BAB II
TINJAUAN PUSTAKA

II.1 Tinjauan Tempat Penelitian

Awalnya ide membangun E-Kosan bermula dari tugas besar mata kuliah Analisis dan Desain Sistem Informasi pada program studi Teknik Informatika Universitas Komputer Indonesia. Kemudian E-Kosan telah mengalami beberapa tahap pengembangan dan didirikan pada tanggal 22 November 2012 oleh pendirinya yang bernama Supriadi dan mulai dipublikasikan pada tanggal 12 Desember 2012.

Berdasarkan data dari *Google Analytics* bahwa sampai April 2015 jumlah pengguna yang mengunjungi situs E-Kosan adalah sebanyak 84.943 orang. Sedangkan untuk *fanpage Facebook* sudah mencapai 3.252 penggemar dan akun *Twitter* E-Kosan sudah mencapai 4.694 pengikut. Melihat respon yang baik dari pengguna situs E-Kosan dan keseriusan pemilik E-Kosan membuat situs ini dapat bertahan hingga sekarang.

II.1.1 Profil E-Kosan

E-Kosan adalah salah satu aplikasi berorientasi website yang bergerak di bidang jasa penyebaran informasi kosan. Saat ini E-Kosan fokus untuk menyediakan informasi kosan yang *up to date* untuk daerah Bandung, Jawa Barat. Pengguna E-Kosan terdiri dari pemilik kosan dan pencari kosan. Pemilik kosan dapat mendaftarkan kosan miliknya agar dipublikasikan di situs E-Kosan. Sedangkan pencari kosan dapat mendaftarkan dirinya untuk mencari kosan yang diinginkan.

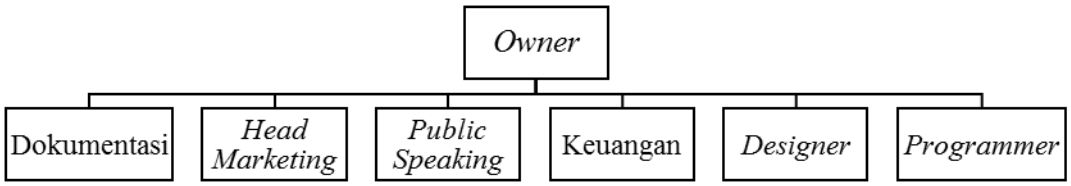
Visi E-Kosan adalah menjadi yang terbaik di bidang pengiklanan informasi kos-kosan dan memiliki cabang di kota-kota pendidikan yang ada di Indonesia. Kemudian misi E-Kosan adalah mengutamakan informasi yang benar dan pelayanan yang baik kepada pencari kos, mencakup semua rumah kos-kosan yang

ada di Bandung, dan menjadi peserta dalam *event-event* kampus sebagai bentuk promosi.

E-Kosan dapat dijadikan solusi bagi pencari kosan terutama mahasiswa ketika ingin mencari kosan. Pencari kosan tidak perlu mencari kosan dengan mendatangi kosan satu per satu secara manual. Pencari kosan cukup membuka situs E-Kosan untuk mencari dan memilih kosan yang kira-kira sesuai dengan keinginan kemudian bisa langsung menghubungi pemilik kosan atau langsung mendatangi kosan tersebut.

II.1.2 Struktur Organisasi E-Kosan

Struktur organisasi E-Kosan dapat dilihat pada Gambar II.1.



Gambar II.1 Struktur Organisasi E-Kosan

Penjelasan setiap bagian dari struktur organisasi diatas adalah sebagai berikut:

a. *Owner*

   Owner memiliki kewajiban dalam menyediakan modal yang harus dikeluarkan untuk kebutuhan perusahaan dalam mengembangkan e-kosan.com.

b. Dokumentasi

   Dokumentasi mempunyai tugas pokok melaksanakan pengumpulan pengolahan data dalam rangka pelaksanaan kegiatan evaluasi pendokumentasian dan pelaporan kegiatan unsur-unsur organisasi di e-kosan.com.

c. *Head Marketing*

   Head marketing bertanggung jawab terhadap kelancaran dan terlaksananya tujuan pemasaran sesuai dengan strategi dan sasaran pemasaran yang ditetapkan oleh e-kosan.com.

d. *Public Speaking*

Public speaking mempunyai tugas pokok yaitu menyampaikan pesan atau ide kepada publik dengan metode yang sesuai sehingga publik bisa memahami pesan atau ide yang disampaikan dan memperoleh manfaat dari pesan tersebut.

e. Keuangan

Keuangan mempunyai tugas pokok melaksanakan penyusunan rencana kegiatan, pengkoordinasian, pembinaan, dan pengendalian terhadap anggaran yang terjadi di e-kosan.com

f. *Designer*

Designer mempunyai tugas pokok membuat desain website e-kosan.com. Tujuannya adalah mengembangkan desain website e-kosan.com menjadi lebih baik dari desain sebelumnya sehingga pengguna dapat menikmati tampilan website e-kosan pada layar komputer, *mobile*, maupun tablet.

g. *Programmer*

Programmer mempunyai tugas pokok untuk membangun atau mengembangkan website e-kosan.com terutama pada tahap *construction* dengan melakukan *coding*, mengimplementasikan *requirement* dan desain proses bisnis ke komputer dengan menggunakan algoritma atau logika dan bahasa pemrograman, serta melakukan *testing* terhadap website e-kosan.com.

II.1.3 Logo E-Kosan

Logo yang digunakan oleh E-Kosan dapat dilihat pada Gambar II.2.



Gambar II.2 Logo E-Kosan

II.2 Landasan Teori

II.2.1 *Usability*

Menurut *International Standards Organization* (ISO) yaitu ISO 9241-11 tahun 1995 yang mengatur mengenai standar ergonomi untuk pekerjaan kantor dengan *visual display terminal*, *usability* adalah sejauh mana sebuah produk bisa digunakan oleh pengguna tertentu untuk tujuan tertentu dengan efektif, efisien, dan kepuasan dalam konteks penggunaan tertentu. Efektivitas mengacu pada keakuratan dan kelengkapan pengguna untuk mencapai tujuan tertentu. Efisiensi berkaitan dengan sumber daya yang dikeluarkan dalam hubungannya dengan akurasi dan kelengkapan pengguna untuk mencapai tujuan. Kepuasan mengacu pada kenyamanan dan penerimaan penggunaan .

Sedangkan menurut Jakob Nielsen, *usablility* adalah set atribut kualitas yang menilai seberapa mudah *user interface* yang digunakan. *Usability website* terdiri atas lima kualitas komponen yaitu:

a. *Learnability*

   Learnability adalah konsep yang menilai kemampuan situs web untuk menyediakan kemudahan dalam menyelesaikan tujuan pengunjung situs web saat pertama kali penggunaannya.

b. *Memorability*

   Memorability adalah konsep yang menilai kemampuan situs web untuk menyediakan kemudahan bagi pengunjung situs web setelah mereka sudah lama tidak menggunakan situs tersebut dalam jangka waktu tertentu.

c. *Efficiency*

   Efficiency adalah konsep yang menilai bagaimana pengunjung situs web dengan cepat menyelesaikan tujuannya, setelah mereka mengetahui dengan baik situs web tersebut.

d. *Satisfaction*

   Satisfaction adalah konsep yang menilai apakah pengunjung dapat nyaman dengan desain pada sebuah situs web.

e. *Errors*

Errors adalah konsep yang menilai jumlah kesalahan yang pengunjung situs web sebabkan ketika dalam penggunaannya, tingkat keparahan pada kesalahan, dan bagaimana kemudahan bagi pengunjung dalam penanggulangan kesalahannya.

Berdasarkan kedua pernyataan diatas maka dapat disimpulkan bahwa pengertian *usability* adalah sebuah tolak ukur yang menilai tentang penggunaan sebuah produk (website) oleh pengguna berdasarkan lima aspek yaitu *learnability*, *memorability*, *efficiency*, *satisfaction*, dan *errors*. Daya guna (*usability*) merupakan unsur penting dalam *Human Computer Interaction* (HCI). *Usability* yang baik menggambarkan sistem tersebut mudah digunakan, tidak menimbulkan kekhawatiran bagi pengguna, dan mudah dipelajari.

II.2.2 *User Interface* (UI)

Menurut *e-book* karangan Wilbert O Galitz yang berjudul *The Essential Guide to User Interface Design An Introduction to GUI Design Priciples And Techniques, user interface* adalah bagian dari komputer dan perangkat lunak yang dapat dilihat, didengar, disentuh, diajak bicara, dan yang dapat dimengerti secara langsung oleh manusia . Dengan kata lain *user interface* dapat dikatakan sebagai teknik dan mekanisme dari tampilan antarmuka untuk berinteraksi dengan pengguna.

Menurut situs website www.quora.com *user interface* adalah desain yang sebenarnya dari antarmuka yang akan memfasilitasi interaksi yang menyenangkan dan bermanfaat antara pengguna dan produk (situs website) .

Berdasarkan kedua pernyataan tersebut, maka dapat ditarik kesimpulan bahwa *user interface* adalah bagian dari komputer dan perangkat lunak yang mengatur tampilan antarmuka untuk pengguna dan memfasilitasi interaksi yang menyenangkan antara pengguna dan sistem (situs website). UI juga bisa diartikan sebagai hasil akhir dari UX yang dapat dilihat.

II.2.3 *User Experience* (UX)

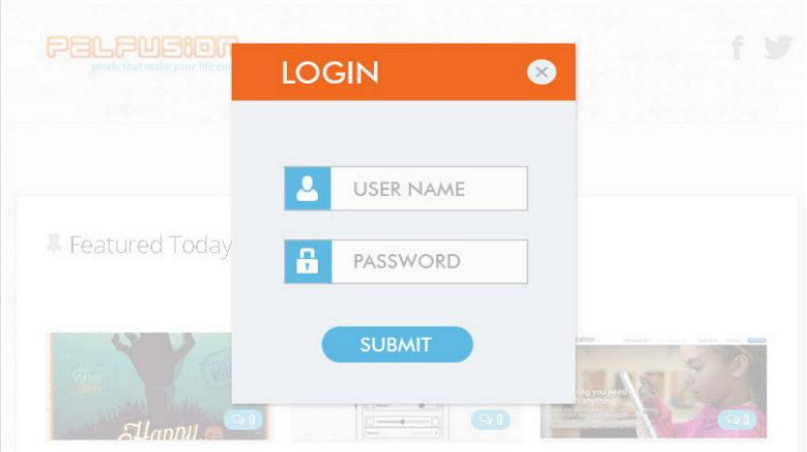
Menurut definisi dari ISO 9241-210, *user experience* adalah persepsi atau pengalaman seseorang dan responnya dari penggunaan sebuah produk, sistem, atau jasa. *User experience* menilai seberapa kepuasan dan kenyamanan seseorang terhadap sebuah produk, sistem, dan jasa .

Menurut Jakob Nielsen , *user experience* mencakup seluruh aspek interaksi terhadap pengguna dengan perusahaan, layanan, dan produk-produknya.

Berdasarkan kedua pernyataan di atas dapat disimpulkan bahwa *user experience* adalah hal-hal yang dirasakan seseorang terhadap penggunaan produk (situs website) yang menilai tentang tingkat kemudahan dan kenyamanan terhadap fungsionalitas yang disajikan oleh sebuah website.

II.2.4 Hubungan *User Interface* (UI) dan *User Experience* (UX)

UI dan UX memiliki konsep yang berbeda akan tetapi mereka memiliki hubungan yang dekat satu dengan yang lain. Berdasarkan pengertian UI dan UX pada subbab II.2.2 dan II.2.3 di atas, dapat disimpulkan bahwa sebuah website yang memiliki tampilan antarmuka yang baik akan tetapi sulit untuk digunakan menggambarkan UI yang baik tetapi tidak untuk UX. Sedangkan sebuah website yang terlihat sangat berguna akan tetapi terlihat berantakan atau kurang menarik menggambarkan UX yang hebat tetapi tidak untuk UI . Contoh UI yang diterapkan pada tampilan sebuah website dapat dilihat pada gambar Gambar II.3.



Gambar II.3 Contoh UI pada Website

Dalam penerapannya UX terkadang tidak membutuhkan sebuah UI untuk terlihat. Contohnya yaitu halaman *newsfeed Facebook*, ketika pengguna melakukan *scrolling newsfeed Facebook* maka pada saat tampilan *newsfeed* telah mencapai dasar layar maka secara otomatis halaman *newsfeed* akan melakukan *refresh* tanpa memerlukan reaksi pengguna untuk mengklik sebuah tombol. Hal itu termasuk UX karena pengguna dapat merasakan sebuah fungsionalitas dari website.

II.2.5 *User Experience Questionnaire*

User Experience Questionnaire (UEQ) adalah sebuah kuesioner yang bertujuan untuk mengukur *user experience* dengan cepat. Terdapat 6 skala pengukuran yang terbagi menjadi 26 pertanyaan di dalam UEQ, yaitu .

a. *Attractive* (Daya Tarik)

   Seberapa besar daya tarik dari sebuah produk. Misal: bagus atau jelek, atraktif atau tidak atraktif.

b. *Perpicuity* (Kejelasan)

   Seberapa besar kejelasan dari sebuah produk. Misal: mudah dipahami atau sulit dipahami.

c. *Efficiency* (Efisiensi)

   Seberapa besar pengguna dapat menyelesaikan tugasnya tanpa usaha yang besar atau efisien. Misal: cepat atau lambat, praktis atau tidak praktis.

d. *Dependability* (Ketepatan)

   Seberapa besar ketepatan yang dirasakan oleh pengguna melalui kontrol yang ia miliki. Misal: dapat diprediksi atau tidak dapat diprediksi, mendukung atau menghalangi.

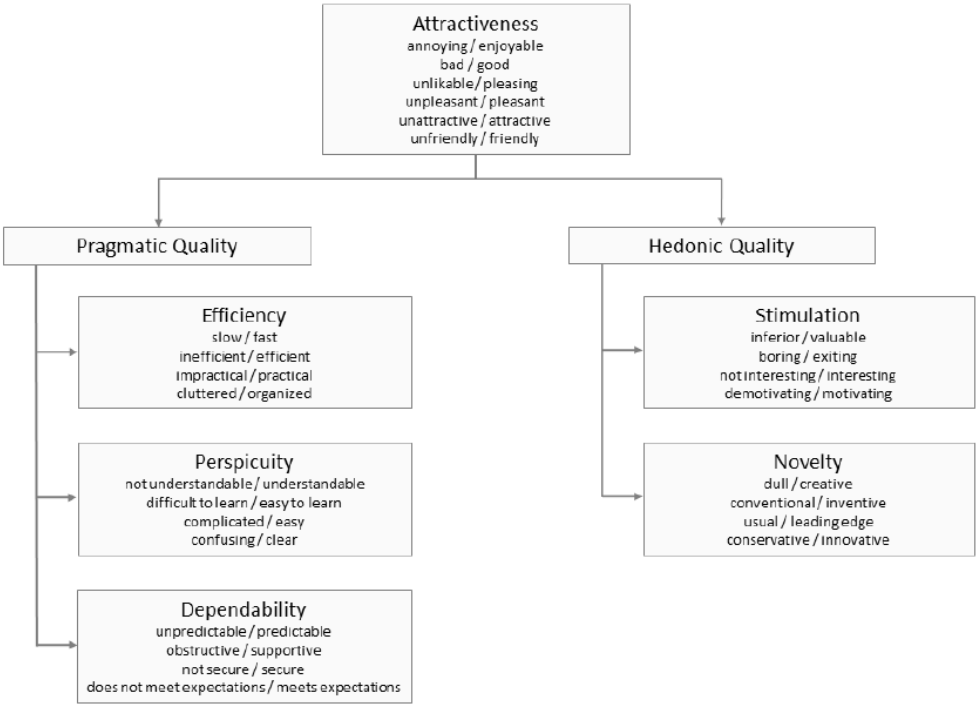
e. *Stimulation* (Stimulasi)

   Seberapa besar motivasi untuk menggunakan produk. Misal: bermanfaat atau kurang bermanfaat, menarik atau tidak menarik.

f. *Novelty* (Kebaruan)

Seberapa besar kebaruan dari produk. Misal: kreatif atau tidak kreatif, konservatif atau inovatif.

Daya tarik merupakan dimensi valensi yang murni, tidak termasuk aspek kualitas apapun. Kejelasan, efisiensi, dan ketepatan tergolong ke dalam aspek kualitas pragmatis yang berorientasi kepada tujuan sehingga pengguna harus melakukan tugas untuk mencapai tujuan melalui website. Sedangkan stimulasi dan kebaruan merupakan aspek kualitas hedonis yang tidak berorientasi pada tujuan sehingga pengguna hanya melakukan akses saja pada website tanpa perlu mencapai tujuan. Berikut adalah struktur pertanyaan yang tergolong ke dalam skala pengukuran pada UEQ .



Gambar II.4 Struktur Skala Pengukuran UEQ

UEQ menggunakan skala pengukuran *semantic differential* yang hampir mirip dengan skala *Likert* yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial . Berikut

adalah daftar pertanyaan yang diajukan oleh UEQ yang ditunjukkan oleh Gambar II.5 .



	1	2	3	4	5	6	7		
menyusahkan	○	○	○	○	○	○	○	menyenangkan	1
tak dapat dipahami	○	○	○	○	○	○	○	dapat dipahami	2
kreatif	○	○	○	○	○	○	○	monoton	3
mudah dipelajari	○	○	○	○	○	○	○	sulit dipelajari	4
bermanfaat	○	○	○	○	○	○	○	kurang bermanfaat	5
membosankan	○	○	○	○	○	○	○	mengasyikkan	6
tidak menarik	○	○	○	○	○	○	○	menarik	7
tak dapat diprediksi	○	○	○	○	○	○	○	dapat diprediksi	8
cepat	○	○	○	○	○	○	○	lambat	9
berdaya cipta	○	○	○	○	○	○	○	konvensional	10
menghalangi	○	○	○	○	○	○	○	mendukung	11
baik	○	○	○	○	○	○	○	buruk	12
rumit	○	○	○	○	○	○	○	sederhana	13
tidak disukai	○	○	○	○	○	○	○	menggembirakan	14
lazim	○	○	○	○	○	○	○	terdepan	15
tidak nyaman	○	○	○	○	○	○	○	nyaman	16
aman	○	○	○	○	○	○	○	tidak aman	17
memotivasi	○	○	○	○	○	○	○	tidak memotivasi	18
memenuhi ekspektasi	○	○	○	○	○	○	○	tidak memenuhi ekspektasi	19
tidak efisien	○	○	○	○	○	○	○	efisien	20
jelas	○	○	○	○	○	○	○	membingungkan	21
tidak praktis	○	○	○	○	○	○	○	praktis	22
terorganisasi	○	○	○	○	○	○	○	berantakan	23
atraktif	○	○	○	○	○	○	○	tidak atraktif	24
ramah pengguna	○	○	○	○	○	○	○	tidak ramah pengguna	25
konservatif	○	○	○	○	○	○	○	inovatif	26

Gambar II.5 Daftar Pertanyaan UEQ

II.2.6 Metode *Heuristic Evaluation*

Menurut Jakob Nielsen *heuristic evaluation* adalah metode *usability engineering* untuk menemukan masalah *usability* di dalam *user interface*. *Heuristic evaluation* melibatkan evaluator dalam jumlah sedikit untuk memeriksa tampilan antarmuka dan mengkritiknya dengan mengenali prinsip *usability* (*heuristic*) .

Kelebihan dari metode ini yaitu relatif mudah digunakan, tidak menghabiskan banyak biaya, tidak membuang banyak waktu pengguna, dan dapat mengidentifikasi banyak masalah. Sedangkan kelemahan dari metode ini adalah pengevaluasi harus memiliki keahlian *user interface* yang baik untuk menilai sebuah website . Berikut adalah lima tahap pada standar proses *heuristic evaluation*:

1. Perencanaan Evaluasi (*Plan The Evaluation*)

   Pada tahap ini diharuskan menetapkan tujuan dari penelitian untuk kemudian dikomunikasikan kepada pengevaluasi untuk persiapan tahap selanjutnya.
2. Pemilihan Evaluator (*Choose Your Evaluators*)

   Pada tahap ini dilakukan pemilihan siapa pengevaluasi yang akan diikutsertakan dalam penelitian. Pengevaluasi dapat terdiri dari kalangan *usability specialist*, sesama *programmer*, dan pengguna internet yang potensial. Berdasarkan penelitian Nielsen Nielsen menyarankan menggunakan tiga hingga lima evaluator untuk melakukan metode *heuristic evalution*. Hal ini dinilai cukup untuk mengetahui sebanyak 65% hingga 75% dari beragam masalah *usability* yang ada .
3. Beri Penjelasan Singkat Pada Evaluator (*Brief The Evaluators*)

   Pada tahap ini akan memberikan penjelasan singkat kepada calon pengevaluasi tentang metode *heuristic evaluation*. Pengevaluasi akan diberikan pedoman penilaian berupa sekumpulan pertanyaan tentang *usability* situs website.
4. Pelaksanaan Evaluasi (*Conduct The Evaluation*)

   Pada tahap ini setiap pengevaluasi akan mulai melakukan evaluasi terhadap website dengan menggunakan pedoman penilaian yang mengacu pada metode *heuristic evaluation*.
5. Analisa Hasil Evaluasi (*Analyze The Results*)

   Pada tahap ini akan dilakukan analisa hasil dari observasi yang dilakukan oleh pengevaluasi pada website. Hasilnya akan berupa rekomendasi yang akan bermanfaat bagi pengembangan website.

Dalam melakukan evaluasi, menurut Jakob Nielsen terdapat sepuluh prinsip dalam *heuristic evaluation* . Akan tetapi prinsip heuristik yang

dikembangkan oleh Jakob Nielsen memiliki cakupan yang cukup luas, sehingga hal tersebut menjadi kekurangan tersendiri . Oleh karena itu pada penelitian ini menggunakan prinsip heuristik yang dikembangkan oleh Dr. David Travis, meliputi :

1. *Home Page Usability*

   Home page adalah wajah organisasi untuk dunia di mana pengguna memutuskan untuk berinteraksi dengan organisasi tersebut. Dibutuhkan keseimbangan antara menampilkan jumlah *item* yang ditawarkan dan jumlah konten sehingga pengguna dapat dengan mudah melakukan tugasnya.

2. *Task Orientation*

   Sebuah website dikatakan *task oriented* ketika mendukung pengguna secara efektif dan efisien untuk melengkapi tugas mereka.

3. *Navigation and Information Architecture* (IA)

   Navigation and information architecture merupakan komponen dari halaman website yang mendukung pengguna menemukan informasi dan menjelajahi situs konten website.

4. *Forms and Data Entry*

   Forms merupakan komponen website yang mengizinkan pengguna untuk berinteraksi dengan organisasi. *Forms* yang baik menyediakan akses ke fungsionalitas yang kaya walaupun pengguna hanya memberi masukan yang minim.

5. *Trust and Credibility*

   Sebuah website memiliki kredibilitas ketika pengguna percaya kepada konten dan kepada organisasi tersebut. Hal ini penting untuk membentuk opini pengguna terhadap sebuah *brand*.

6. *Writing and Content Quality*

   Menulis pada website tidak sama dengan menulis untuk cetakan. Orang-orang membaca dengan cara yang berbeda pada website dan melakukan *scanning* untuk menemukan informasi yang dibutuhkan.

7. *Page Layout and Visual Design*

Poin ini menanyakan apakah dialog pada situs website memiliki estetika dan minimalis. Desain visual yang tepat memiliki arti bahwa huruf, ikon, warna, dan tata letak membantu pengguna untuk menyelesaikan tugasnya dan halaman website tidak mengandung informasi yang tidak relevan.

8. *Search Usability*

   Search merupakan salah satu cara dominan yang dilakukan pengguna untuk berinteraksi dengan website. Sebuah mesin pencarian yang baik perlu memahami sisi manusiawi dari aktifitas pencarian yang berarti berhubungan dengan kesalahan ejaan dan sinonim (seperti "laptop" untuk "notebook"). *Google* telah menetapkan standar mengenai bagaimana seharusnya pencarian terlihat dan berperilaku serta banyak pedoman yang didasarkan dari praktek terbaik ini.

9. *Help, Feedback and Error Tolerance*

   Poin ini berguna untuk sebuah situs website dalam mencegah pengguna melakukan kesalahan serta memiliki toleransi terhadap kesalahan tanpa adanya tindakan minimal dari pengguna untuk melakukan perbaikan.

Akan tetapi dari sembilan prinsip tersebut tidak semua prinsip digunakan untuk mengevaluasi website E-Kosan. Prinsip yang tidak digunakan yaitu prinsip *Task Orientation*. Jadi jumlah prinsip yang digunakan hanya berjumlah delapan buah. Alasan tidak digunakannya *Task Orientation* yaitu pada website E-Kosan tidak memberikan tugas yang spesifik kepada pengguna yang menyebabkan pengguna bekerja dalam situs website.

II.2.6.1 Kuesioner dan Sistem Penilaian

Kuesioner yang digunakan dalam metode *heuristic evaluation* menggunakan kuesioner yang dibuat oleh Dr. David Travis dan dapat diunduh pada http://www.userfocus.co.uk/resources/guidelines.html. Kuesioner tersebut sudah tersedia dalam format *Excel workbook* yang menyertakan instruksi, 247 pertanyaan, dan perhitungan kuesioner. Penggunaan paket pertanyaan kuesioner ini dapat disesuaikan dengan kebutuhan masing-masing website yang akan dinilai.

Sistem penilaian yang digunakan juga mengikuti sistem penilaian yang dikembangkan Dr. David Travis. Pada sistem penilaiannya, Dr. David Travis menggunakan tiga poin penilaian. Tabel II.1 menjelaskan mengenai poin penilaian yang dipakai.

Tabel II.1 Penilaian dalam *Heuristic Evaluation*

Nilai	Keterangan
-1	Tidak memenuhi petunjuk pernyataan yang disediakan
0	Mendekati petunjuk pernyataan yang disediakan
1	Memenuhi petunjuk pernyataan yang disediakan

Setelah evaluator memberikan nilai pada masing-masing pernyataan maka langkah selanjutnya yaitu menjumlahkan semua nilai dari masing-masing prinsip. Penjumlahan nilai dari setiap prinsip ini disebut *raw score*. Kemudian setelah didapatkan *raw score* didapatkanlah nilai per prinsip heuristik dengan rumus sebagai berikut:

$$\text{Nilai per prinsip heuristik} = \frac{(\text{raw score} + \text{jumlah pernyataan yang dijawab})}{(2 \times \text{jumlah pernyataan yang dijawab})} \text{ x } 100\ \% \quad (1)$$

Lalu setelah didapatkan *raw score* dan nilai per prinsip heuristik maka dilakukan perhitungan rata-rata dengan rumus sebagai berikut:

$$\overline{x} = \frac{x_1 + x_2 + x_3 + \cdots + x_n}{n} \quad (2)$$

Keterangan:

$\overline{x}$ = rata-rata hitung

x_i = nilai sampel ke-i (nilai P1, nilai P2, nilai P3, dan seterusnya)

n = banyak data

II.2.7 *Code Refactoring*

Code refactoring adalah proses merubah sistem perangkat lunak tanpa merubah perilaku eksternal dari kode tetapi ditujukan untuk memperbaiki struktur internal dari kode tersebut. Langkah-langkah ketika akan melakukan *code refactoring* adalah sebagai berikut .

1. Melakukan tes terhadap kode

   Tes perlu dilakukan dengan baik agar dapat menemukan letak kesalahan pada kode. Menguji validasi standar kode dengan bantuan *tools online* juga dapat membantu dalam menemukan kesalahan pada kode.

2. Melakukan perbaikan terhadap kode

   Perbaikan terhadap kode dilakukan setelah ditemukannya kesalahan pada kode. Perbaikan yang baik akan terus dilakukan hingga kesalahan pada program dapat teratasi secara keseluruhan ketika dilakukan tes kembali.

Ketika memperbaiki struktur internal kode diperlukan juga standar pengkodean sebagai acuan untuk melakukan perbaikan. Penggunaan standar pengkodean dimaksudkan agar kode pemrograman memiliki sebuah standar atau dasar umum. Masing-masing kode pemrograman memiliki standar pengkodean yang berbeda. Berikut adalah penjelasan mengenai standar pengkodean yang digunakan dalam skripsi ini:

1. Standar Pengkodean *World Wide Web Consortium* (W3C)

Standar pengkodean dari W3C ini digunakan sebagai analisis standar kode HTML dan CSS. W3C adalah organisasi internasional yang bekerja untuk mengembangkan standar web . Ketika melakukan pengecekan menggunakan HTML dan CSS *validator* secara *online* yang disediakan oleh W3C biasanya terdapat beberapa kesalahan yang sering ditemui. Berikut adalah contoh kesalahan yang sering ditemui di HTML *validator* .

a. Tidak ada *DOCTYPE*



Gambar II.6 Kesalahan HTML – Tidak Ada *DOCTYPE*

Tidak adanya *DOCTYPE* menyebabkan *web browser* harus menebak bahasa apa yang digunakan pada halaman yang ditulis *developer*. Oleh karena itu untuk memperbaiki kesalahan ini perlu ditambahkan HTML atau XHTML *doctype* pada awal halaman.

b. Lupa untuk menutup elemen



Gambar II.7 Kesalahan HTML – Lupa Menutup Elemen

Jika telah membuka *tag* di dalam dokumen HTML, *developer* harus ingat untuk menutup *tag* tersebut di tempat yang sesuai. Jika lupa menutup elemen dapat menyebabkan masalah yang serius pada *layout*. Peringatan yang biasa muncul pada W3C *validator* adalah pesan seperti "*end tag for "p" omitted, but OMITTAG NO was specified*" yang berarti Anda lupa untuk menutup *tag* paragraf.

c. *Missing self closing elementer*



Gambar II.8 Kesalahan HTML – *Missing Self Closing Elementer*

Kebanyakan elemen HTML memiliki *closing tag* yang terpisah seperti `</div>` tetapi elemen lain seperti `input`, `img`, dan `meta` memiliki *self closing* yang berarti mereka memiliki tanda garis miring (*slash*) sebelum *tag* penutup seperti berikut `/>`.

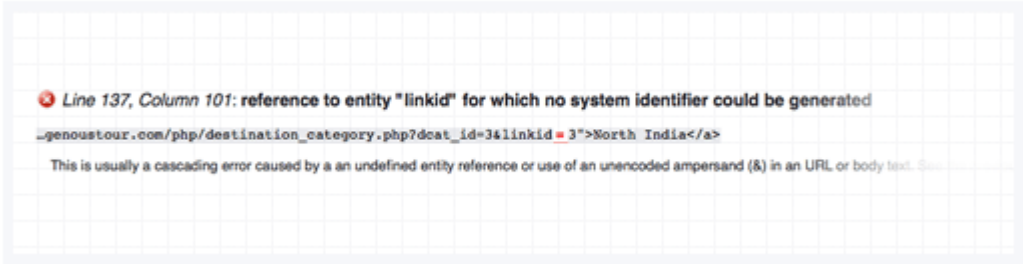
d. Lupa untuk mengkonversi karakter spesial



Gambar II.9 Kesalahan HTML – Lupa Mengkonversi Karakter Spesial

Salah satu karakter spesial yang harus dikonversi ke dalam entitas HTML adalah tanda &. Tanda & diubah menjadi `&amp`.

e. Tidak menyandikan karakter di URL



Gambar II.10 Kesalahan HTML – Tidak Menyandikan Karakter di URL

Mengikuti poin sebelumnya, karakter spesial seperti *ampersand* juga harus dikonversi ketika tampak di URL. Contohnya *link* yang digunakan PHP memakai karakter & jadi akan lebih baik jika menulisnya dengan `&` untuk menghindari *error*.

f. *Block* elemen di dalam *inline* elemen



Gambar II.11 Kesalahan HTML – *Block* Elemen Dalam *Inline* Elemen

Peraturan pokok HTML adalah *blok* elemen tidak boleh berada di dalam *inline* elemen. Contoh yang sering dijumpai adalah *anchor* di dalam *header* sebagai berikut `<a href="#"><h3>bananas</h3></a>`. `<h3>` adalah *block* elemen, jadi seharusnya `<h3>` membungkus *inline* elemen menjadi `<h3><a href="#">bananas</a></h3>`.

g. Tidak ada atribut ALT



Gambar II.12 Kesalahan HTML – Tidak Ada Atribut ALT

Setiap *image* yang dimasukkan ke dalam dokumen HTML harus menggunakan atribut ALT untuk mendeskripsikan konten. Contohnya adalah sebagai berikut: `<img src=”cat.png” alt=”cat picture”>`.

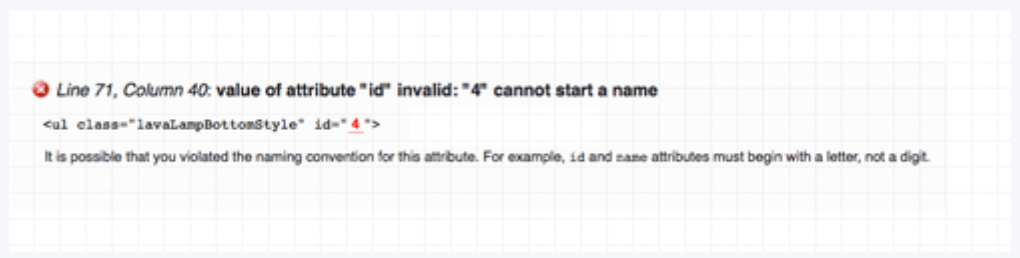
h. Menggunakan atribut "*width*" dan "*height*"



Gambar II.13 Kesalahan HTML – Menggunakan Atribut "*width*" dan "*height*"

Atribut *width* dan *height* tidak disarankan untuk disimpan pada dokumen HTML. Akan lebih baik jika menyimpan atribut *width* dan *height* di CSS *stylesheet* sebagai bagian dari *styling*.

i. Mengawali *class* atau *id* dengan angka



Gambar II.14 Kesalahan HTML – Mengawali Nama "*class*" Atau "*id*" dengan Angka

Nama *class*, *id*, atau atribut tidak dapat diawali dengan angka. Mereka dapat mencantumkan angka tetapi tidak sebagai awal penamaan.

Sedangkan pada CSS *validator* contoh kesalahan yang sering dijumpai adalah sebagai berikut

a. Lupa untuk menutup dengan baik

Salah satu kesalahan yang sering disebabkan karena ketidaktelitian *developer* adalah lupa untuk menutup dengan baik. Perlu diingat pada aturan penulisan *style* CSS setelah penulisan *selector* pasti diawali dengan penulisan kurung

kurawal buka dan diakhiri dengan kurung kurawal tutup. Kemudian di akhir *value* setiap *properties* diakhir dengan tanda titik koma (;).

b. Lupa menggunakan *hashtag* (#) ketika mengetik warna dalam heksa

   Ini merupakan contoh yang salah karena tidak menggunakan tanda # `color: ea6bc2;` seharusnya penulisan yang benar adalah seperti berikut `color: #ea6bc2;`

c. Tidak menggunakan *shorthand code*

   Sebenarnya tidak masalah jika tidak menggunakan *shorthand code*. Akan tetapi *shorthand code* berguna untuk membantu *developer* agar baris *stylesheet* mereka lebih singkat serta dapat mempercepat waktu *loading* pengguna ketika membuka halaman dan memudahkan mereka ketika akan mengubah kode tersebut di lain waktu. Berikut adalah contoh penulisan kode tanpa menggunakan *shorthand code* dan dengan menggunakan *shorthand code*:

Tabel II.2 Contoh Penulisan *Shorthand Code*

Tanpa *Shorthand Code*	*Shorthand Code*

`border-top:1px solid #00f;`
`border-right:1px solid #00f;`
`border-bottom:1px solid #00f;`
`border-left:1px solid #00f;`
	`border:1px solid #00f;`

2. Standar Pengkodean PHP *Standards Recommendations* (PSRs)

PSRs digunakan sebagai standar analisis pengkodean untuk PHP. PSRs adalah spesifikasi PHP yang diterbitkan oleh PHP *Framework Interop* Group. Tujuannya adalah untuk memungkinkan interoperabilitas komponen dan untuk memberikan dasar teknis umum terhadap implementasi pembuktian konsep untuk praktek pemrograman dan pengujian optimal. PHP-FIG dibentuk oleh beberapa pendiri *framework* PHP . PSRs terdiri dari beberapa tingkatan dengan spesifikasi yang berbeda-beda. Pada analisis standar kode PHP nanti, PSR-1: *Basic Coding Standard* dan PSR-2: *Coding Style Guide* dipilih sebagai aturan yang digunakan.

3. Standar Pengkodean Google JavaScript Style Guide

Google JavaScript Style Guide digunakan sebagai standar analisis pengkodean JavaScript. *Google JavaScript Style Guide* memiliki dua konten di dalamnya yaitu *JavaScript Language Rules* dan *JavaScript Style Rules*. *Style Guide* ini terdiri dari kumpulan *list* mengenai hal-hal yang boleh dan tidak boleh untuk program *JavaScript* .

II.2.8 *Object Oriented Programming*

Object Oriented Programming (OOP) merupakan sebuah pendekatan pengembangan perangkat lunak, dimana struktur perangkat lunak dianalisis berdasarkan objek yang saling berinteraksi untuk menyelesaikan tugasnya berdasarkan urutan logika yang telah ditentukan. Interaksi ini membutuhkan pertukaran pesan bolak-balik antara objek yang satu dengan objek yang lainnya. Setelah pesan diterima, objek dapat melakukan *action* atau *method* .

Object Oriented Programming memiliki enam konsep dasar, yaitu:

1. *Object*

   Object merupakan struktur yang menggabungkan data dan prosedur agar dapat berjalan dengan suatu data.

2. *Abstraction*

   Abstraction merupakan kemampuan untuk menyaring sejumlah informasi dengan kata lain tidak perlu untuk mengetahui seluruh sifat-sifat asing suatu objek tetapi hanya menggunakan apa yang diperlukan saja. Kemampuan abstraksi diperlukan karena sangat sulit untuk memproses sejumlah informasi tanpa kemampuan untuk menyaring sifat objek.

3. *Encapsulation*

   Enkapsulasi adalah proses di mana akses data secara langsung tidak diizinkan. Untuk mengakses data tersebut dapat menggunakan objek yang bertanggung jawab dalam pengaksesan data. Dengan enkapsulasi data membuat sistem yang dibuat lebih aman.

4. *Inheritance*

   Inheritance atau pewarisan digunakan untuk mengklasifikasikan objek menurut karakteristik dan fungsi. Dengan menggunakan *inheritance* membuat pekerjaan dengan menggunakan objek menjadi lebih mudah.

5. *Polymorphysm*

   Polymorphysm adalah kemampuan dua buah objek berbeda untuk menerima pesan permintaan yang sama dengan cara yang berbeda pada tiap objek. Pada implementasinya dapat menggunakan objek dengan nama yang sama.

6. *Agregation*

   Agregation adalah dimana ketika sebuah objek yang terdiri dari gabungan objek yang saling bekerjasama. Dengan menggunakan agregasi memungkinkan untuk mengimplementasitakan proses bisnis sesuai dengan model.

II.2.9 *Post-Study System Usability Questionnaire*

Post-Study System Usability Questionnaire (PSSUQ) adalah kuesioner yang didesain untuk menaksir tingkat kepuasan pengguna tentang sistem atau aplikasi . Saat ini PSSUQ yang digunakan merupakan versi ketiga dengan total jumlah pertanyaan sebanyak enam belas pertanyaan. Setiap *item* dari kuesioner memiliki skala antara satu hingga tujuh (sangat tidak setuju sampai sangat setuju) yang merupakan skala *Likert* . Berikut adalah daftar pertanyaan PSSUQ versi tiga yang digunakan:

1. Secara keseluruhan saya puas dengan betapa mudahnya menggunakan sistem.
2. Penggunaan website sederhana.
3. Saya dapat dengan cepat mencari informasi kosan menggunakan website.
4. Saya merasa nyaman ketika menggunakan website.
5. Mudah untuk mempelajari cara menggunakan website.
6. Saya percaya saya dapat produktif menggunakan website.
7. Website memberikan pesan kesalahan yang jelas tentang bagaimana memperbaiki kesalahan tersebut.
8. Kapanpun saya membuat kesalahan ketika menggunakan website, saya dapat mengatasinya dengan mudah dan cepat.

9. Informasi (seperti bantuan online, pesan di layar, dan dokumentasi lain) jelas tersedia pada website.
10. Mudah untuk menemukan informasi yang saya butuhkan.
11. Informasi yang disediakan efektif untuk membantu saya dalam mencari kosan.
12. Informasi perusahaan ditampilkan dengan jelas pada layar website.
13. Tampilan antarmuka dari website menyenangkan.
14. Saya senang berinteraksi dengan antarmuka website.
15. Website ini memiliki fungsi dan kemampuan yang saya harapkan.
16. Secara keseluruhan saya puas dengan website ini.

PSSUQ menghasilkan empat penilaian, secara menyeluruh dan tiga sub skala. Penjabarannya adalah sebagai berikut :

a. *Overall*, rata-rata dari pertanyaan nomor 1 sampai 16.
b. *System Quality* (*SysQual*), rata-rata dari pertanyaan 1 sampai 6.
c. *Information Quality* (*InfoQual*), rata-rata dari pertanyaan 7 sampai 12.
d. *Interface Quality* (*IntQual*), rata-rata dari pertanyaan 13 sampai 16.

Rumus untuk menghitung rata-rata sub skala dapat dilakukan menggunakan rumus berikut ini :

$$Sub\ skala = \frac{\text{jumlah skor penilaian responden dari setiap sub skala}}{\text{jumlah nomor item pertanyaan setiap sub skala}} \quad (3)$$