi sebelum kelas. Kode dalam kelas diberi inden 4 spasi. Dalam satu dokumen php hanya terdapat satu kelas. Untuk deklarasi *method*, mekanisme parameter yang digunakan hanya pass-by-reference. Tidak diperkenankan All Time pass-by-reference. *Return value* tidak boleh diakhiri dengan t*and*a kurung
	Control statement	*If/Else/Elseif* Terdapat satu spasi antar keyword *if* dengan t*and*a kurung buka. Operator dan operan dipisahkan dengan spasi agar dapat dibaca. Kurung kurawal lokal dianjurkan untuk kondisional yang banyak. Kurung kurawal pembuka berada dalam satu baris, sedangkan kurung kurawal penutup berada di barisnya sendiri.
		Switch Semua konten di dalam *switch*, menggunakan identasi dengan 4 spasi. *Construct default* harus selalu ada dalam sebuah *switch*.
Documentation Format	*File*	Setiap *File* yang meng*and*ung kode PHP harus disertai *docblock* di bagian paling atas *File*, yang berisi deskripsi

Topik	Kategori	Penjelasan
		singkat, lisensi, versi dan *link* package, software rilis.
	Classes	Setiap kelas harus meng*and*ung *docblock* dengan isi minimal deskripsi singkat, copyright, lisensi, versi, *link*, tanggal tersedia dan tanggal dicabut.
	Function	Setiap fungsi, termasuk *method* harus mempunyai *docblock* yang berisi deskripsi fungsi, semua argumen dan kemungkinan *return value*

I.2.8 *Heuristic* Evaluation

Heuristic Evaluation merupakan sebuah metode untuk menyusun saran terhadap sebuah sistem menggunakan kumpulan *Heuristic* yang relatif sederhana dan umum. *Heuristic* sendiri merupakan panduan umum yang dapat digunakan untuk keputusan-keputusan perancangan atau untuk meninjau keputusan yang sudah dibuat [23].

Metode ini pertama kali dikembangkan oleh Jakob Nielsen dan Rolf Molich, pada tahun 1990. Ide umum dari metode ini adalah dengan mengumpulkan beberapa *evaluator* untuk meneliti dan mengevaluasi sebuah desain dengan menggunakan pedoman [6]. Berikut adalah 10 pedoman yang dikembangkan pertama kali oleh Nielsen dan Molich [23]:

1. Visibilitas dari status sistem, pengguna harus selalu diberikan informasi mengenai apa yang terjadi saat ini pada sistem.
2. Kecocokan antara sistem dengan dunia nyata, sistem harus berbicara dengan bahasa yang digunakan oleh pengguna dan menggunakan konsep yang familiar dengan pengguna.
3. Kontrol dan kebebasan pengguna, terkadang pengguna memilih fungsi secara tidak sengaja, dan harus diberikan jalan keluar dari situasi tersebut.
4. Konsistensi dan standar, pengguna tidak boleh bertanya-tanya apakah istilah atau aksi mempunyai arti sama atau tidak pada konteks berbeda.
5. Pencegahan *error*, buatlah sistem atau pengguna kesulitan untuk melakukan *error*.

6. Pengenalan dibandingkan mengingat ulang, buatlah objek dan opsi terlihat konsisten sehingga pengguna tidak perlu mengingat informasi tapi mudah dikenali.
7. Fleksibilitas dan efisiensi penggunaan, memungkinkan pengguna dengan mudah melakukan aksi yang mempunyai frekuensi sering dilakukan.
8. Desain yang estetik dan minimalis, dialog menampilkan pesan yang singkat tapi efektif, tidak meng*and*ung informasi yang tidak diperlukan.
9. Bantu pengguna mengenal, mendiagnosa, dan keluar dari *error*, error ditampilkan dalam bentuk pesan yang sederhana menggunakan bahasa pengguna bukan kode.
10. Bantu dan dokumentasikan, sediakan dokumentasi dan *help* untuk pengguna sehingga dapat mempelajari sistem.

I.2.8.1 Daftar *Heuristic* dan Sistem Penilaian

Seperti yang telah dipaparkan pada batasan masalah *Heuristic* yang digunakan adalah daftar *Heuristic* yang disusun oleh Dr. David Travis yang tersedia secara online dan dapat diunduh pada alamat http://www.userfocus.co.uk/resources/guidelines.html. Daftar tersedia dalam bentuk file *workbook*, yang terdiri dari 247 *Heuristic*, dibagi dalam 9 kategori beserta halaman hasil penilaian keseluruhan *Heuristic*. Kategori dalam *Heuristic* dibagi menjadi *home page usability*, *task orientation*, *navigation and information architecture*, *forms and data entry*, *trust and credibility*, *writing and content quality*, *page layout and visual design*, *search usability*, dan *help, feedback, and error tolerance.*

Sistem penilaian yang digunakan adalah sistem penilaian menggunakan skala *likert-like* dengan 3 poin, dalam bahasa Inggris skala ini disebut dengan *Likert-like type scale* [24]. Nilai tersebut digunakan pada masing-masing *Heuristic* yang sesuai untuk website yang dievaluasi. Berikut adalah sistem penilaian yang digunakan [25].

Tabel I.5 Poin penilaian *Heuristic*

Poin	**Keterangan**
+1	*Website* **memenuhi** kriteria *Heuristic* yang tersedia

Poin	Keterangan
0	*Website* **mendekati** kriteria *Heuristic* yang tersedia
-1	*Website* **tidak memenuhi** kriteria *Heuristic* yang tersedia

Poin yang diberikan oleh *evaluator* pada tiap *Heuristic* akan dijumlahkan dan disebut sebagai *raw score*. Beberapa pertanyaan dapat tidak diisi apabila evaluator merasa hal tersebut tidak relevan, sehingga jumlah pertanyaan yang diisi akan dihitung dan debut sebagai *answer*. *Raw score* dan *answer* dijumlahkan lalu hasilnya akan dibagi dengan *answer* dikali 2. Berikut adalah rumus perhitungan skor tersebut.

$$\text{Skor} = \frac{(\text{raw score} + \text{answer})}{(2 \times \text{answer})} \text{ x } 100\ \% \qquad \text{(I.1)}$$

I.2.9 *Cognitive walkthrough*

Cognitive walkthrough membutuhkan *review* yang detil mengenai rangkaian aksi pada website [23]. *Walkthrough* atau penelusuran membutuhkan review yang detail dari urutan aksi. Urutan aksi merujuk pada tahap-tahap antarmuka yang akan muncul ketika pengguna perlu menyelesaikan sebuah *task.* Fokus utama dari *cognitive walkthrough* adalah untuk mengetahui semudah apa sistem untuk dipelajari.

Untuk melakukan penelusuran, diperlukan 4 hal berikut [23]:

1. Spesifikasi atau *prototype* dari sistem, tidak harus lengkap tetapi detail.
2. Deskripsi dari *task* yang mungkin dilakukan pengguna dalam sistem.
3. Sebuah daftar lengkap mengenai aksi yang diperlukan untuk menyelesaikan *task*.
4. Indikasi mengenai siapa itu pengguna serta pengalaman dan pengetahuan apa yang mungkin dimiliki pengguna.

I.2.10 UEQ (User Experience Questionnaire)

Merupakan kuesioner untuk mengukur *user experience* secara kuantitatif dan diukur dari pengguna akhir (*end-user*) satu sistem atau aplikasi. UEQ dikembangkan oleh Dr. Martin Schrepp dan koleganya di Jerman [4]. Kuesioner ini awalnya berbahasa Jerman, dan kemudian diterjemahkan ke dalam bahasa Inggris

pada tahun 2005. Kuesioner dibuat khusus untuk evaluasi usability yang dapat dilakukan dengan sederhana dan membutuhkan hasil yang cepat, serta tidak harus dilakukan oleh ahli [4].

Kuesioner ini melakukan evaluasi terhadap 3 aspek utama *user experience*, yaitu daya tarik, kualitas pragmatis yang berorientasi terhadap tujuan, dan kualitas hedonis yang tidak berorientasi terhadap tujuan. Kuesioner menghasilkan 6 skala penilaian [3], yaitu

1. Daya tarik

   Penilaian terhadap ketertarikan pengguna terhadap produk, apakah pengguna menyukai produk atau tidak.
2. Kejelasan

   Mencakup penilaian terhadap penggunaan produk, apakah mudah dipelajari bagaimana cara penggunaannya atau tidak.
3. Efisiensi

   Menilai apakah *task* dapat diselesaikan dengan efisien atau tidak.
4. Ketepatan

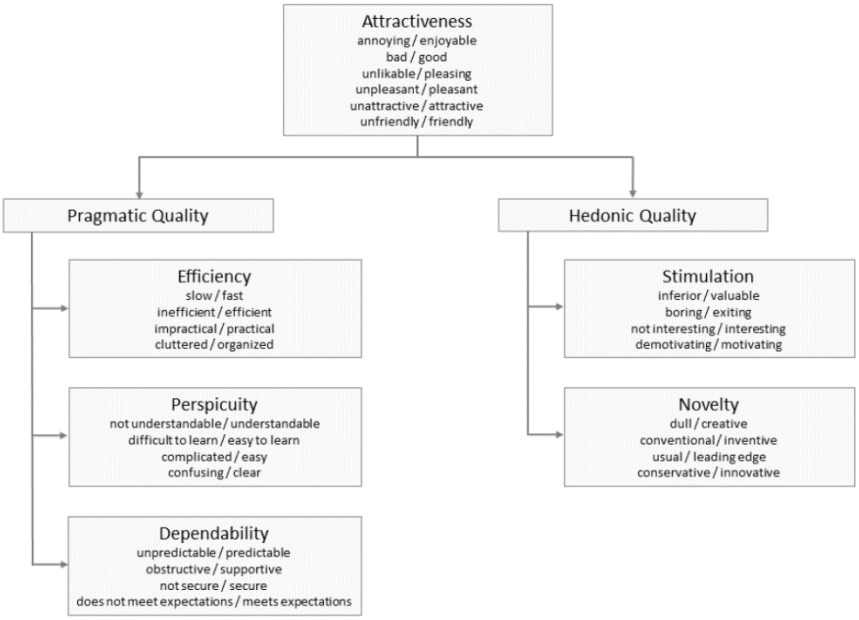
   Menilai kontrol pengguna terhadap produk, apakah pengguna mempunyai kontrol atau sebaliknya.
5. Stimulasi

   Menilai keterkesanan dari pengguna, apakah pengguna terangsang untuk melakukan eksplorasi produk atau tidak.
6. Kebaruan

   Mencakup penilaian terhadap inovasi produk, apakah produk inovatif atau kreatif, serta dapat memancing ketertarikan pengguna.

Kategori daya tarik merupakan kategori dengan valensi dimensi murni yang berdiri sendiri, sehingga tidak termasuk pada kategori kualitas tertentu. Untuk kategori kejelasan, efisiensi, dan ketepatan termasuk pada kualitas pragmatis. Seperti yang telah dipaparkan, kualitas pragmatis merupakan kualitas yang berorientasi pada tujuan, dimana untuk mengukur kualitas tersebut, responden harus terlebih dahulu menyelesaikan sebuah tugas atau tujuan yang dapat dicapai melalui situs web. Selain kualitas pragmatis, terdapat pula kualitas hedonis, yaitu

kualitas yang tidak berorientasi pada tujuan, sehingga responden hanya perlu melakukan akses, tanpa harus menyelesaikan tujuan [3]. Berikut adalah struktur dari kategori dan pertanyaan-pertanyaan pada *user experience questionnaire.*



Gambar I.3 Struktur Kategori *User Experience Questionnaire* [3]

I.2.11 GTMetrix

GT Metrix merupakan *website performance tool* yang dikeluarkan oleh GT.Net perusahaan *hosting* dari Kanada. GT Metrix menampilkan penilaian dari Google PageSpeed Insight dan Yahoo YSlow serta memberikan rincian request yang dilakukan pada website juga menampilkan *history* akses selama satu minggu ke belakang dari hari situs dianalisis.

I.2.12 PSSUQ

Post-Study System Usability Questionnaire (PSSUQ) adalah kuesioner yang didesain oleh Lewis untuk mengetahui tingkat kepuasan pengguna terhadap sebuah sistem atau aplikasi yang diujikan [7, 26]. PSSUQ yang digunakan merupakan versi ketiga dengan total jumlah pertanyaan sebanyak enam belas pertanyaan. Pengukuran penilaian terhadap masing-masing *item* menggunakan

skala likert 1-7 dimana 1 mewakili nilai sangat setuju dan 7 mewakili nilai sangat tidak setuju.

Pertanyaan-pertanyaan tersebut mewakili kategori tertentu dalam aspek *usability* [27]. Berikut merupakan kategori dari masing-masing pertanyaan, diantaranya adalah sebagai berikut.

1. *Overall* diwakili oleh pertanyaaan 1-19
2. *System Usefulness* diwakili oleh pertanyaan 1-8
3. *Interface Quality* diwakili oleh pertanyaan 9-15
4. *Information Quality* diwakili oleh pertanyaan 16-19

Sistem penilaian menggunakan skala likert 7 poin, berikut adalah rincian masing-masing poin.

Tabel I.6 Skala Likert untuk PSSUQ

Nilai Skala	Penjelasan
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Agak Tidak Setuju
4	Netral
5	Agak Setuju
6	Setuju
7	Sangat Setuju

Dari nilai yang diberikan responden dilakukan perhitungan rata-rata untuk masing-masing kategori, sehingga menghasilkan skor PSSUQ untuk masing-masing kategori. Berikut adalah rumus untuk menghitung skor dari masing-masing kategori PSSUQ.

$$Skor = \frac{\text{jumlah skor penilaian responden dari setiap kategori}}{\text{jumlah nomor item pertanyaan setiap kategori}} \quad \text{(I.2)}$$